User's Manual

32

RX74N Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX700 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.10 Jul 2026

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation.
All trademarks and registered trademarks are the property of their respective owners.

© 2026 Renesas Electronics Corporation. All rights reserved.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.
The manual comprises an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes can be found within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX74N Group. Make sure to refer to the latest versions of these documents. The latest versions of the listed documents are available from the Renesas Electronics website.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	RX74N Group Datasheet	R01DS0434EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RX74N Group User's Manual: Hardware	This User's manual
User's Manual: Software	Description of CPU instruction set	RX Family RXv3e Instruction Set Architecture User's Manual: Software	R01US0745EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Information on using peripheral functions and application examples Sample programs	Available from Renesas Electronics website.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X ... Register

Address(es): xxxx xxxxh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	...[1:0]		...4	—	—	—	...0
Value after reset:	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: Setting prohibited (3)	R/W (1)
b3 to b1	—	Reserved (2)	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: 1:	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... Settings other than above are prohibited. (3)	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved.
Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input / Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver / Transmitter
VCO	Voltage Controlled Oscillator

Contents

Features 110

1. Overview 111

1.1 Outline of Specifications 111

1.2 List of Products 123

1.3 Block Diagram 126

1.4 Pin Functions 127

1.5 Pin Assignments 134

1.5.1 320-Pin LFBGA 134

1.5.2 224-Pin LFBGA 135

1.5.3 176-Pin LFBGA 136

1.5.4 176-Pin LFQFP 137

1.5.5 144-Pin LFQFP 138

1.5.6 100-Pin LFQFP 139

1.6 List of Pin and Pin Functions 140

1.6.1 320-Pin LFBGA 140

1.6.2 224-Pin LFBGA 158

1.6.3 176-Pin LFBGA 173

1.6.4 176-Pin LFQFP 184

1.6.5 144-Pin LFQFP 195

1.6.6 100-Pin LFQFP 204

2. CPU 211

2.1 Features 211

2.2 Register Set of the CPU 212

2.2.1 General-Purpose Registers (R0 to R15) 213

2.2.2 Control Registers 213

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP) 214

2.2.2.2 Interrupt Table Register (INTB) 214

2.2.2.3 Program Counter (PC) 214

2.2.2.4 Processor Status Word (PSW) 215

2.2.2.5 Backup PC (BPC) 217

2.2.2.6 Backup PSW (BPSW) 218

2.2.2.7 Fast Interrupt Vector Register (FINTV) 218

2.2.2.8 Single-Precision Floating-Point Status Word (FPSW) 219

2.2.2.9 Exception Table Register (EXTB) 221

2.2.2.10 Interrupt Vector Control Register (IVC) 222

2.2.2.11 World ID Register (WID) 224

2.2.2.12 Root Call Vector Table Register (TRCTB) 225

2.2.2.13 Vector Table Register for Interrupts that Cause World Transitions (WINTB) 225

2.2.2.14 World-Shared Processor Interrupt Priority Level Register (WPRI) 226

2.2.3 Accumulator 227

2.3 Processor Mode 228

2.3.1 Supervisor Mode 228
2.3.2 User Mode 228
2.3.3 Privileged Instruction 228
2.3.4 Switching Between Processor Modes 229
2.4 Data Types 230
2.4.1 Integer 230
2.4.2 Single-Precision Floating-Point Numbers 231
2.4.3 Bitwise Operations 231
2.4.4 Strings 232
2.5 Endian 233
2.5.1 Switching the Endian 233
2.5.2 Access to I/O Registers 236
2.5.3 Notes on Access to I/O Registers 236
2.5.4 Data Arrangement 237
2.5.4.1 Data Arrangement in Registers 237
2.5.4.2 Data Arrangement in Memory 237
2.5.5 Notes on the Allocation of Instruction Codes 237
2.6 Vector Tables 238
2.6.1 Exception Vector Table 238
2.6.2 Interrupt Vector Table 239
2.6.3 Root Call Vector Table 240
2.6.4 Vector Table for Interrupts that Cause World Transitions 240
2.7 Register Bank Save Function 241
2.8 Double-Precision Floating-Point Coprocessor 242
2.8.1 Features 242
2.8.2 Double-Precision Floating-Point Register Set 243
2.8.2.1 Double-Precision Floating-Point Data Registers (DR0 to DR15) 243
2.8.2.2 Double-Precision Floating-Point Control Registers 243
2.8.3 Data Types (for the Double-Precision Floating-Point Coprocessor) 248
2.8.3.1 Double-Precision Floating-Point Numbers 248
2.8.4 Data Arrangement (for the Double-Precision Floating-Point Coprocessor) 248
2.8.4.1 Arrangement of Data in the Double-Precision Floating-Point Registers 248
2.8.4.2 Arrangement of Data for Double-Precision Floating-Point Numbers in Memory 249
2.8.5 Switching the Endian (for the Double-Precision Floating-Point Coprocessor) 250
2.9 Acceleration Functions 253
2.9.1 Features 253
2.9.2 Block Diagrams 254
2.9.3 Cache Control Registers 256
2.9.3.1 Cache Security Attribute Register (CACHESAR) 257
2.9.3.2 Instruction Cache Control Register (IFCACTL) 258
2.9.3.3 Instruction Cache Flush Control Register (IFCAFCT) 259

2.9.3.4 Instruction Cache Error Detection Status Register (IFCAEDST) 260
2.9.3.5 Instruction Cache Test Access Address Register (IFCATAA) 262
2.9.3.6 Instruction Cache Test Access Data Register (IFCATAD_DATA, IFCATAD_ECC, IFCATAD_TAG, IFCATAD_LRU, IFCATAD_TAGECC) 264
2.9.3.7 Data Cache Control Register (OACACTL) 267
2.9.3.8 Data Cache Flush Control Register (OACAFCT) 268
2.9.3.9 Data Cache Write Attribute Register (OACAWTA) 269
2.9.3.10 Data Cache Error Detection Status Register (OACAEDST) 270
2.9.3.11 Data Cache Test Access Address Register (OACATAA) 272
2.9.3.12 Data Cache Test Access Data Register (OACATAD_DATA, OACATAD_ECC, OACATAD_TAG, OACATAD_LRU, OACATAD_TAGECC) 274
2.9.3.13 Operation Settings After Cache Error Detection Register (CAOAD) 277
2.9.3.14 Cache Protection Register (CAPRCR) 278
2.9.4 Branch Instruction Processing Accelerator Registers 279
2.9.4.1 Branch Instruction Processing Accelerator Security Attribute Register (IFUSAR) 279
2.9.4.2 Branch Instruction Processing Accelerator Enable Register (CPUBAEN) 280
2.9.4.3 Branch Instruction Processing Accelerator Invalidation Setting Register (CPUBAINV) 281
2.9.5 Local SRAM (LRAM) Control Registers 282
2.9.5.1 LRAM Security Attribute Setting Register (LRMSAR) 282
2.9.5.2 LRAMn Boundary Address Setting Register (LRMSABARn) (n = 0, 1) 283
2.9.5.3 LRAM Protection Control Register for Secure Applications (LRMPRCR_S), LRAM Protection Control Register for Non-Secure Applications (LRMPRCR_NS) 285
2.9.5.4 LRAMn Control Register (LRMCRn) (n = 0, 1) 286
2.9.5.5 LRAM Error Flag Register (LRMESR) 287
2.9.5.6 LRAM Error Flag Clear Register (LRMESCLR) 288
2.9.5.7 LRAMn Error Address Register m (LRMEARnm) (n = 0, 1) (m = 1, 2) 289
2.9.6 Operations of Caches 291
2.9.6.1 Basic Operation of Caches 291
2.9.6.2 Cache Flush 291
2.9.6.3 LRU and Replace 291
2.9.6.4 Interrupt Source 291
2.9.6.5 Bus Error 292
2.9.7 Operations of the Branch Instruction Processing Accelerator 293
2.9.7.1 Judgment of Hit or Miss and Registration of Data 293
2.9.7.2 Algorithm for Replacement 293
2.9.7.3 Invalidation through the Branch Instruction Processing Accelerator Invalidation Setting Register 293
2.9.8 Operations of LRAM 294
2.9.8.1 Interrupt Source 294
2.9.8.2 Security Function 294
2.9.9 ECC 295

2.9.9.1 ECC Function ... 295
2.9.9.2 Methods of Testing the ECC Decoder Circuits for the Caches ... 296
2.9.9.3 Methods of Testing the ECC Decoder Circuits for the LRAM ... 300
2.10 CPU Control Registers ... 301
2.10.1 CPU Function Lock Control Register (CPULOCKCR) ... 301
2.10.2 CPU Non-secure Function Lock Control Register (CPULOCKCRNS) ... 302
2.11 Operation of Instructions ... 303
2.11.1 Restrictions on RMPA and String-Manipulation Instructions ... 303
2.11.1.1 Transfer Size and Data Prefetching ... 303
2.11.1.2 Access to the External Address Space ... 303
2.11.1.3 Access to I/O Registers ... 303
2.12 Numbers of Cycles ... 304
2.12.1 Instructions and Numbers of Cycles ... 304
2.12.2 Instructions and Numbers of Cycles (for the Double-Precision Floating-Point Coprocessor) ... 308
2.12.3 Numbers of Cycles for Response to Interrupts ... 309
2.12.4 Numbers of Cycles of Waiting ... 309
2.13 Usage Notes ... 310
2.13.1 Notes on Self-Diagnosis of the RAM for the Register-Saving Banks ... 310
2.13.2 Points to Note on Cache Operations ... 310
2.13.3 Note on Cache Coherency ... 311
2.13.4 Note on Cache Flush and Write-Back ... 311
2.13.5 Note on Testing LRAM Memory Cells ... 311
2.13.6 Notes on Modifying Instruction Codes ... 312
3. Operating Modes ... 313
3.1 Operating Mode Types and Selection ... 313
3.2 Register Descriptions ... 314
3.2.1 System Control Register 0 (SYSCR0) ... 314
3.2.2 Operating Mode Security Attribution Register (MDSAR) ... 314
3.3 Details of Operating Modes ... 315
3.3.1 Single-Chip Mode ... 315
3.3.2 Boot Mode (SCI Interface) ... 315
3.3.3 Boot Mode (USB Interface) ... 315
3.3.4 Boot Mode (JTAG/SWD Interface) ... 315
3.3.5 Boot Mode (FINE Interface) ... 315
3.4 Transitions of Operating Modes ... 316
3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pin ... 316
3.4.2 Operating Mode Transitions According to Register Setting ... 317
4. Address Space ... 318
4.1 Address Space ... 318
4.2 External Address Space ... 321

5. I/O Registers 322
5.1 I/O Register Addresses (Address Order) 324
6. Resets 335
6.1 Overview 335
6.2 Register Descriptions 346
6.2.1 Reset Security Attribute Register (RSTSAR) 346
6.2.2 Reset Status Register 0 (RSTSR0) 347
6.2.3 Reset Status Register 1 (RSTSR1) 350
6.2.4 Reset Status Register 2 (RSTSR2) 352
6.2.5 Reset Status Register 3 (RSTSR3) 353
6.2.6 System Reset Mask Control Register 0 (SYRSTMSK0) 354
6.2.7 System Reset Mask Control Register 2 (SYRSTMSK2) 355
6.2.8 Temperature Monitoring Reset Control Register (TEMPRCR) 356
6.2.9 Temperature Monitoring Reset Lock Register (TEMPRLR) 357
6.2.10 Reset Output Status Register 0 (ROUTSR0) 358
6.2.11 Reset Output Status Register 1 (ROUTSR1) 360
6.2.12 Reset Output Status Register 2 (ROUTSR2) 362
6.2.13 Software Reset Register (SWRR) 363
6.3 Operation 364
6.3.1 RES# Pin Reset 364
6.3.2 Power-On Reset and Voltage Monitoring 0 Reset 364
6.3.3 Voltage Monitoring 1, 2, 4 and 5 Resets 365
6.3.4 Core Voltage Monitoring Reset 369
6.3.5 Deep Software Standby Reset 369
6.3.6 Independent Watchdog Timer Reset 370
6.3.7 Watchdog Timer Reset 370
6.3.8 Software Reset 370
6.3.9 Bus Error Reset 370
6.3.10 Common Memory Error Reset 371
6.3.11 Local Memory Error Reset 371
6.3.12 Temperature Monitoring Reset 372
6.3.13 Determination of Cold/Warm Start 374
6.3.14 Determining the Source of Reset Generation 375
6.3.15 Reset Output Function 376
6.4 Protection from Reset Control by a Non-Secure Program 378
7. Option-Setting Memory (OFSM) 379
7.1 Overview 379
7.2 Register Descriptions 381
7.2.1 Secure Switchable OTP Memory Type Setting Register (SOTPMS_S), Non-Secure Switchable OTP Memory Type Setting Register (SOTPMS_NS) 381
7.2.2 Option Function Select Register 0 (OFS0) 382

7.2.3 Option Function Select Register 2 (OFS2) ... 386
7.2.4 Endian Select Register (MDE) ... 387
7.2.5 Bank Mode Select Register (DUALSEL) ... 388
7.2.6 Bank Select Register for Secure Application (BANKSEL_S),
Bank Select Register for Non-Secure Application (BANKSEL_NS) ... 389
7.2.7 BANKSEL Attribute Select Register (BANKSEL_SEL) ... 390
7.2.8 Block Select Register for Secure Application (BLSEL_S),
Block Select Register for Non-Secure Application (BLSEL_NS) ... 391
7.2.9 BLSEL Attribute Select Register (BLSEL_SEL) ... 392
7.2.10 Block Swap Setting Register n for Secure Application (BLKSWPn_S) (n = 0 to 3),
Block Swap Setting Register n for Non-Secure Application (BLKSWPn_NS) (n = 0 to 3) ... 393
7.2.11 Start-Up Area Setting Register (SAS) ... 395
7.2.12 Start-Up Area Setting Register 2 (SAS2) ... 396
7.2.13 Option Function Select Register 1 for Secure Application (OFS1_S),
Option Function Select Register 1 for Non-Secure Application (OFS1_NS) ... 397
7.2.14 OFS1 Attribute Select Register (OFS1_SEL) ... 399
7.2.15 MRAM Security Attribute Boundary Register (MRAMSABR) ... 400
7.2.16 Block Protect Register n for Secure Application (BPSn_S) (n = 0 to 15, 64 to 71),
Block Protect Register n for Non-Secure Application (BPSn_NS) (n = 0 to 15, 64 to 71) ... 402
7.2.17 Data MRAM Block Protect Register 0 for Secure Application (BPSD0_S),
Data MRAM Block Protect Register 0 for Non-Secure Application (BPSD0_NS) ... 404
7.2.18 Permanent Block Protect Register m for Secure Application (PBPSm_S)
(m = 0 to 3, 16, 17),
Permanent Block Protect Register m for Non-Secure Application (PBPSm_NS)
(m = 0 to 3, 16, 17) ... 406
7.2.19 Data MRAM Permanent Block Protect Register 0 for Secure Application (PBPSD0_S),
Data MRAM Permanent Block Protect Register 0 for Non-Secure Application
(PBPSD0_NS) ... 407
7.2.20 Secure MRAM Option Setting Memory Protect Register n (MOFSPSn_S) (n = 0 to 3) ... 408
7.2.21 Non-secure MRAM Option Setting Memory Protect Register 0 (MOFSPS0_NS) ... 408
7.2.22 Secure Switchable OTP Option Setting Memory Protect Register n (SOFSPSn_S)
(n = 0 to 3) ... 409
7.2.23 Non-secure Switchable OTP Option Setting Memory Protect Register 0 (SOFSPS0_NS) ... 409
7.2.24 FSBL Control Register 0 (FSBLCTRL0) ... 410
7.2.25 FSBL Control Register 1 (FSBLCTRL1) ... 411
7.2.26 FSBL Control Register 2 (FSBLCTRL2) ... 412
7.2.27 Code Certificate 0 Start Address Register n (SACC0n) (n = 0 to 3) ... 413
7.2.28 Code Certificate 1 Start Address Register n (SACC1n) (n = 0 to 3) ... 413
7.2.29 Measurement Report Start Address Register (SAMR) ... 414
7.2.30 OEM_ROOT_PK Revoke Register (REVOKE) ... 415
7.2.31 OEM_ROOT_PKn Hash Register [i] (HOEMRTPKn[i]) (n = 0 to 3; i = 0 to 11) ... 416
7.2.32 HUK Zeroization Register (ZHUK) ... 417
7.2.33 Anti-Rollback Counter Setting Register (ARCCS) ... 418
7.2.34 Anti-Rollback Counter for Secure Application [i] (ARCS[i]) (i = 0 to 31) ... 419

7.2.35 Anti-Rollback Counter n for Non-Secure Application [i] (ARCNSn[i]) (n = 0 to 3, i = 0 to 7) ... 419
7.2.36 Anti-Rollback Counter for Non-Secure Application [i] (ARCNS[i]) (i = 0 to 31) ... 420
7.2.37 Anti-Rollback Counter for OEM Boot Loader [i] (ARCOEMBL[i]) (i = 0, 1) ... 420
7.2.38 User Extra MRAM Configuration Register (UMRECR) ... 421
7.3 Setting the Option-Setting Memory ... 422
7.3.1 Allocation of Data in the Option-Setting Memory ... 422
7.3.2 Programming to Option-Setting Memory ... 422
7.3.3 Security Attributes of Option-Setting Memory ... 422
7.3.4 Registers with Selectable Security Attributes for Each Bit ... 423
7.3.5 Reflection Timing of the Setting Value ... 423
7.4 Usage Note ... 424
7.4.1 Set Value for Reserved Area ... 424
7.4.2 Notes on the First-Stage Boot Loader (FSBL) ... 424
8. Programmable Voltage Detection Circuit (PVD) ... 425
8.1 Overview ... 425
8.2 Register Descriptions ... 428
8.2.1 Programmable Voltage Detection Security Attribute Setting Register (PVDSAR) ... 428
8.2.2 Voltage Monitoring m Comparator Control Register (PVDmCMPCR) (m = 1, 2) ... 429
8.2.3 Voltage Monitoring n Comparator Control Register (PVDnCMPCR) (n = 4, 5) ... 430
8.2.4 Voltage Monitoring m Circuit Control Register 0 (PVDmCR0) (m = 1, 2) ... 431
8.2.5 Voltage Monitoring n Circuit Control Register 0 (PVDnCR0) (n = 4, 5) ... 433
8.2.6 Voltage Monitoring m Circuit Control Register 1 (PVDmCR1) (m = 1, 2) ... 434
8.2.7 Voltage Monitoring m Circuit Status Register (PVDmSR) (m = 1, 2) ... 435
8.2.8 Voltage Monitoring m Function Control Register (PVDmFCR) (m = 1, 2) ... 436
8.2.9 Voltage Monitoring n Function Control Register (PVDnFCR) (n = 4, 5) ... 437
8.2.10 Voltage Monitoring Lock Register (PVDLR) ... 438
8.3 VCC Input Voltage Monitoring ... 439
8.3.1 Monitoring Vdet0 ... 439
8.3.2 Monitoring Vdetm ... 439
8.4 Voltage Monitoring 0 Reset ... 440
8.5 Voltage Monitoring m Interrupt and Voltage Monitoring m Reset (m = 1, 2) ... 441
8.6 Voltage Monitoring n Reset (n = 4, 5) ... 445
8.7 Event Link Output ... 447
8.7.1 Interrupt Handling and Event Linking ... 447
9. Clock Generation Circuit ... 448
9.1 Overview ... 448
9.2 Register Descriptions ... 454
9.2.1 Clock Generation Function Security Attribute Register (CGFSAR) ... 454
9.2.2 System Clock Frequency Division Control Register (SCKDIVCR) ... 456
9.2.3 System Clock Frequency Division Control Register 2 (SCKDIVCR2) ... 459
9.2.4 System Register Access Status Register (SYRACCR) ... 460

9.2.5 System Clock Source Control Register (SCKSCR) 461
9.2.6 PLL Clock Control Register (PLLCCR) 462
9.2.7 PLL Clock Control Register 2 (PLLCCR2) 463
9.2.8 PLL Control Register (PLLCR) 464
9.2.9 PLL2 Clock Control Register (PLL2CCR) 465
9.2.10 PLL2 Clock Control Register 2 (PLL2CCR2) 466
9.2.11 PLL2 Control Register (PLL2CR) 467
9.2.12 External Bus Clock Control Register (BCKCR) 468
9.2.13 Main Clock Oscillator Control Register (MOSCCR) 469
9.2.14 Sub-Clock Oscillator Control Register (SOSCCR) 470
9.2.15 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 471
9.2.16 High-Speed On-Chip Oscillator Control Register (HOCOCR) 472
9.2.17 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2) 473
9.2.18 Middle-Speed On-Chip Oscillator Control Register (MOCOCR) 474
9.2.19 FLL Control Register 1 (FLLCR1) 475
9.2.20 FLL Control Register 2 (FLLCR2) 477
9.2.21 Oscillation Stabilization Flag Register (OSCSF) 478
9.2.22 Oscillator Monitor Register (OSCMONR) 480
9.2.23 Oscillation Stop Detection Control Register (OSTDCR) 481
9.2.24 Oscillation Stop Detection Status Register (OSTDSR) 482
9.2.25 Sub-Clock Oscillation Stop Detection Control Register (SOSTDCR) 483
9.2.26 Sub-Clock Oscillation Stop Detection Status Register (SOSTDSR) 484
9.2.27 Main Clock Oscillator Wait Control Register (MOSCWTCR) 485
9.2.28 Main Clock Oscillator Mode Oscillation Control Register (MOMCR) 486
9.2.29 Sub-Clock Oscillator Mode Control Register (SOMCR) 487
9.2.30 Clock Output Control Register (CKOCR) 488
9.2.31 External Bus Clock Output Control Register (EBCKOCR) 489
9.2.32 SDRAM Clock Output Control Register (SDCKOCR) 489
9.2.33 LOCO User Trimming Control Register (LOCOUTCR) 490
9.2.34 MOCO User Trimming Control Register (MOCOUTCR) 491
9.2.35 HOCO User Trimming Control Register (HOCOUTCR) 492
9.2.36 Main Clock Oscillator Standby Control Register (MOSCSCR) 493
9.2.37 High-Speed On-Chip Oscillator Standby Control Register (HOCOSCR) 494
9.2.38 Middle-Speed On-Chip Oscillator Standby Control Register (MOCOSCR) 495
9.2.39 USB Clock Frequency Division Control Register (USBCKDIVCR) 496
9.2.40 XSPI Clock Frequency Division Control Register (XSPICKDIVCR) 497
9.2.41 CANFD Core Clock Frequency Division Control Register (CANFDCKDIVCR) 498
9.2.42 USBHS-PHY Clock Frequency Division Control Register (USB60CKDIVCR) 499
9.2.43 I3C Clock Frequency Division Control Register (I3CCKDIVCR) 500
9.2.44 USB Clock Control Register (USBCKCR) 501
9.2.45 XSPI Clock Control Register (XSPICKCR) 503

9.2.46 CANFD Core Clock Control Register (CANFDCKCR) 505
9.2.47 USBHS-PHY Clock Control Register (USB60CKCR) 507
9.2.48 I3C Clock Control Register (I3CCKCR) 509
9.2.49 SCI Clock Frequency Division Control Register (SCICKDIVCR) 510
9.2.50 SPI Clock Frequency Division Control Register (SPICKDIVCR) 511
9.2.51 ADC Clock Frequency Division Control Register (ADCCKDIVCR) 512
9.2.52 GPT Clock Frequency Division Control Register (GPTCKDIVCR) 513
9.2.53 SCI Clock Control Register (SCICKCR) 514
9.2.54 SPI Clock Control Register (SPICKCR) 516
9.2.55 ADC Clock Control Register (ADCCKCR) 518
9.2.56 GPT Clock Control Register (GPTCKCR) 520
9.2.57 Asynchronous External Bus Clock Frequency Division Control Register (BCKADIVCR) 521
9.2.58 ESWM Clock Frequency Division Control Register (ESWCKDIVCR) 522
9.2.59 ESWM-PHY Clock Frequency Division Control Register (ESWPCKDIVCR) 523
9.2.60 Ethernet-PHY Clock Frequency Division Control Register (ETHPCKDIVCR) 524
9.2.61 Asynchronous External Bus Clock Control Register (BCKACR) 525
9.2.62 ESWM Clock Control Register (ESWCKCR) 527
9.2.63 ESWM-PHY Clock Control Register (ESWPCKCR) 529
9.2.64 Ethernet-PHY Clock Control Register (ETHPCKCR) 531
9.3 Main Clock Oscillator 533
9.3.1 Connecting a Resonator 533
9.3.2 Inputting an External Clock 534
9.3.3 Note on the External Clock Input 534
9.3.4 Notes on Oscillation Keeping in Software Standby Mode 534
9.4 Sub-Clock Oscillator 535
9.4.1 Connecting a 32.768-kHz Crystal Resonator 535
9.4.2 Handling of Pins when the Sub-Clock Oscillator is Not in Use 535
9.4.3 Input of an External Clock 535
9.5 Low-Speed On-Chip Oscillator (LOCO) 536
9.5.1 LOCO Operating Conditions 536
9.5.2 LOCO Stop Conditions 536
9.5.3 Restrictions on LOCO Operation and Stopping 537
9.6 Middle-Speed On-Chip Oscillator (MOCO) 538
9.6.1 MOCO Operating Conditions 538
9.6.2 MOCO Stop Conditions 538
9.6.3 Restrictions on MOCO Operation and Stopping 539
9.7 High-Speed On-Chip Oscillator (HOCO) 540
9.7.1 Frequency Locked Loop Function 540
9.7.2 Notes on Oscillation Keeping in Software Standby Mode 540
9.8 PLL Circuits 541
9.9 Oscillation Stop Detection Function 543

9.9.1 Main Clock Oscillation Stop Detection and Operation after Detection 543
9.9.2 Main Clock Oscillation Stop Detection Interrupts 545
9.9.3 Sub-Clock Oscillation Stop Detection and Operation after Detection 545
9.9.4 Sub-Clock Oscillation Stop Detection Interrupts 548
9.10 Internal Clocks 549
9.10.1 CPU Clock (CPUCLK0) 550
9.10.2 MRAM Code Bus Clock (MRICLK) 550
9.10.3 MRAM Peripheral Bus Clock (MRPCLK) 550
9.10.4 System Clock (ICLK) 551
9.10.5 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE) 552
9.10.6 External Bus Clock (BCLK, EBCLK) 552
9.10.7 SDRAM Clock (SDCLK) 552
9.10.8 SCI Clock (SCICLK) 553
9.10.9 SPI Clock (SPICLK) 553
9.10.10 XSPI Clock (XSPICLK, XSPIDIVCLK) 553
9.10.11 CANFD Core Clock (CANFDCLK) 554
9.10.12 USB Clock (USBCLK) 554
9.10.13 USBHS-PHY Clock (USB60CLK) 554
9.10.14 MOSC Clock for USBHS (USBMCLK) 555
9.10.15 ADC Clock (ADCCLK) 555
9.10.16 GPT Clock (GPTCLK) 555
9.10.17 I3C Clock (I3CCLK) 556
9.10.18 Asynchronous External Bus Clock (BCLKA, EBCLK) 556
9.10.19 ESWM Clock (ESWCLK) 556
9.10.20 ESWM-PHY Clock (ESWPHYCLK) 557
9.10.21 Ethernet-PHY Clock (ETHPHYCLK) 557
9.10.22 CANFD Main Clock (CANFDMCLK) 557
9.10.23 CAC Clock (CACCLK) 557
9.10.24 RTC Clock (RTCSCLK, RTCLCLK) 558
9.10.25 IWDT Clock (IWDTCLK) 558
9.10.26 PDMIF Clock (PDMIFCLK) 558
9.10.27 External Pin Output Clock (CLKOUT) 558
9.10.28 JTAG Clock 558
9.11 Clock Setting 559
9.11.1 Setting the System Clock 559
9.11.1.1 Initial Settings of the System Clock 559
9.11.1.2 Setting the System Clock to Increase the Frequency 560
9.11.1.3 Setting the System Clock to Decrease the Frequency 561
9.11.2 Setting Peripheral Module-Dedicated Clocks 562
9.11.2.1 Initial Settings of a Peripheral Module-Dedicated Clock 562
9.11.2.2 Changing the Clock Source of a Peripheral Module-Dedicated Clock 563

9.11.2.3 Changing the Division Ratio of a Peripheral Module-Dedicated Clock 564
9.11.3 Setting the Main Clock Oscillator 565
9.11.4 Setting the Sub-Clock Oscillator 565
9.11.5 Setting the High-Speed On-Chip Oscillator 566
9.11.5.1 Initial Settings of the HOCO Clock 566
9.11.5.2 Setting the HOCO by the FLL Function 567
9.11.6 Setting PLL Circuits 568
9.11.6.1 Initial Settings of a PLL Clock 568
9.11.6.2 Changing the PLL Settings 569
9.12 Usage Notes 570
9.12.1 Notes on the Clock Generation Circuit 570
9.12.2 Restrictions on a Resonator 570
9.12.3 Notes on Board Design 571
9.12.4 Notes on Resonator Connection Pins 571
9.12.5 Notes on Using the Sub-Clock Oscillator 571
9.12.6 Notes on the Reset Conditions of the SOSC and LOCO 572
10. Clock Frequency Accuracy Measurement Circuit (CAC) 573
10.1 Overview 573
10.2 Register Descriptions 575
10.2.1 CAC Control Register 0 (CACR0) 575
10.2.2 CAC Control Register 1 (CACR1) 576
10.2.3 CAC Control Register 2 (CACR2) 578
10.2.4 CAC Interrupt Request Enable Register (CAICR) 580
10.2.5 CAC Status Register (CASTR) 581
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 582
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 582
10.2.8 CAC Counter Buffer Register (CACNTBR) 582
10.3 Operation 583
10.3.1 Measuring Clock Frequency 583
10.3.2 Digital Filtering of Signals on the CACREF Pin 584
10.4 Interrupt Requests 584
10.5 Usage Notes 585
10.5.1 Module Stop Function Setting 585
11. Low Power Consumption 586
11.1 Overview 586
11.2 Register Descriptions 593
11.2.1 Low Power Consumption Mode Security Attribution Register (LPMSAR) 593
11.2.2 Power Gating Security Attribution Register (PGCSAR) 595
11.2.3 Deep Software Standby Interrupt Source Security Attribution Register (DPFSAR) 596
11.2.4 Deep Software Standby Interrupt Source Security Attribution Register 1 (DPFSAR1) 600
11.2.5 RAM Standby Control Security Attribution Register (RSCSAR) 602

11.2.6 Module Stop Control Register A (MSTPCRA) 604
11.2.7 Module Stop Control Register B (MSTPCRB) 605
11.2.8 Module Stop Control Register C (MSTPCRC) 607
11.2.9 Module Stop Control Register D (MSTPCRD) 609
11.2.10 Module Stop Control Register E (MSTPCRE) 610
11.2.11 Module Stop Control Register F (MSTPCRF) 611
11.2.12 Power Gating Shift Control Register (PGSCR) 612
11.2.13 Power Switch Control Start Time Control Register n (PSSTCRn) (n = 2, 4, 5) 612
11.2.14 ESWM Power Domain Control Register (PDCTRESWM) 613
11.2.15 SRAM Power Domain Standby Control Register 0 (PDRAMSCR0) 614
11.2.16 SRAM Power Domain Standby Control Register 2 (PDRAMSCR2) 616
11.2.17 Standby Control Register (SBYCR) 617
11.2.18 Software Standby Control Register 1 (SSCR1) 618
11.2.19 Low Power Consumption State Control Register (LPSCR) 619
11.2.20 Deep Software Standby Control Register (DPSBYCR) 621
11.2.21 Deep Software Standby Interrupt Enable Register 0 (DPSIER0) 622
11.2.22 Deep Software Standby Interrupt Enable Register 1 (DPSIER1) 623
11.2.23 Deep Software Standby Interrupt Enable Register 2 (DPSIER2) 624
11.2.24 Deep Software Standby Interrupt Enable Register 3 (DPSIER3) 625
11.2.25 Deep Software Standby Interrupt Enable Register 4 (DPSIER4) 626
11.2.26 Deep Software Standby Interrupt Enable Register 5 (DPSIER5) 627
11.2.27 Deep Software Standby Interrupt Flag Register 0 (DPSIFR0) 628
11.2.28 Deep Software Standby Interrupt Flag Register 1 (DPSIFR1) 629
11.2.29 Deep Software Standby Interrupt Flag Register 2 (DPSIFR2) 630
11.2.30 Deep Software Standby Interrupt Flag Register 3 (DPSIFR3) 632
11.2.31 Deep Software Standby Interrupt Flag Register 4 (DPSIFR4) 634
11.2.32 Deep Software Standby Interrupt Flag Register 5 (DPSIFR5) 635
11.2.33 Deep Software Standby Interrupt Edge Register 0 (DPSIEGR0) 636
11.2.34 Deep Software Standby Interrupt Edge Register 1 (DPSIEGR1) 637
11.2.35 Deep Software Standby Interrupt Edge Register 2 (DPSIEGR2) 638
11.2.36 Deep Software Standby Interrupt Edge Register 3 (DPSIEGR3) 639
11.2.37 Deep Software Standby Interrupt Edge Register 4 (DPSIEGR4) 640
11.2.38 PLL1-LDO Control Register (PLL1LDOCR) 641
11.2.39 PLL2-LDO Control Register (PLL2LDOCR) 642
11.2.40 HOCO/LDO Control Register (HOCOLDOCR) 643
11.2.41 Voltage Scaling Control Register (VSCR) 644
11.2.42 SSTBY Voltage Scaling Control Register (SVSCR) 645
11.2.43 LDO Efficiency Control Register (LDOECR) 646
11.2.44 DSTBY Condition Control Register (DCCR) 647
11.2.45 CPU Low Power Control Register (CLPCR) 648
11.3 Reducing Power Consumption by Switching Clock Signals 649

11.4 Module-Stop Function 649
11.5 Power Gating Control 650
11.5.1 Power Gating Control Using the Power Domain Control Register 650
11.6 Low Power Consumption Modes 651
11.6.1 CPU Sleep Mode/CPU Deep Sleep Mode 651
11.6.1.1 Transition to CPU Sleep Mode or CPU Deep Sleep Mode 651
11.6.1.2 Release from CPU Sleep Mode/CPU Deep Sleep Mode 652
11.6.2 Software Standby Mode 654
11.6.2.1 Transitioning to Software Standby Mode 654
11.6.2.2 Release from Software Standby Mode 656
11.6.2.3 Example of Software Standby Mode Application 657
11.6.3 Deep Software Standby Mode 658
11.6.3.1 Transition to Deep Software Standby Mode 658
11.6.3.2 Release from Deep Software Standby Mode 659
11.6.3.3 Pin States at the Time of Release from Deep Software Standby Mode 660
11.6.3.4 Example of Deep Software Standby Mode Application 660
11.6.3.5 Flowchart to Use Deep Software Standby Mode 662
11.7 Voltage Scaling Control 663
11.8 Usage Notes 665
11.8.1 Register Access 665
11.8.2 I/O Port States 665
11.8.3 Module-Stop State of DMAC and DTC 665
11.8.4 On-Chip Peripheral Module Interrupts 665
11.8.5 Input Buffer Control by the DIRQnE Bit (n = 0 to 31) 665
11.8.6 Timing of the WAIT Instruction 666
11.8.7 Writing to the WDT/IWDT Registers by the DMAC or DTC 666
11.8.8 Timing of Writing to the Module Stop Bits 666
11.8.9 Notes on Transition to and Return from Low Power Modes 666
12. Battery Backup Function (VBATTBa) 669
12.1 Overview 669
12.2 Register Descriptions 672
12.2.1 VBATT Security Attribution Register (VBSAR) 672
12.2.2 Backup Register Security Attribute Boundary Register (BKRSABR) 673
12.2.3 Battery Voltage Measurement Function Control Register (VBATTMCR) 674
12.2.4 Battery Backup Function Control Register 1 (VBATTCR1) 675
12.2.5 Battery Backup Function Control Register 2 (VBATTCR2) 676
12.2.6 Backup Domain Power Status Register (BKPSR) 677
12.2.7 Tamper Status Register (TAMPSR) 678
12.2.8 Tamper Control Register (TAMPCR) 679
12.2.9 Tamper Control Register 2 (TAMPCR2) 680
12.2.10 Time Capture Event Control Register (TCECR) 681

12.2.11 Tamper/RTCIC Input Control Register 1 (TAMPICR1) 682
12.2.12 Tamper/RTCIC Input Control Register 2 (TAMPICR2) 683
12.2.13 Tamper/RTCIC Input Control Register 3 (TAMPICR3) 684
12.2.14 Tamper/RTCIC Input Monitoring Register (TAMPIMR) 685
12.2.15 Backup Register Control Register (BKRCR) 686
12.2.16 Backup Register n (BKRn) (n = 0 to 127) 686
12.3 Operation 687
12.3.1 Battery Backup Function 687
12.3.2 Power-Down Detector 689
12.3.3 Backup Registers 689
12.3.4 Tamper Detector 690
12.4 Interrupt 692
12.5 Usage Notes 692
12.5.1 Voltage Monitor 0 Reset 692
12.5.2 When the Battery Backup Function is Not to be Used 692
12.5.3 Current Injection Flowing into the VBATT Pin 693
13. Register Write Protection Function 694
13.1 Register Descriptions 695
13.1.1 Protect Register (PRCR_S) 695
13.1.2 Protect Register (PRCR_NS) 696
14. Exception Handling 697
14.1 Exception Events 697
14.1.1 Undefined Instruction Exception 698
14.1.2 Privileged Instruction Exception 698
14.1.3 Access Exception 698
14.1.4 Secure Exception 698
14.1.5 Address Exception 698
14.1.6 Single-Precision Floating-Point Exception 698
14.1.7 Reset 698
14.1.8 Non-Maskable Interrupt 699
14.1.9 Interrupt 699
14.1.10 Unconditional Trap 700
14.2 Exception Handling Procedure 701
14.2.1 Exception Handling Procedure without World Transitions 702
14.2.2 Exception Handling Procedure with World Transitions 704
14.3 Acceptance of Exception Events 706
14.3.1 Timing of Acceptance and Saved PC Value 706
14.3.2 Vector and Location for Saving the PC and PSW 707
14.4 Hardware Processing for Accepting and Returning from Exceptions 708
14.4.1 Hardware Processing for Accepting and Returning from Exceptions Not Involving World Transitions 708

14.4.2 Hardware Processing for Accepting and Returning from Exceptions Involving World Transitions ... 709
14.5 Hardware Pre-Processing ... 710
14.5.1 Undefined Instruction Exception ... 710
14.5.2 Privileged Instruction Exception ... 710
14.5.3 Access Exception ... 710
14.5.4 Secure Exception ... 710
14.5.5 Address Exceptions ... 711
14.5.6 Single-Precision Floating-Point Exceptions ... 711
14.5.7 Reset ... 711
14.5.8 Non-Maskable Interrupts (with Normal Acceptance) ... 711
14.5.9 Non-Maskable Interrupts (with World-Transition Acceptance) ... 711
14.5.10 Interrupts (with Normal Acceptance) ... 712
14.5.11 Interrupts (with World-Transition Acceptance) ... 712
14.5.12 Unconditional Trap (with Normal Acceptance) ... 712
14.5.13 Unconditional Trap (with World-Transition Acceptance) ... 713
14.6 Return from Exception Handling Routine ... 714
14.7 Priority of Exception Events ... 714
14.8 Exception Generated in Coprocessor ... 715
14.8.1 Double-Precision Floating-Point Exceptions ... 715
15. Interrupt Controller (ICUM) ... 716
15.1 Overview ... 716
15.2 Register Descriptions ... 718
15.2.1 Interrupt Request Register n (IRn) (n = 16 to 255) ... 718
15.2.2 Interrupt Request Enable Register m (IERm) (m = 2 to 31) ... 720
15.2.3 Interrupt Source Priority Register r (IPRr) (r = 0 to 255) ... 721
15.2.4 Fast Interrupt Set Register (FIR) ... 722
15.2.5 Software Interrupt Generation Register (SWINTR) ... 723
15.2.6 Software Interrupt 2 Generation Register (SWINT2R) ... 723
15.2.7 DTC Transfer Request Enable Register n (DTCERn) (n = 26 to 255) ... 724
15.2.8 DMAC Trigger Select Register m (DMRSRm) (m = 0 to 7) ... 725
15.2.9 IRQi Pin Interrupt Control Register (IRQCRi) (i = 0 to 31) ... 726
15.2.10 Non-Maskable Interrupt Status Register (NMISR) ... 727
15.2.11 Non-Maskable Interrupt Enable Register (NMIER) ... 731
15.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR) ... 733
15.2.13 NMI Pin Interrupt Control Register (NMICR) ... 734
15.2.14 Group ILi Non-Secure Interrupt Request Register (GRPILi_NS) (i = 0, 1), Group BLi Non-Secure Interrupt Request Register (GRPBLi_NS) (i = 0, 1), Group ALi Non-Secure Interrupt Request Register (GRPALi_NS) (i = 0 to 6) ... 735
15.2.15 Group ILi Secure Interrupt Request Register (GRPILi_S) (i = 0, 1), Group BLi Secure Interrupt Request Register (GRPBLi_S) (i = 0, 1), Group ALi Secure Interrupt Request Register (GRPALi_S) (i = 0 to 6) ... 737

15.2.16 Group ILi Non-Secure Interrupt Request Enable Register (GENILi_NS) (i = 0, 1), Group BLi Non-Secure Interrupt Request Enable Register (GENBLi_NS) (i = 0, 1), Group ALi Non-Secure Interrupt Request Enable Register (GENALi_NS) (i = 0 to 6) 739
15.2.17 Group ILi Secure Interrupt Request Enable Register (GENILi_S) (i = 0, 1), Group BLi Secure Interrupt Request Enable Register (GENBLi_S) (i = 0, 1), Group ALi Secure Interrupt Request Enable Register (GENALi_S) (i = 0 to 6) 741
15.2.18 Software Configurable Interrupt Request Register Xk (PIXRk) (k = 0, 1) 743
15.2.19 Software Configurable Interrupt Request Register k (PIRk) (k = 0 to 127) 744
15.2.20 Software Configurable Interrupt Source Select Register Xn (SLIXRn) (n = 128 to 143) 745
15.2.21 Software Configurable Interrupt Source Select Register n (SLIRn) (n = 144 to 255) 746
15.2.22 EXDMAC Trigger Select Register n (SELEXDRn) (n = 0, 1) 757
15.2.23 Non-Secure Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR_NS) 758
15.2.24 Secure Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR_S) 759
15.2.25 ICU Security Attribution Register A (ICUSARA) 760
15.2.26 ICU Security Attribution Register B (ICUSARB) 761
15.2.27 ICU Security Attribution Register C (ICUSARC) 762
15.2.28 ICU Security Attribution Register D (ICUSARD) 763
15.2.29 ICU Security Attribution Register 0 (ICU0SAR0) 764
15.2.30 ICU Security Attribution Register 1 (ICU0SAR1) 766
15.2.31 ICU Security Attribution Register 2 (ICU0SAR2) 768
15.2.32 ICU Security Attribution Register 3 (ICU0SAR3) 770
15.2.33 ICU Security Attribution Register 4 (ICU0SAR4) 772
15.2.34 ICU Security Attribution Register 5 (ICU0SAR5) 774
15.2.35 ICU Security Attribution Register 6 (ICU0SAR6) 776
15.2.36 ICU Security Attribution Register 7 (ICU0SAR7) 778
15.2.37 Trusted Event Route Control Register (TEVTRCR) 779
15.2.38 ICU Non-Secure Access Control Register (ICUNSACR) 780
15.2.39 Protect Zone Interrupt Control Register (PRTZCR) 781
15.3 Vector Table 782
15.3.1 Interrupt Vector Table 782
15.3.2 Fast Interrupt Vector Area 790
15.3.3 Non-maskable Interrupt Vector Area 790
15.4 Types of Interrupts 791
15.4.1 Peripheral Interrupts 791
15.4.2 Software Interrupts 791
15.4.3 External Pin Interrupt 791
15.4.4 Group Interrupts 792
15.4.5 Software Configurable Interrupts 799
15.4.6 Non-Maskable Interrupts 800
15.5 Interrupt Detection 801
15.5.1 Edge Detection 801

15.5.2 Level Detection ... 803
15.5.3 Interrupt Detection for Group Interrupts ... 804
15.5.4 Interrupt Detection for Software Configurable Interrupts ... 806
15.6 Determining Priority of Interrupt Requests ... 806
15.7 Interrupt Setting Procedure ... 807
15.7.1 Enabling Interrupt Requests ... 807
15.7.2 Disabling Interrupt Requests ... 807
15.7.3 Selecting Interrupt Request Destination ... 807
15.7.3.1 Interrupt Request Destination Setting Procedure ... 807
15.7.3.2 Operations When the DTC/DMAC Selected ... 809
15.7.3.3 Changing the Interrupt Request Destination ... 809
15.7.4 Setting the External Pin Interrupt ... 810
15.7.5 Setting Non-Maskable Interrupts ... 810
15.7.6 Digital Noise Filter ... 811
15.7.7 Setting Software Configurable Interrupts ... 811
15.7.7.1 Polling for Software Configurable Interrupts ... 812
15.8 Multiple Interrupt ... 812
15.9 Fast interrupt ... 812
15.10 Exiting Low Power Consumption State ... 813
15.10.1 Exiting Sleep Mode or Deep Sleep Mode ... 813
15.10.2 Exiting Software Standby Mode ... 813
15.11 Usage Notes ... 814
15.11.1 Notes on the WAIT instruction When Using the Non-Maskable Interrupt ... 814
15.11.2 Interrupt Requests in Software Standby Mode ... 814
15.11.3 Notes on Bus Error during DTC/DMA Transfer ... 814
16. Buses ... 815
16.1 Overview ... 815
16.2 Description of Buses ... 819
16.2.1 Arbitration ... 819
16.2.2 External Bus ... 819
16.2.3 Parallel Operation ... 823
16.2.4 Bus Settings ... 823
16.2.5 Restrictions ... 824
16.2.5.1 Restriction on Endian ... 824
16.2.5.2 Bufferable Write ... 824
16.2.5.3 Notes on the Timing of Writing Completion ... 824
16.3 Register Descriptions ... 825
16.3.1 Bus Security Attribution Register A (BUSSARA) ... 825
16.3.2 Bus Security Attribution Register B (BUSSARB) ... 826
16.3.3 Bus Security Attribution Register C (BUSSARC) ... 827
16.3.4 CSn Control Register (CSnCR) (n = 0 to 7) ... 828

16.3.5 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7) 830
16.3.6 CS Recovery Cycle Insertion Enable Register (CSRECEN) 832
16.3.7 CSn Mode Register (CSnMOD) (n = 0 to 7) 834
16.3.8 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7) 836
16.3.9 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7) 838
16.3.10 SDC Control Register (SDCCR) 841
16.3.11 SDC Mode Register (SDCMOD) 842
16.3.12 SDRAM Access Mode Register (SDAMOD) 843
16.3.13 SDRAM Self-Refresh Control Register (SDSELF) 844
16.3.14 SDRAM Refresh Control Register (SDRFCR) 845
16.3.14.1 Auto-Refresh Request Interval and RFC Set Value 846
16.3.15 SDRAM Auto-Refresh Control Register (SDRFEN) 846
16.3.16 SDRAM Initialization Sequence Control Register (SDICR) 847
16.3.17 SDRAM Initialization Register (SDIR) 848
16.3.18 SDRAM Address Register (SDADR) 849
16.3.19 SDRAM Timing Register (SDTR) 850
16.3.20 SDRAM Mode Register (SDMOD) 852
16.3.21 SDRAM Status Register (SDSR) 853
16.3.22 EXBUS Error Status Register (EXBERR) 854
16.3.23 EXBUS Error Enable Register (EXBEREN) 855
16.3.24 EXBUS Error Status Clear Register (EXBECLR) 856
16.3.25 EXBUS Error Address Register (EXBEAD) 857
16.3.26 EXBUS Hold Enable Register (EXHLDEN) 857
16.3.27 Bus Slave Arbitration Control Register 0 (BUSSABT0MRE0BI, BUSSABT0S0BI, BUSSABT0S1BI, BUSSABT0ECBI, BUSSABT0XSPI0BI, BUSSABT0XSPI1BI, BUSSABT0PBBI, BUSSABT0PABI, BUSSABT0PIBI, BUSSABT0PSBI, BUSSABT0PS0BI) 858
16.3.28 Bus Slave Arbitration Control Register 1 (BUSSABT1MRC0BI, BUSSABT1MRC1BI) 860
16.3.29 Bus Error Status Register (BUSERRSTATCPU0IF, BUSERRSTATCPU0OA, BUSERRSTATDMAC0, BUSERRSTATEXDMAC, BUSERRSTATEDMAC, BUSERRSTATXSPI0, BUSERRSTATXSPI1) 861
16.3.30 Bus Error Clear Register (BUSERRCLRCPU0IF, BUSERRCLRCPU0OA, BUSERRCLRDMAC0, BUSERRCLREXDMAC, BUSERRCLREDMAC, BUSERRCLRXSPI0, BUSERRCLRXSPI1) 863
16.3.31 Bus Operation after Detection Register (BUSOAD) 864
16.3.32 Bus Operation after Detection Protect Register (BUSOADPT) 865
16.3.33 Master Bufferable Write Error Status Register (MBWERRSTAT) 866
16.3.34 Master Bufferable Write Error Clear Register (MBWERRCLR) 867
16.3.35 Bus Error Interrupt Enable Register (BUSINTENCPU0IF, BUSINTENCPU0OA, BUSINTENDMAC0, BUSINTENEXDMAC, BUSINTENEDMAC, BUSINTENXSPI0, BUSINTENXSPI1) 868
16.4 Endian and Data Alignment 869
16.4.1 Data Alignment Control for CS Area 869

16.4.2 Data Alignment Control for SDRAM Area 875
16.5 Operation of CS Area Controller 882
16.5.1 Separate Bus 882
16.5.2 Address/Data Multiplexed Bus 899
16.5.3 External Wait Function 901
16.5.4 Insertion of Recovery Cycles 905
16.5.5 No Access State 909
16.5.6 Limitations 909
16.6 SDRAM Area Controller Operation 911
16.6.1 Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width 911
16.6.2 No Access State 911
16.6.3 Insertion of Recovery Cycles 911
16.6.4 SDRAM Commands 912
16.6.5 Conditions for Setting SDRAMC Registers 912
16.6.6 Self-Refresh 913
16.6.7 Auto-Refresh 916
16.6.8 Initialization Sequencer 918
16.6.9 Setting Mode Register 919
16.6.10 Read/Write Access 920
16.6.11 SDRAMC Setting Examples 922
16.6.11.1 SDRAMC Access Procedure 922
16.6.11.2 Procedure for Transition to and Recovery from Self-Refresh Mode 923
16.6.11.3 Timing Register Settings and Access Timing 925
16.6.12 Address Multiplexing 934
16.6.13 Examples for Connecting with SDRAMs 935
16.6.13.1 32-Bit Bus Space 935
16.6.13.2 16-Bit Bus Space 938
16.6.14 Restrictions 941
16.7 External Bus Hold 941
16.8 Bus Error Monitoring Section 942
16.8.1 Types of Bus Error 942
16.8.2 Operations When a Bus Error Occurs 942
16.8.3 Conditions Leading to Illegal Address Access Errors 946
16.8.4 Timeout 951
16.8.4.1 Peripheral Bus 951
16.8.4.2 External Bus Access Wait with WAIT# Signal 951
16.8.4.3 External Bus Hold with BREQ# and BACK# Signals 951
17. Memory-Protection Unit (MPU) 952
17.1 Overview 952
17.2 CPU MPU 953
17.2.1 Overview of CPU MPU 953

17.2.1.1 Types of Access Control 954
17.2.1.2 Access Control Regions 954
17.2.1.3 Cache Attribute 954
17.2.1.4 Background Region 955
17.2.1.5 Overlap between Regions 955
17.2.1.6 Instructions and Data that Span Regions 955
17.2.2 Register Descriptions of CPU MPU 956
17.2.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7) 956
17.2.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7) 957
17.2.2.3 Memory-Protection Enable Register (MPEN) 958
17.2.2.4 Background Access Control Register (MPBAC) 959
17.2.2.5 Memory-Protection Error Status Clear Register (MPECLR) 960
17.2.2.6 Memory-Protection Error Status Register (MPESTS) 961
17.2.2.7 Data Memory-Protection Error Address Register (MPDEA) 962
17.2.2.8 Region Search Address Register (MPSA) 962
17.2.2.9 Region Search Operation Register (MPOPS) 963
17.2.2.10 Region Invalidation Operation Register (MPOPI) 964
17.2.2.11 Instruction-Hit Region Register (MHITI) 965
17.2.2.12 Data-Hit Region Register (MHITD) 967
17.2.3 Functions of CPU MPU 969
17.2.3.1 Memory Protection Function 969
17.2.3.2 Cache Attribute Setting Function 969
17.2.3.3 Region Search 969
17.2.3.4 Protection of Registers Related to the Memory-Protection Unit 969
17.2.3.5 Flow for Determining Access by the Memory Protection Function 970
17.2.3.6 Flow for Determining the Cache Attributes 972
17.2.4 Procedures for Using the Memory Protection Function 973
17.2.4.1 Setting the Access Control Information and Cache Attributes 973
17.2.4.2 Enabling the Memory Protection and Cache Attribute Setting Functions 973
17.2.4.3 Transition to User Mode 973
17.2.4.4 Modifying the Cache Attributes 974
17.2.4.5 Processing in Response to a Memory-Protection Error 975
17.3 Bus Master MPU 976
17.3.1 Overview of Bus Master MPU 976
17.3.2 Register Descriptions of Bus Master MPU 977
17.3.2.1 Master Memory-Protection Unit Security Attribute Register B (MMPUSARB) 977
17.3.2.2 MMPU Start Address Register for xSPI (MMPUSXSPImn) (m = 0, 1; n = 0 to 3) 978
17.3.2.3 MMPU End Address Register for xSPI (MMPUEXSPImn) (m = 0, 1; n = 0 to 3) 979
17.3.2.4 MMPU Access Control Register for xSPI (MMPUACXSPImn) (m = 0, 1, n = 0 to 3) 980

17.3.2.5 MMPU Enable Register for xSPI (MMPUENXSPIm) (m = 0, 1) 981
17.3.2.6 MMPU Enable Protect Register for xSPI (MMPUENPTXSPIm) (m = 0, 1) 982
17.3.2.7 MMPU Region Protect Register for xSPI (MMPURPTXSPIm) (m = 0, 1) 983
17.3.2.8 MMPU Post-Detection Operation Register (MMPUOAD) 984
17.3.2.9 MMPU Post-Detection Operation Protect Register (MMPUOADPT) 985
17.3.3 Functions of Bus Master MPU 986
17.3.3.1 Bus Master Memory Protection 986
17.3.3.2 Register Protection 988
17.3.3.3 Memory-Protection Error 988
18. DMA Controller (DMACAa) 989
18.1 Overview 989
18.2 Register Descriptions 991
18.2.1 DMAC Channel Security Attribution Register (DMACCHSAR) 991
18.2.2 DMA Controller Security Attribution Register (DMACSAR) 992
18.2.3 DMA Source Address Register (DMSAR) 993
18.2.4 DMA Source Reload Address Register (DMSRR) 994
18.2.5 DMA Destination Address Register (DMDAR) 995
18.2.6 DMA Destination Reload Address Register (DMDRR) 996
18.2.7 DMA Transfer Count Register (DMCRA) 997
18.2.8 DMA Block Transfer Count Register (DMCRB) 998
18.2.9 DMA Transfer Mode Register (DMTMD) 999
18.2.10 DMA Interrupt Setting Register (DMINT) 1000
18.2.11 DMA Address Mode Register (DMAMD) 1002
18.2.12 DMA Offset Register (DMOFR) 1005
18.2.13 DMA Transfer Enable Register (DMCNT) 1006
18.2.14 DMA Software Start Register (DMREQ) 1007
18.2.15 DMA Status Register (DMSTS) 1008
18.2.16 DMAC Trigger Source Flag Control Register (DMCSL) 1009
18.2.17 DMA Source Buffer Size Register (DMSBS) 1010
18.2.18 DMA Destination Buffer Size Register (DMDBS) 1011
18.2.19 DMA Bufferable Write Enable Register (DMBWR) 1012
18.2.20 DMAC Module Start Register (DMAST) 1013
18.2.21 DMAC Error Channel Register (DMECHR) 1014
18.2.22 DMAC74 Interrupt Status Monitor Register (DMIST) 1016
18.2.23 DMAC Control Register (DMCTL) 1017
18.3 Operation 1018
18.3.1 Transfer Mode 1018
18.3.2 Extended Repeat Area Function 1028
18.3.3 Free-Running Function 1030
18.3.4 Address Update Function using Offset 1032
18.3.5 Address-Update Function in Repeat-Block Transfer Mode 1036

18.3.6 Example of Using Repeat-Block Transfer Mode 1040
18.3.7 Trigger Sources 1042
18.3.8 Operation Timing 1043
18.3.9 Activating the DMAC 1044
18.3.10 Starting DMA Transfer 1046
18.3.11 Registers during DMA Transfer 1046
18.3.12 Channel Priority 1047
18.3.13 Channel Security 1047
18.4 Ending DMA Transfer 1048
18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 1048
18.4.2 Transfer End by Repeat Size End Interrupt 1048
18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 1049
18.5 Processing on DMA Transfer Error 1050
18.6 Interrupts 1051
18.6.1 Transfer End Interrupt 1051
18.6.2 Transfer Error Interrupt 1053
18.7 Event Link 1054
18.8 Low-Power Consumption Function 1055
18.9 Usage Notes 1056
18.9.1 DMA Transfer to Reserved Areas 1056
18.9.2 Setting of DMAC Trigger Select Register of the Interrupt Controller (ICU.DMRSRm) 1056
18.9.3 Suspending or Restarting DMA Transfer 1056
19. EXDMA Controller (EXDMACAa) 1057
19.1 Overview 1057
19.2 Register Descriptions 1059
19.2.1 EXDMAC Channel Security Attribution Register (EXDMACCHSAR) 1059
19.2.2 EXDMAC Controller Security Attribution Register (EXDMACSAR) 1060
19.2.3 EXDMA Source Address Register (EDMSAR) 1061
19.2.4 EXDMA Source Reload Address Register (EDMSRR) 1061
19.2.5 EXDMA Destination Address Register (EDMDAR) 1062
19.2.6 EXDMA Destination Reload Address Register (EDMDRR) 1062
19.2.7 EXDMA Transfer Count Register (EDMCRA) 1063
19.2.8 EXDMA Block Transfer Count Register (EDMCRB) 1064
19.2.9 EXDMA Transfer Mode Register (EDMTMD) 1065
19.2.10 EXDMA Interrupt Setting Register (EDMINT) 1066
19.2.11 EXDMA Address Mode Register (EDMAMD) 1068
19.2.12 EXDMA Offset Register (EDMOFR) 1071
19.2.13 EXDMA Transfer Enable Register (EDMCNT) 1072
19.2.14 EXDMA Software Start Register (EDMREQ) 1073
19.2.15 EXDMA Status Register (EDMSTS) 1074
19.2.16 EXDMA Source Buffer Size Register (EDMSBS) 1076

19.2.17 EXDMA Destination Buffer Size Register (EDMDBS) 1077
19.2.18 EXDMA Bufferable Write Enable Register (EDMBWR) 1078
19.2.19 EXDMAC Module Start Register (EDMAST) 1079
19.2.20 EXDMAC Error Channel Register (EDMECHR) 1080
19.2.21 EXDMAC Control Register (EDMCTL) 1082
19.2.22 EXDMAC Peripheral Module Request Flag Register (EDMPRF) 1083
19.3 Operation 1084
19.3.1 Transfer Mode 1084
19.3.2 Extended Repeat Area Function 1094
19.3.3 Free-Running Function 1096
19.3.4 Address Update Function using Offset 1098
19.3.5 Address Update Function in Repeat-Block Transfer Mode 1102
19.3.6 Trigger Sources 1106
19.3.7 Operation Timing 1107
19.3.8 Activating the EXDMAC 1108
19.3.9 Starting DMA Transfer 1110
19.3.10 Registers during DMA Transfer 1110
19.3.11 Channel Priority 1111
19.3.12 Channel Security 1111
19.4 Ending DMA Transfer 1112
19.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 1112
19.4.2 Transfer End by Repeat Size End Interrupt 1112
19.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 1113
19.5 Processing on DMA Transfer Error 1114
19.6 Interrupts 1115
19.6.1 Transfer End Interrupt 1115
19.6.2 Transfer Error Interrupt 1116
19.7 Low-Power Consumption Function 1117
19.8 Usage Notes 1118
19.8.1 Bus Response to DMA Write Access 1118
19.8.2 Access to the Registers during DMA Transfer 1118
19.8.3 DMA Transfer to Reserved Areas 1118
20. Data Transfer Controller (DTCA) 1119
20.1 Overview 1119
20.2 Register Descriptions 1121
20.2.1 DTC Mode Register A (MRA) 1121
20.2.2 DTC Mode Register B (MRB) 1123
20.2.3 DTC Mode Register C (MRC) 1125
20.2.4 DTC Transfer Source Register (SAR) 1126
20.2.5 DTC Transfer Destination Register (DAR) 1126
20.2.6 DTC Transfer Count Register A (CRA) 1127

20.2.7 DTC Transfer Count Register B (CRB) 1128
20.2.8 DTC Control Register for Secure Application (DTCCR_S), DTC Control Register for Non-secure Application (DTCCR_NS) 1128
20.2.9 DTC Vector Base Register for Secure Application (DTCVBR_S), DTC Vector Base Register for Non-secure Application (DTCVBR_NS) 1129
20.2.10 DTC Module Start Register (DTCST) 1130
20.2.11 DTC Status Register (DTCSTS) 1131
20.2.12 DTC Index Table Base Register (DTCIBR) 1132
20.2.13 DTC Operation Register (DTCOR) 1133
20.2.14 DTC Sequence Transfer Enable Register (DTCSQE) 1134
20.2.15 DTC Address Displacement Register (DTCDISP) 1134
20.2.16 DTC Error Vector Register (DTCEVR) 1135
20.2.17 DTC Security Attribution Register (DTCSAR) 1136
20.3 Request Sources 1137
20.3.1 Allocating Transfer Information and DTC Vector Table 1137
20.4 Operation 1139
20.4.1 Transfer Information Read Skip Function 1141
20.4.2 Transfer Information Write-Back Skip Function 1142
20.4.2.1 Write-Back Skip by Fixing Addresses 1142
20.4.2.2 Write-Back Skip by the MRA.WBDIS Bit 1142
20.4.3 Normal Transfer Mode 1143
20.4.4 Repeat Transfer Mode 1144
20.4.5 Block Transfer Mode 1145
20.4.6 Chain Transfer 1146
20.4.7 Operation Timing 1147
20.4.8 Sequence Transfer 1149
20.4.9 DTC Index Table 1151
20.4.10 Example of Sequence Transfer 1153
20.5 DTC Setting Procedure 1159
20.6 Examples of DTC Usage 1160
20.6.1 Normal Transfer 1160
20.6.2 Chain Transfer 1160
20.6.3 Chain Transfer When the Counter is 0 1162
20.6.4 Sequence Transfer 1163
20.7 Interrupt Source 1164
20.8 Event Link 1164
20.9 Low Power Consumption Function 1165
20.10 Usage Notes 1166
20.10.1 Start Address of Transfer Information 1166
20.10.2 Allocating Transfer Information 1166
20.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn) 1166

20.10.4 Notes on Using the Sequence Transfer 1166
21. Event Link Controller (ELC) 1167
21.1 Overview 1167
21.2 Register Descriptions 1168
21.2.1 ELC Security Attribution Register A (ELCSARA) 1168
21.2.2 ELC Security Attribution Register B (ELCSARB) 1169
21.2.3 ELC Security Attribution Register C (ELCSARC) 1170
21.2.4 ELC Security Attribution Register D (ELCSARD) 1171
21.2.5 Event Link Control Register (ELCR) 1172
21.2.6 Software Event Generation Register m (ELSEGRm) (m = 0 to 3) 1173
21.2.7 Event Link Setting Register n (ELSRn) (n = 0 to 26, 46, 47, 50 to 52, 73 to 78, 88 to 91) 1174
21.2.8 Event Link Option Setting Register A (ELOPA) 1185
21.2.9 Event Link Option Setting Register B (ELOPB) 1185
21.2.10 Event Link Option Setting Register C (ELOPC) 1186
21.2.11 Event Link Option Setting Register D (ELOPD) 1187
21.3 Operation 1188
21.3.1 Relation between Interrupt Handling and Event Linking 1188
21.3.2 Event Linkage 1189
21.3.3 Operation of Peripheral Timer Modules When Event Signal is Input 1190
21.3.4 Operation of GPTW When Event Signal is Input 1190
21.3.5 Operation of A/D and D/A Converters When Event Signal is Input 1190
21.3.6 Example of Procedure for Linking Events 1191
21.4 Usage Notes 1192
21.4.1 Setting ELSR88, ELSR89, ELSR90, and ELSR91 Registers 1192
21.4.2 Linking DMA/DTC Transfer End Signal as Event 1192
21.4.3 Clock Settings 1192
21.4.4 Module Stop Function Setting 1192
21.4.5 Internal Propagation Delay Times in the ELC 1193
22. I/O Ports 1194
22.1 Overview 1194
22.2 Configuration of I/O Ports 1197
22.3 Register Descriptions 1198
22.3.1 Port Control Register 1 (PCNTR1) 1198
22.3.2 Port Direction Register (PDR) 1200
22.3.3 Port Output Data Register (PODR) 1201
22.3.4 Port Control Register 2 (PCNTR2) 1202
22.3.5 Port Input Data Register (PIDR) 1203
22.3.6 Port Event Input Data Register (EIDR) 1204
22.3.7 Port Control Register 3 (PCNTR3) 1205
22.3.8 Port Output Set Register (POSR) 1207
22.3.9 Port Output Reset Register (PORR) 1208

22.3.10 Port Control Register 4 (PCNTR4) ... 1209
22.3.11 Event Output Set Register (EOSR) ... 1210
22.3.12 Event Output Reset Register (EORR) ... 1211
22.3.13 Pmn Pin Function Selection Register (PmnPFS) (m = 0 to 9, A to G; n = 00 to 15) ... 1212
22.3.14 Non-secure World Write Protection Register (PWPR_NS) ... 1241
22.3.15 Secure World Write Protection Register (PWPR_S) ... 1242
22.3.16 Port m Security Attribute Setting Register (PmSAR) (m = 0 to 9, A to G) ... 1243
22.3.17 Low-Voltage Operating Characteristics Control Register (LVOCR) ... 1244
22.4 Operation ... 1245
22.4.1 General-Purpose I/O Port ... 1245
22.4.2 Selecting Pin Functions ... 1245
22.4.3 Port Group Function of the ELC ... 1245
22.4.3.1 Operation when the ELC_PORTn (n = B to E) Signal is Input from the ELC ... 1245
22.4.3.2 Operation when an Event Pulse is Applied to the ELC ... 1247
22.5 Handling of Unused Pins ... 1248
22.6 Notes ... 1250
22.6.1 Procedure for Setting Pin Functions ... 1250
22.6.2 Procedure for Using the Event Output Function to ELC ... 1250
22.6.3 Method of Setting the Port Output Data Register (PODR) ... 1250
22.6.4 Note on Using Analog Functions ... 1251
22.6.5 Notes on Using the RTC Time-Capture Event Inputs and Tampering Detection ... 1251
22.6.6 Selecting the USB0_DP and USB0_DM Pins ... 1251
22.6.7 Pulling the PF00 and PF01 Pins Used for USB or General I/O Signals Up or Down ... 1252
22.6.8 Note on Using the NMI Pin ... 1252
22.6.9 Note on Using the Main Clock Oscillator ... 1252
22.6.10 Note on Using the Sub-Clock Oscillator ... 1252
22.6.11 Initialization of the Port Direction Register (PDR) ... 1253
23. Software Configurable 32-Bit Programmable I/O Port (PIO) ... 1256
23.1 Overview ... 1256
23.2 Register Descriptions ... 1257
23.2.1 PIO Operation Enable Register (PIOENR) ... 1257
23.2.2 PIO Port Input Enable Register (PIOIER) ... 1259
23.2.3 PIO Port Output Enable Register (PIOOER) ... 1260
23.2.4 PIO Port Input Data Register (PIOIDR) ... 1261
23.2.5 PIO Port Output Data Register (PIOODR) ... 1262
23.2.6 PIO Port Output Setting Register (PIOOSR) ... 1263
23.2.7 PIO Port Output Resetting Register (PIOORR) ... 1265
23.2.8 PIO Port Selection Register 0 (PIOSEL0) ... 1267
23.2.9 PIO Port Selection Register 1 (PIOSEL1) ... 1267
23.2.10 PIO Port Selection Register 2 (PIOSEL2) ... 1268

23.2.11 PIO Port Selection Register 3 (PIOSEL3) 1268
23.2.12 PIO Port Selection Register 4 (PIOSEL4) 1269
23.2.13 PIO Port Selection Register 5 (PIOSEL5) 1269
23.2.14 PIO Port Selection Register 6 (PIOSEL6) 1270
23.2.15 PIO Port Selection Register 7 (PIOSEL7) 1270
23.3 Operation 1271
23.3.1 Input 1271
23.3.2 Output 1271
23.4 Initial Settings 1272
24. General PWM Timer (GPTW) 1273
24.1 Overview 1273
24.2 Register Descriptions 1281
24.2.1 General PWM Timer Write-Protection Register (GTWP) 1281
24.2.2 General PWM Timer Software Start Register (GTSTR) 1284
24.2.3 General PWM Timer Software Stop Register (GTSTP) 1285
24.2.4 General PWM Timer Software Clear Register (GTCLR) 1287
24.2.5 General PWM Timer Start Source Select Register (GTSSR) 1288
24.2.6 General PWM Timer Stop Source Select Register (GTPSR) 1290
24.2.7 General PWM Timer Clear Source Select Register (GTCSR) 1292
24.2.8 General PWM Timer Count-Up Source Select Register (GTUPSR) 1296
24.2.9 General PWM Timer Count-Down Source Select Register (GTDNSR) 1299
24.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR) 1302
24.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR) 1305
24.2.12 General PWM Timer Switch Source Select Register (GTSWSR) 1308
24.2.13 General PWM Timer Switch Offset Setting Register (GTSWOS) 1310
24.2.14 General PWM Timer Control Register (GTCR) 1311
24.2.15 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC) 1318
24.2.16 General PWM Timer I/O Control Register (GTIOR) 1321
24.2.17 General PWM Timer Interrupt Output Setting Register (GTINTAD) 1326
24.2.18 General PWM Timer Status Register (GTST) 1330
24.2.19 General PWM Timer Buffer Enable Register (GTBER) 1336
24.2.20 General PWM Timer Buffer Enable Register 2 (GTBER2) 1340
24.2.21 General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register (GTITC) 1345
24.2.22 General PWM Timer Counter (GTCNT) 1347
24.2.23 General PWM Timer Compare Capture Register m (GTCCRm) (m = A to F) 1348
24.2.24 General PWM Timer Period Setting Register (GTPR) 1349
24.2.25 General PWM Timer Period Setting Buffer Register (GTPBR) 1349
24.2.26 General PWM Timer Period Setting Double-Buffer Register (GTPDBR) 1350
24.2.27 A/D Conversion Start Request Timing Register m (GTADTRm) (m = A, B) 1351
24.2.28 A/D Conversion Start Request Timing Buffer Register m (GTADTBRm) (m = A, B) 1352

24.2.29 A/D Conversion Start Request Timing Double-Buffer Register m (GTADTDBRm) (m = A, B) ... 1353
24.2.30 General PWM Timer Dead Time Control Register (GTDTCR) ... 1354
24.2.31 General PWM Timer Dead Time Value Register m (GTDVm) (m = U, D) ... 1356
24.2.32 General PWM Timer Dead Time Value Buffer Register m (GTDBm) (m = U, D) ... 1357
24.2.33 General PWM Timer Output Protection Function Status Register (GTSOS) ... 1358
24.2.34 General PWM Timer Output Protection Function Temporary Release Register (GTSOTR) ... 1359
24.2.35 General PWM Timer A/D Conversion Start Request Signal Monitoring Register (GTADSMR) ... 1360
24.2.36 General PWM Timer Extended Interrupt Skipping Counter Control Register (GTEITC) ... 1362
24.2.37 General PWM Timer Extended Interrupt Skipping Setting Register 1 (GTEITLI1) ... 1365
24.2.38 General PWM Timer Extended Interrupt Skipping Setting Register 2 (GTEITLI2) ... 1367
24.2.39 General PWM Timer Extended Buffer Transfer Skipping Setting Register (GTEITLB) ... 1369
24.2.40 General PWM Timer Inter Channel Logical Operation Function Setting Register (GTICLF) ... 1372
24.2.41 General PWM Timer Cycle Count Register (GTPC) ... 1374
24.2.42 General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register (GTADCMSC) ... 1376
24.2.43 General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register (GTADCMSS) ... 1378
24.2.44 General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register (GTSECSR) ... 1381
24.2.45 General PWM Timer Operation Enable Bit Simultaneous Control Register (GTSECR) ... 1383
24.2.46 General PWM Timer Output Level Buffer Register (GTOLBR) ... 1386
24.2.47 General PWM Timer Inter Channel Cooperation Input Capture Control Register (GTICCR) ... 1387
24.2.48 General PWM Timer-linked Synchronization Control Register (GTSYNCFCR) ... 1391
24.2.49 General PWM Timer-linked Synchronization Signal Output Control Register (GTSYNCSCR) ... 1392
24.2.50 Output Phase Switching Control Register (OPSCR) ... 1393
24.2.51 General PWM Timer Clock Control Register (GTCLKCR) ... 1396
24.3 Operation ... 1397
24.3.1 Basic Operation ... 1397
24.3.1.1 Counter Operation ... 1397
24.3.1.2 Waveform Output by Compare Match ... 1406
24.3.1.3 Input Capture Function ... 1410
24.3.2 Buffer Operation ... 1415
24.3.2.1 GTPR Register Buffer Operation ... 1415
24.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers ... 1421
24.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers ... 1427
24.3.2.4 Buffer Operation for the GTIOA[4:0] and GTIOB[4:0] Bits ... 1431
24.3.3 PWM Output Operating Mode ... 1434
24.3.3.1 Sawtooth-Wave PWM Mode 1 ... 1434

24.3.3.2 Sawtooth-Wave PWM Mode 2 ... 1437
24.3.3.3 Sawtooth-Wave One-Shot Pulse Mode ... 1441
24.3.3.4 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough) ... 1444
24.3.3.5 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough) ... 1446
24.3.3.6 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough) ... 1448
24.3.3.7 Complementary PWM mode 1, 2, 3 ... 1451
24.3.3.8 Complementary PWM Mode 4 ... 1475
24.3.4 Automatic Dead Time Setting Function ... 1490
24.3.5 Count Direction Changing Function ... 1495
24.3.6 Duty Cycle 0%/100% Output Function ... 1496
24.3.7 Hardware Count Start/Count Stop and Clear Operation ... 1500
24.3.7.1 Hardware Start Operation ... 1500
24.3.7.2 Hardware Stop Operation ... 1504
24.3.7.3 Hardware Clear Operation ... 1509
24.3.8 Hardware Count Switch Operation ... 1515
24.3.9 Inter-Channel Synchronous Operation ... 1525
24.3.9.1 Software Control of Inter-Channel Synchronous Operation ... 1525
24.3.9.2 Hardware Source Control of Inter-Channel Synchronous Operation ... 1529
24.3.9.3 Synchronous Clear Operation by Inter Channel Cooperation ... 1533
24.3.9.4 Input Capture Operation by Inter Channel Cooperation ... 1537
24.3.10 PWM Output Operation Examples ... 1539
24.3.11 Cycle Count Function ... 1545
24.3.12 Phase Counting Mode ... 1546
24.3.13 External Pulse Width Measuring Function ... 1554
24.3.14 Output Phase Switching Control Function (OPS) ... 1556
24.3.14.1 Input Selection and Sampling ... 1558
24.3.14.2 Rotation Direction Control ... 1559
24.3.14.3 Input Phase Decode ... 1559
24.3.14.4 Output Selection Control ... 1560
24.3.14.5 Output Selection Control (Group Output Disable Function) ... 1561
24.3.14.6 Event Link Controller (ELC) Output ... 1561
24.3.14.7 OPS Start Operation Setting Flow ... 1561
24.3.15 Inter Channel Logical Operation Function ... 1562
24.3.16 GTIOCnA/GTIOCnB Input/Output Pin Polarity Reversal Function ... 1564
24.3.17 Timer-Linked Synchronization Signal Output Function ... 1565
24.4 Interrupt Sources ... 1570
24.4.1 Interrupt Sources and Priorities ... 1570
24.4.2 DMAC/DTC Activation ... 1574
24.4.3 Interrupt and A/D Conversion Start Request Skipping Function ... 1574
24.4.3.1 Interrupt Skipping Function by GTITC Register ... 1574
24.4.3.2 Extended Interrupt Skipping Function ... 1578

24.4.3.3 A/D conversion Start Request Compare Match Skipping Function 1590
24.5 A/D Conversion Start Request 1595
24.6 Operations Linked by the ELC 1599
24.6.1 Event Signal Output to the ELC 1599
24.6.2 GPTW Operations in Response to Receiving Event Signals from the ELC 1599
24.7 Noise Filter Function 1600
24.8 Protection Function 1601
24.8.1 Write-Protection for Registers 1601
24.8.2 Disabling of Buffer Operation 1601
24.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels 1603
24.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer 1606
24.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 15; m = A, B) 1612
24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 15; m = A, B) 1613
24.9 Initialization Method of Output Pins 1619
24.9.1 Pin Settings after Reset 1619
24.9.2 Pin Initialization Due to Error during Operation 1619
24.10 Usage Notes 1620
24.10.1 Module Stop Function Setting 1620
24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F) 1620
24.10.3 Range of Settings for the GTPBR and GTPDBR Registers in the Complementary PWM Mode 1621
24.10.4 Setting Range of the GTCNT Counter 1621
24.10.5 The GTCNT Counter Start/Stop 1622
24.10.6 Order of Priority in Events 1622
24.10.7 Note on Counter Clearing in the Complementary PWM Mode 1623
24.10.8 Note on Disabling PWM Initial Output After Synchronous Clearing in the Complementary PWM Mode 1623
24.10.9 Correspondence to Count Switch Operation of Each Function 1623
24.10.10 Note on Transitions to Software Standby Mode 1625
24.10.11 Interval between Interrupt Requests 1625
25. High Resolution PWM Waveform Generation Circuit (HRPWMb) 1626
25.1 Overview 1626
25.2 Register Descriptions 1628
25.2.1 HRPWM Operation Control Register (HROCR) 1628
25.2.2 HRPWM Operation Control Register 2 (HROCR2) 1629
25.2.3 GTIOCnA Pin Rising Edge Adjustment Register (HRREARnA) (n = 0 to 3) 1630
25.2.4 GTIOCnA Pin Falling Edge Adjustment Register (HRFEARnA) (n = 0 to 3) 1631
25.2.5 GTIOCnB Pin Rising Edge Adjustment Register (HRREARnB) (n = 0 to 3) 1632
25.2.6 GTIOCnB Pin Falling Edge Adjustment Register (HRFEARnB) (n = 0 to 3) 1633
25.3 Operation 1634
25.3.1 Adjustment to the Timing of Rising and Falling Edges in PWM Waveforms 1634

25.3.2 Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3) ... 1636
25.4 Usage Notes ... 1638
25.4.1 Settings for the Module-Stop Function ... 1638
25.4.2 Note on Selecting Timer Prescaler in the GTCNT Counter While Using the HRPWM ... 1638
25.4.3 Notes on Delay Settings for HRPWM ... 1638
26. GPTW Port Output Enable (POEGd) ... 1639
26.1 Overview ... 1639
26.2 Register Descriptions ... 1642
26.2.1 POEG Group x Setting Register (POEGGx) (x = A to D) ... 1642
26.2.2 POEGx Pin Input Control Register (POEGPICRx) (x = A to D) ... 1644
26.3 Operation ... 1646
26.3.1 Output Disabling Request by Detection of Input Level on the GTETRGx Pin (x = A to D) ... 1646
26.3.1.1 Digital Noise Filter ... 1646
26.3.2 Output Disabling Request by Detection of Input Level or Input Edge on the POEGx Pin (x = A to D) ... 1647
26.3.2.1 Digital Noise Filter ... 1647
26.3.3 Output Disabling Request by Output Disabling Condition Detection Signal from GPTW ... 1647
26.3.4 Output Disabling Request by Oscillation Stop Detection Signal ... 1648
26.3.5 Output Disabling Request by a Register ... 1648
26.3.6 Canceling Output Disabling Request ... 1649
26.3.6.1 Cancellation of Output Disabling Request by Reset ... 1649
26.3.6.2 Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register ... 1649
26.4 Interrupt Sources ... 1650
26.5 External Trigger Output to GPTW ... 1651
26.6 Usage Notes ... 1651
26.6.1 Transitions to Low-Power Consumption Mode ... 1651
26.6.2 Setting the Function for Stopping the Module ... 1651
26.6.3 Duplication of Output Disabling Request ... 1651
27. 8-Bit Timer (TMRa) ... 1652
27.1 Overview ... 1652
27.2 Register Descriptions ... 1657
27.2.1 Timer Counter (TCNT) ... 1657
27.2.2 Time Constant Register A (TCORA) ... 1658
27.2.3 Time Constant Register B (TCORB) ... 1658
27.2.4 Timer Control Register (TCR) ... 1659
27.2.5 Timer Counter Control Register (TCCR) ... 1660
27.2.6 Timer Control/Status Register (TCSR) ... 1662
27.2.7 Timer Counter Start Register (TCSTR) ... 1664
27.3 Operation ... 1665

27.3.1 Pulse Output 1665
27.3.2 External Counter Reset Input 1666
27.4 Operation Timing 1667
27.4.1 TCNT Count Timing 1667
27.4.2 Timing of Interrupt Signal Output on a Compare Match 1668
27.4.3 Timing of Timer Output Signal at Compare Match 1668
27.4.4 Timing of Counter Clear by Compare Match 1669
27.4.5 Timing of the External Reset for TCNT 1669
27.4.6 Timing of Interrupt Signal Output on an Overflow 1670
27.5 Operation with Cascaded Connection 1671
27.5.1 16-Bit Count Mode 1671
27.5.2 Compare Match Count Mode 1671
27.6 Interrupt Sources 1672
27.6.1 Interrupt Sources and DTC Activation 1672
27.7 Link Operation by ELC 1673
27.7.1 Event Signal Output to ELC 1673
27.7.2 TMR Operation when Receiving an Event Signal from ELC 1673
27.7.3 Notes on Operating TMR According to an Event Signal from ELC 1674
27.8 Usage Notes 1675
27.8.1 Module Stop State Setting 1675
27.8.2 Notes on Setting Cycle 1675
27.8.3 Conflict between TCNT Write and Counter Clear 1675
27.8.4 Conflict between TCNT Write and Increment 1676
27.8.5 Conflict between TCORA or TCORB Write and Compare Match 1676
27.8.6 Conflict between Compare Matches A and B 1677
27.8.7 Switching of Internal Clocks and TCNT Operation 1677
27.8.8 Clock Source Setting with Cascaded Connection 1679
27.8.9 Continuous Output of Compare Match Interrupt Signal 1679
27.8.10 Transition to Software Standby Mode 1679
28. Compare Match Timer (CMT) 1680
28.1 Overview 1680
28.2 Register Descriptions 1682
28.2.1 Compare Match Timer Start Register 0 (CMSTR0) 1682
28.2.2 Compare Match Timer Start Register 1 (CMSTR1) 1682
28.2.3 Compare Match Timer Control Register (CMCR) 1683
28.2.4 Compare Match Counter (CMCNT) 1684
28.2.5 Compare Match Constant Register (CMCOR) 1684
28.3 Operation 1685
28.3.1 Cyclic Count Operation 1685
28.3.2 CMCNT Count Timing 1685
28.4 Interrupts 1686

28.4.1 Interrupt Sources ... 1686
28.4.2 Timing of Compare Match Interrupt Generation ... 1686
28.5 Link Operations by ELC ... 1687
28.5.1 Event Signal Output to ELC ... 1687
28.5.2 CMT Operation When Receiving an Event Signal from ELC ... 1687
28.5.3 Notes on Operating CMT According to an Event Signal from ELC ... 1687
28.6 Usage Notes ... 1688
28.6.1 Setting the Module Stop Function ... 1688
28.6.2 Conflict between CMCNT Counter Writing and Compare Match ... 1688
28.6.3 Conflict between CMCNT Counter Writing and Incrementing ... 1688
28.6.4 Transition to Software Standby Mode ... 1689
29. Compare Match Timer W (CMTW) ... 1690
29.1 Overview ... 1690
29.2 Register Descriptions ... 1693
29.2.1 Timer Start Register (CMWSTR) ... 1693
29.2.2 Timer Control Register (CMWCR) ... 1694
29.2.3 Timer I/O Control Register (CMWIOR) ... 1696
29.2.4 Timer Counter (CMWCNT) ... 1697
29.2.5 Compare Match Constant Register (CMWCOR) ... 1697
29.2.6 Input Capture Register n (CMWICRn) (n = 0, 1) ... 1698
29.2.7 Output Compare Register n (CMWOCRn) (n = 0, 1) ... 1698
29.3 Operation ... 1699
29.3.1 Cyclic Count Operation ... 1699
29.3.2 Compare Match Function ... 1699
29.3.3 Output Compare Function ... 1701
29.3.4 Input Capture Function ... 1702
29.3.5 Counter Size ... 1703
29.3.6 Count Timing of CMWCNT Counter ... 1703
29.3.7 Output Compare Output Timing ... 1703
29.3.8 Input Capture Timing ... 1704
29.4 Interrupts ... 1705
29.4.1 CMTW Interrupt Sources and DTC/DMAC Transfer Requests ... 1705
29.4.2 Timing of Compare Match Interrupt Generation ... 1706
29.5 Link Operations by ELC ... 1708
29.5.1 Event Signal Output to ELC ... 1708
29.5.2 CMTW Operation When Receiving an Event Signal from ELC ... 1709
29.5.3 Conflict between Event Link Operation and Register Access ... 1711
29.6 Usage Notes ... 1713
29.6.1 Setting the Module Stop Function ... 1713
29.6.2 Conflict between CMWCNT Counter Writing and Compare Match ... 1713
29.6.3 Conflict between CMWCNT Counter Writing and Incrementing or Clearing ... 1714

29.6.4 Conflict between CMWCOR Register Writing and Compare Match 1714
29.6.5 Conflict between CMWOCRn Register Writing and Compare Match (n = 0, 1) 1715
29.6.6 Conflict between CMWCNT Counter Reading and Incrementing or Clearing 1715
29.6.7 Conflict between CMWICRn Register Reading and Input Capture (n = 0, 1) 1716
29.6.8 Transition to Software Standby Mode 1716
30. Realtime Clock (RTCBb) 1717
30.1 Overview 1717
30.2 Register Descriptions 1719
30.2.1 64-Hz Counter (R64CNT) 1719
30.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) 1720
30.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) 1721
30.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) 1722
30.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) 1723
30.2.6 Date Counter (RDAYCNT) 1724
30.2.7 Month Counter (RMONCNT) 1725
30.2.8 Year Counter (RYRCNT) 1726
30.2.9 Second Alarm Register (RSECAR)/ Binary Counter 0 Alarm Register (BCNT0AR) 1727
30.2.10 Minute Alarm Register (RMINAR)/ Binary Counter 1 Alarm Register (BCNT1AR) 1728
30.2.11 Hour Alarm Register (RHRAR)/ Binary Counter 2 Alarm Register (BCNT2AR) 1729
30.2.12 Day-of-Week Alarm Register (RWKAR)/ Binary Counter 3 Alarm Register (BCNT3AR) 1730
30.2.13 Date Alarm Register (RDAYAR)/ Binary Counter 0 Alarm Enable Register (BCNT0AER) 1731
30.2.14 Month Alarm Register (RMONAR)/ Binary Counter 1 Alarm Enable Register (BCNT1AER) 1732
30.2.15 Year Alarm Register (RYRAR)/ Binary Counter 2 Alarm Enable Register (BCNT2AER) 1733
30.2.16 Year Alarm Enable Register (RYRAREN)/ Binary Counter 3 Alarm Enable Register (BCNT3AER) 1734
30.2.17 RTC Control Register 1 (RCR1) 1735
30.2.18 RTC Control Register 2 (RCR2) 1737
30.2.19 RTC Control Register 4 (RCR4) 1739
30.2.20 Frequency Register H/L (RFRH/RFRL) 1739
30.2.21 Time Error Adjustment Register (RADJ) 1741
30.2.22 Time Capture Control Register n (RTCCRn) (n = 0 to 2) 1742
30.2.23 Second Capture Register n (RSECCPn) (n = 0 to 2)/ BCNT0 Capture Register n (BCNT0CPn) (n = 0 to 2) 1744
30.2.24 Minute Capture Register n (RMINCPn) (n = 0 to 2)/ BCNT1 Capture Register n (BCNT1CPn) (n = 0 to 2) 1745
30.2.25 Hour Capture Register n (RHRCPn) (n = 0 to 2)/ BCNT2 Capture Register n (BCNT2CPn) (n = 0 to 2) 1746

30.2.26 Date Capture Register n (RDAYCPn) (n = 0 to 2)/ BCNT3 Capture Register n (BCNT3CPn) (n = 0 to 2) ... 1747
30.2.27 Month Capture Register n (RMONCPn) (n = 0 to 2) ... 1748
30.3 Operation ... 1749
30.3.1 Outline of Initial Settings of Registers after Power On ... 1749
30.3.2 Clock and Count Mode Setting Procedure ... 1750
30.3.3 Setting the Time ... 1753
30.3.4 30-Second Adjustment ... 1755
30.3.5 Reading 64-Hz Counter and Time ... 1756
30.3.6 Alarm Function ... 1757
30.3.7 Procedure for Disabling Alarm Interrupt ... 1758
30.3.8 Time Error Adjustment Function ... 1758
30.3.8.1 Automatic Adjustment ... 1758
30.3.8.2 Adjustment by Software ... 1759
30.3.8.3 Procedure for Changing the Mode of Adjustment ... 1760
30.3.8.4 Procedure for Stopping Adjustment ... 1760
30.3.9 Time Capture Function ... 1761
30.4 Interrupt Sources ... 1762
30.5 Event Link Output ... 1764
30.5.1 Interrupt Handling and Event Linking ... 1764
30.6 Usage Notes ... 1765
30.6.1 Register Writing during Counting ... 1765
30.6.2 Use of Periodic Interrupts ... 1765
30.6.3 RTCOUT (1-Hz/64-Hz) Clock Output ... 1765
30.6.4 Transitions to Low Power Consumption Modes after Setting Registers ... 1766
30.6.5 Notes on Writing to and Reading from Registers ... 1766
30.6.6 Changing the Count Mode ... 1766
30.6.7 Notes on Using the Time Capture Function ... 1766
30.6.8 Initialization Procedure When the Realtime Clock is Not to be Used ... 1767
31. Watchdog Timer (WDTA) ... 1769
31.1 Overview ... 1769
31.2 Register Descriptions ... 1771
31.2.1 WDT Refresh Register (WDTRR) ... 1771
31.2.2 WDT Control Register (WDTCR) ... 1772
31.2.3 WDT Status Register (WDTSR) ... 1775
31.2.4 WDT Reset Control Register (WDTRCR) ... 1776
31.2.5 WDT Count Stop Control Register (WDTCSTPR) ... 1777
31.2.6 Option Function Select Register 0 (OFS0) ... 1777
31.3 Operation ... 1778
31.3.1 Count Operation in Each Start Mode ... 1778
31.3.1.1 Register Start Mode ... 1778

31.3.1.2 Auto-Start Mode ... 1780
31.3.2 Refresh Operation ... 1781
31.3.3 Reset Output ... 1782
31.3.4 Interrupt Source ... 1782
31.3.5 Reading the Counter Value ... 1782
31.3.6 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers ... 1783
31.4 Link Operation by ELC ... 1783
32. Independent Watchdog Timer (IWDTa) ... 1784
32.1 Overview ... 1784
32.2 Register Descriptions ... 1786
32.2.1 IWDT Refresh Register (IWDTRR) ... 1786
32.2.2 IWDT Control Register (IWDTCR) ... 1787
32.2.3 IWDT Status Register (IWDTSR) ... 1790
32.2.4 IWDT Reset Control Register (IWDTRCR) ... 1791
32.2.5 IWDT Count Stop Control Register (IWDTCSTPR) ... 1792
32.2.6 Option Function Select Register 0 (OFS0) ... 1792
32.3 Operation ... 1793
32.3.1 Count Operation in Each Start Mode ... 1793
32.3.1.1 Register Start Mode ... 1793
32.3.1.2 Auto-Start Mode ... 1795
32.3.2 Refresh Operation ... 1796
32.3.3 Status Flags ... 1798
32.3.4 Reset Output ... 1798
32.3.5 Interrupt Sources ... 1798
32.3.6 Reading the Counter Value ... 1799
32.3.7 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers ... 1800
32.4 Link Operation by ELC ... 1800
32.5 Usage Notes ... 1801
32.5.1 Refresh Operations ... 1801
32.5.2 Count Clock Frequency Setting ... 1801
32.5.3 Reading Counter after Returning from Deep Software Standby Mode 1 ... 1801
32.5.4 Transitions to Low Power Consumption Mode ... 1801
33. Ethernet Module ... 1802
33.1 Overview ... 1802
33.2 Register Description ... 1806
33.2.1 ESWM Interface Control Registers ... 1806
33.2.1.1 Media Interface Reset Register (MIIRR) ... 1806
33.2.1.2 Media Interface Control Register 0 (MIICR0) ... 1807
33.2.1.3 Media Clock Capture Event Select Register (MCCESR) ... 1808
33.2.1.4 TAS Status Monitor Signal Select Register (TASSTSR) ... 1809

33.2.1.5 Ether Bus Control Register (ETHBUSCR) ... 1810
33.2.2 Interrupt Mapping Function Registers ... 1811
33.2.2.1 Error and Status Interrupt Mapping Register 1 (TPEMIMC1) ... 1811
33.2.2.2 Error and Status Interrupt Mapping Register 2 (TPEMIMC2) ... 1813
33.2.2.3 Error and Status Interrupt Mapping Register 3 (TPEMIMC3) ... 1814
33.2.2.4 Error and Status Interrupt Mapping Register 6n (TPEMIMC6n) (n = 0, 1) ... 1815
33.2.2.5 Error and Status Interrupt Mapping Register 7n (TPEMIMC7n) (n = 0 to 63) ... 1816
33.2.3 Interrupt Mirroring Registers ... 1817
33.2.3.1 ESWM Interrupt Mirroring Register (TSIM) ... 1817
33.2.3.2 MFWD Interrupt Mirroring Register (TFIM) ... 1818
33.2.3.3 COMA Interrupt Mirroring Register (TCIM) ... 1819
33.2.3.4 GWCA0 Interrupt Mirroring Register (TGIM0) ... 1820
33.2.3.5 ETHA0 Interrupt Mirroring Register (TEIM0) ... 1821
33.3 Operation ... 1822
33.3.1 Operating Clocks ... 1822
33.3.2 Interface Modes ... 1822
33.3.2.1 MII/GMII Interface ... 1822
33.3.2.2 RGMII Interface ... 1823
33.3.2.3 RMII Interface ... 1823
33.3.2.4 MAC-to-MAC Connection ... 1823
33.3.3 Software Flows ... 1824
33.3.3.1 Initialization Flow ... 1824
33.3.3.2 Disable Flow ... 1824
33.3.3.3 Reset Flow ... 1825
33.3.3.4 Emergency Reset Flow ... 1825
33.4 Interrupts and Events ... 1826
33.5 Usage Notes ... 1827
33.5.1 Module-Stop Function Setting ... 1827
33.5.2 Transition to Software Standby Mode ... 1827
33.5.3 Return from Software Standby Mode ... 1827
33.5.4 Notes on Cutting Off the Power Supply to ESWM ... 1827
33.5.5 Note on Resuming Power Supply to ESWM ... 1827
33.5.6 Notes on Using Half-Duplex Mode ... 1827
34. Ethernet Message Forwarding Engine (MFWD) ... 1828
34.1 Overview ... 1828
34.2 Register Descriptions ... 1831
34.2.1 General Function Registers ... 1831
34.2.1.1 General Configuration Register (FWGC) ... 1831
34.2.1.2 TAG TPID Configuration Register 0 (FWTTC0) ... 1832
34.2.1.3 TAG TPID Configuration Register 1 (FWTTC1) ... 1833
34.2.1.4 CPU Exceptional Path Target Configuration Register (FWCEPTC) ... 1834

34.2.1.5 CPU Exceptional Path Reason Configuration Register 0 (FWCEPRC0) 1835
34.2.1.6 CPU Exceptional Path Reason Configuration Register 2 (FWCEPRC2) 1837
34.2.1.7 CPU Mirroring Path Target Configuration Register (FWCMPTC) 1837
34.2.1.8 Ethernet Mirroring Path Target Configuration Register (FWEMPTC) 1839
34.2.1.9 Source-Destination Mirroring Path Target Configuration Register (FWSDMPTC) 1840
34.2.1.10 Source-Destination Mirroring Path Vector Configuration Register (FWSDMPVC) 1841
34.2.1.11 Port i Level Based Watermark Configuration Register (FWLBWMCi) (i = 0, 2) 1842
34.2.1.12 Port i Configuration Register 1 (FWPCi1) (i = 0, 2) 1846
34.2.2 Port Based Forwarding Function Registers 1847
34.2.2.1 Port Based Forwarding Configuration Register i (FWPBFCi) (i = 0, 2) 1847
34.2.2.2 Port i Port Based Forwarding CSD Configuration Register 0 (FWPBFCSDCi0) (i = 0, 2) 1848
34.2.3 MFWD Counter Registers 1849
34.2.3.1 Port 2 Direct Descriptor Forwarded Descriptor Counter Register (FWDDFDCN2) 1849
34.2.3.2 Port i Port Based Forwarded Descriptor Counter Register (FWPBFDCNi) (i = 0, 2) 1850
34.2.3.3 Port i Watermark Rejected Descriptor Counter Register (FWWMRDCNi) (i = 0 to 2) 1851
34.2.3.4 Port 2 Direct Descriptor Rejected Descriptor Counter Register (FWDDRDCN2) 1852
34.2.3.5 Port i Port Based Rejected Descriptor Counter Register (FWPBRDCNi) (i = 0, 2) 1853
34.2.4 MFWD Interrupt Registers 1854
34.2.4.1 Port i Error Interrupt Status Register 0 (FWEIS0i) (i = 0, 2) 1854
34.2.4.2 Port i Error Interrupt Enable Register 0 (FWEIE0i) (i = 0, 2) 1857
34.2.4.3 Port i Error Interrupt Disable Register 0 (FWEID0i) (i = 0, 2) 1859
34.3 Operations 1860
34.3.1 General Functionalities 1860
34.3.1.1 Descriptors 1860
34.3.1.2 ESWM Modes 1861
34.3.1.3 ESWM TPIDs 1861
34.3.1.4 ESWM Frame Input Priority 1862
34.3.1.5 ESWM Debug Functions 1863
34.3.1.6 Watermark 1866
34.3.2 Forwarding/Routing 1868
34.3.2.1 Direct Descriptor Forwarding 1869
34.3.2.2 Port Based Forwarding 1872
34.3.3 Data Representation in RAMs 1875
34.3.3.1 RAM Address Dependencies 1875
34.3.3.2 RAM Access by Agents 1876
34.3.4 Fabric Time Arbiter 1878
34.3.4.1 Agents Done Signal Distribution 1878
34.3.4.2 Done Signal Patterns 1879

34.3.4.3 Fabric Jitter 1880
34.3.4.4 Fabric Minimum Latency 1880
34.3.5 Source Port Generation 1881
34.3.6 Software Flows 1882
34.3.6.1 Interrupt Handling Flow 1882
34.3.6.2 Port i Setting Flow 1882
34.3.6.3 Port Based Forwarding Flows 1883
34.3.6.4 Called Software Flows 1884
34.3.6.5 Register Writable Without Software Flow After Initialization 1887
35. Ethernet Common Agent (COMA) 1888
35.1 Overview 1888
35.1.1 Features 1888
35.1.2 COMA Block Diagram 1889
35.2 Register Description 1890
35.2.1 Global Registers 1890
35.2.1.1 IP Version Register (RIPV) 1890
35.2.1.2 Reset Configuration Register (RRC) 1891
35.2.1.3 Clock Enable Configuration Register (RCEC) 1892
35.2.1.4 Clock Disable Configuration Register (RCDC) 1893
35.2.2 Buffer Pool Function Registers 1894
35.2.2.1 Buffer Pool IPV Based Watermark Configuration Register i (CABPIBWMCi) (i = 0 to 7) 1894
35.2.2.2 Buffer Pool Watermark Level Configuration Register (CABPWMLC) 1895
35.2.2.3 Buffer Pointer Pause Frame Level i Configuration Register (CABPPFLCi) (i = 0, 1) 1896
35.2.2.4 Port i Buffer Pool Watermark Level Configuration Register (CABPPWMLCi) (i = 0 to 2) 1897
35.2.2.5 Port i Buffer Pointer Pause Frame Level j Configuration Register (CABPPPFLCij) (i = 0 to 2; j = 0, 1) 1898
35.2.2.6 Port i Buffer Pointer Utilization Level Configuration Register (CABPULCi) (i = 0 to 2) 1899
35.2.2.7 Buffer Pool Initialization Register Monitoring Register (CABPIRM) 1900
35.2.2.8 Buffer Pool Pointer Count Monitoring Register (CABPPCM) 1901
35.2.2.9 Buffer Pool Pointer Least Count Monitoring Register (CABPLCM) 1902
35.2.2.10 Port i Buffer Pointer Count Monitoring Register (CABPCPMi) (i = 0 to 2) 1903
35.2.2.11 Port i Buffer Pointer Maximum Count Monitoring Register (CABPMCPMi) (i = 0 to 2) 1904
35.2.3 Reject RAM Function Registers 1905
35.2.3.1 Rejected Descriptor Number Monitoring Register (CARDNM) 1905
35.2.3.2 Rejected Descriptor Maximum Number Monitoring Register (CARDMNM) 1906
35.2.4 Reject RAM Counter Registers 1907
35.2.4.1 Rejected Descriptor Counter Register (CARDCN) 1907
35.2.5 Error Interrupt Registers 1908

35.2.5.1 Error Interrupt Status Register 0 (CAEIS0) 1908
35.2.5.2 Error Interrupt Enable Register 0 (CAEIE0) 1910
35.2.5.3 Error Interrupt Disable Register 0 (CAEID0) 1911
35.2.5.4 Error Interrupt Status Register 1 (CAEIS1) 1912
35.2.5.5 Error Interrupt Enable Register 1 (CAEIE1) 1913
35.2.5.6 Error Interrupt Disable Register 1 (CAEID1) 1914
35.2.6 Monitoring Interrupt Registers 1915
35.2.6.1 Monitoring Interrupt Status Register 0 (CAMIS0) 1915
35.2.6.2 Monitoring Interrupt Enable Register 0 (CAMIE0) 1916
35.2.6.3 Monitoring Interrupt Disable Register 0 (CAMID0) 1916
35.2.6.4 Monitoring Interrupt Status Register 1 (CAMIS1) 1917
35.2.6.5 Monitoring Interrupt Enable Register 1 (CAMIE1) 1918
35.2.6.6 Monitoring Interrupt Disable Register 1 (CAMID1) 1919
35.3 Operations 1920
35.3.1 Watermark Function 1920
35.3.1.1 IPV Based Watermark 1920
35.3.1.2 Global Level-Based Watermark 1921
35.3.1.3 Per-Port Level-Based Watermark 1922
35.3.2 Pause Function 1923
35.3.2.1 Global Pause Function 1923
35.3.2.2 Per-Port Pause Function 1924
35.3.3 Per Port Memory Allocation Function 1925
35.3.3.1 Shared Memory Setting 1925
35.3.3.2 Reduced Memory Setting 1 1926
35.3.3.3 Reduced Memory Setting 2 1927
35.3.3.4 Hybrid Memory Setting 1928
35.3.4 Software Flows 1929
35.3.4.1 Interrupt Handling Flow 1929
35.3.4.2 Called Software Flows 1930
36. Ethernet Agent (ETHA) 1934
36.1 Overview 1934
36.1.1 Features 1934
36.1.2 Ethernet Agent Block Diagram 1935
36.2 Register Description 1936
36.2.1 ETHA Mode Function Registers 1936
36.2.1.1 Mode Configuration Register (EAMC) 1936
36.2.1.2 Mode Status Register (EAMS) 1937
36.2.2 Transmission Function Registers 1938
36.2.2.1 IPV Remapping Configuration Register [802.1Q] (EAIRC) 1938
36.2.2.2 TX Descriptor Queue Security Configuration Register (EATDQSC) 1939
36.2.2.3 TX Descriptor Queue Configuration Register (EATDQC) 1940

36.2.2.4 TX Descriptor Queue Arbitration Configuration Register (EATDQAC) 1942
36.2.2.5 Transmission Maximum Frame Size Configuration Register q (EATMFSCq) (q = 0 to 7) 1943
36.2.2.6 Transmission Descriptor Queue Depth Configuration Register q (EATDQDCq) (q = 0 to 7) 1944
36.2.2.7 Transmission Descriptor Queue q Monitoring Register (EATDQMq) (q = 0 to 7) 1945
36.2.2.8 Transmission Descriptor Queue q Max Level Monitoring Register (EATDQMLMq) (q = 0 to 7) 1946
36.2.3 TAG Function Registers 1947
36.2.3.1 VLAN Control Configuration Register (EAVCC) 1947
36.2.3.2 VLAN TAG Configuration Register (EAVTC) 1947
36.2.3.3 Reception TAG Filtering Configuration Register (EARTFC) 1948
36.2.4 ETHA Counter Registers 1949
36.2.4.1 Under Minimum Frame Size Error Counter Register (EAUSMFSECN) 1949
36.2.4.2 TAG Filtering Error Counter Register (EATFECN) 1950
36.2.4.3 Frame Size Error Counter Register (EAFSECN) 1951
36.2.4.4 Descriptor Queue Overflow Error Counter Register (EADQOECN) 1952
36.2.4.5 Descriptor Queue Security Error Counter Register (EADQSECN) 1953
36.2.5 Error Interrupt Registers 1954
36.2.5.1 Error Interrupt Status Register 0 (EAEIS0) 1954
36.2.5.2 Error Interrupt Enable Register 0 (EAEIE0) 1956
36.2.5.3 Error Interrupt Disable Register 0 (EAEID0) 1958
36.2.5.4 Error Interrupt Status Register 2 (EAEIS2) 1959
36.2.5.5 Error Interrupt Enable Register 2 (EAEIE2) 1961
36.2.5.6 Error Interrupt Disable Register 2 (EAEID2) 1963
36.2.5.7 Error Interrupt Status Register 3 (EAEIS3) 1964
36.2.5.8 Error Interrupt Enable Register 3 (EAEIE3) 1966
36.2.5.9 Error Interrupt Disable Register 3 (EAEID3) 1968
36.3 Operation 1969
36.3.1 Data Transmission 1969
36.3.1.1 Descriptor Store 1969
36.3.1.2 Frame Size Check 1971
36.3.1.3 TX Data Fetch 1972
36.3.1.4 RMAC TX Interface Control 1972
36.3.2 Data Reception 1973
36.3.2.1 RX Data Store 1973
36.3.3 Operating Modes 1975
36.3.3.1 Operating Mode Transitions 1976
36.3.4 Software Flows 1977
36.3.4.1 Mode Transition Flow 1978
36.3.4.2 Reset Flow 1979
36.3.4.3 Initialization Flow 1980

36.3.4.4 Reinitialization Flow 1981
36.3.4.5 Interrupt Handling Flow 1982
36.3.4.6 Called Software Flows 1983
37. Ethernet MAC (RMAC) 1984
37.1 Overview 1984
37.1.1 Features 1984
37.1.2 RMAC Block Diagram 1986
37.2 Register Description 1987
37.2.1 Function Registers 1987
37.2.1.1 PHY Station Management Register (MPSM) 1987
37.2.1.2 PHY Interfaces Configuration Register (MPIC) 1989
37.2.1.3 PHY Interfaces Monitoring Register (MPIM) 1991
37.2.1.4 I/O Configuration Register (MIOC) 1992
37.2.1.5 Transmission Frame Format Configuration Register (MTFFC) 1993
37.2.1.6 Transmission Pause or PFC Frame Configuration Register (MTPFC) 1994
37.2.1.7 Transmission Pause or PFC Frame Configuration Register 2 (MTPFC2) 1996
37.2.1.8 Transmission Pause or PFC Frame Configuration Register 3t (MTPFC3t) (t = 0, 1) ... 1998
37.2.1.9 Transmission Flow Control JAM Configuration Register (MTFCJC) 1999
37.2.1.10 Reception General Configuration Register (MRGC) 2000
37.2.1.11 Reception MAC Address Configuration Register 0 (MRMAC0) 2002
37.2.1.12 Reception MAC Address Configuration Register 1 (MRMAC1) 2002
37.2.1.13 Reception Address Filter Configuration Register (MRAFC) 2003
37.2.1.14 Reception Storm Configuration for E-Frames Register (MRSCE) 2005
37.2.1.15 Reception Storm Counter Configuration Register (MRSCC) 2006
37.2.1.16 Reception Frame Size Configuration for E-Frames Register (MRFSCE) 2007
37.2.1.17 Timestamp Reception Configuration Register (MTRC) 2008
37.2.1.18 Reception Pause or PFC Frame Monitoring Register (MRPFM) 2009
37.2.1.19 PTP Filtering Register Configuration Register t (MPFCt) (t = 0 to 15) 2010
37.2.1.20 Energy Efficient Ethernet Configuration Register (MEEEC) 2011
37.2.1.21 Loopback Configuration Register (MLBC) 2012
37.2.2 Counter Registers 2013
37.2.2.1 Manual Pause Frame Transmit Counter Register (MMPFTCT) 2013
37.2.2.2 Automatic Pause Frame Transmit Counter Register (MAPFTCT) 2014
37.2.2.3 Pause Frame Receive Counter Register (MPFRCT) 2015
37.2.2.4 False Carrier Indication Counter Register (MFCICT) 2016
37.2.2.5 Energy Efficient Ethernet Counter Register (MEEECT) 2017
37.2.2.6 Manual PFC Frame Transmit Counter Register t (MMPCFTCTt) (t = 0, 1) 2018
37.2.2.7 Automatic PFC Frame Transmit Counter Register t (MAPCFTCTt) (t = 0, 1) 2019
37.2.2.8 PFC Frame Receive Counter Register t (MPCFRCTt) (t = 0 to 7) 2020
37.2.2.9 Receive Overflow Counter Register (MROVFC) 2021
37.2.2.10 Received Byte Counter E-Frames Upper Side Register (MRXBCEU) 2022

37.2.2.11 Received Byte Counter E-Frames Lower Side Register (MRXBCEL) 2023
37.2.2.12 Received Good Frame Counter E-Frames Register (MRGFCE) 2024
37.2.2.13 Received Good Broadcast Frame Counter Register (MRBFC) 2025
37.2.2.14 Received Good Multicast Frame Counter Register (MRMFC) 2026
37.2.2.15 Received Good Unicast Frame Counter Register (MRUFC) 2027
37.2.2.16 Received PHY Error Frame Count Register (MRPEFC) 2028
37.2.2.17 Received Nibble Error Frame Count Register (MRNEFC) 2029
37.2.2.18 Received FCS/mCRC Error Frame Count Register (MRFMEFC) 2030
37.2.2.19 Received Final Fragment Missing Error Frame Count Register (MRFFMEFC) 2031
37.2.2.20 Received C-Fragment Count Error Frame Count Register (MRCFCEFC) 2032
37.2.2.21 Received Fragment Count Error Frame Count Register (MRFCEFC) 2033
37.2.2.22 Received MAC Filter Error Frame Count Register (MRRCFEFC) 2034
37.2.2.23 Received Frame Count Register (MRFC) 2034
37.2.2.24 Received Good Undersize Error Frame Count Register (MRGUEFC) 2035
37.2.2.25 Received Bad Undersize Error Frame Count Register (MRBUEFC) 2035
37.2.2.26 Received Good Oversize Error Frame Count Register (MRGOEFC) 2036
37.2.2.27 Received Bad Oversize Error Frame Count Register (MRBOEFC) 2036
37.2.2.28 Transmitted Byte Counter E-Frames Upper Side Register (MTXBCEU) 2037
37.2.2.29 Transmitted Byte Counter E-Frames Lower Side Register (MTXBCEL) 2038
37.2.2.30 Transmitted Good Frame Counter E-Frames Register (MTGFCE) 2039
37.2.2.31 Transmitted Broadcast Frame Counter Register (MTBFC) 2040
37.2.2.32 Transmitted Multicast Frame Counter Register (MTMFC) 2040
37.2.2.33 Transmitted Unicast Frame Counter Register (MTUFC) 2041
37.2.2.34 Transmitted Error Frame Counter Register (MTEFC) 2041
37.2.2.35 Transmitted Half Duplex Good Frame Counter Register (MTHDGFC) 2042
37.2.3 Interrupt Registers 2043
37.2.3.1 Error Interrupt Status Register (MEIS) 2043
37.2.3.2 Error Interrupt Enable Register (MEIE) 2050
37.2.3.3 Error Interrupt Disable Register (MEID) 2055
37.2.3.4 Monitoring Interrupt Status Register 0 (MMIS0) 2057
37.2.3.5 Monitoring Interrupt Enable Register 0 (MMIE0) 2059
37.2.3.6 Monitoring Interrupt Disable Register 0 (MMID0) 2060
37.2.3.7 Monitoring Interrupt Status Register 1 (MMIS1) 2061
37.2.3.8 Monitoring Interrupt Enable Register 1 (MMIE1) 2063
37.2.3.9 Monitoring Interrupt Disable Register 1 (MMID1) 2065
37.2.3.10 Monitoring Interrupt Status Register 2 (MMIS2) 2066
37.2.3.11 Monitoring Interrupt Enable Register 2 (MMIE2) 2068
37.2.3.12 Monitoring Interrupt Disable Register 2 (MMID2) 2069
37.3 Operation 2070
37.3.1 Flow Control 2070
37.3.1.1 Flow Control by PAUSE Frame 2070

37.3.1.2 Flow Control by PFC Frame 2072
37.3.1.3 Flow Control by JAM 2076
37.3.2 Timestamping 2077
37.3.2.1 Timestamp Capture 2077
37.3.3 Station Management 2079
37.3.3.1 Management Register Access 2079
37.3.3.2 Management Data Clock (MDC) Setting 2082
37.3.3.3 Management Data Timing Adjustment 2082
37.3.4 Energy Efficient Ethernet 2083
37.3.4.1 LPI Control 2083
37.3.5 MAC Address Filtering on Reception 2084
37.3.6 Operation Modes 2086
37.3.6.1 Frame Handling 2086
37.3.7 Software Flows 2087
37.3.7.1 Initialization Flow 2087
37.3.7.2 Interrupt Handling Flow 2088
37.3.7.3 PHY MDIO and eMDIO Access Flows 2089
37.3.7.4 LPI Transition for Transmit Port 2091
37.3.7.5 LINK Verification 2092
38. Ethernet CPU Agent (GWCA) 2093
38.1 Overview 2093
38.1.1 Features 2093
38.1.2 GWCA Block Diagram 2094
38.2 Register Descriptions 2095
38.2.1 GWCA Mode Function Registers 2095
38.2.1.1 Mode Configuration Register (GWMC) 2095
38.2.1.2 Mode Status Register (GWMS) 2096
38.2.2 Reception Function Registers 2097
38.2.2.1 IPV Remapping Configuration Register [802.1Q] (GWIRC) 2097
38.2.2.2 RX Descriptor Queue Security Configuration Register (GWRDQSC) 2098
38.2.2.3 RX Descriptor Queue Control Register (GWRDQC) 2099
38.2.2.4 RX Descriptor Queue Arbitration Control Register (GWRDQAC) 2100
38.2.2.5 RX General Configuration Register (GWRGC) 2101
38.2.2.6 Reception Maximum Frame Size Configuration Register q (GWRMFSCq) (q = 0 to 7) 2102
38.2.2.7 Reception Descriptor Queue q Depth Configuration Register (GWRDQDCq) (q = 0 to 7) 2103
38.2.2.8 RX Descriptor Queue q Monitoring Register (GWRDQMq) (q = 0 to 7) 2104
38.2.2.9 RX Descriptor Queue q Max Level Monitoring Register (GWRDQMLMq) (q = 0 to 7) 2105
38.2.3 Reception Multicast Function Registers 2106
38.2.3.1 Multicast Table Initialization Register Monitoring Register (GWMTIRM) 2106

38.2.3.2 Multicast Table Learning Setting Register (GWMSTLS) 2107
38.2.3.3 Multicast Table Learning Result Register (GWMSTLR) 2108
38.2.3.4 Multicast Table Searching Setting Register (GWMSTSS) 2109
38.2.3.5 Multicast Table Searching Result Register (GWMSTSR) 2110
38.2.4 MAC Function Registers 2111
38.2.4.1 MAC Address Configuration Register 0 (GWMAC0) 2111
38.2.4.2 MAC Address Configuration Register 1 (GWMAC1) 2111
38.2.5 TAG Function Registers 2112
38.2.5.1 VLAN Control Configuration Register (GWVCC) 2112
38.2.5.2 VLAN TAG Configuration Register (GWVTC) 2113
38.2.5.3 Transmission TAG Filtering Configuration Register (GWTTFC) 2114
38.2.6 Timestamp Function Registers 2115
38.2.6.1 Timestamp Descriptor Chain s Address Configuration Register 0 (GWTDCACs0) (s = 0, 1) 2115
38.2.6.2 Timestamp Descriptor Chain s Address Configuration Register 1 (GWTDCACs1) (s = 0, 1) 2116
38.2.6.3 Timestamp Descriptor Chain s Configuration Register (GWTSDCCs) (s = 0, 1) 2117
38.2.6.4 Timestamp Number Monitoring Register (GWTSNM) 2118
38.2.6.5 Timestamp Maximum Number Monitoring Register (GWTSMNM) 2118
38.2.7 AXI Master Registers 2119
38.2.7.1 AXI Control Register (GWAC) 2119
38.2.7.2 Descriptor Chain Base Address Configuration Register 0 (GWDCBAC0) 2120
38.2.7.3 Descriptor Chain Base Address Configuration Register 1 (GWDCBAC1) 2120
38.2.7.4 Maximum Descriptor Number Configuration Register (GWMDNC) 2121
38.2.7.5 Transmission Request Configuration Register i (GWTRCi) (i = 0, 1) 2122
38.2.7.6 Transmission Pause Configuration Register p (GWTPCp) (p = 0, 1) 2125
38.2.7.7 AXI RAM Initialization Register Monitoring Register (GWARIRM) 2126
38.2.7.8 Descriptor Chain i Configuration Register (GWDCCi) (i = 0 to 63) 2127
38.2.7.9 AXI Address RAM Searching Setting Register (GWAARSS) 2129
38.2.7.10 AXI Address RAM Searching Result Register 0 (GWAARSR0) 2130
38.2.7.11 AXI Address RAM Searching Result Register 1 (GWAARSR1) 2131
38.2.7.12 Incremental Data Area i Used Area Size Register (GWIDAUASi) (i = 0 to 3) 2132
38.2.7.13 Incremental Data Area i Size Monitoring Register (GWIDASMi) (i = 0 to 3) 2133
38.2.7.14 Incremental Data Area i Start Address Monitoring Register 0 (GWIDASAMi0) (i = 0 to 3) 2134
38.2.7.15 Incremental Data Area i Start Address Monitoring Register 1 (GWIDASAMi1) (i = 0 to 3) 2135
38.2.7.16 Incremental Data Area i Current Address Monitoring Register 0 (GWIDACAMi0) (i = 0 to 3) 2136
38.2.7.17 Incremental Data Area i Current Address Monitoring Register 1 (GWIDACAMi1) (i = 0 to 3) 2137

38.2.8 Rate Limiter Function Registers 2138
38.2.8.1 Global Rate Limiter Configuration Register (GWGRLC) 2138
38.2.8.2 Global Rate Limiter Upper Limit Configuration Register (GWGRLULC) 2139
38.2.8.3 Rate Limiter i Configuration Register (GWRLCi) (i = 0 to 7) 2140
38.2.8.4 Rate Limiter i Upper Limit Configuration Register (GWRLULCi) (i = 0 to 7) 2141
38.2.9 Interrupt Function Registers 2142
38.2.9.1 Interrupt Delay Prescaler Configuration Register (GWIDPC) 2142
38.2.9.2 Interrupt Delay Configuration Register (GWIDCi) (i = 0 to 63) 2143
38.2.10 Counter Registers 2144
38.2.10.1 Received Data Counter Register (GWRDCN) 2144
38.2.10.2 Transmitted Data Counter Register (GWTDCN) 2145
38.2.10.3 Timestamp Counter Register (GWTSCN) 2146
38.2.10.4 Timestamp Overflow Error Counter Register (GWTSOVFECN) 2147
38.2.10.5 Under Minimum Frame Size Error Counter Register (GWUSMFSECN) 2148
38.2.10.6 TAG Filtering Error Counter Register (GWTFECN) 2149
38.2.10.7 Sequence Error Counter Register (GWSEQECN) 2150
38.2.10.8 TX Descriptor Number Error Counter Register (GWTXDNECN) 2151
38.2.10.9 Frame Size Error Counter Register (GWFSECN) 2152
38.2.10.10 Timestamp Descriptor Full Error Counter Register (GWTDFECN) 2153
38.2.10.11 Timestamp Descriptor Number Error Counter Register (GWTSDNECN) 2154
38.2.10.12 Descriptor Queue Overflow Error Counter Register (GWDQOECN) 2155
38.2.10.13 Descriptor Queue Security Error Counter Register (GWDQSECN) 2156
38.2.10.14 Descriptor Full Error Counter Register (GWDFECN) 2157
38.2.10.15 Descriptor Security Error Counter Register (GWDSECN) 2158
38.2.10.16 Data Size Error Counter Register (GWDSZECN) 2159
38.2.10.17 Descriptor Chain Type Error Counter Register (GWDCTECN) 2160
38.2.10.18 RX Descriptor Number Error Counter Register (GWRXDNECN) 2161
38.2.11 Data Interrupt Registers 2162
38.2.11.1 Data Interrupt Status Register i (GWDISi) (i = 0, 1) 2162
38.2.11.2 Data Interrupt Enable Register i (GWDIEi) (i = 0, 1) 2166
38.2.11.3 Data Interrupt Disable Register i (GWDIDi) (i = 0, 1) 2170
38.2.11.4 Data Interrupt Delayed Status Register i (GWDIDSi) (i = 0, 1) 2174
38.2.11.5 Timestamp Data Interrupt Status Register (GWTSDIS) 2178
38.2.11.6 Timestamp Data Interrupt Enable Register (GWTSDIE) 2179
38.2.11.7 Timestamp Data Interrupt Disable Register (GWTSDID) 2179
38.2.12 Error Interrupt Registers 2180
38.2.12.1 Error Interrupt Status Register 0 (GWEIS0) 2180
38.2.12.2 Error Interrupt Enable Register 0 (GWEIE0) 2185
38.2.12.3 Error Interrupt Disable Register 0 (GWEID0) 2188
38.2.12.4 Error Interrupt Status Register 1 (GWEIS1) 2190

38.2.12.5 Error Interrupt Enable Register 1 (GWEIE1) 2192
38.2.12.6 Error Interrupt Disable Register 1 (GWEID1) 2194
38.2.12.7 Error Interrupt Status Register 2i (GWEIS2i) (i = 0, 1) 2195
38.2.12.8 Error Interrupt Enable Register 2i (GWEIE2i) (i = 0, 1) 2199
38.2.12.9 Error Interrupt Disable Register 2i (GWEID2i) (i = 0, 1) 2203
38.2.12.10 Error Interrupt Status Register 3 (GWEIS3) 2207
38.2.12.11 Error Interrupt Enable Register 3 (GWEIE3) 2208
38.2.12.12 Error Interrupt Disable Register 3 (GWEID3) 2209
38.2.12.13 Error Interrupt Status Register 4 (GWEIS4) 2210
38.2.12.14 Error Interrupt Enable Register 4 (GWEIE4) 2213
38.2.12.15 Error Interrupt Disable Register 4 (GWEID4) 2214
38.2.12.16 Error Interrupt Status Register 5 (GWEIS5) 2215
38.2.12.17 Error Interrupt Enable Register 5 (GWEIE5) 2217
38.2.12.18 Error Interrupt Disable Register 5 (GWEID5) 2218
38.3 Operations 2219
38.3.1 AXI Master General Information 2219
38.3.1.1 AXI Descriptor Queue Mapping 2219
38.3.1.2 AXI Descriptor General Description 2220
38.3.1.3 LINKFIX Table 2224
38.3.1.4 AXI Common Descriptor Utilization 2226
38.3.1.5 AXI Descriptor Queue Format 2229
38.3.1.6 AXI Address RAM Searching 2231
38.3.2 Data Transmission 2232
38.3.2.1 TX Queue Arbitration 2233
38.3.2.2 AXI Master Interface 2235
38.3.2.3 TX Data Store 2244
38.3.3 Data Reception 2251
38.3.3.1 Descriptor Store 2252
38.3.3.2 Multicast Control 2258
38.3.3.3 RX Data Fetch 2262
38.3.3.4 AXI Master Interface 2267
38.3.4 Timestamp Reception [GPTP] 2286
38.3.4.1 Timestamp Control 2286
38.3.4.2 AXI Master Interface 2287
38.3.5 gPTP Triggered Transmission 2289
38.3.6 Interrupt Delay Function 2290
38.3.7 Operation Modes 2291
38.3.7.1 Operation Mode Transitions 2292
38.3.8 Software Flows 2293
38.3.8.1 Mode Transition Flow 2293
38.3.8.2 Reset Flow 2294

38.3.8.3 Initialization Flow 2295
38.3.8.4 Reinitialization Flow 2296
38.3.8.5 Multicast Table Reset Flow 2297
38.3.8.6 Multicast Table Setting Flow 2298
38.3.8.7 Multicast Table Read Flow 2299
38.3.8.8 AXI RAM Reset Flow 2300
38.3.8.9 AXI RAM Read Flow 2301
38.3.8.10 Global Rate Limiter Setting Flow 2302
38.3.8.11 Global Rate Limiter Disabling Flow 2302
38.3.8.12 Rate Limiter i Setting Flow 2303
38.3.8.13 Rate Limiter i Disabling Flow 2303
38.3.8.14 AXI Table Read Flow 2304
38.3.8.15 Interrupt Handling Flow 2305
38.3.8.16 Called Software Flows 2306
38.3.8.17 Register Writable Without Software Flow 2308
39. Ethernet Generic PTP Timer (GPTP) 2309
39.1 Overview 2309
39.1.1 Features 2309
39.1.2 Block Diagram 2310
39.2 Register Descriptions 2312
39.2.1 Version Register 2312
39.2.1.1 IP Version Register (PTPIPV) 2312
39.2.2 Timer Configuration Registers 2313
39.2.2.1 Timer Enable Configuration Register (PTPTMEC) 2313
39.2.2.2 Timer Disable Configuration Register (PTPTMDC) 2313
39.2.2.3 Timer 0 Increment Value Configuration Register (PTPTIVC0) 2314
39.2.2.4 Timer 0 Offset Value Configuration Register L (PTPTOVC0L) 2315
39.2.2.5 Timer 0 Offset Value Configuration Register M (PTPTOVC0M) 2316
39.2.2.6 Timer 0 Offset Value Configuration Register U (PTPTOVC0U) 2316
39.2.3 Timer Monitoring Registers 2317
39.2.3.1 AVTP Timer 0 Monitoring Register L (PTPAVTPTM0L) 2317
39.2.3.2 AVTP Timer 0 Monitoring Register U (PTPAVTPTM0U) 2317
39.2.3.3 gPTP Timer 0 Monitoring Register L (PTPGPTPTM0L) 2318
39.2.3.4 gPTP Timer 0 Monitoring Register M (PTPGPTPTM0M) 2318
39.2.3.5 gPTP Timer 0 Monitoring Register U (PTPGPTPTM0U) 2319
39.2.4 Media Clock Capture Registers 2320
39.2.4.1 Media Clock m Capture Configuration Register (PTPMCCCm) (m = 0, 1) 2320
39.2.4.2 Media Clock m Capture Monitoring Register L (PTPMCCMmL) (m = 0, 1) 2321
39.2.4.3 Media Clock m Capture Monitoring Register M (PTPMCCMmM) (m = 0, 1) 2322
39.2.4.4 Media Clock m Capture Monitoring Register U (PTPMCCMmU) (m = 0, 1) 2323
39.2.5 Media Clock Recovery Registers 2325

39.2.5.1 Media Clock m Recovery Configuration Register (PTPMCRCm) (m = 0, 1) 2325
39.2.5.2 Media Clock m Recovery Time Configuration Register L (PTPMCRTCmL) (m = 0, 1) 2326
39.2.5.3 Media Clock m Recovery Time Configuration Register M (PTPMCRTCmM) (m = 0, 1) 2326
39.2.5.4 Media Clock m Recovery Time Configuration Register U (PTPMCRTCmU) (m = 0, 1) 2327
39.2.5.5 Media Clock m Pin Configuration Register (PTPMCPCm) (m = 0, 1) 2328
39.2.6 Cyclic Compare Registers 2329
39.2.6.1 Cyclic Compare c Configuration Register 0 (PTPCCCc0) (c = 0 to 7) 2329
39.2.6.2 Cyclic Compare c Configuration Register 1 (PTPCCCc1) (c = 0 to 7) 2329
39.2.7 Interrupt Registers 2330
39.2.7.1 Interrupt Status Register 0 (PTPIS0) 2330
39.2.7.2 Interrupt Enable Register 0 (PTPIE0) 2331
39.2.7.3 Interrupt Disable Register 0 (PTPID0) 2332
39.2.7.4 Interrupt Status Register 1 (PTPIS1) 2333
39.2.7.5 Interrupt Enable Register 1 (PTPIE1) 2334
39.2.7.6 Interrupt Disable Register 1 (PTPID1) 2335
39.2.8 Pulse Output Timer Configuration Registers 2336
39.2.8.1 Pulse Output Timer Configuration Register (POTCFGR) 2336
39.2.8.2 Pulse Output Timer Control Register n (POTCRn) (n = 0, 1) 2337
39.2.8.3 Pulse Output Start Time Setting Register nU (POTSTRnU) (n = 0, 1) 2338
39.2.8.4 Pulse Output Start Time Setting Register nM (POTSTRnM) (n = 0, 1) 2339
39.2.8.5 Pulse Output Start Time Setting Register nL (POTSTRnL) (n = 0, 1) 2340
39.2.8.6 Period Setting Register nU (POTPERnU) (n = 0, 1) 2341
39.2.8.7 Period Setting Register nM (POTPERnM) (n = 0, 1) 2342
39.2.8.8 Period Setting Register nL (POTPERnL) (n = 0, 1) 2343
39.2.8.9 Pulse Width Setting Register n (POTPWRn) (n = 0, 1) 2344
39.2.9 Pulse Output Timer Monitoring Registers 2345
39.2.9.1 Time Capture Register nU (POTCPRnU) (n = 0, 1) 2345
39.2.9.2 Time Capture Register nM (POTCPRnM) (n = 0, 1) 2346
39.2.9.3 Time Capture Register nL (POTCPRnL) (n = 0, 1) 2347
39.3 Operations 2348
39.3.1 Timer 0 2348
39.3.1.1 gPTP Timer 0 2348
39.3.1.2 64-bit AVTP Timer 0 2348
39.3.1.3 32-bit AVTP Timer 0 2349
39.3.1.4 PPS Signal 2349
39.3.2 Media Clock Capture m (m = 0, 1) 2350
39.3.3 Media Clock Recovery m (m = 0, 1) 2351
39.3.4 mediaClock[m] Mapping to Output Pins 2352

39.3.5 Cyclic Compare c (c = 0 to 7) 2352
39.3.6 Pulse Output Timer n (n = 0, 1) 2353
39.3.6.1 gPTP Timer Selection 2353
39.3.6.2 Pulse Output Generation Registers 2353
39.3.6.3 Pulse Output Start 2353
39.3.6.4 Pulse Output Stop 2354
39.3.6.5 Timing of Pulse Output Permission 2354
39.3.6.6 Interrupt Output 2354
39.3.7 Pulse Output Time Chart 2355
39.3.8 Interrupt Output Time Chart 2359
39.3.9 Software Flows 2360
39.3.9.1 Interrupt Handling Flow 2360
39.3.9.2 Timer 0 Flows 2361
39.3.9.3 Media Clock m Capture Flows (m = 0, 1) 2364
39.3.9.4 Media Clock Recovery m Flows (m = 0, 1) 2367
39.3.9.5 Cyclic Compare c Flows (c = 0 to 7) 2369
39.3.9.6 Pulse Output Timer n Operation Flow (n = 0, 1) 2369
39.4 Usage Notes 2370
39.4.1 Note on Register Setting 2370
39.4.2 Restrictions on Register Setting 2370
40. USB 2.0 Full-Speed Host/Function Module (USBf) 2371
40.1 Overview 2371
40.2 Register Descriptions 2373
40.2.1 System Configuration Control Register (SYSCFG) 2373
40.2.2 System Configuration Status Register 0 (SYSSTS0) 2375
40.2.3 Device State Control Register 0 (DVSTCTR0) 2377
40.2.4 CFIFO Port Register (CFIFO),
D0FIFO Port Register (D0FIFO),
D1FIFO Port Register (D1FIFO) 2380
40.2.5 CFIFO Port Select Register (CFIFOSEL),
D0FIFO Port Select Register (D0FIFOSEL),
D1FIFO Port Select Register (D1FIFOSEL) 2382
40.2.6 CFIFO Port Control Register (CFIFOCTR),
D0FIFO Port Control Register (D0FIFOCTR),
D1FIFO Port Control Register (D1FIFOCTR) 2386
40.2.7 Interrupt Enable Register 0 (INTENB0) 2388
40.2.8 Interrupt Enable Register 1 (INTENB1) 2389
40.2.9 BRDY Interrupt Enable Register (BRDYENB) 2390
40.2.10 NRDY Interrupt Enable Register (NRDYENB) 2391
40.2.11 BEMP Interrupt Enable Register (BEMPENB) 2392
40.2.12 SOF Output Configuration Register (SOFCFG) 2393
40.2.13 Interrupt Status Register 0 (INTSTS0) 2394
40.2.14 Interrupt Status Register 1 (INTSTS1) 2397

40.2.15 BRDY Interrupt Status Register (BRDYSTS) 2400
40.2.16 NRDY Interrupt Status Register (NRDYSTS) 2401
40.2.17 BEMP Interrupt Status Register (BEMPSTS) 2402
40.2.18 Frame Number Register (FRMNUM) 2403
40.2.19 Device State Change Register (DVCHGR) 2404
40.2.20 USB Address Register (USBADDR) 2405
40.2.21 USB Request Type Register (USBREQ) 2406
40.2.22 USB Request Value Register (USBVAL) 2407
40.2.23 USB Request Index Register (USBINDX) 2407
40.2.24 USB Request Length Register (USBLENG) 2408
40.2.25 DCP Configuration Register (DCPCFG) 2409
40.2.26 DCP Maximum Packet Size Register (DCPMAXP) 2410
40.2.27 DCP Control Register (DCPCTR) 2411
40.2.28 Pipe Window Select Register (PIPESEL) 2414
40.2.29 Pipe Configuration Register (PIPECFG) 2415
40.2.30 Pipe Maximum Packet Size Register (PIPEMAXP) 2417
40.2.31 Pipe Cycle Control Register (PIPEPERI) 2418
40.2.32 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9) 2419
40.2.33 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5) 2427
40.2.34 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5) 2428
40.2.35 Device Address n Configuration Register (DEVADDn) (n = 0 to 5) 2429
40.2.36 Deep Standby USB Transceiver Control/Pin Monitoring Register (DPUSR0R) 2430
40.2.37 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R) 2431
40.3 Operation 2433
40.3.1 System Control 2433
40.3.1.1 Setting Data to the USB Related Register 2433
40.3.1.2 Controller Function Selection 2433
40.3.1.3 Controlling USB Data Bus Resistors 2433
40.3.1.4 Example of USB External Connection Circuit 2434
40.3.1.5 Release from Deep Software Standby Mode 1 Due to USB Suspend/Resume Interrupts 2438
40.3.2 Interrupt Sources 2442
40.3.3 Interrupt Descriptions 2444
40.3.3.1 BRDY Interrupt 2444
40.3.3.2 NRDY Interrupt 2448
40.3.3.3 BEMP Interrupt 2451
40.3.3.4 Device State Transition Interrupt 2452
40.3.3.5 Control Transfer Stage Transition Interrupt 2453
40.3.3.6 Frame Update Interrupt 2454
40.3.3.7 VBUS Interrupt 2454
40.3.3.8 Resume Interrupt 2454

40.3.3.9 OVRCR Interrupt 2454
40.3.3.10 BCHG Interrupt 2454
40.3.3.11 DTCH Interrupt 2454
40.3.3.12 SACK Interrupt 2455
40.3.3.13 SIGN Interrupt 2455
40.3.3.14 ATTCH Interrupt 2455
40.3.3.15 EOFERR Interrupt 2455
40.3.4 Pipe Control 2456
40.3.4.1 Pipe Control Register Switching Procedures 2457
40.3.4.2 Transfer Types 2457
40.3.4.3 Endpoint Number 2457
40.3.4.4 Maximum Packet Size Setting 2458
40.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction) 2458
40.3.4.6 Response PID 2459
40.3.4.7 Data PID Sequence Bit 2460
40.3.4.8 Response PID = NAK Function 2460
40.3.4.9 Auto Response Mode 2460
40.3.4.10 OUT-NAK Mode 2460
40.3.4.11 Null Auto Response Mode 2461
40.3.5 FIFO Buffer Memory 2461
40.3.5.1 FIFO Buffer Memory 2461
40.3.5.2 FIFO Buffer Clearing 2462
40.3.5.3 FIFO Port Functions 2463
40.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports) 2464
40.3.6 Control Transfers Using DCP 2465
40.3.6.1 Control Transfers When the Host Controller is Selected 2465
40.3.6.2 Control Transfers When the Function Controller is Selected 2466
40.3.7 Bulk Transfers (Pipes 1 to 5) 2467
40.3.8 Interrupt Transfers (Pipes 6 to 9) 2467
40.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected 2467
40.3.9 Isochronous Transfers (Pipes 1 and 2) 2468
40.3.9.1 Error Detection in Isochronous Transfers 2468
40.3.9.2 Data PID 2469
40.3.9.3 Interval Counter 2469
40.3.10 SOF Interpolation Function 2476
40.3.11 Pipe Schedule 2477
40.3.11.1 Conditions for Generating a Transaction 2477
40.3.11.2 Transfer Schedule 2477
40.3.11.3 Enabling USB Communication 2477
40.4 Usage Notes 2478
40.4.1 Setting the Module-Stop Function 2478

41. USB 2.0 High-Speed Host/Function Module (USBHS) 2479
41.1 Overview 2479
41.2 Register Descriptions 2481
41.2.1 System Configuration Control Register (SYSCFG) 2481
41.2.2 CPU Bus Wait Register (BUSWAIT) 2483
41.2.3 System Configuration Status Register (SYSSTS0) 2484
41.2.4 PLL Status Register (PLLSTA) 2485
41.2.5 Device State Control Register 0 (DVSTCTR0) 2486
41.2.6 USB Test Mode Register (TESTMODE) 2489
41.2.7 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO) 2491
41.2.8 CFIFO Port Select Register (CFIFOSEL) 2494
41.2.9 D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL) 2496
41.2.10 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR) 2498
41.2.11 Interrupt Enable Register 0 (INTENB0) 2500
41.2.12 Interrupt Enable Register 1 (INTENB1) 2501
41.2.13 BRDY Interrupt Enable Register (BRDYENB) 2502
41.2.14 NRDY Interrupt Enable Register (NRDYENB) 2502
41.2.15 BEMP Interrupt Enable Register (BEMPENB) 2503
41.2.16 SOF Output Configuration Register (SOFCFG) 2504
41.2.17 PHY Setting Register (PHYSET) 2505
41.2.18 Interrupt Status Register 0 (INTSTS0) 2507
41.2.19 Interrupt Status Register 1 (INTSTS1) 2509
41.2.20 BRDY Interrupt Status Register (BRDYSTS) 2512
41.2.21 NRDY Interrupt Status Register (NRDYSTS) 2512
41.2.22 BEMP Interrupt Status Register (BEMPSTS) 2513
41.2.23 Frame Number Register (FRMNUM) 2514
41.2.24 Microframe Number Register (UFRMNUM) 2515
41.2.25 USB Address Register (USBADDR) 2516
41.2.26 USB Request Type Register (USBREQ) 2517
41.2.27 USB Request Value Register (USBVAL) 2518
41.2.28 USB Request Index Register (USBINDX) 2518
41.2.29 USB Request Length Register (USBLENG) 2519
41.2.30 Default Control Pipe Configuration Register (DCPCFG) 2520
41.2.31 Default Control Pipe Maximum Packet Size Register (DCPMAXP) 2521
41.2.32 Default Control Pipe Control Register (DCPCTR) 2522
41.2.33 Pipe Window Select Register (PIPESEL) 2526
41.2.34 Pipe Configuration Register (PIPECFG) 2527
41.2.35 Pipe Buffer Register (PIPEBUF) 2530

41.2.36 Pipe Maximum Packet Size Register (PIPEMAXP) 2532
41.2.37 Pipe Cycle Control Register (PIPEPERI) 2533
41.2.38 Pipe n Control Register (PIPEnCTR) (n = 1 to 9) 2534
41.2.39 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5) 2539
41.2.40 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5) 2540
41.2.41 Device Address m Configuration Register (DEVADDm) (m = 0 to A) 2541
41.2.42 Low Power Control Register (LPCTRL) 2542
41.2.43 Low Power Status Register (LPSTS) 2543
41.2.44 Battery Charging Control Register (BCCTRL) 2544
41.2.45 Function L1 Control Register 1 (PL1CTRL1) 2546
41.2.46 Function L1 Control Register 2 (PL1CTRL2) 2548
41.2.47 Host L1 Control Register 1 (HL1CTRL1) 2549
41.2.48 Host L1 Control Register 2 (HL1CTRL2) 2550
41.2.49 Deep Standby USB Transceiver Pin Monitor Register (DPUSR0R) 2551
41.2.50 Deep Standby USB Resume Interrupt Register (DPUSR1R) 2552
41.2.51 Deep Standby USB Resume Interrupt Register 2 (DPUSR2R) 2554
41.2.52 Deep Standby USB Transceiver Control Register (DPUSRCR) 2555
41.3 Operation 2556
41.3.1 System Control 2556
41.3.1.1 Setting Registers Associated with the USBHS 2556
41.3.1.2 Selecting the Controller Function 2556
41.3.2 Controlling the USB Data Bus Resistors 2556
41.3.3 Supplying Clocks 2556
41.3.4 Notes on Stopping Clock 2558
41.3.5 Interrupts 2559
41.3.5.1 Setting the USBHSR Interrupt Signal Output Method 2560
41.3.6 Interrupt Descriptions 2563
41.3.6.1 BRDY Interrupt 2563
41.3.6.2 NRDY Interrupt 2566
41.3.6.3 BEMP Interrupt 2570
41.3.6.4 Device State Transition Interrupt in Function Controller Operation 2571
41.3.6.5 Control Transfer Stage Transition Interrupt in Function Controller Operation 2572
41.3.6.6 Frame Number Refresh Interrupt 2573
41.3.6.7 VBUS Interrupt 2573
41.3.6.8 Resume Interrupt 2573
41.3.6.9 OVRCR Interrupt 2573
41.3.6.10 BCHG Interrupt 2573
41.3.6.11 DTCH Interrupt 2573
41.3.6.12 SACK Interrupt 2574
41.3.6.13 SIGN Interrupt 2574
41.3.6.14 ATTCH Interrupt 2574

41.3.6.15 EOFERR Interrupt 2574
41.3.6.16 Portable Device Detection Interrupt 2574
41.3.6.17 LPMEND Interrupt 2574
41.3.6.18 L1RSMEND Interrupt 2574
41.3.7 Pipe Control 2575
41.3.7.1 Pipe Control Register Switching Procedures 2576
41.3.7.2 Transfer Types 2577
41.3.7.3 Endpoint Number 2577
41.3.7.4 Setting the Maximum Packet Size 2577
41.3.7.5 Transaction Counter (For Pipes 1 to 5 in the Reading Direction) 2577
41.3.7.6 Response PID 2578
41.3.7.7 Data PID Sequence Bit 2579
41.3.7.8 Response PID = NAK Function 2579
41.3.7.9 Auto Response Mode 2579
41.3.7.10 OUT-NAK Mode 2579
41.3.7.11 Null Auto Response Mode 2580
41.3.8 FIFO Buffers 2581
41.3.8.1 FIFO Buffers 2581
41.3.8.2 Clearing the FIFO Buffers 2582
41.3.8.3 FIFO Port Functions 2582
41.3.8.4 DMA/DTC Transfers (D0FIFO and D1FIFO Ports) 2584
41.3.8.5 Allocating the FIFO Buffers 2585
41.3.9 Control Transfer Using the Default Control Pipe 2586
41.3.9.1 Control Transfer in Host Controller Operation 2586
41.3.9.2 Control Transfer in Function Controller Operation 2587
41.3.10 Bulk Transfer for Pipes 1 to 5 2588
41.3.10.1 Controlling the PING Token in Host Controller Operation 2588
41.3.10.2 Controlling the NYET Handshake in Function Controller Operation 2589
41.3.11 Interrupt Transfer for Pipes 6 to 9 2590
41.3.11.1 Interval Counter for Interrupt Transfer in Host Controller Operation 2590
41.3.12 Isochronous Transfer for Pipes 1 and 2 2591
41.3.12.1 Error Detection in Isochronous Transfers 2591
41.3.12.2 Data PID 2592
41.3.12.3 Interval Counter 2592
41.3.13 SOF Recovery Function 2599
41.3.14 Pipe Schedule 2600
41.3.14.1 Conditions for Generating a Transaction 2600
41.3.14.2 Transfer Schedule 2600
41.3.14.3 Enabling USB Communication 2600
41.3.15 Battery Charging Detection Processing 2601
41.3.15.1 Processing in Function Controller Operation 2601

41.3.15.2 Processing in Host Controller Operation 2603
41.3.16 Link Power Management Processing 2606
41.3.16.1 Processing in Function Controller Operation 2606
41.3.16.2 Processing in Host Controller Operation 2607
41.3.17 USB External Connection Circuit 2608
41.3.18 Release from Deep Software Standby Mode 1 in Response to the USBHS Resume Interrupt 2612
41.4 Notes on Using the USBHS 2615
41.4.1 Setting the USBHS Module-Stop Function 2615
41.4.2 Setting to Transition to Deep Software Standby Mode 2615
42. Serial Communications Interface (RSCI) 2616
42.1 Overview 2616
42.2 Register Descriptions 2625
42.2.1 Receive Shift Register (RSR) 2625
42.2.2 Receive Data Register (RDR) 2625
42.2.3 Transmit Data Register (TDR) 2627
42.2.4 Transmit Shift Register (TSR) 2628
42.2.5 Control Register 0 (SCR0) 2629
42.2.6 Control Register 1 (SCR1) 2632
42.2.7 Control Register 2 (SCR2) 2637
42.2.8 Control Register 3 (SCR3) 2653
42.2.9 Control Register 4 (SCR4) 2658
42.2.10 I^2C Mode Register (SIMR) 2660
42.2.11 FIFO Control Register (FCR) 2663
42.2.12 Manchester Mode Control Register (MMCR) 2665
42.2.13 DE Signal Control Register (DECR) 2669
42.2.14 Extended Serial Mode Control Register 0 (XCR0) 2670
42.2.15 Extended Serial Mode Control Register 1 (XCR1) 2673
42.2.16 Extended Serial Mode Control Register 2 (XCR2) 2675
42.2.17 Status Register (SSR) 2676
42.2.18 I^2C Status Register (SISR) 2682
42.2.19 Receive FIFO Status Register (RFSR) 2683
42.2.20 Transmit FIFO Status Register (TFSR) 2685
42.2.21 Manchester Mode Status Register (MMSR) 2686
42.2.22 Extended Serial Mode Status Register 0 (XSR0) 2689
42.2.23 Extended Serial Mode Status Register 1 (XSR1) 2691
42.2.24 Status Clear Register (SSCR) 2692
42.2.25 I^2C Status Clear Register (SISCR) 2693
42.2.26 Receive FIFO Status Clear Register (RFSCR) 2693
42.2.27 Manchester Mode Status Clear Register (MMSCR) 2694
42.2.28 Extended Serial Mode Status Clear Register (XSCR) 2695

42.2.29 Internal Signal Status Register (SISSR) 2696
42.2.30 HBS Support Mode Control Register (HBSCR) 2697
42.2.31 RXD Input Path Select Register (RXDSEL) 2698
42.2.32 GTIOC Input Path Select Register (GTIOCSEL) 2698
42.3 Operation in Asynchronous Mode 2699
42.3.1 Serial Data Transfer Format 2699
42.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 2701
42.3.3 Clock 2703
42.3.4 Double-Speed Mode, Divide-by-6 Mode, and Divided-by-4 Mode 2703
42.3.5 CTS and RTS Functions 2704
42.3.6 Data Match Detection 2705
42.3.7 RSCI Initialization (Asynchronous Mode) 2708
42.3.8 Serial Data Transmission (Asynchronous Mode) 2712
42.3.9 Serial Data Reception (Asynchronous Mode) 2718
42.3.10 Receive Data Sampling Timing Adjustment 2726
42.3.11 Transmit Data Transition Timing Adjustment 2727
42.4 Multi-Processor Communication Function 2728
42.4.1 Multi-Processor Serial Data Transmission 2730
42.4.2 Multi-Processor Serial Data Reception 2733
42.5 Manchester Mode 2739
42.5.1 Frame Format 2739
42.5.2 Clock 2743
42.5.3 RSCI Initialization of Manchester Mode 2743
42.5.4 Double Speed Operation 2744
42.5.5 CTS, RTS Functions 2745
42.5.6 Serial Data Transmission in Manchester Mode 2745
42.5.7 Serial Data Reception in Manchester Mode 2749
42.5.8 Operation When Multi-Processor Bit is Used 2753
42.5.9 Receive Retiming 2753
42.5.10 Polarity Setting for Manchester Code 2754
42.5.11 Errors in Manchester Mode 2756
42.6 HBS Support Mode 2761
42.6.1 Reception in HBS Support Mode 2761
42.6.2 Transmission in HBS Support Mode 2762
42.6.3 Register Setting for HBS Support Mode 2764
42.7 Smart Card Interface Mode 2765
42.7.1 Sample Connection 2765
42.7.2 Data Format (Except in Block Transfer Mode) 2766
42.7.3 Block Transfer Mode 2767
42.7.4 Receive Data Sampling Timing and Reception Margin 2768
42.7.5 RSCI Initialization (Smart Card Interface Mode) 2769

42.7.6 Serial Data Transmission (Except in Block Transfer Mode) 2771
42.7.7 Serial Data Reception (Except in Block Transfer Mode) 2774
42.7.8 Clock Output Control 2776
42.8 Extended Serial Mode 2777
42.8.1 Serial Transfer Protocol 2777
42.8.2 Transmitting a Start Frame 2778
42.8.3 Receiving a Start Frame 2781
42.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used) 2781
42.8.3.2 Priority Interrupt Bit 2786
42.8.4 Detection of Bus Collisions 2787
42.8.5 Bit Rate Measurement 2789
42.9 Operation in Simple I2C Mode 2790
42.9.1 Generation of Start, Restart, and Stop Conditions 2791
42.9.2 Clock Synchronization 2793
42.9.3 SDA Output Delay 2794
42.9.4 RSCI Initialization (Simple I2C Mode) 2795
42.9.5 Operation in Master Transmission (Simple I2C Mode) 2796
42.9.6 Master Reception (Simple I2C Mode) 2801
42.10 Operation in Clock Synchronous Mode 2804
42.10.1 Clock 2804
42.10.2 CTS and RTS Functions 2805
42.10.3 RSCI Initialization (Clock Synchronous Mode) 2806
42.10.4 Serial Data Transmission (Clock Synchronous Mode) 2807
42.10.5 Serial Data Reception (Clock Synchronous Mode) 2813
42.10.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) 2818
42.11 Operation in Simple SPI Mode 2820
42.11.1 Clock 2820
42.11.2 States of Pins in Master and Slave Modes 2821
42.11.3 SS Function in Master Mode 2821
42.11.4 SS Function in Slave Mode 2821
42.11.5 Relationship between Clock and Transmit/Receive Data 2822
42.11.6 RSCI Initialization (Simple SPI Mode) 2823
42.11.7 Transmission and Reception of Serial Data (Simple SPI Mode) 2823
42.12 Bit Rate Modulation Function 2824
42.13 Noise Cancellation Function 2825
42.14 RS-485 Driver Control Function 2826
42.15 Loopback Function 2827
42.16 Half-Duplex Communication Function 2828
42.17 Synchronizer Bypass Function 2828
42.18 Interrupt Signal 2829
42.18.1 Buffer Operations for TXI and RXI Interrupts 2829

42.18.2 Interrupt in Asynchronous Mode, Manchester Mode, Clock Synchronous Mode, and Simple SPI Mode ... 2830
42.18.3 Interrupt in Smart Card Interface Mode ... 2832
42.18.4 Interrupts in Simple I2C Mode ... 2833
42.18.5 Interrupts in Extended Serial Mode ... 2834
42.19 Event Linking ... 2835
42.20 Usage Notes ... 2837
42.20.1 Setting the Module Stop Function ... 2837
42.20.2 RSCI Operations during Low Power Consumption State ... 2837
42.20.3 Break Detection and Processing ... 2841
42.20.4 Mark State and Sending Breaks ... 2841
42.20.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode) ... 2841
42.20.6 Writing Data to the TDR Register ... 2841
42.20.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode) ... 2842
42.20.8 Restrictions on Using DMAC or DTC ... 2844
42.20.9 Notes on Starting Transfer ... 2844
42.20.10 Limitations on Simple SPI Mode ... 2845
42.20.11 Note on Transmit Enable Bit (TE Bit) ... 2846
42.20.12 Note on Extended Serial Mode ... 2846
42.20.13 Notes on RS-485 Driver Control Function ... 2846
42.20.14 Notes on Loopback Function ... 2846
42.20.15 Note on Register Access When the Frequency of the Operating Clock (TCLK) is Lower Than the Frequency of the Peripheral Module Clock (PCLK) ... 2846
42.20.16 Note on Aborting Operation ... 2846
43. I3C Bus Interface (RI3C) ... 2847
43.1 Overview ... 2847
43.2 Register Descriptions ... 2852
43.2.1 Mode Register (ICMR) ... 2852
43.2.2 Control Register (ICCR) ... 2853
43.2.3 Controller Device Address Register (ICCAR) ... 2855
43.2.4 Reset Control Register (ICRCR) ... 2856
43.2.5 Operating Mode Monitor Register (ICMMR) ... 2858
43.2.6 Internal Status Register (ICISR) ... 2860
43.2.7 Internal Status Detection Enable Register (ICISER) ... 2861
43.2.8 Internal Status Interrupt Enable Register (ICISIER) ... 2861
43.2.9 Device Characteristics Table Index Register (ICDCTIR) ... 2862
43.2.10 IBI Notify Control Register (ICINCR) ... 2863
43.2.11 Function Enable Register (ICFER) ... 2864
43.2.12 Target Mode Control Register (ICTCR) ... 2866
43.2.13 Slave Mode Control Register (ICSCR) ... 2867

43.2.14 Reference Clock Control Register (ICRCCR) 2869
43.2.15 Standard Bitrate Register (ICSBR) 2870
43.2.16 F/S-Mode Bitrate Register (ICFBR) 2872
43.2.17 Extended Bitrate Register (ICEBR) 2874
43.2.18 Hs-Mode Bitrate Register (ICHBR) 2876
43.2.19 Bus Free Time Setting Register (ICBFTR) 2877
43.2.20 Bus Available Time Setting Register (ICBATR) 2878
43.2.21 Bus Idle Time Setting Register (ICBITR) 2878
43.2.22 Output Signal Control Register (ICOCR) 2879
43.2.23 Input Signal Control Register (ICICR) 2881
43.2.24 Timeout Control Register (ICTOR) 2882
43.2.25 Wake-Up Unit Control Register (ICWUCR) 2884
43.2.26 Acknowledgment Bit Control Register (ICACKR) 2886
43.2.27 Clock Stretch Control Register (ICCSCR) 2887
43.2.28 Clock Stall Control Register (ICSTCR) 2888
43.2.29 Target Transmit/Receive Data Length Register (ICTDLR) 2889
43.2.30 Synchronous Timing Control Register (ICSYTCR) 2890
43.2.31 Asynchronous Timing Control Register (ICASTCR) 2890
43.2.32 Asynchronous Timing Trigger Register (ICASTTR) 2891
43.2.33 Asynchronous Timing Counter Control Register (ICASTCCR) 2891
43.2.34 Condition Generation Request Register (ICCGR) 2892
43.2.35 Command Queue Register (ICCQR) 2894
43.2.36 Response Queue Register (ICRQR) 2894
43.2.37 Transmit/Receive Data Register (ICDR) 2895
43.2.38 IBI Queue Register (ICIQR) 2897
43.2.39 Receive Status Queue Register (ICSQR) 2897
43.2.40 High-Priority Command Queue Register (ICHCQR) 2898
43.2.41 High-Priority Response Queue Register (ICHRQR) 2898
43.2.42 High-Priority Transmit/Receive Data Register (ICHDR) 2899
43.2.43 Queue Buffer Threshold Control Register (ICQBTCR) 2900
43.2.44 Data Buffer Threshold Control Register (ICDBTCR) 2902
43.2.45 Receive Status Queue Threshold Control Register (ICSQTCR) 2903
43.2.46 High-Priority Queue Buffer Threshold Control Register (ICHQBTCR) 2904
43.2.47 High-Priority Data Buffer Threshold Control Register (ICHDBTCR) 2905
43.2.48 Status Register 2 (ICSR2) 2907
43.2.49 Status Detection Enable Register (ICSER) 2910
43.2.50 Status Interrupt Enable Register (ICSIER) 2911
43.2.51 Communication Status Register (ICCSR) 2912
43.2.52 Communication Status Detection Enable Register (ICCSER) 2916
43.2.53 Communication Status Interrupt Enable Register (ICCSIER) 2917
43.2.54 High-Priority Communication Status Register (ICHCSR) 2918

43.2.55 High-Priority Communication Status Detection Enable Register (ICHCSER) 2920
43.2.56 High-Priority Communication Status Interrupt Enable Register (ICHCSIER) 2921
43.2.57 Bus Status Register (ICBSR) 2922
43.2.58 Slave Mode Status Register (ICSSR) 2924
43.2.59 Wake-Up Unit Status Register (ICWUSR) 2927
43.2.60 C_SyncCNT Capture Register (ICCSCCR) 2928
43.2.61 Target Device Address Table Register m (ICTDATRm) (m = 0 to 7) 2929
43.2.62 Extended Target Device Address Table Register (ICEDATR) 2930
43.2.63 Device Address Register y (ICDARy) (y = 0 to 2) 2931
43.2.64 Slave Address Register y (SARy) (y = 0 to 2) 2932
43.2.65 Target Device Characteristics Table Register m (ICTDCTRm) (m = 0 to 7) 2933
43.2.66 Device Characteristics Table Register (ICDCTR) 2935
43.2.67 Provisioned ID Low Register (ICPIDLR) 2937
43.2.68 Provisioned ID High Register (ICPIDHR) 2937
43.2.69 Device Address Monitor Register y (ICDAMRy) (y = 0 to 2) 2938
43.2.70 Slave Address Monitor Register y (SAMRy) (y = 0 to 2) 2940
43.2.71 Target Event Register (ICTEVR) 2942
43.2.72 Activity State Register (ICASR) 2943
43.2.73 Max Write Length Register (ICMWLR) 2945
43.2.74 Max Read Length Register (ICMRLR) 2946
43.2.75 Test Mode Register (ICTMR) 2947
43.2.76 Device Status Register (ICDSR) 2948
43.2.77 Max Write Speed Register (ICMWSR) 2949
43.2.78 Max Read Speed Register (ICMRSR) 2950
43.2.79 Maximum Read Turnaround Time Register (ICMTTR) 2951
43.2.80 Timing Support Information Register (ICTSIR) 2952
43.2.81 Timing Support Information Register 2 (ICTSIR2) 2953
43.2.82 Bit Count Register (ICBCR) 2954
43.2.83 Queue Buffer Status Register (ICQBSR) 2955
43.2.84 Data Buffer Status Register (ICDBSR) 2955
43.2.85 Receive Status Queue Status Register (ICSQSR) 2956
43.2.86 High-Priority Queue Buffer Status Register (ICHQBSR) 2956
43.2.87 High-Priority Data Buffer Status Register (ICHDBSR) 2957
43.2.88 Internal Status Monitor Register (ICIMR) 2957
43.2.89 Controller Error Count Register (ICCECR) 2958
43.2.90 T_C1 Capture Register (ICTC1CR) 2958
43.2.91 T_C2 Capture Register (ICTC2CR) 2959
43.2.92 Communication Clock Control Register (ICCCR) 2959
43.3 Data Structures 2960
43.3.1 Command Descriptor 2960
43.3.1.1 Address Assignment Command 2961

43.3.1.2 Immediate Data Transfer Command 2963
43.3.1.3 Regular Data Transfer Command 2965
43.3.1.4 Combo Transfer Command 2968
43.3.1.5 Internal Control Command 2970
43.3.2 Response Descriptor 2971
43.3.3 IBI Status Descriptor 2974
43.3.4 Receive Status Descriptor 2976
43.4 Operation 2978
43.4.1 Transfer Methods 2978
43.4.2 I3C Protocol 2981
43.4.2.1 Communication Protocol 2981
43.4.2.2 Bus Conditions 2985
43.4.3 I2C Protocol 2986
43.4.4 Initial Settings 2987
43.4.5 I3C Communication Flow 2989
43.4.6 I3C Controller Operation 2991
43.4.6.1 Dynamic Address Assignment Procedure 2991
43.4.6.2 SDR Write Transfer 2993
43.4.6.3 SDR Read Transfer 2995
43.4.6.4 IBI Transfer 2997
43.4.6.5 I3C Controller Transmission Flow (FIFO Buffer Transfer) 3000
43.4.6.6 I3C Controller Transmission Flow (High-Priority FIFO Buffer Transfer) 3001
43.4.6.7 I3C Controller Reception Flow (FIFO Buffer Transfer) 3002
43.4.6.8 I3C Controller Reception Flow (High-Priority FIFO Buffer Transfer) 3003
43.4.6.9 I3C Controller IBI Reception Flow 3004
43.4.7 I3C Target Operation 3005
43.4.7.1 Dynamic Address Assignment Procedure 3005
43.4.7.2 SDR Write Transfer 3007
43.4.7.3 SDR Read Transfer 3009
43.4.7.4 IBI Transfer 3011
43.4.7.5 I3C Target Transmission Flow (FIFO Buffer Transfer) 3013
43.4.7.6 I3C Target Reception Flow (FIFO Buffer Transfer) 3015
43.4.7.7 I3C Target IBI Transmission Flow 3016
43.4.8 I2C Master Operation 3017
43.4.8.1 Master Transmit Operation 3017
43.4.8.2 Master Receive Operation 3021
43.4.9 I2C Slave Operation 3026
43.4.9.1 Slave Transmit Operation 3026
43.4.9.2 Slave Receive Operation 3029
43.5 Details of Function 3032
43.5.1 SCL Synchronization Circuit (I2C Protocol Mode) 3032

43.5.2 SDA Output Delay Function (I^2C Protocol Mode) 3033
43.5.3 Digital Noise Filters (I^2C Protocol Mode) 3034
43.5.4 Address Match Detection (I^2C Protocol Mode) 3035
43.5.4.1 Slave-Address Match Detection 3035
43.5.4.2 General Call Address Detection 3037
43.5.4.3 Device-ID Address Detection 3038
43.5.4.4 Hs-Mode Master Code Detection 3040
43.5.4.5 Host Address Detection 3042
43.5.5 Automatic Low-Hold Function for SCL (I^2C Protocol Mode) 3043
43.5.5.1 Function to Prevent Wrong Transmission of Transmit Data 3043
43.5.5.2 NACK Reception Transfer Suspension Function 3044
43.5.5.3 Function to Prevent Failure to Receive Data 3045
43.5.6 Arbitration-Lost Detection Functions (I^2C Protocol Mode) 3047
43.5.6.1 Master Arbitration-Lost Detection Function (MALE Bit) 3047
43.5.6.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit) 3049
43.5.6.3 Slave Arbitration-Lost Detection Function (SALE Bit) 3050
43.5.7 START Condition/Repeated START Condition/STOP Condition Generating Function (I^2C Protocol Mode) 3051
43.5.7.1 Generating a START Condition 3051
43.5.7.2 Generating a Repeated START Condition 3051
43.5.7.3 Generating a STOP Condition 3054
43.5.8 Bus Hanging (I^2C Protocol Mode) 3055
43.5.8.1 Timeout Function 3055
43.5.8.2 Additional SCL Output Function 3056
43.5.9 Wake-Up Function (I^2C Protocol Mode) 3057
43.5.9.1 Normal Wake-Up Mode 1 3057
43.5.9.2 Normal Wake-Up Mode 2 3059
43.5.9.3 Command Recovery Mode and EEP Response Mode 3061
43.5.9.4 Notes on Using Wake-Up Function 3063
43.5.10 SMBus Operation (I^2C Protocol Mode) 3064
43.5.10.1 SMBus Timeout Measurement 3064
43.5.10.2 Packet Error Code (PEC) 3065
43.5.10.3 SMBus Host Notification Protocol (Notify ARP Master Command) 3065
43.5.11 CCC detection function (I3C protocol mode) 3066
43.5.12 Clock Stalling (I3C Protocol Mode) 3068
43.5.13 IBI (I3C Protocol Mode) 3072
43.5.13.1 Target Interrupt Request 3072
43.5.13.2 Controller Role Request 3075
43.5.14 Timing Control (I3C Protocol Mode) 3078
43.5.14.1 Sync Mode 3078
43.5.14.2 Async Mode 0 (Asynchronous Basic Mode) 3079

43.5.14.3 Async Mode 1 (Asynchronous Advanced Mode) ... 3079
43.5.15 Common Command Codes (CCC) (I3C Protocol Mode) ... 3081
43.5.16 Error Detection (I3C Protocol Mode) ... 3081
43.5.16.1 SDR Error Detection and Recovery Methods for I3C Target Devices ... 3081
43.5.16.2 SDR Error Detection and Recovery Methods for I3C Controller Devices ... 3082
43.5.16.3 Timeout Detection ... 3082
43.5.16.4 Resume Operation ... 3083
43.5.16.5 Abort Operation ... 3083
43.5.16.6 Error Recovery Operation ... 3084
43.5.17 Wake-Up Function (I3C Protocol Mode) ... 3086
43.5.17.1 I3C Controller Wake-Up Function ... 3086
43.5.17.2 I3C Target Wake-Up Function ... 3088
43.5.17.3 Notes on Using Wake-Up Function ... 3091
43.6 Interrupt Sources ... 3092
43.6.1 Buffer Operation for TXI and RXI Interrupts ... 3093
43.7 Event Link Function ... 3094
43.7.1 Interrupt Handling and Event Linking ... 3094
43.8 Reset Description ... 3095
43.9 Usage Notes ... 3102
43.9.1 Settings of the Module Stop Function ... 3102
44. High-Speed I2C-bus Interface (RIICHS) ... 3103
44.1 Overview ... 3103
44.2 Register Descriptions ... 3106
44.2.1 Control Register (ICCR) ... 3106
44.2.2 Reset Control Register (ICRCR) ... 3107
44.2.3 Operating Mode Monitor Register (ICMMR) ... 3108
44.2.4 Function Enable Register (ICFER) ... 3110
44.2.5 Slave Mode Control Register (ICSCR) ... 3112
44.2.6 Reference Clock Control Register (ICRCCR) ... 3114
44.2.7 F/S-Mode Bitrate Register (ICFBR) ... 3115
44.2.8 Hs-Mode Bitrate Register (ICHBR) ... 3117
44.2.9 Bus Free Time Setting Register (ICBFTR) ... 3118
44.2.10 Output Signal Control Register (ICOCR) ... 3119
44.2.11 Input Signal Control Register (ICICR) ... 3121
44.2.12 Timeout Control Register (ICTOR) ... 3122
44.2.13 Wake-Up Unit Control Register (ICWUCR) ... 3124
44.2.14 Acknowledgment Bit Control Register (ICACKR) ... 3126
44.2.15 Clock Stretch Control Register (ICCSCR) ... 3127
44.2.16 Condition Generation Request Register (ICCGR) ... 3128
44.2.17 Transmit/Receive Data Register (ICDR) ... 3130
44.2.18 Status Register 2 (ICSR2) ... 3131

44.2.19 Status Detection Enable Register (ICSER) 3134
44.2.20 Status Interrupt Enable Register (ICSIER) 3135
44.2.21 Communication Status Register (ICCSR) 3136
44.2.22 Communication Status Detection Enable Register (ICCSER) 3137
44.2.23 Communication Status Interrupt Enable Register (ICCSIER) 3138
44.2.24 Bus Status Register (ICBSR) 3139
44.2.25 Slave Mode Status Register (ICSSR) 3140
44.2.26 Wake-Up Unit Status Register (ICWUSR) 3143
44.2.27 Slave Address Register y (SARy) (y = 0 to 2) 3144
44.2.28 Slave Address Monitor Register y (SAMRy) (y = 0 to 2) 3145
44.2.29 Bit Count Register (ICBCR) 3147
44.2.30 Internal Status Monitor Register (ICIMR) 3148
44.3 Operation 3149
44.3.1 Communication Data Format 3149
44.3.2 Initial Settings 3150
44.3.3 I^2C Master Operation 3151
44.3.3.1 Master Transmit Operation 3151
44.3.3.2 Master Receive Operation 3155
44.3.4 I^2C Slave Operation 3160
44.3.4.1 Slave Transmit Operation 3160
44.3.4.2 Slave Receive Operation 3163
44.4 Details of Function 3166
44.4.1 SCL Synchronization Circuit 3166
44.4.2 SDA Output Delay Function 3167
44.4.3 Digital Noise Filters 3168
44.4.4 Address Match Detection 3169
44.4.4.1 Slave-Address Match Detection 3169
44.4.4.2 General Call Address Detection 3171
44.4.4.3 Device-ID Address Detection 3172
44.4.4.4 Hs-Mode Master Code Detection 3174
44.4.4.5 Host Address Detection 3176
44.4.5 Automatic Low-Hold Function for SCL 3177
44.4.5.1 Function to Prevent Wrong Transmission of Transmit Data 3177
44.4.5.2 NACK Reception Transfer Suspension Function 3178
44.4.5.3 Function to Prevent Failure to Receive Data 3179
44.4.6 Arbitration-Lost Detection Functions 3181
44.4.6.1 Master Arbitration-Lost Detection Function (MALE Bit) 3181
44.4.6.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit) 3183
44.4.6.3 Slave Arbitration-Lost Detection Function (SALE Bit) 3184
44.4.7 START Condition/Repeated START Condition/STOP Condition Generating Function 3185
44.4.7.1 Generating a START Condition 3185

44.4.7.2 Generating a Repeated START Condition ... 3185
44.4.7.3 Generating a STOP Condition ... 3188
44.4.8 Bus Hanging ... 3189
44.4.8.1 Timeout Function ... 3189
44.4.8.2 Additional SCL Output Function ... 3190
44.4.9 Wake-Up Function ... 3191
44.4.9.1 Normal Wake-Up Mode 1 ... 3191
44.4.9.2 Normal Wake-Up Mode 2 ... 3193
44.4.9.3 Command Recovery Mode and EEP Response Mode ... 3195
44.4.9.4 Notes on Using Wake-Up Function ... 3197
44.4.10 SMBus Operation ... 3198
44.4.10.1 SMBus Timeout Measurement ... 3198
44.4.10.2 Packet Error Code (PEC) ... 3199
44.4.10.3 SMBus Host Notification Protocol (Notify ARP Master Command) ... 3199
44.5 Interrupt Sources ... 3200
44.5.1 Buffer Operation for TXI and RXI Interrupts ... 3200
44.6 Event Link Function ... 3201
44.6.1 Interrupt Handling and Event Linking ... 3201
44.7 Reset Description ... 3202
44.8 Usage Notes ... 3204
44.8.1 Settings of the Module Stop Function ... 3204
45. CAN FD Module with Gateway Functionality (CANFDG) ... 3205
45.1 Overview ... 3205
45.2 Register Descriptions ... 3208
45.2.1 Nominal Bit Rate Configuration Register (NBCR) ... 3208
45.2.2 Channel Control Register (CHCR) ... 3210
45.2.3 Channel Status Register (CHSR) ... 3214
45.2.4 Channel Error Status Register (CHESR) ... 3217
45.2.5 Data Bit Rate Configuration Register (DBCR) ... 3222
45.2.6 CAN FD Configuration Register (FDCFG) ... 3224
45.2.7 CAN FD Control Register (FDCTR) ... 3228
45.2.8 CAN FD Status Register (FDSTS) ... 3229
45.2.9 CAN FD CRC Register (FDCRC) ... 3231
45.2.10 Global Configuration Register (GCFG) ... 3232
45.2.11 Global Control Register (GCR) ... 3234
45.2.12 Global Status Register (GSR) ... 3236
45.2.13 Global Error Status Register (GESR) ... 3237
45.2.14 Transmit Interrupt Status Register (TISR) ... 3239
45.2.15 Timestamp Counter Register (TSCR) ... 3243
45.2.16 Acceptance Filter List Control Register (AFCR) ... 3244
45.2.17 Acceptance Filter List Configuration Register (AFCFG) ... 3245

45.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15) 3246
45.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15) 3247
45.2.20 Acceptance Filter List n Pointer Register 0 (AFLn.PTR0) (n = 0 to 15) 3248
45.2.21 Acceptance Filter List n Pointer Register 1 (AFLn.PTR1) (n = 0 to 15) 3250
45.2.22 Receive Message Buffer Configuration Register (RMCR) 3252
45.2.23 Receive Message Buffer New Data Register 0 (RMNDR0) 3253
45.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0 to 7) 3254
45.2.25 Receive FIFO n Status Register (RFSRn) (n = 0 to 7) 3256
45.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0 to 7) 3258
45.2.27 Common FIFO n Configuration Register (CFCRn) (n = 0 to 5) 3259
45.2.28 Common FIFO n Extended Configuration Register (CFECRn) (n = 0 to 5) 3262
45.2.29 Common FIFO n Status Register (CFSRn) (n = 0 to 5) 3264
45.2.30 Common FIFO n Pointer Control Register (CFPCRn) (n = 0 to 5) 3268
45.2.31 FIFO Empty Status Register (FESR) 3269
45.2.32 FIFO Full Status Register (FFSR) 3271
45.2.33 FIFO Message Lost Status Register (FMLSR) 3273
45.2.34 Receive FIFO Interrupt Status Register (RFISR) 3274
45.2.35 Common FIFO Receive Interrupt Status Register (CFRISR) 3276
45.2.36 Common FIFO Transmit Interrupt Status Register (CFTISR) 3277
45.2.37 FIFO Full Interrupt Status Register (FFISR) 3278
45.2.38 Common FIFO Message Overwrite Status Register (CFOWSR) 3279
45.2.39 Common FIFO One Frame Receive Interrupt Status Register (CFORISR) 3280
45.2.40 Common FIFO One Frame Transmit Interrupt Status Register (CFOTISR) 3281
45.2.41 DMA Transfer Control Register (DTCR) 3282
45.2.42 DMA Transfer Status Register (DTSR) 3284
45.2.43 Transmit DMA Transfer Control Register (TDTCR) 3286
45.2.44 Transmit DMA Transfer Status Register (TDTSR) 3288
45.2.45 Receive Interrupt Status Register (RISR) 3289
45.2.46 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3291
45.2.47 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3293
45.2.48 Transmit Message Buffer Transmission Request Status Register m (TMTRSRm) (m = 0 to 3) 3294
45.2.49 Transmit Message Buffer Transmission Abort Request Status Register m (TMARSRm) (m = 0 to 3) 3296
45.2.50 Transmit Message Buffer Transmission Completion Status Register m (TMTCSRm) (m = 0 to 3) 3297
45.2.51 Transmit Message Buffer Transmission Abort Status Register m (TMTASRm) (m = 0 to 3) 3298
45.2.52 Transmit Message Buffer Interrupt Enable Register m (TMIERm) (m = 0 to 3) 3299
45.2.53 Transmit Queue n Configuration Register (TQCRn) (n = 0 to 3) 3300
45.2.54 Transmit Queue n Status Register (TQSRn) (n = 0 to 3) 3303

45.2.55 Transmit Queue n Pointer Control Register (TQPCRn) (n = 0 to 3) 3306
45.2.56 Transmit Queue Empty Status Register (TQESR) 3307
45.2.57 Transmit Queue Full Interrupt Status Register (TQFISR) 3308
45.2.58 Transmit Queue Message Lost Status Register (TQMLSR) 3309
45.2.59 Transmit Queue Interrupt Status Register (TQISR) 3310
45.2.60 Transmit Queue One Frame Transmit Interrupt Status Register (TQOTISR) 3311
45.2.61 Transmit Queue One Frame Receive Interrupt Status Register (TQORISR) 3312
45.2.62 Transmit Queue Full Status Register (TQFSR) 3313
45.2.63 Transmission History Configuration Register (THCR) 3314
45.2.64 Transmission History Status Register (THSR) 3316
45.2.65 Transmission History Access Register 0 (THACR0) 3318
45.2.66 Transmission History Access Register 1 (THACR1) 3319
45.2.67 Transmission History Pointer Control Register (THPCR) 3320
45.2.68 Global Reset Control Register (GRCR) 3321
45.2.69 Global Test Mode Configuration Register (GTMCR) 3322
45.2.70 Global Test Mode Enable Register (GTMER) 3323
45.2.71 Global CAN FD Configuration Register (GFDCFG) 3324
45.2.72 Global Test Mode Lock Key Register (GTMLKR) 3325
45.2.73 RAM Test Page Access Register k (RTPARk) (k = 0 to 63) 3325
45.2.74 Bus Load Control Register (BLCCR) 3326
45.2.75 Bus Load Counter Register (BLCR) 3327
45.2.76 Acceptance Filter List Ignore Entry Setting Register (AFIGSR) 3328
45.2.77 Acceptance Filter List Ignore Entry Enable Register (AFIGER) 3329
45.2.78 Identifier Bits Alignment 3330
45.2.79 Message Buffer Structure 3331
45.2.79.1 Start Addresses 3331
45.2.79.2 Receive Message Buffer n (RMBn) (n = 0 to 31) 3332
45.2.79.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 31) 3333
45.2.79.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 31) 3334
45.2.79.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 31) 3335
45.2.79.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 31; p = 0 to 15) 3337
45.2.79.7 Receive Message Buffer n Data k (RMBn.DATAk) (n = 0 to 31; k = 0 to 63) 3338
45.2.79.8 Receive FIFO n (RFBn) (n = 0 to 7) 3339
45.2.79.9 Receive FIFO n Header Field 0 (RFBn.HF0) (n = 0 to 7) 3340
45.2.79.10 Receive FIFO n Header Field 1 (RFBn.HF1) (n = 0 to 7) 3341
45.2.79.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0 to 7) 3342
45.2.79.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0 to 7; p = 0 to 15) 3343
45.2.79.13 Receive FIFO n Data k (RFBn.DATAk) (n = 0 to 7; k = 0 to 63) 3343
45.2.79.14 Common FIFO n (CFBn) (n = 0 to 5) 3344
45.2.79.15 Common FIFO n Header Field 0 (CFBn.HF0) (n = 0 to 5) 3345

45.2.79.16 Common FIFO n Header Field 1 (CFBn.HF1) (n = 0 to 5) 3346
45.2.79.17 Common FIFO n Header Field 2 (CFBn.HF2) (n = 0 to 5) 3347
45.2.79.18 Common FIFO n Data Field p (CFBn.DFp) (n = 0 to 5; p = 0 to 15) 3349
45.2.79.19 Common FIFO n Data k (CFBn.DATAk) (n = 0 to 5; k = 0 to 63) 3349
45.2.79.20 Transmit Message Buffer n (TMBn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3350
45.2.79.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3351
45.2.79.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3352
45.2.79.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103) 3353
45.2.79.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103; p = 0 to 15) 3355
45.2.79.25 Transmit Message Buffer n Data k (TMBn.DATAk) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103; k = 0 to 63) 3356
45.2.80 ECC Control/Status Register (AFLRAM0.ECCSR, AFLRAM1.ECCSR, MBRAM.ECCSR) 3357
45.2.81 ECC Test Mode Register (AFLRAM0.ECTMR, AFLRAM1.ECTMR, MBRAM.ECTMR) 3361
45.2.82 ECC Decoder Test Data Register (AFLRAM0.ECTDR, AFLRAM1.ECTDR, MBRAM.ECTDR) 3362
45.2.83 ECC Error Address Register (AFLRAM0.ECEAR, AFLRAM1.ECEAR, MBRAM.ECEAR) 3362
45.3 Operating Mode 3363
45.3.1 Global Modes 3363
45.3.1.1 GL_SLEEP Mode 3364
45.3.1.2 GL_RESET Mode 3365
45.3.1.3 GL_HALT Mode 3366
45.3.1.4 GL_OPERATION Mode 3368
45.3.2 Channel Modes 3369
45.3.2.1 CH_SLEEP Mode 3369
45.3.2.2 CH_RESET Mode 3370
45.3.2.3 CH_HALT Mode 3370
45.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State) 3371
45.3.2.5 CH_OPERATION Mode (in Bus-Off State) 3372
45.3.3 Global Mode and Channel Mode Transition Interactions 3375
45.3.3.1 Transition Time for Global Mode Change 3375
45.3.3.2 Transition Time for Channel Mode Change 3376
45.4 Initialization of CANFDG Module 3377
45.4.1 Initialization of CAN Clock, Bit Timing, and Bit Rate 3377
45.4.1.1 Bit Timing Conditions 3377
45.4.1.2 Bit Timing 3378
45.4.1.3 Bit Rate 3379
45.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate 3381

45.4.1.5 Transceiver Delay Compensation 3382
45.4.2 CANFDG Module Configuration after a Reset 3384
45.5 Filtering Using Acceptance Filter List (AFL) 3386
45.5.1 Acceptance Filtering Process 3386
45.5.2 DLC Filtering Process 3386
45.5.3 Message Storage 3387
45.5.4 Payload Overflow Process 3387
45.5.5 Allocation of AFL Entries to Each CAN Channel 3388
45.5.6 AFL Entry Description 3389
45.5.7 Entering Entries in the AFL 3391
45.5.8 Loop-Back Configuration 3393
45.5.9 IDE Masking 3394
45.5.10 Updating AFL Entry during Communication 3394
45.6 FIFO Buffers and Message Buffer Configuration 3397
45.6.1 Receive Message Buffers 3398
45.6.1.1 Receive Message Buffer Configuration 3398
45.6.2 FIFO Buffers 3399
45.6.2.1 FIFO Buffers Configuration 3399
45.6.2.2 FIFO Buffers Control 3403
45.7 Reception and Transmission 3404
45.7.1 Reception 3404
45.7.1.1 Message Storage in Receive Message Buffers 3404
45.7.1.2 Message Storage in FIFO Buffers 3406
45.7.1.3 Timestamp 3408
45.7.2 Transmission 3409
45.7.2.1 Transmission Priority 3410
45.7.2.2 Message Transmission from Transmit Message Buffer 3411
45.7.2.3 Message Transmission from FIFO Buffer 3414
45.7.2.4 Transmit Queue 3419
45.7.2.5 Transmission History 3422
45.7.2.6 Transmit Data Padding 3424
45.8 ECC Check 3425
45.8.1 ECC Function Setting 3425
45.8.2 ECC Decoder Testing 3426
45.9 Test Mode 3427
45.9.1 Channel Specific Test Modes 3427
45.9.1.1 Basic Test Mode 3427
45.9.1.2 Listen-Only Mode 3427
45.9.1.3 Self Test Mode 0 (External Loop-Back Mode) 3428
45.9.1.4 Self Test Mode 1 (Internal Loop-Back Mode) 3429
45.9.1.5 Restricted Operation Mode 3429

45.9.2 Global Test Modes ... 3430
45.9.2.1 RAM Test Mode ... 3431
45.9.2.2 Internal CAN Bus Communication Test Mode ... 3433
45.10 Bus Load Measurement ... 3434
45.10.1 How to Count the CAN Bus Idle Time ... 3434
45.10.2 Measurement Procedure ... 3435
45.11 Interrupts and DTC/DMA Requests ... 3436
45.11.1 CANFDG Interrupts ... 3436
45.11.2 ECC Interrupts ... 3441
45.11.3 DTC/DMA Transfer Requests ... 3441
45.12 Usage Notes ... 3443
45.12.1 Setting the Module Stop Function ... 3443
45.12.2 Notes on the DLL Clock Settings ... 3443
45.12.3 Note on Configuration of Receive Message Buffers and FIFO Buffers ... 3443
45.12.4 Transition to Software Standby Mode ... 3443
46. CAN FD Module (CANFD) ... 3444
46.1 Overview ... 3444
46.2 Register Descriptions ... 3446
46.2.1 Nominal Bit Rate Configuration Register (NBCR) ... 3446
46.2.2 Channel Control Register (CHCR) ... 3448
46.2.3 Channel Status Register (CHSR) ... 3452
46.2.4 Channel Error Status Register (CHESR) ... 3455
46.2.5 Data Bit Rate Configuration Register (DBCR) ... 3460
46.2.6 CAN FD Configuration Register (FDCFG) ... 3462
46.2.7 CAN FD Control Register (FDCTR) ... 3465
46.2.8 CAN FD Status Register (FDSTS) ... 3466
46.2.9 CAN FD CRC Register (FDCRC) ... 3468
46.2.10 Global Configuration Register (GCFG) ... 3469
46.2.11 Global Control Register (GCR) ... 3471
46.2.12 Global Status Register (GSR) ... 3473
46.2.13 Global Error Status Register (GESR) ... 3474
46.2.14 Transmit Interrupt Status Register (TISR) ... 3476
46.2.15 Timestamp Counter Register (TSCR) ... 3478
46.2.16 Acceptance Filter List Control Register (AFCR) ... 3479
46.2.17 Acceptance Filter List Configuration Register (AFCFG) ... 3480
46.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15) ... 3481
46.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15) ... 3483
46.2.20 Acceptance Filter List n Pointer 0 Register (AFLn.PTR0) (n = 0 to 15) ... 3485
46.2.21 Acceptance Filter List n Pointer 1 Register (AFLn.PTR1) (n = 0 to 15) ... 3487
46.2.22 Receive Message Buffer Configuration Register (RMCR) ... 3489
46.2.23 Receive Message Buffer New Data Register (RMNDR) ... 3490

46.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0, 1) ... 3491
46.2.25 Receive FIFO n Status Register (RFSRn) (n = 0, 1) ... 3493
46.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0, 1) ... 3495
46.2.27 Common FIFO 0 Configuration Register (CFCR0) ... 3496
46.2.28 Common FIFO 0 Status Register (CFSR0) ... 3499
46.2.29 Common FIFO 0 Pointer Control Register (CFPCR0) ... 3502
46.2.30 FIFO Empty Status Register (FESR) ... 3503
46.2.31 FIFO Full Status Register (FFSR) ... 3504
46.2.32 FIFO Message Lost Status Register (FMLSR) ... 3505
46.2.33 Receive FIFO Interrupt Status Register (RFISR) ... 3506
46.2.34 DMA Transfer Control Register (DTCR) ... 3507
46.2.35 DMA Transfer Status Register (DTSR) ... 3508
46.2.36 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 3) ... 3509
46.2.37 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 3) ... 3511
46.2.38 Transmit Message Buffer Transmission Request Status Register 0 (TMTRSR0) ... 3512
46.2.39 Transmit Message Buffer Transmission Abort Request Status Register 0 (TMARSR0) ... 3513
46.2.40 Transmit Message Buffer Transmission Completion Status Register 0 (TMTCSR0) ... 3514
46.2.41 Transmit Message Buffer Transmission Abort Status Register 0 (TMTASR0) ... 3515
46.2.42 Transmit Message Buffer Interrupt Enable Register (TMIER0) ... 3516
46.2.43 Transmit Queue 0 Configuration Register (TQCR0) ... 3517
46.2.44 Transmit Queue 0 Status Register (TQSR0) ... 3519
46.2.45 Transmit Queue 0 Pointer Control Register (TQPCR0) ... 3521
46.2.46 Transmission History Configuration Register (THCR) ... 3522
46.2.47 Transmission History Status Register (THSR) ... 3523
46.2.48 Transmission History Access Register 0 (THACR0) ... 3525
46.2.49 Transmission History Access Register 1 (THACR1) ... 3526
46.2.50 Transmission History Pointer Control Register (THPCR) ... 3527
46.2.51 Global Reset Control Register (GRCR) ... 3528
46.2.52 Global Test Mode Configuration Register (GTMCR) ... 3529
46.2.53 Global Test Mode Enable Register (GTMER) ... 3530
46.2.54 Global CAN FD Configuration Register (GFDCFG) ... 3531
46.2.55 Global Test Mode Lock Key Register (GTMLKR) ... 3532
46.2.56 RAM Test Page Access Register k (RTPARk) (k = 0 to 63) ... 3532
46.2.57 Acceptance Filter List Ignore Entry Setting Register (AFIGSR) ... 3533
46.2.58 Acceptance Filter List Ignore Entry Enable Register (AFIGER) ... 3534
46.2.59 Receive Message Buffer Interrupt Enable Register (RMIER) ... 3535
46.2.60 Identifier Bits Alignment ... 3536
46.2.61 Message Buffer Structure ... 3537
46.2.61.1 Start Addresses ... 3537
46.2.61.2 Receive Message Buffer n (RMBn) (n = 0 to 15) ... 3538
46.2.61.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 15) ... 3539

46.2.61.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 15) 3540
46.2.61.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 15) 3541
46.2.61.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 15; p = 0 to 15) 3542
46.2.61.7 Receive Message Buffer n Data k (RMBn.DATAk) (n = 0 to 15; k = 0 to 63) 3543
46.2.61.8 Receive FIFO n (RFBn) (n = 0, 1) 3544
46.2.61.9 Receive FIFO n Header Field 0 (RFBn.HF0) (n = 0, 1) 3545
46.2.61.10 Receive FIFO n Header Field 1 (RFBn.HF1) (n = 0, 1) 3546
46.2.61.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0, 1) 3547
46.2.61.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0, 1; p = 0 to 15) 3548
46.2.61.13 Receive FIFO n Data k (RFBn.DATAk) (n = 0, 1; k = 0 to 63) 3548
46.2.61.14 Common FIFO 0 (CFB0) 3549
46.2.61.15 Common FIFO 0 Header Field 0 (CFB0.HF0) 3550
46.2.61.16 Common FIFO 0 Header Field 1 (CFB0.HF1) 3551
46.2.61.17 Common FIFO 0 Header Field 2 (CFB0.HF2) 3552
46.2.61.18 Common FIFO 0 Data Field p (CFB0.DFp) (p = 0 to 15) 3554
46.2.61.19 Common FIFO 0 Data k (CFB0.DATAk) (k = 0 to 63) 3554
46.2.61.20 Transmit Message Buffer n (TMBn) (n = 0 to 3) 3555
46.2.61.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 3) 3556
46.2.61.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 3) 3557
46.2.61.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 3) 3558
46.2.61.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 3; p = 0 to 15) 3559
46.2.61.25 Transmit Message Buffer n Data k (TMBn.DATAk) (n = 0 to 3; k = 0 to 63) 3559
46.2.62 ECC Control/Status Register (ECCSR) 3560
46.2.63 ECC Test Mode Register (ECTMR) 3564
46.2.64 ECC Decoder Test Data Register (ECTDR) 3565
46.2.65 ECC Error Address Register (ECEAR) 3565
46.3 Operating Mode 3566
46.3.1 Global Modes 3566
46.3.1.1 GL_SLEEP Mode 3567
46.3.1.2 GL_RESET Mode 3568
46.3.1.3 GL_HALT Mode 3569
46.3.1.4 GL_OPERATION Mode 3571
46.3.2 Channel Modes 3572
46.3.2.1 CH_SLEEP Mode 3572
46.3.2.2 CH_RESET Mode 3573
46.3.2.3 CH_HALT Mode 3573
46.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State) 3574
46.3.2.5 CH_OPERATION Mode (in Bus-Off State) 3575
46.3.3 Global Mode and Channel Mode Transition Interactions 3578
46.3.3.1 Transition Time for Global Mode Change 3578

46.3.3.2 Transition Time for Channel Mode Change 3579
46.4 Initialization of CANFD Module 3580
46.4.1 Initialization of CAN Clock, Bit Timing and Bit Rate 3580
46.4.1.1 Bit Timing Conditions 3580
46.4.1.2 Bit Timing 3581
46.4.1.3 Bit Rate 3582
46.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate 3584
46.4.1.5 Transceiver Delay Compensation 3584
46.4.2 CANFD Module Configuration after a Reset 3587
46.5 Filtering Using Acceptance Filter List (AFL) 3589
46.5.1 Acceptance Filtering Process 3589
46.5.2 DLC Filtering Process 3589
46.5.3 Message Storage 3590
46.5.4 Payload Overflow Process 3590
46.5.5 Allocation of AFL Entries 3590
46.5.6 AFL Entry Description 3591
46.5.7 Entering Entries in the AFL 3593
46.5.8 Loop-Back Configuration 3594
46.5.9 IDE Masking 3595
46.5.10 Updating AFL Entry during Communication 3595
46.6 FIFO Buffers and Message Buffer Configuration 3598
46.6.1 Receive Message Buffers 3599
46.6.1.1 Receive Message Buffer Configuration 3599
46.6.2 FIFO Buffers 3600
46.6.2.1 FIFO Buffers Configuration 3600
46.6.2.2 FIFO Buffers Control 3603
46.7 Reception and Transmission 3604
46.7.1 Reception 3604
46.7.1.1 Message Storage in Receive Message Buffers 3604
46.7.1.2 Message Storage in FIFO Buffers 3606
46.7.1.3 Timestamp 3608
46.7.2 Transmission 3608
46.7.2.1 Transmission Priority 3609
46.7.2.2 Message Transmission from Transmit Message Buffer 3610
46.7.2.3 Message Transmission from FIFO Buffer 3614
46.7.2.4 Transmit Queue 3617
46.7.2.5 Transmission History 3619
46.7.2.6 Transmit Data Padding 3620
46.8 ECC Check 3621
46.8.1 ECC Function Setting 3621
46.8.2 ECC Decoder Testing 3622

46.9 Test Mode ... 3623
46.9.1 Channel Specific Test Modes ... 3623
46.9.1.1 Basic Test Mode ... 3623
46.9.1.2 Listen-Only Mode ... 3623
46.9.1.3 Self Test Mode 0 (External Loop-Back Mode) ... 3624
46.9.1.4 Self Test Mode 1 (Internal Loop-Back Mode) ... 3625
46.9.1.5 Restricted Operation Mode ... 3625
46.9.2 Global Test Modes ... 3626
46.9.2.1 RAM Test Mode ... 3627
46.9.2.2 Bit Flip Test ... 3629
46.10 Interrupts and DTC/DMA Requests ... 3630
46.10.1 CANFD Interrupts ... 3630
46.10.2 ECC Interrupts ... 3634
46.10.3 DTC/DMA Transfer Requests ... 3634
46.11 Usage Notes ... 3635
46.11.1 Setting the Module Stop Function ... 3635
46.11.2 Notes on the DLL Clock Settings ... 3635
46.11.3 Note on Configuration of Receive Message Buffers and FIFO Buffers ... 3635
46.11.4 Transition to Software Standby Mode ... 3636
47. Serial Peripheral Interface (RSPIA) ... 3637
47.1 Overview ... 3637
47.2 Register Descriptions ... 3641
47.2.1 RSPI Data Register (SPDR) ... 3641
47.2.2 RSPI Clock Delay Register (SPCKD) ... 3645
47.2.3 RSPI Slave Select Negation Delay Register (SSLND) ... 3646
47.2.4 RSPI Next-Access Delay Register (SPND) ... 3647
47.2.5 RSPI Control Register (SPCR) ... 3648
47.2.6 RSPI Receive-Only Mode Control Register (SPRMCR) ... 3653
47.2.7 RSPI Receive Data Ready Detect Condition Setting Register (SPDRCSR) ... 3654
47.2.8 RSPI Pin Control Register (SPPCR) ... 3655
47.2.9 RSPI Slave Select Polarity Register (SSLP) ... 3656
47.2.10 RSPI Bit Rate Register (SPBR) ... 3657
47.2.11 RSPI Sequence Control Register (SPSCR) ... 3658
47.2.12 RSPI Command Register m (SPCMDm) (m = 0 to 7) ... 3659
47.2.13 RSPI Data Control Register (SPDCR) ... 3662
47.2.14 RSPI FIFO Control Register (SPFCR) ... 3663
47.2.15 RSPI Sequence Status Register (SPSSR) ... 3664
47.2.16 RSPI Status Register (SPSR) ... 3665
47.2.17 RSPI Transmit FIFO Status Register (SPTFSR) ... 3671
47.2.18 RSPI Receive FIFO Status Register (SPRFSR) ... 3671
47.2.19 RSPI Polling Status Register (SPPSR) ... 3672

47.2.20 RSPI Status Clear Register (SPSCLR) 3673
47.2.21 RSPI FIFO Clear Register (SPFCLR) 3674
47.3 Operation 3675
47.3.1 Overview of RSPI Operations 3675
47.3.2 Controlling RSPI Pins 3677
47.3.3 RSPI System Configuration Examples 3677
47.3.3.1 Single Master/Single Slave (with This MCU Acting as Master) 3677
47.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave) 3678
47.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master) 3679
47.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave) 3680
47.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master) 3681
47.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master) 3682
47.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave) 3682
47.3.4 Data Format 3683
47.3.4.1 Data Format of One Frame 3683
47.3.4.2 When Parity is Disabled (SPCR.SPPE = 0) 3684
47.3.4.3 When Parity is Enabled (SPCR.SPPE = 1) 3688
47.3.4.4 Byte Swap Transmission 3692
47.3.4.5 Byte Swap Reception 3694
47.3.5 Transfer Format (Frame Format) 3698
47.3.5.1 When CPHA = 0 3698
47.3.5.2 When CPHA = 1 3699
47.3.6 Communication Mode 3701
47.3.6.1 Transmit-Receive Mode (SPCR.CMMD[1:0] = 00b) 3701
47.3.6.2 Transmit-Only Mode (SPCR.CMMD[1:0] = 01b) 3702
47.3.6.3 Receive-Only Mode (SPCR.CMMD[1:0] = 10b) 3703
47.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts 3704
47.3.8 Idle Interrupts 3706
47.3.9 Communication End Interrupts 3708
47.3.9.1 Transmit-Receive Master Mode or Transmit-Only Master Mode 3708
47.3.9.2 Receive-Only Master Mode 3710
47.3.9.3 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on SPI Operation 3712
47.3.9.4 Receive-Only Slave Mode on SPI Operation 3714
47.3.9.5 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on Clock Synchronous Operation 3716
47.3.9.6 Receive-Only Slave Mode on Clock Synchronous Operation 3717
47.3.10 Error Detection 3718
47.3.10.1 Overrun Error 3719
47.3.10.2 Parity Error 3724

47.3.10.3 Mode Fault Error ... 3725
47.3.10.4 Underrun Error ... 3725
47.3.11 Received Data Ready Detection ... 3726
47.3.12 Initializing RSPIA ... 3727
47.3.12.1 Initialization by Clearing the SPE Bit ... 3727
47.3.12.2 System Reset ... 3727
47.3.13 SPI Operation ... 3728
47.3.13.1 Master Mode Operation ... 3728
47.3.13.2 Slave Mode Operation ... 3741
47.3.14 Clock Synchronous Operation ... 3747
47.3.14.1 Master Mode Operation ... 3747
47.3.14.2 Slave Mode Operation ... 3751
47.3.15 Loopback Mode ... 3753
47.3.16 Self-Diagnosis of Parity Bit Function ... 3754
47.3.17 Interrupt Sources ... 3755
47.3.18 Link Operation by Event Linking ... 3756
47.3.18.1 Receive Buffer Full Event Output ... 3756
47.3.18.2 Transmit Buffer Empty Event Output ... 3756
47.3.18.3 Error Event Output ... 3757
47.3.18.4 Idle Event Output ... 3758
47.3.18.5 Communication End Event Output ... 3759
47.3.19 Synchronizer Bypass Function ... 3761
47.4 Usage Notes ... 3762
47.4.1 Settings of the Module Stop Function ... 3762
47.4.2 Notes on Starting Transfer ... 3762
47.4.3 Note on Error Events Output ... 3762
47.4.4 Note on Low Power Consumption Functions ... 3762
47.4.5 Note on the SPRF and SPTEF flags ... 3762
47.4.6 Notes on Burst Transfer in Master Mode ... 3762
47.4.7 Notes on Operating in TI SSP Slave Mode ... 3762
47.4.8 Notes on Data Length and Frequency Ratio ... 3763
47.4.9 Notes on Rewriting Registers ... 3763
48. Expanded Serial Peripheral Interface (XSPI) ... 3765
48.1 Overview ... 3765
48.2 Register Descriptions ... 3769
48.2.1 Operating Mode Setting Register (DIPCFG) ... 3769
48.2.2 Controller Mode Registers ... 3770
48.2.2.1 Timing Configuration Register (WRAPCFG) ... 3770
48.2.2.2 Common Configuration Register (COMCFG) ... 3771
48.2.2.3 Memory-Mapped Mode Configuration Register (BMCFGCH0) ... 3772

48.2.2.4 CSi Area Memory-Mapped Mode Configuration Register 0 (CMCFG0CSi) (i = 0, 1) ... 3774
48.2.2.5 CSi Area Memory-Mapped Mode Configuration Register 1 (CMCFG1CSi) (i = 0, 1) ... 3776
48.2.2.6 CSi Area Memory-Mapped Mode Configuration Register 2 (CMCFG2CSi) (i = 0, 1) ... 3777
48.2.2.7 CSi Area I/O Signal Configuration Register (LIOCFGCSi) (i = 0, 1) ... 3778
48.2.2.8 Memory Access Extension Configuration Register (MEMEXTCFG) ... 3781
48.2.2.9 Memory-Mapped Mode Control Register 0 (BMCTL0) ... 3782
48.2.2.10 Memory-Mapped Mode Control Register 1 (BMCTL1) ... 3783
48.2.2.11 Memory-Mapped Mode Command Control Register (CMCTLCH0) ... 3784
48.2.2.12 Indirect Access Mode Control Register 0 (CDCTL0) ... 3785
48.2.2.13 Indirect Access Mode Control Register 1 (CDCTL1) ... 3786
48.2.2.14 Indirect Access Mode Control register 2 (CDCTL2) ... 3787
48.2.2.15 Command Buffer m Transaction Type Register (CDTBUFm) (m = 0 to 3) ... 3788
48.2.2.16 Command Buffer m Address Register (CDABUFm) (m = 0 to 3) ... 3790
48.2.2.17 Command Buffer m Data Register 0 (CDD0BUFm) (m = 0 to 3) ... 3791
48.2.2.18 Command Buffer m Data Register 1 (CDD1BUFm) (m = 0 to 3) ... 3791
48.2.2.19 Signal Pattern Control Register 0 (LPCTL0) ... 3792
48.2.2.20 Signal Pattern Control Register 1 (LPCTL1) ... 3794
48.2.2.21 I/O Signal Control Register (LIOCTL) ... 3796
48.2.2.22 CSi Area Calibration Control Register 0 (CCCTL0CSi) (i = 0, 1) ... 3797
48.2.2.23 CSi Area Calibration Control Register 1 (CCCTL1CSi) (i = 0, 1) ... 3799
48.2.2.24 CSi Area Calibration Control Register 2 (CCCTL2CSi) (i = 0, 1) ... 3800
48.2.2.25 CSi Area Calibration Control Register 3 (CCCTL3CSi) (i = 0, 1) ... 3801
48.2.2.26 CSi Area Calibration Control Register 4 (CCCTL4CSi) (i = 0, 1) ... 3801
48.2.2.27 CSi Area Calibration Control Register 5 (CCCTL5CSi) (i = 0, 1) ... 3802
48.2.2.28 CSi Area Calibration Control Register 6 (CCCTL6CSi) (i = 0, 1) ... 3802
48.2.2.29 CSi Area Calibration Control Register 7 (CCCTL7CSi) (i = 0, 1) ... 3803
48.2.2.30 Common Status Register (COMSTT) ... 3804
48.2.2.31 CSi Area Calibration Status Register (CASTTCSi) (i = 0, 1) ... 3806
48.2.2.32 Interrupt Status Register (INTS) ... 3808
48.2.2.33 Interrupt Clear Register (INTC) ... 3810
48.2.2.34 Interrupt Enable Register (INTE) ... 3811
48.2.3 Target Mode Registers ... 3812
48.2.3.1 L1.5 Configuration Register 0 (L15CFG0) ... 3812
48.2.3.2 L1.5 Configuration Register 1 (L15CFG1) ... 3813
48.2.3.3 L1.5 Configuration Register 2 (L15CFG2) ... 3814
48.2.3.4 L2 Configuration Register 0 (L2CFG0) ... 3815
48.2.3.5 L2 Configuration Register 1 (L2CFG1) ... 3816
48.2.3.6 Target Write Address Register (TWADR) ... 3817
48.2.3.7 Target Write Length Register (TWLEN) ... 3817

48.2.3.8 Target Write Control Register (TWCNT) ... 3818
48.2.3.9 Target Read Address Register (TRADR) ... 3818
48.2.3.10 Target Read Length Register (TRLEN) ... 3819
48.2.3.11 Target Read Control Register (TRCNT) ... 3819
48.2.3.12 Transfer Control Register (TRNSC) ... 3820
48.2.3.13 Target Write Buffer Status Register (TWBS) ... 3821
48.2.3.14 Target Write Transfer Status Register (TWTS) ... 3821
48.2.3.15 Target Read Buffer Status Register (TRBS) ... 3822
48.2.3.16 Target Read Transfer Status Register (TRTS) ... 3822
48.2.3.17 Target Status Register (TS) ... 3823
48.2.3.18 Target Status Clear Register (TSC) ... 3825
48.2.3.19 Target Status Set Register (TSS) ... 3826
48.2.3.20 Target Status xSPI Interrupt Enable Register (TSXIE) ... 3827
48.2.3.21 Target Status System Interrupt Enable Register (TSSIE) ... 3828
48.2.3.22 Target Basic Information Register (TBI) ... 3829
48.3 Operation ... 3830
48.3.1 XSPI Controller Module ... 3830
48.3.1.1 xSPI Bus ... 3831
48.3.1.2 Indirect Access Mode ... 3839
48.3.1.3 Memory-Mapped Mode ... 3840
48.3.1.4 Signal Pattern Control ... 3848
48.3.1.5 Error Detection ... 3850
48.3.1.6 Interrupt Sources ... 3851
48.3.1.7 Operation Flows ... 3852
48.3.2 XSPI Target Module ... 3860
48.3.2.1 XSPI Target Configuration ... 3860
48.3.2.2 Register Access ... 3860
48.3.2.3 Memory Access ... 3861
48.3.2.4 Changing IO Protocol Mode ... 3866
48.3.2.5 xSPI Reset ... 3867
48.3.2.6 Error Detection ... 3868
48.3.2.7 Interrupt Sources ... 3869
48.4 Usage Notes ... 3870
48.4.1 Settings of the Module Stop Function ... 3870
48.4.2 Notes on the XSPI Controller ... 3870
48.4.2.1 Software Standby Mode ... 3870
48.4.2.2 Restrictions Regarding the Prefetch Function ... 3870
48.4.2.3 Note on the Prefetch Function ... 3870
48.4.2.4 Note on Using Special Memory Devices ... 3870
48.4.3 Notes on the XSPI Target ... 3870
48.4.3.1 Software Standby Mode ... 3870

48.4.3.2 Endian Setting 3870
49. Decryption On The Fly (DOTF) 3871
49.1 Overview 3871
49.2 Register Descriptions 3872
49.2.1 Decryption Area Start Address Register (CONVAREAST) 3872
49.2.2 Decryption Area End Address Register (CONVAREAED) 3872
49.3 Operation 3873
49.3.1 Initialization Flow 3873
49.4 Usage Notes 3874
49.4.1 Module Stop Function Setting 3874
50. CRC Calculator (CRCA) 3875
50.1 Overview 3875
50.2 Register Descriptions 3876
50.2.1 CRC Control Register (CRCCR) 3876
50.2.2 CRC Data Input Register (CRCDIR) 3877
50.2.3 CRC Data Output Register (CRCDOR) 3878
50.3 Operation 3879
50.4 Usage Notes 3882
50.4.1 Module Stop Function Setting 3882
50.4.2 Note on Transmission 3882
51. Serial Sound Interface (SSIE) 3883
51.1 Overview 3883
51.2 Register Descriptions 3886
51.2.1 Control Register (SSICR) 3886
51.2.2 Status Register (SSISR) 3891
51.2.3 FIFO Control Register (SSIFCR) 3893
51.2.4 FIFO Status Register (SSIFSR) 3896
51.2.5 Transmit FIFO Data Register (SSIFTDR) 3898
51.2.6 Receive FIFO Data Register (SSIFRDR) 3899
51.2.7 Audio Format Register (SSIOFR) 3900
51.2.8 FIFO Status Control Register (SSISCR) 3901
51.3 Operation 3902
51.3.1 Data Formats 3902
51.3.1.1 I^2S Format 3903
51.3.1.2 Left-Justified Format 3904
51.3.1.3 Right-Justified Format 3904
51.3.1.4 Monaural Format 3905
51.3.1.5 TDM Format 3906
51.3.2 Output Control 3908
51.3.2.1 Mute Function 3908
51.3.2.2 LRCK Continuous Output Function 3909

51.3.2.3 BCK Continuous Output Disable Function ... 3909
51.3.3 Transfer Modes ... 3910
51.3.3.1 Slave Mode ... 3910
51.3.3.2 Master Mode ... 3910
51.3.3.3 Transmission ... 3910
51.3.3.4 Reception ... 3911
51.3.3.5 Transception ... 3911
51.3.4 State Transition ... 3912
51.3.4.1 Idle State ... 3912
51.3.4.2 Transfer States ... 3913
51.3.5 SSIE Initialization ... 3914
51.3.6 Transmission ... 3915
51.3.7 Reception ... 3916
51.3.8 Stop Data Transfer ... 3917
51.3.9 Error Processing ... 3918
51.4 Interrupts ... 3919
51.4.1 SSIFn Interrupt ... 3919
51.4.2 SSITXIn Interrupt ... 3920
51.4.3 SSIRXIn Interrupt ... 3921
51.5 Notes ... 3922
51.5.1 Setting the Module Stop Function ... 3922
51.5.2 Notes on AUDIO_CLK ... 3922
51.5.3 Note on LRCK in Slave Mode ... 3922
51.5.4 Transition to Software Standby Mode ... 3922
52. SD/MMC Host Interface (SDHIc) ... 3923
52.1 Overview ... 3923
52.2 Register Descriptions ... 3925
52.2.1 Command Register (SDCMD) ... 3925
52.2.2 Argument Register (SDARG) ... 3928
52.2.3 Data Stop Register (SDSTOP) ... 3929
52.2.4 Block Count Register (SDBLKCNT) ... 3930
52.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76) ... 3931
52.2.6 SD Status Register 1 (SDSTS1) ... 3932
52.2.7 SD Status Register 2 (SDSTS2) ... 3935
52.2.8 SD Interrupt Mask Register 1 (SDIMSK1) ... 3939
52.2.9 SD Interrupt Mask Register 2 (SDIMSK2) ... 3940
52.2.10 SDHI Clock Control Register (SDCLKCR) ... 3941
52.2.11 Transfer Data Size Register (SDSIZE) ... 3942
52.2.12 Card Access Option Register (SDOPT) ... 3943
52.2.13 SD Error Status Register 1 (SDERSTS1) ... 3945

52.2.14 SD Error Status Register 2 (SDERSTS2) 3946
52.2.15 SD Buffer Register (SDBUFR) 3947
52.2.16 SDIO Mode Control Register (SDIOMD) 3948
52.2.17 SDIO Status Register (SDIOSTS) 3950
52.2.18 SDIO Interrupt Mask Register (SDIOIMSK) 3951
52.2.19 DMA Transfer Enable Register (SDDMAEN) 3952
52.2.20 SDHI Software Reset Register (SDRST) 3953
52.2.21 SD/MMC Interface Mode Register (SDIFMD) 3954
52.2.22 Swap Control Register (SDSWAP) 3955
52.3 SDHI Operation 3956
52.3.1 Data Block Format of the SD Card/MMC 3956
52.3.2 SD Buffer and the SDBUFR Register 3957
52.3.3 SD Card/MMC Detection 3958
52.3.3.1 Using the SD0_CD Pin to Detect an SD Card/MMC 3958
52.3.3.2 Using the SD0_DAT3 Pin to Detect an SD Card 3959
52.3.4 SD Card/MMC Write Protection 3959
52.3.4.1 Using the SD0_WP Pin to Enable Write Protection 3959
52.3.4.2 Using a Command to Enable Write Protection 3959
52.3.5 Communication Errors and Timeouts 3960
52.3.6 Examples of Issuing a Command 3961
52.3.6.1 Command Absent of Response Reception and Data Transfer 3961
52.3.6.2 Command Absent of Data Transfer 3962
52.3.6.3 Single Block Read Command (CMD17) 3963
52.3.6.4 Single Block Write Command (CMD24) 3965
52.3.6.5 Multi-Block Read Command (CMD18) 3967
52.3.6.6 Multi-Block Write Command (CMD25) 3969
52.3.6.7 Multi-Block Write Command (CMD54) 3971
52.3.6.8 IO_RW_DIRECT Command (CMD52) 3973
52.3.6.9 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read)) 3974
52.3.6.10 IO_RW_EXTENDED (CMD53 Multi-Block Write) 3976
52.3.6.11 DMA Transfer 3978
52.4 Interrupts 3980
52.4.1 DMA Transfer Triggered by Interrupt Requests 3981
52.5 Notes on Using the SDHI 3982
52.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It 3982
52.5.2 SDBUFR Register Illegal Write Error 3982
52.5.3 Automatic Control of the SDHI Clock Output 3983
52.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence 3983
52.5.5 Note on Setting the SDCLKCR Register 3983
52.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence 3983
52.5.7 Controlling Module Operation 3983

53. Parallel Data Capture Unit (PDC) 3984
53.1 Overview 3984
53.2 Register Descriptions 3986
53.2.1 PDC Control Register 0 (PCCR0) 3986
53.2.2 PDC Control Register 1 (PCCR1) 3988
53.2.3 PDC Status Register (PCSR) 3989
53.2.4 PDC Pin Monitor Register (PCMONR) 3991
53.2.5 PDC Receive Data Register (PCDR) 3991
53.2.6 Vertical Capture Register (VCR) 3993
53.2.7 Horizontal Capture Register (HCR) 3994
53.3 Operation 3995
53.3.1 Transfer Formats 3995
53.3.2 Transfer Timing 3996
53.3.3 Settings of the VCR and HCR Registers and the Captured Range 3997
53.3.4 Operations for Reception 3999
53.3.5 Operation during the Horizontal Blanking Period 4000
53.3.6 Operations for Continued Reception at the Time of Frame End 4000
53.3.7 Error Detection 4001
53.3.8 Initial Settings 4004
53.3.9 Flows of Operations 4005
53.3.10 Interrupt Source 4007
53.3.11 Reset State 4008
53.4 Usage Notes 4009
53.4.1 Setting of the Module Stop Function 4009
53.4.2 Notes on the Power Consumption Reduction Function 4009
53.4.3 Notes on Error Interrupts 4009
53.4.4 Notes on Use of the DTC 4009
53.4.5 Notes on Use of the DMAC 4009
53.4.6 Notes on Start of Transfer 4010
54. Boundary Scan 4011
54.1 Overview 4011
54.2 Register Descriptions 4012
54.2.1 Instruction Register (JTIR) 4013
54.2.2 ID Code Register (JTIDR) 4013
54.2.3 Bypass Register (JTBPR) 4014
54.2.4 Boundary Scan Register (JTBSR) 4014
54.3 Operations 4043
54.3.1 TAP Controller 4043
54.3.2 List of Commands 4044
54.4 Usage Notes 4045

55. Arithmetic Unit for Trigonometric Functions (TFUv2) 4046
55.1 Overview 4046
55.2 Register Descriptions 4048
55.2.1 Fixed-Point Sincos Input and Output Setting Register (FXSCIOC) 4048
55.2.2 Fixed-Point Atanhypot_k Input and Output Setting Register (FXATIOC) 4049
55.2.3 Trigonometric Status Register (TRGSTS) 4050
55.2.4 Floating-Point Sincos Data Register 0 (FPSCDT0) 4051
55.2.5 Floating-Point Sincos Data Register 1 (FPSCDT1) 4051
55.2.6 Floating-Point Atanhypot_k Data Register 0 (FPATDT0) 4052
55.2.7 Floating-Point Atanhypot_k Data Register 1 (FPATDT1) 4052
55.2.8 Fixed-Point Sincos Data Register 0 (FXSCDT0) 4053
55.2.9 Fixed-Point Sincos Data Register 1 (FXSCDT1) 4053
55.2.10 Fixed-Point Atanhypot_k Data Register 0 (FXATDT0) 4055
55.2.11 Fixed-Point Atanhypot_k Data Register 1 (FXATDT1) 4055
55.2.12 Data Saving and Restoring Register 0 (DTSR0) 4057
55.2.13 Data Saving and Restoring Register 1 (DTSR1) 4057
55.2.14 TFU Security Attribution Register (TFUSAR) 4058
55.3 Operation 4059
55.3.1 Arithmetic Processing 4059
55.3.2 Input and Output Value Formats 4059
55.3.2.1 Single-Precision Floating-Point Numbers 4059
55.3.2.2 Fixed-Point Numbers 4060
55.3.3 Units of Angles 4061
55.3.4 Relationship Between Input and Output Values for Sincos Operation 4061
55.3.5 Relationship Between Input and Output Values for Atan Operation 4062
55.3.6 Relationship Between Input and Output Values for hypot_k Operation 4062
55.4 Usage Procedures 4063
55.4.1 Procedures for Using the TFU 4063
55.4.2 How to Use Interrupts 4065
56. Security Features 4066
56.1 Features 4066
56.2 Detection of Tampering 4067
56.3 Protect Zone Feature 4068
56.3.1 Protect Zone Technology 4068
56.3.2 Flat-Project and Non-Flat-Project 4068
56.3.3 Relationship Between Interrupts and Worlds 4069
56.3.3.1 When Set for Operation for Non-Flat-Project 4070
56.3.3.2 When Set for Operation for Flat-Project 4070
56.3.4 Protect Zone Initial State 4070
56.4 Bus Access and Address Space 4071
56.5 Security Attribution Unit (SAU) 4074

56.5.1 Types of CPU Access 4074
56.5.2 Address-Space Security Attribute 4074
56.5.3 Transaction Types Issued 4074
56.5.4 Access Across Different Security Attribute Spaces 4074
56.5.5 Functions 4075
56.5.5.1 Secure Memory Protection Function 4075
56.5.5.2 Protection of Security Attribution Management Registers 4075
56.5.5.3 Access Determination Flow for the Secure Memory Protection Function 4075
56.5.6 Instructions for Use 4077
56.5.6.1 Handling Secure Exceptions 4077
56.6 Memory Security Attributes 4077
56.7 Security Attribution of Peripherals 4080
56.8 Protect Zone Access Error 4080
56.9 Register Descriptions 4081
56.9.1 Secure Address-Space Attribute Setting Register (AAS) 4081
56.9.2 Secure Exception Status Register (SESTS) 4082
56.9.3 Secure Exception Status Clear Register (SECLR) 4083
56.9.4 Secure Exception Address Register (SEA) 4084
56.9.5 Peripheral Security Attribution Register B (PSARB) 4085
56.9.6 Peripheral Security Attribution Register C (PSARC) 4087
56.9.7 Peripheral Security Attribution Register D (PSARD) 4089
56.9.8 Peripheral Security Attribution Register E (PSARE) 4090
56.9.9 Peripheral Security Attribution Register F (PSARF) 4091
56.9.10 Module Stop Security Attribution Register (MSSAR) 4092
56.9.11 Code MRAM Security Attribution Monitor Register (CMSAMON) 4093
56.9.12 Data MRAM Security Attribution Monitor Register (DMSAMON) 4094
56.9.13 Device Lifecycle Management State Monitor Register (DLMMON) 4095
56.9.14 Master Security Attribution Operation-After-Detection Register (MSAOAD) 4096
56.9.15 Master Security Attribution Protect Register (MSAPT) 4097
56.10 Secure Key Injection 4098
56.11 Secure Factory Programming 4100
56.12 Secure Boot 4102
56.13 Field Updating in Dual Mode 4109
56.14 Device Lifecycle Management 4110
56.14.1 Changing Lifecycle States 4111
56.14.2 Protection and Authentication Levels 4111
56.14.3 Serial Programming 4113
56.14.4 Device Lifecycle States and PL Change Examples 4113
56.14.5 Failure Analysis 4113
56.15 Usage Notes 4114
56.15.1 Notes on Settings of the Security Attribution Bits 4114

56.15.2 Security Attribution Bit Write Timing 4114
56.15.3 Executing the WAIT Instruction in Non-Flat-Project Operation 4114
57. Renesas Secure IP (RSIP-E50D) 4115
57.1 Overview 4115
57.2 Operation 4117
57.2.1 Encryption Engine 4117
57.2.2 Encryption and Decryption 4118
57.3 Usage Notes 4118
57.3.1 Standby Mode 4118
57.3.2 Setting the Module Stop Function 4118
57.3.3 Protection against SPA/DPA 4118
57.3.4 RSIP Library 4118
58. Pulse-Density Modulation Interface (PDM-IF) 4119
58.1 Overview 4119
58.2 Register Descriptions 4121
58.2.1 Register Map 4121
58.2.2 Channel Software Start Trigger Register (PDCSTRTR) 4125
58.2.3 Channel Software Stop Trigger Register (PDCSTPTR) 4125
58.2.4 Channel Software Change Trigger Register (PDCCHGTR) 4126
58.2.5 Channel Interrupt Control Register (PDCICR) 4126
58.2.6 Channel Status Register (PDCSR) 4127
58.2.7 Channel Status Clear Register (PDCSCR) 4129
58.2.8 Channel Audio Detection Control Register (PDCSDCR) 4129
58.2.9 Channel Data Read Control Register (PDCDRCR) 4130
58.2.10 Channel Data Clear Register (PDCDCR) 4130
58.2.11 Version Register (PDVR) 4131
58.2.12 Software Start Trigger Register Channel 0 (PDSTRTRCH0) 4131
58.2.13 Software Stop Trigger Register Channel 0 (PDSTPTRCH0) 4132
58.2.14 Software Change Trigger Register Channel 0 (PDCHGTRCH0) 4132
58.2.15 Interrupt Control Register Channel 0 (PDICRCH0) 4133
58.2.16 Status Detection Control Register Channel 0 (PDSDCRCH0) 4134
58.2.17 Status Register Channel 0 (PDSRCH0) 4135
58.2.18 Status Clear Register Channel 0 (PDSCRCH0) 4137
58.2.19 Mode Setting Register Channel 0 (PDMDSRCH0) 4138
58.2.20 Sinc Filter Control Register Channel 0 (PDSFCRCH0) 4139
58.2.21 High-Pass Filter Coefficient s (0) Register Channel 0 (PDHFCS0RCH0) 4141
58.2.22 High-Pass Filter Coefficient k (1) Register Channel 0 (PDHFCK1RCH0) 4141
58.2.23 High-Pass Filter Coefficient h (0) Register Channel 0 (PDHFCH0RCH0) 4142
58.2.24 High-Pass Filter Coefficient h (1) Register Channel 0 (PDHFCH1RCH0) 4142
58.2.25 Compensation Filter Coefficient h (0) Register Channel 0 (PDCFCH00RCH0) 4143
58.2.26 Compensation Filter Coefficient h (1) Register Channel 0 (PDCFCH01RCH0) 4143

58.2.27 Compensation Filter Coefficient h (2) Register Channel 0 (PDCFCH02RCH0) 4144
58.2.28 Compensation Filter Coefficient h (3) Register Channel 0 (PDCFCH03RCH0) 4144
58.2.29 Compensation Filter Coefficient h (4) Register Channel 0 (PDCFCH04RCH0) 4145
58.2.30 Compensation Filter Coefficient h (5) Register Channel 0 (PDCFCH05RCH0) 4145
58.2.31 Compensation Filter Coefficient h (6) Register Channel 0 (PDCFCH06RCH0) 4146
58.2.32 Compensation Filter Coefficient h (7) Register Channel 0 (PDCFCH07RCH0) 4146
58.2.33 Compensation Filter Coefficient h (8) Register Channel 0 (PDCFCH08RCH0) 4147
58.2.34 Compensation Filter Coefficient h (9) Register Channel 0 (PDCFCH09RCH0) 4147
58.2.35 Compensation Filter Coefficient h (10) Register Channel 0 (PDCFCH10RCH0) 4148
58.2.36 Low-Pass Filter Coefficient h0 (10) Register Channel 0 (PDLFCH010RCH0) 4148
58.2.37 Low-Pass Filter Coefficient h1 (0) Register Channel 0 (PDLFCH100RCH0) 4149
58.2.38 Low-Pass Filter Coefficient h1 (1) Register Channel 0 (PDLFCH101RCH0) 4149
58.2.39 Low-Pass Filter Coefficient h1 (2) Register Channel 0 (PDLFCH102RCH0) 4150
58.2.40 Low-Pass Filter Coefficient h1 (3) Register Channel 0 (PDLFCH103RCH0) 4150
58.2.41 Low-Pass Filter Coefficient h1 (4) Register Channel 0 (PDLFCH104RCH0) 4151
58.2.42 Low-Pass Filter Coefficient h1 (5) Register Channel 0 (PDLFCH105RCH0) 4151
58.2.43 Low-Pass Filter Coefficient h1 (6) Register Channel 0 (PDLFCH106RCH0) 4152
58.2.44 Low-Pass Filter Coefficient h1 (7) Register Channel 0 (PDLFCH107RCH0) 4152
58.2.45 Low-Pass Filter Coefficient h1 (8) Register Channel 0 (PDLFCH108RCH0) 4153
58.2.46 Low-Pass Filter Coefficient h1 (9) Register Channel 0 (PDLFCH109RCH0) 4153
58.2.47 Low-Pass Filter Coefficient h1 (10) Register Channel 0 (PDLFCH110RCH0) 4154
58.2.48 Low-Pass Filter Coefficient h1 (11) Register Channel 0 (PDLFCH111RCH0) 4154
58.2.49 Low-Pass Filter Coefficient h1 (12) Register Channel 0 (PDLFCH112RCH0) 4155
58.2.50 Low-Pass Filter Coefficient h1 (13) Register Channel 0 (PDLFCH113RCH0) 4155
58.2.51 Low-Pass Filter Coefficient h1 (14) Register Channel 0 (PDLFCH114RCH0) 4156
58.2.52 Low-Pass Filter Coefficient h1 (15) Register Channel 0 (PDLFCH115RCH0) 4156
58.2.53 Low-Pass Filter Coefficient h1 (16) Register Channel 0 (PDLFCH116RCH0) 4157
58.2.54 Low-Pass Filter Coefficient h1 (17) Register Channel 0 (PDLFCH117RCH0) 4157
58.2.55 Low-Pass Filter Coefficient h1 (18) Register Channel 0 (PDLFCH118RCH0) 4158
58.2.56 Low-Pass Filter Coefficient h1 (19) Register Channel 0 (PDLFCH119RCH0) 4158
58.2.57 Audio Detection Lower Threshold Register Channel 0 (PDSDLTRCH0) 4159
58.2.58 Audio Detection Upper Threshold Register Channel 0 (PDSDUTRCH0) 4159
58.2.59 Data Buffer Control Register Channel 0 (PDDBCRCH0) 4160
58.2.60 Short-Circuit Threshold Setting Register Channel 0 (PDSCTSRCH0) 4160
58.2.61 Overvoltage Lower Threshold Register Channel 0 (PDOVLTRCH0) 4161
58.2.62 Overvoltage Upper Threshold Register Channel 0 (PDOVUTRCH0) 4161
58.2.63 Data Read Control Register Channel 0 (PDDRCRCH0) 4162
58.2.64 Data Clear Register Channel 0 (PDDCRCH0) 4162
58.2.65 Data Read Register Channel 0 (PDDRRCH0) 4163
58.2.66 Data Status Register Channel 0 (PDDSRCH0) 4163
58.2.67 Mirroring Registers 4164

58.3 Operation 4165
58.3.1 Clock Specifications 4165
58.3.1.1 Output Sampling Frequency 4165
58.3.2 Channel Start and Channel Stop 4166
58.3.3 Changing the PDMCLK0 Frequency 4166
58.3.4 Filters 4167
58.3.4.1 Data Clipping and Shifting Among Filters 4167
58.3.4.2 Sinc Filter 4172
58.3.4.3 High-Pass Filter 4174
58.3.4.4 Compensation Filter 4175
58.3.4.5 Low-Pass (Half-Band Decimation) Filter 4176
58.3.4.6 Moving Average Filter 4177
58.3.4.7 Settling Times of Channel Start and Setting Change 4178
58.3.5 Audio Detection 4179
58.3.6 Data Buffer 4180
58.3.7 Short-Circuit Detection 4181
58.3.8 Overvoltage Detection 4182
58.3.9 Errors 4183
58.3.10 Interrupt Functions 4183
58.4 Setting Flows 4184
58.4.1 Start Flow 4187
58.4.2 Stop Flow 4188
58.4.3 Normal Processing Flow 4189
58.4.4 Low Power Consumption Mode Transition Flow 4190
58.4.5 Normal Mode Transition Flow 4191
58.5 Usage Notes 4192
58.5.1 Setting the Data Receive Interrupt Threshold 4192
58.5.2 Default Filter Frequency Characteristics 4192
59. Phase-Counting Interface (PCIF) 4194
59.1 Overview 4194
59.2 Register Descriptions 4198
59.2.1 PCIF Clock Enable Register (CLKER) 4198
59.2.2 PCIF Clock Status Register (CLKSR) 4199
59.2.3 PCIF Clock Control Register (CLKCR) 4200
59.2.4 PCIF Control Register (CTLR) 4201
59.2.5 PCIF Maximum Position Setting Register (POSMAXR) 4202
59.2.6 PCIF Start/Stop Trigger Register (TRGR) 4203
59.2.7 PCIF Status Register (SR) 4204
59.2.8 PCIF Phase Counting Mode Control Register (PHCCTLR) 4205
59.2.9 Carrier Period Event Difference Threshold Select Register for PCIF Phase Counting Mode (PHCCEDTHR) 4207

59.2.10 PCIF Phase Counting Mode Carrier Period Setting Register (PHCPERIODR) 4208
59.2.11 PCIF Phase Counting Mode Edge Count Setting Register (PHCECR) 4209
59.2.12 PCIF Phase Counting Mode Position Count Monitor Register (PHCPCMR) 4210
59.2.13 PCIF Phase Counting Mode Position Counter Write Register (PHCPCWDR) 4211
59.2.14 PCIF Phase Counting Mode Status Register (PHCSR) 4212
59.2.15 PCIF Phase Counting Mode Status Clear Register (PHCSCR) 4213
59.2.16 PCIF Phase Counting Mode Status Enable Register (PHCSER) 4214
59.2.17 PCIF Incremental Mode Control Register (INCCTLR) 4215
59.2.18 PCIF Incremental Mode Z-Phase Generation Control Register (INCZGCTLR) 4217
59.2.19 PCIF Incremental Mode Position Counter Clear Enable Register (INCPCCER) 4219
59.2.20 PCIF Incremental Mode Software Trigger Register (INCSWTRGR) 4220
59.2.21 PCIF Incremental Mode Position Counter Monitor Register (INCPCMR) 4221
59.2.22 PCIF Software Reset Register (SWRSTR) 4222
59.3 Operation 4223
59.3.1 Phase Counting Mode 4223
59.3.1.1 Startup Control 4224
59.3.1.2 Stop Control 4225
59.3.1.3 Carrier Period Control 4227
59.3.1.4 Position Counter Control 4230
59.3.1.5 Generating A-, B-, and Z-Phase Signals 4232
59.3.1.6 States 4233
59.3.2 Incremental Mode 4238
59.3.2.1 Startup/Stop Control 4239
59.3.2.2 Generating A- and B-Phase Signals 4241
59.3.2.3 Generating Z-Phase Signals 4250
59.3.3 Output Controller 4257
59.3.3.1 ABZ-Phase Output Inversion Function 4257
59.3.4 Operating Flows 4258
59.3.4.1 Overall Operation 4258
59.3.4.2 PCIF Startup Processing 4259
59.3.4.3 Flows of Phase Counting Mode 4260
59.3.4.4 Flows of Incremental Mode 4264
59.3.4.5 PCIF Stop Processing 4267
59.4 Event Link Function 4268
59.4.1 Carrier Period Event Input (ELC_IN) 4268
59.4.2 Clearing the Internal Counters by Crest/Trough Event Trigger Input (PEK_IN, VLY_IN) 4268
59.4.3 Generating Z-Phase Output by Crest/Trough Event Trigger Input (PEK_IN, VLY_IN) 4268
59.5 Usage Notes 4269
59.5.1 Restrictions on Input Pins 4269
59.5.2 Restriction on Register Settings 4269

59.5.3 Restrictions on Operation 4270
59.5.4 Transition to Software Standby Mode 4270
59.5.5 Return from Software Standby Mode 4270
60. 16-Bit A/D Converter (ADC16H) 4271
60.1 Overview 4271
60.2 Register Descriptions 4282
60.2.1 A/D Conversion Clock Enable Register (ADCLKENR) 4282
60.2.2 A/D Conversion Clock Status Register (ADCLKSR) 4283
60.2.3 A/D Conversion Clock Control Register (ADCLKCR) 4284
60.2.4 A/D Converter Synchronous Operation Control Register (ADSYCR) 4285
60.2.5 A/D Converter Unit Sleep Control Register 0 (ADUSLPCR0) 4287
60.2.6 A/D Converter Unit Sleep Control Register 1 (ADUSLPCR1) 4287
60.2.7 A/D Converter Unit Sleep Control Register 2 (ADUSLPCR2) 4288
60.2.8 A/D Conversion Error Interrupt Enable Register (ADERINTCR) 4288
60.2.9 A/D Conversion Overflow Interrupt Enable Register (ADOVFINTCR) 4289
60.2.10 Calibration interrupt Enable Register (ADCALINTCR) 4289
60.2.11 A/D Converter Mode Selection Register (ADMDR) 4290
60.2.12 A/D Group scan Priority Control Register (ADGSPCR) 4291
60.2.13 Scan Group Enable Register (ADSGER) 4292
60.2.14 Scan Group Control Register 0 (ADSGCR0) 4293
60.2.15 Scan Group Control Register 1 (ADSGCR1) 4294
60.2.16 Scan Group Control Register 2 (ADSGCR2) 4295
60.2.17 Scan End Interrupt Enable Register (ADINTCR) 4296
60.2.18 External Trigger Enable Register n (ADTRGEXTn) (n = 0 to 11) 4296
60.2.19 ELC Trigger Enable Register n (ADTRGELCn) (n = 0 to 11) 4297
60.2.20 GPT Trigger Enable Register n (ADTRGGPTn) (n = 0 to 11) 4298
60.2.21 A/D Conversion Start Trigger Enable Register (ADTRGENR) 4299
60.2.22 A/D Conversion Start Trigger Delay Register 0 (ADTRGDLR0) 4300
60.2.23 A/D Conversion Start Trigger Delay Register 1 (ADTRGDLR1) 4301
60.2.24 A/D Conversion Start Trigger Delay Register 2 (ADTRGDLR2) 4302
60.2.25 A/D Conversion Start Trigger Delay Register 3 (ADTRGDLR3) 4303
60.2.26 A/D Conversion Start Trigger Delay Register 4 (ADTRGDLR4) 4304
60.2.27 A/D Conversion Start Trigger Delay Register 5 (ADTRGDLR5) 4305
60.2.28 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 0 (ADSWNR0) 4306
60.2.29 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 1 (ADSWNR1) 4308
60.2.30 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 2 (ADSWNR2) 4310
60.2.31 Oversampling Mode (Same Channel Continuous Scan Mode) Decimation Control Register (ADDECCR) 4312
60.2.32 A/D Converter Accuracy Mode Register (ADACMDR) 4313

60.2.33 A/D Conversion Channel Configuration Register n (ADCHCRn) (n = 0 to 49) 4314
60.2.34 A/D Conversion Data Operation Control A Register n (ADDOPCRAn) (n = 0 to 49) 4317
60.2.35 A/D Conversion Data Operation Control B Register n (ADDOPCRBn) (n = 0 to 49) 4319
60.2.36 A/D Conversion Data Operation Control C Register n (ADDOPCRCn) (n = 0 to 49) 4321
60.2.37 Scan Group Diagnosis Function Control Register n (ADSGDCRn) (n = 0 to 11) 4323
60.2.38 Sampling State Table Register 0 (ADSSTR0) 4325
60.2.39 Sampling State Table Register 1 (ADSSTR1) 4325
60.2.40 Sampling State Table Register 2 (ADSSTR2) 4326
60.2.41 Sampling State Table Register 3 (ADSSTR3) 4326
60.2.42 Sampling State Table Register 4 (ADSSTR4) 4327
60.2.43 Sampling State Table Register 5 (ADSSTR5) 4327
60.2.44 Sampling State Table Register 6 (ADSSTR6) 4328
60.2.45 Sampling State Table Register 7 (ADSSTR7) 4328
60.2.46 A/D Conversion State Register (ADCNVSTR) 4329
60.2.47 Channel-Dedicated Sample-and-Hold Circuit Control Register 0 (ADSHCR0) 4330
60.2.48 Channel-Dedicated Sample-and-Hold Circuit Diagnosis Control Register 0 (ADSHDCR0) 4331
60.2.49 Channel-Dedicated Sample-and-Hold Circuit State Register 0 (ADSHSTR0) 4332
60.2.50 Channel-Dedicated Sample-and-Hold Circuit Single-Ended Input Mode Bypass Control Register (ADSHSBPCR) 4333
60.2.51 Channel-Dedicated Sample-and-Hold Circuit Differential Input Mode Bypass Control Register (ADSHDBPCR) 4335
60.2.52 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Control Register 0 (ADSHSDCR0) 4337
60.2.53 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Control Register 1 (ADSHSDCR1) 4339
60.2.54 Channel-Dedicated Sample-and-Hold Circuit Constant Sampling Control Register (ADSHCSCR) 4340
60.2.55 Internal Reference Voltage Monitor Enable Register (ADREFCR) 4341
60.2.56 A/D Converter Digital Filter Selection Register n (ADDFSRn) (n = 0 to 2) 4342
60.2.57 A/D Converter Self-Calibration State Register (ADCALSTCR) 4343
60.2.58 Channel-Dedicated Sample-and-Hold Circuit Self-Calibration State Register (ADCALSHCR) 4344
60.2.59 User Offset Table Register n (ADUOFTRn) (n = 0 to 7) 4345
60.2.60 User Gain Table Register n (ADUGTRn) (n = 0 to 7) 4346
60.2.61 Limiter Clip Interrupt Enable Register (ADLIMINTCR) 4346
60.2.62 Limiter Clip Table Register n (ADLIMTRn) (n = 0 to 7) 4347
60.2.63 Limiter Clip Scan Group Status Register (ADLIMGRSR) 4347
60.2.64 Limiter Clip Channel Status Register 0 (ADLIMCHSR0) 4348
60.2.65 Limiter Clip Channel Status Register 1 (ADLIMCHSR1) 4348
60.2.66 Extended Analog Limiter Clip Status Register (ADLIMEXSR) 4349

60.2.67 Limiter Clip Scan Group Status Clear Register (ADLIMGRSCR) 4350
60.2.68 Limiter Clip Channel Status Clear Register 0 (ADLIMCHSCR0) 4351
60.2.69 Limiter Clip Channel Status Clear Register 1 (ADLIMCHSCR1) 4351
60.2.70 Extended Analog Limiter Clip Status Clear Register (ADLIMEXSCR) 4352
60.2.71 Compare Match Enable Register (ADCMPENR) 4353
60.2.72 Compare Match Interrupt Enable Register (ADCMPINTCR) 4353
60.2.73 Composite Compare Match Configuration Register n (ADCCMPCRn) (n = 0 to 7) 4354
60.2.74 Compare Match Mode Selection Register 0 (ADCMPMDR0) 4355
60.2.75 Compare Match Mode Selection Register 1 (ADCMPMDR1) 4356
60.2.76 Compare Match Table Register n (ADCMPTBRn) (n = 0 to 7) 4357
60.2.77 Compare Match Table Status Register (ADCMPTBSR) 4357
60.2.78 Compare Match Table Status Clear Register (ADCMPTBSCR) 4358
60.2.79 Compare Match Channel Status Register 0 (ADCMPCHSR0) 4359
60.2.80 Compare Match Channel Status Register 1 (ADCMPCHSR1) 4359
60.2.81 Extended Analog Compare Match Status Register (ADCMPEXSR) 4360
60.2.82 Compare Match Channel Status Clear Register 0 (ADCMPCHSCR0) 4361
60.2.83 Compare Match Channel Status Clear Register 1 (ADCMPCHSCR1) 4361
60.2.84 Extended Analog Compare Match Status Clear Register (ADCMPEXSCR) 4362
60.2.85 A/D Converter Self-Calibration Start Register (ADCALSTR) 4363
60.2.86 A/D Conversion Synchronous Software Start Register (ADSYSTR) 4365
60.2.87 A/D Conversion Software Start Register n (ADSTRn) (n = 0 to 11) 4366
60.2.88 A/D Conversion Stop Register (ADSTOPR) 4367
60.2.89 A/D Conversion Status Register (ADSR) 4368
60.2.90 Scan Group Status Register (ADGRSR) 4369
60.2.91 Scan End Status Register (ADSCANENDSR) 4369
60.2.92 Scan End Status Clear Register (ADSCANENDSCR) 4370
60.2.93 A/D Conversion Error Status Register (ADERSR) 4370
60.2.94 A/D Conversion Error Status Clear Register (ADERSCR) 4371
60.2.95 A/D Converter Calibration End Status Register (ADCALENDSR) 4371
60.2.96 A/D Converter Calibration End Status Clear Register (ADCALENDSCR) 4372
60.2.97 A/D Conversion Overflow Error Status Register (ADOVFERSR) 4372
60.2.98 A/D Conversion Overflow Channel Status Register 0 (ADOVFCHSR0) 4373
60.2.99 A/D Conversion Overflow Channel Status Register 1 (ADOVFCHSR1) 4373
60.2.100 Extended Analog A/D Conversion Overflow Status Register (ADOVFEXSR) 4374
60.2.101 A/D Conversion Overflow Error Status Clear Register (ADOVFERSCR) 4375
60.2.102 A/D Conversion Overflow Channel Status Clear Register 0 (ADOVFCHSCR0) 4376
60.2.103 A/D Conversion Overflow Channel Status Clear Register 1 (ADOVFCHSCR1) 4376
60.2.104 Extended Analog A/D Conversion Overflow Status Clear Register (ADOVFEXSCR) 4377
60.2.105 FIFO Control Register (ADFIFOCR) 4378
60.2.106 FIFO Interrupt Control Register (ADFIFOINTCR) 4379

60.2.107 FIFO Interrupt Generation Level Register 0 (ADFIFOINTLR0) 4379
60.2.108 FIFO Interrupt Generation Level Register 1 (ADFIFOINTLR1) 4380
60.2.109 FIFO Interrupt Generation Level Register 2 (ADFIFOINTLR2) 4381
60.2.110 FIFO Interrupt Generation Level Register 3 (ADFIFOINTLR3) 4382
60.2.111 FIFO Interrupt Generation Level Register 4 (ADFIFOINTLR4) 4383
60.2.112 FIFO Interrupt Generation Level Register 5 (ADFIFOINTLR5) 4384
60.2.113 FIFO Status Register 0 (ADFIFOSR0) 4385
60.2.114 FIFO Status Register 1 (ADFIFOSR1) 4385
60.2.115 FIFO Status Register 2 (ADFIFOSR2) 4386
60.2.116 FIFO Status Register 3 (ADFIFOSR3) 4386
60.2.117 FIFO Status Register 4 (ADFIFOSR4) 4387
60.2.118 FIFO Status Register 5 (ADFIFOSR5) 4387
60.2.119 FIFO Data Clear Register (ADFIFODCR) 4388
60.2.120 FIFO Error Status Register (ADFIFOERSR) 4389
60.2.121 FIFO Error Status Clear Register (ADFIFOERSCR) 4390
60.2.122 A/D Data Register n (ADDRn) (n = 0 to 35) 4391
60.2.123 A/D Extended Analog Data Register n (ADEXDRn) (n = 0 to 2, 4 to 6, 8, 9, 16 to 18) 4392
60.2.124 FIFO Data Register n (ADFIFODRn) (n = 0 to 11) 4394
60.3 Operation 4395
60.3.1 A/D Conversion Clock 4395
60.3.2 A/D Converter Operation Mode 4395
60.3.2.1 SAR Mode 4395
60.3.2.2 Oversampling Mode 4395
60.3.2.3 Hybrid Mode 4396
60.3.3 Single-Ended Input and Differential Input 4396
60.3.4 Analog Channel 4397
60.3.5 Virtual Channel 4397
60.3.6 Scan Group 4398
60.3.7 Scan Mode 4399
60.3.7.1 Single Scan Mode 4399
60.3.7.2 Continuous Scan Mode 4399
60.3.7.3 Background Continuous Scan Mode 4400
60.3.7.4 Same Channel Continuous Scan Mode 4400
60.3.7.5 Fixed Channels Continuous Scan Mode 4400
60.3.8 Example of the Scanning Operation 4402
60.3.8.1 SAR Mode – Single Scan Mode 4402
60.3.8.2 SAR Mode – Continuous Scan Mode 4403
60.3.8.3 Oversampling Mode – Single Scan Mode 4404
60.3.8.4 Oversampling Mode – Continuous Scan Mode 4405
60.3.8.5 Oversampling Mode – Same Channel Continuous Scan Mode 4406

60.3.8.6 Hybrid Mode – Single Scan Mode 4407
60.3.8.7 Hybrid Mode – Continuous Scan Mode 4408
60.3.8.8 Hybrid Mode – Background Continuous Scan Mode 4410
60.3.8.9 Hybrid Mode – Fixed Channels Continuous Scan Mode 4412
60.3.9 Self-Calibration 4414
60.3.9.1 Conditions under Which Self-Calibration is Required 4414
60.3.9.2 Self-Calibration Procedure 4415
60.3.9.3 Restrictions on Self-Calibration 4416
60.3.10 Analog Input Channel 4417
60.3.11 Extended Analog Function 4417
60.3.12 Self-Diagnosis Function 4417
60.3.12.1 Self-Diagnosis 4417
60.3.12.2 Self-Diagnosis Mode 4418
60.3.12.3 Notes on Self-Diagnosis Function 4419
60.3.13 Internal Reference Voltage 4419
60.3.14 Temperature Sensor 4419
60.3.15 D/A Converter 4419
60.3.16 VBATT 1/6 Voltage Monitoring Output 4420
60.3.17 Channel-Dedicated Sample-and-hold Circuit 4420
60.3.17.1 Configuration of Channel-Dedicated Sample-and-Hold Circuit 4420
60.3.17.2 Operation in SAR Mode with Channel-Dedicated Sample-and-Hold Circuit 4422
60.3.17.3 Operation in Hybrid Mode with Channel-Dedicated Sample-and-Hold Circuit 4432
60.3.17.4 Restrictions on Channel-Dedicated Sample-and-Hold Circuit 4439
60.3.18 Disconnection Detection Assist Function 4441
60.3.19 Group Priority Operation 4444
60.3.19.1 Group Priority Operation in SAR Mode – Single Scan Mode 4444
60.3.19.2 Group Priority Operation in SAR Mode – Continuous Scan Mode 4446
60.3.19.3 Restrictions on Group Priority Operation 4450
60.3.20 Synchronous Operation 4451
60.3.20.1 Synchronous Operation Examples 4451
60.3.20.2 Restrictions on Synchronous Operation 4453
60.3.21 A/D Conversion Operation Monitor Function 4454
60.4 A/D Conversion Data 4455
60.4.1 Internal Data Processing Flow 4455
60.4.2 Digital Filter Function 4456
60.4.2.1 Configuration and Characteristics 4456
60.4.2.2 Operation of Digital Filter Function 4457
60.4.3 Calibration and Adjustment 4458
60.4.3.1 Gain Error and Offset Error Calibration 4458
60.4.3.2 User’s Gain Adjustment 4458
60.4.3.3 User’s Offset Adjustment 4460

60.4.4 A/D-Converted Value Addition/Averaging Function 4462
60.4.5 Limiter Clip Function 4463
60.4.6 Data Formatting process 4464
60.4.6.1 Signed/Unsigned Data Process 4464
60.4.6.2 Data Rounding Process 4464
60.4.7 Data Format 4465
60.4.8 Compare Match Function 4471
60.4.8.1 Compare Match 4471
60.4.8.2 Composite Compare Match 4472
60.4.9 Data Registers 4474
60.4.10 FIFO Function 4474
60.4.10.1 Operation When Writing A/D Conversion Data to FIFO 4475
60.4.10.2 Operation When Reading A/D Conversion Data from FIFO 4475
60.4.10.3 FIFO Data Read Request and Overflow 4475
60.4.10.4 Clearing FIFO Data 4475
60.4.10.5 Clearing FIFO Data by Starting/Resuming Scanning 4475
60.4.10.6 Usage Notes on FIFO 4476
60.4.11 A/D Conversion Data Output Decimation 4477
60.4.12 A/D Conversion Data Error Detection 4478
60.5 Start and Stop Control of A/D Conversion 4479
60.5.1 Software Trigger 4479
60.5.2 Peripheral Module Triggers 4479
60.5.2.1 ELC Trigger 4479
60.5.2.2 GPT Trigger 4480
60.5.2.3 External Trigger 4480
60.5.3 Trigger Delay 4482
60.5.4 Force Stops the A/D Conversion Operation 4482
60.5.4.1 Forced Stop Procedure 4482
60.5.4.2 Waiting Time after Disabling Trigger Input for Forced Stop Processing 4482
60.6 Error Detection 4484
60.6.1 A/D Converter Error 4484
60.6.2 A/D Conversion Overflow 4484
60.6.3 FIFO Overflow 4485
60.7 Procedure for Setting up and Changing 4486
60.7.1 Initial Setup Procedure 4486
60.7.2 Procedure for Changing ADCLK Settings 4486
60.7.3 Procedure for Changing the Settings of the A/D Converter 4488
60.7.4 Procedure for the A/D Converter Entering to and Returning from the Software Standby Mode 4489
60.8 Interrupt Sources 4490
60.9 Scan Conversion Time 4494

60.9.1 Scan Start Processing Time 4494
60.9.2 Conversion Processing Time 4496
60.9.3 Scan End Processing Time 4500
60.10 Usage Notes 4502
60.10.1 Prohibition of Changing the Operation Settings During A/D Conversion Operation 4502
60.10.2 Usage Notes on Forced Stop of A/D Conversion 4502
60.10.3 Usage Notes on A/D Data Registers 4502
60.10.4 Settings for the Module-Stop Function 4503
60.10.5 Settings for the Sleep Function 4503
60.10.6 Restrictions on Entering and Releasing the Low-Power States 4503
60.10.7 Notes on Board Design 4503
60.10.8 Notes on Synchronous Operation 4505
60.10.9 Notes on Channel-Dedicated Sample-and-Hold Circuit 4505
60.10.10 Restrictions on Analog Channel Shared Among Multiple A/D Converters 4506
60.10.11 Notes on A/D Conversion Start Trigger 4506
60.10.12 Notes on Self-Calibration 4506
60.10.13 Notes on Group Priority Operation 4506
60.10.14 Restrictions on SAR Mode 4506
60.10.15 Restrictions on Oversampling Mode 4506
60.10.16 Restrictions on Hybrid Mode 4507
60.10.17 Restrictions on the High Accuracy Mode 4508
60.10.18 Restrictions on Composite Compare Match 4508
60.10.19 Estimating the Sampling Error and Sampling Time 4509
60.10.20 Restriction on A/D Conversion Overflow in Single-Ended Input Mode 4511
61. 12-Bit D/A Converter (DAC12R) 4512
61.1 Overview 4512
61.2 Register Descriptions 4514
61.2.1 D/A Data Register (DADR) 4514
61.2.2 D/A Control Register 0 (DACR0) 4515
61.2.3 D/A Control Register 1 (DACR1) 4517
61.2.4 D/A Control Register 2 (DACR2) 4518
61.3 Operation 4519
61.3.1 D/A Conversion 4519
61.4 Linked Operation with the Use of the Event Link Function 4522
61.4.1 Action in Response to Receiving an Event Signal 4522
61.4.2 Usage Notes on Event Link Operation 4522
61.5 Usage Notes 4523
61.5.1 Module Stop Function Setting 4523
61.5.2 Operation of the 12-Bit D/A Converter in Module Stop State 4523
61.5.3 Operation of the 12-Bit D/A Converter in Software Standby Mode or Deep Software Standby Mode 4523

61.5.4 Notes on Output from the DAn Pin ... 4523
61.5.5 Notes on Simultaneous Output to the DAn Pin and Internal Peripheral Modules ... 4523
62. Temperature Sensor (TEMPS) ... 4524
62.1 Overview ... 4524
62.2 Register Descriptions ... 4525
62.2.1 Temperature Sensor Control Register (TSCR) ... 4525
62.2.2 Temperature Sensor Calibration Data Register (TSCDR) ... 4525
62.2.3 Temperature Sensor Calibration Data Register 2 (TSCDR2) ... 4526
62.3 Using the Temperature Sensor ... 4527
62.3.1 Preparation for Using the Temperature Sensor ... 4527
62.3.2 Procedure for Using the Temperature Sensor ... 4529
62.3.3 Timing of A/D Conversion of Temperature Sensor Output ... 4530
62.4 Usage Note ... 4531
62.4.1 Module-Stop Function Setting ... 4531
62.4.2 Settings for the Software Standby Mode ... 4531
63. Data Operation Circuit (DOCA) ... 4533
63.1 Overview ... 4533
63.2 Register Descriptions ... 4535
63.2.1 DOC Control Register (DOCR) ... 4535
63.2.2 DOC Status Register (DOSR) ... 4536
63.2.3 DOC Status Clear Register (DOSCR) ... 4536
63.2.4 DOC Data Input Register (DODIR) ... 4537
63.2.5 DOC Data Setting Register 0 (DODSR0) ... 4537
63.2.6 DOC Data Setting Register 1 (DODSR1) ... 4538
63.3 Operation ... 4539
63.3.1 Data Comparison Mode ... 4539
63.3.2 Data Addition Mode ... 4543
63.3.3 Data Subtraction Mode ... 4544
63.4 Interrupt Requests ... 4544
63.5 Event Link Output ... 4545
63.5.1 Interrupt Handling and Event Linking ... 4545
63.6 Usage Note ... 4545
63.6.1 Module Stop Function Setting ... 4545
64. SRAM ... 4546
64.1 Overview ... 4546
64.2 Register Descriptions ... 4547
64.2.1 SRAMn Security Attribute Boundary Address Register (SRAMSABARn) (n = 0, 1) ... 4547
64.2.2 SRAM Security Attribution Register (SRAMSAR) ... 4548
64.2.3 SRAM Protection Control Register for Secure (SRAMPRCR_S) ... 4549
64.2.4 SRAM Protection Control Register for Non-Secure (SRAMPRCR_NS) ... 4550

64.2.5 SRAM Wait State Control Register (SRAMWTSC) 4551
64.2.6 SRAM Control Register n (SRAMCRn) (n = 0, 1) 4552
64.2.7 SRAM ECC Region Control Register 0 (SRAMECCRGN0) 4553
64.2.8 SRAM ECC Region Control Register 1 (SRAMECCRGN1) 4554
64.2.9 SRAM Error Status Register (SRAMESR) 4555
64.2.10 SRAM Error Status Clear Register (SRAMESCLR) 4556
64.2.11 SRAMn Error Address m Register (SRAMEARnm) (n = 0, 1; m = 0, 1) 4557
64.3 Operation 4558
64.3.1 ECC Function 4558
64.3.2 ECC Error Interrupt Function 4558
64.3.3 ECC Decoder Testing 4559
64.3.4 Interrupt Source 4559
64.3.5 Wait State 4560
64.3.6 Access Cycle 4560
64.4 Usage Notes 4561
64.4.1 Low Power Consumption Function 4561
64.4.2 Instruction Fetch from SRAM Area 4561
64.4.3 Notes when using the SRAM error check function 4561
65. MRAM 4562
65.1 Overview 4562
65.2 Structure of Memory 4565
65.3 Address Space 4568
65.4 MRAM Read and Program Function 4569
65.4.1 Prefetch Buffer 4569
65.4.2 Code MRAM Programming Procedure 4569
65.4.3 Data MRAM Programming Procedure 4574
65.4.4 MRAM Programming Procedure with MRAM Program Ready Interrupt 4577
65.4.5 Frequency Change Procedure 4581
65.5 Register Descriptions 4583
65.5.1 Code MRAM Prefetch Buffer Enable Register (MRCPFB) 4583
65.5.2 Code MRAM Frequency Notifications Register (MRCFREQ) 4584
65.5.3 Extra MRAM Frequency Notifications Register (MREFREQ) 4585
65.5.4 Code MRAM ECC Decoder Control Register (MRCDECC) 4586
65.5.5 Code MRAM Read Access Error Interrupt Enable Register (MRCRAEINT) 4587
65.5.6 Code MRAM Read Access Error Status Register (MRCRAES) 4588
65.5.7 Code MRAM TED Error Address Register (MRCRTEA) 4589
65.5.8 Code MRAM DEC Error Address Register (MRCRDEA) 4590
65.5.9 Extra MRAM ECC Decoder Control Register (MREDECC) 4591
65.5.10 Extra MRAM Read Access Error Interrupt Enable Register (MRERAEINT) 4592
65.5.11 Extra MRAM Read Access Error Status Register (MRERAES) 4593
65.5.12 Extra MRAM TED Error Address Register (MRERTEA) 4594

65.5.13 Extra MRAM DEC Error Address Register (MRERDEA) 4595
65.5.14 MRAM Security Attribution Register (MSAR) 4596
65.5.15 Extra MRAM Zeroization Status Register (MREZS) 4598
65.5.16 Extra MRAM Zeroization Control Register (MREZC) 4599
65.5.17 Extra MRAM Access Status Register (MASTAT) 4600
65.5.18 Extra MRAM Access Error Interrupt Enable Register (MPAEINT) 4601
65.5.19 Extra MRAM Ready Interrupt Enable Register (MRDYIE) 4602
65.5.20 MACI Command Start Address Register (MSADDR) 4603
65.5.21 MRAM Counter Select Register (MCNTSELR) 4604
65.5.22 MRAM Counter Data Register n (MCNTDTRn) (n = 0, 1) 4605
65.5.23 MRAM Configuration Update Transfer Control Register (MCTRCNTR) 4606
65.5.24 MRAM Configuration Update Transfer Status Register (MCTRSTATR) 4607
65.5.25 Extra MRAM Status Register (MSTATR) 4608
65.5.26 Extra MRAM Program Mode Entry Register (MENTRYR) 4611
65.5.27 Extra MRAM Sequencer Set-Up Initialization Register (MSUINITR) 4612
65.5.28 MACI Command Register (MCMDR) 4613
65.5.29 MRAM Startup Area Select Monitor Register (MSUASMON) 4614
65.5.30 MRAM Startup Area Control Register (MSUACR) 4615
65.5.31 MACI Reset Transfer Status Register (MRSTTRST) 4616
65.5.32 MRAM Program Speed Control Register (MRPSC) 4617
65.5.33 Extra MRAM ECC Encoder Control Register (MREEECC) 4618
65.5.34 Code MRAM Program Control Register for Non-Secure Application (MRCPCNS) 4619
65.5.35 Code MRAM Program Control Register for Secure Application (MRCPCS) 4620
65.5.36 Code MRAM Block Protection Register for Non-Secure Application (MRCBPROTNS) 4621
65.5.37 Code MRAM Block Protection Register for Secure Application (MRCBPROTS) 4622
65.5.38 Code MRAM Program Status Register (MRCPS) 4623
65.5.39 Code MRAM Program Access Error Interrupt Enable Register (MRCPAEINT) 4625
65.5.40 Code MRAM Program Error Address Register (MRCPEA) 4626
65.5.41 Code MRAM Flush Register (MRCFLR) 4627
65.5.42 Code MRAM Program Ready Interrupt Control Register (MRCPRICR) 4628
65.5.43 Data MRAM Program Status Register (MRDPS) 4629
65.5.44 Data MRAM Program Access Error Interrupt Enable Register (MRDPAEINT) 4631
65.5.45 Data MRAM Program Error Address Register (MRDPEA) 4632
65.5.46 Data MRAM Flush Register (MRDFLR) 4633
65.5.47 Data MRAM Program Ready Interrupt Control Register (MRDPRICR) 4634
65.5.48 Code MRAM ECC Encoder Control Register (MRCEECC) 4635
65.5.49 Unique ID Register n (UIDRn) (n = 0 to 3) 4636
65.5.50 Part Numbering Register n (PNRn) (n = 0 to 3) 4637
65.5.51 MCU Version Register (MCUVER) 4638
65.5.52 MRAM-OTP Write Mode Control Register (MWMCR) 4639
65.6 Operating Mode 4640

65.6.1 MRAM Read/Program Mode 4640
65.6.2 MRAM High-Speed Program Mode 4642
65.7 MACI Commands 4643
65.7.1 Programmable Area with MACI Commands 4643
65.7.2 List of MACI Commands 4644
65.7.3 Relationship Between the Extra MRAM Sequencer State and MACI Commands 4645
65.7.4 Usage of MACI Commands 4646
65.7.4.1 Overview of Command Usage in Extra MRAM Program Mode 4646
65.7.4.2 Transition to Extra MRAM Program Mode 4647
65.7.4.3 Transition to Read Mode 4648
65.7.4.4 Recovery from the Command-Locked State 4649
65.7.4.5 Program Command 4650
65.7.4.6 Status Clear Command 4652
65.7.4.7 Forced Stop Command 4653
65.7.4.8 Configuration Set Command 4654
65.7.4.9 Increment Counter Command 4656
65.7.4.10 Read Counter Command 4657
65.8 Security Function 4658
65.8.1 Permanent OFS Protect Setting 4658
65.8.2 Security Flag for Startup Area and Startup Size Select 4659
65.8.3 Permanent Block Protect Setting 4659
65.8.4 MRAM Protection for Protect Zone 4663
65.8.4.1 Protection for Code MRAM and Data MRAM Areas (Program) 4663
65.8.4.2 Protection for extra MRAM Area (Program) 4663
65.8.4.3 Protection for MRAM Area (Read) 4664
65.8.4.4 Protection for Registers 4665
65.8.5 Anti-Rollback Counter 4667
65.8.6 HUK Protection 4668
65.8.7 OEM Root Key Protection 4668
65.9 Protection Function 4669
65.9.1 Software Protection 4669
65.9.1.1 Code MRAM Program Protection 4669
65.9.1.2 Block Protection of Code MRAM 4669
65.9.1.3 Block Protection of Data MRAM 4674
65.9.2 Error Protection 4675
65.9.3 Startup Program Protection 4677
65.9.4 Dual Bank Function 4681
65.9.4.1 Bank Mode Switching Function 4681
65.9.4.2 Startup Bank Selection Function 4682
65.9.5 Block Swap Function 4684
65.9.5.1 Block Swap Selection 4685

65.9.6 Fault Correction and Detection ... 4688
65.9.6.1 Test Method for ECC Circuits ... 4688
65.10 Configuration Update Transfer ... 4692
65.11 Boot Mode ... 4693
65.11.1 Boot Mode (SCI Interface) ... 4694
65.11.2 Boot Mode (USB Interface) ... 4695
65.11.3 Boot Mode (FINE Interface) ... 4696
65.11.3.1 Operating Conditions in Boot Mode (FINE Interface) ... 4696
65.12 Using the Serial Programmer for Rewriting ... 4697
65.12.1 Environments for Serial Programming ... 4697
65.13 Programming through Self-Programming ... 4698
65.13.1 Overview ... 4698
65.13.2 Parallel Accessibility ... 4699
65.13.3 Reading MRAM ... 4699
65.14 Usage Notes ... 4700
65.14.1 Area where Programming was Interrupted by W-HUK Zeroing ... 4700
65.14.2 Actions Prohibited during Programming ... 4700
65.14.3 Prohibition of Additional Programming ... 4700
65.14.4 Resets During Programming ... 4700
65.14.5 Allocation of Vectors for Interrupts and Other Exceptions During Programming ... 4700
65.14.6 Synchronization of Code MRAM Program Completion and Subsequent Instruction Generation ... 4700
65.14.7 Constraints During Programming of MRAM ... 4700
65.14.8 Power Supply in Any Type of Boot Mode ... 4701
66. Internal Voltage Regulator ... 4702
66.1 Overview ... 4702
66.2 Operation ... 4702
66.2.1 DCDC Mode ... 4702
66.2.2 External VDD Mode ... 4703
66.2.3 LDO Mode ... 4705
66.3 Usage Notes ... 4705
67. Electrical Characteristics ... 4706
67.1 Test Conditions ... 4706
67.1.1 Maximum and Minimum Values (Max./Min.) ... 4706
67.1.2 Typical Values (Typ.) ... 4706
67.2 Absolute Maximum Ratings ... 4707
67.3 Recommended Operating Conditions ... 4708
67.4 DC Characteristics ... 4709
67.4.1 Characteristics of the Input Voltage of the I/O Pins ... 4709
67.4.2 Characteristics of the Output Voltage of the I/O Pins ... 4712
67.4.3 Characteristics of the Other I/O Pins ... 4717
67.4.4 Operating and Standby Currents ... 4717

67.4.4.1 Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = A, B, C, D) ... 4722
67.4.4.2 Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = E, F) ... 4730
67.4.4.3 Currents in Deep Software Standby Mode ... 4740
67.4.4.4 Other Operating Currents ... 4743
67.4.4.5 Permissible Output Currents ... 4744
67.4.5 Thermal Characteristics ... 4749
67.4.5.1 Thermal Resistance Value ... 4749
67.4.5.2 Supply Currents for the Peripheral Functions ... 4750
67.4.5.3 Example of Calculation of T_a ... 4751
67.4.6 VCC Rising and Falling Slopes and Ripple Frequency ... 4753
67.5 AC Characteristics ... 4754
67.5.1 Operating Frequency ... 4754
67.5.2 Clock Timing ... 4755
67.5.3 Reset Timing ... 4762
67.5.4 Timing of Return from Low Power Consumption Modes ... 4764
67.5.5 Control Signal Timing ... 4772
67.5.6 Bus Timing ... 4773
67.5.7 Timing of On-Chip Peripheral Modules ... 4789
67.5.7.1 I/O Port ... 4789
67.5.7.2 TMR ... 4789
67.5.7.3 CMTW ... 4790
67.5.7.4 POEG ... 4790
67.5.7.5 GPTW ... 4791
67.5.7.6 RSCI ... 4792
67.5.7.7 RSPIA ... 4800
67.5.7.8 XSPI ... 4806
67.5.7.9 RIICHS ... 4812
67.5.7.10 RI3C ... 4816
67.5.7.11 SSIE ... 4826
67.5.7.12 SD/MMC Host Interface ... 4829
67.5.7.13 PDC ... 4830
67.5.7.14 HRPWM ... 4832
67.5.7.15 CANFD/CANFDG ... 4832
67.5.7.16 ESWM ... 4833
67.5.7.17 PDMIF ... 4840
67.5.7.18 PCIF ... 4841
67.5.7.19 PIO ... 4843
67.5.7.20 A/D Converter Trigger ... 4844
67.5.7.21 CAC ... 4844
67.6 USB Characteristics ... 4845

67.6.1 USB Timing 4845
67.6.2 USBHS Timing 4847
67.7 A/D Conversion Characteristics 4850
67.7.1 Analog Input Voltage Range 4850
67.7.2 Common Characteristics of the 16-Bit A/D Converter 4851
67.7.3 SAR Mode Characteristics 4853
67.7.4 Common Characteristics of Oversampling Mode and Hybrid Mode 4857
67.7.5 Oversampling Mode Characteristics 4859
67.7.6 Hybrid Mode Characteristics 4861
67.8 D/A Conversion Characteristics 4863
67.9 Temperature Sensor Characteristics 4865
67.10 Power-on Reset Circuit and Voltage Detection Circuit Characteristics 4867
67.11 Core Voltage Monitoring Reset Characteristics 4870
67.12 Oscillation Stop Detection Timing 4871
67.13 External VDD Timing Characteristics 4872
67.14 Battery Backup Function Characteristics 4873
67.15 MRAM Characteristics 4875
67.15.1 Code MRAM Characteristics 4875
67.15.2 Option-Setting Memory (MRAM Area) Characteristics 4875
67.15.3 Option-Setting Memory (OTP Area with ECC) Characteristics 4876
67.15.4 Option-Setting Memory (OTP Area without ECC) Characteristics 4876
67.15.5 MACI Command Characteristics 4877
67.15.6 W-HUK Zeroization Characteristics 4877
67.15.7 MRAM Magnetic Field Resistance Characteristics 4877
67.15.8 Data MRAM Characteristics 4878
67.16 Boundary Scan 4879
67.17 Usage Note 4880
67.17.1 Connecting VCL Capacitor and Bypass Capacitors 4880
Appendix 1. Port States in Each Processing Mode 4881
Appendix 2. Package Dimensions 4883
Appendix 3. Cycles for Access 4889
REVISION HISTORY 4893

600-MHz 32-bit RX MCU, on-chip double-precision floating-point coprocessor, 3604 CoreMark, arithmetic unit for trigonometric functions, up to 1-MB code MRAM (supporting the dual bank function), 1.5-MB SRAM, Ethernet MAC unit compliant with IEEE 802.3, SD/MMC host interface, expanded serial peripheral interfaces, various communications interfaces including CAN FD, 16-bit A/D converters, RTC, encryption functions, serial sound interfaces

Features

■ 32-bit RXv3e CPU core

- Maximum operating frequency: 600 MHz
  Capable of 3604 CoreMark performance in operation at 600 MHz
- 64-bit double-precision IEEE-754 floating point
- A collective register bank save function is available.
- Supports memory-protection units (MPUs)
- JTAG, SWD, and FINE (one-line) debugging interfaces
- Supports the Protect Zone function
- Instruction cache and data cache: 32 Kbytes each; support for ECC
- Two local RAMs: 128 Kbytes each; support for ECC

■ Low-power design and architecture

- Operation from a single 1.62- to 3.63-V supply
- The RTC and backup registers are capable of operation from a battery power supply.
- Seven low power consumption modes

■ On-chip user extra MRAM

- 128 Kbytes (switchable to operate as MTJ-OTP memory)
- User code is programmable by on-board or off-board programming.
- Capable of holding the user boot program, data, etc.

■ On-chip code MRAM

- Devices with up to 1 Mbyte are available.
- No wait cycles when the prefetch buffer is hit
- Supports reading at 100 MHz
- User code is programmable by on-board or off-board programming.
- Programming as a background operation (BGO)
- A dual-bank structure allows swapping the start-up bank.

■ On-chip data MRAM

- 48 Kbytes; programmable up to 100,000 times
- Programming as a background operation (BGO)

■ On-chip SRAM

- 1.5 Mbytes; support for ECC and reading at 150 MHz

■ Data transfer

- Eight DMA controllers (DMACs), DTCA, two EXDMA controllers (EXDMACs), Ethernet DMAC (GWCA), and two XSPIs serving as target modules

■ Reset and supply management

- Power-on reset (POR)
- Programmable voltage detection (PVD)
- Core voltage monitoring (CVM)
- Temperature monitoring reset
- Backup area reset
- Reset output function

■ Clock functions

- Main clock oscillator for an 8- to 48-MHz external resonator
- Sub-clock oscillator for a 32.768-kHz crystal resonator
- Two PLL circuits
- 8-MHz MOCO, HOCO with 16, 18, 20, 32, or 48 MHz selectable, and 32.768-kHz LOCO
- Clock output function

■ Realtime clock

- Adjustment functions (for 30 seconds, leap year, and time error)
- Realtime clock counting and binary counting modes are selectable.
- Time capture in response to event-signal input

■ Independent watchdog timer

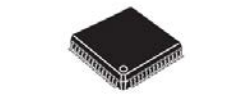
- Operation at 16.384 kHz. Operation in deep software standby mode 1 is possible.

■ Useful functions for IEC60730 compliance

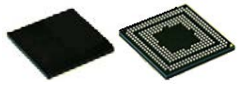
- Oscillation stop detection (MOSC, SOSC), clock frequency accuracy measurement circuits, CRCA, IWDTa, DOCA, self-diagnosis and disconnection detection assist functions for the A/D converters, etc.
- Register write protection function protects important registers against overwriting.

■ External address space

- Buses for high-speed data transfer (max. operating frequency of 80 MHz)
- 8 CS areas with 8-, 16-, or 32-bit bus space selectable per area
- Independent SDRAM area (128 Mbytes) and XSPI areas (1 Gbyte)



PLQP0176KJ-A 24 × 24 mm, 0.5-mm pitch
PLQP0144KA-B 20 × 20 mm, 0.5-mm pitch
PLQP0100KP-A 14 × 14 mm, 0.5-mm pitch



PLBG0320GD-A 17 × 17 mm, 0.8-mm pitch
PLBG0224GE-A 13 × 13 mm, 0.8-mm pitch
PLBG0176GG-A 13 × 13 mm, 0.8-mm pitch

■ Various communications interfaces

- One Ethernet MAC unit with a single port, compliant with IEEE 802.3
- One high-speed USB 2.0 port for host/function, OTG controller, or BC 1.2, with on-chip PHY layer
- One full-speed USB 2.0 port for host/function or OTG controller, with on-chip PHY layer
- Three CAN FD modules compliant with the specifications of ISO 11898-1:2015
- 13 RSCIs with 16-byte transmission and reception FIFO buffers
  The serial communications mode can be selected from among asynchronous, clock synchronous, smart card interface, simple SPI, simple I^2C, and extended serial.
  Communications in Manchester code available, support for HBS functionality
- Two I^2C-bus interfaces for transfer at up to 3.4 Mbps
- I3C bus interface for transfer at up to 12 Mbps in SDR mode
- Three RSPIAs with single I/O for transmission and reception at up to 50 Mbps
- Parallel data capture unit (PDC) for the CMOS camera interface
- SD/MMC host interface with a 1- or 4-bit SD bus for use with SD memory or SDIO
- Two SSIEs for various audio formats such as I^2S
- Two XSPIs for transfer at up to 266 MT/s, switchable between controller and target, support for 1-, 2-, 4-, and 8-bit protocols and decryption on the fly (DOTF)
- PDM-IF connectable with an external microphone

■ Up to 26 extended-function timers

- 16 32-bit general PWM timers (GPTWs), two 8-bit timers (TMRs) with 2 channels per unit, two 16-bit CMTs with 2 channels per unit, two 32-bit compare match timers W (CMTWs) with a single channel per unit

■ Three 16-bit A/D converters

- Up to 36 analog input channels
- Self-diagnosis, detection of analog input disconnection

■ Two 12-bit D/A converters

■ Temperature sensor for measuring temperature within the chip

■ Arithmetic unit for trigonometric functions

■ Two phase-counting interfaces

■ Security features

- Renesas Secure IP (RSIP-E50D) (true-random number generator (TRNG), AES, ChaCha20-Poly1305, RSA, ECC, SHA-512, SHA3-512, and 128-bit unique ID)
- Zeroization
- Device lifecycle management
- Support for decryption on the fly (DOTF)
- Secure boot
- Detection of tampering
- The security attribute is specifiable per pin.
- Protect Zone function

■ Up to 197 pins for general I/O ports

- 5-V tolerance, open drain, input pull-up, switchable driving ability
- Software configurable 32-bit programmable I/O port

■ Operating temperature range

- Tj = –40 to +125°C

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 give a comparison of the functions of products in different packages.
Table 1.1 is an outline of maximum specifications, and the peripheral modules and the number of channels of the modules differ depending on the number of pins on the package. For details, refer to Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/10)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 600 MHz • 32-bit RX CPU (RXv3e) • Minimum instruction execution time: One instruction per state (cycle of the system clock) • Address space: 4-Gbyte linear • Protect Zone: Three worlds (non-secure, secure, and root) • Register set of the CPU General purpose: Sixteen 32-bit registers Control: Fourteen 32-bit registers Accumulator: Two 72-bit registers • 117 instructions Instructions installed as standard: 111 Basic instructions: 77 Single-precision floating-point operation instructions: 11 DSP instructions: 23 Instructions for register bank save function: 2 Protect Zone instructions: 4 • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 × 32 → 64 bits • On-chip divider: 32 / 32 → 32 bits • Barrel shifter: 32 bits • Instruction cache and data cache: 32 Kbytes each • Accelerator for branch instruction processing: 272 bytes
	FPU	• Single-precision floating-point numbers (32 bits) • Data types and floating-point exceptions in conformance with the IEEE754 standard
	Double-precision floating point coprocessor	• Double-precision floating-point register set Double-precision floating-point data registers: 16, each with 64-bit width Double-precision floating-point control registers: Four, each with 32-bit width • Double-precision floating-point processing instructions: 21 • Notifying the interrupt controller of double-precision floating-point exceptions
	Register bank save function	• Fast collective saving and restoration of the values of CPU registers • Sixteen save register banks are incorporated in each of the root world, secure world, and non-secure world.

Table 1.1 Outline of Specifications (2/10)

Classification	Module/Function	Description
Memory	Code MRAM	• Capacity: 512 Kbytes/1 Mbyte • On-board programming: Five types • A dual-bank structure allows programming during reading or exchanging the start-up areas
	Data MRAM	• Capacity: 48 Kbytes • Programming: 100,000 times
	User extra MRAM	• Capacity: 128 Kbytes • On-board programming: Five types • Switchable to operate as MTJ-OTP
	Unique ID	• 128-bit unique ID for each device of this product family
	RAM	• Capacity: 1.5 Mbytes • Error-correcting code (ECC), single-error correcting and double-error detecting code (SEC-DED)
	Local RAM	• Capacity: 128 Kbytes × 2 • Up to 600 MHz, no-wait access • Error-correcting code (ECC), single-error correcting and double-error detecting code (SEC-DED)
Operating modes		• Operating modes by the mode-setting pins at the time of release from the reset state Single-chip mode Boot mode (SCI interface) Boot mode (USB interface) Boot mode (FINE interface) Boot mode (JTAG interface) Boot mode (SWD interface) • Selection of operating mode by register setting Single-chip mode • Endian selectable
Clocks	Clock generation circuit	• Main clock oscillator, sub-clock oscillator, low-speed/middle-speed/high-speed on-chip oscillator, and PLL frequency synthesizer (two circuits) • The peripheral module clocks can be set to frequencies above that of the system clock. • Main-clock oscillation stoppage detection • Sub-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the CPU clock (CPUCLK0), system clock (ICLK), MRAM code bus clock (MRICLK), MRAM peripheral clock (MRPCLK), peripheral module clocks (PCLKA, PCLKB, PCLKD, PCLKE), and external bus clock (BCLK) The CPU and local RAM run in synchronization with CPUCLK0: Up to 600 MHz The MRAM bus and code MRAM run in synchronization with MRICLK: Up to 300 MHz The data MRAM runs in synchronization with MRPCLK: Up to 150 MHz The RAM and other bus masters run in synchronization with the system clock (ICLK): Up to 300 MHz Peripheral modules of CRCA, DOCA, RSPIA, RSCI, ESWM, GPTW, HRPWM, POEG, ADC16H, CANFDG, CANFD, RSIP, RI3C, RIICHS, and USBHS run in synchronization with PCLKA, which operates at up to 150 MHz. Other peripheral modules run in synchronization with PCLKB: Up to 75 MHz The counter reference clock for the GPTW runs in synchronization with PCLKD: Up to 150 MHz The CANFDRAM runs in synchronization with PCLKE: Up to 300 MHz Devices connected to the external bus run in synchronization with BCLK: Up to 150 MHz • The high-speed on-chip oscillator (HOCO) can be obtained through frequency-multiplication of the PLL1 or PLL2 reference clock • External clock input frequency for the main clock oscillator: Up to 48 MHz • External clock input frequency for the sub-clock oscillator: 32.768 kHz • Clock output function

Table 1.1 Outline of Specifications (3/10)

Classification	Module/Function	Description
Resets		Sixteen types of reset • RES# pin reset: Triggered when the RES# pin is driven low. • Power-on reset: Triggered when the RES# pin is driven high and VCC rises above specific levels. • Core voltage monitoring reset: Triggered when the core voltage falls below or rises above specific levels. • Voltage-monitoring 0 reset: Triggered when VCC falls below specific levels. • Voltage-monitoring 1, 2, 4, and 5 resets: Triggered when VCC falls below or rises above specific levels. • Temperature-monitoring reset: Triggered due to the detection of temperatures outside the specified range. • Deep software standby reset: Triggered in response to an interrupt to trigger release from deep software standby. • Independent watchdog timer reset: Triggered when the independent watchdog timer underflows, or a refresh error occurs. • Watchdog timer reset: Triggered when the watchdog timer underflows, or a refresh error occurs. • Software reset: Triggered by register setting. • Bus error reset: Triggered in response to an illicit address or size error • Common memory error reset: Triggered in response to an ECC error in the RAM • Local memory error reset: Triggered in response to an ECC error in the CPU cache or local RAM Reset output is available.
Power-on reset		• If the RES# pin is at the high level when power is supplied, an internal reset is generated. After VCC has exceeded the voltage detection level and the specified period has elapsed, the reset is cancelled.
Programmable voltage detection circuit (PVD)		Monitors the voltage being input to the VCC pins and generates an internal reset or interrupt. • Voltage detection circuit 0 Capable of generating an internal reset The option-setting memory can be used to select enabling or disabling of the reset. Voltage detection level: Selectable from seven different levels (2.85 V, 2.58 V, 2.15 V, 2.00 V, 1.90 V, 1.80 V, 1.67 V) • Voltage detection circuits 1, 2, 4 and 5 Voltage detection level: Selectable from twelve different levels (3.14 V, 3.10 V, 3.08 V, 2.85 V, 2.83 V, 2.80 V, 2.62 V, 2.33 V, 1.90 V, 1.86 V, 1.74 V, 1.71 V) Digital filtering (1/2, 1/4, 1/8, and 1/16 LOCO frequency) Capable of generating an internal reset • Two types of timing are selectable for release from reset An internal interrupt can be requested only from voltage detection circuits 1 and 2. • Detection of voltage rising above and falling below thresholds is selectable. • Maskable or non-maskable interrupt is selectable Voltage detection monitoring Event linking
Low power consumption	Low power consumption function	• Module stop function • Seven low power consumption modes Sleep mode, deep sleep mode, software standby modes 1 and 2, and deep software standby modes 1 to 3
	Battery backup function	• When the voltage on the VCC pin drops below a certain point, battery power from the VBATT pin is supplied to the backup power area. • Backup power area: sub clock oscillator real-time clock backup register Tamper detection • Detection of voltage drop in the backup power area

Table 1.1 Outline of Specifications (4/10)

Classification	Module/Function	Description
Interrupt	Interrupt controller (ICUM)	• Number of interrupt vectors: 256 • External interrupts: 32 (pins IRQ0 to IRQ31) • Software interrupts: 2 sources • Non-maskable interrupts: 13 sources • 32 levels specifiable for the order of priority • Method of interrupt source selection: The interrupt vectors consist of 256 vectors (128 sources are fixed. The remaining 128 vectors are selected from among the other 342 sources.)
External bus extension		• The external address space can be divided into eight areas (CS0 to CS7), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7) A chip-select signal (CS0# to CS7#) can be output for each area. Each area is specifiable as an 8-, 16-, or 32-bit bus space. The data arrangement in each area is selectable as little or big endian (only for data). • SDRAM interface connectable • Bus format: Separate bus, multiplex bus • Wait control • Write buffer facility • Connection for a memory device compliant with the expanded serial peripheral interface (xSPI) is possible (JEDEC Standards JESD 251, JESD 251-1, and JESD 252).
DMA	DMA controller (DMACAa)	• 8 channels • Four transfer modes: Normal transfer, repeat transfer, block transfer, and repeat block transfer • Request sources: Software trigger and interrupt requests from peripheral functions
	EXDMA controller (EXDMACAa)	• 2 channels • Four transfer modes: Normal transfer, repeat transfer, block transfer, and repeat block transfer • Request sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	Data transfer controller (DTCA)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: External interrupts and interrupt requests from peripheral functions • Sequence transfer
I/O ports	Programmable I/O ports	• I/O ports for the 320-pin LFBGA I/O pins: 197 Input pins: 3 Pull-up resistors: 198 Open-drain outputs: 197 5-V tolerant pins: 16 • I/O ports for the 224-pin LFBGA I/O pins: 159 Input pins: 3 Pull-up resistors: 160 Open-drain outputs: 159 5-V tolerant pins: 16 • I/O ports for the 176-pin LFBGA and 176-pin LFQFP I/O pins: 120 Input pins: 3 Pull-up resistors: 121 Open-drain outputs: 120 5-V tolerant pins: 16 • I/O ports for the 144-pin LFQFP I/O pins: 100 Input pins: 3 Pull-up resistors: 101 Open-drain outputs: 100 5-V tolerant pins: 16 • I/O ports for the 100-pin LFQFP I/O pins: 72 Input pins: 3 Pull-up resistors: 73 Open-drain outputs: 72 5-V tolerant pins: 9

Table 1.1 Outline of Specifications (5/10)

Classification	Module/Function	Description
I/O ports	Software configurable 32-bit programmable I/O port (PIO)	• Of the 64 port pins, up to 32 can be selected for simultaneous use. • Input or output can be enabled, and for outputs, the high or low level can be specified through register settings.
Event link controller (ELC)		• Event signals such as interrupt request signals can be interlinked with the operation of functions such as timer counting, eliminating the need for intervention by the CPU to control the functions. • 377 internal event signals can be freely combined for interlinked operation with connected functions. • Event signals from peripheral modules can be used to change the states of output pins (of ports B, C, D, and E). • Changes in the states of pins (of ports B, C, D, and E) being used as inputs can be interlinked with the operation of peripheral modules.
Timers	General PWM timer (GPTW)	• 32 bits × 16 channels (GPTW0 to GPTW15) • Counting up or down (sawtooth-wave), counting up and down (triangle-wave) selectable for all channels • Clock sources independently selectable for each channel • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Generation of dead times in PWM operation • Generation of highly accurate PWM waveforms with duty cycles close to the settings for values in the vicinity of duty 0% and 100%. • In output compare operation, settings of the comparison registers are immediately reflected, and the generation of PWM waveforms with secured dead times is possible. • Switching the direction of counting during operations in the triangle-wave PWM mode is possible. • The counter can be synchronously started, stopped, cleared, or have its direction of counting switched in any channel. • The counter can be started, stopped, cleared, or have its direction of counting switched, incremented, or decremented in response to up to eight ELC events. • The counter can be started, stopped, cleared, or have its direction of counting switched, incremented, or decremented in response to detecting a state transition in one event input signal while another input signal is at a particular level. • The counter can be started, stopped, cleared, or have its direction of counting switched, incremented, or decremented in response to up to four external triggers. • Disabling of output on detection of a dead time error or a short-circuit among output pins is possible. • Using a timer to generate a trigger signal to start A/D conversion is possible. The trigger signal can be output to the external pin is possible. • Output of events, such as compare-matches from A to F, overflow, and underflow, to the ELC is possible. • The noise filter of input capture is available. • Cycle counting • Measurement of the pulse width of the input signal • Output of the result of a logical operation on the output signal from one channel and that from another channel is possible. • Synchronous counter setting/counter clearing/input capture among channels • The timer-linked synchronization signal output function enables synchronous operation between channels synchronized with the timer period, and synchronous operation with other IPs synchronized with the timer period. • GTIOCnA/GTIOCnB pin input/output polarity inversion
	High resolution PWM waveform generation circuit (HRPWMb)	• Finely adjusting a PWM waveform generated by GPTW0 to GPTW3 with a minimum resolution of approximately 49 ps is possible.

Table 1.1 Outline of Specifications (6/10)

Classification	Module/Function	Description
Timers	GPTW port output enable (POEGd)	Various types of request to stop output from general PWM timers (GPTWs) are available. • Stopping output in response to detecting the input level on the GTETRGx pin • Stopping output in response to an output stopping signal from the GPTW • Stopping output in response to detecting the stopping of oscillation, or to requests by software • Stopping output in response to detecting the input level or an edge on the POEG pin
	8-bit timer (TMRa)	• (8 bits × 2 channels) × 2 units • Select from among seven internal clock signals (PCLKB/1, PCLKB/2, PCLKB/8, PCLKB/32, PCLKB/64, PCLKB/1024, PCLKB/8192) and one external clock signal • Capable of output of pulse trains with desired duty cycles or of PWM signals • The 2 channels of each unit can be cascaded to create a 16-bit timer • Capable of generating baud-rate clocks for RSCI • Event linking by the ELC
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512)
	Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events. • Event linking by the ELC
	Realtime clock (RTCBb)*1	• Clock sources: Low-speed on-chip oscillator, sub-clock • Selection of the 32-bit binary count in time count/second unit possible • Clock and calendar functions • Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt • Battery backup operation • Time-capture facility for three values • Event linking by the ELC
	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among 6 counter-input clock signals (PCLKB/4, PCLKB/64, PCLKB/128, PCLKB/512, PCLKB/2048, PCLKB/8192) • Event linking by the ELC
	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Counter-input clock: Low-speed on-chip oscillator (IWDT-dedicated clock) • Dedicated clock/1, dedicated clock/16, dedicated clock/32, dedicated clock/64, dedicated clock/128, dedicated clock/256 • Window function: The positions where the window starts and ends are specifiable (the window defines the timing with which refreshing is enabled and disabled). • Event linking by the ELC • Operation in the deep software standby mode 1 is possible.
Communications functions	Ethernet module (ESWM)	• One channel (one port) • Transfer at 10, 100, or 1000 Mbps • Full- and half-duplex modes • Detection of Magic Packets™*2 or output of a “Wake-On-LAN” signal (WOL) • Management frame transmission/reception with PHY-LSI compliant with the following specifications: MII (Media-Independent Interface), RMII (Reduced Media-Independent Interface), GMII (Gigabit Media-Independent Interface), and RGMII (Reduced Gigabit Media-Independent Interface) • Support for MII MAC-to-MAC connection • Two gPTP channels

Table 1.1 Outline of Specifications (7/10)

Classification	Module/Function	Description
Communications functions	USB 2.0 full-speed host/function module (USBf)	• Includes a USB device controller (UDC) and transceiver for USB 2.0 FS • One port • Compliant with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-powered mode and bus-powered mode are supported • On-the-Go (OTG) operation is possible (low-speed is not supported) • Incorporates 2 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	USB 2.0 high-speed host/function module (USBHS)	• Includes a USB device controller (UDC) and transceiver for USB 2.0 HS • One port (320-pin/224-pin/176-pin products only) • Compliant with the USB 2.0 specification • Transfer rate: High speed (480 Mbps), full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-powered mode and bus-powered mode are supported • On-the-Go (OTG) operation is possible (low-speed is not supported) • Compliant with Battery Charging Specification Revision 1.2 (BC1.2) • Incorporates 8.5 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	Serial communications interface (RSCI)	• 13 channels • Serial communications modes: Asynchronous, clock synchronous, and smart-card interface • Multi-processor function • On-chip baud rate generator allows selection of the desired bit rate • Choice of LSB-first or MSB-first transfer • Start-bit detection: Level or edge detection is selectable. • Simple I^2C • Simple SPI • 9-bit transfer mode • Bit rate modulation • Double-speed mode • Average transfer rate clock can be input from TMR timers • Event linking by the ELC • Supports the serial communications protocol, which contains the start frame and information frame • Supports the LIN format • Data can be transmitted or received in sequence by the 16-byte FIFO buffers of the transmission and reception unit • Manchester encoding is supported. • RSCI has some home bus system (HBS) functionality. • Data match detection • Adjustment of the timing of sampling of the RXD signals
	I3C bus interface (RI3C)	• Single channel (I^2C/I3C) • I^2C protocol mode Communication formats: I^2C-bus format/SMBus format Supports the multi-master Max. transfer rate: 3.4 Mbps • I3C protocol mode Communication protocols: SDR mode, Legacy I^2C message support Supports the multi-master Max. transfer rate: 12 Mbps • Event linking by the ELC
	High-speed I^2C-bus interface (RIICHS)	• Two channels (I^2C only) • I^2C-bus interface Communication formats: I^2C-bus format/SMBus format Supports the multi-master Max. transfer rate: 3.4 Mbps

Table 1.1 Outline of Specifications (8/10)

Classification	Module/Function	Description
Communications functions	CAN FD module with gateway functionality (CANFDG)	• 2 channels • Compliance with the ISO11898-1:2015 specification (standard frame and extended frame) • Gateway function • Message buffers 32 receive message buffers 16 transmit message buffers per channel Four transmit queues per channel • FIFO Eight receive FIFOs Six common FIFOs
	CAN FD module (CANFD)	• 1 channel • Compliance with the ISO11898-1:2015 specification (standard frame and extended frame) • Message buffers 16 receive message buffers Four transmit message buffers One transmit queue • FIFO Tow receive FIFOs One common FIFO
	Serial peripheral interface (RSPIA)	• 3 channels • RSPI transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) Capable of handling serial transfer as a master or slave • Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) Transmit/receive data can be swapped in byte units • Buffered structure The transmitter and receiver each have four-stage 32-bit FIFO buffers to enable continuous transmission and reception. • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC • Communications protocol: RSPIA supports the Texas Instruments Synchronous Serial Protocol (TI SSP).
	Expanded serial peripheral interface (XSPI)	• 2 units (switchable between controller and target) • Compliant with xSPI (JEDEC Standards JESD 251, JESD 251-1, and JESD 252) • 1-bit, 2-bit, 4-bit, and 8-bit protocols supported • Supports Octal SPI in the JESD 251 Profile 1.0, and HyperBus™ (HyperRAM™ and HyperFlash™) in the JESD 251 Profile 2.0. • Supports decryption on the fly (DOTF) • Maximum operating rate as a controller: 266 MT/s (133-MHz clock) • Maximum operating rate as a target: 200 MT/s (100-MHz clock)
Serial sound interface (SSIE)		• 2 channels • Full-duplex transmission (for both channel 0 and channel 1) • Various types of serial audio formatting are supported. • Master and slave operations are supported. • The bit-clock frequency is selectable from among 13 frequencies (1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, or 1/128). • Data formats with 8, 16, 18, 20, 22, 24, and 32 bits are supported. • 32-stage FIFO buffers for transmission and reception • Stopping or not stopping the SSILRCK signal on stopping of data transmission is selectable. • Clock signal from a GPTW is selectable.

Table 1.1 Outline of Specifications (9/10)

Classification	Module/Function	Description
SD/MMC host interface (SDHIc)		• 1 channel • SD bus interface One interface for SD memory and I/O cards (supporting 1- and 4-bit SD buses) Compatible with the SD, SDHC, and SDXC formats • MMC bus interface Data buses: Support for 1-, 4-, and 8-bit MMC buses • Interrupt requests: Card access interrupt, SDIO access interrupt, card detection interrupt, interrupt of SD buffer access • DMA transfer requests: SD_BUF write and SD_BUF read • Support for card detection and write protection
Arithmetic unit for trigonometric functions (TFUv2)		• Calculation of sine, cosine, arctangent, and hypotenuse Simultaneous calculation of sine and cosine Simultaneous calculation of arctangent and hypotenuse
Parallel data capture unit (PDC)		• 1 channel • Acquisition of synchronization through external 8-bit horizontal and vertical synchronization signals • Setting of the image size when clipping of the output for a one-frame image is required
Pulse-density modulation interface (PDM-IF)		• 1 channel • Connectable with an external microphone • 1-bit digital input data can be converted to 20- or 16-bit data. • Four filters: Sinc filter, high-pass filter, correction filter, and half-band decimation filter • Incorporates an internal buffer
Phase-counting interface (PCIF)		• Two units • Outputs ABZ-phase waveforms similar to incremental encoder outputs • Carrier-cycle event input from the ELC
16-bit A/D converter (ADC16H)		• Three units • Hybrid architecture that combines successive comparison and delta-sigma modulation • Up to 16-bit resolution (switchable between 16, 14, 12, and 10 bits) • High-speed conversion: 0.16 µs at the 50-MHz A/D conversion clock • Self-calibration • Extended analog functions: Self-diagnostic function, temperature sensor, internal reference voltage, VBATT voltage, D/A converter • Channel-dedicated sample-and-hold function (3 channels only in unit 0) • Supports single-ended and differential inputs
12-bit D/A converter (DAC12R)		• 2 channels • 12-bit resolution • Output voltage: 0.2 V to AVCC0 – 0.2 V (buffered output), 0 V to AVCC0 (unbuffered output) • Buffered output or unbuffered output can be selected. • Event linking by the ELC
Temperature sensor		• 1 channel • Relative precision: ± 1°C • The voltage of the temperature is converted into a digital value by the 16-bit A/D converter. • Detection of abnormally high and low temperatures
Safety	Memory-protection unit (MPU)	• Protection area: Eight areas (max.) can be specified in the range from 0000 0000h to FFFF FFFFh. • Minimum protection unit: 32 bytes • Reading from, writing to, and enabling the execution access can be specified for each area. • An access exception occurs when the detected access is not in the permitted area.
	Register write protection function	• Protects important registers from being overwritten for in case a program runs out of control.

Table 1.1 Outline of Specifications (10/10)

Classification	Module/Function	Description
Safety	CRC calculator (CRCA)	• Generation of CRC codes for 8-/32-bit data 8-bit data Selectable from the following three polynomials $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, $X^{16} + X^{12} + X^5 + 1$ 32-bit data Selectable from the following two polynomials $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$, $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable
	Main clock oscillation stop detection	• Main clock oscillation stop detection: Available
	Sub clock oscillation stop detection	• Sub clock oscillation stop detection: Available
	Clock frequency accuracy measurement circuit (CAC)	• Three units • Monitors the clock output from the main clock oscillator, sub-clock oscillator, low-, middle-, and high-speed on-chip oscillators, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, PLL2R, and PCLKB, and generates interrupts when the setting range is exceeded.
	Data operation circuit (DOCA)	• This handles the comparison, addition, subtraction, comparison in terms of which is larger or smaller, or window comparison of 32-bit values.
Security	Security function	• Protect Zone function Security state of the CPU: Secure or non-secure Access to the secure area is only possible through the CPU when it is in the secure state. • Privilege control • Device lifecycle management • Configurable range for access by the debugger at the debug authentication level • Secure key injection • Secure multi-function pin controller • Secure boot • Secure factory programming • The RSIP can be used for data confidentiality and protection against tampering with data. 256-bit hardware unique key (HUK) 128-bit unique ID OEM boot loader version Key data for use with circuits for decryption on the fly (DOTFs) Protection against SPA or DPA
	Decryption on the fly (DOTF)	Decrypt encrypted content stored in external memory in real time.
Encryption function	Renesas Secure IP (RSIP-E50D)	• Common key encryption: AES, ChaCha20-Poly1305 • Public key encryption: RSA, ECC • Hash value generation: SHA-512, SHA3-512 • TRNG (true-random number generator) : 128-bit
Operating frequency		Up to 600 MHz
Power supply voltage		VCC = VCC2 = 1.62 to 3.63 V VCC_USB = VCC_USBHS = AVCC_USBHS = 1.62 to 3.63 V AVCC0 = 1.62 to 3.63 V, 1.62 ≤ VREFH0 ≤ AVCC0 V_{BATT} = 1.62 to 3.63 V The level of each power supply voltage can independently be set. Connect the VCC_DCDC pin to VCC or VSS in accord with the power-supply mode.
Junction temperature		'4' version: –40 to +125°C
Package		320-pin LFBGA (PLBG0320GD-A) 224-pin LFBGA (PLBG0224GE-A) 176-pin LFBGA (PLBG0176GG-A) 176-pin LFQFP (PLQP0176KJ-A) 144-pin LFQFP (PLQP0144KA-B) 100-pin LFQFP (PLQP0100KP-A)
On-chip debugging system		• JTAG, SWD, and FINE interfaces

Note 1. When the realtime clock is not used, initialize the registers in the time clock according to description in section 30.6.8, Initialization Procedure When the Realtime Clock is Not to be Used.

Note 2. Magic Packet™ is a registered trademark of Advanced Micro Devices, Inc.

Table 1.2 Comparison of Functions for Different Packages (1/2)

Functions		RX74N Group					
Package		320-Pin LFBGA	224-Pin LFBGA	176-Pin LFBGA	176-Pin LFQFP	144-Pin LFQFP	100-Pin LFQFP
Operating frequency (max.)		600/300 MHz			480/300 MHz		
External bus	External bus width	32 bits/16 bits/8 bits				16 bits/8 bits	Not available
	SDRAM area controller	Available					Not available
DMA	DMA controller	Ch. 0 to 7					
	Data transfer controller	Available					
	EXDMA controller	Ch. 0 and 1					Not available
Timers	General PWM timer	Ch. 0 to 15					
	High resolution PWM	Ch. 0 to 3					
	Port output enable for GPTW	Available					
	8-bit timers	Ch. 0 to 3					
	Compare match timer	Ch. 0 to 3					
	Compare match timer W	Ch. 0 and 1					Ch. 1
	Realtime clock	Available					
	Watchdog timer	Available					
	Independent watchdog timer	Available					
Communications functions	Ethernet module	ch0 (GMII/MII/ RMII/RGMII)	ch0 (MII/RMII/RGMII)		ch0 (MII/RMII)		
	USB 2.0 full-speed host/ function module	Ch. 0					
	USB 2.0 high-speed host/ function module	Ch. 0				Not available	
	Serial communications interface	Ch. 0 to 12					Ch. 0 to 5, 7, 8, 10 to 12
	I3C bus interface	Ch. 0					
	High-speed I^2C-bus interface	Ch. 1 and 2					
	Serial peripheral interface	Ch. 0 to 2					
	CAN FD module	Ch. 0 to 2					
	Expanded serial peripheral interface	Unit 0 (1 bit, 2 bits, 4 bits, 8 bits), Unit 1 (1 bit, 2 bits, 4 bits, 8 bits)	Unit 0 (1 bit, 2 bits, 4 bits, 8 bits), Unit 1 (1 bit, 2 bits, 4 bits)				Unit 0 (1 bit, 2 bits, 4 bits), Unit 1 (1 bit, 2 bits, 4 bits)
	Serial sound interface	Ch. 0					
	SD/MMC host interface	Ch. 0					
	Pulse-density modulation interface	Ch. 0					
	Phase-counting interface	Ch. 0 and 1					
	Parallel data capture unit	Available					Not available

Table 1.2 Comparison of Functions for Different Packages (2/2)

Functions		RX74N Group					
Package		320-Pin LFBGA	224-Pin LFBGA	176-Pin LFBGA	176-Pin LFQFP	144-Pin LFQFP	100-Pin LFQFP
16-bit A/D converter	Unit 0	AN000 to 007 (8 channels)				AN000 to 005 (6 channels)	AN000, 002, 004 (3 channels)
	Unit 1	AN008 to 035 (28 channels)	AN014 to 035 (22 channels)			AN014 to 033 (20 channels)	AN014, 017 to 031 (16 channels)
	Unit 2	AN008 to 035 (28 channels)	AN014 to 035 (22 channels)			AN014 to 033 (20 channels)	AN014, 017 to 031 (16 channels)
12-bit D/A converter		Ch. 0 and 1					Ch. 0
Temperature sensor		Available					
Arithmetic unit for trigonometric functions		Available					
CRC calculator		Available					
Data operation circuit		Available					
Clock frequency accuracy measurement circuit		Available					
Renesas Secure IP		Available					
Event link controller		Available					

1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no.

Table 1.3 List of Products (1/2)

Group	Part No.	Package	Code MRAM Capacity	RAM Capacity	Operating Frequency (Max.)	CANFD	Junction temperature (°C)
RX74N	R5K574NCC4FC	PLQP0176KJ-A	1 Mbyte	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574NCD4FC	PLQP0176KJ-A	1 Mbyte	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574N6C4FC	PLQP0176KJ-A	512 Kbytes	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574N6D4FC	PLQP0176KJ-A	512 Kbytes	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574NCC4FB	PLQP0144KA-B	1 Mbyte	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574NCD4FB	PLQP0144KA-B	1 Mbyte	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574N6C4FB	PLQP0144KA-B	512 Kbytes	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574N6D4FB	PLQP0144KA-B	512 Kbytes	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574NCC4FP	PLQP0100KP-A	1 Mbyte	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574NCD4FP	PLQP0100KP-A	1 Mbyte	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574N6C4FP	PLQP0100KP-A	512 Kbytes	1.5 Mbytes	480 MHz	Available*1	–40 to +125
	R5K574N6D4FP	PLQP0100KP-A	512 Kbytes	1.5 Mbytes	480 MHz	Available	–40 to +125
	R5K574NCA4AE	PLBG0320GD-A	1 Mbyte	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574NCB4AE	PLBG0320GD-A	1 Mbyte	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574N6A4AE	PLBG0320GD-A	512 Kbytes	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574N6B4AE	PLBG0320GD-A	512 Kbytes	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574NCA4BD	PLBG0224GE-A	1 Mbyte	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574NCB4BD	PLBG0224GE-A	1 Mbyte	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574N6A4BD	PLBG0224GE-A	512 Kbytes	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574N6B4BD	PLBG0224GE-A	512 Kbytes	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574NCA4BG	PLBG0176GG-A	1 Mbyte	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574NCB4BG	PLBG0176GG-A	1 Mbyte	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574N6A4BG	PLBG0176GG-A	512 Kbytes	1.5 Mbytes	600 MHz	Available*1	–40 to +125
	R5K574N6B4BG	PLBG0176GG-A	512 Kbytes	1.5 Mbytes	600 MHz	Available	–40 to +125
	R5K574NCE4FC	PLQP0176KJ-A	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4FC	PLQP0176KJ-A	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4FC	PLQP0176KJ-A	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4FC	PLQP0176KJ-A	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574NCE4FB	PLQP0144KA-B	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4FB	PLQP0144KA-B	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4FB	PLQP0144KA-B	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4FB	PLQP0144KA-B	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574NCE4FP	PLQP0100KP-A	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4FP	PLQP0100KP-A	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4FP	PLQP0100KP-A	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4FP	PLQP0100KP-A	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574NCE4AE	PLBG0320GD-A	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4AE	PLBG0320GD-A	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4AE	PLBG0320GD-A	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4AE	PLBG0320GD-A	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125

Table 1.3 List of Products (2/2)

Group	Part No.	Package	Code MRAM Capacity	RAM Capacity	Operating Frequency (Max.)	CANFD	Junction temperature (°C)
RX74N	R5K574NCE4BD	PLBG0224GE-A	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4BD	PLBG0224GE-A	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4BD	PLBG0224GE-A	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4BD	PLBG0224GE-A	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574NCE4BG	PLBG0176GG-A	1 Mbyte	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574NCF4BG	PLBG0176GG-A	1 Mbyte	1.5 Mbytes	300 MHz	Available	–40 to +125
	R5K574N6E4BG	PLBG0176GG-A	512 Kbytes	1.5 Mbytes	300 MHz	Available*1	–40 to +125
	R5K574N6F4BG	PLBG0176GG-A	512 Kbytes	1.5 Mbytes	300 MHz	Available	–40 to +125

Note 1. Products with this part number support only CAN 2.0 protocol.

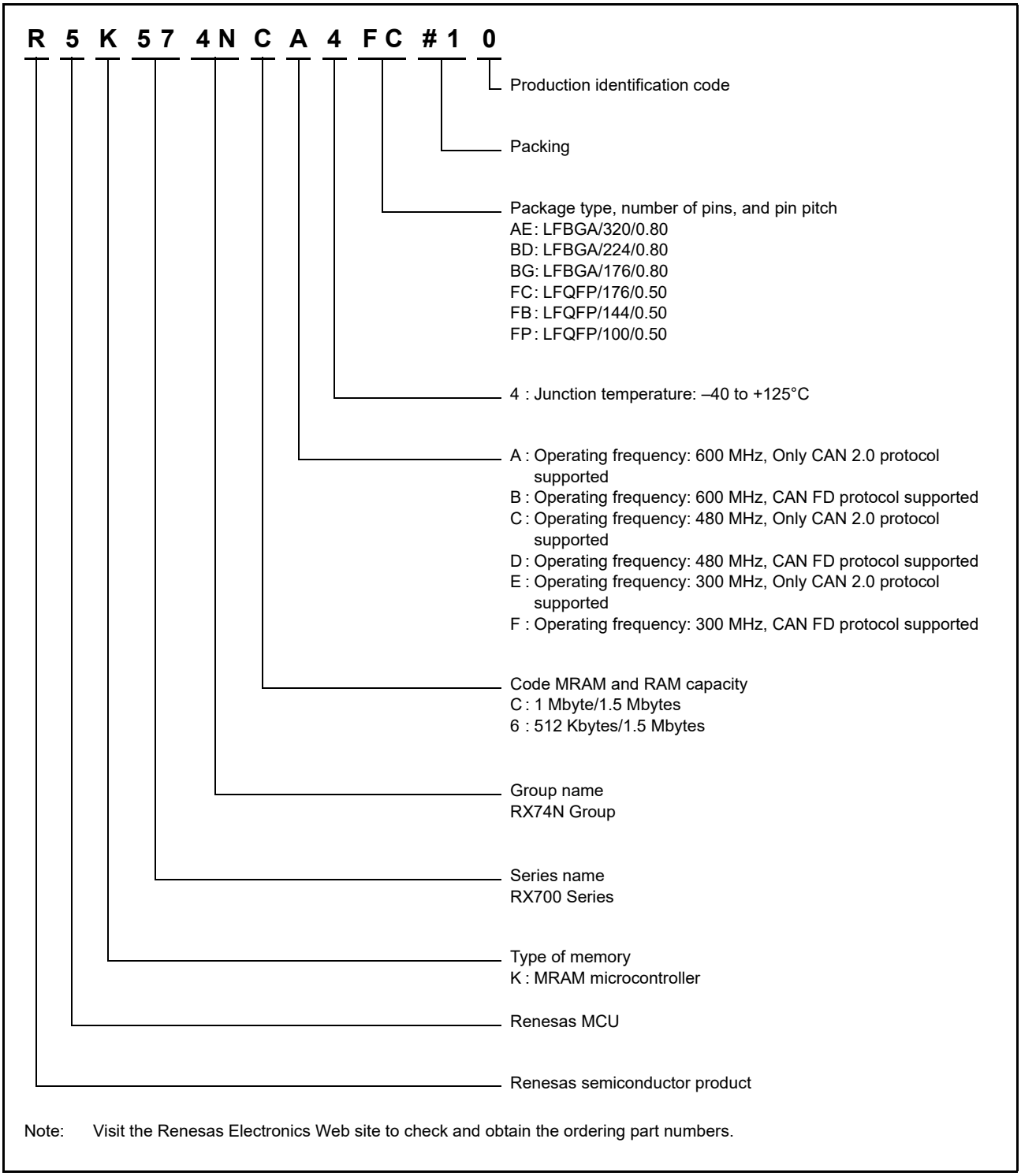
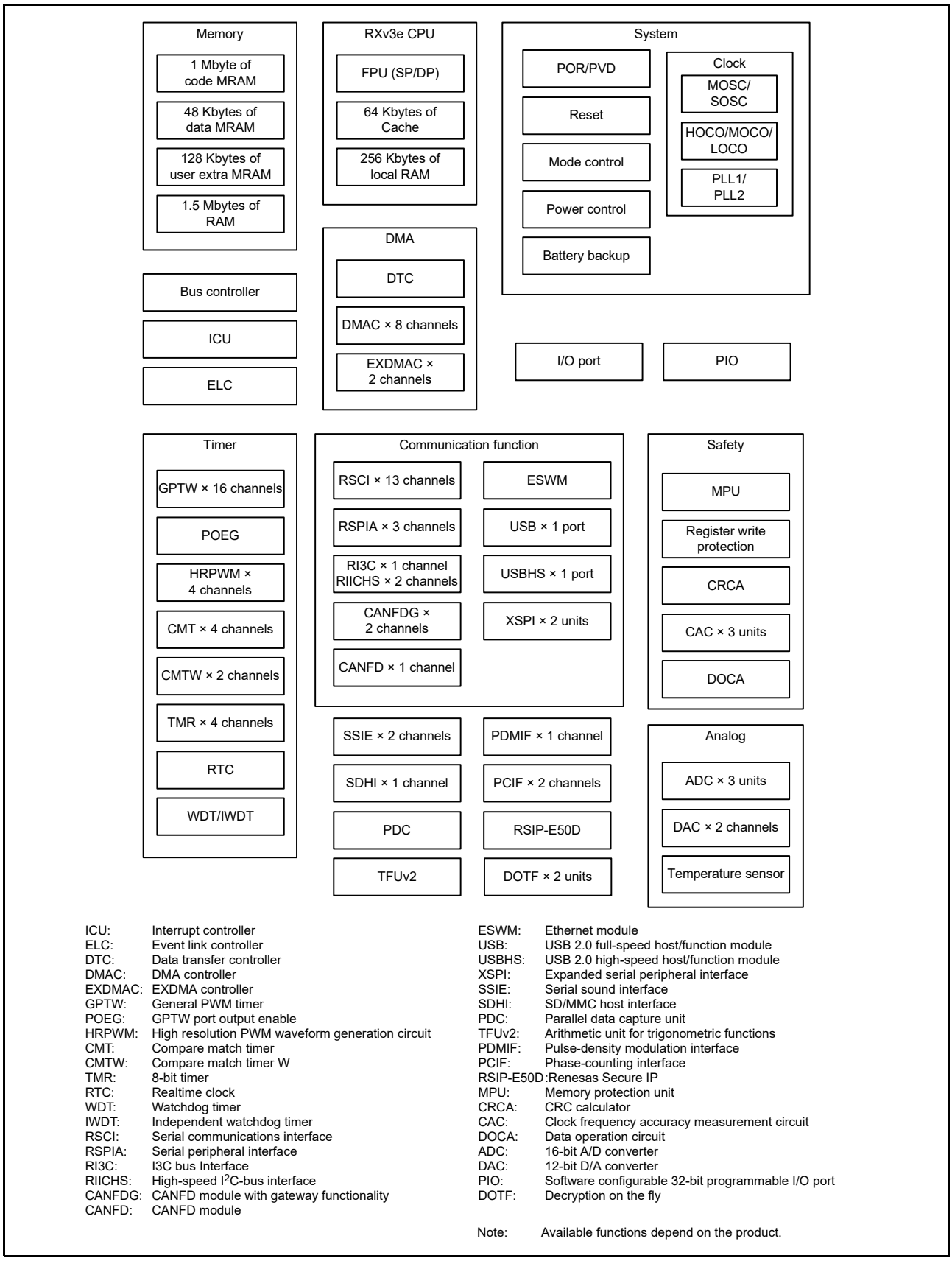


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICU: Interrupt controller
ELC: Event link controller
DTC: Data transfer controller
DMAC: DMA controller
EXDMAC: EXDMA controller
GPTW: General PWM timer
POEG: GPTW port output enable
HRPWM: High resolution PWM waveform generation circuit
CMT: Compare match timer
CMTW: Compare match timer W
TMR: 8-bit timer
RTC: Realtime clock
WDT: Watchdog timer
IWDT: Independent watchdog timer
RSCI: Serial communications interface
RSPIA: Serial peripheral interface
RI3C: I3C bus Interface
RIICHS: High-speed I^2C-bus interface
CANFDG: CANFD module with gateway functionality
CANFD: CANFD module

ESWM: Ethernet module
USB: USB 2.0 full-speed host/function module
USBHS: USB 2.0 high-speed host/function module
XSPI: Expanded serial peripheral interface
SSIE: Serial sound interface
SDHI: SD/MMC host interface
PDC: Parallel data capture unit
TFUv2: Arithmetic unit for trigonometric functions
PDMIF: Pulse-density modulation interface
PCIF: Phase-counting interface
RSIP-E50D: Renesas Secure IP
MPU: Memory protection unit
CRCA: CRC calculator
CAC: Clock frequency accuracy measurement circuit
DOCA: Data operation circuit
ADC: 16-bit A/D converter
DAC: 12-bit D/A converter
PIO: Software configurable 32-bit programmable I/O port
DOTF: Decryption on the fly

Note: Available functions depend on the product.

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/7)

Classifications	Pin Name	I/O	Description
Digital power supply	VCC	Input	Power supply pin. Connect this pin to the system power supply. Connect the pin to VSS via a 0.1-μF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VCC2	Input	Power supply pin. Connect this pin to the system power supply. Connect the pin to VSS via a 0.1-μF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VCL	Input	The pin setting depends on the power-supply mode. For details, see section 66, Internal Voltage Regulator.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBATT	Input	Backup power pin
Switching regulator	VCC_DCDC	Input	Power supply pin of the switching regulator
	VLO	Output	Output pin of the switching regulator Connect this pin to an external inductor and capacitor.
	VSS_DCDC	Input	Ground pin of the switching regulator
Clock	XTAL	Output	Input/output pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	EBCLK	Output	Outputs the external bus clock for external devices.
	SDCLK	Output	Outputs the SDRAM-dedicated clock.
	XCOUT	Output	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
	EXCIN	Input	External sub-clock input pin
	CLKOUT	Output	Clock output pin.
Clock frequency accuracy measurement	CACREF	Input	Reference clock input pin for the clock frequency accuracy measurement circuit
Operating mode control	MD	Input	Input pin for setting the operating mode. The signal level on this pin must not be changed during operation.
System control	RES#/RSTOUT#	I/O	Reset signal input pin. This MCU enters the reset state when this signal goes low.
On-chip emulator	FINED	I/O	FINE interface pin
	TMS	Input	On-chip emulator or boundary scan pins.
	TDI	Input	
	TCK	Input	
	TDO	Output	
	SWDIO	I/O	Serial wire debugging data input/output pin
	SWCLK	Input	Serial wire clock pin
Address bus	A0 to A23	Output	Output pins for the address
Data bus	D0 to D31	I/O	Input and output pins for the bidirectional data bus
Multiplexed bus	A0/D0 to A15/D15	I/O	Address/data multiplexed bus

Table 1.4 Pin Functions (2/7)

Classifications	Pin Name	I/O	Description
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in 1-write strobe mode
	WR0# to WR3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode
	BC0# to BC3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode
	ALE	Output	Address latch signal when address/data multiplexed bus is selected
	WAIT#	Input	Input pin for wait request signals in access to the external space
	CS0# to CS7#	Output	Select signals for CS areas
	BREQ#	Input	External bus request signal from an external device
	BACK#	Output	External bus acknowledge signal from the MCU
SDRAM interface	CKE	Output	SDRAM clock enable signal
	SDCS#	Output	SDRAM chip select signal
	RAS#	Output	SDRAM row address strobe signal
	CAS#	Output	SDRAM column address strobe signal
	WE#	Output	SDRAM write enable pin
	DQM0 to DQM3	Output	SDRAM I/O data mask enable signals
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ31, IRQ0-DS to IRQ31-DS	Input	Maskable interrupt request pins
General PWM timer	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	Input pins for the external trigger signals
	GTIOC0A to GTIOC15A, GTIOC0B to GTIOC15B	I/O	Input capture input/output compare output/PWM output pins
	GTCPPO0 to GTCPPO15	Output	PWM synchronous output
	GTIU, GTIV, GTIW	Input	Hall sensor input pins
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)
	GTADSM0, GTADSM1	Output	Output pins for monitoring A/D conversion start requests.
GPTW port output enable	POEGA, POEGB, POEGC, POEGD	Input	Input pins for request signals to place GPTW outputs in the high-impedance state
8-bit timer	TMO0 to TMO3	Output	Compare match output pins
	TMCI0 to TMCI3	Input	Input pins for external clocks to be input to the counter
	TMRI0 to TMRI3	Input	Input pins for the counter reset
Compare match timer W	TIC0 to TIC3	Input	Input pins for CMTW
	TOC0 to TOC3	Output	Output pins for CMTW

Table 1.4 Pin Functions (3/7)

Classifications	Pin Name	I/O	Description
Serial communications interface	• Asynchronous mode/clock synchronous mode		
	SCK000 to SCK012	I/O	Input/output pins for the clock
	RXD000 to RXD012	Input	Input pins for received data
	TXD000 to TXD012	Output	Output pins for transmitted data
	CTS000# to CTS012#	Input	Input pins for controlling the start of transmission and reception
	RTS000# to RTS012#	Output	Output pins for controlling the start of transmission and reception
	DE000 to DE012	Output	Output pins for controlling the RS-485 driver
	• Simple I^2C mode		
	SSCL000 to SSCL012	I/O	Input/output pins for the I^2C clock
	SSDA000 to SSDA012	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK000 to SCK012	I/O	Input/output pins for the clock
	SMISO000 to SMISO012	I/O	Input/output pins for slave transmission of data
	SMOSI000 to SMOSI012	I/O	Input/output pins for master transmission of data
	SS000# to SS012#	Input	Chip-select input pins
	• HBS support mode		
	RXD000 to RXD012	Input	Input pins for received data
	TXDA000 to TXDA012	Output	Output pins for transmitted data
	TXDB000 to TXDB012		
I3C bus interface	SCL00	I/O	Input/output pins for clocks
	SDA00	I/O	Input/output pins for data
High-speed I^2C-bus interface	SCLHS1, SCLHS2	I/O	Input/output pins for clocks
	SDAHS1, SDAHS2	I/O	Input/output pins for data
Ethernet module	• RMII mode		
	RMII0_REF50CK	Input	50-MHz reference clocks. These pins input reference signals for transmission/reception timing in RMII mode.
	RMII0_CRS_DV	Input	These pins indicate that there are carrier detection signals and valid receive data on RMIIn_RXD1 and RMIIn_RXD0 in RMII mode.
	RMII0_TXD0, RMII0_TXD1	Output	2-bit transmit data in RMII mode
	RMII0_RXD0, RMII0_RXD1	Input	2-bit receive data in RMII mode
	RMII0_TX_EN	Output	Output pin for data transmit enable signal in RMII mode
	RMII0_RX_ER	Input	These pins indicate an error has occurred during reception of data in RMII mode.
	• RGMII mode		
	RGMII0_TXC	Output	Transmission clock signals in RGMII mode
	RGMII0_RXC	Input	Reception clock signals in RGMII mode
	RGMII0_TX_CTL	Output	Control of transmission in RGMII mode
	RGMII0_TXD0 to RGMII0_TXD3	Output	4-bit transmission data in RGMII mode
	RGMII0_RX_CTL	Input	Control of reception in RGMII mode

Table 1.4 Pin Functions (4/7)

Classifications	Pin Name	I/O	Description
Ethernet module	RGMII0_RXD0 to RGMII0_RXD3	Input	4-bit reception data in RGMII mode
	• GMII/MII mode		
	ET0_GTX_CLK	Output	1000-Mbps transmission clock pin
	ET0_CRS	Input	Carrier detection/data reception enable pins
	ET0_RX_DV	Input	These pins indicate that there are valid receive data.
	ET0_TXD0 to ET0_TXD7	Output	8 bits of GMII transmit data/4 bits of MII transmit data
	ET0_RXD0 to ET0_RXD7	Input	8 bits of GMII receive data/4 bits of MII receive data
	ET0_TX_EN	Output	Transmit enable pins.
	ET0_TX_ER	Output	Transmit error pins. These pins function as signals notifying the PHY-LSI of an error during transmission.
	ET0_RX_ER	Input	Receive error pins. These pins function as signals to recognize an error during reception.
	ET0_TX_CLK	I/O	100-Mbps/10-Mbps transmission clock pins (input for MII, output for MAC-to-MAC connection)
	ET0_RX_CLK	Input	Receive clock pins.
	ET0_COL	Input	Input collision detection signals.
	• Common		
	ET0_LINKSTA	Input	Input link status from the PHY-LSI.
	ET0_INT	Input	Input for an interrupt signal from PHY-LSI
	ET0_WOL	Output	Signals indicating receipt of Magic Packet.
	ET0_MDC	Output	Output reference clock signals for information transfer via ETn_MDIO.
	ET0_MDIO	I/O	Input or output bidirectional signals for exchange of management information between this MCU and the PHY-LSI.
	GPTP_CAPTURE0, GPTP_CAPTURE1	Input	Media clock capture input
	GPTP_MATCH0, GPTP_MATCH1	Output	Media clock recovery output
	GPTP0_PPS	Output	PPS signals
	GPTP_PTPOUT0, GPTP_PTPOUT1	Output	PTP pulse generation signals
	ETHPHYCLK	Output	Clock output pin for PHY clock input
USB 2.0 host/function module	VCC_USB, VCC_USBHS	Input	Power supply pins
	VSS_USB, VSS_USBHS	Input	Ground pins
	AVCC_USBHS	Input	Analog voltage supply pin for USBHS
	AVSS_USBHS	Input	Analog ground pin for USBHS
	USBHS_RREF	I/O	Reference voltage supply pin for USBHS
	USB0_DP, USBHS_DP	I/O	Input or output USB transceiver D+ data.
	USB0_DM, USBHS_DM	I/O	Input or output USB transceiver D- data.
	USB0_EXICEN, USBHS_EXICEN	Output	Connect to the OTG power IC.
	USB0_ID, USBHS_ID	Input	Connect to the OTG power IC.
	USB0_VBUSEN, USBHS_VBUSEN	Output	USB VBUS power enable pins

Table 1.4 Pin Functions (5/7)

Classifications	Pin Name	I/O	Description
USB 2.0 host/function module	USB0_OVRCURA/ USB0_OVRCURB, USBHS_OVRCURA/ USBHS_OVRCURB, USB0_OVRCURA-DS/ USB0_OVRCURB-DS, USBHS_OVRCURA-DS/ USBHS_OVRCURB-DS	Input	USB overcurrent pins
	USB0_VBUS, USBHS_VBUS, USB0_VBUS-DS, USBHS_VBUS-DS	Input	USB cable connection/disconnection detection input pin
CANFD module	CRX0, CRX1, CRX2	Input	CAN input pins
	CTX0, CTX1, CTX2	Output	CAN output pins
Serial peripheral interface	RSPCK0 to RSPCK2	I/O	Clock input/output pins
	MOSI0 to MOSI2	I/O	Input or output data output from the master
	MISO0 to MISO2	I/O	Input or output data output from the slave
	SSL00, SSL10, SSL20	I/O	Input or output pins for slave selection
	SSL01 to SSL03, SSL11 to SSL13, SSL21 to SSL23	Output	Output pins for slave selection
Expanded serial peripheral interface	XSPI0_CK, XSPI1_CK	I/O	Clock input/output pins (output in the controller mode; input in the target mode)
	XSPI0_CKN#, XSPI1_CKN#	Output	Inverted clock output (output in the controller mode; N/A in the target mode)
	XSPI0_CS0#, XSPI0_CS1#, XSPI1_CS0#, XSPI1_CS1#	I/O	Chip select signals (output in the controller mode; input in the target mode)
	XSPI0_DS, XSPI1_DS	I/O	Read data strobe/write data mask signal
	XSPI0_IO0 to XSPI0_IO7, XSPI1_IO0 to XSPI1_IO7	I/O	Data input/output pins
	XSPI0_RESET0#, XSPI0_RESET1#, XSPI1_RESET0#, XSPI1_RESET1#	I/O	Signals from the controller device to reset the target device (output in the controller mode; input in the target mode)
	XSPI0_ECS0#, XSPI0_ECS1#, XSPI1_ECS0#, XSPI1_ECS1#	Input	Signals that indicate the state of error correction in the target device
	XSPI0_INT0#, XSPI1_INT0#	I/O	Interrupt signals from the target device (input in the controller mode; output in the target mode)
	XSPI0_INT1#, XSPI1_INT1#	Input	Interrupt signals from the target device (input in the controller mode; N/A in the target mode)
	XSPI0_RSTO0#, XSPI1_RSTO0#	I/O	Signals from the target device to reset the controller device (input in the controller mode; output in the target mode)
	XSPI0_RSTO1#, XSPI1_RSTO1#	Input	Signals from the target device to reset the controller device (input in the controller mode; N/A in the target mode)
	XSPI0_WP0#, XSPI0_WP1#, XSPI1_WP0#, XSPI1_WP1#	Output	Signals to control protection of the target device against writing by the controller device
Serial sound interface	SSIBCK0, SSIBCK1	I/O	SSIE serial bit-clock pins
	SSILRCK0, SSILRCK1	I/O	LR clock
	SSITXD0, SSITXD1	Output	Serial data output pins
	SSIRXD0, SSIRXD1	Input	Serial data input pins
	AUDIO_CLK	Input	External clock pin for audio (input for an oversampling clock)

Table 1.4 Pin Functions (6/7)

Classifications	Pin Name	I/O	Description
SD/MMC host interface	SD0_CLK	Output	SD clock output pin
	SD0_CMD	I/O	SD command output or response input signal pin
	SD0_DAT0 to SD0_DAT7	I/O	SD/MMC data bus pins
	SD0_CD	Input	SD card detection pin
	SD0_WP	Input	SD write-protect signal
Parallel data capture unit	PIXCLK	Input	Image transfer clock pin
	VSYNC	Input	Vertical synchronization signal pin
	HSYNC	Input	Horizontal synchronization signal pin
	PIXD0 to PIXD7	Input	8-bit image data pins
	PCKO	Output	Output pin for dot clock
Pulse-density modulation interface	PDMCLK0	Output	Output pin for the clock
	PDMDAT0	Input	Input pin for data
Phase-counting interface	PCINA0, PCINA1	Input	A-phase input pins
	PCINB0, PCINB1	Input	B-phase input pins
	PCINZ0, PCINZ1	Input	Z-phase input pins
	PCOUTA0, PCOUTA1	Output	A-phase output pins
	PCOUTB0, PCOUTB1	Output	B-phase output pins
	PCOUTZ0, PCOUTZ1	Output	Z-phase output pins
Realtime clock	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
Battery backup	TAMPI0 to TAMPI2	Input	Input pins for detecting tampering
16-bit A/D converter	AN000 to AN035	Input	Input pins for the analog signals to be processed by the A/D converter
	ADTRG0# to ADTRG2#	Input	Input pins for the external trigger signals that start the A/D conversion
	ADST0 to ADST2	Output	Scan period monitor output
	ADCNV0 to ADCNV2	Output	Successive approximation period monitor output
	ADSYNC	Output	Synchronous A/D conversion start signal monitor output
12-bit D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Analog power supply	AVCC0	Input	Analog power supply pin for the 16-bit A/D converter, and analog power supply and reference voltage pin for the D/A converter.
	AVSS0	Input	Analog ground pin for the 16-bit A/D converter, and analog ground and reference ground pin for the D/A converter
	VREFH0	Input	Analog reference voltage supply pin for the 16-bit A/D converter
	VREFL0	Input	Analog reference ground pin for the 16-bit A/D converter
	VREFH00 to VREFH02*1	Input	Analog reference voltage supply pin for the 16-bit A/D converter
	VREFL00 to VREFL02*1	Input	Analog reference ground pin for the 16-bit A/D converter
Software Configurable 32-Bit Programmable I/O Port	PIO_000 to PIO_063	I/O	Pins composing the software configured general-purpose I/O port

Table 1.4 Pin Functions (7/7)

Classifications	Pin Name	I/O	Description
I/O ports	P000 to P015	I/O	16-bit input/output pins
	P100 to P115	I/O	16-bit input/output pins
	P200 to P215	I/O	16-bit input/output pins
	P300 to P312, P315	I/O	14-bit input/output pins (P305, P308, P309: input pin)
	P400 to P415	I/O	16-bit input/output pins
	P500 to P515	I/O	16-bit input/output pins
	P600 to P607	I/O	8-bit input/output pins
	P700 to P715	I/O	16-bit input/output pins
	P800 to P807	I/O	8-bit input/output pins
	P900 to P907	I/O	8-bit input/output pins
	PA00 to PA07	I/O	8-bit input/output pins
	PB00 to PB07	I/O	8-bit input/output pins
	PC00 to PC07	I/O	8-bit input/output pins
	PD00 to PD07	I/O	8-bit input/output pins
	PE00 to PE07	I/O	8-bit input/output pins
	PF00 to PF09	I/O	10-bit input/output pins
	PG00 to PG15	I/O	16-bit input/output pins

Note: Note the following points regarding pin names. For details, refer to section 1.6, List of Pin and Pin Functions.
- Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.
- When a pin function has “-DS” appended to its name, it can also be used as a trigger for release from deep software standby mode.

Note 1. This pin is only present in 320-pin products.

1.5 Pin Assignments

1.5.1 320-Pin LFBGA

RX74N Group
PLBG0320GD-A (320-pin LFBGA)
(Upper Perspective View)

INDEX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	VSS	P514	P515	PC01	PC03	PC04	PC07	P801	P800	VLO	VSS_DCDC	VSS_DCDC	VCC_DCDC	P504	P103	P104	P106	P200	P300/TMS	VSS	A
B	PG03	P512	P513	PC00	PC02	VSS	PC05	PC06	P802	VLO	VSS_DCDC	VSS_DCDC	VCC_DCDC	P505	P102	VSS	P107	P201	P301/TCK	P302/TDO	B
C	PG02	PG01	PB01	PB03	PB04	PB06	PB07	PB05	P803	VLO	VLO	VCC_DCDC	VCC_DCDC	P506	P105	P805	P806	P807	P304	P303/TDI	C
D	PA06	PG00	P511															P108	P109	P110	D
E	PA05	PA07	P510		PG05	PB02	PB00	VCC	VCC	P500	P502	P100	VCC	VCC	P507	P804		P111	P112	P113	E
F	PA03	PA04	P509		PG07	PG06	PG04	VSS	VSS	P501	P503	P101	VSS	VSS	PF02	P115		P114	VSS	VCC	F
G	PA02	PA01	P508		VCC	VSS									PF03	PF04		VSS_USB	PF00/USB0_DP	PF01/USB0_DM	G
H	PA00	PG13	PG15		VCC	VSS		VSS	VCL	VSS	VSS	VCL	VSS		P204	P203		PF05	VCC_USB	VCC_USB	H
J	PG11	PG14	PG12		VCC	VSS		VSS	VCL	VSS	VSS	VSS	VCL		P205	P202		AVSS_USBHS	USBHS_DP	USBHS_DM	J
K	PG09	PG10	P705		P704	P702		VCL	VSS	VCL	VCL	VCC	VSS		P207	P206		USBHS_RREF	VSS_USBHS	VCC_USBHS	K
L	PG08	VSS	P707		P706	P703		VSS	VSS	VCL	VCL	VSS	VCC		P310	P305/NMI		AVCC_USBHS	VCC	VCC	L
M	P701	P715	P714		VCC	VSS		VCL	VCL	VSS	VSS	VSS	VCL		P312	P311		VSS	P306/EXTAL	P307/XTAL	M
N	P700	P712	P713		VCC	VSS		VSS	VCL	VSS	VSS	VCL	VSS		P209	P208		P315/MD	VSS	VSS	N
P	PE06	PE07	PF09		VCC	VSS									P211	P210		VSS	P309/XCIN/EXCIN	P308/XCOUT	P
R	PE03	PE05	PF08		P607	P606	VSS	VSS	VSS	P906	P405	P404	VSS	VSS	P215	P214		RES#	VCC	VBATT/VCC	R
T	PE01	PE04	PF07		P604	P602	VCC2	VCC2	VCC2	P907	P406	P407	VCC	VCC	P013	P012		P213	P212	P000	T
U	PE00	PE02	PF06															P008	P009	P001	U
V	P709	P711	P605	P603	P600	P601	P905	P904	AVCC0	AVSS0	VREFL00	VREFH00	P412	P413	P414	P415	P014	P011	P010	P002	V
W	P708	P710	PD05	PD03	VSS	PD02	P902	P900	AVCC0	AVSS0	VREFL01	VREFH01	P403	P401	P411	P409	VSS	P006	P004	P003	W
Y	VSS	PD07	PD06	PD04	PD01	PD00	P903	P901	AVCC0	AVSS0	VREFL02	VREFH02	P402	P400	P410	P408	P015	P007	P005	VSS	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.5, List of Pin and Pin Functions (320-Pin LFBGA).

Figure 1.3 Pin Assignment (320-Pin LFBGA)

1.5.2 224-Pin LFBGA

RX74N Group
PLBG0224GE-A (224-pin LFBGA)
(Upper Perspective View)

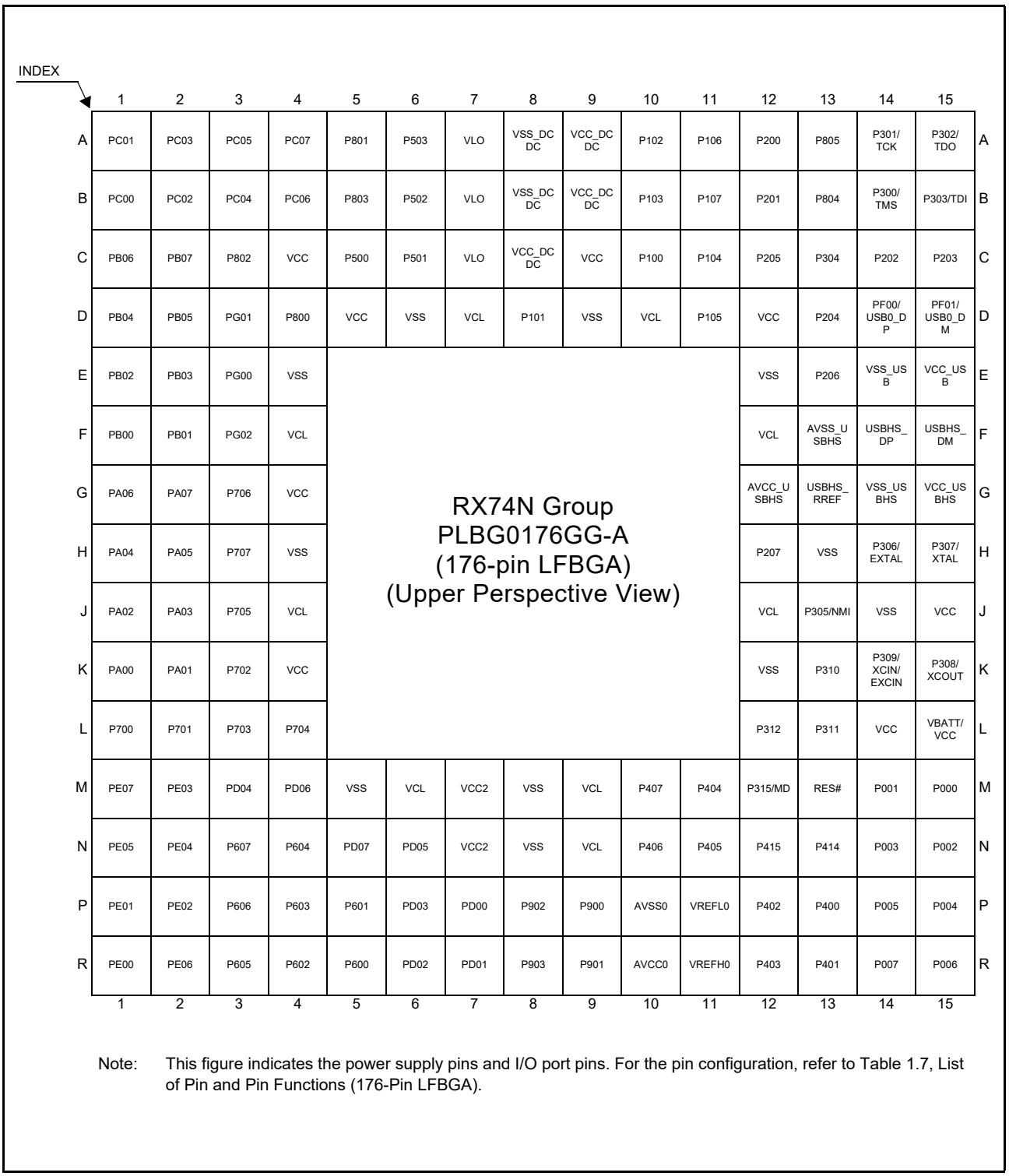
INDEX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	VSS	PC03	PC07	P801	P802	VLO	VSS_DC DC	VCC_DC DC	P103	P104	P106	P200	P301/ TCK	P302/ TDO	VSS	A
B	PC04	PC01	PC05	PC06	P803	VLO	VSS_DC DC	VCC_DC DC	P102	P105	P107	P201	P300/ TMS	P303/TDI	P304	B
C	PC00	PC02	PB06	PB05	P800	VLO	P100	VCC_DC DC	P804	P805	P114	P111	P108	P110	P113	C
D	PG00	PG02	PB04	PB07	P502	P500	P101	P504	P506	P806	P109	P115	P112	VSS	VCC	D
E	PG01	PG03	PB02	PB03	VCC	P501	P503	P505	P507	P807	PF02	PF04	PF03	PF00/ USB0_D P	PF01/ USB0_D M	E
F	PA05	PA07	PB01	PB00	VSS	VCL	VSS	VCC	VSS	VSS	PF05	P202	P203	VCC_US B	VSS_US B	F
G	PA04	PA06	PG05	PG07	PF09	VSS	VCL	VSS	VCL	VCL	P205	P204	VCC_US BHS	USBHS_ DP	USBHS_ DM	G
H	PA02	PA03	PG06	PG04	PF08	VCC	VSS	VCL	VSS	VCL	P206	AVSS_U SBHS	USBHS_ RREF	VSS_US BHS	VCC	H
J	PA00	PA01	P703	PF06	PF07	VSS	VCL	VSS	VCL	VSS	P207	P305/NMI	AVCC_U SBHS	P307/ XTAL	P306/ EXTAL	J
K	P707	P706	P702	P605	VSS	VCC	VSS	VCC2	VCL	VSS	P310	P311	P315/MD	VSS	VSS	K
L	P705	P704	P604	P606	VCC2	P904	P906	P407	P406	P215	P209	P312	VCC	P309/ XCIN/ EXCIN	P308/ XCOUT	L
M	P700	P701	P607	P603	P907	P905	P902	P900	P405	P415	P212	P214	RES#	VSS	VBATT/ VCC	M
N	PE07	PE04	PE06	P602	P601	P600	P903	P901	P404	P414	P210	P213	P208	P211	P000	N
P	PE05	PE03	PE02	PD07	PD05	PD03	PD02	AVSS0	VREFL0	P403	P401	P006	P004	P003	P001	P
R	VSS	PE01	PE00	PD06	PD04	PD00	PD01	AVCC0	VREFH0	P402	P400	P007	P005	P002	NC	R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.6, List of Pin and Pin Functions (224-Pin LFBGA).

Figure 1.4 Pin Assignment (224-Pin LFBGA)

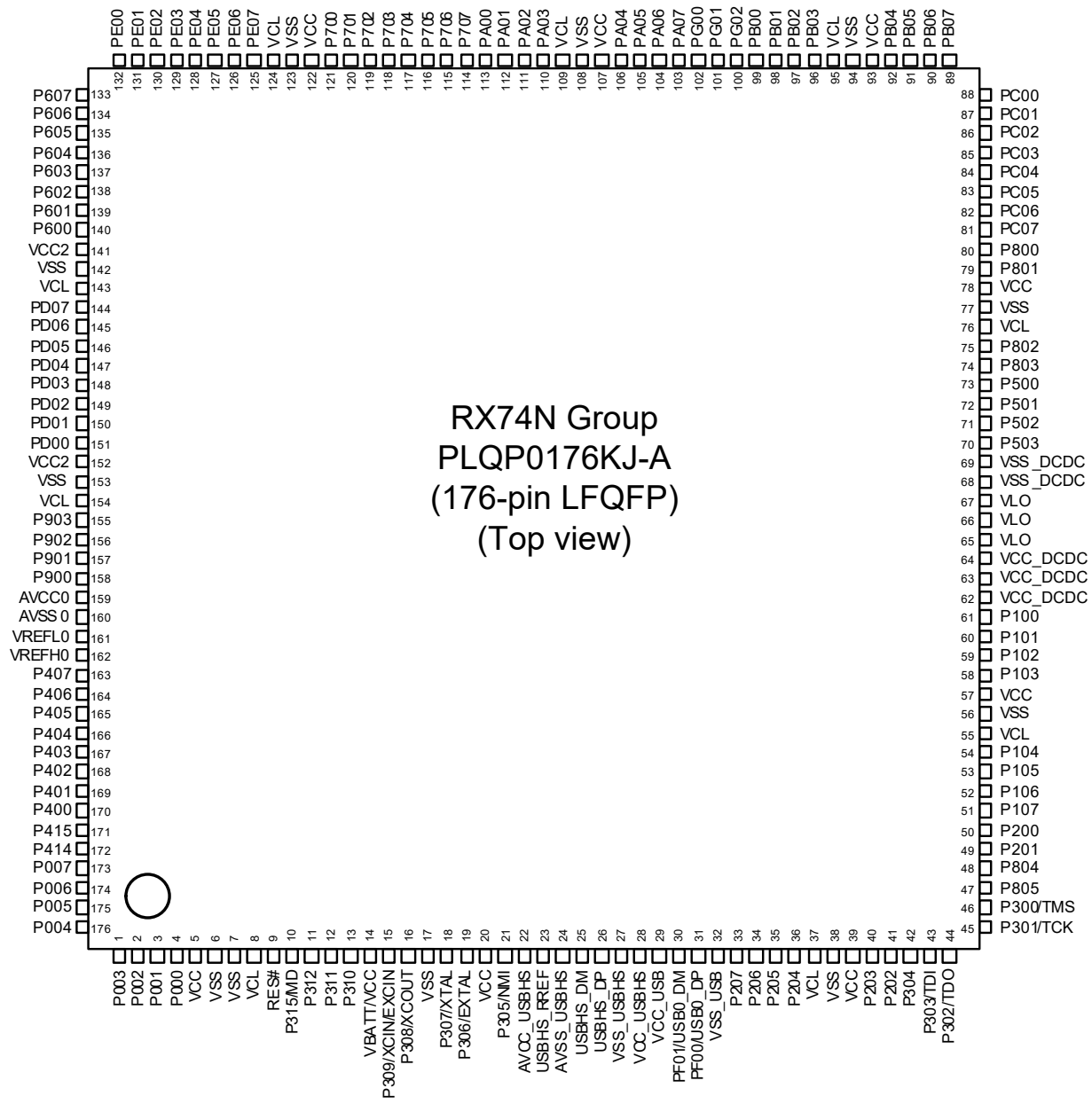
1.5.3 176-Pin LFBGA



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.7, List of Pin and Pin Functions (176-Pin LFBGA).

Figure 1.5 Pin Assignment (176-Pin LFBGA)

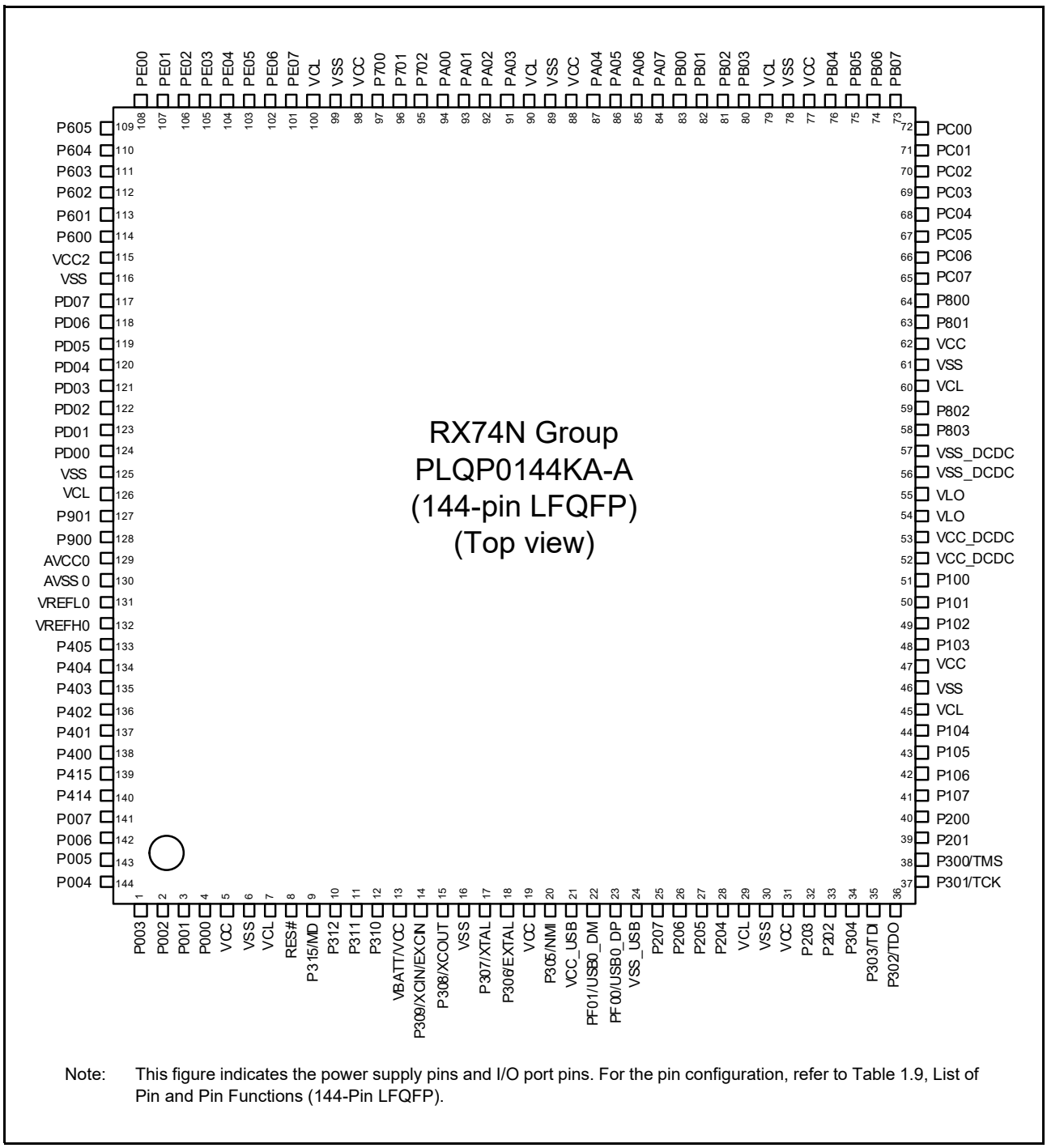
1.5.4 176-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.8, List of Pin and Pin Functions (176-Pin LFQFP).

Figure 1.6 Pin Assignment (176-Pin LFQFP)

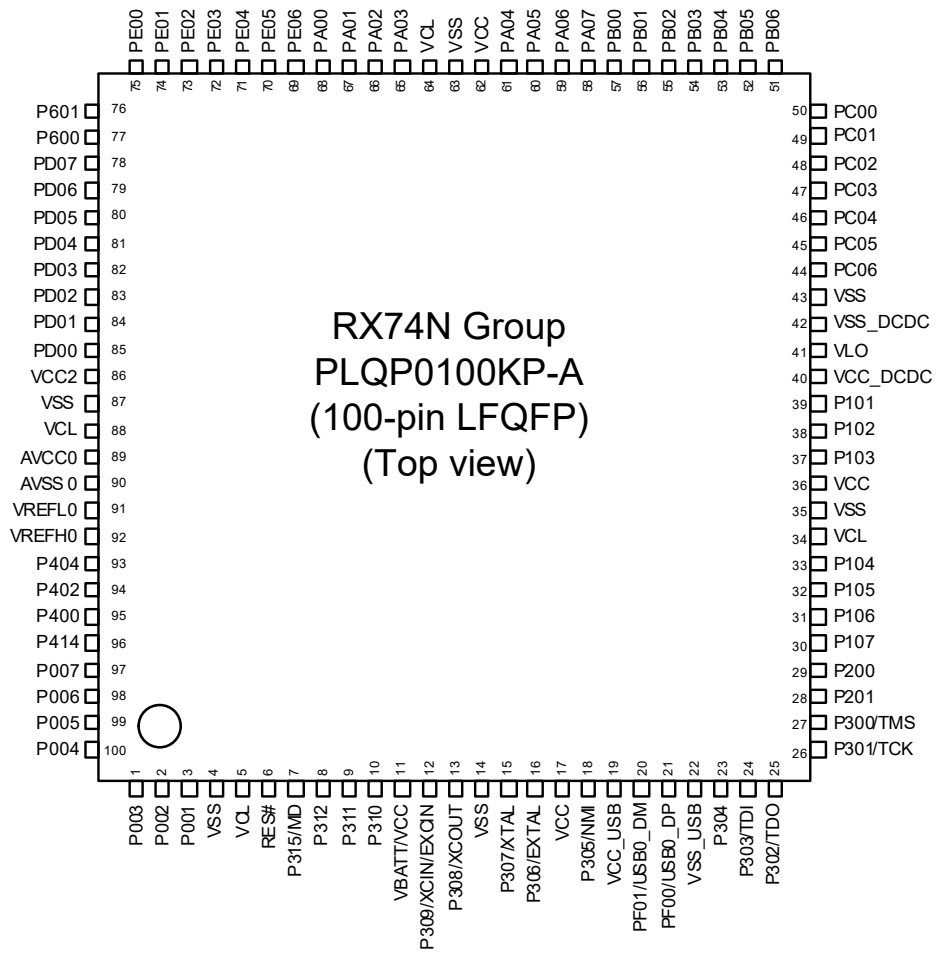
1.5.5 144-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.9, List of Pin and Pin Functions (144-Pin LFQFP).

Figure 1.7 Pin Assignment (144-Pin LFQFP)

1.5.6 100-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, refer to Table 1.10, List of Pin and Pin Functions (100-Pin LFQFP).

Figure 1.8 Pin Assignment (100-Pin LFQFP)

1.6 List of Pin and Pin Functions

1.6.1 320-Pin LFBGA

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (1/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A1	VSS											
A2		P514	PIO_006	D17-A		GTIOC9A	CTS008#/ SSL23-D	ET0_TXD6-A	SD0_WP	IRQ22		
A3		P515	PIO_007	D16-A		GTIOC9B	CTS010#	ET0_TXD7-A	SD0_CD	IRQ23		
A4		PC01	PIO_025	D6-A		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A/ RGMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
A5		PC03	PIO_027	D4-A		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A/ RGMII0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
A6		PC04	PIO_028	D3-A	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A/ RGMII0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
A7	ETHPHYCLK	PC07	PIO_031	D0-A	TMO2/ TOC0/ CACREF	GTETRGD/ GTADSM1/ GTIOC15B/ POEGA	TXD008-A/ SMOSI008-A/ SSDA008/ TXDA008/ MISO0-A/ SSITXD0	ET0_GTX_CLK-A/ RGMII0_TXC-A	SD0_CD	IRQ14		PDMDAT0
A8	CLKOUT	P801	PIO_057	EBCLK-A		GTIV/ GTIOC0B/ POEGC	TXD010-B/ SMOSI010-B/ SSDA010/ TXDA010	ET0_CRS		IRQ9	ADCNV1	
A9		P800	PIO_056	CS1#-A/ DQM0-B		GTIU/ GTIOC0A/ POEGD	RXD010-B/ SMISO010-B/ SSCL010	ET0_COL		IRQ8	ADCNV0	
A10	VLO											
A11	VSS_DCDC											
A12	VSS_DCDC											
A13	VCC_DCDC											
A14		P504	PIO_036		TMCI1	GTETRGA/ GTIOC4A	CTS002#/ RXD007-B/ SMISO007-B/ SSCL007/ MOSI2-B	ET0_TX_ER-C/ GPTP_CAPTURE0		IRQ4		
A15		P103	PIO_011		TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (2/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A16		P104	PIO_012		TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRC URA-DS/ USBHS_VBU S	ET0_TXD0-B/ RMII0_TXD 0-B	PIXD0	IRQ4	ADST0	PCINA0
A17	CLKOUT	P106	PIO_014		TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C/PIXD2	IRQ30-DS	ADTRG0#	PCINA1
A18		P200	PIO_016		TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_ EN-B	SD0_CMD-C/PIXD4	IRQ8	ADTRG2#	PCINZ1
A19	TMS/ SWDIO	P300	PIO_024		TMO1	GTCPPO10	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
A20	VSS											
B1		PG03	PIO_059	DQM2-A/ WR2#-C/ BC2#-C		GTIOC11B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008	ET0_INT/ GPTP_MAT CH1		IRQ11		
B2		P512	PIO_004	D19-A		GTIOC8A	SSL21-D	ET0_TXD4-A	SD0_DAT0-D	IRQ20		PCINZ0
B3		P513	PIO_005	D18-A		GTIOC8B	SSL22-D	ET0_TXD5-A	SD0_DAT1-D	IRQ21		PCINZ1
B4		PC00	PIO_024	D7-A		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
B5		PC02	PIO_026	D5-A		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD 1-A/ RGMII0_TX D1-A	SD0_CMD-A	IRQ10		PCOUTZ0
B6	VSS											
B7		PC05	PIO_029	D2-A	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_ EN-A/ RGMII0_TX _CTL-A	SD0_DAT1-A	IRQ5		PCOUTZ1

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (3/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
B8		PC06	PIO_030	D1-A	TMCI2/ TIC0	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER -A	SD0_WP	IRQ13		PDMCLK0
B9		P802	PIO_058	CS2#-A/ WR#-B/ WR0#-B		GTIW/ GTIOC1A/ POEGB	SCK010-B/ CTS010#/ TXDB010/ CRX2/ USB0_EXICE N	ET0_MDIO-B		IRQ2	ADCNV2	
B10	VLO											
B11	VSS_DCD C											
B12	VSS_DCD C											
B13	VCC_DCD C											
B14		P505	PIO_037		TMO3	GTETRGB/ GTIOC4B	TXD007-B/ SMOSI007-B/ SSDA007/ TXDA007/ MISO2-B	ET0_TX_CL K-C/ GPTP_MAT CH0		IRQ10		
B15		P102	PIO_010		TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
B16	VSS											
B17		P107	PIO_015		TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C/PIXD3	IRQ31-DS	ADTRG1#	PCINB1
B18		P201	PIO_017		TMCI0	GTIW/ GTIOC3B/ GTIOC14B	SCK006-A/ CTS006#/ TXDB006/ MISO0-C/ SCLHS1-A/ USB0_EXICE N/SSILRCK0	ET0_RX_E R-B/ RMII0_RX_ ER-B	SD0_CLK-C/PIXD5	IRQ9		PCINZ0
B19	TCK/ SWCLK	P301	PIO_025		TMCI3	GTCPPO1 1	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		
B20	TDO	P302	PIO_026		TMRI3	GTCPPO1 2	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
C1		PG02	PIO_058	DQM3-A/ WR3#-C/ BC3#-C		GTIOC11A	TXD008-C/ SMOSI008-C/ SSDA008/ TXDA008			IRQ10		
C2		PG01	PIO_057	DQM0-A/ WR#-C/ WR0#-C		GTCPPO1	RXD008-C/ SMISO008-C/ SSCL008			IRQ9		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (4/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
C3		PB01	PIO_017	D14-A	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A/ RGMII0_RXD1-A		IRQ4-DS		PCINB1
C4		PB03	PIO_019	D12-A	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A/ RGMII0_RXD3-A		IRQ3		
C5		PB04	PIO_020	D11-A		GTOVUP/ GTIOC5A	CTS004#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009	ET0_RX_CLK-A/ RMII0_REF50CK-A/ RGMII0_RXC-A		IRQ4		
C6		PB06	PIO_022	D9-A		GTOWUP/ GTIOC6A	RXD009-A/ SMISO009-A/ SSCL009	ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
C7		PB07	PIO_023	D8-A		GTOWLO/ GTIOC6B	TXD009-A/ SMOSI009-A/ SSDA009/ TXDA009	ET0_LINKSTA		IRQ15		
C8		PB05	PIO_021	D10-A	TMRI1	GTOVLO/ GTIOC5B	SCK009-A/ CTS009#/ TXDB009	ET0_RX_DV-A/ RMII0_CRS_DV-A/ RGMII0_RX_CTL-A		IRQ13		
C9		P803	PIO_059	CS3#-A/ WR1#-B/ BC1#-B		GTIOC1B/ POEGA	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTX2/ USB0_ID	ET0_MDC-B		IRQ3	ADSYNC	
C10	VLO											
C11	VLO											
C12	VCC_DCDC											
C13	VCC_DCDC											
C14		P506	PIO_038			GTETRGC/ GTIOC5A/ POEGD	SCK007-B/ CTS007#/ TXDB007/ RSPCK2-B	ET0_TXD2-C/ GPTP0_PPS		IRQ6		
C15		P105	PIO_013		TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ USBHS_OVRCURB/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B	PIXD1	IRQ29-DS	ADST1	PCINB0

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (5/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
C16		P805	PIO_061			GTADSM1/ GTIOC15B	SCK005-B/ CTS005#/ TXDB005/ SSL22-B/ CRX0/ USB0_VBUS EN/ USBHS_VBU SEN	ET0_TXD0-C/ RMII0_TXD 0-C	HSYNC	IRQ5	ADSYNC	PCINB0
C17		P806	PIO_062			GTCPPO7/ GTIOC14A	RXD005-B/ SMISO005-B/ SSCL005/ SSL23-B/ CTX1	ET0_TXD1-C/ RMII0_TXD 1-C	VSYNC	IRQ14		PCINA1
C18		P807	PIO_063			GTCPPO4/ GTIOC14B/ POEGD	TXD005-B/ SMOSI005-B/ SSDA005/ TXDA005/ CRX1	ET0_RX_ER-C/ RMII0_RX_ ER-C	PIXCLK	IRQ15		PCINB1
C19	ETHPHYC LK	P304	PIO_028	ALE-C/ WAIT#-B	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUS EN/ USBHS_ID	ET0_INT		IRQ0-DS	ADTRG0 #	
C20	TDI	P303	PIO_027		TMCI2	GTCPPO1 3	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		
D1		PA06	PIO_014	A1-A/ DQM2-B	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
D2		PG00	PIO_056	DQM1-A/ WR1#-C/ BC1#-C		GTCPPO0	SCK008-C/ CTS008#/ TXDB008			IRQ8		
D3		P511	PIO_003	D20-A		GTETRGA/ GTIOC7B/ POEGC	SSL20-D	ET0_RXD7-A	SD0_CLK-D	IRQ19		
D18		P108	PIO_040		TIC0	GTCPPO8	SCK009-C/ CTS009#/ TXDB009/ RSPCK2-C	ET0_RX_DV-C/ RMII0_CRS _DV-C	SD0_DAT4-C/PIXD0	IRQ16	ADCNV0	
D19		P109	PIO_041		TOC0	GTCPPO9	RXD009-C/ SMISO009-C/ SSCL009/ MOSI2-C	ET0_RX_CLK-C/ RMII0_REF 50CK-C	SD0_DAT5-C/PIXD1	IRQ17	ADCNV1	
D20		P110	PIO_042			GTCPPO1 0	TXD009-C/ SMOSI009-C/ SSDA009/ TXDA009/ MISO2-C	ET0_RXD0-C/ RMII0_RXD 0-C	SD0_DAT6-C/PIXD2	IRQ18	ADCNV2	
E1		PA05	PIO_013	A2-A		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0
E2	ETHPHYC LK	PA07	PIO_015	A0-A/ BC0#-A/ DQM1-B		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0
E3	CLKOUT	P510	PIO_002	D21-A		GTETRGB/ GTIOC7A/ POEGB	MISO2-D	ET0_RXD6-A	SD0_CMD-D	IRQ18		PCOUTZ0

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (6/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
E5		PG05	PIO_061	A10-A		GTIOC12B	RXD003-C/ SMISO003-C/ SSCL003	ET0_MDIO-A		IRQ13		
E6		PB02	PIO_018	D13-A		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS009#	ET0_RXD2-A/ RGMII0_RXD2-A		IRQ2		
E7		PB00	PIO_016	D15-A		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A/ RGMII0_RXD0-A		IRQ12	ADST2	PCINA1
E8	VCC											
E9	VCC											
E10		P500	PIO_032	WR#-A/ WR0#-A/ A16-C		GTIOC2A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS007#/ SSL11-C/ CTX0	ET0_LINKSTA	SD0_DAT4-A	IRQ0	ADST0	
E11		P502	PIO_034	WR2#-A/ BC2#-A/ A18-C		GTIOC3A	TXD002-C/ SMOSI002-C/ SSDA002/ TXDA002/ SSL13-C/ CTX1	ET0_WOL	SD0_DAT6-A	IRQ2	ADST2	
E12		P100	PIO_008	CS1#-B/ WAIT#-A	TMRI3/ CACREF	GTIU/ GTADSM0/ GTIOC12A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS005#/ USB0_OVRCURB-DS/ USBHS_VBUS	ET0_TX_ER-B		IRQ0	ADTRG2#	
E13	VCC											
E14	VCC											
E15		P507	PIO_039			GTETRGD/ GTIOC5B/ POEGC	CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ SSL20-B	ET0_TXD3-C		IRQ7		
E16		P804	PIO_060			GTADSM0/ GTIOC15A	CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL21-B/ CTX0/ USB0_OVRCURA/ USBHS_OVRCURA	ET0_TX_EN-C/ RMII0_TX_EN-C	PCKO	IRQ4		PCINA0
E18		P111	PIO_043			GTCPPO11	CTS012#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ SSL20-C	ET0_RXD1-C/ RMII0_RXD1-C	SD0_DAT7-C/PIXD3	IRQ19		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (7/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
E19		P112	PIO_044			GTCPPO12	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ CTS009#/ SSL21-C	ET0_RXD2-C	PIXD4	IRQ20	ADST0	PDMCLK0
E20		P113	PIO_045			GTCPPO13	SCK012-B/ CTS012#/ TXDB012/ SSL22-C	ET0_RXD3-C	PIXD5	IRQ21	ADST1	PDMDAT0
F1		PA03	PIO_011	A4-A		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
F2		PA04	PIO_012	A3-A	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
F3		P509	PIO_001	D22-A	TOC2	GTIOC6B	MOSI2-D	ET0_RXD5-A	SD0_DAT3-D	IRQ17		PCOUTB0
F5		PG07	PIO_063	A8-A		GTIOC13B	CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003	ET0_CRS		IRQ15		
F6		PG06	PIO_062	A9-A		GTIOC13A	TXD003-C/ SMOSI003-C/ SSDA003/ TXDA003	ET0_COL		IRQ14		
F7		PG04	PIO_060	A11-A		GTIOC12A	SCK003-C/ CTS003#/ TXDB003	ET0_MDC-A		IRQ12		
F8	VSS											
F9	VSS											
F10		P501	PIO_033	WR1#-A/ BC1#-A/ A17-C		GTIOC2B	SCK002-C/ CTS002#/ TXDB002/ SSL12-C/ CRX0	ET0_INT	SD0_DAT5-A	IRQ1	ADST1	
F11		P503	PIO_035	WR3#-A/ BC3#-A/ A19-C		GTIOC3B	RXD002-C/ SMISO002-C/ SSCL002/ CRX1/ USB0_OVRCURB/ SSIRXD0		SD0_DAT7-A	IRQ3		
F12		P101	PIO_009	CS0#-A/ ALE-B	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUSDS/ USBHS_EXICEN	ET0_TX_CLK-B		IRQ1	ADTRG1#	
F13	VSS											
F14	VSS											
F15		PF02	PIO_050	CS6#-B/ BREQ#-C		GTCPPO1/ GTIOC7A	RXD000-C/ SMISO000-C/ SSCL000/ RXD007-C/ SMISO007-C/ SSCL007			IRQ2	ADSYNC	

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (8/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
F16		P115	PIO_047			GTCPPO15	RXD012-B/ SMISO012-B/ SSCL012		PIXD7	IRQ23		
F18		P114	PIO_046			GTCPPO14	TXD012-B/ SMOSI012-B/ SSDA012/ TXDA012/ SSL23-C		PIXD6	IRQ22	ADST2	
F19	VSS											
F20	VCC											
G1		PA02	PIO_010	A5-A		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
G2		PA01	PIO_009	A6-A		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
G3		P508	PIO_000	D23-A	TIC2	GTIOC6A	RSPCK2-D	ET0_RXD4-A	SD0_DAT2-D	IRQ16		PCOUTA0
G5	VCC											
G6	VSS											
G15		PF03	PIO_051	CS7#-B/ BACK#-C		GTCPPO2/ GTIOC7B	TXD000-C/ SMOSI000-C/ SSDA000/ TXDA000/ TXD007-C/ SMOSI007-C/ SSDA007/ TXDA007			IRQ3		
G16		PF04	PIO_052	CS4#-B		GTCPPO3/ GTIOC8A	SCK000-C/ CTS000#/ TXDB000/ SCK007-C/ CTS007#/ TXDB007			IRQ4		
G18	VSS_USB											
G19		PF00	PIO_048			GTCPPO14	USB0_DP			IRQ0		
G20		PF01	PIO_049			GTCPPO15	USB0_DM			IRQ1		
H1	ETHPHYCLK	PA00	PIO_008	A7-A	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	
H2		PG13	PIO_029	D26-A		GTCPPO13	CTS003#	GPTP_PTPOUT1	SD0_DAT5-D/ XSPI1_RESET1#/PIXD5	IRQ29		PCOUTB0
H3		PG15	PIO_031	D24-A		GTCPPO15	CTS011#		SD0_DAT7-D/ XSPI1_RSTO1#/PIXD7	IRQ31		
H5	VCC											
H6	VSS											
H8	VSS											
H9	VCL											
H10	VSS											

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (9/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
H11	VSS											
H12	VCL											
H13	VSS											
H15	ETHPHYCLK	P204	PIO_020	A20-C/ CS7#-C	TMRI1	GTOVUP/ GTIOC1A/ GTIOC5A	CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-B/ SCLHS1-B/ CTX0/ USB0_VBUSEN/ USBHS_VBUS-DS/ SSIBCK1	ET0_RXD0-B/ RMII0_RXD0-B	SD0_WP/ PIXCLK	IRQ12		PCOUTA1
H16		P203	PIO_019	CS2#-B/ BACK#-A		GTOULO/ GTIOC0B/ GTIOC4B	RXD006-A/ SMISO006-A/ SSCL006/ CTS003#/ SSL00-C/ SDA00-B/ CRX2/ USB0_OVRCURB/ USBHS_ID/ SSIBCK0	ET0_RX_CLK-B/ RMII0_REF50CK-B	SD0_DAT1-C/PIXD7	IRQ3		PDMDAT0
H18		PF05	PIO_053	CS5#-B		GTCPPO5/ GTIOC8B/ POEGD	CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ SSL01-C			IRQ5		
H19	VCC_USB											
H20	VCC_USB											
J1		PG11	PIO_027	D28-A		GTCPPO13	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012		PIXD3	IRQ27		PCOUTZ1
J2		PG14	PIO_030	D25-A		GTCPPO14			SD0_DAT6-D/ XSPI1_WP1#/PIXD6	IRQ30		PCOUTZ0
J3		PG12	PIO_028	D27-A		GTCPPO12	CTS008#/ CTS005#	GPTP_PTPOUT0	SD0_DAT4-D/ XSPI1_INT1#/ XSPI1_ECS1#/PIXD4	IRQ28		PCOUTA0
J5	VCC											
J6	VSS											
J8	VSS											
J9	VCL											
J10	VSS											
J11	VSS											
J12	VSS											
J13	VCL											

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (10/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
J15	CLKOUT	P205	PIO_021	A21-C/ CS6#-C		GTOVLO/ GTIOC1B/ GTIOC5B	SCK001-B/ CTS001#/ TXDB001/ RSPCK1-B/ SDAHS1-B/ CRX0/ USB0_VBUS/ USBHS_VBUSEN/ SSIRXD1	ET0_RXD1-B/ RMII0_RXD1-B	SD0_CD/ HSYNC	IRQ5	ADTRG0#	PCOUTB1
J16		P202	PIO_018	CS3#-B/ BREQ#-A	TMO0	GTOUUP/ GTIOC0A/ GTIOC4A	TXD006-A/ SMOSI006-A/ SSDA006/ TXDA006/ RSPCK0-C/ SCL00-B/ CTX2/ USB0_OVRCURA/ USBHS_EXICEN/ AUDIO_CLK	ET0_RX_DV-B/ RMII0_CRS_DV-B	SD0_DAT0-C/PIXD6	IRQ15		PDMCLK0
J18	AVSS_USBHS											
J19							USBHS_DP					
J20							USBHS_DM					
K1		PG09	PIO_025	D30-A	TOC3	GTCPPO11	RXD012-C/ SMISO012-C/ SSCL012		PIXD1	IRQ25		PCOUTA1
K2		PG10	PIO_026	D29-A		GTCPPO12	TXD012-C/ SMOSI012-C/ SSDA012/ TXDA012		PIXD2	IRQ26		PCOUTB1
K3		P705	PIO_053	RAS#-A/ CS5#-A		GTIOC12B	SCK011-B/ CTS011#/ TXDB011		XSPI1_RSTO1#	IRQ13	ADCNV1	
K5		P704	PIO_052	SDCS#-A/ CS4#-A		GTIOC12A	CTS004#/ CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011		XSPI1_INT1#/ XSPI1_ECS1#	IRQ12	ADCNV0	
K6		P702	PIO_050	RD#-A/ DQM3-B		GTETRGA/ GTIOC11A	SCK004-B/ CTS004#/ TXDB004		XSPI1_WP0#/ XSPI1_RESET1#/ HSYNC	IRQ10	ADST2	
K8	VCL											
K9	VSS											
K10	VCL											
K11	VCL											
K12	VCC											
K13	VSS											
K15		P207	PIO_023	A23-C/ CS4#-C	TMRI3	GTOWLO/ GTIOC2B/ GTIOC6B	RXD001-B/ SMISO001-B/ SSCL001/ MISO1-B/ SDAHS2-B/ CRX1/ USB0_ID/ USBHS_OVRCURB-DS	ET0_RXD3-B	PCKO	IRQ3-DS	ADTRG2#	

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (11/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
K16		P206	PIO_022	A22-C/ CS5#-C	TMO3/ RTCOUT	GTOWUP/ GTIOC2A/ GTIOC6A	CTS006#/ TXD001-B/ SMOSI001-B/ SSDA001/ TXDA001/ MOSI1-B/ SCLHS2-B/ CTX1/ USB0_EXICEN/ USBHS_OVRCURA-DS/ SSITXD1	ET0_RXD2-B	VSYNC	IRQ2-DS	ADTRG1#	PCOUTZ1
K18	USBHS_REF											
K19	VSS_USBHS											
K20	VCC_USBHS											
L1		PG08	PIO_024	D31-A	TIC3	GTCPPO10	SCK012-C/ CTS012#/ TXDB012		PIXD0	IRQ24		
L2	VSS											
L3		P707	PIO_055	WE#-A/ CS7#-A		GTIOC13B	TXD011-B/ SMOSI011-B/ SSDA011/ TXDA011		XSPI1_RESET1#	IRQ7		
L5		P706	PIO_054	CAS#-A/ CS6#-A		GTIOC13A	RXD011-B/ SMISO011-B/ SSCL011		XSPI1_WP1#	IRQ14	ADCNV2	
L6		P703	PIO_051	ALE-A/ WR2#-B/ BC2#-B	CACREF	GTETRGB/ GTIOC11B	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004		XSPI1_RESET0#/PCKO	IRQ8		
L8	VSS											
L9	VSS											
L10	VCL											
L11	VCL											
L12	VSS											
L13	VCC											
L15		P310	PIO_058		RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
L16		P305								NMI/IRQ0		
L18	AVCC_USBHS											
L19	VCC											
L20	VCC											
M1		P701	PIO_049	CKE-A/ WR3#-B/ BC3#-B		GTIOC10B	RXD004-B/ SMISO004-B/ SSCL004		XSPI1_RSTO0#/ XSPI1_RESET0#/ VSYNC	IRQ1	ADST1	
M2		P715	PIO_023	A12-A		GTCPPO9/ GTIOC15B	RXD010-C/ SMISO010-C/ SSCL010		HSYNC	IRQ23		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (12/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
M3		P714	PIO_022	A13-A		GTCPPO8/ GTIOC15A	TXD010-C/ SMOSI010-C/ SSDA010/ TXDA010		VSYNC	IRQ22		
M5	VCC											
M6	VSS											
M8	VCL											
M9	VCL											
M10	VSS											
M11	VSS											
M12	VSS											
M13	VCL											
M15		P312	PIO_060	BACK#-B	RTCIC2	GTETRGD/ GTCPPO1 5/POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2 #	TAMPI2
M16		P311	PIO_059	BREQ#-B	RTCIC1	GTETRGC/ GTCPPO1 4/POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1 #	TAMPI1
M18	VSS											
M19	EXTAL	P306										
M20	XTAL	P307										
N1	ETHPHYC LK	P700	PIO_048	SDCLK-A		GTIOC10A	TXD004-B/ SMOSI004-B/ SSDA004/ TXDA004		XSPI1_INT0 #/ XSPI1_ECS 0#/PIXCLK	IRQ0	ADST0	
N2		P712	PIO_020	A15-A		GTCPPO5/ GTIOC14A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTS011#		PCKO	IRQ20		
N3		P713	PIO_021	A14-A		GTCPPO6/ GTIOC14B	SCK010-C/ CTS010#/ TXDB010		PIXCLK	IRQ21		
N5	VCC											
N6	VSS											
N8	VSS											
N9	VCL											
N10	VSS											
N11	VSS											
N12	VCL											
N13	VSS											
N15		P209	PIO_049		TOC1	GTETRGB	RXD004-C/ SMISO004-C/ SSCL004/ MOSI1-D			IRQ25		PCINB1
N16		P208	PIO_048		TIC1	GTETRGA	SCK004-C/ CTS004#/ TXDB004/ RSPCK1-D			IRQ24		PCINA1
N18	MD/FINED	P315										
N19	VSS											
N20	VSS											

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (13/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
P1		PE06	PIO_046	A9-B/A21-D	TIC1	GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS/ PIXD6	IRQ6		
P2		PE07	PIO_047	A8-B/A20-D	TOC1	GTETRGD/ GTCPPO7/ GTIOC10B/ POEGD			XSPI1_CS0#/PIXD7	IRQ7		
P3		PF09	PIO_017	A20-A/ A16-D			CTS012#	GPTP_CAPTURE1	XSPI1_INT0#/ XSPI1_ECS0#	IRQ17		
P5	VCC											
P6	VSS											
P15		P211	PIO_051			GTETRGD	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS005#/ SSL10-D/ USBHS_OVRCURB			IRQ27		
P16		P210	PIO_050			GTETRGC	TXD004-C/ SMOSI004-C/ SSDA004/ TXDA004/ MISO1-D			IRQ26		PCINZ1
P18	VSS											
P19	XCIN/ EXCIN	P309										
P20	XCOUT	P308										
R1		PE03	PIO_043	A12-B	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/ XSPI1_IO1/ PIXD3	IRQ11		
R2		PE05	PIO_045	A10-B/ A22-D		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3/ PIXD5	IRQ5		
R3		PF08	PIO_016	A21-A/ A17-D		GTCPPO12/POEGA			XSPI1_RESET0#	IRQ16		
R5		P607	PIO_015	CS0#-B/ D16-B		GTETRGB/ GTADSM0/ GTIOC15B/ POEGA	CTS008#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ CRX2		XSPI0_RSTO0#	IRQ15		
R6		P606	PIO_014	CS1#-C/ D17-B		GTETRGA/ GTCPPO4/ GTIOC15A/ POEGC	SCK009-B/ CTS009#/ TXDB009/ SSL03-D/ CTX2		XSPI0_RESET0#	IRQ14		
R7	VSS											
R8	VSS											
R9	VSS											
R10		P906	PIO_006			GTCPPO2/ POEGA	SCK006-B/ CTS006#/ TXDB006	ET0_MDC-C		IRQ14		
R11		P405	PIO_053							IRQ13-DS	AN005	

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (14/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
R12		P404	PIO_052							IRQ12-DS	AN004	
R13	VSS											
R14	VSS											
R15		P215	PIO_055			GTCPPO13	CTS004#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ USBHS_ID			IRQ31		PDMDAT0 /PCINZ0
R16		P214	PIO_054			GTCPPO12	TXD005-C/ SMOSI005-C/ SSDA005/ TXDA005/ SSL13-D/ USBHS_EXICEN			IRQ30	ADTRG2#	PDMCLK0 /PCINB0
R18	RES#											
R19	VCC											
R20	VBATT/ VCC											
T1	CLKOUT	PE01	PIO_041	A14-B		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/ XSPI1_CK/ PIXD1	IRQ9		
T2		PE04	PIO_044	A11-B/ A23-D		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0#/PIXD4	IRQ12		
T3		PF07	PIO_055	A22-A/ A18-D		GTCPPO11/GTIOC9B	SSL03-C		XSPI1_RSTO0#	IRQ7		
T5		P604	PIO_012	CS3#-C/ D19-B		GTIOC6A	TXD009-B/ SMOSI009-B/ SSDA009/ TXDA009/ SSL01-D		XSPI0_CS1#/ XSPI0_RSTO0#	IRQ4		
T6		P602	PIO_010	CAS#-B/ D21-B		GTIOC5A	SCK008-B/ CTS008#/ TXDB008/ MOSI0-D		XSPI0_IO6/ HSYNC	IRQ2	ADTRG2#	
T7	VCC2											
T8	VCC2											
T9	VCC2											
T10		P907	PIO_007			GTCPPO3/ POEGD	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ CTS001#	ET0_MDIO-C	XSPI0_WP0#	IRQ15		
T11		P406	PIO_054							IRQ14-DS	AN006	
T12		P407	PIO_055							IRQ15-DS	AN007	
T13	VCC											
T14	VCC											
T15		P013	PIO_037			GTCPPO13		GPTP_CAPTURE1		IRQ29		
T16		P012	PIO_036			GTCPPO12	CTS006#	GPTP0_PPS		IRQ28		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (15/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
T18		P213	PIO_053			GTCPPO11	RXD005-C/ SMISO005-C/ SSCL005/ SSL12-D/ USBHS_VBUSEN			IRQ29	ADTRG1#	PCINA0
T19		P212	PIO_052			GTCPPO10	SCK005-C/ CTS005#/ TXDB005/ SSL11-D/ USBHS_OVRCURA			IRQ28	ADTRG0#	
T20		P000	PIO_000		TMRI0	GTCPPO7/ GTIOC10A/ POEGB	CTS000#/ CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ AUDIO_CLK	ET0_INT		IRQ16-DS	AN016	
U1	ETHPHYCLK	PE00	PIO_040	A15-B		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN#/ XSPI1_CS1#/PIXD0	IRQ8		
U2		PE02	PIO_042	A13-B	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/ XSPI1_IO0/ PIXD2	IRQ7-DS		
U3		PF06	PIO_054	A23-A/ A19-D		GTCPPO6/ GTIOC9A/ POEGC	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS007#/ SSL02-C		XSPI1_CS1#/ XSPI1_WP0#	IRQ6		
U18		P008	PIO_032			GTCPPO8/ GTIOC14A	TXD006-C/ SMOSI006-C/ SSDA006/ TXDA006			IRQ24	ADTRG1#	
U19		P009	PIO_033			GTCPPO9/ GTIOC14B	RXD006-C/ SMISO006-C/ SSCL006	ET0_LINKSTA		IRQ25	ADTRG2#	
U20		P001	PIO_001		TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
V1		P709	PIO_017	A18-A		GTETRGD/ GTCPPO7/ GTIOC10B	RXD011-C/ SMISO011-C/ SSCL011		XSPI1_IO5	IRQ17		
V2		P711	PIO_019	A16-A		GTETRGB/ GTADSM1/ GTIOC11B	CTS010#/ CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011		XSPI1_IO7	IRQ19		
V3		P605	PIO_013	CS2#-C/ D18-B		GTIOC6B	RXD009-B/ SMISO009-B/ SSCL009/ SSL02-D		XSPI0_INT0#/ XSPI0_ECS0#	IRQ13		
V4		P603	PIO_011	WE#-B/ D20-B		GTIOC5B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS009#/ MISO0-D		XSPI0_IO7/ PCKO	IRQ3		

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (16/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
V5		P600	PIO_008	SDCS#-B/ D23-B		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTP OUT0	XSPI0_IO4/ PIXCLK	IRQ0	ADTRG0 #	
V6		P601	PIO_009	RAS#-B/ D22-B		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTP OUT1	XSPI0_IO5/ VSYNC	IRQ1	ADTRG1 #	
V7		P905	PIO_005			GTCPPO1/ POEGB	RXD006-B/ SMISO006-B/ SSCL006	ET0_CRS/ GPTP_PTP OUT1		IRQ13		
V8		P904	PIO_004			GTCPPO0/ POEGC	TXD006-B/ SMOSI006-B/ SSDA006/ TXDA006	ET0_COL/ GPTP_PTP OUT0		IRQ12		
V9	AVCC0											
V10	AVSS0											
V11	VREFL00											
V12	VREFH00											
V13		P412	PIO_060							IRQ28-DS	AN012	
V14		P413	PIO_061							IRQ29	AN013	
V15		P414	PIO_062							IRQ30	AN014/ DA0	
V16		P415	PIO_063							IRQ31	AN015/ DA1	
V17		P014	PIO_038			GTCPPO1 4		ET0_COL/ GPTP_MAT CH1		IRQ30		
V18		P011	PIO_035			GTCPPO1 1/ GTIOC15B	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006	GPTP_MAT CH0		IRQ27		
V19		P010	PIO_034			GTCPPO1 0/ GTIOC15A	SCK006-C/ CTS006#/ TXDB006	ET0_WOL/ GPTP_CAP TURE0		IRQ26		
V20		P002	PIO_002		TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
W1		P708	PIO_016	A19-A		GTETRGC/ GTCPPO4/ GTIOC10A	SCK011-C/ CTS011#/ TXDB011	ET0_INT	XSPI1_IO4	IRQ16		
W2		P710	PIO_018	A17-A		GTETRGA/ GTADSM0/ GTIOC11A	TXD011-C/ SMOSI011-C/ SSDA011/ TXDA011		XSPI1_IO6	IRQ18		
W3		PD05	PIO_037	A18-B/ D26-B		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1-B/ XSPI0_IO3/ PIXD5	IRQ5	AN029	
W4		PD03	PIO_035	A20-B/ D28-B	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK-B/ XSPI0_IO1/ PIXD3	IRQ3	AN027	
W5	VSS											

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (17/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
W6		PD02	PIO_034	A21-B/ D29-B	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD-B/ XSPI0_IO0/ PIXD2	IRQ2	AN026	
W7		P902	PIO_002			GTETRGC/ GTCPPO8	SCK001-C/ CTS001#/ TXDB001		XSPI0_RESET1#	IRQ10	AN034	
W8		P900	PIO_000			GTETRGA/ GTCPPO5	RXD001-C/ SMISO001-C/ SSCL001		XSPI0_INT1#/ XSPI0_ECS1#	IRQ0	AN032	
W9	AVCC0											
W10	AVSS0											
W11	VREFL01											
W12	VREFH01											
W13		P403	PIO_051							IRQ11-DS	AN003	
W14		P401	PIO_049							IRQ9-DS	AN001	
W15		P411	PIO_059							IRQ27-DS	AN011	
W16		P409	PIO_057							IRQ25-DS	AN009	
W17	VSS											
W18		P006	PIO_006			GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1#/AN022	PDMCLK0/ PCOUTZ1
W19		P004	PIO_004			GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1
W20		P003	PIO_003			GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0
Y1	VSS											
Y2		PD07	PIO_039	A16-B/ D24-B		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0#/ XSPI0_RESET1#/PIXD7	IRQ7	AN031	
Y3		PD06	PIO_038	A17-B/ D25-B		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS/ PIXD6	IRQ6	AN030	
Y4		PD04	PIO_036	A19-B/ D27-B		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0-B/ XSPI0_IO2/ XSPI0_WP0#/PIXD4	IRQ4	AN028	
Y5		PD01	PIO_033	A22-B/ D30-B		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3-B/ XSPI0_CK/ PIXD1	IRQ1	AN025	

Table 1.5 List of Pin and Pin Functions (320-Pin LFBGA) (18/18)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
320-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
Y6		PD00	PIO_032	A23-B/ D31-B		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2-B/ XSPI0_CKN#/ XSPI0_CS1#/PIXD0	IRQ0	AN024	
Y7		P903	PIO_003			GTETRGD/ GTCPPO9/ POEGD	CTS006#/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001		XSPI0_WP1#	IRQ11	AN035	
Y8		P901	PIO_001			GTETRGB/ GTCPPO6	TXD001-C/ SMOSI001-C/ SSDA001/ TXDA001		XSPI0_RSTO1#/ XSPI0_RESET0#	IRQ9	AN033	
Y9	AVCC0											
Y10	AVSS0											
Y11	VREFL02											
Y12	VREFH02											
Y13		P402	PIO_050							IRQ10-DS	AN002	
Y14		P400	PIO_048							IRQ8-DS	AN000	
Y15		P410	PIO_058							IRQ26-DS	AN010	
Y16		P408	PIO_056							IRQ24-DS	AN008	
Y17		P015	PIO_039			GTCPPO15		ET0_CRS		IRQ31		
Y18		P007	PIO_007			GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0#/AN023	PDMDAT0
Y19		P005	PIO_005			GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2#/AN021	PCOUTB1
Y20	VSS											

1.6.2 224-Pin LFBGA

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (1/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A1	VSS											
A2		PC03	PIO_027	D4-A		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A/ RGMII0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
A3	ETHPHYCLK	PC07	PIO_031	D0-A	TMO2/ TOC0/ CACREF	GTETRGD/ GTADSM1/ GTIOC15B/ POEGA	TXD008-A/ SMOSI008-A/ SSDA008/ TXDA008/ MISO0-A/ SSITXD0	RGMII0_TXC-A	SD0_CD	IRQ14		PDMDAT0
A4	CLKOUT	P801	PIO_057	EBCLK-A		GTIV/ GTIOC0B/ POEGC	TXD010-B/ SMOSI010-B/ SSDA010/ TXDA010	ET0_CRS		IRQ9	ADCNV1	
A5		P802	PIO_058	CS2#-A/ WR#-B/ WR0#-B		GTIW/ GTIOC1A/ POEGB	SCK010-B/ CTS010#/ TXDB010/ CRX2/ USB0_EXICEN	ET0_MDIO-B		IRQ2	ADCNV2	
A6	VLO											
A7	VSS_DCDC											
A8	VCC_DCDC											
A9		P103	PIO_011		TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		
A10		P104	PIO_012		TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRCURA-DS/ USBHS_VBUS	ET0_TXD0-B/ RMII0_TXD0-B	PIXD0	IRQ4	ADST0	PCINA0
A11	CLKOUT	P106	PIO_014		TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C/PIXD2	IRQ30-DS	ADTRG0#	PCINA1
A12		P200	PIO_016		TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_EN-B	SD0_CMD-C/PIXD4	IRQ8	ADTRG2#	PCINZ1
A13	TCK/ SWCLK	P301	PIO_025		TMCI3	GTCPPO11	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (2/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A14	TDO	P302	PIO_026		TMRI3	GTCPPO12	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
A15	VSS											
B1		PC04	PIO_028	D3-A	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A/ RGMII0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
B2		PC01	PIO_025	D6-A		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A/ RGMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
B3		PC05	PIO_029	D2-A	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_EN-A/ RGMII0_TX_CTL-A	SD0_DAT1-A	IRQ5		PCOUTZ1
B4		PC06	PIO_030	D1-A	TMCI2/ TIC0	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER-A	SD0_WP	IRQ13		PDMCLK0
B5		P803	PIO_059	CS3#-A/ WR1#-B/ BC1#-B		GTIOC1B/ POEGA	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTX2/ USB0_ID	ET0_MDC-B		IRQ3	ADSYNC	
B6	VLO											
B7	VSS_DCDC											
B8	VCC_DCDC											
B9		P102	PIO_010		TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
B10		P105	PIO_013		TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ USBHS_OVRCURB/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B	PIXD1	IRQ29-DS	ADST1	PCINB0
B11		P107	PIO_015		TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C/PIXD3	IRQ31-DS	ADTRG1#	PCINB1

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (3/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
B12		P201	PIO_017		TMCI0	GTIW/ GTIOC3B/ GTIOC14B	SCK006-A/ CTS006#/ TXDB006/ MISO0-C/ SCLHS1-A/ USB0_EXICEN/SSILRCK0	ET0_RX_ER-B/ RMII0_RX_ER-B	SD0_CLK-C/PIXD5	IRQ9		PCINZ0
B13	TMS/ SWDIO	P300	PIO_024		TMO1	GTCPPO10	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
B14	TDI	P303	PIO_027		TMCI2	GTCPPO13	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		
B15	ETHPHYCLK	P304	PIO_028	ALE-C/ WAIT#-B	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUSEN/ USBHS_ID	ET0_INT		IRQ0-DS	ADTRG0#	
C1		PC00	PIO_024	D7-A		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
C2		PC02	PIO_026	D5-A		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD1-A/ RGMII0_TXD1-A	SD0_CMD-A	IRQ10		PCOUTZ0
C3		PB06	PIO_022	D9-A		GTOWUP/ GTIOC6A	RXD009-A/ SMISO009-A/ SSCL009	ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
C4		PB05	PIO_021	D10-A	TMRI1	GTOVLO/ GTIOC5B	SCK009-A/ CTS009#/ TXDB009	ET0_RX_DV-A/ RMII0_CRS_DV-A/ RGMII0_RX_CTL-A		IRQ13		
C5		P800	PIO_056	CS1#-A/ DQM0-B		GTIU/ GTIOC0A/ POEGD	RXD010-B/ SMISO010-B/ SSCL010	ET0_COL		IRQ8	ADCNV0	
C6	VLO											
C7		P100	PIO_008	CS1#-B/ WAIT#-A	TMRI3/ CACREF	GTIU/ GTADSM0/ GTIOC12A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS005#/ USB0_OVRCURB-DS/ USBHS_VBUS	ET0_TX_ER-B		IRQ0	ADTRG2#	

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (4/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
C8	VCC_DCDC											
C9		P804	PIO_060			GTADSM0/ GTIOC15A	CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL21-B/ CTX0/ USB0_OVRCURA/ USBHS_OVRCURA	ET0_TX_EN-C/ RMII0_TX_EN-C	PCKO	IRQ4		PCINA0
C10		P805	PIO_061			GTADSM1/ GTIOC15B	SCK005-B/ CTS005#/ TXDB005/ SSL22-B/ CRX0/ USB0_VBUSEN/ USBHS_VBUSEN	ET0_TXD0-C/ RMII0_TXD0-C	HSYNC	IRQ5	ADSYNC	PCINB0
C11		P114	PIO_046			GTCPPO14	TXD012-B/ SMOSI012-B/ SSDA012/ TXDA012/ SSL23-C		PIXD6	IRQ22	ADST2	
C12		P111	PIO_043			GTCPPO11	CTS012#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ SSL20-C	ET0_RXD1-C/ RMII0_RXD1-C	SD0_DAT7-C/PIXD3	IRQ19		
C13		P108	PIO_040		TIC0	GTCPPO8	SCK009-C/ CTS009#/ TXDB009/ RSPCK2-C	ET0_RX_DV-C/ RMII0_CRS_DV-C	SD0_DAT4-C/PIXD0	IRQ16	ADCNV0	
C14		P110	PIO_042			GTCPPO10	TXD009-C/ SMOSI009-C/ SSDA009/ TXDA009/ MISO2-C	ET0_RXD0-C/ RMII0_RXD0-C	SD0_DAT6-C/PIXD2	IRQ18	ADCNV2	
C15		P113	PIO_045			GTCPPO13	SCK012-B/ CTS012#/ TXDB012/ SSL22-C	ET0_RXD3-C	PIXD5	IRQ21	ADST1	PDMDAT0
D1		PG00	PIO_056	DQM1-A/ WR1#-C/ BC1#-C		GTCPPO0	SCK008-C/ CTS008#/ TXDB008			IRQ8		
D2		PG02	PIO_058	DQM3-A/ WR3#-C/ BC3#-C		GTIOC11A	TXD008-C/ SMOSI008-C/ SSDA008/ TXDA008			IRQ10		
D3		PB04	PIO_020	D11-A		GTOVUP/ GTIOC5A	CTS004#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009	ET0_RX_CLK-A/ RMII0_REF50CK-A/ RGMII0_RXC-A		IRQ4		
D4		PB07	PIO_023	D8-A		GTOWLO/ GTIOC6B	TXD009-A/ SMOSI009-A/ SSDA009/ TXDA009	ET0_LINKSTA		IRQ15		

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (5/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
D5		P502	PIO_034	WR2#-A/ BC2#-A/ A18-C		GTIOC3A	TXD002-C/ SMOSI002-C/ SSDA002/ TXDA002/ SSL13-C/ CTX1	ET0_WOL	SD0_DAT6-A	IRQ2	ADST2	
D6		P500	PIO_032	WR#-A/ WR0#-A/ A16-C		GTIOC2A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS007#/ SSL11-C/ CTX0	ET0_LINKSTA	SD0_DAT4-A	IRQ0	ADST0	
D7		P101	PIO_009	CS0#-A/ ALE-B	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUS-DS/ USBHS_EXICEN	ET0_TX_CLK-B		IRQ1	ADTRG1#	
D8		P504	PIO_036		TMCI1	GTETRGA/ GTIOC4A	CTS002#/ RXD007-B/ SMISO007-B/ SSCL007/ MOSI2-B	ET0_TX_ER-C/ GPTP_CAPTURE0		IRQ4		
D9		P506	PIO_038			GTETRGC/ GTIOC5A/ POEGD	SCK007-B/ CTS007#/ TXDB007/ RSPCK2-B	ET0_TXD2-C/ GPTP0_PPS		IRQ6		
D10		P806	PIO_062			GTCPPO7/ GTIOC14A	RXD005-B/ SMISO005-B/ SSCL005/ SSL23-B/ CTX1	ET0_TXD1-C/ RMII0_TXD1-C	VSYNC	IRQ14		PCINA1
D11		P109	PIO_041		TOC0	GTCPPO9	RXD009-C/ SMISO009-C/ SSCL009/ MOSI2-C	ET0_RX_CLK-C/ RMII0_REF50CK-C	SD0_DAT5-C/PIXD1	IRQ17	ADCNV1	
D12		P115	PIO_047			GTCPPO15	RXD012-B/ SMISO012-B/ SSCL012		PIXD7	IRQ23		
D13		P112	PIO_044			GTCPPO12	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ CTS009#/ SSL21-C	ET0_RXD2-C	PIXD4	IRQ20	ADST0	PDMCLK0
D14	VSS											
D15	VCC											
E1		PG01	PIO_057	DQM0-A/ WR#-C/ WR0#-C		GTCPPO1	RXD008-C/ SMISO008-C/ SSCL008			IRQ9		
E2		PG03	PIO_059	DQM2-A/ WR2#-C/ BC2#-C		GTIOC11B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008	ET0_INT/ GPTP_MATCH1		IRQ11		
E3		PB02	PIO_018	D13-A		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS009#	ET0_RXD2-A/ RGMII0_RXD2-A		IRQ2		

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (6/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
E4		PB03	PIO_019	D12-A	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A/ RGMII0_RXD3-A		IRQ3		
E5	VCC											
E6		P501	PIO_033	WR1#-A/ BC1#-A/ A17-C		GTIOC2B	SCK002-C/ CTS002#/ TXDB002/ SSL12-C/ CRX0	ET0_INT	SD0_DAT5-A	IRQ1	ADST1	
E7		P503	PIO_035	WR3#-A/ BC3#-A/ A19-C		GTIOC3B	RXD002-C/ SMISO002-C/ SSCL002/ CRX1/ USB0_OVRCURB/ SSIRXD0		SD0_DAT7-A	IRQ3		
E8		P505	PIO_037		TMO3	GTETRGB/ GTIOC4B	TXD007-B/ SMOSI007-B/ SSDA007/ TXDA007/ MISO2-B	ET0_TX_CLK-C/ GPTP_MATCH0		IRQ10		
E9		P507	PIO_039			GTETRGD/ GTIOC5B/ POEGC	CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ SSL20-B	ET0_TXD3-C		IRQ7		
E10		P807	PIO_063			GTCPPO4/ GTIOC14B/ POEGD	TXD005-B/ SMOSI005-B/ SSDA005/ TXDA005/ CRX1	ET0_RX_ER-C/ RMII0_RX_ER-C	PIXCLK	IRQ15		PCINB1
E11		PF02	PIO_050	CS6#-B/ BREQ#-C		GTCPPO1/ GTIOC7A	RXD000-C/ SMISO000-C/ SSCL000/ RXD007-C/ SMISO007-C/ SSCL007			IRQ2	ADSYNC	
E12		PF04	PIO_052	CS4#-B		GTCPPO3/ GTIOC8A	SCK000-C/ CTS000#/ TXDB000/ SCK007-C/ CTS007#/ TXDB007			IRQ4		
E13		PF03	PIO_051	CS7#-B/ BACK#-C		GTCPPO2/ GTIOC7B	TXD000-C/ SMOSI000-C/ SSDA000/ TXDA000/ TXD007-C/ SMOSI007-C/ SSDA007/ TXDA007			IRQ3		
E14		PF00	PIO_048			GTCPPO14	USB0_DP			IRQ0		
E15		PF01	PIO_049			GTCPPO15	USB0_DM			IRQ1		
F1		PA05	PIO_013	A2-A		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0
F2	ETHPHYCLK	PA07	PIO_015	A0-A/ BC0#-A/ DQM1-B		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (7/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
F3		PB01	PIO_017	D14-A	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A/ RGMII0_RXD1-A		IRQ4-DS		PCINB1
F4		PB00	PIO_016	D15-A		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A/ RGMII0_RXD0-A		IRQ12	ADST2	PCINA1
F5	VSS											
F6	VCL											
F7	VSS											
F8	VCC											
F9	VSS											
F10	VSS											
F11		PF05	PIO_053	CS5#-B		GTCPPO5/ GTIOC8B/ POEGD	CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ SSL01-C			IRQ5		
F12		P202	PIO_018	CS3#-B/ BREQ#-A	TMO0	GTOUUP/ GTIOC0A/ GTIOC4A	TXD006-A/ SMOSI006-A/ SSDA006/ TXDA006/ RSPCK0-C/ SCL00-B/ CTX2/ USB0_OVRCURA/ USBHS_EXICEN/ AUDIO_CLK	ET0_RX_DV-B/ RMII0_CRS_DV-B	SD0_DAT0-C/PIXD6	IRQ15		PDMCLK0
F13		P203	PIO_019	CS2#-B/ BACK#-A		GTOULO/ GTIOC0B/ GTIOC4B	RXD006-A/ SMISO006-A/ SSCL006/ CTS003#/ SSL00-C/ SDA00-B/ CRX2/ USB0_OVRCURB/ USBHS_ID/ SSIBCK0	ET0_RX_CLK-B/ RMII0_REF50CK-B	SD0_DAT1-C/PIXD7	IRQ3		PDMDAT0
F14	VCC_USB											
F15	VSS_USB											
G1		PA04	PIO_012	A3-A	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
G2		PA06	PIO_014	A1-A/ DQM2-B	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
G3		PG05	PIO_061	A10-A		GTIOC12B	RXD003-C/ SMISO003-C/ SSCL003	ET0_MDIO-A		IRQ13		

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (8/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
G4		PG07	PIO_063	A8-A		GTIOC13B	CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003	ET0_CRS		IRQ15		
G5		PF09	PIO_017	A20-A/ A16-D			CTS012#	GPTP_CAPTURE1	XSPI1_INT0#/ XSPI1_ECS0#	IRQ17		
G6	VSS											
G7	VCL											
G8	VSS											
G9	VCL											
G10	VCL											
G11	CLKOUT	P205	PIO_021	A21-C/ CS6#-C		GTOVLO/ GTIOC1B/ GTIOC5B	SCK001-B/ CTS001#/ TXDB001/ RSPCK1-B/ SDAHS1-B/ CRX0/ USB0_VBUS/ USBHS_VBUSEN/ SSIRXD1	ET0_RXD1-B/ RMII0_RXD1-B	SD0_CD/ HSYNC	IRQ5	ADTRG0#	PCOUTB1
G12	ETHPHYCLK	P204	PIO_020	A20-C/ CS7#-C	TMRI1	GTOVUP/ GTIOC1A/ GTIOC5A	CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-B/ SCLHS1-B/ CTX0/ USB0_VBUSEN/ USBHS_VBUS-DS/ SSIBCK1	ET0_RXD0-B/ RMII0_RXD0-B	SD0_WP/ PIXCLK	IRQ12		PCOUTA1
G13	VCC_USBHS											
G14							USBHS_DP					
G15							USBHS_DM					
H1		PA02	PIO_010	A5-A		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
H2		PA03	PIO_011	A4-A		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
H3		PG06	PIO_062	A9-A		GTIOC13A	TXD003-C/ SMOSI003-C/ SSDA003/ TXDA003	ET0_COL		IRQ14		
H4		PG04	PIO_060	A11-A		GTIOC12A	SCK003-C/ CTS003#/ TXDB003	ET0_MDC-A		IRQ12		
H5		PF08	PIO_016	A21-A/ A17-D		GTCPPO12/POEGA			XSPI1_RESET0#	IRQ16		
H6	VCC											
H7	VSS											
H8	VCL											
H9	VSS											

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (9/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
H10	VCL											
H11		P206	PIO_022	A22-C/ CS5#-C	TMO3/ RTCOUT	GTOWUP/ GTIOC2A/ GTIOC6A	CTS006#/ TXD001-B/ SMOSI001-B/ SSDA001/ TXDA001/ MOSI1-B/ SCLHS2-B/ CTX1/ USB0_EXICEN/ USBHS_OVRCURA-DS/ SSITXD1	ET0_RXD2-B	VSYNC	IRQ2-DS	ADTRG1#	PCOUTZ1
H12	AVSS_USBHS											
H13	USBHS_REF											
H14	VSS_USBHS											
H15	VCC											
J1	ETHPHYCLK	PA00	PIO_008	A7-A	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	
J2		PA01	PIO_009	A6-A		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
J3		P703	PIO_051	ALE-A/ WR2#-B/ BC2#-B	CACREF	GTETRGB/ GTIOC11B	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004		XSPI1_RESET0#/PCKO	IRQ8		
J4		PF06	PIO_054	A23-A/ A19-D		GTCPPO6/ GTIOC9A/ POEGC	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS007#/ SSL02-C		XSPI1_CS1#/ XSPI1_WP0#	IRQ6		
J5		PF07	PIO_055	A22-A/ A18-D		GTCPPO11/GTIOC9B	SSL03-C		XSPI1_RSTO0#	IRQ7		
J6	VSS											
J7	VCL											
J8	VSS											
J9	VCL											
J10	VSS											
J11		P207	PIO_023	A23-C/ CS4#-C	TMRI3	GTOWLO/ GTIOC2B/ GTIOC6B	RXD001-B/ SMISO001-B/ SSCL001/ MISO1-B/ SDAHS2-B/ CRX1/ USB0_ID/ USBHS_OVRCURB-DS	ET0_RXD3-B	PCKO	IRQ3-DS	ADTRG2#	
J12		P305								NMI/IRQ0		
J13	AVCC_USBHS											

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (10/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
J14	XTAL	P307										
J15	EXTAL	P306										
K1		P707	PIO_055	WE#-A/ CS7#-A		GTIOC13B	TXD011-B/ SMOSI011-B/ SSDA011/ TXDA011		XSPI1_RESET1#	IRQ7		
K2		P706	PIO_054	CAS#-A/ CS6#-A		GTIOC13A	RXD011-B/ SMISO011-B/ SSCL011		XSPI1_WP1#	IRQ14	ADCNV2	
K3		P702	PIO_050	RD#-A/ DQM3-B		GTETRGA/ GTIOC11A	SCK004-B/ CTS004#/ TXDB004		XSPI1_WP0#/ XSPI1_RESET1#/ HSYNC	IRQ10	ADST2	
K4		P605	PIO_013	CS2#-C/ D18-B		GTIOC6B	RXD009-B/ SMISO009-B/ SSCL009/ SSL02-D		XSPI0_INT0#/ XSPI0_ECS0#	IRQ13		
K5	VSS											
K6	VCC											
K7	VSS											
K8	VCC2											
K9	VCL											
K10	VSS											
K11		P310	PIO_058		RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
K12		P311	PIO_059	BREQ#-B	RTCIC1	GTETRGC/ GTCPPO14/POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1#	TAMPI1
K13	MD/FINED	P315										
K14	VSS											
K15	VSS											
L1		P705	PIO_053	RAS#-A/ CS5#-A		GTIOC12B	SCK011-B/ CTS011#/ TXDB011		XSPI1_RSTO1#	IRQ13	ADCNV1	
L2		P704	PIO_052	SDCS#-A/ CS4#-A		GTIOC12A	CTS004#/ CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011		XSPI1_INT1#/ XSPI1_ECS1#	IRQ12	ADCNV0	
L3		P604	PIO_012	CS3#-C/ D19-B		GTIOC6A	TXD009-B/ SMOSI009-B/ SSDA009/ TXDA009/ SSL01-D		XSPI0_CS1#/ XSPI0_RSTO0#	IRQ4		
L4		P606	PIO_014	CS1#-C/ D17-B		GTETRGA/ GTCPPO4/ GTIOC15A/ POEGC	SCK009-B/ CTS009#/ TXDB009/ SSL03-D/ CTX2		XSPI0_RESET0#	IRQ14		
L5	VCC2											
L6		P904	PIO_004			GTCPPO0/ POEGC	TXD006-B/ SMOSI006-B/ SSDA006/ TXDA006	ET0_COL/ GPTP_PTPOUT0		IRQ12		

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (11/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
L7		P906	PIO_006			GTCPPO2/ POEGA	SCK006-B/ CTS006#/ TXDB006	ET0_MDC-C		IRQ14		
L8		P407	PIO_055							IRQ15-DS	AN007	
L9		P406	PIO_054							IRQ14-DS	AN006	
L10		P215	PIO_055			GTCPPO13	CTS004#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ USBHS_ID			IRQ31		PDMDAT0 /PCINZ0
L11		P209	PIO_049		TOC1	GTETRGB	RXD004-C/ SMISO004-C/ SSCL004/ MOSI1-D			IRQ25		PCINB1
L12		P312	PIO_060	BACK#-B	RTCIC2	GTETRGD/ GTCPPO15/POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2#	TAMPI2
L13	VCC											
L14	XCIN/ EXCIN	P309										
L15	XCOUT	P308										
M1	ETHPHYCLK	P700	PIO_048	SDCLK-A		GTIOC10A	TXD004-B/ SMOSI004-B/ SSDA004/ TXDA004		XSPI1_INT0#/ XSPI1_ECS0#/PIXCLK	IRQ0	ADST0	
M2		P701	PIO_049	CKE-A/ WR3#-B/ BC3#-B		GTIOC10B	RXD004-B/ SMISO004-B/ SSCL004		XSPI1_RSTO0#/ XSPI1_RESET0#/ VSYNC	IRQ1	ADST1	
M3		P607	PIO_015	CS0#-B/ D16-B		GTETRGB/ GTADSM0/ GTIOC15B/ POEGA	CTS008#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ CRX2		XSPI0_RSTO0#	IRQ15		
M4		P603	PIO_011	WE#-B/ D20-B		GTIOC5B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS009#/ MISO0-D		XSPI0_IO7/ PCKO	IRQ3		
M5		P907	PIO_007			GTCPPO3/ POEGD	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ CTS001#	ET0_MDIO-C	XSPI0_WP0#	IRQ15		
M6		P905	PIO_005			GTCPPO1/ POEGB	RXD006-B/ SMISO006-B/ SSCL006	ET0_CRS/ GPTP_PTPOUT1		IRQ13		
M7		P902	PIO_002			GTETRGC/ GTCPPO8	SCK001-C/ CTS001#/ TXDB001		XSPI0_RESET1#	IRQ10	AN034	
M8		P900	PIO_000			GTETRGA/ GTCPPO5	RXD001-C/ SMISO001-C/ SSCL001		XSPI0_INT1#/ XSPI0_ECS1#	IRQ0	AN032	

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (12/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
M9		P405	PIO_053							IRQ13-DS	AN005	
M10		P415	PIO_063							IRQ31	AN015/ DA1	
M11		P212	PIO_052			GTCPPO10	SCK005-C/ CTS005#/ TXDB005/ SSL11-D/ USBHS_OVRCURA			IRQ28	ADTRG0#	
M12		P214	PIO_054			GTCPPO12	TXD005-C/ SMOSI005-C/ SSDA005/ TXDA005/ SSL13-D/ USBHS_EXICEN			IRQ30	ADTRG2#	PDMCLK0 /PCINB0
M13	RES#											
M14	VSS											
M15	VBATT/ VCC											
N1		PE07	PIO_047	A8-B/A20-D	TOC1	GTETRGD/ GTCPPO7/ GTIOC10B/ POEGD			XSPI1_CS0#/PIXD7	IRQ7		
N2		PE04	PIO_044	A11-B/ A23-D		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0#/PIXD4	IRQ12		
N3		PE06	PIO_046	A9-B/A21-D	TIC1	GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS/ PIXD6	IRQ6		
N4		P602	PIO_010	CAS#-B/ D21-B		GTIOC5A	SCK008-B/ CTS008#/ TXDB008/ MOSI0-D		XSPI0_IO6/ HSYNC	IRQ2	ADTRG2#	
N5		P601	PIO_009	RAS#-B/ D22-B		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTPOUT1	XSPI0_IO5/ VSYNC	IRQ1	ADTRG1#	
N6		P600	PIO_008	SDCS#-B/ D23-B		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTPOUT0	XSPI0_IO4/ PIXCLK	IRQ0	ADTRG0#	
N7		P903	PIO_003			GTETRGD/ GTCPPO9/ POEGD	CTS006#/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001		XSPI0_WP1#	IRQ11	AN035	
N8		P901	PIO_001			GTETRGB/ GTCPPO6	TXD001-C/ SMOSI001-C/ SSDA001/ TXDA001		XSPI0_RSTO1#/ XSPI0_RESET0#	IRQ9	AN033	
N9		P404	PIO_052							IRQ12-DS	AN004	
N10		P414	PIO_062							IRQ30	AN014/ DA0	
N11		P210	PIO_050			GTETRGC	TXD004-C/ SMOSI004-C/ SSDA004/ TXDA004/ MISO1-D			IRQ26		PCINZ1

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (13/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
N12		P213	PIO_053			GTCPPO11	RXD005-C/ SMISO005-C/ SSCL005/ SSL12-D/ USBHS_VBUSEN			IRQ29	ADTRG1#	PCINA0
N13		P208	PIO_048		TIC1	GTETRGA	SCK004-C/ CTS004#/ TXDB004/ RSPCK1-D			IRQ24		PCINA1
N14		P211	PIO_051			GTETRGD	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS005#/ SSL10-D/ USBHS_OVRCURB			IRQ27		
N15		P000	PIO_000		TMRI0	GTCPPO7/ GTIOC10A/ POEGB	CTS000#/ CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ AUDIO_CLK	ET0_INT		IRQ16-DS	AN016	
P1		PE05	PIO_045	A10-B/ A22-D		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3/ PIXD5	IRQ5		
P2		PE03	PIO_043	A12-B	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/ XSPI1_IO1/ PIXD3	IRQ11		
P3		PE02	PIO_042	A13-B	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/ XSPI1_IO0/ PIXD2	IRQ7-DS		
P4		PD07	PIO_039	A16-B/ D24-B		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0#/ XSPI0_RESET1#/PIXD7	IRQ7	AN031	
P5		PD05	PIO_037	A18-B/ D26-B		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1-B/ XSPI0_IO3/ PIXD5	IRQ5	AN029	
P6		PD03	PIO_035	A20-B/ D28-B	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK-B/ XSPI0_IO1/ PIXD3	IRQ3	AN027	
P7		PD02	PIO_034	A21-B/ D29-B	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD-B/ XSPI0_IO0/ PIXD2	IRQ2	AN026	
P8	AVSS0											
P9	VREFL0											
P10		P403	PIO_051							IRQ11-DS	AN003	
P11		P401	PIO_049							IRQ9-DS	AN001	
P12		P006	PIO_006			GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1#/AN022	PDMCLK0/ PCOUTZ1

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (14/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
P13		P004	PIO_004			GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1
P14		P003	PIO_003			GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0
P15		P001	PIO_001		TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
R1	VSS											
R2	CLKOUT	PE01	PIO_041	A14-B		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/ XSPI1_CK/ PIXD1	IRQ9		
R3	ETHPHYCLK	PE00	PIO_040	A15-B		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN#/ XSPI1_CS1#/PIXD0	IRQ8		
R4		PD06	PIO_038	A17-B/ D25-B		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS/ PIXD6	IRQ6	AN030	
R5		PD04	PIO_036	A19-B/ D27-B		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0-B/ XSPI0_IO2/ XSPI0_WP0#/PIXD4	IRQ4	AN028	
R6		PD00	PIO_032	A23-B/ D31-B		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2-B/ XSPI0_CKN#/ XSPI0_CS1#/PIXD0	IRQ0	AN024	
R7		PD01	PIO_033	A22-B/ D30-B		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3-B/ XSPI0_CK/ PIXD1	IRQ1	AN025	
R8	AVCC0											
R9	VREFH0											
R10		P402	PIO_050							IRQ10-DS	AN002	
R11		P400	PIO_048							IRQ8-DS	AN000	
R12		P007	PIO_007			GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0#/AN023	PDMDAT0
R13		P005	PIO_005			GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2#/AN021	PCOUTB1

Table 1.6 List of Pin and Pin Functions (224-Pin LFBGA) (15/15)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
224-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
R14		P002	PIO_002		TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
R15	NC											

1.6.3 176-Pin LFBGA

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (1/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A1		PC01	PIO_025	D6-A		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A/ RGMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
A2		PC03	PIO_027	D4-A		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A/ RGMII0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
A3		PC05	PIO_029	D2-A	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_EN-A/ RGMII0_TX_CTL-A	SD0_DAT1-A	IRQ5		PCOUTZ1
A4	ETHPHYCLK	PC07	PIO_031	D0-A	TMO2/ TOC0/ CACREF	GTETRGD/ GTADSM1/ GTIOC15B/ POEGA	TXD008-A/ SMOSI008-A/ SSDA008/ TXDA008/ MISO0-A/ SSITXD0	RGMII0_TXC-A	SD0_CD	IRQ14		PDMDAT0
A5	CLKOUT	P801	PIO_057	EBCLK-A		GTIV/ GTIOC0B/ POEGC	TXD010-B/ SMOSI010-B/ SSDA010/ TXDA010	ET0_CRS		IRQ9	ADCNV1	
A6		P503	PIO_035	WR3#-A/ BC3#-A/ A19-C		GTIOC3B	RXD002-C/ SMISO002-C/ SSCL002/ CRX1/ USB0_OVRCURB/ SSIRXD0		SD0_DAT7-A	IRQ3		
A7	VLO											
A8	VSS_DCDC											
A9	VCC_DCDC											
A10		P102	PIO_010		TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
A11	CLKOUT	P106	PIO_014		TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C/PIXD2	IRQ30-DS	ADTRG0#	PCINA1
A12		P200	PIO_016		TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_EN-B	SD0_CMD-C/PIXD4	IRQ8	ADTRG2#	PCINZ1
A13		P805	PIO_061			GTADSM1/ GTIOC15B	SCK005-B/ CTS005#/ TXDB005/ SSL22-B/ CRX0/ USB0_VBUSEN/ USBHS_VBUSEN	ET0_TXD0-C/ RMII0_TXD0-C	HSYNC	IRQ5	ADSYNC	PCINB0

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (2/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
A14	TCK/ SWCLK	P301	PIO_025		TMCI3	GTCPPO11	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		
A15	TDO	P302	PIO_026		TMRI3	GTCPPO12	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
B1		PC00	PIO_024	D7-A		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
B2		PC02	PIO_026	D5-A		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD1-A/ RGMII0_TXD1-A	SD0_CMD-A	IRQ10		PCOUTZ0
B3		PC04	PIO_028	D3-A	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A/ RGMII0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
B4		PC06	PIO_030	D1-A	TMCI2/ TIC0	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER-A	SD0_WP	IRQ13		PDMCLK0
B5		P803	PIO_059	CS3#-A/ WR1#-B/ BC1#-B		GTIOC1B/ POEGA	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTX2/ USB0_ID	ET0_MDC-B		IRQ3	ADSYNC	
B6		P502	PIO_034	WR2#-A/ BC2#-A/ A18-C		GTIOC3A	TXD002-C/ SMOSI002-C/ SSDA002/ TXDA002/ SSL13-C/ CTX1	ET0_WOL	SD0_DAT6-A	IRQ2	ADST2	
B7	VLO											
B8	VSS_DCDC											
B9	VCC_DCDC											
B10		P103	PIO_011		TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		
B11		P107	PIO_015		TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C/PIXD3	IRQ31-DS	ADTRG1#	PCINB1

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (3/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
B12		P201	PIO_017		TMCI0	GTIW/ GTIOC3B/ GTIOC14B	SCK006-A/ CTS006#/ TXDB006/ MISO0-C/ SCLHS1-A/ USB0_EXICEN/SSILRCK0	ET0_RX_ER-B/ RMII0_RX_ER-B	SD0_CLK-C/PIXD5	IRQ9		PCINZ0
B13		P804	PIO_060			GTADSM0/ GTIOC15A	CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL21-B/ CTX0/ USB0_OVRCURA/ USBHS_OVRCURA	ET0_TX_EN-C/ RMII0_TX_EN-C	PCKO	IRQ4		PCINA0
B14	TMS/ SWDIO	P300	PIO_024		TMO1	GTCPPO10	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
B15	TDI	P303	PIO_027		TMCI2	GTCPPO13	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		
C1		PB06	PIO_022	D9-A		GTOWUP/ GTIOC6A	RXD009-A/ SMISO009-A/ SSCL009	ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
C2		PB07	PIO_023	D8-A		GTOWLO/ GTIOC6B	TXD009-A/ SMOSI009-A/ SSDA009/ TXDA009	ET0_LINKSTA		IRQ15		
C3		P802	PIO_058	CS2#-A/ WR#-B/ WR0#-B		GTIW/ GTIOC1A/ POEGB	SCK010-B/ CTS010#/ TXDB010/ CRX2/ USB0_EXICEN	ET0_MDIO-B		IRQ2	ADCNV2	
C4	VCC											
C5		P500	PIO_032	WR#-A/ WR0#-A/ A16-C		GTIOC2A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS007#/ SSL11-C/ CTX0	ET0_LINKSTA	SD0_DAT4-A	IRQ0	ADST0	
C6		P501	PIO_033	WR1#-A/ BC1#-A/ A17-C		GTIOC2B	SCK002-C/ CTS002#/ TXDB002/ SSL12-C/ CRX0	ET0_INT	SD0_DAT5-A	IRQ1	ADST1	
C7	VLO											
C8	VCC_DCDC											
C9	VCC											

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (4/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
C10		P100	PIO_008	CS1#-B/ WAIT#-A	TMRI3/ CACREF	GTIU/ GTADSM0/ GTIOC12A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS005#/ USB0_OVRCURB-DS/ USBHS_VBUS	ET0_TX_ER-B		IRQ0	ADTRG2#	
C11		P104	PIO_012		TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRCURA-DS/ USBHS_VBUS	ET0_TXD0-B/ RMII0_TXD0-B	PIXD0	IRQ4	ADST0	PCINA0
C12	CLKOUT	P205	PIO_021	A21-C/ CS6#-C		GTOVLO/ GTIOC1B/ GTIOC5B	SCK001-B/ CTS001#/ TXDB001/ RSPCK1-B/ SDAHS1-B/ CRX0/ USB0_VBUS/ USBHS_VBUSEN/ SSIRXD1	ET0_RXD1-B/ RMII0_RXD1-B	SD0_CD/ HSYNC	IRQ5	ADTRG0#	PCOUTB1
C13	ETHPHYCLK	P304	PIO_028	ALE-C/ WAIT#-B	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUSEN/ USBHS_ID	ET0_INT		IRQ0-DS	ADTRG0#	
C14		P202	PIO_018	CS3#-B/ BREQ#-A	TMO0	GTOUUP/ GTIOC0A/ GTIOC4A	TXD006-A/ SMOSI006-A/ SSDA006/ TXDA006/ RSPCK0-C/ SCL00-B/ CTX2/ USB0_OVRCURA/ USBHS_EXICEN/ AUDIO_CLK	ET0_RX_DV-B/ RMII0_CRS_DV-B	SD0_DAT0-C/PIXD6	IRQ15		PDMCLK0
C15		P203	PIO_019	CS2#-B/ BACK#-A		GTOULO/ GTIOC0B/ GTIOC4B	RXD006-A/ SMISO006-A/ SSCL006/ CTS003#/ SSL00-C/ SDA00-B/ CRX2/ USB0_OVRCURB/ USBHS_ID/ SSIBCK0	ET0_RX_CLK-B/ RMII0_REF50CK-B	SD0_DAT1-C/PIXD7	IRQ3		PDMDAT0
D1		PB04	PIO_020	D11-A		GTOVUP/ GTIOC5A	CTS004#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009	ET0_RX_CLK-A/ RMII0_REF50CK-A/ RGMII0_RXC-A		IRQ4		

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (5/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
D2		PB05	PIO_021	D10-A	TMRI1	GTOVLO/ GTIOC5B	SCK009-A/ CTS009#/ TXDB009	ET0_RX_DV-A/ RMII0_CRS_DV-A/ RGMII0_RX_CTL-A		IRQ13		
D3		PG01	PIO_057	DQM0-A/ WR#-C/ WR0#-C		GTCPPO1	RXD008-C/ SMISO008-C/ SSCL008			IRQ9		
D4		P800	PIO_056	CS1#-A/ DQM0-B		GTIU/ GTIOC0A/ POEGD	RXD010-B/ SMISO010-B/ SSCL010	ET0_COL		IRQ8	ADCNV0	
D5	VCC											
D6	VSS											
D7	VCL											
D8		P101	PIO_009	CS0#-A/ ALE-B	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUSDS/ USBHS_EXICEN	ET0_TX_CLK-B		IRQ1	ADTRG1#	
D9	VSS											
D10	VCL											
D11		P105	PIO_013		TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ USBHS_OVRCURB/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B	PIXD1	IRQ29-DS	ADST1	PCINB0
D12	VCC											
D13	ETHPHYCLK	P204	PIO_020	A20-C/ CS7#-C	TMRI1	GTOVUP/ GTIOC1A/ GTIOC5A	CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-B/ SCLHS1-B/ CTX0/ USB0_VBUSEN/ USBHS_VBUS-DS/ SSIBCK1	ET0_RXD0-B/ RMII0_RXD0-B	SD0_WP/ PIXCLK	IRQ12		PCOUTA1
D14		PF00	PIO_048			GTCPPO14	USB0_DP			IRQ0		
D15		PF01	PIO_049			GTCPPO15	USB0_DM			IRQ1		
E1		PB02	PIO_018	D13-A		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS009#	ET0_RXD2-A/ RGMII0_RXD2-A		IRQ2		
E2		PB03	PIO_019	D12-A	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A/ RGMII0_RXD3-A		IRQ3		

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (6/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
E3		PG00	PIO_056	DQM1-A/ WR1#-C/ BC1#-C		GTCPPO0	SCK008-C/ CTS008#/ TXDB008			IRQ8		
E4	VSS											
E12	VSS											
E13		P206	PIO_022	A22-C/ CS5#-C	TMO3/ RTCOUT	GTOWUP/ GTIOC2A/ GTIOC6A	CTS006#/ TXD001-B/ SMOSI001-B/ SSDA001/ TXDA001/ MOSI1-B/ SCLHS2-B/ CTX1/ USB0_EXICEN/ USBHS_OVRCURA-DS/ SSITXD1	ET0_RXD2-B	VSYNC	IRQ2-DS	ADTRG1#	PCOUTZ1
E14	VSS_USB											
E15	VCC_USB											
F1		PB00	PIO_016	D15-A		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A/ RGMII0_RXD0-A		IRQ12	ADST2	PCINA1
F2		PB01	PIO_017	D14-A	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A/ RGMII0_RXD1-A		IRQ4-DS		PCINB1
F3		PG02	PIO_058	DQM3-A/ WR3#-C/ BC3#-C		GTIOC11A	TXD008-C/ SMOSI008-C/ SSDA008/ TXDA008			IRQ10		
F4	VCL											
F12	VCL											
F13	AVSS_USBHS											
F14							USBHS_DP					
F15							USBHS_DM					
G1		PA06	PIO_014	A1-A/ DQM2-B	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
G2	ETHPHYCLK	PA07	PIO_015	A0-A/ BC0#-A/ DQM1-B		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0
G3		P706	PIO_054	CAS#-A/ CS6#-A		GTIOC13A	RXD011-B/ SMISO011-B/ SSCL011		XSPI1_WP1#	IRQ14	ADCNV2	
G4	VCC											
G12	AVCC_USBHS											
G13	USBHS_REF											
G14	VSS_USBHS											

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (7/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
G15	VCC_USBHS											
H1		PA04	PIO_012	A3-A	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
H2		PA05	PIO_013	A2-A		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0
H3		P707	PIO_055	WE#-A/ CS7#-A		GTIOC13B	TXD011-B/ SMOSI011-B/ SSDA011/ TXDA011		XSPI1_RESET1#	IRQ7		
H4	VSS											
H12		P207	PIO_023	A23-C/ CS4#-C	TMRI3	GTOWLO/ GTIOC2B/ GTIOC6B	RXD001-B/ SMISO001-B/ SSCL001/ MISO1-B/ SDAHS2-B/ CRX1/ USB0_ID/ USBHS_OVRCURB-DS	ET0_RXD3-B	PCKO	IRQ3-DS	ADTRG2#	
H13	VSS											
H14	EXTAL	P306										
H15	XTAL	P307										
J1		PA02	PIO_010	A5-A		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
J2		PA03	PIO_011	A4-A		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
J3		P705	PIO_053	RAS#-A/ CS5#-A		GTIOC12B	SCK011-B/ CTS011#/ TXDB011		XSPI1_RSTO1#	IRQ13	ADCNV1	
J4	VCL											
J12	VCL											
J13		P305								NMI/IRQ0		
J14	VSS											
J15	VCC											
K1	ETHPHYCLK	PA00	PIO_008	A7-A	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	
K2		PA01	PIO_009	A6-A		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
K3		P702	PIO_050	RD#-A/ DQM3-B		GTETRGA/ GTIOC11A	SCK004-B/ CTS004#/ TXDB004		XSPI1_WP0#/ XSPI1_RESET1#/ HSYNC	IRQ10	ADST2	
K4	VCC											

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (8/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
K12	VSS											
K13		P310	PIO_058		RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
K14	XCIN/ EXCIN	P309										
K15	XCOUT	P308										
L1	ETHPHYCLK	P700	PIO_048	SDCLK-A		GTIOC10A	TXD004-B/ SMOSI004-B/ SSDA004/ TXDA004		XSPI1_INT0#/ XSPI1_ECS0#/PIXCLK	IRQ0	ADST0	
L2		P701	PIO_049	CKE-A/ WR3#-B/ BC3#-B		GTIOC10B	RXD004-B/ SMISO004-B/ SSCL004		XSPI1_RSTO0#/ XSPI1_RESET0#/ VSYNC	IRQ1	ADST1	
L3		P703	PIO_051	ALE-A/ WR2#-B/ BC2#-B	CACREF	GTETRGB/ GTIOC11B	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004		XSPI1_RESET0#/PCKO	IRQ8		
L4		P704	PIO_052	SDCS#-A/ CS4#-A		GTIOC12A	CTS004#/ CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011		XSPI1_INT1#/ XSPI1_ECS1#	IRQ12	ADCNV0	
L12		P312	PIO_060	BACK#-B	RTCIC2	GTETRGD/ GTCPPO15/POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2#	TAMPI2
L13		P311	PIO_059	BREQ#-B	RTCIC1	GTETRGC/ GTCPPO14/POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1#	TAMPI1
L14	VCC											
L15	VBATT/ VCC											
M1		PE07	PIO_047	A8-B/A20-D	TOC1	GTETRGD/ GTCPPO7/ GTIOC10B/ POEGD			XSPI1_CS0#/PIXD7	IRQ7		
M2		PE03	PIO_043	A12-B	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/ XSPI1_IO1/ PIXD3	IRQ11		
M3		PD04	PIO_036	A19-B/ D27-B		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0-B/ XSPI0_IO2/ XSPI0_WP0#/PIXD4	IRQ4	AN028	
M4		PD06	PIO_038	A17-B/ D25-B		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS/ PIXD6	IRQ6	AN030	
M5	VSS											
M6	VCL											
M7	VCC2											

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (9/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
M8	VSS											
M9	VCL											
M10		P407	PIO_055							IRQ15-DS	AN007	
M11		P404	PIO_052							IRQ12-DS	AN004	
M12	MD/FINED	P315										
M13	RES#											
M14		P001	PIO_001		TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
M15		P000	PIO_000		TMRI0	GTCPPO7/ GTIOC10A/ POEGB	CTS000#/ CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ AUDIO_CLK	ET0_INT		IRQ16-DS	AN016	
N1		PE05	PIO_045	A10-B/ A22-D		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3/ PIXD5	IRQ5		
N2		PE04	PIO_044	A11-B/ A23-D		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0#/PIXD4	IRQ12		
N3		P607	PIO_015	CS0#-B/ D16-B		GTETRGB/ GTADSM0/ GTIOC15B/ POEGA	CTS008#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ CRX2		XSPI0_RSTO0#	IRQ15		
N4		P604	PIO_012	CS3#-C/ D19-B		GTIOC6A	TXD009-B/ SMOSI009-B/ SSDA009/ TXDA009/ SSL01-D		XSPI0_CS1#/ XSPI0_RSTO0#	IRQ4		
N5		PD07	PIO_039	A16-B/ D24-B		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0#/ XSPI0_RESET1#/PIXD7	IRQ7	AN031	
N6		PD05	PIO_037	A18-B/ D26-B		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1-B/ XSPI0_IO3/ PIXD5	IRQ5	AN029	
N7	VCC2											
N8	VSS											
N9	VCL											
N10		P406	PIO_054							IRQ14-DS	AN006	
N11		P405	PIO_053							IRQ13-DS	AN005	
N12		P415	PIO_063							IRQ31	AN015/ DA1	
N13		P414	PIO_062							IRQ30	AN014/ DA0	
N14		P003	PIO_003			GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (10/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
N15		P002	PIO_002		TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
P1	CLKOUT	PE01	PIO_041	A14-B		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/ XSPI1_CK/ PIXD1	IRQ9		
P2		PE02	PIO_042	A13-B	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/ XSPI1_IO0/ PIXD2	IRQ7-DS		
P3		P606	PIO_014	CS1#-C/ D17-B		GTETRGA/ GTCPPO4/ GTIOC15A/ POEGC	SCK009-B/ CTS009#/ TXDB009/ SSL03-D/ CTX2		XSPI0_RESET0#	IRQ14		
P4		P603	PIO_011	WE#-B/ D20-B		GTIOC5B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS009#/ MISO0-D		XSPI0_IO7/ PCKO	IRQ3		
P5		P601	PIO_009	RAS#-B/ D22-B		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTPOUT1	XSPI0_IO5/ VSYNC	IRQ1	ADTRG1#	
P6		PD03	PIO_035	A20-B/ D28-B	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK-B/ XSPI0_IO1/ PIXD3	IRQ3	AN027	
P7		PD00	PIO_032	A23-B/ D31-B		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2-B/ XSPI0_CKN#/ XSPI0_CS1#/PIXD0	IRQ0	AN024	
P8		P902	PIO_002			GTETRGC/ GTCPPO8	SCK001-C/ CTS001#/ TXDB001		XSPI0_RESET1#	IRQ10	AN034	
P9		P900	PIO_000			GTETRGA/ GTCPPO5	RXD001-C/ SMISO001-C/ SSCL001		XSPI0_INT1#/ XSPI0_ECS1#	IRQ0	AN032	
P10	AVSS0											
P11	VREFL0											
P12		P402	PIO_050							IRQ10-DS	AN002	
P13		P400	PIO_048							IRQ8-DS	AN000	
P14		P005	PIO_005			GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2#/AN021	PCOUTB1
P15		P004	PIO_004			GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1

Table 1.7 List of Pin and Pin Functions (176-Pin LFBGA) (11/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFBGA	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
R1	ETHPHYCLK	PE00	PIO_040	A15-B		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN#/ XSPI1_CS1#/PIXD0	IRQ8		
R2		PE06	PIO_046	A9-B/A21-D	TIC1	GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS/ PIXD6	IRQ6		
R3		P605	PIO_013	CS2#-C/ D18-B		GTIOC6B	RXD009-B/ SMISO009-B/ SSCL009/ SSL02-D		XSPI0_INT0#/ XSPI0_ECS0#	IRQ13		
R4		P602	PIO_010	CAS#-B/ D21-B		GTIOC5A	SCK008-B/ CTS008#/ TXDB008/ MOSI0-D		XSPI0_IO6/ HSYNC	IRQ2	ADTRG2#	
R5		P600	PIO_008	SDCS#-B/ D23-B		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTPOUT0	XSPI0_IO4/ PIXCLK	IRQ0	ADTRG0#	
R6		PD02	PIO_034	A21-B/ D29-B	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD-B/ XSPI0_IO0/ PIXD2	IRQ2	AN026	
R7		PD01	PIO_033	A22-B/ D30-B		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3-B/ XSPI0_CK/ PIXD1	IRQ1	AN025	
R8		P903	PIO_003			GTETRGD/ GTCPPO9/ POEGD	CTS006#/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001		XSPI0_WP1#	IRQ11	AN035	
R9		P901	PIO_001			GTETRGB/ GTCPPO6	TXD001-C/ SMOSI001-C/ SSDA001/ TXDA001		XSPI0_RSTO1#/ XSPI0_RESET0#	IRQ9	AN033	
R10	AVCC0											
R11	VREFH0											
R12		P403	PIO_051							IRQ11-DS	AN003	
R13		P401	PIO_049							IRQ9-DS	AN001	
R14		P007	PIO_007			GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0#/AN023	PDMDAT0
R15		P006	PIO_006			GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1#/AN022	PDMCLK0/ PCOUTZ1

1.6.4 176-Pin LFQFP

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (1/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
1		P003	PIO_003			GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0
2		P002	PIO_002		TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
3		P001	PIO_001		TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
4		P000	PIO_000		TMRI0	GTCPPO7/ GTIOC10A/ POEGB	CTS000#/ CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ AUDIO_CLK	ET0_INT		IRQ16-DS	AN016	
5	VCC											
6	VSS											
7	VSS											
8	VCL											
9	RES#											
10	MD/FINED	P315										
11		P312	PIO_060	BACK#-B	RTCIC2	GTETRGD/ GTCPPO15/POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2#	TAMPI2
12		P311	PIO_059	BREQ#-B	RTCIC1	GTETRGC/ GTCPPO14/POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1#	TAMPI1
13		P310	PIO_058		RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
14	VBATT/ VCC											
15	XCIN/ EXCIN	P309										
16	XCOUT	P308										
17	VSS											
18	XTAL	P307										
19	EXTAL	P306										
20	VCC											
21		P305								NMI/IRQ0		
22	AVCC_USBHS											
23	USBHS_REF											
24	AVSS_USBHS											
25							USBHS_DM					
26							USBHS_DP					

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (2/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
27	VSS_USBHS											
28	VCC_USBHS											
29	VCC_USB											
30		PF01	PIO_049			GTCPPO15	USB0_DM			IRQ1		
31		PF00	PIO_048			GTCPPO14	USB0_DP			IRQ0		
32	VSS_USB											
33		P207	PIO_023	A23-C/ CS4#-C	TMRI3	GTOWLO/ GTIOC2B/ GTIOC6B	RXD001-B/ SMISO001-B/ SSCL001/ MISO1-B/ SDAHS2-B/ CRX1/ USB0_ID/ USBHS_OVRCURB-DS	ET0_RXD3-B	PCKO	IRQ3-DS	ADTRG2#	
34		P206	PIO_022	A22-C/ CS5#-C	TMO3/ RTCOUT	GTOWUP/ GTIOC2A/ GTIOC6A	CTS006#/ TXD001-B/ SMOSI001-B/ SSDA001/ TXDA001/ MOSI1-B/ SCLHS2-B/ CTX1/ USB0_EXICEN/ USBHS_OVRCURA-DS/ SSITXD1	ET0_RXD2-B	VSYNC	IRQ2-DS	ADTRG1#	PCOUTZ1
35	CLKOUT	P205	PIO_021	A21-C/ CS6#-C		GTOVLO/ GTIOC1B/ GTIOC5B	SCK001-B/ CTS001#/ TXDB001/ RSPCK1-B/ SDAHS1-B/ CRX0/ USB0_VBUS/ USBHS_VBUSEN/ SSIRXD1	ET0_RXD1-B/ RMII0_RXD1-B	SD0_CD/ HSYNC	IRQ5	ADTRG0#	PCOUTB1
36	ETHPHYCLK	P204	PIO_020	A20-C/ CS7#-C	TMRI1	GTOVUP/ GTIOC1A/ GTIOC5A	CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-B/ SCLHS1-B/ CTX0/ USB0_VBUSEN/ USBHS_VBUS-DS/ SSIBCK1	ET0_RXD0-B/ RMII0_RXD0-B	SD0_WP/ PIXCLK	IRQ12		PCOUTA1
37	VCL											
38	VSS											
39	VCC											

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (3/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
40		P203	PIO_019	CS2#-B/ BACK#-A		GTOULO/ GTIOC0B/ GTIOC4B	RXD006-A/ SMISO006-A/ SSCL006/ CTS003#/ SSL00-C/ SDA00-B/ CRX2/ USB0_OVRC URB/ USBHS_ID/ SSIBCK0	ET0_RX_CL K-B/ RMII0_REF 50CK-B	SD0_DAT1- C/PIXD7	IRQ3		PDMDAT0
41		P202	PIO_018	CS3#-B/ BREQ#-A	TMO0	GTOUUP/ GTIOC0A/ GTIOC4A	TXD006-A/ SMOSI006-A/ SSDA006/ TXDA006/ RSPCK0-C/ SCL00-B/ CTX2/ USB0_OVRC URA/ USBHS_EXIC EN/ AUDIO_CLK	ET0_RX_D V-B/ RMII0_CRS _DV-B	SD0_DAT0- C/PIXD6	IRQ15		PDMCLK0
42	ETHPHYC LK	P304	PIO_028	ALE-C/ WAIT#-B	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUS EN/ USBHS_ID	ET0_INT		IRQ0-DS	ADTRG0 #	
43	TDI	P303	PIO_027		TMCI2	GTCPPO1 3	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		
44	TDO	P302	PIO_026		TMRI3	GTCPPO1 2	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
45	TCK/ SWCLK	P301	PIO_025		TMCI3	GTCPPO1 1	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		
46	TMS/ SWDIO	P300	PIO_024		TMO1	GTCPPO1 0	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
47		P805	PIO_061			GTADSM1/ GTIOC15B	SCK005-B/ CTS005#/ TXDB005/ SSL22-B/ CRX0/ USB0_VBUS EN/ USBHS_VBU SEN	ET0_TXD0- C/ RMII0_TXD 0-C	HSYNC	IRQ5	ADSYNC	PCINB0

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (4/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
48		P804	PIO_060			GTADSM0/ GTIOC15A	CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL21-B/ CTX0/ USB0_OVRCURA/ USBHS_OVRCURA	ET0_TX_EN-C/ RMII0_TX_EN-C	PCKO	IRQ4		PCINA0
49		P201	PIO_017		TMCI0	GTIW/ GTIOC3B/ GTIOC14B	SCK006-A/ CTS006#/ TXDB006/ MISO0-C/ SCLHS1-A/ USB0_EXICEN/SSILRCK0	ET0_RX_ER-B/ RMII0_RX_ER-B	SD0_CLK-C/PIXD5	IRQ9		PCINZ0
50		P200	PIO_016		TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_EN-B	SD0_CMD-C/PIXD4	IRQ8	ADTRG2#	PCINZ1
51		P107	PIO_015		TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C/PIXD3	IRQ31-DS	ADTRG1#	PCINB1
52	CLKOUT	P106	PIO_014		TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C/PIXD2	IRQ30-DS	ADTRG0#	PCINA1
53		P105	PIO_013		TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ USBHS_OVRCURB/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B	PIXD1	IRQ29-DS	ADST1	PCINB0
54		P104	PIO_012		TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRCURA-DS/ USBHS_VBUS	ET0_TXD0-B/ RMII0_TXD0-B	PIXD0	IRQ4	ADST0	PCINA0
55	VCL											
56	VSS											
57	VCC											
58		P103	PIO_011		TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (5/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
59		P102	PIO_010		TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
60		P101	PIO_009	CS0#-A/ ALE-B	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUS-DS/ USBHS_EXICEN	ET0_TX_CLK-B		IRQ1	ADTRG1#	
61		P100	PIO_008	CS1#-B/ WAIT#-A	TMRI3/ CACREF	GTIU/ GTADSM0/ GTIOC12A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS005#/ USB0_OVRCURB-DS/ USBHS_VBUS	ET0_TX_ER-B		IRQ0	ADTRG2#	
62	VCC_DCDC											
63	VCC_DCDC											
64	VCC_DCDC											
65	VLO											
66	VLO											
67	VLO											
68	VSS_DCDC											
69	VSS_DCDC											
70		P503	PIO_035	WR3#-A/ BC3#-A/ A19-C		GTIOC3B	RXD002-C/ SMISO002-C/ SSCL002/ CRX1/ USB0_OVRCURB/ SSIRXD0		SD0_DAT7-A	IRQ3		
71		P502	PIO_034	WR2#-A/ BC2#-A/ A18-C		GTIOC3A	TXD002-C/ SMOSI002-C/ SSDA002/ TXDA002/ SSL13-C/ CTX1	ET0_WOL	SD0_DAT6-A	IRQ2	ADST2	
72		P501	PIO_033	WR1#-A/ BC1#-A/ A17-C		GTIOC2B	SCK002-C/ CTS002#/ TXDB002/ SSL12-C/ CRX0	ET0_INT	SD0_DAT5-A	IRQ1	ADST1	
73		P500	PIO_032	WR#-A/ WR0#-A/ A16-C		GTIOC2A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS007#/ SSL11-C/ CTX0	ET0_LINKSTA	SD0_DAT4-A	IRQ0	ADST0	
74		P803	PIO_059	CS3#-A/ WR1#-B/ BC1#-B		GTIOC1B/ POEGA	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTX2/ USB0_ID	ET0_MDC-B		IRQ3	ADSYNC	

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (6/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
75		P802	PIO_058	CS2#-A/ WR#-B/ WR0#-B		GTIW/ GTIOC1A/ POEGB	SCK010-B/ CTS010#/ TXDB010/ CRX2/ USB0_EXICEN	ET0_MDIO-B		IRQ2	ADCNV2	
76	VCL											
77	VSS											
78	VCC											
79	CLKOUT	P801	PIO_057	EBCLK-A		GTIV/ GTIOC0B/ POEGC	TXD010-B/ SMOSI010-B/ SSDA010/ TXDA010	ET0_CRS		IRQ9	ADCNV1	
80		P800	PIO_056	CS1#-A/ DQM0-B		GTIU/ GTIOC0A/ POEGD	RXD010-B/ SMISO010-B/ SSCL010	ET0_COL		IRQ8	ADCNV0	
81	ETHPHYCLK	PC07	PIO_031	D0-A	TMO2/ TOC0/ CACREF	GTETRGD/ GTADSM1/ GTIOC15B/ POEGA	TXD008-A/ SMOSI008-A/ SSDA008/ TXDA008/ MISO0-A/ SSITXD0		SD0_CD	IRQ14		PDMDAT0
82		PC06	PIO_030	D1-A	TMCI2/ TIC0	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER-A	SD0_WP	IRQ13		PDMCLK0
83		PC05	PIO_029	D2-A	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_EN-A	SD0_DAT1-A	IRQ5		PCOUTZ1
84		PC04	PIO_028	D3-A	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
85		PC03	PIO_027	D4-A		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
86		PC02	PIO_026	D5-A		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD1-A	SD0_CMD-A	IRQ10		PCOUTZ0
87		PC01	PIO_025	D6-A		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
88		PC00	PIO_024	D7-A		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
89		PB07	PIO_023	D8-A		GTOWLO/ GTIOC6B	TXD009-A/ SMOSI009-A/ SSDA009/ TXDA009	ET0_LINKSTA		IRQ15		

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (7/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
90		PB06	PIO_022	D9-A		GTOWUP/ GTIOC6A	RXD009-A/ SMISO009-A/ SSCL009	ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
91		PB05	PIO_021	D10-A	TMRI1	GTOVLO/ GTIOC5B	SCK009-A/ CTS009#/ TXDB009	ET0_RX_DV-A/ RMII0_CRS_DV-A		IRQ13		
92		PB04	PIO_020	D11-A		GTOVUP/ GTIOC5A	CTS004#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009	ET0_RX_CLK-A/ RMII0_REF50CK-A		IRQ4		
93	VCC											
94	VSS											
95	VCL											
96		PB03	PIO_019	D12-A	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A		IRQ3		
97		PB02	PIO_018	D13-A		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS009#	ET0_RXD2-A		IRQ2		
98		PB01	PIO_017	D14-A	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A		IRQ4-DS		PCINB1
99		PB00	PIO_016	D15-A		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A		IRQ12	ADST2	PCINA1
100		PG02	PIO_058	DQM3-A/ WR3#-C/ BC3#-C		GTIOC11A	TXD008-C/ SMOSI008-C/ SSDA008/ TXDA008			IRQ10		
101		PG01	PIO_057	DQM0-A/ WR#-C/ WR0#-C		GTCPPO1	RXD008-C/ SMISO008-C/ SSCL008			IRQ9		
102		PG00	PIO_056	DQM1-A/ WR1#-C/ BC1#-C		GTCPPO0	SCK008-C/ CTS008#/ TXDB008			IRQ8		
103	ETHPHYCLK	PA07	PIO_015	A0-A/ BC0#-A/ DQM1-B		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0
104		PA06	PIO_014	A1-A/ DQM2-B	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
105		PA05	PIO_013	A2-A		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0
106		PA04	PIO_012	A3-A	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
107	VCC											

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (8/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
108	VSS											
109	VCL											
110		PA03	PIO_011	A4-A		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
111		PA02	PIO_010	A5-A		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
112		PA01	PIO_009	A6-A		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
113	ETHPHYCLK	PA00	PIO_008	A7-A	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	
114		P707	PIO_055	WE#-A/ CS7#-A		GTIOC13B	TXD011-B/ SMOSI011-B/ SSDA011/ TXDA011		XSPI1_RESET1#	IRQ7		
115		P706	PIO_054	CAS#-A/ CS6#-A		GTIOC13A	RXD011-B/ SMISO011-B/ SSCL011		XSPI1_WP1#	IRQ14	ADCNV2	
116		P705	PIO_053	RAS#-A/ CS5#-A		GTIOC12B	SCK011-B/ CTS011#/ TXDB011		XSPI1_RSTO1#	IRQ13	ADCNV1	
117		P704	PIO_052	SDCS#-A/ CS4#-A		GTIOC12A	CTS004#/ CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011		XSPI1_INT1#/ XSPI1_ECS1#	IRQ12	ADCNV0	
118		P703	PIO_051	ALE-A/ WR2#-B/ BC2#-B	CACREF	GTETRGB/ GTIOC11B	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004		XSPI1_RESET0#/PCKO	IRQ8		
119		P702	PIO_050	RD#-A/ DQM3-B		GTETRGA/ GTIOC11A	SCK004-B/ CTS004#/ TXDB004		XSPI1_WP0#/ XSPI1_RESET1#/ HSYNC	IRQ10	ADST2	
120		P701	PIO_049	CKE-A/ WR3#-B/ BC3#-B		GTIOC10B	RXD004-B/ SMISO004-B/ SSCL004		XSPI1_RSTO0#/ XSPI1_RESET0#/ VSYNC	IRQ1	ADST1	
121	ETHPHYCLK	P700	PIO_048	SDCLK-A		GTIOC10A	TXD004-B/ SMOSI004-B/ SSDA004/ TXDA004		XSPI1_INT0#/ XSPI1_ECS0#/PIXCLK	IRQ0	ADST0	
122	VCC											
123	VSS											
124	VCL											
125		PE07	PIO_047	A8-B/A20-D	TOC1	GTETRGD/ GTCPPO7/ GTIOC10B/ POEGD			XSPI1_CS0#/PIXD7	IRQ7		

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (9/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
126		PE06	PIO_046	A9-B/A21-D	TIC1	GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS/ PIXD6	IRQ6		
127		PE05	PIO_045	A10-B/ A22-D		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3/ PIXD5	IRQ5		
128		PE04	PIO_044	A11-B/ A23-D		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0#/PIXD4	IRQ12		
129		PE03	PIO_043	A12-B	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/ XSPI1_IO1/ PIXD3	IRQ11		
130		PE02	PIO_042	A13-B	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/ XSPI1_IO0/ PIXD2	IRQ7-DS		
131	CLKOUT	PE01	PIO_041	A14-B		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/ XSPI1_CK/ PIXD1	IRQ9		
132	ETHPHYCLK	PE00	PIO_040	A15-B		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN#/ XSPI1_CS1#/PIXD0	IRQ8		
133		P607	PIO_015	CS0#-B/ D16-B		GTETRGB/ GTADSM0/ GTIOC15B/ POEGA	CTS008#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009/ CRX2		XSPI0_RSTO0#	IRQ15		
134		P606	PIO_014	CS1#-C/ D17-B		GTETRGA/ GTCPPO4/ GTIOC15A/ POEGC	SCK009-B/ CTS009#/ TXDB009/ SSL03-D/ CTX2		XSPI0_RESET0#	IRQ14		
135		P605	PIO_013	CS2#-C/ D18-B		GTIOC6B	RXD009-B/ SMISO009-B/ SSCL009/ SSL02-D		XSPI0_INT0#/ XSPI0_ECS0#	IRQ13		
136		P604	PIO_012	CS3#-C/ D19-B		GTIOC6A	TXD009-B/ SMOSI009-B/ SSDA009/ TXDA009/ SSL01-D		XSPI0_CS1#/ XSPI0_RSTO0#	IRQ4		
137		P603	PIO_011	WE#-B/ D20-B		GTIOC5B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS009#/ MISO0-D		XSPI0_IO7/ PCKO	IRQ3		
138		P602	PIO_010	CAS#-B/ D21-B		GTIOC5A	SCK008-B/ CTS008#/ TXDB008/ MOSI0-D		XSPI0_IO6/ HSYNC	IRQ2	ADTRG2#	
139		P601	PIO_009	RAS#-B/ D22-B		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTPOUT1	XSPI0_IO5/ VSYNC	IRQ1	ADTRG1#	

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (10/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
140		P600	PIO_008	SDCS#-B/ D23-B		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTP OUT0	XSPI0_IO4/ PIXCLK	IRQ0	ADTRG0 #	
141	VCC2											
142	VSS											
143	VCL											
144		PD07	PIO_039	A16-B/ D24-B		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0 #/ XSPI0_RES ET1#/PIXD7	IRQ7	AN031	
145		PD06	PIO_038	A17-B/ D25-B		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS/ PIXD6	IRQ6	AN030	
146		PD05	PIO_037	A18-B/ D26-B		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1- B/ XSPI0_IO3/ PIXD5	IRQ5	AN029	
147		PD04	PIO_036	A19-B/ D27-B		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0- B/ XSPI0_IO2/ XSPI0_WP0 #/PIXD4	IRQ4	AN028	
148		PD03	PIO_035	A20-B/ D28-B	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK- B/ XSPI0_IO1/ PIXD3	IRQ3	AN027	
149		PD02	PIO_034	A21-B/ D29-B	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD- B/ XSPI0_IO0/ PIXD2	IRQ2	AN026	
150		PD01	PIO_033	A22-B/ D30-B		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3- B/ XSPI0_CK/ PIXD1	IRQ1	AN025	
151		PD00	PIO_032	A23-B/ D31-B		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2- B/ XSPI0_CKN #/ XSPI0_CS1 #/PIXD0	IRQ0	AN024	
152	VCC2											
153	VSS											
154	VCL											
155		P903	PIO_003			GTETRGD/ GTCPPO9/ POEGD	CTS006#/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001		XSPI0_WP1 #	IRQ11	AN035	
156		P902	PIO_002			GTETRGC/ GTCPPO8	SCK001-C/ CTS001#/ TXDB001		XSPI0_RES ET1#	IRQ10	AN034	

Table 1.8 List of Pin and Pin Functions (176-Pin LFQFP) (11/11)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
176-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
157		P901	PIO_001			GTETRGB/ GTCPPO6	TXD001-C/ SMOSI001-C/ SSDA001/ TXDA001		XSPI0_RST O1#/ XSPI0_RES ET0#	IRQ9	AN033	
158		P900	PIO_000			GTETRGA/ GTCPPO5	RXD001-C/ SMISO001-C/ SSCL001		XSPI0_INT1 #/ XSPI0_ECS 1#	IRQ0	AN032	
159	AVCC0											
160	AVSS0											
161	VREFL0											
162	VREFH0											
163		P407	PIO_055							IRQ15-DS	AN007	
164		P406	PIO_054							IRQ14-DS	AN006	
165		P405	PIO_053							IRQ13-DS	AN005	
166		P404	PIO_052							IRQ12-DS	AN004	
167		P403	PIO_051							IRQ11-DS	AN003	
168		P402	PIO_050							IRQ10-DS	AN002	
169		P401	PIO_049							IRQ9-DS	AN001	
170		P400	PIO_048							IRQ8-DS	AN000	
171		P415	PIO_063							IRQ31	AN015/ DA1	
172		P414	PIO_062							IRQ30	AN014/ DA0	
173		P007	PIO_007			GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0 #/AN023	PDMDAT0
174		P006	PIO_006			GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1 #/AN022	PDMCLK0 / PCOUTZ1
175		P005	PIO_005			GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2 #/AN021	PCOUTB1
176		P004	PIO_004			GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1

1.6.5 144-Pin LFQFP

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (1/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
1		P003	PIO_003			GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0
2		P002	PIO_002		TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
3		P001	PIO_001		TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
4		P000	PIO_000		TMRI0	GTCPPO7/ GTIOC10A/ POEGB	CTS000#/ CTS007#/ RTS007#/ SS007#/ DE007/ TXDB007/ AUDIO_CLK	ET0_INT		IRQ16-DS	AN016	
5	VCC											
6	VSS											
7	VCL											
8	RES#											
9	MD/FINED	P315										
10		P312	PIO_060	BACK#-B	RTCIC2	GTETRGD/ GTCPPO15/POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2#	TAMPI2
11		P311	PIO_059	BREQ#-B	RTCIC1	GTETRGC/ GTCPPO14/POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1#	TAMPI1
12		P310	PIO_058		RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
13	VBATT/ VCC											
14	XCIN/ EXCIN	P309										
15	XCOUT	P308										
16	VSS											
17	XTAL	P307										
18	EXTAL	P306										
19	VCC											
20		P305								NMI/IRQ0		
21	VCC_USB											
22		PF01	PIO_049			GTCPPO15	USB0_DM			IRQ1		
23		PF00	PIO_048			GTCPPO14	USB0_DP			IRQ0		
24	VSS_USB											

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (2/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
25		P207	PIO_023	A23-C/ CS4#-C	TMRI3	GTOWLO/ GTIOC2B/ GTIOC6B	RXD001-B/ SMISO001-B/ SSCL001/ MISO1-B/ SDAHS2-B/ CRX1/ USB0_ID	ET0_RXD3-B	PCKO	IRQ3-DS	ADTRG2#	
26		P206	PIO_022	A22-C/ CS5#-C	TMO3/ RTCOUT	GTOWUP/ GTIOC2A/ GTIOC6A	CTS006#/ TXD001-B/ SMOSI001-B/ SSDA001/ TXDA001/ MOSI1-B/ SCLHS2-B/ CTX1/ USB0_EXICEN/SSITXD1	ET0_RXD2-B	VSYNC	IRQ2-DS	ADTRG1#	PCOUTZ1
27	CLKOUT	P205	PIO_021	A21-C/ CS6#-C		GTOVLO/ GTIOC1B/ GTIOC5B	SCK001-B/ CTS001#/ TXDB001/ RSPCK1-B/ SDAHS1-B/ CRX0/ USB0_VBUS/ SSIRXD1	ET0_RXD1-B/ RMII0_RXD1-B	SD0_CD/ HSYNC	IRQ5	ADTRG0#	PCOUTB1
28	ETHPHYCLK	P204	PIO_020	A20-C/ CS7#-C	TMRI1	GTOVUP/ GTIOC1A/ GTIOC5A	CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-B/ SCLHS1-B/ CTX0/ USB0_VBUSEN/SSIBCK1	ET0_RXD0-B/ RMII0_RXD0-B	SD0_WP/ PIXCLK	IRQ12		PCOUTA1
29	VCL											
30	VSS											
31	VCC											
32		P203	PIO_019	CS2#-B/ BACK#-A		GTOULO/ GTIOC0B/ GTIOC4B	RXD006-A/ SMISO006-A/ SSCL006/ CTS003#/ SSL00-C/ SDA00-B/ CRX2/ USB0_OVRCURB/ SSIBCK0	ET0_RX_CLK-B/ RMII0_REF50CK-B	SD0_DAT1-C/PIXD7	IRQ3		PDMDAT0
33		P202	PIO_018	CS3#-B/ BREQ#-A	TMO0	GTOUUP/ GTIOC0A/ GTIOC4A	TXD006-A/ SMOSI006-A/ SSDA006/ TXDA006/ RSPCK0-C/ SCL00-B/ CTX2/ USB0_OVRCURA/ AUDIO_CLK	ET0_RX_DV-B/ RMII0_CRS_DV-B	SD0_DAT0-C/PIXD6	IRQ15		PDMCLK0
34	ETHPHYCLK	P304	PIO_028	ALE-C/ WAIT#-B	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUSEN	ET0_INT		IRQ0-DS	ADTRG0#	
35	TDI	P303	PIO_027		TMCI2	GTCPPO13	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (3/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
36	TDO	P302	PIO_026		TMRI3	GTCPPO12	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
37	TCK/ SWCLK	P301	PIO_025		TMCI3	GTCPPO11	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		
38	TMS/ SWDIO	P300	PIO_024		TMO1	GTCPPO10	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
39		P201	PIO_017		TMCI0	GTIW/ GTIOC3B/ GTIOC14B	SCK006-A/ CTS006#/ TXDB006/ MISO0-C/ SCLHS1-A/ USB0_EXICEN/SSILRCK0	ET0_RX_ER-B/ RMII0_RX_ER-B	SD0_CLK-C/PIXD5	IRQ9		PCINZ0
40		P200	PIO_016		TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	CTS006#/ RTS006#/ SS006#/ DE006/ TXDB006/ MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_EN-B	SD0_CMD-C/PIXD4	IRQ8	ADTRG2#	PCINZ1
41		P107	PIO_015		TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C/PIXD3	IRQ31-DS	ADTRG1#	PCINB1
42	CLKOUT	P106	PIO_014		TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C/PIXD2	IRQ30-DS	ADTRG0#	PCINA1
43		P105	PIO_013		TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B	PIXD1	IRQ29-DS	ADST1	PCINB0

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (4/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
44		P104	PIO_012		TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRCURA-DS	ET0_TXD0-B/ RMII0_TXD0-B	PIXD0	IRQ4	ADST0	PCINA0
45	VCL											
46	VSS											
47	VCC											
48		P103	PIO_011		TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		
49		P102	PIO_010		TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
50		P101	PIO_009	CS0#-A/ ALE-B	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUS-DS	ET0_TX_CLK-B		IRQ1	ADTRG1#	
51		P100	PIO_008	CS1#-B/ WAIT#-A	TMRI3/ CACREF	GTIU/ GTADSM0/ GTIOC12A	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS005#/ USB0_OVRCURB-DS	ET0_TX_ER-B		IRQ0	ADTRG2#	
52	VCC_DCDC											
53	VCC_DCDC											
54	VLO											
55	VLO											
56	VSS_DCDC											
57	VSS_DCDC											
58		P803	PIO_059	CS3#-A/ WR1#-B/ BC1#-B		GTIOC1B/ POEGA	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ CTX2/ USB0_ID	ET0_MDC-B		IRQ3	ADSYNC	
59		P802	PIO_058	CS2#-A/ WR#-B/ WR0#-B		GTIW/ GTIOC1A/ POEGB	SCK010-B/ CTS010#/ TXDB010/ CRX2/ USB0_EXICEN	ET0_MDIO-B		IRQ2	ADCNV2	
60	VCL											
61	VSS											
62	VCC											
63	CLKOUT	P801	PIO_057	EBCLK-A		GTIV/ GTIOC0B/ POEGC	TXD010-B/ SMOSI010-B/ SSDA010/ TXDA010	ET0_CRS		IRQ9	ADCNV1	

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (5/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
64		P800	PIO_056	CS1#-A/ DQM0-B		GTIU/ GTIOC0A/ POEGD	RXD010-B/ SMISO010-B/ SSCL010	ET0_COL		IRQ8	ADCNV0	
65	ETHPHYCLK	PC07	PIO_031	D0-A	TMO2/ TOC0/ CACREF	GTETRGD/ GTADSM1/ GTIOC15B/ POEGA	TXD008-A/ SMOSI008-A/ SSDA008/ TXDA008/ MISO0-A/ SSITXD0		SD0_CD	IRQ14		PDMDAT0
66		PC06	PIO_030	D1-A	TMCI2/ TIC0	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER-A	SD0_WP	IRQ13		PDMCLK0
67		PC05	PIO_029	D2-A	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_EN-A	SD0_DAT1-A	IRQ5		PCOUTZ1
68		PC04	PIO_028	D3-A	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
69		PC03	PIO_027	D4-A		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
70		PC02	PIO_026	D5-A		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD1-A	SD0_CMD-A	IRQ10		PCOUTZ0
71		PC01	PIO_025	D6-A		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
72		PC00	PIO_024	D7-A		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
73		PB07	PIO_023	D8-A		GTOWLO/ GTIOC6B	TXD009-A/ SMOSI009-A/ SSDA009/ TXDA009	ET0_LINKSTA		IRQ15		
74		PB06	PIO_022	D9-A		GTOWUP/ GTIOC6A	RXD009-A/ SMISO009-A/ SSCL009	ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
75		PB05	PIO_021	D10-A	TMRI1	GTOVLO/ GTIOC5B	SCK009-A/ CTS009#/ TXDB009	ET0_RX_DV-A/ RMII0_CRS_DV-A		IRQ13		
76		PB04	PIO_020	D11-A		GTOVUP/ GTIOC5A	CTS004#/ CTS009#/ RTS009#/ SS009#/ DE009/ TXDB009	ET0_RX_CLK-A/ RMII0_REF50CK-A		IRQ4		

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (6/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
77	VCC											
78	VSS											
79	VCL											
80		PB03	PIO_019	D12-A	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A		IRQ3		
81		PB02	PIO_018	D13-A		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004/ CTS009#	ET0_RXD2-A		IRQ2		
82		PB01	PIO_017	D14-A	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A		IRQ4-DS		PCINB1
83		PB00	PIO_016	D15-A		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A		IRQ12	ADST2	PCINA1
84	ETHPHYCLK	PA07	PIO_015	A0-A/ BC0#-A/ DQM1-B		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0
85		PA06	PIO_014	A1-A	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
86		PA05	PIO_013	A2-A		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0
87		PA04	PIO_012	A3-A	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
88	VCC											
89	VSS											
90	VCL											
91		PA03	PIO_011	A4-A		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
92		PA02	PIO_010	A5-A		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
93		PA01	PIO_009	A6-A		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
94	ETHPHYCLK	PA00	PIO_008	A7-A	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (7/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
95		P702	PIO_050	RD#-A		GTETRGA/ GTIOC11A	SCK004-B/ CTS004#/ TXDB004		XSPI1_WP0 #/ XSPI1_RES ET1#/ HSYNC	IRQ10	ADST2	
96		P701	PIO_049	CKE-A		GTIOC10B	RXD004-B/ SMISO004-B/ SSCL004		XSPI1_RST O0#/ XSPI1_RES ET0#/ VSYNC	IRQ1	ADST1	
97	ETHPHYC LK	P700	PIO_048	SDCLK-A		GTIOC10A	TXD004-B/ SMOSI004-B/ SSDA004/ TXDA004		XSPI1_INT0 #/ XSPI1_ECS 0#/PIXCLK	IRQ0	ADST0	
98	VCC											
99	VSS											
100	VCL											
101		PE07	PIO_047	A8-B/A20-D	TOC1	GTETRGD/ GTCPPO7/ GTIOC10B/ POEGD			XSPI1_CS0 #/PIXD7	IRQ7		
102		PE06	PIO_046	A9-B/A21-D	TIC1	GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS/ PIXD6	IRQ6		
103		PE05	PIO_045	A10-B/ A22-D		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3/ PIXD5	IRQ5		
104		PE04	PIO_044	A11-B/ A23-D		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0 #/PIXD4	IRQ12		
105		PE03	PIO_043	A12-B	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/ XSPI1_IO1/ PIXD3	IRQ11		
106		PE02	PIO_042	A13-B	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/ XSPI1_IO0/ PIXD2	IRQ7-DS		
107	CLKOUT	PE01	PIO_041	A14-B		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/ XSPI1_CK/ PIXD1	IRQ9		
108	ETHPHYC LK	PE00	PIO_040	A15-B		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN #/ XSPI1_CS1 #/PIXD0	IRQ8		
109		P605	PIO_013	CS2#-C		GTIOC6B	RXD009-B/ SMISO009-B/ SSCL009/ SSL02-D		XSPI0_INT0 #/ XSPI0_ECS 0#	IRQ13		
110		P604	PIO_012	CS3#-C		GTIOC6A	TXD009-B/ SMOSI009-B/ SSDA009/ TXDA009/ SSL01-D		XSPI0_CS1 #/ XSPI0_RST O0#	IRQ4		

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (8/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
111		P603	PIO_011	WE#-B		GTIOC5B	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS009#/ MISO0-D		XSPI0_IO7/ PCKO	IRQ3		
112		P602	PIO_010	CAS#-B		GTIOC5A	SCK008-B/ CTS008#/ TXDB008/ MOSI0-D		XSPI0_IO6/ HSYNC	IRQ2	ADTRG2#	
113		P601	PIO_009	RAS#-B		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTPOUT1	XSPI0_IO5/ VSYNC	IRQ1	ADTRG1#	
114		P600	PIO_008	SDCS#-B		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTPOUT0	XSPI0_IO4/ PIXCLK	IRQ0	ADTRG0#	
115	VCC2											
116	VSS											
117		PD07	PIO_039	A16-B		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0#/ XSPI0_RESET1#/PIXD7	IRQ7	AN031	
118		PD06	PIO_038	A17-B		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS/ PIXD6	IRQ6	AN030	
119		PD05	PIO_037	A18-B		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1-B/ XSPI0_IO3/ PIXD5	IRQ5	AN029	
120		PD04	PIO_036	A19-B		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0-B/ XSPI0_IO2/ XSPI0_WP0#/PIXD4	IRQ4	AN028	
121		PD03	PIO_035	A20-B	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK-B/ XSPI0_IO1/ PIXD3	IRQ3	AN027	
122		PD02	PIO_034	A21-B	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD-B/ XSPI0_IO0/ PIXD2	IRQ2	AN026	
123		PD01	PIO_033	A22-B		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3-B/ XSPI0_CK/ PIXD1	IRQ1	AN025	
124		PD00	PIO_032	A23-B		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2-B/ XSPI0_CKN#/ XSPI0_CS1#/PIXD0	IRQ0	AN024	
125	VSS											

Table 1.9 List of Pin and Pin Functions (144-Pin LFQFP) (9/9)

Pin Number					Timer		Communication		Memory I/F Camera I/F			Others
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	Bus EXDMAC SDRAMC	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI, PDC)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
126	VCL											
127		P901	PIO_001			GTETRGB/ GTCPPO6	TXD001-C/ SMOSI001-C/ SSDA001/ TXDA001		XSPI0_RSTO1#/ XSPI0_RESET0#	IRQ9	AN033	
128		P900	PIO_000			GTETRGA/ GTCPPO5	RXD001-C/ SMISO001-C/ SSCL001		XSPI0_INT1#/ XSPI0_ECS1#	IRQ0	AN032	
129	AVCC0											
130	AVSS0											
131	VREFL0											
132	VREFH0											
133		P405	PIO_053							IRQ13-DS	AN005	
134		P404	PIO_052							IRQ12-DS	AN004	
135		P403	PIO_051							IRQ11-DS	AN003	
136		P402	PIO_050							IRQ10-DS	AN002	
137		P401	PIO_049							IRQ9-DS	AN001	
138		P400	PIO_048							IRQ8-DS	AN000	
139		P415	PIO_063							IRQ31	AN015/ DA1	
140		P414	PIO_062							IRQ30	AN014/ DA0	
141		P007	PIO_007			GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0#/AN023	PDMDAT0
142		P006	PIO_006			GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1#/AN022	PDMCLK0/ PCOUTZ1
143		P005	PIO_005			GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2#/AN021	PCOUTB1
144		P004	PIO_004			GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1

1.6.6 100-Pin LFQFP

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (1/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
1		P003	PIO_003		GTCPPO2/ GTIOC11B/ GTIOC8A	RXD007-A/ SMISO007-A/ SSCL007/ SSIRXD1	ET0_MDIO-D		IRQ19-DS	AN019	PCOUTB0
2		P002	PIO_002	TMCI1	GTCPPO1/ GTIOC11A/ GTIOC7B	TXD007-A/ SMOSI007-A/ SSDA007/ TXDA007/ SSIBCK1	ET0_MDC-D		IRQ18-DS	AN018	PCOUTA0
3		P001	PIO_001	TMCI0	GTCPPO0/ GTIOC10B/ GTIOC7A	SCK007-A/ CTS007#/ TXDB007/ SSIBCK0	ET0_LINKSTA		IRQ17-DS	AN017	PCOUTZ0
4	VSS										
5	VCL										
6	RES#										
7	MD/FINED	P315									
8		P312	PIO_060	RTCIC2	GTETRGD/ GTCPPO15/ POEGA	SSL13-B/ SSILRCK0			IRQ20	ADTRG2#	TAMPI2
9		P311	PIO_059	RTCIC1	GTETRGC/ GTCPPO14/ POEGD	CTS007#/ SSL12-B/ SSIRXD0	ET0_WOL		IRQ19	ADTRG1#	TAMPI1
10		P310	PIO_058	RTCIC0/ CACREF	GTETRGB/ GTCPPO1/ POEGC	CTS000#/ CTS001#/ SSL11-B/ SSITXD0			IRQ18	ADTRG0#	TAMPI0
11	VBATT/ VCC										
12	XCIN/ EXCIN	P309									
13	XCOUT	P308									
14	VSS										
15	XTAL	P307									
16	EXTAL	P306									
17	VCC										
18		P305							NMI/IRQ0		
19	VCC_USB										
20		PF01	PIO_049		GTCPPO15	USB0_DM			IRQ1		
21		PF00	PIO_048		GTCPPO14	USB0_DP			IRQ0		
22	VSS_USB										
23	ETHPHYCLK	P304	PIO_028	TMCI3	GTETRGD/ GTCPPO0/ POEGA	CTS000#/ CTS001#/ USB0_VBUSEN	ET0_INT		IRQ0-DS	ADTRG0#	
24	TDI	P303	PIO_027	TMCI2	GTCPPO13	RXD000-A/ SMISO000-A/ SSCL000/ RXD001-A/ SMISO001-A/ SSCL001/ MISO1-C			IRQ1		

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (2/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
25	TDO	P302	PIO_026	TMRI3	GTCPPO12	TXD000-A/ SMOSI000-A/ SSDA000/ TXDA000/ TXD001-A/ SMOSI001-A/ SSDA001/ TXDA001/ MOSI1-C			IRQ4		
26	TCK/ SWCLK	P301	PIO_025	TMCI3	GTCPPO11	SCK000-A/ CTS000#/ TXDB000/ SCK001-A/ CTS001#/ TXDB001/ RSPCK1-C			IRQ7		
27	TMS/ SWDIO	P300	PIO_024	TMO1	GTCPPO10	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000/ CTS001#/ RTS001#/ SS001#/ DE001/ TXDB001/ SSL10-C	ET0_WOL		IRQ6		
28		P201	PIO_017	TMCI0	GTIW/ GTIOC3B/ GTIOC14B	MISO0-C/ SCLHS1-A/ USB0_EXICEN/SSILRCK0	ET0_RX_ER-B/ RMII0_RX_ER-B	SD0_CLK-C	IRQ9		PCINZ0
29		P200	PIO_016	TMRI0	GTETRGA/ GTIOC3A/ GTIOC14A	MOSI0-C/ SDAHS1-A/ USB0_ID/ SSIRXD0	ET0_TX_EN-B/ RMII0_TX_EN-B	SD0_CMD-C	IRQ8	ADTRG2#	PCINZ1
30		P107	PIO_015	TMO1	GTIV/ GTIOC15B/ POEGA	TXD003-B/ SMOSI003-B/ SSDA003/ TXDA003/ SDAHS2-A/ SSITXD0	ET0_TXD3-B	SD0_DAT3-C	IRQ31-DS	ADTRG1#	PCINB1
31	CLKOUT	P106	PIO_014	TMO2/ RTCOUT	GTIU/ GTIOC15A/ POEGB	RXD003-B/ SMISO003-B/ SSCL003/ SCLHS2-A	ET0_TXD2-B	SD0_DAT2-C	IRQ30-DS	ADTRG0#	PCINA1
32		P105	PIO_013	TMCI2	GTIV/ GTIOC14B	TXD002-B/ SMOSI002-B/ SSDA002/ TXDA002/ SCK003-B/ CTS003#/ TXDB003/ CRX2/ USB0_VBUS/ SSILRCK1	ET0_TXD1-B/ RMII0_TXD1-B		IRQ29-DS	ADST1	PCINB0

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (3/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
33		P104	PIO_012	TMRI2	GTIW/ GTIOC14A	RXD002-B/ SMISO002-B/ SSCL002/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ CTX2/ USB0_OVRCURA-DS	ET0_TXD0-B/ RMII0_TXD0-B		IRQ4	ADST0	PCINA0
34	VCL										
35	VSS										
36	VCC										
37		P103	PIO_011	TMO3	GTADSM1/ GTIOC13B	CTS002#/ SDA00-A	ET0_CRS		IRQ3		
38		P102	PIO_010	TMCI1	GTADSM0/ GTIOC13A	SCL00-A	ET0_COL		IRQ2		
39		P101	PIO_009	TMCI3	GTETRGC/ GTADSM1/ GTIOC12B	SCK002-B/ CTS002#/ TXDB002/ USB0_VBUS-DS	ET0_TX_CLK-B		IRQ1	ADTRG1#	
40	VCC_DCDC										
41	VLO										
42	VSS_DCDC										
43	VSS										
44		PC06	PIO_030	TMCI2	GTETRGC/ GTCPPO7/ GTIOC15A/ POEGB	RXD008-A/ SMISO008-A/ SSCL008/ MOSI0-A/ SSILRCK0	ET0_TX_ER-A	SD0_WP	IRQ13		PDMCLK0
45		PC05	PIO_029	TMRI2	GTOWLO/ GTIOC9B	SCK008-A/ CTS008#/ TXDB008/ RSPCK0-A/ SSIBCK0	ET0_TX_EN-A/ RMII0_TX_EN-A	SD0_DAT1-A	IRQ5		PCOUTZ1
46		PC04	PIO_028	TMCI1	GTOWUP/ GTIOC9A	CTS008#/ RTS008#/ SS008#/ DE008/ TXDB008/ CTS011#/ SSL00-A/ AUDIO_CLK	ET0_TXD3-A	SD0_DAT0-A	IRQ12		PCOUTB1
47		PC03	PIO_027		GTOVLO/ GTIOC8B	TXD011-A/ SMOSI011-A/ SSDA011/ TXDA011/ SDAHS1-C	ET0_TXD2-A	SD0_CLK-A	IRQ11		PCOUTA1
48		PC02	PIO_026		GTOVUP/ GTIOC8A	RXD011-A/ SMISO011-A/ SSCL011/ SSL03-A/ SCLHS1-C	ET0_TXD1-A/ RMII0_TXD1-A	SD0_CMD-A	IRQ10		PCOUTZ0

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (4/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
49		PC01	PIO_025		GTOULO/ GTIOC7B	SCK011-A/ CTS011#/ TXDB011/ SSL02-A/ SDA00-C	ET0_TXD0-A/ RMII0_TXD0-A	SD0_DAT3-A	IRQ12		PCOUTB0
50		PC00	PIO_024		GTOUUP/ GTIOC7A	CTS011#/ RTS011#/ SS011#/ DE011/ TXDB011/ SSL01-A/ SCL00-C	ET0_TX_CLK-A	SD0_DAT2-A	IRQ14		PCOUTA0
51		PB06	PIO_022		GTOWUP/ GTIOC6A		ET0_RX_ER-A/ RMII0_RX_ER-A		IRQ6		
52		PB05	PIO_021	TMRI1	GTOVLO/ GTIOC5B		ET0_RX_DV-A/ RMII0_CRS_DV-A		IRQ13		
53		PB04	PIO_020		GTOVUP/ GTIOC5A	CTS004#	ET0_RX_CLK-A/ RMII0_REF50CK-A		IRQ4		
54		PB03	PIO_019	TMO0	GTOULO/ GTIOC4B	SCK004-A/ CTS004#/ TXDB004	ET0_RXD3-A		IRQ3		
55		PB02	PIO_018		GTOUUP/ GTIOC4A	CTS004#/ RTS004#/ SS004#/ DE004/ TXDB004	ET0_RXD2-A		IRQ2		
56		PB01	PIO_017	TMCI0	GTETRGB/ GTCPPO4/ GTIOC14B/ POEGA	TXD004-A/ SMOSI004-A/ SSDA004/ TXDA004	ET0_RXD1-A/ RMII0_RXD1-A		IRQ4-DS		PCINB1
57		PB00	PIO_016		GTETRGA/ GTADSM0/ GTIOC14A/ POEGC	RXD004-A/ SMISO004-A/ SSCL004	ET0_RXD0-A/ RMII0_RXD0-A		IRQ12	ADST2	PCINA1
58	ETHPHYCLK	PA07	PIO_015		GTIOC3B/ GTIOC13B	RXD010-A/ SMISO010-A/ SSCL010/ MISO0-B			IRQ7		PCINB0
59		PA06	PIO_014	TMCI3	GTIOC3A/ GTIOC13A	TXD010-A/ SMOSI010-A/ SSDA010/ TXDA010/ MOSI0-B			IRQ14		PCINZ1
60		PA05	PIO_013		GTIOC2B/ GTIOC12B	SCK010-A/ CTS010#/ TXDB010/ RSPCK0-B			IRQ5		PCINA0

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (5/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
61		PA04	PIO_012	TMRI0	GTIOC2A/ GTIOC12A	CTS010#/ RTS010#/ SS010#/ DE010/ TXDB010/ SSL00-B			IRQ5-DS		PCINZ0
62	VCC										
63	VSS										
64	VCL										
65		PA03	PIO_011		GTIOC1B/ GTIOC11B	RXD005-A/ SMISO005-A/ SSCL005			IRQ6-DS	ADSYNC	
66		PA02	PIO_010		GTIW/ GTIOC1A/ GTIOC11A	TXD005-A/ SMOSI005-A/ SSDA005/ TXDA005/ SSL03-B			IRQ10	ADCNV2	
67		PA01	PIO_009		GTIV/ GTIOC0B/ GTIOC10B	SCK005-A/ CTS005#/ TXDB005/ SSL02-B			IRQ11	ADCNV1	
68	ETHPHYCLK	PA00	PIO_008	CACREF	GTIU/ GTIOC0A/ GTIOC10A	CTS010#/ CTS005#/ RTS005#/ SS005#/ DE005/ TXDB005/ SSL01-B			IRQ1-DS	ADCNV0	
69		PE06	PIO_046		GTETRGC/ GTADSM1/ GTIOC10A/ POEGB	SSL13-A		XSPI1_DS	IRQ6		
70		PE05	PIO_045		GTOWLO/ GTIOC9B	SSL12-A		XSPI1_IO3	IRQ5		
71		PE04	PIO_044		GTOWUP/ GTIOC9A	CTS012#/ SSL11-A		XSPI1_IO2/ XSPI1_WP0#	IRQ12		
72		PE03	PIO_043	TOC3	GTOVLO/ GTIOC8B	CTS012#/ RTS012#/ SS012#/ DE012/ TXDB012/ MISO1-A		SD0_DAT7-B/XSPI1_IO1	IRQ11		
73		PE02	PIO_042	TIC3	GTOVUP/ GTIOC8A	RXD012-A/ SMISO012-A/ SSCL012/ MOSI1-A		SD0_DAT6-B/XSPI1_IO0	IRQ7-DS		

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (6/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
74	CLKOUT	PE01	PIO_041		GTOULO/ GTIOC7B	TXD012-A/ SMOSI012-A/ SSDA012/ TXDA012/ RSPCK1-A		SD0_DAT5-B/XSPI1_CK	IRQ9		
75	ETHPHYCLK	PE00	PIO_040		GTOUUP/ GTIOC7A	SCK012-A/ CTS012#/ TXDB012/ SSL10-A		SD0_DAT4-B/ XSPI1_CKN#/ XSPI1_CS1#	IRQ8		
76		P601	PIO_009		GTIOC4B	RXD008-B/ SMISO008-B/ SSCL008/ RSPCK0-D	GPTP_PTPOUT1		IRQ1	ADTRG1#	
77		P600	PIO_008		GTIOC4A	TXD008-B/ SMOSI008-B/ SSDA008/ TXDA008/ SSL00-D	GPTP_PTPOUT0		IRQ0	ADTRG0#	
78		PD07	PIO_039		GTIW/ GTIOC3B/ POEGA	TXD003-A/ SMOSI003-A/ SSDA003/ TXDA003/ CTX1		SD0_CD/ XSPI0_CS0#/ XSPI0_RESET1#	IRQ7	AN031	
79		PD06	PIO_038		GTIV/ GTIOC3A/ POEGB	RXD003-A/ SMISO003-A/ SSCL003/ SSL23-A/ CRX1		SD0_WP/ XSPI0_DS	IRQ6	AN030	
80		PD05	PIO_037		GTIU/ GTIOC2B/ POEGD	SCK003-A/ CTS003#/ TXDB003/ SSL22-A/ CTX0		SD0_DAT1-B/XSPI0_IO3	IRQ5	AN029	
81		PD04	PIO_036		GTETRGA/ GTIOC2A/ POEGC	CTS002#/ CTS003#/ RTS003#/ SS003#/ DE003/ TXDB003/ SSL21-A/ CRX0		SD0_DAT0-B/ XSPI0_IO2/ XSPI0_WP0#	IRQ4	AN028	
82		PD03	PIO_035	TOC2	GTIOC1B/ POEGA	TXD002-A/ SMOSI002-A/ SSDA002/ TXDA002/ MISO2-A		SD0_CLK-B/ XSPI0_IO1	IRQ3	AN027	

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (7/7)

Pin Number				Timer		Communication		Memory I/F Camera I/F			Others
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	PIO	(TMR, CMTW, RTC, CAC)	(GPTW, POEG)	(RSCI, RSPIA, RI3C, RIICHS, CANFD, USB, SSIE)	(ESWM)	(XSPI, SDHI)	Interrupt	A/D D/A	(PDMIF, PCIF, Tamper detection)
83		PD02	PIO_034	TIC2	GTETRGB/ GTIOC1A/ POEGB	RXD002-A/ SMISO002-A/ SSCL002/ MOSI2-A		SD0_CMD-B/ XSPI0_IO0	IRQ2	AN026	
84		PD01	PIO_033		GTETRGC/ GTIOC0B/ POEGC	SCK002-A/ CTS002#/ TXDB002/ RSPCK2-A		SD0_DAT3-B/XSPI0_CK	IRQ1	AN025	
85		PD00	PIO_032		GTETRGD/ GTIOC0A/ POEGD	CTS002#/ RTS002#/ SS002#/ DE002/ TXDB002/ CTS003#/ SSL20-A		SD0_DAT2-B/ XSPI0_CKN#/ XSPI0_CS1#	IRQ0	AN024	
86	VCC2										
87	VSS										
88	VCL										
89	AVCC0										
90	AVSS0										
91	VREFL0										
92	VREFH0										
93		P404	PIO_052						IRQ12-DS	AN004	
94		P402	PIO_050						IRQ10-DS	AN002	
95		P400	PIO_048						IRQ8-DS	AN000	
96		P414	PIO_062						IRQ30	AN014/ DA0	
97		P007	PIO_007		GTCPPO6/ GTIOC13B/ POEGA	CTS000#/ RTS000#/ SS000#/ DE000/ TXDB000			IRQ23-DS	ADTRG0#/ AN023	PDMDAT0
98		P006	PIO_006		GTCPPO5/ GTIOC13A/ GTIOC9B	SCK000-B/ CTS000#/ TXDB000			IRQ22-DS	ADTRG1#/ AN022	PDMCLK0/ PCOUTZ1
99		P005	PIO_005		GTCPPO4/ GTIOC12B/ GTIOC9A	RXD000-B/ SMISO000-B/ SSCL000/ SSILRCK1			IRQ21-DS	ADTRG2#/ AN021	PCOUTB1
100		P004	PIO_004		GTCPPO3/ GTIOC12A/ GTIOC8B	TXD000-B/ SMOSI000-B/ SSDA000/ TXDA000/ SSITXD1			IRQ20-DS	AN020	PCOUTA1

2. CPU

The RXv3e CPU is based on the RXv3e instruction set architecture. The architecture is based on the addition of a security function as an expansion to the architecture for the RXv3 CPU.
The RXv3e instruction set architecture (RXv3e) provides upward compatibility with the RXv3 instruction set architecture (RXv3), the RXv2 instruction set architecture (RXv2), and the RXv1 instruction set architecture (RXv1).

- Protect Zone
  The RXv3e CPU is equipped with Protect Zone technology. This is used to define and manage the security states of the CPU core, and supports the partitioning of software execution environments in terms of these security states.
- Adoption of variable-length instruction format
  The CPU has short formats for frequently used instructions, facilitating the development of programs that are efficient in terms of taking up less memory.
- Powerful instruction set
  DSP instructions and floating-point operation instructions realize high-speed arithmetic processing.
- Versatile addressing modes
  The CPU has versatile addressing modes, with register-register operations, register-memory operations, and bitwise operations included. Data transfer between memory locations is also possible.

2.1 Features

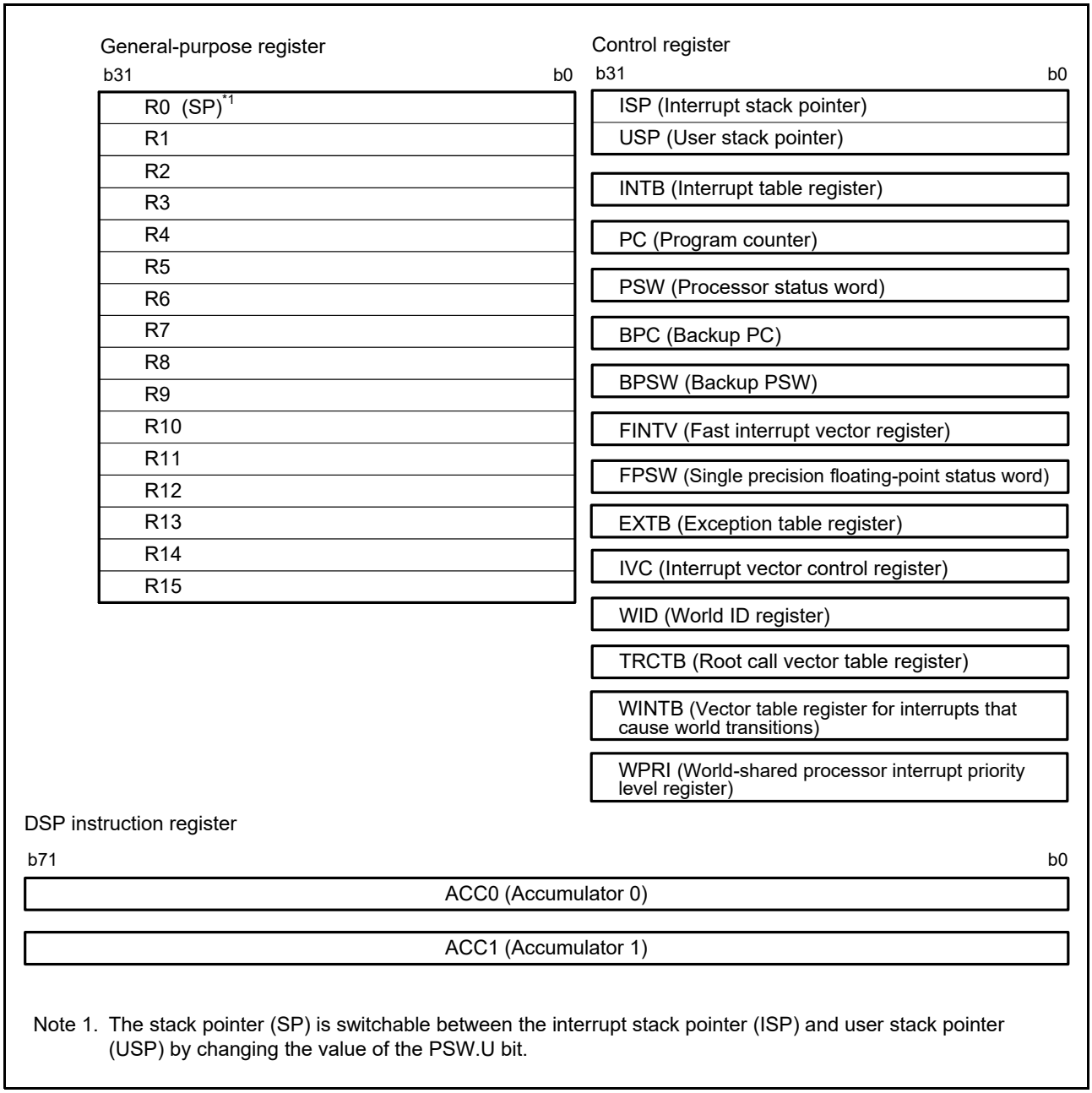
- Minimum instruction execution time: One clock cycle
- Address space: 4-Gbyte linear addresses
- Protect Zone: Three “worlds” (non-secure, secure, and root)
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Fifteen 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 117 instructions
  Standard provided instructions: 111
  Basic instructions: 77
  Single-precision floating point instructions: 11
  DSP instructions: 23
  Instructions for register bank save function: 2
  Protect Zone instructions: 4
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table, interrupt vector table, root call vector table, and world transition interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian
- Register bank save function
- On-chip double-precision floating point coprocessor
  Double-precision floating-point processing instructions: 21
- Instruction cache and data cache
  Instruction cache: 32 Kbytes (4-way set associative cache)

Data cache: 32 Kbytes (4-way set associative cache)

- Branch instruction processing accelerator
  Accelerator for MRAM and LRAM: 272 bytes (Fully associative cache)
- Local SRAM (LRAM)
  128 Kbytes × 2 units

2.2 Register Set of the CPU

The CPU has sixteen general-purpose registers, fifteen control registers, and two accumulators used for DSP instructions.



Note 1. The stack pointer (SP) is switchable between the interrupt stack pointer (ISP) and user stack pointer (USP) by changing the value of the PSW.U bit.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

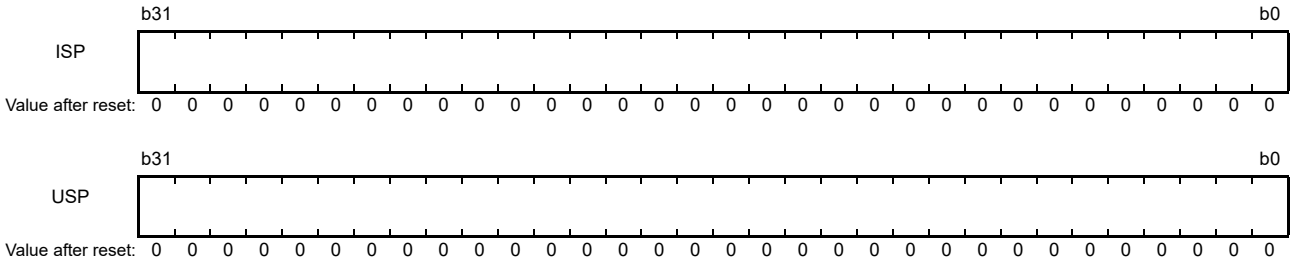
2.2.2 Control Registers

The CPU has sixteen control registers: fifteen are accessible through dedicated instructions.

- Interrupt stack pointer (ISP)*1
- User stack pointer (USP)
- Interrupt table register (INTB)*1
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)*1
- Backup PSW (BPSW)*1
- Fast interrupt vector register (FINTV)
- Single-precision floating-point status word (FPSW)
- Exception table register (EXTB)*1
- Interrupt vector control register (IVC)*1
- World ID register (WID)
- Root call vector table register (TRCTB)
- Vector table register for interrupts that cause world transitions (WINTB)
- World-shared processor interrupt priority level register (WPRI)

Note 1. This control register is actually implemented as three registers to support the Protect Zone function. Where this is required, the three registers are distinguished by postfixes appended to the register names, specifically _TR for the root world register, _S for the secure world register, and _NS for the non-secure world register.

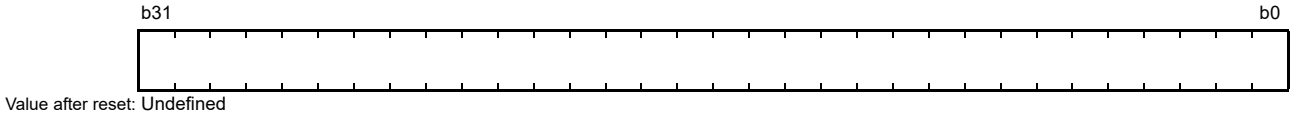
2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
The ISP is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: ISP_TR (only accessible in the root world)
- For the secure world: ISP_S (only accessible in the secure world or root world)
- For the non-secure world: IPS_NS (only accessible in the non-secure world or root world)

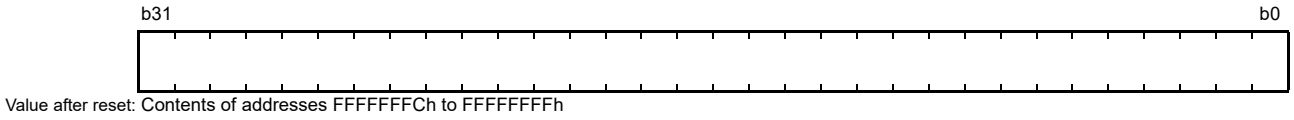
2.2.2.2 Interrupt Table Register (INTB)



The interrupt table register (INTB) specifies the address where the interrupt vector table starts. This register is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: INTB_TR (only accessible in the root world)
- For the secure world: INTB_S (only accessible in the secure world or root world)
- For the non-secure world: INTB_NS (only accessible in the non-secure world or root world)

2.2.2.3 Program Counter (PC)



The program counter (PC) indicates the address of the instruction being executed.

2.2.2.4 Processor Status Word (PSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	IPL[4:0]					—	—	—	PM	—	—	U	I
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C	Carry Flag	0: No carry has occurred. 1: A carry has occurred.	R/W
b1	Z	Zero Flag	0: Result is non-zero. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	I*1	Interrupt Enable	0: See Table 2.1 1: Interrupt enabled.	R/W
b17	U*1	Stack Pointer Select	0: Interrupt stack pointer (ISP) is selected. 1: User stack pointer (USP) is selected.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PM*1, *2, *3	Processor Mode Select	0: Supervisor mode is selected. 1: User mode is selected.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b28 to b24	IPL[4:0]*1, *4	Processor Interrupt Priority Level	b28 b24 0 0 0 0 0: Priority level 0 (lowest) 0 0 0 0 1: Priority level 1 0 0 0 1 0: Priority level 2 0 0 0 1 1: Priority level 3 0 0 1 0 0: Priority level 4 0 0 1 0 1: Priority level 5 0 0 1 1 0: Priority level 6 0 0 1 1 1: Priority level 7 0 1 0 0 0: Priority level 8 0 1 0 0 1: Priority level 9 0 1 0 1 0: Priority level 10 0 1 0 1 1: Priority level 11 0 1 1 0 0: Priority level 12 0 1 1 0 1: Priority level 13 0 1 1 1 0: Priority level 14 0 1 1 1 1: Priority level 15 (non-secure, highest) 1 0 0 0 0: Priority level 16 1 0 0 0 1: Priority level 17 1 0 0 1 0: Priority level 18 1 0 0 1 1: Priority level 19 1 0 1 0 0: Priority level 20 1 0 1 0 1: Priority level 21 1 0 1 1 0: Priority level 22 1 0 1 1 1: Priority level 23 1 1 0 0 0: Priority level 24 1 1 0 0 1: Priority level 25 1 1 0 1 0: Priority level 26 1 1 0 1 1: Priority level 27 1 1 1 0 0: Priority level 28 1 1 1 0 1: Priority level 29 1 1 1 1 0: Priority level 30 1 1 1 1 1: Priority level 31 (highest)	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In user mode, writing to the IPL[4:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[4:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.
Note 4. Writing to the IPL[4:0] bits in response to an MVTC, POPC, or MVTIPL instruction in the non-secure world always causes clearing of the IPL[4] bit (to 0).
Executing an RTE or RTFI instruction in the non-secure world also always causes clearing of the IPL[4] bit (to 0).

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

C Flag (Carry Flag)

This flag retains the state of the bit after a carry, borrow, or shift-out has occurred.

Z Flag (Zero Flag)

This flag is set to 1 if the result of an operation is 0; otherwise its value is set to 0.

S Flag (Sign Flag)

This flag is set to 1 if the result of an operation is negative; otherwise its value is set to 0.

O Flag (Overflow Flag)

This flag is set to 1 if the result of an operation overflows; otherwise its value is set to 0.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When an exception is accepted, the value of this bit becomes 0. Even if the value of

this bit is 0, an interrupt request will only be accepted in the non-secure world when the conditions of both the setting of the NSIC bit in the interrupt vector control register (IVC) and the priority level of the requested interrupt are met as listed in Table 2.1.

Table 2.1 Conditions of Interrupt Acceptance

Current World	I bit	IVC.NSIC	IPR	Enabling or Disabling of Interrupts
Root or secure	0	—	—	Disabled
	1	—	—	Enabled
Non-secure	0	0	—	Disabled
		1	Less than 16	Disabled
			16 or greater	Enabled
	1	—	—	Enabled

IPR: The priority level of a requested interrupt

U Bit (Stack Pointer Select)

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit is set to 0. When the processor mode is switched from supervisor mode to user mode, this bit is set to 1.

PM Bit (Processor Mode Select)

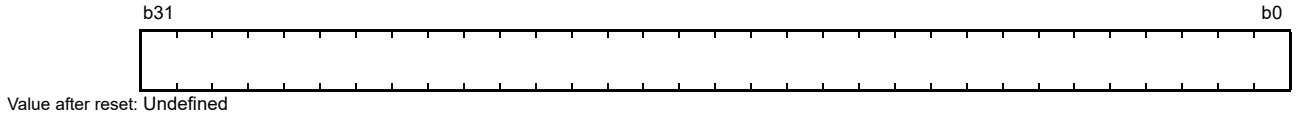
This bit specifies the processor mode. When an exception is accepted, the value of this bit becomes 0.

IPL[4:0] Bits (Processor Interrupt Priority Level)

The IPL[4:0] bits specify the processor interrupt priority level in the root world or secure world as one of 32 levels from 0 to 31, wherein priority level 0 is the lowest and priority level 31 the highest. In the non-secure world, the level is specified as one of 16 levels from 0 to 15, wherein level 0 is the lowest and level 15 the highest. Priority level 16 or a higher level cannot be specified in the non-secure world.

When the priority level of a requested interrupt is higher than the processor interrupt priority level specified in these bits, the interrupt is enabled. Setting the IPL[4:0] bits to level 31 (1Fh) in the root world or secure world disables all interrupt requests. In the non-secure world, setting the IPL[4:0] bits to level 15 (0Fh) disables all interrupt requests for the non-secure world. The IPL[4:0] bits are set to level 31 (1Fh) when a non-maskable interrupt for the root world or secure world is generated. The IPL[4:0] bits are set to level 15 (0Fh) when a non-maskable interrupt for the non-secure world is generated while execution is in progress in the non-secure world. When interrupts in general are generated, the bits are set to the priority levels of accepted interrupts.

2.2.2.5 Backup PC (BPC)



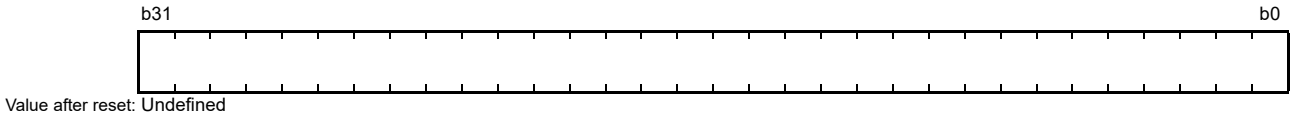
Value after reset: Undefined

The backup PC (BPC) is provided to speed up response to interrupts.

After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register. This register is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: BPC_TR (only accessible in the root world)
- For the secure world: BPC_S (only accessible in the secure world or root world)
- For the non-secure world: BPC_NS (only accessible in the non-secure world or root world)

2.2.2.6 Backup PSW (BPSW)



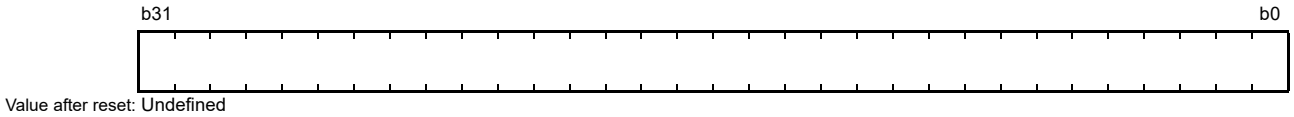
The backup PSW (BPSW) is provided to speed up response to interrupts.

After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW. This register is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: BPSW_TR (only accessible in the root world)
- For the secure world: BPSW_S (only accessible in the secure world or root world)
- For the non-secure world: BPSW_NS (only accessible in the non-secure world or root world)

In the case of an attempt to write a value to bit 28 in BPSW_NS, 0 is written to the bit.

2.2.2.7 Fast Interrupt Vector Register (FINTV)



The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.

The FINTV register specifies a branch destination address when a fast interrupt has been generated. This register is only accessible in the root world.

2.2.2.8 Single-Precision Floating-Point Status Word (FPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Single-Precision Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W
b31	FS	Single-Precision Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the EV bit is set to 0, the FV flag is enabled.
Note 4. When the EO bit is set to 0, the FO flag is enabled.
Note 5. When the EZ bit is set to 0, the FZ flag is enabled.
Note 6. When the EU bit is set to 0, the FU flag is enabled.
Note 7. When the EX bit is set to 0, the FX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is set to 0 by software.

The single-precision floating-point status word (FPSW) indicates the results of single-precision floating-point arithmetic operations.
When the corresponding exception handling enable bits (Ej) are set to enable processing of the exceptions (Ej = 1), the Cj flags can be used by the exception handling routine to identify the source of that exception. If handling of an exception is masked (Ej = 0), the Fj flag can be used to check for the generation of the exception at the end of a sequence of processing. The Fj flags operate in an accumulative fashion (j = X, U, Z, O, or V).

RM[1:0] Bits (Single-Precision Floating-Point Rounding-Mode Setting)

These bits specify the single-precision floating-point rounding-mode.

Explanation of Single-Precision Floating-Point Rounding Modes

- Rounding towards the nearest value (the default behavior): An inexact result is rounded to the available value that is closest to the result of a hypothetical calculation with infinite precision. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0: An inexact result is rounded to the smallest available absolute value, i.e. in the direction of zero (simple truncation).
- Rounding towards +∞: An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards –∞: An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

CV Flag (Invalid Operation Cause Flag), CO Flag (Overflow Cause Flag), CZ Flag (Division-by-Zero Cause Flag), CU Flag (Underflow Cause Flag), CX Flag (Inexact Cause Flag), and CE Flag (Unimplemented Processing Cause Flag)

Single-precision floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further single-precision floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (CE) is set to 1.

- If an exception or processing that is not implemented is not encountered in the execution of a single-precision floating-point arithmetic instruction, the corresponding flags become 0.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Bit (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number. When this bit is set to 1, a denormalized number is handled as 0.

EV Bit (Invalid Operation Exception Enable), EO Bit (Overflow Exception Enable), EZ Bit (Division-by-Zero Exception Enable), EU Bit (Underflow Exception Enable), and EX Bit (Inexact Exception Enable)

When any of five single-precision floating-point exceptions specified in the IEEE754 standard is generated by the single-precision floating-point operation instruction, the bit decides whether the CPU will start handling the exception. When the bit is set to 0, the exception handling is masked; when the bit is set to 1, the exception handling is enabled.

FV Flag (Invalid Operation Flag), FO Flag (Overflow Flag), FZ Flag (Division-by-Zero Flag), FU Flag (Underflow Flag), and FX Flag (Inexact Flag)

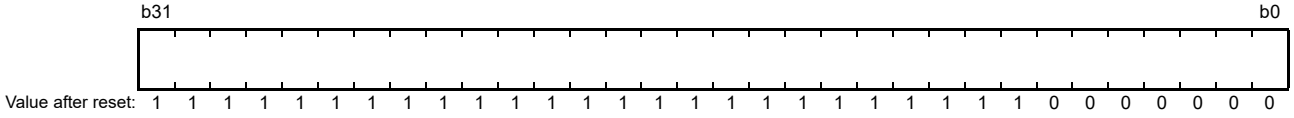
While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five single-precision floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is set to 0 by software (accumulation flag).

FS Flag (Single-Precision Floating-Point Error Summary Flag)

This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.2.9 Exception Table Register (EXTB)



The exception table register (EXTB) specifies the address where the exception vector table starts. This register is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: EXTB_TR (only accessible in the root world)
- For the secure world: EXTB_S (only accessible in the secure world or root world)
- For the non-secure world: EXTB_NS (only accessible in the non-secure world or root world)

The initial value depends on the world.

- EXTB_TR, EXTB_S: FFFF FF80h
- EXTB_NS: 7FFF FF80h

2.2.2.10 Interrupt Vector Control Register (IVC)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	EXCSE	—	—	—	NSIC	—	—	—	—	—	—	UIR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BWINM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BWINM[7:0]	Backup Interrupt Source Number for World Transition	These bits hold the source number of an interrupt accepted through the world-transition acceptance method.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	UIR[1:0]	Software Interrupt Restriction	These bits control the range of interrupt numbers that can be specified as the operand in an INT instruction to be executed in user mode. A number outside this range being specified as the operand will cause a privileged instruction exception. In supervisor mode, a number from 0 to 255 can be specified as the operand regardless of the setting in these bits. • For INT #imm:8 b17 b16 0 0: A number from 0 to 255 can be specified. 0 1: A number from 0 to 15 can be specified. 1 x: Setting prohibited • For INT Rs b17 b16 0 0: A number from 0 to 255 can be specified. 0 1: Software interrupts cannot be executed. 1 x: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	NSIC	Non-secure Interrupt Enable Control	This bit controls the function of the PSW.I bit in the non-secure world. 0: Interrupts are disabled while the PSW.I bit is 0. 1: Interrupts up to priority level 15 are disabled while the PSW.I bit is 0.	R/W *1
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	EXCSE	Secure Exception Detection during Hardware Pre-processing	This bit is set if a secure exception occurs during hardware pre-processing. 0: A secure exception has not occurred during hardware pre-processing. 1: A secure exception has occurred during hardware pre-processing.	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In the IVC_NS and IVC_S registers, this bit is always read as 0.

This register is actually implemented as the following three registers to support the Protect Zone function.

- For the root world: IVC_TR (only accessible in the root world)
- For the secure world: IVC_S (only accessible in the secure world or root world)
- For the non-secure world: IVC_NS (only accessible in the non-secure world or root world)

These actual registers are only modifiable during execution in supervisor mode while in the root world.

BWINM[7:0] Bits (Backup Interrupt Source Number for World Transition)

When an interrupt accepted through the world-transition acceptance method is generated, the vector number in the

interrupt vector table is stored in the IVC_TR.BWINM[7:0] bits and in the IVC.BWINM[7:0] bits in the destination world of the interrupt. The same value is written to the BWINM[7:0] bits of both the IVC_TR register and IVC register in the destination world for the transition (that is, the IVC_S.BWINM[7:0] bits in the secure world or the IVC_NS.BWINM[7:0] bits in the non-secure world).

UIR[1:0] Bits (Software Interrupt Restriction)

These bits are used to restrict the interrupt numbers that can be specified as the operand in an INT instruction to be executed in user mode. This setting is ignored in supervisor mode.

NSIC Bit (Non-secure Interrupt Enable Control)

This bit is used to control the function of the PSW.I bit for interrupts with a priority level 16 or higher in the non-secure world. When the NSIC bit is set to 1, interrupts with a priority level of 16 or higher are accepted regardless of the setting of the PSW.I bit. Interrupts with a priority level up to 15 are only accepted while the setting of the PSW.I bit is 1.
When the NSIC bit is set to 0, interrupts, whatever their priority level, are only accepted while the PSW.I bit is 1.

EXCSE Bit (Secure Exception Detection during Hardware Pre-processing)

This bit indicates that a secure exception has been detected during hardware pre-processing.
If a secure exception occurs during hardware pre-processing, the pre-processing is aborted and the secure exception is accepted. In this case, execution cannot be returned from the processing of the secure exception by using the TRRTE instruction. This bit can be used to identify such situations.

2.2.2.11 World ID Register (WID)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BWI	—	—	—	—	—	—	—	—	—	—	—	—	—	BITW[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BWID[1:0]		—	—	—	—	—	—	WID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	WID[1:0]	World ID	These bits hold the ID of the world in which execution is currently in progress.	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	BWID[1:0]	Backup World ID	These bits hold the world ID that was saved as the backup in a transition to root world.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	BITW[1:0]	Backup Destination World ID	These bits hold the destination world ID for a backup interrupt (with world-transition acceptance) or non-maskable interrupt (with world-transition acceptance).	R
b30 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	BWI	Backup Interrupt Enable for Source World of Transition	This bit holds the value of the interrupt enable bit (PSW.I) specified in the source world of a transition for an unconditional trap (with world-transition acceptance).	R

Note 1. The bit is read as 0 during execution in the secure world or non-secure world.

The following world IDs are defined.

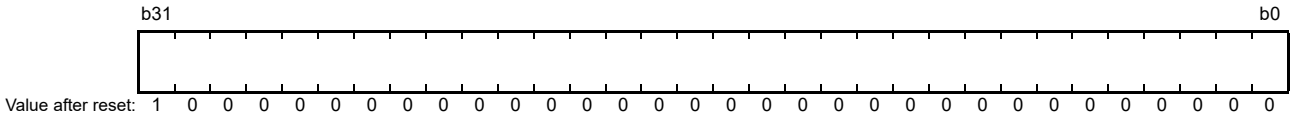
00: Root world

01: Secure world

10: Non-secure world

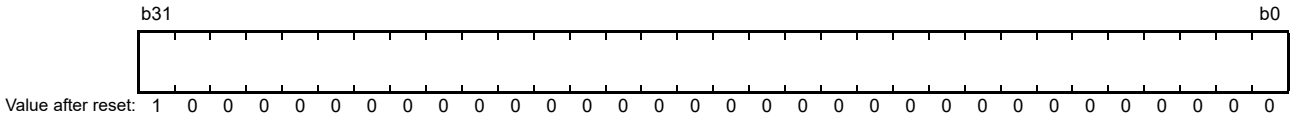
This register is only modifiable during execution in supervisor mode while in the root world. It is read-only during execution in the secure world or non-secure world.

2.2.2.12 Root Call Vector Table Register (TRCTB)



Use the root call vector table register (TRCTB) to specify the first address of the root call vector table to be used for unconditional traps accepted through the world-transition acceptance method. The initial value of this register is 8000 0000h. This register is only accessible during execution in supervisor mode while in the root world.

2.2.2.13 Vector Table Register for Interrupts that Cause World Transitions (WINTB)



Use the vector table register for interrupts that cause world transitions (WINTB) to specify the first address of the vector table for interrupts that cause world transitions to be used for interrupts accepted through the world-transition acceptance method. The initial value of this register is 8000 0000h. This register is only accessible during execution in supervisor mode while in the root world.

2.2.2.14 World-Shared Processor Interrupt Priority Level Register (WPRI)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	WIPL[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28 to b24	WIPL[4:0]	World-Shared Processor Interrupt Priority Level	b28 b24 0 0 0 0 0: Priority level 0 (lowest) 0 0 0 0 1: Priority level 1 0 0 0 1 0: Priority level 2 0 0 0 1 1: Priority level 3 0 0 1 0 0: Priority level 4 0 0 1 0 1: Priority level 5 0 0 1 1 0: Priority level 6 0 0 1 1 1: Priority level 7 0 1 0 0 0: Priority level 8 0 1 0 0 1: Priority level 9 0 1 0 1 0: Priority level 10 0 1 0 1 1: Priority level 11 0 1 1 0 0: Priority level 12 0 1 1 0 1: Priority level 13 0 1 1 1 0: Priority level 14 0 1 1 1 1: Priority level 15 (non-secure, highest) 1 0 0 0 0: Priority level 16 1 0 0 0 1: Priority level 17 1 0 0 1 0: Priority level 18 1 0 0 1 1: Priority level 19 1 0 1 0 0: Priority level 20 1 0 1 0 1: Priority level 21 1 0 1 1 0: Priority level 22 1 0 1 1 1: Priority level 23 1 1 0 0 0: Priority level 24 1 1 0 0 1: Priority level 25 1 1 0 1 0: Priority level 26 1 1 0 1 1: Priority level 27 1 1 1 0 0: Priority level 28 1 1 1 0 1: Priority level 29 1 1 1 1 0: Priority level 30 1 1 1 1 1: Priority level 31 (secure/root, highest)	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is only readable and writable in the secure or root world.
This register is only writable in supervisor mode.

WIPL[4:0] Bits (World-Shared Processor Interrupt Priority Level)

The WIPL[4:0] bits specify the world-shared processor interrupt priority level as one of 32 levels from 0 to 31, wherein priority level 0 is the lowest and priority level 31 the highest.
When the priority level of a requested interrupt is higher than the current processor interrupt priority level specified in PSW.IPL[4:0] and higher than the world-shared processor interrupt priority level specified in WPRI.WIPL[4:0], the interrupt is enabled.

2.2.3 Accumulator

The accumulator (ACC0 or ACC1) is a 72-bit register used for DSP instructions. The accumulator is handled as a 96-bit register for reading and writing. At this time, when bits 95 to 72 of the accumulator are read, the value where the value of bit 71 is sign extended is read. Writing to bits 95 to 72 of the accumulator is ignored. ACC0 is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in ACC0 is modified by execution of the instruction.

Use the MVTACGU, MVTACHI, and MVTACLO instructions for writing to the accumulator. The MVTACGU, MVTACHI, and MVTACLO instructions write data to the guard bits (bits 95 to 64), the higher-order 32 bits (bits 63 to 32), and the lower-order 32 bits (bits 31 to 0), respectively.

Use the MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions for reading data from the accumulator. The MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions read data from the guard bits (bits 95 to 64), higher-order 32 bits (bits 63 to 32), the middle 32 bits (bits 47 to 16), and the lower-order 32 bits (bits 31 to 0), respectively.

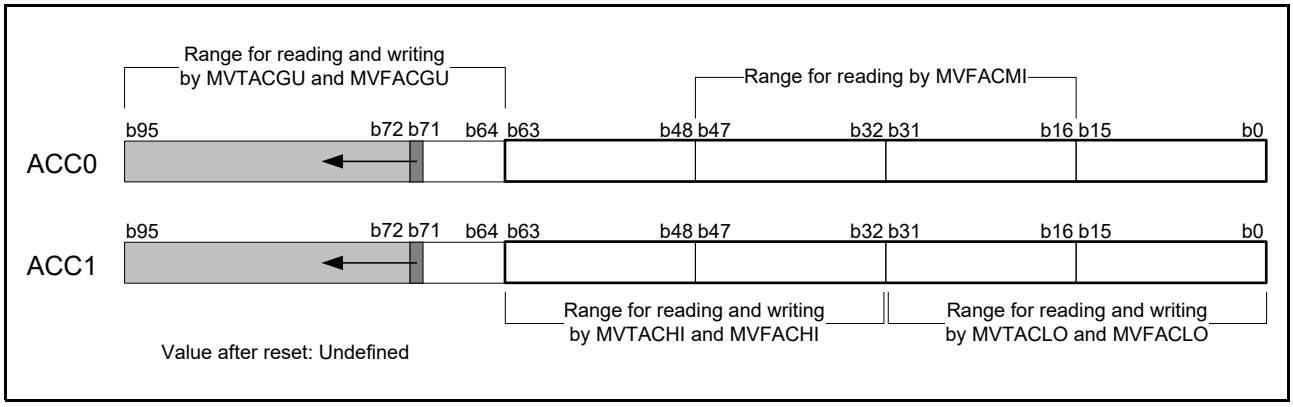


Figure 2.2 Accumulator

Note: When bits 95 to 72 of the accumulator are read, the value is the sign-extended value of bit 71. Writing to this area is ignored.

2.3 Processor Mode

The CPU supports two processor modes, supervisor and user. These processor modes and the memory protection function enable the realization of a hierarchical CPU resource protection and memory protection mechanism. Each processor mode imposes a level on rights of access to memory and the instructions that can be executed. Supervisor mode carries greater rights than user mode. The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

In supervisor mode, all CPU resources that are allowed in the world in which execution is in progress are accessible and all instructions that are allowed in the world in which execution is in progress can be executed. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.4, Processor Status Word (PSW).

2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[4:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)
- Exception table register (EXTB)
- Interrupt vector control register (IVC)
- World ID register (WID)
- Root call vector table register (TRCTB)
- Vector table register for interrupts that cause world transitions (WINTB)
- World-shared processor interrupt priority level register (WPRI)

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, WAIT, TRCALL, TRMVFC, TRMVTC, TRRTE, SAVE, and RSTR instructions. If the INT instruction is executed in user mode with a number outside the specified allowable range, a privileged instruction exception occurs. For details, see the description of the IVC.UIR bit.

2.3.4 Switching Between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC, TRMVTC, or POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

(1) Switching from user mode to supervisor mode

After an exception has been generated, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the PSW.PM bit that is saved on the stack.

(2) Switching from supervisor mode to user mode

Executing an RTE or TRRTE instruction when the value of the PSW.PM bit that has been saved on the stack is 1 or an RTFI instruction when the value of the PSW.PM bit that has been saved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the PSW.U bit) becomes 1.

2.4 Data Types

The CPU can handle four types of data: integer, single-precision floating-point number, bit, and string.
For details, refer to RX Family RXv3e Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

An integer can be signed or unsigned. For signed integers, negative values are represented by two's complements.

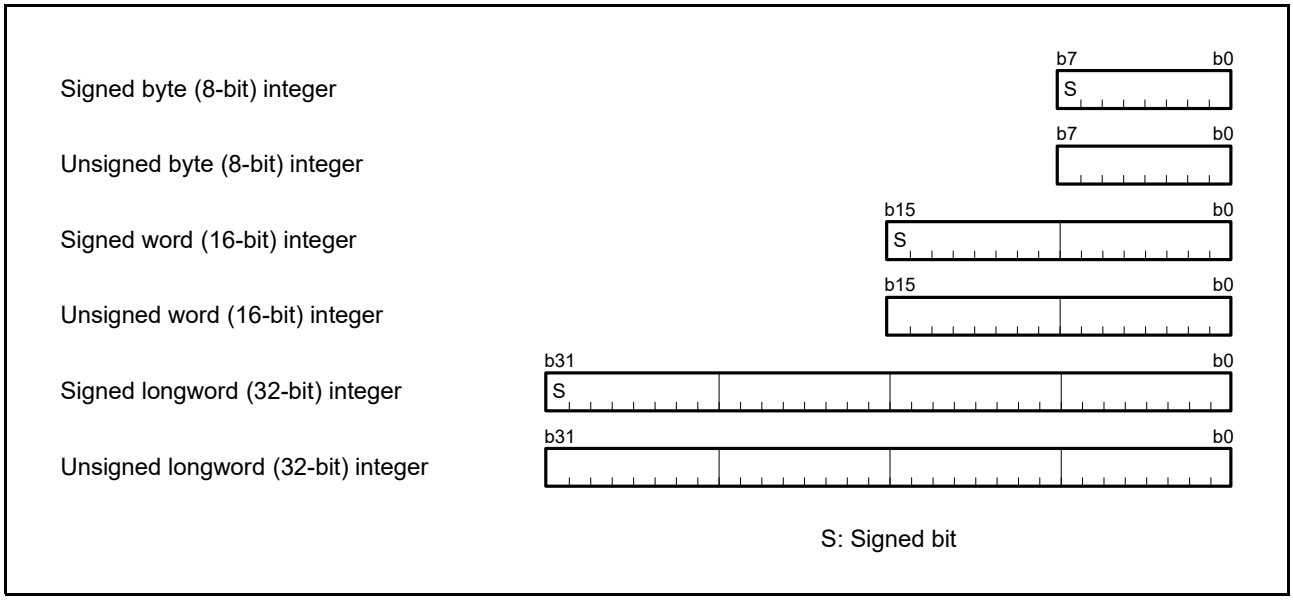


Figure 2.3 Integer

2.4.2 Single-Precision Floating-Point Numbers

The single-precision floating-point number is compliant with that specified in the IEEE754 standard; operands of this type can be used in eleven single-precision floating-point operation instructions: FADD, FCMP, FDIV, FMUL, FSQRT, FSUB, FTOI, FTOU, ITOF, ROUND, and UTOF.

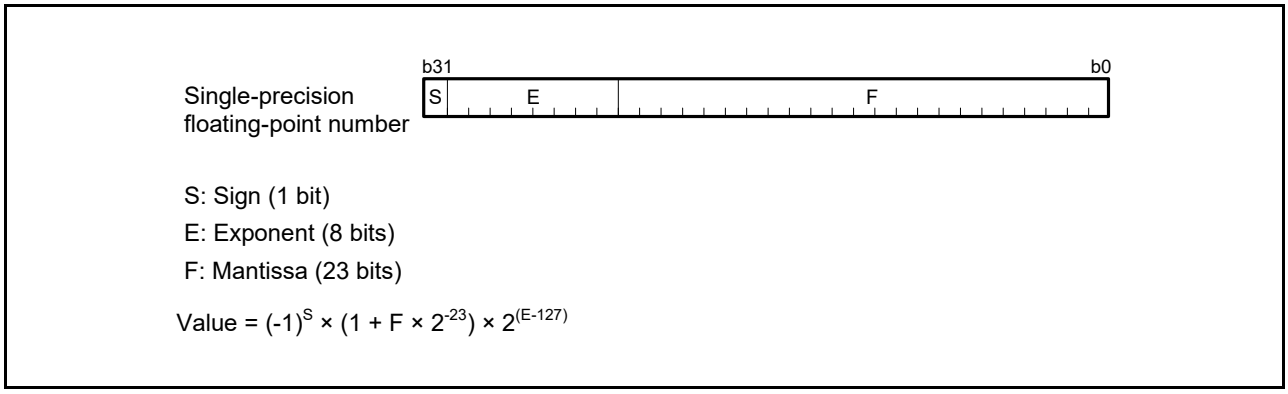


Figure 2.4 Single-Precision Floating-Point Number

The single-precision floating-point number can represent the values listed below.

- 0 < E < 255 (normal numbers)
- E = 0 and F = 0 (signed zero)
- E = 0 and F > 0 (denormalized numbers)*1
- E = 255 and F = 0 (infinity)
- E = 255 and F > 0 (NaN: Not-a-Number)

Note 1. The number is treated as 0 when the FPSW.DN bit is 1. When the DN bit is 0, an unimplemented processing exception is generated.

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BM*Cnd*, BNOT, BSET, and BTST.
A bit in a register is specified as the destination register and a bit number in the range from 31 to 0.
A bit in memory is specified as the destination address and a bit number from 7 to 0. The addressing modes available to specify addresses are register indirect and register relative.

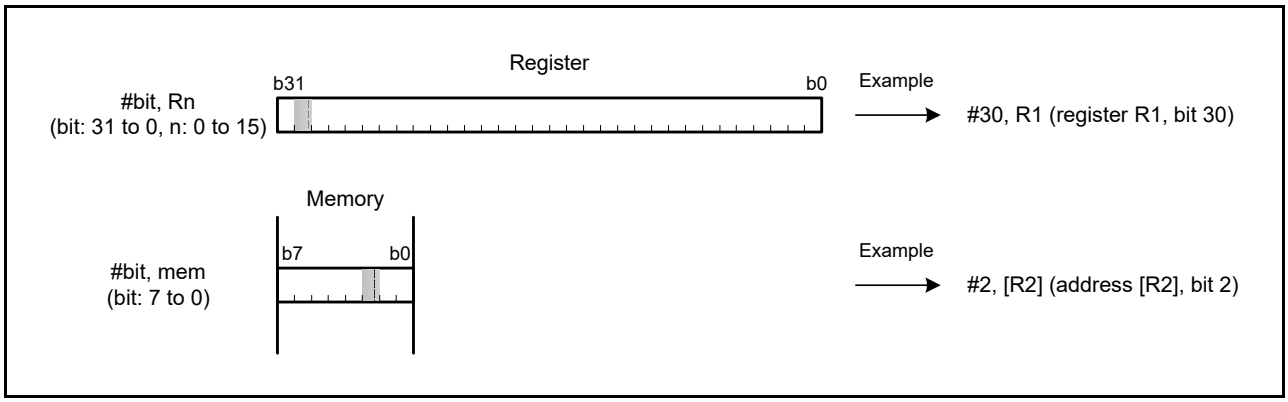


Figure 2.5 Bit

2.4.4 Strings

The string data type consists of an arbitrary number of consecutive byte (8-bit), word (16-bit), or longword (32-bit) units. Seven string manipulation instructions are provided for use with strings: SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE.

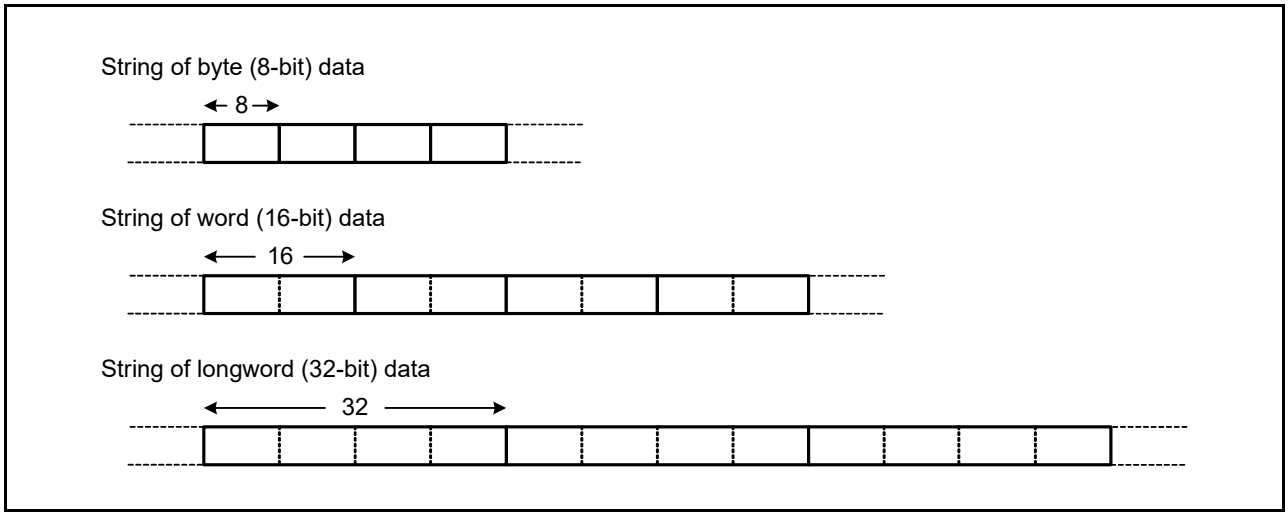


Figure 2.6 String

2.5 Endian

For the CPU, instructions are little endian, but the treatment of data is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, see section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.2 to Table 2.13.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

	D31 to D24	D23 to D16	D15 to D8	D7 to D0
General purpose register: Rm	HH	HL	LH	LL

Table 2.2 32-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to LL	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—
Address 2	Transfer to HL	Transfer to LH	Transfer to LL	—	—
Address 3	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL	—
Address 4	—	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL
Address 5	—	—	Transfer to HH	Transfer to HL	Transfer to LH
Address 6	—	—	—	Transfer to HH	Transfer to HL
Address 7	—	—	—	—	Transfer to HH

Table 2.3 32-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to HH	—	—	—	—
Address 1	Transfer to HL	Transfer to HH	—	—	—
Address 2	Transfer to LH	Transfer to HL	Transfer to HH	—	—
Address 3	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH	—
Address 4	—	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH
Address 5	—	—	Transfer to LL	Transfer to LH	Transfer to HL
Address 6	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	Transfer to LL

Table 2.4 32-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from LL	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—
Address 2	Transfer from HL	Transfer from LH	Transfer from LL	—	—
Address 3	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL	—
Address 4	—	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL
Address 5	—	—	Transfer from HH	Transfer from HL	Transfer from LH
Address 6	—	—	—	Transfer from HH	Transfer from HL
Address 7	—	—	—	—	Transfer from HH

Table 2.5 32-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from HH	—	—	—	—
Address 1	Transfer from HL	Transfer from HH	—	—	—
Address 2	Transfer from LH	Transfer from HL	Transfer from HH	—	—
Address 3	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH	—
Address 4	—	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH
Address 5	—	—	Transfer from LL	Transfer from LH	Transfer from HL
Address 6	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	Transfer from LL

Table 2.6 16-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LL	—	—	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—	—	—
Address 2	—	Transfer to LH	Transfer to LL	—	—	—	—
Address 3	—	—	Transfer to LH	Transfer to LL	—	—	—
Address 4	—	—	—	Transfer to LH	Transfer to LL	—	—
Address 5	—	—	—	—	Transfer to LH	Transfer to LL	—
Address 6	—	—	—	—	—	Transfer to LH	Transfer to LL
Address 7	—	—	—	—	—	—	Transfer to LH

Table 2.7 16-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LH	—	—	—	—	—	—
Address 1	Transfer to LL	Transfer to LH	—	—	—	—	—
Address 2	—	Transfer to LL	Transfer to LH	—	—	—	—
Address 3	—	—	Transfer to LL	Transfer to LH	—	—	—
Address 4	—	—	—	Transfer to LL	Transfer to LH	—	—
Address 5	—	—	—	—	Transfer to LL	Transfer to LH	—
Address 6	—	—	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	—	—	Transfer to LL

Table 2.8 16-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.9 16-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LH	—	—	—	—	—	—
Address 1	Transfer from LL	Transfer from LH	—	—	—	—	—
Address 2	—	Transfer from LL	Transfer from LH	—	—	—	—
Address 3	—	—	Transfer from LL	Transfer from LH	—	—	—
Address 4	—	—	—	Transfer from LL	Transfer from LH	—	—
Address 5	—	—	—	—	Transfer from LL	Transfer from LH	—
Address 6	—	—	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	—	—	Transfer from LL

Table 2.10 8-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.11 8-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.12 8-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

Table 2.13 8-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of eight bits is indicated, use instructions having operands of the same width (eight bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.7 shows the relation between the sizes of registers and bit numbers.

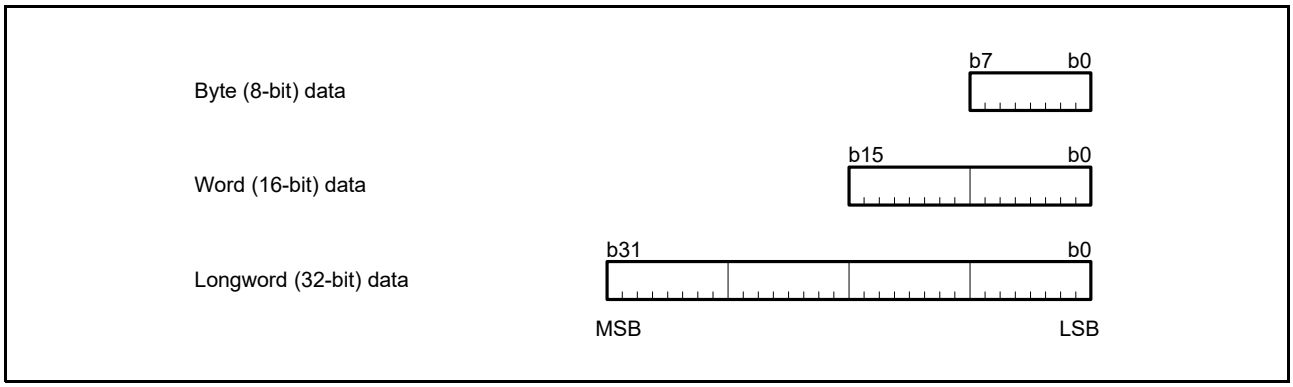


Figure 2.7 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.8 shows the arrangement of data in memory.

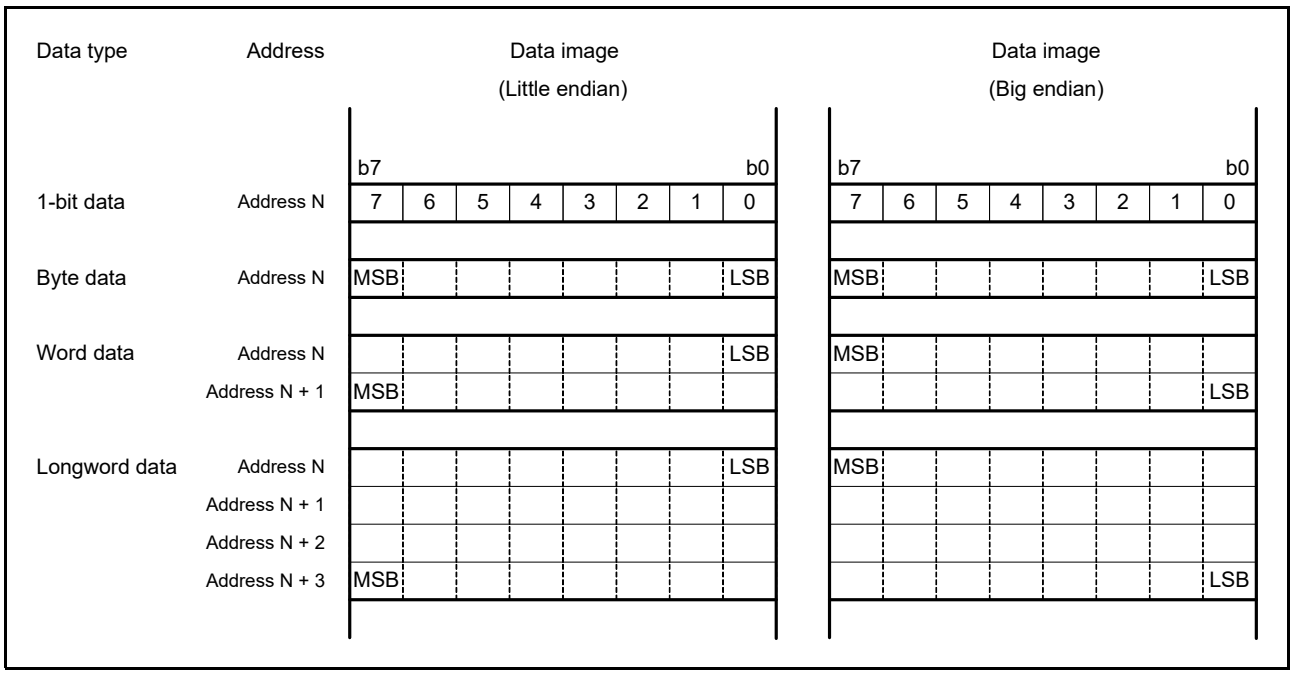


Figure 2.8 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external address space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external address space, they must be allocated to areas where the endian setting is the same as that for the chip.

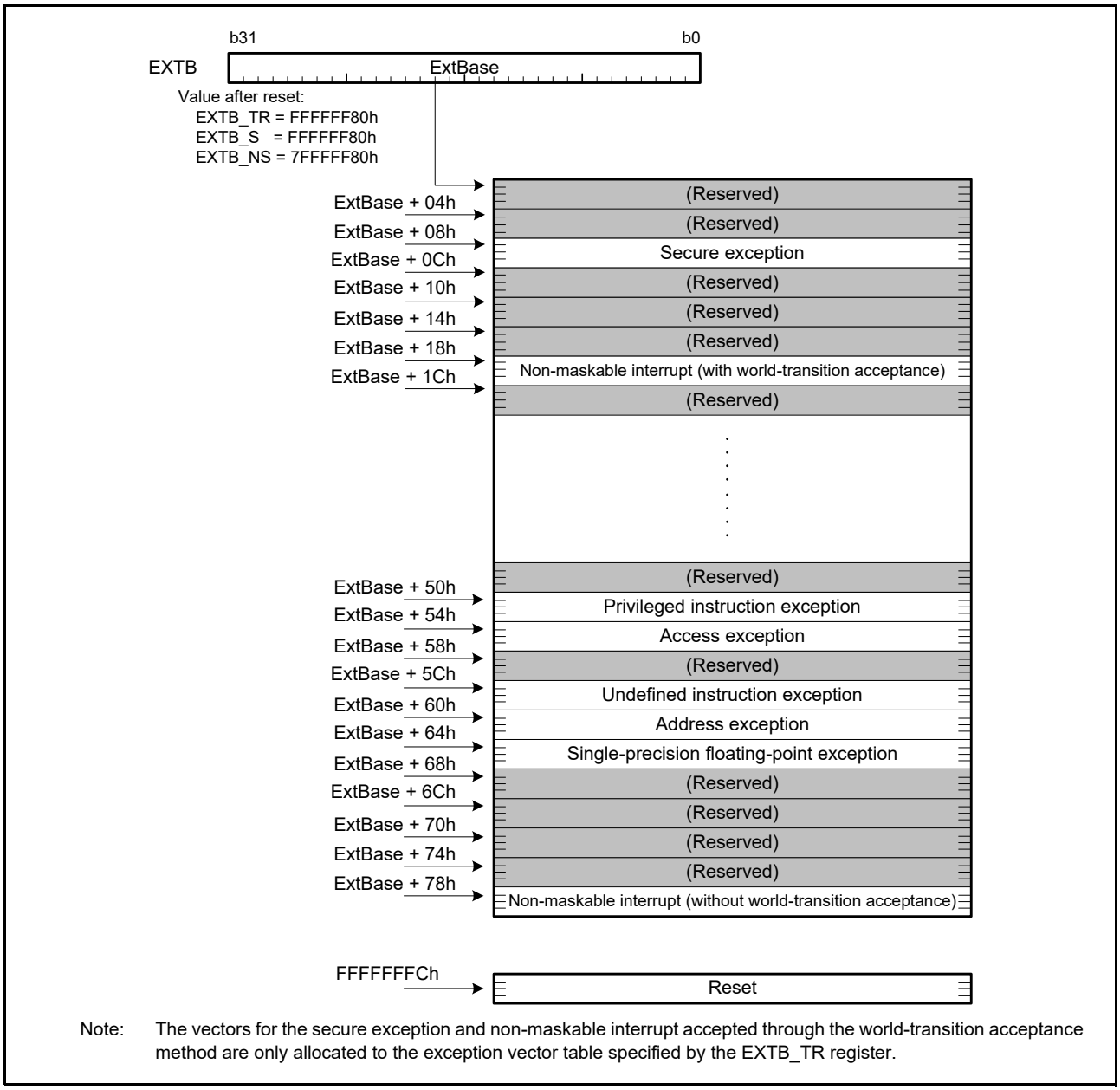
2.6 Vector Tables

There are four types of vector tables: exception vector table, interrupt vector table, root call vector table, and vector table for interrupt that cause world transitions. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Exception Vector Table

In the exception vector table, the individual vectors for the secure exceptions, non-maskable interrupt (with world-transition acceptance), privileged instruction exception, access exception, address exception, undefined instruction exception, single-precision floating-point exception, and non-maskable interrupt (with normal acceptance) are allocated to the 124-byte area where the value indicated by the exception table register (EXTB) is used as the starting address (ExtBase). The reset vector is always allocated to FFFFFFFCh, regardless of the value of the exception vector table register.

Figure 2.9 shows the exception vector table.



Note: The vectors for the secure exception and non-maskable interrupt accepted through the world-transition acceptance method are only allocated to the exception vector table specified by the EXTB_TR register.

Figure 2.9 Exception Vector Table

2.6.2 Interrupt Vector Table

The address where the interrupt vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.10 shows the interrupt vector table.

Each vector in the interrupt vector table has a vector number from 0 to 255. Vector numbers from 0 to 255 are assigned for the INT instruction, which generates an unconditional trap, and the vector corresponding to the INT instruction number is read from the vector table. Vector number 0 is assigned for the BRK instruction and the vector is always read from the entry with vector number 0 in the table. In addition, fixed vector numbers (0 to 255) are assigned to the individual interrupt sources and the assigned numbers vary with the product type. For more on interrupt vector numbers, see section 15.3.1, Interrupt Vector Table.

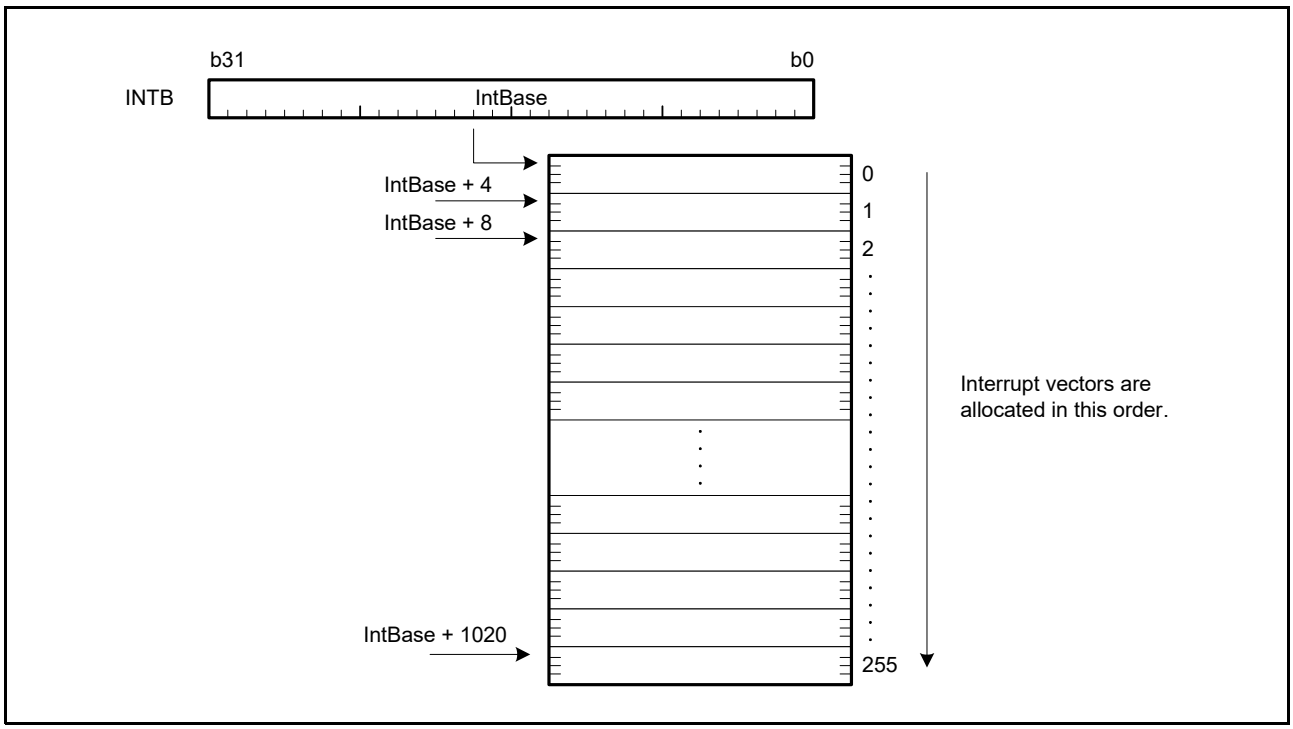


Figure 2.10 Interrupt Vector Table

2.6.3 Root Call Vector Table

The root call vector table is a 1024-byte region starting at the address (TRCallBase) specified in the root call vector table register (TRCTB). Each vector called by the TRCALL instruction is assigned to this vector table. Figure 2.11 shows the root call vector table.

Each vector in the root call vector table has a vector number from 0 to 255. Vector numbers 0 to 255 are assigned for the TRCALL instruction and the vector corresponding to the number specified in the operand of the instruction is read from the vector table.

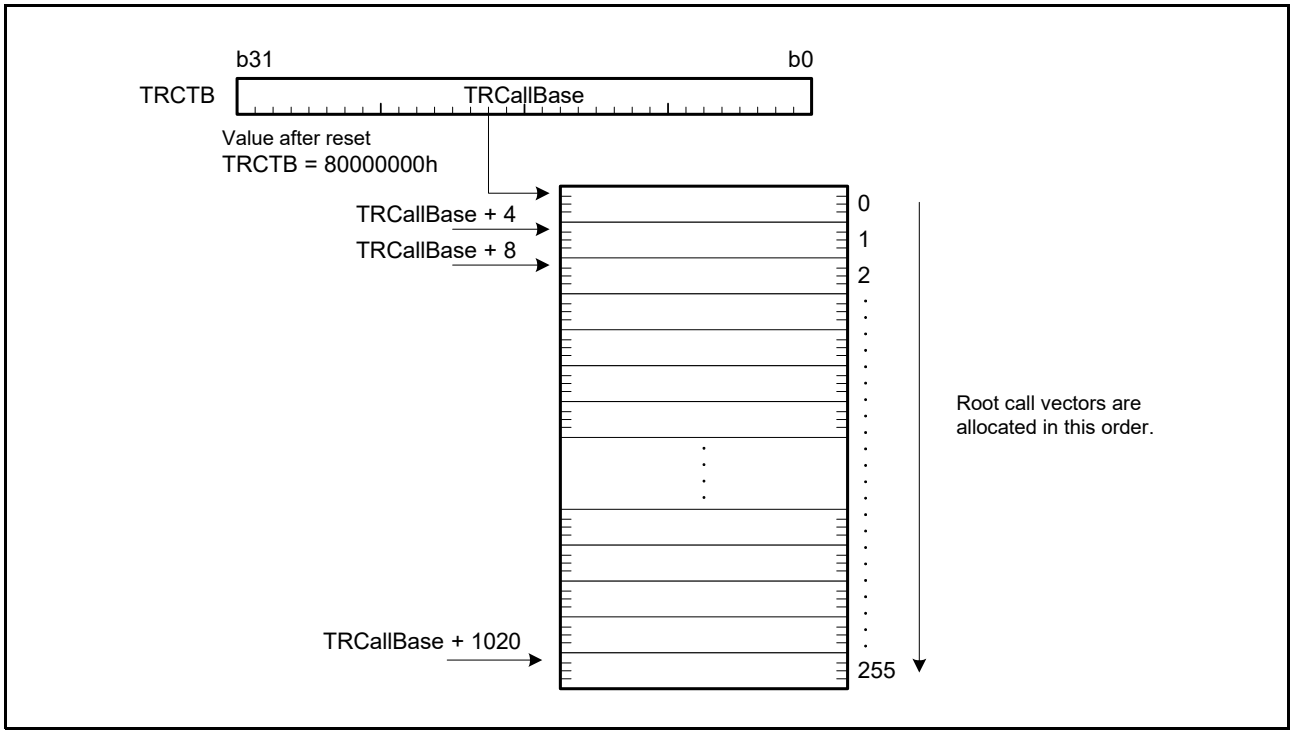


Figure 2.11 Root Call Vector Table

2.6.4 Vector Table for Interrupts that Cause World Transitions

The vector table for interrupts that cause world transition is a 12-byte region starting at the address (WIntBase) specified in the vector table register for interrupts that cause world transitions (WINTB). The individual vectors for the interrupts accepted through the world-transition acceptance method are allocated to this vector table. Figure 2.12 shows the vector table for interrupts that cause world transitions.

Each vector in this vector table has a vector number from 0 to 2. The vector number corresponds to the ID of the destination world for the interrupt accepted through the world-transition acceptance method.

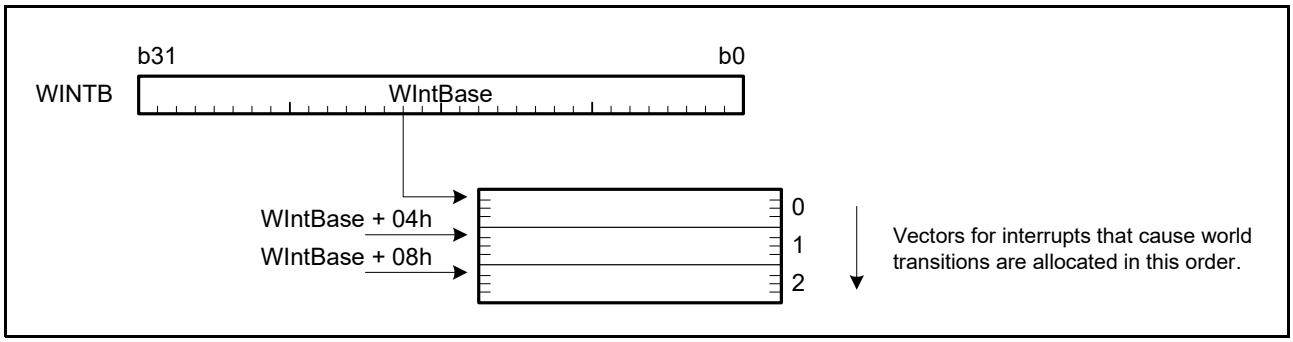


Figure 2.12 Vector Table for Interrupts that Cause World Transitions

2.7 Register Bank Save Function

The CPU has dedicated register-saving banks and functionality for using them for the fast saving and restoring of the values of CPU registers (see Figure 2.13). The register-saving banks enable the fast collective saving at the start of the exception handling routine and fast collective restoring of register values at the end of the exception handling routine. The register-saving banks are only accessible by the SAVE and RSTR instructions, and are independent of the 4-Gbyte address space. Each of the multiple banks is used to save and restore the values of the following CPU registers: all general purpose registers except R0, the USP, FPSW, and accumulators (ACC0, ACC1). Values in the register-saving banks are undefined after a reset.

A unique number (bank number) is allocated to each register-saving bank.

The MCU has 16 save register banks per world, to which the bank numbers 0 to 15 are assigned.

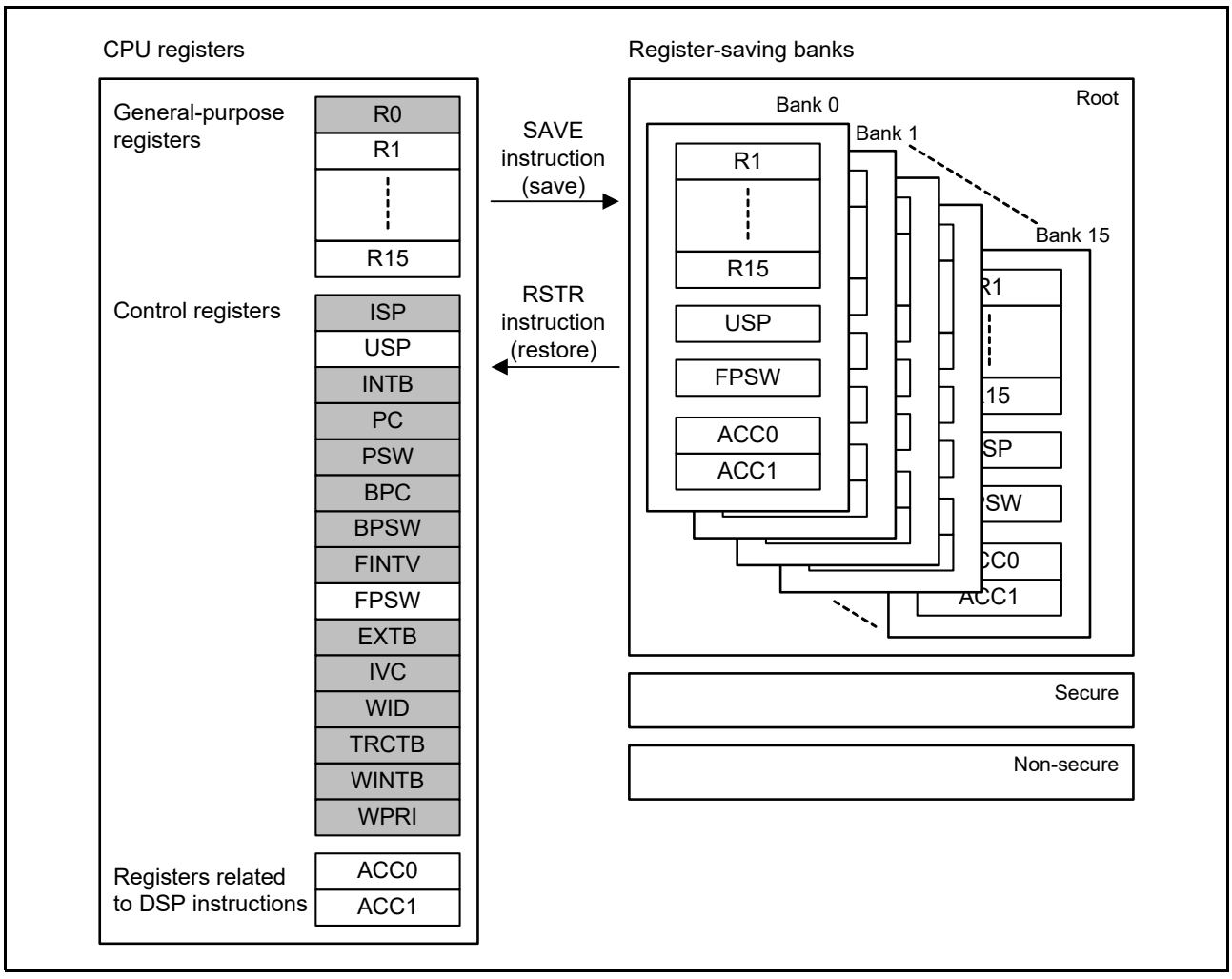


Figure 2.13 Register-Saving Banks

Each world has its own register-saving bank. There is a TR register-saving bank for root, an S register-saving bank for secure, and an NS register-saving bank for non-secure. The register-saving bank corresponding to the world in which the SAVE or RSTR instruction was executed is used.

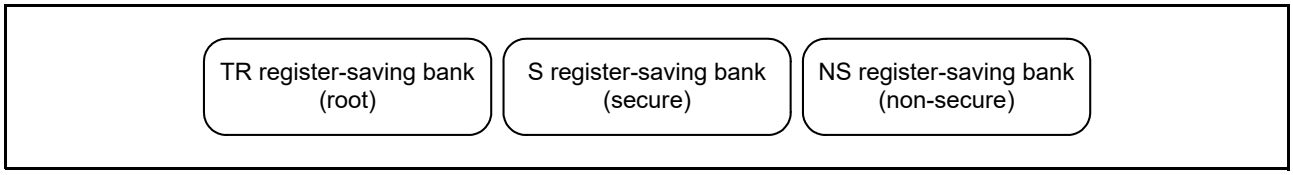


Figure 2.14 Configuration of Register-Saving Banks to Support the Protect Zone Function

2.8 Double-Precision Floating-Point Coprocessor

The double-precision floating-point coprocessor operates as a coprocessor of the CPU, and executes double-precision floating-point processing instructions. Using the double-precision floating-point coprocessor considerably accelerates the processing of double-precision floating point arithmetic.

2.8.1 Features

- Double-precision floating-point register set
  Double-precision floating-point data registers: Sixteen 64-bit registers
  Double-precision floating-point control registers: Four 32-bit registers
- Double-precision floating-point processing instructions: 21
- Notifying the interrupt controller of double-precision floating-point exceptions

2.8.2 Double-Precision Floating-Point Register Set

The double-precision floating-point coprocessor consists of 16 double-precision floating-point data registers and 4 double-precision floating-point control registers.

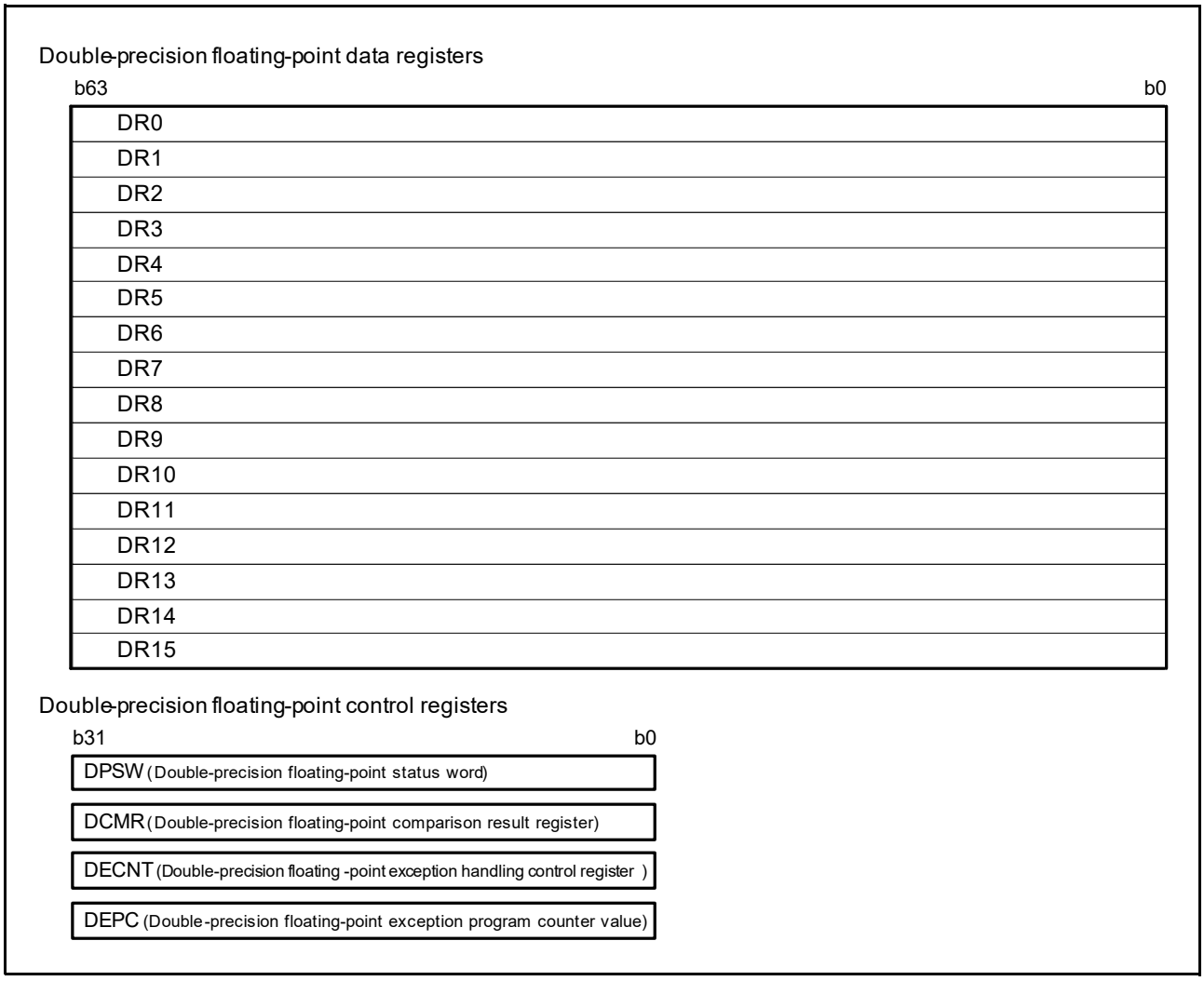


Figure 2.15 Double-Precision Floating-Point Register Set

2.8.2.1 Double-Precision Floating-Point Data Registers (DR0 to DR15)

16 double-precision floating-point data registers with 64-bit width are provided (DR0 to DR15). To specify 32-bit values, use the upper 32 bits (DRH0 to DRH15) or lower 32 bits (DRL0 to DRL15) as separate units.

2.8.2.2 Double-Precision Floating-Point Control Registers

Four double-precision floating-point control registers are provided.

- Double-precision floating-point status word (DPSW)
- Double-precision floating-point comparison result register (DCMR)
- Double-precision floating-point exception handling control register (DECNT)
- Double-precision floating-point exception program counter (DEPC)

(1) Double-Precision Floating-Point Status Word (DPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFS	DFX	DFU	DFZ	DFO	DFV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DEX	DEU	DEZ	DEO	DEV	—	DDN	DCE	DCX	DCU	DCZ	DCO	DCV	DRM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DRM[1:0]	Double-Precision Floating-Point Rounding-Mode Setting	b1 b0 0 0: Round to the nearest value 0 1: Round towards 0 1 0: Round towards +∞ 1 1: Round towards –∞	R/W
b2	DCV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W)*1
b3	DCO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W)*1
b4	DCZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W)*1
b5	DCU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W)*1
b6	DCX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W)*1
b7	DCE	Unimplemented Processing Cause Flag	0:No unimplemented processing has been encountered. 1:Unimplemented processing has been encountered.	R/(W)*1
b8	DDN	0 Flush Bit of Denormalized Number	0:A denormalized number is handled as a denormalized number. 1:A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	DEV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	DEO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	DEZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	DEU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	DEX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	DFV	Invalid Operation Flag*3	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	DFO	Overflow Flag*4	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	DFZ	Division-by-Zero Flag*5	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	DFU	Underflow Flag*6	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	DFX	Inexact Flag*7	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W

Bit	Symbol	Bit Name	Description	R/W
b31	DFS	Double-Precision Floating-Point Error Summary Flag	This bit reflects the logical OR of the DFU, DFZ, DFO, and DFV flags.	R

Note 1. When 0 is written to the bit, the setting of the bit will be 0; the bit retains the previous value in response to the writing of 1.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the DEV bit is set to 0, the DFV flag is enabled.
Note 4. When the DEO bit is set to 0, the DFO flag is enabled.
Note 5. When the DEZ bit is set to 0, the DFZ flag is enabled.
Note 6. When the DEU bit is set to 0, the DFU flag is enabled.
Note 7. When the DEX bit is set to 0, the DFX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is set to 0 by software.

The double-precision floating-point status word (DPSW) indicates the results of double-precision floating-point arithmetic operations.
When the corresponding exception handling enable bits (DEj) are set to enable processing of the exceptions (DEj = 1), the DCj flags can be used by the exception handling routine (the interrupt handling routine with the double-precision floating-point exception as its source) to identify the source of that exception. If handling of an exception is masked (DEj = 0), the DFj flag can be used to check for the generation of the exception at the end of a sequence of processing. The DFj flags operate in an accumulative fashion (j =X, U, Z, O, or V).
The single-precision floating-point status word (FPSW) is neither referred to nor updated in double-precision floating-point arithmetic operations.

DRM[1:0] Bits (Double-Precision Floating-Point Rounding-Mode Setting)

These bits specify the double-precision floating-point rounding-mode.

Explanation of Double-Precision Floating-Point Rounding Modes

- Rounding to the nearest value (the default behavior): An inexact result is rounded to the available value that is closest to the result of a hypothetical calculation with infinite precision. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0: An inexact result is rounded to the smallest available absolute value; i.e., in the direction of zero (simple truncation).
- Rounding towards +∞: An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards –∞: An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

DCV Flag (Invalid Operation Cause Flag), DCO Flag (Overflow Cause Flag), DCZ Flag (Division-by-Zero Cause Flag), DCU Flag (Underflow Cause Flag), DCX Flag (Inexact Cause Flag), and DCE Flag (Unimplemented Processing Cause Flag)

Double-precision floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further double-precision floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (DCE) is set to 1.

- If an exception or processing that is not implemented is not encountered in the execution of a double-precision floating-point arithmetic instruction other than DABS or DNEG, the corresponding flags become 0.
- When 0 is written to the bit, the setting of the bit will be 0; the bit retains the previous value in response to the writing of 1.

DDN Bit (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number.
When this bit is set to 1, a denormalized number is handled as 0.

DEV Bit (Invalid Operation Exception Enable), DEO Bit (Overflow Exception Enable), DEZ Bit (Division-by-Zero Exception Enable), DEU Bit (Underflow Exception Enable), and DEX Bit (Inexact Exception Enable)

When any of the five floating-point exceptions specified in the IEEE754 standard is generated in the execution of a double-precision floating-point operation instructions, these bits determine whether the CPU will start handling the exception (interrupt information will be held by setting the DECNT.EHS bit and an interrupt request will be sent to the interrupt controller).

When the bit corresponding to an exception is 0, the DECNT.EHS bit is not set to 1. When the bit corresponding to an exception is 1, the DECNT.EHS bit is set to 1. However, if the DECNT.EHM bit is 0, the DECNT.EHS bit will not be set to 1 regardless of the value of the bit corresponding to the exception.

DFV Flag (Invalid Operation Flag), DFO Flag (Overflow Flag), DFZ Flag (Division-by-Zero Flag), DFU Flag (Underflow Flag), and DFX Flag (Inexact Flag)

While the exception handling enable bit (DEj) is 0 (exception handling is masked), if any of five floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

- When DEj is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is set to 0 by software (Accumulation flag).

DFS Flag (Double-Precision Floating-Point Error Summary Flag)

This bit reflects the logical OR of the DFU, DFZ, DFO, and DFV flags.

The value of this register is not updated when the EHM and EHS bits in the DECNT register are 1.

(2) Double-Precision Floating-Point Comparison Result Register (DCMR)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RES	Double-Precision Floating-Point Compare Instruction Result Flag	0: Condition for comparison was not satisfied. 1: Condition for comparison was satisfied.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(3) Double-Precision Floating-Point Exception Handling Control Register (DECNT)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EHS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EHM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EHM	Double-Precision Floating-Point Exception Information Saving Mode	0: Mode in which information on the sources of exceptions is not saved. Information on the generation of double-precision floating-point exceptions is not saved when they occur. No interrupt request is sent to the interrupt controller. 1: Mode in which information on the generation of exceptions is saved. When a double-precision floating-point exception is generated and an interrupt request is sent to the interrupt controller, the EHS bit is changed to 1 to save information at the time the exception was generated, and the interrupt request is sent to the interrupt controller.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EHS	Double-Precision Floating-Point Exception Information Saving Status	0: Exception information is not being saved. 1: Exception information is being saved. Updating of the following registers stops when the EHM and EHS bits are 1, and interrupts due to double-precision floating-point exceptions do not occur. • Double-precision floating-point status word • Double-precision floating-point exception program counter In addition, an interrupt request is sent to the interrupt controller. Writing 0 to this bit releases the information recorded at the time the exception was generated from the retained state.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(4) Double-Precision Floating-Point Exception Program Counter (DEPC)

	b31	b0
Value after reset:	Undefined	

When a double-precision floating-point exception is generated and the corresponding DEX, DEU, DEZ, DEO, or DEV bit of the DPSW is 1, the value of the program counter for the instruction that has caused the most recent double-precision floating-point exception is saved in this register. This register is read-only. The value of this register is not updated when the value of the DECNT.EHS bit is 1 while that of the DECNT.EHM bit is also 1.

2.8.3 Data Types (for the Double-Precision Floating-Point Coprocessor)

The double-precision floating-point coprocessor can handle double-precision floating-point numbers.
In addition, it can handle single-precision floating-point numbers with the DTOF and FTOD instructions, and 32-bit integers with the DROUND, DTOI, DTOU, ITOD and UTOD instructions.
For single-precision floating-point numbers, refer to section 2.4.2, Single-Precision Floating-Point Numbers, and for 32-bit integers, refer to section 2.4.1, Integer.

2.8.3.1 Double-Precision Floating-Point Numbers

The double-precision floating-point number is compliant with that specified in the IEEE754 standard. Operands of this type can be used in fifteen double-precision floating-point arithmetic instructions: DABS, DADD, DCMPcm, DDIV, DMUL, DNEG, DROUND, DSUB, DSQRT, DTOF, DTOI, DTOU, FTOD, ITOD, and UTOD.



Figure 2.16 Double-Precision Floating-Point Number

The double-precision floating-point number can represent the values listed below.
0 < E < 2047 (normal numbers)
E = 0 and F = 0 (signed zero)
E = 0 and F > 0 (denormalized numbers)*1
E = 2047 and F = 0 (infinity (∞))
E = 2047 and F > 0 (Not-a-Number (NaN))
Note 1. The number is treated as 0 when the DPSW.DDN bit is 1. When the DDN bit is 0, an unimplemented processing exception is generated.

2.8.4 Data Arrangement (for the Double-Precision Floating-Point Coprocessor)

The double-precision floating-point coprocessor can handle 32-bit and 64-bit values.
For 32-bit values, refer to section 2.5.4, Data Arrangement, and section 2.5.1, Switching the Endian.

2.8.4.1 Arrangement of Data in the Double-Precision Floating-Point Registers

Figure 2.17 shows the relationship between the sizes of registers DR0 to DR15 and the bit numbers.

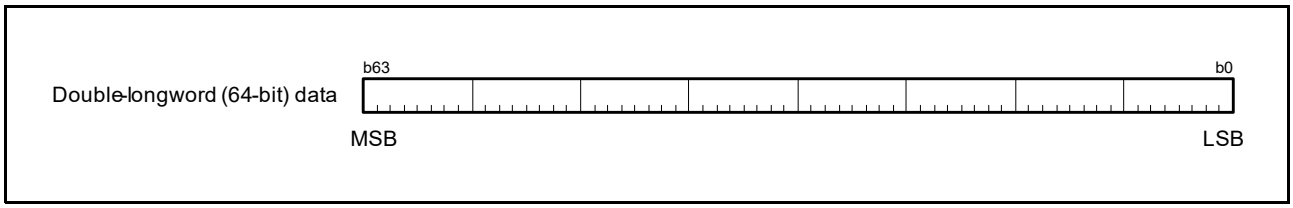


Figure 2.17 Data Arrangement in Registers

2.8.4.2 Arrangement of Data for Double-Precision Floating-Point Numbers in Memory

A double-precision floating-point number is always represented as a double-longword (64 bits) in memory. The data arrangement is selectable as little endian or big endian. Figure 2.18 shows the arrangement of data in memory.

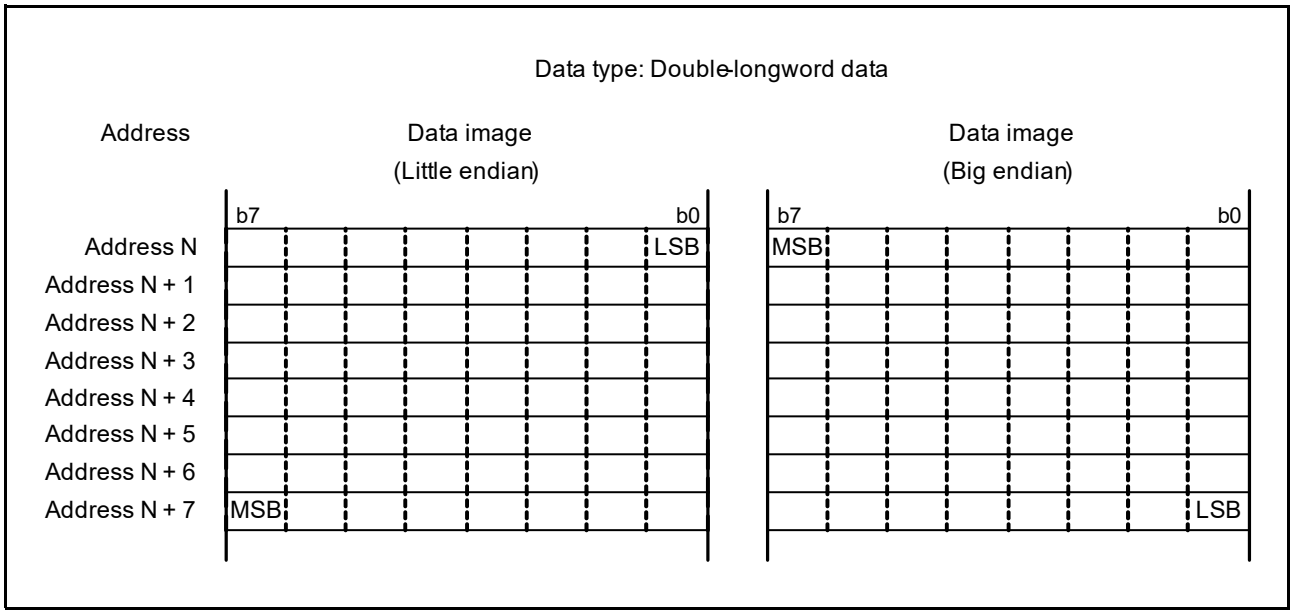


Figure 2.18 Data Arrangement in Memory

2.8.5 Switching the Endian (for the Double-Precision Floating-Point Coprocessor)

The operation in access with 64-bit values by double-precision floating-point processing instructions depends on the endian setting. Table 2.14 to Table 2.17 show operations in access for each endian and in the cases of reading and writing.

Address exceptions are generated in response to 64-bit access to addresses that are not on 32-bit boundaries.

In the tables,

LLL indicates bits D7 to D0 of the double-precision floating-point data registers,
LLH indicates bits D15 to D8 of the double-precision floating-point data registers,
LHL indicates bits D23 to D16 of the double-precision floating-point data registers,
LHH indicates bits D31 to D24 of the double-precision floating-point data registers,
HLL indicates bits D39 to D32 of the double-precision floating-point data registers,
HLH indicates bits D47 to D40 of the double-precision floating-point data registers,
HHL indicates bits D55 to D48 of the double-precision floating-point data registers, and
HHH indicates bits D63 to D56 of the double-precision floating-point data registers.

Double-precision floating-point data register: DRm	D63 to D56	D55 to D48	D47 to D40	D39 to D32
	HHH	HHL	HLH	HLL
	D31 to D24	**D23 to D16**	**D15 to D8**	**D7 to D0**
	LHH	LHL	LLH	LLL

Table 2.14 64-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 64-bit unit from address 0	Reading a 64-bit unit from address 4	Reading a 64-bit unit from address 8
Address 0	Transfer to LLL	—	—
Address 1	Transfer to LLH	—	—
Address 2	Transfer to LHL	—	—
Address 3	Transfer to LHH	—	—
Address 4	Transfer to HLL	Transfer to LLL	—
Address 5	Transfer to HLH	Transfer to LLH	—
Address 6	Transfer to HHL	Transfer to LHL	—
Address 7	Transfer to HHH	Transfer to LHH	—
Address 8	—	Transfer to HLL	Transfer to LLL
Address 9	—	Transfer to HLH	Transfer to LLH
Address A	—	Transfer to HHL	Transfer to LHL
Address B	—	Transfer to HHH	Transfer to LHH
Address C	—	—	Transfer to HLL
Address D	—	—	Transfer to HLH
Address E	—	—	Transfer to HHL
Address F	—	—	Transfer to HHH

Table 2.15 64-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 64-bit unit from address 0	Reading a 64-bit unit from address 4	Reading a 64-bit unit from address 8
Address 0	Transfer to HHH	—	—
Address 1	Transfer to HHL	—	—
Address 2	Transfer to HLH	—	—
Address 3	Transfer to HLL	—	—
Address 4	Transfer to LHH	Transfer to HHH	—
Address 5	Transfer to LHL	Transfer to HHL	—
Address 6	Transfer to LLH	Transfer to HLH	—
Address 7	Transfer to LLL	Transfer to HLL	—
Address 8	—	Transfer to LHH	Transfer to HHH
Address 9	—	Transfer to LHL	Transfer to HHL
Address A	—	Transfer to LLH	Transfer to HLH
Address B	—	Transfer to LLL	Transfer to HLL
Address C	—	—	Transfer to LHH
Address D	—	—	Transfer to LHL
Address E	—	—	Transfer to LLH
Address F	—	—	Transfer to LLL

Table 2.16 64-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 64-bit unit to address 0	Writing a 64-bit unit to address 4	Writing a 64-bit unit to address 8
Address 0	Transfer to LLL	—	—
Address 1	Transfer to LLH	—	—
Address 2	Transfer to LHL	—	—
Address 3	Transfer to LHH	—	—
Address 4	Transfer to HLL	Transfer to LLL	—
Address 5	Transfer to HLH	Transfer to LLH	—
Address 6	Transfer to HHL	Transfer to LHL	—
Address 7	Transfer to HHH	Transfer to LHH	—
Address 8	—	Transfer to HLL	Transfer to LLL
Address 9	—	Transfer to HLH	Transfer to LLH
Address A	—	Transfer to HHL	Transfer to LHL
Address B	—	Transfer to HHH	Transfer to LHH
Address C	—	—	Transfer to HLL
Address D	—	—	Transfer to HLH
Address E	—	—	Transfer to HHL
Address F	—	—	Transfer to HHH

Table 2.17 64-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 64-bit unit to address 0	Writing a 64-bit unit to address 4	Writing a 64-bit unit to address 8
Address 0	Transfer to HHH	—	—
Address 1	Transfer to HHL	—	—
Address 2	Transfer to HLH	—	—
Address 3	Transfer to HLL	—	—
Address 4	Transfer to LHH	Transfer to HHH	—
Address 5	Transfer to LHL	Transfer to HHL	—
Address 6	Transfer to LLH	Transfer to HLH	—
Address 7	Transfer to LLL	Transfer to HLL	—
Address 8	—	Transfer to LHH	Transfer to HHH
Address 9	—	Transfer to LHL	Transfer to HHL
Address A	—	Transfer to LLH	Transfer to HLH
Address B	—	Transfer to LLL	Transfer to HLL
Address C	—	—	Transfer to LHH
Address D	—	—	Transfer to LHL
Address E	—	—	Transfer to LLH
Address F	—	—	Transfer to LLL

2.9 Acceleration Functions

This MCU has two caches, one each for instructions and data, a branch instruction processing accelerator, and local SRAM (LRAM) for acceleration of processing. All are capable of operating at the same frequency as that of the CPU core.

2.9.1 Features

- Instruction cache and data cache
  Instruction cache: 32 Kbytes (4-way set associative cache)
  Data cache: 32 Kbytes (4-way set associative cache)
- Branch instruction processing accelerator
  Accelerator for MRAM and LRAM: 272 bytes (Fully associative cache)
- Local SRAM (LRAM)
  128 Kbytes × 2 units

Table 2.18 lists the features of the caches, Table 2.19 lists the features of the branch instruction processing accelerator, and Table 2.20 lists the features of the LRAM.

Table 2.18 Features of the Caches

Item	Instruction Cache	Data Cache
Capacity	32 Kbytes	32 Kbytes
Mapping	4-way set associative cache	4-way set associative cache
Line size	256 bits (32 bytes)	256 bits (32 bytes)
Number of entries	256 entries/way	256 entries/way
Write mode	—	Write-through or write-back non-write-allocate/write-allocate
Update Method	LRU	LRU
Error detection method	ECC Data: 1-bit error correction, 2-bit error detection Tag: 1-bit error detection, 2-bit error detection	ECC Data: 1-bit error correction, 2-bit error detection Tag: 1-bit error detection, 2-bit error detection

Table 2.19 Features of the Branch Instruction Processing Accelerator

Item	Description
Capacity	272 bytes (256 bytes for branch Instructions and 16 bytes for RTS instruction)
Mapping	Fully associative cache
Line size	128 bits (One line consists of two 64-bit data fields allocated on 64-bit boundaries.)
Number of entries	16 entries for branch Instructions and 1 entry for RTS instruction
Update Method	LRU
Caching areas	MRAM: 7F00 0000h to 7FFF FFFFh, FF00 0000h to FFFF FFFFh LRAM: 0000 0000h to 00FF FFFFh, 8000 0000h to 80FF FFFFh

Table 2.20 Features of the Local SRAM (LRAM)

Item	LRAM0/1
Capacity	128 Kbytes
Error detection method	ECC 1-bit error correction, 2-bit error detection

2.9.2 Block Diagrams

Figure 2.19 shows the relationship between the caches, branch instruction processing accelerator, and LRAM units.

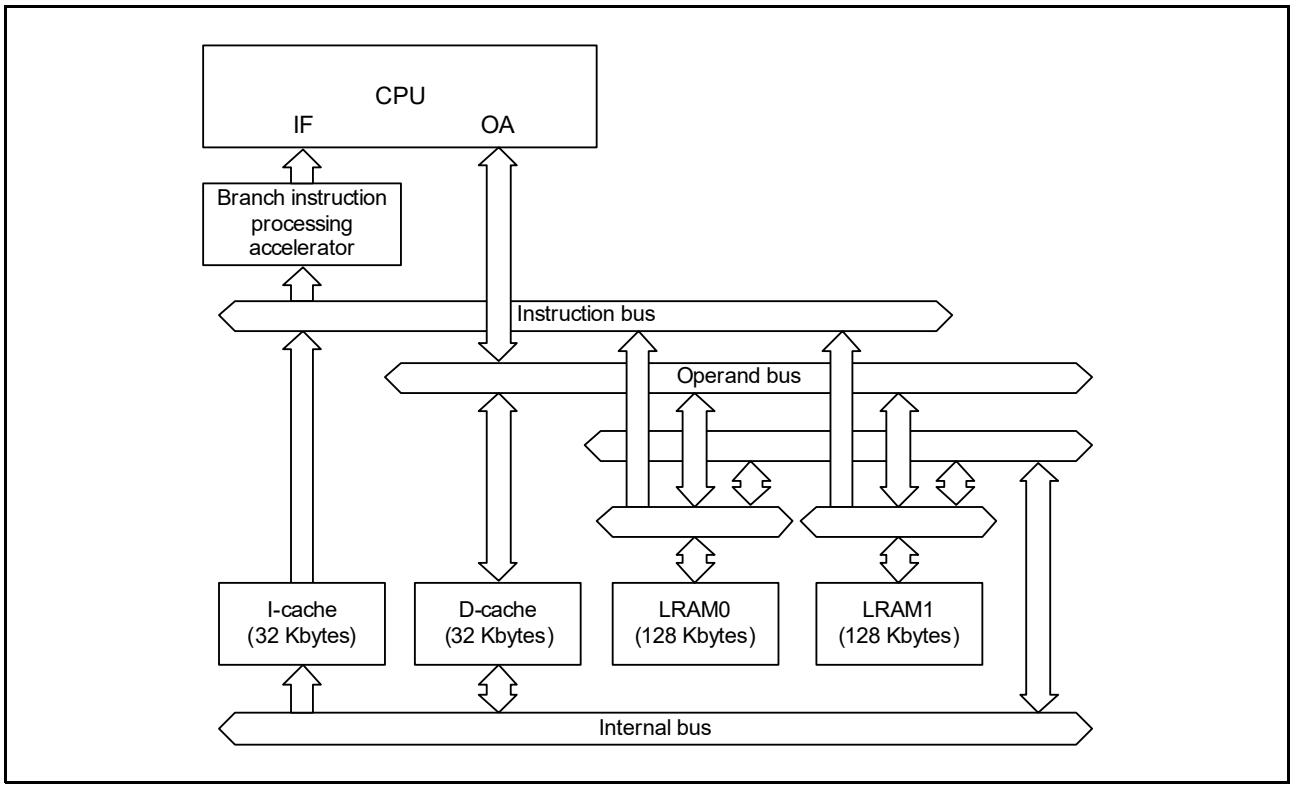


Figure 2.19 Relationship between the Caches, Branch Instruction Accelerator, and LRAM Units

Figure 2.20 shows the structure of the caches.

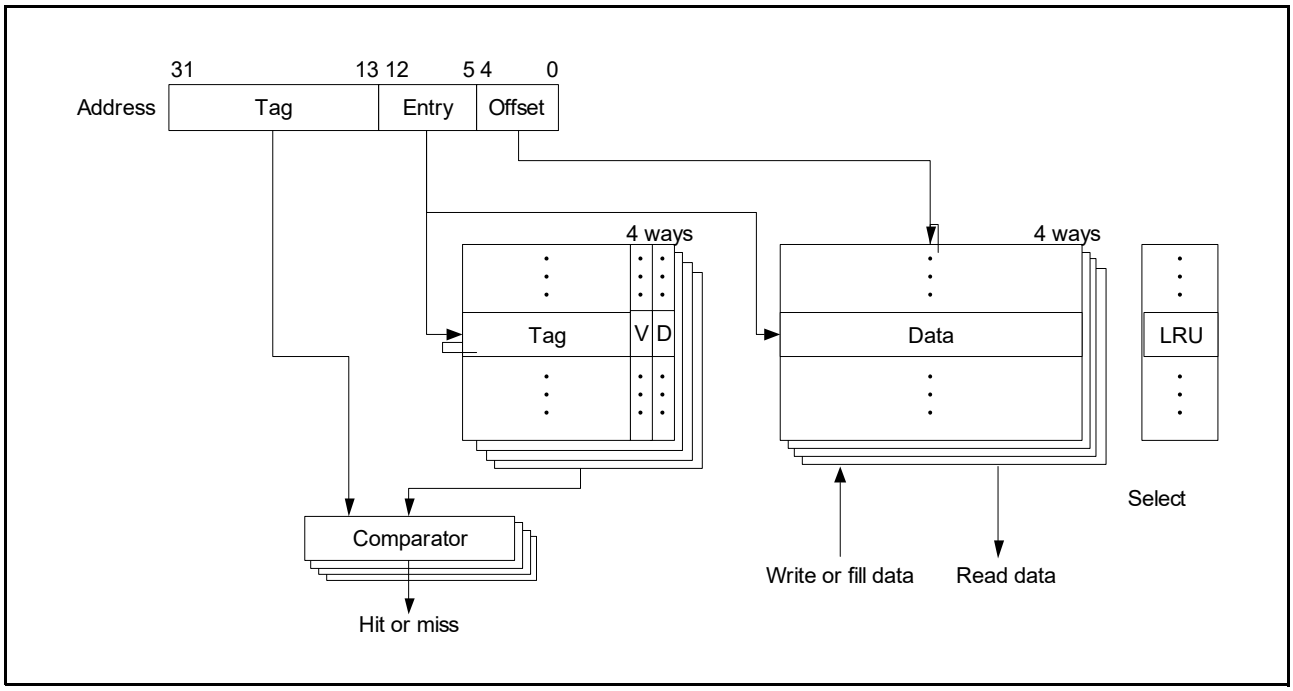


Figure 2.20 Internal Structure of the Caches

- Tag
  Each tag field holds cached address bits 31 to 13. It also holds the ECC code for the 32-bit value consisting of address bits 31 to 13, V bit, and 0-extension bits. The tag fields are cleared by a cache flush.
- V
  Each cache line has a valid (V) bit. This indicates that the valid data has been stored in the cache line. The tag fields are cleared by a cache flush.
- D
  Each cache line has a dirty (D) bit. This indicates that the valid data value stored in the cache line has changed from the initial value and the value needs to be written back through the internal bus. Note that the instruction cache does not have D bits.
- Data
  Each data field consists of 256 bits per cache line. It also holds the ECC codes for every 32-bit unit.
- LRU
  The LRU (least recently used) field indicates the next cache line to be replaced.
- Comparator
  The comparator judges whether the target data value for access is stored in any cache line. When the V bit is set and the target address matches the address stored in the tag field, the access is judged to be a cache hit. Otherwise, the access is judged to be a cache miss. Note that if an ECC error is found in the tag while the access has been judged as a hit, the operation is handled as a miss.

Figure 2.21 shows the structure of the branch instruction processing accelerator.

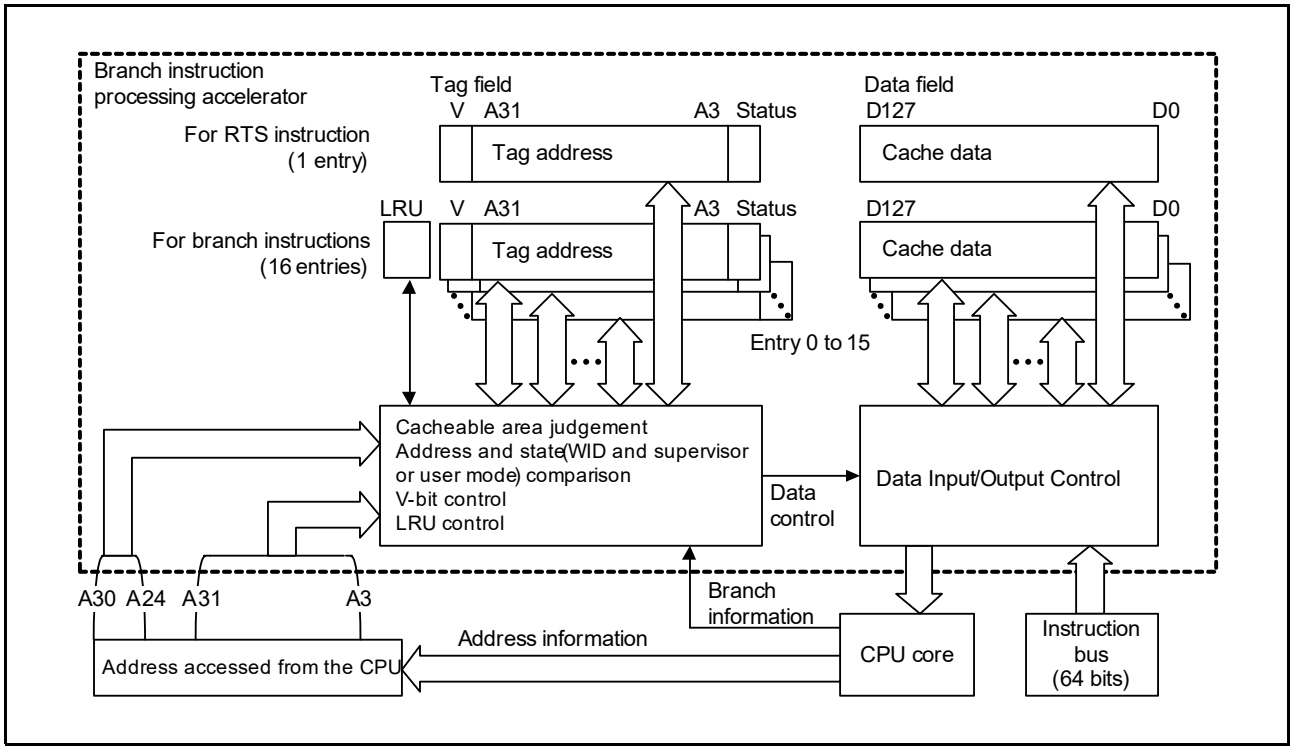


Figure 2.21 Internal Structure of the Branch Instruction Processing Accelerator

2.9.3 Cache Control Registers

The cache has the following fourteen control registers.

- Cache security attribute register (CACHESAR)
- Instruction cache control register (IFCACTL)
- Instruction cache flush control register (IFCAFCT)
- Instruction cache error detection status register (IFCAEDST)
- Instruction cache test access address register (IFCATAA)*1
- Instruction cache test access data register (IFCATAD)*1
- Data cache control register (OACACTL)
- Data cache flush control register (OACAFCT)
- Data cache write attribute register (OACAWTA)
- Data cache error detection status register (OACAEDST)
- Data cache test access address register (OACATAA)*1
- Data cache test access data register (OACATAD)*1
- Operation settings after cache error detection register (CAOAD)
- Cache protection register (CAPRCR)

Note 1. The register can be used in testing of the ECC decoder for function safety.

2.9.3.1 Cache Security Attribute Register (CACHESAR)

Address(es): CPSCU.CACHESAR 8701 8500h, CPSCU_NS.CACHESAR 0701 8500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CACHEESA	—	CACHESA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACHESA	Security Attribute Setting for Control Registers	0: Secure 1: Non-secure	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	CACHEESA	Security Attribute Setting for Error Registers	0: Secure 1: Non-secure	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is write-protected by the PRCR_S.PRC4 bit. It is only writable from secure world.

CACHESA Bit (Security Attribute Setting for Control Registers)

This bit is used to specify the security attribute for the control registers. The following lists the applicable registers.
IFCACTL, IFCAEDST, OACACTL, OACAWTA, OACAEDST

CACHEESA Bit (Security Attribute Setting for Error Registers)

This bit is used to specify the security attribute for the error registers. The following lists the applicable registers.
CAOAD, CAPRCR

To make secure program or secure data cacheable, set the CACHESA and CACHEESA bits to 0 (secure).

2.9.3.2 Instruction Cache Control Register (IFCACTL)

Address(es): CACHE.IFCACTL 8101 0000h, CACHE_NS.IFCACTL 0101 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FL	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN	Instruction Cache Enable	0: The instruction cache is disabled. 1: The instruction cache is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FL	Instruction Cache Flush	0: No effect 1: The instruction cache is flushed: that is, all lines are invalidated. This bit has the same function as that of the IFCAFCT.FL bit.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

EN Bit (Instruction Cache Enable)

This bit is used to enable or disable the instruction cache. When the instruction cache is enabled, whether to cache the fetched instruction depends on the cache attribute of the CPU access. Refer to the related descriptions in the section on the MPU.

FL Bit (Instruction Cache Flush)

Refer to the description of the IFCAFCT.FL bit for details.

2.9.3.3 Instruction Cache Flush Control Register (IFCAFCT)

Address(es): CACHE.IFCAFCT 8101 0004h, CACHE_NS.IFCAFCT 0101 0004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FL	Instruction Cache Flush	0: No effect 1: The instruction cache is flushed: that is, all lines are invalidated.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FL Bit (Instruction Cache Flush)

Writing 1 to this bit starts flushing of the instruction cache (invalidating the contents of the cache). This bit is automatically cleared upon completion of the flush. Do not write to this bit or the IFCAFCT register before automatic clearing of this bit.

2.9.3.4 Instruction Cache Error Detection Status Register (IFCAEDST)

Address(es): CACHE.IFCAEDST 8101 0010h, CACHE_NS.IFCAEDST 0101 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	EST2	—	ESTC	ESD2	ESD1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESD1	Instruction Cache 1-bit ECC Error Flag*1	This flag indicates that a 1-bit ECC error has been detected during reading from a data field in the instruction cache. 0: A 1-bit ECC error has not been detected. 1: A 1-bit ECC error has been detected and the error has been corrected.	R/W
b1	ESD2	Instruction Cache 2-bit ECC Error Flag*1	This flag indicates that a 2-bit ECC error has been detected during reading from a data field in the instruction cache. 0: A 2-bit ECC error has not been detected. 1: A 2-bit ECC error has been detected.	R/W
b2	ESTC	Instruction Cache Tag Clean Line Invalidate Flag*1	This flag indicates that a clean line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the instruction cache. 0: A clean line has not been invalidated. 1: A clean line has been invalidated.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	EST2	Instruction Cache Tag 2-bit ECC Error Flag*1	This flag indicates that a 2-bit ECC error has been detected during reading from a tag field in the instruction cache. 0: A 2-bit ECC error has not been detected. 1: A 2-bit ECC error has been detected.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The write value should be 0. Writing 0 will clear the flag.

ESD1 Flag (Instruction Cache 1-bit ECC Error Flag)

This flag indicates whether a 1-bit ECC error has been detected during reading from a data field in the instruction cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

ESD2 Flag (Instruction Cache 2-bit ECC Error Flag)

This flag indicates whether a 2-bit ECC error has been detected during reading from a data field in the instruction cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

ESTC Flag (Instruction Cache Tag Clean Line Invalidate Flag)

This flag indicates whether a line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the instruction cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and

the error is ignored.

EST2 Flag (Instruction Cache Tag 2-bit ECC Error Flag)

This flag indicates whether a 2-bit ECC error has been detected during reading from a tag field in the instruction cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

2.9.3.5 Instruction Cache Test Access Address Register (IFCATAA)

Address(es): CACHE.IFCATAA 8101 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WAY[1:0]		—	—	—	—	—	—	RW	—	—	—	—	TARGET[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ENTRY[7:0]								OFFSET[2:0]			—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b2	OFFSET[2:0]	Address Offset	Specify the value of bits 4 to 2 of the address for test access to the instruction cache; these bits indicate the value of the three high-order bits of the cache offset for use in testing. This setting is valid for the reading and writing of cache data or ECC codes.	R/W
b12 to b5	ENTRY[7:0]	Address Entry	Specify the value of bits 12 to 5 of the address for test access to the instruction cache; these bits indicate the cache entry for use in testing.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	TARGET[2:0]	Target of Test Access	Specify the target of test access to the instruction cache. The type of access can be selected by the RW bit. b18 b16 0 0 0: Reading or writing of cache data 0 0 1: Reading or writing of ECC codes 0 1 0: Reading or writing of the tag field and V bit 0 1 1: Reading or writing of the LRU 1 0 0: Reading or writing of ECC code for the address in the tag field Settings other than above are prohibited.	R/W
b22 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	RW	Read/Write Select	Select the type of access (read or write) to the target of testing. 0: Read 1: Write	R/W
b29 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31, b30	WAY[1:0]	Address Way	Specify the value of bits 31 and 30 of the address for test access to the instruction cache; these bits indicate the cache way for use in testing. This setting is valid for the reading and writing of cache data, ECC codes, or tag values. The setting is ignored in reading and writing of the LRU value.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register should be accessed in 32-bit units.

When the type of test access is “write”, write to the IFCATAD register before writing to the IFCATAA register. When the type of test access is “read”, read the IFCATAD register after writing to the IFCATAA register. Use the procedure in Table 2.21 for reading and the procedure in Table 2.22 for writing.

Table 2.21 Procedure of Reading

1	Write 0 (reading) to the IFCATAA.RW bit and set the other bits to values that suite the test.
2	Read the IFCATAA register and check the value that is read.
3	Read the IFCATAD register.

Table 2.22 Procedure of Writing

1	Write a value to the IFCATAD register.
2	Write 1 (writing) to the IFCATAA.RW bit and set the other bits to values that suit the test.

OFFSET[2:0] Bits (Address Offset)
ENTRY[7:0] Bits (Address Entry)
WAY[1:0] Bits (Address Way)

Specify the address for test access to the instruction cache.

Table 2.23 Relationship between the TARGET[2:0], WAY[1:0], ENTRY[7:0], and OFFSET[2:0] Bits

TARGET[2:0]	WAY[1:0]	ENTRY[7:0]	OFFSET[2:0]
000b (Cache data)	Valid	Valid	Valid
001b (ECC code)	Valid	Valid	Valid
010b (TAG and V bits)	Valid	Valid	Invalid
011b (LRU)	Invalid	Valid	Invalid
100b (TAG ECC code)	Valid	Valid	Invalid

TARGET[2:0] Bits (Target of Test Access)

Specify the target of test access to the instruction cache.

RW Bit (Read/Write Select)

Select the type of access (read or write) to the target of testing.

2.9.3.6 Instruction Cache Test Access Data Register (IFCATAD_DATA, IFCATAD_ECC, IFCATAD_TAG, IFCATAD_LRU, IFCATAD_TAGECC)

The valid bits and their functions depend on the setting of the IFCATAA.TARGET[2:0] bits.

- IFCATAA.TARGET[2:0] bits = 000b (Cache data)

Address(es): CACHE.IFCATAD_DATA 8101 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DATA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DATA[31:0]	Cache Data	These bits hold the cache data that have been read or are to be written in test access to the instruction cache.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

DATA[31:0] Bits (Cache Data)

These bits hold the cache data that have been read or are to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- IFCATAA.TARGET[2:0] bits = 001b (ECC code)

Address(es): CACHE.IFCATAD_ECC 8101 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	ECC[6:0]	ECC Code	These bits hold the ECC code that has been read or is to be written in test access to the instruction cache.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

ECC[6:0] Bits (ECC Code)

These bits hold the ECC code that has been read or is to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by

test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- IFCATAA.TARGET[2:0] bits = 010b (TAG and V bits)

Address(es): CACHE.IFCATAD_TAG 8101 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG[18:3]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG[2:0]			—	—	—	—	—	—	—	—	—	—	—	V	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	V	Valid	This bit holds the value of the valid bit that has been read or is to be written in test access to the instruction cache.	R/W
b12 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b13	TAG[18:0]	TAG Data	These bits hold the value of the TAG[31:13] bits that has been read or is to be written in test access to the instruction cache.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

V Bit (Valid)

This bit holds the value of the valid bit that has been read or is to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

TAG[18:0] Bits (TAG Data)

These bits hold the value of the TAG[31:13] bits that has been read or is to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- IFCATAA.TARGET[2:0] bits = 011b (LRU)

Address(es): CACHE.IFCATAD_LRU 8101 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	LRU[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	LRU[4:0]	LRU	These bits hold the value of the LRU that has been read or is to be written in test access to the instruction cache.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

LRU[4:0] Bits (LRU)

These bits hold the value of the LRU that has been read or is to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- IFCATAA.TARGET[2:0] bits = 100b (TAG ECC code)

Address(es): CACHE.IFCATAD_TAGECC 8101 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TAG ECC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	TAG ECC[6:0]	TAG ECC Code	These bits hold the ECC code for the tag that has been read or is to be written in test access to the instruction cache.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

TAG ECC[6:0] Bits (TAG ECC Code)

These bits hold the ECC code for the tag that has been read or is to be written in test access to the instruction cache. Before access to this register, disable the instruction cache by setting the IFCACTL.EN bit to 0. If updating of the IFCATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

2.9.3.7 Data Cache Control Register (OACACTL)

Address(es): CACHE.OACACTL 8101 0040h, CACHE_NS.OACACTL 0101 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	WB	FL	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN	Data Cache Enable	0: The data cache is disabled. 1: The data cache is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FL	Data Cache Flush	0: No effect 1: The data cache is flushed: that is, all lines are invalidated. This bit has the same function as that of the OACAFCT.FL bit.	R/W
b9	WB	Data Cache Write-back	0: No effect 1: The contents of the data cache are written back to memory. This bit has the same function as that of the OACAFCT.WB bit.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

EN Bit (Data Cache Enable)

This bit is used to enable or disable the data cache. When the data cache is enabled, whether to cache data depends on the cache attribute of the CPU access. Refer to the related descriptions in the section on the MPU.

FL Bit (Data Cache Flush)

Refer to the description of the OACAFCT.FL bit for details.

WB Bit (Data Cache Write-back)

Refer to the description of the OACAFCT.WB bit for details.

2.9.3.8 Data Cache Flush Control Register (OACAFCT)

Address(es): CACHE.OACAFCT 8101 0044h, CACHE_NS.OACAFCT 0101 0044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WB	FL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FL	Data Cache Flush	0: No effect 1: The data cache is flushed: that is, all lines are invalidated.	R/W
b1	WB	Data Cache Write-back	0: No effect 1: The contents of the data cache are written back to memory.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When the OACAFCT.FL bit is set to 1 while the OACAWTA.WT bit is 0, the cache is flushed and data are written back to memory.

FL Bit (Data Cache Flush)

Writing 1 to this bit starts flushing of the data cache (invalidating the contents of the cache). This bit is automatically cleared upon completion of the flush. Do not write to this bit before automatic clearing of this bit.

WB Bit (Data Cache Write-back)

Writing 1 to this bit writes the contents of the data cache back to memory. This bit is automatically cleared upon completion of writing back. Do not write to this bit before automatic clearing of this bit.

2.9.3.9 Data Cache Write Attribute Register (OACAWTA)

Address(es): CACHE.OACAWTA 8101 004Ch, CACHE_NS.OACAWTA 0101 004Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WA	WT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WT	Data Cache Write-through Setting	0: Whether the data cache is set to the write-through mode or write-back mode depends on the cache attribute. 1: The data cache is set to the write-through mode.	R/W
b1	WA	Data Cache Write Allocation Setting	0: The write allocation mode is disabled. 1: Whether the write allocation mode is enabled or disabled depends on the cache attribute.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

WT Bit (Data Cache Write-through Setting)

This bit is used to specify the method for writing data from the data cache to memory. Before modifying this setting, set the OACACTL.EN bit to 0 to disable the data cache. For details of the cache attributes, refer to the section on the MPU.

WA Bit (Data Cache Write Allocation Setting)

This bit is used to specify the write allocation mode for the data cache. Before modifying this setting, set the OACACTL.EN bit to 0 to disable the data cache. For details of the cache attributes, refer to the section on the MPU.

2.9.3.10 Data Cache Error Detection Status Register (OACAEDST)

Address(es): CACHE.OACAEDST 8101 0050h, CACHE_NS.OACAEDST 0101 0050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	EST2	ESTD	ESTC	ESD2	ESD1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESD1	Data Cache 1-Bit ECC Error Flag *1	This flag indicates that a 1-bit ECC error has been detected during reading from a data field in the data cache. 0: A 1-bit ECC error has not been detected. 1: A 1-bit ECC error has been detected and the error correction has been executed.	R/W
b1	ESD2	Data Cache 2-Bit ECC Error Flag *1	This flag indicates that a 2-bit ECC error has been detected during reading from a data field in the data cache. 0: A 2-bit ECC error has not been detected. 1: A 2-bit ECC error has been detected.	R/W
b2	ESTC	Data Cache Tag Clean Line Invalidate Flag*1	This flag indicates that a clean line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the data cache. 0: A 1-bit ECC error has not been detected. 1: A 1-bit ECC error has been detected and a clean line has been invalidated.	R/W
b3	ESTD	Data Cache Tag Dirty Line Invalidate Flag*1	This flag indicates that a dirty line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the data cache. 0: A 1-bit ECC error has not been detected. 1: A 1-bit ECC error has been detected and a dirty line has been invalidated.	R/W
b4	EST2	Data Cache Tag 2-Bit ECC Error Flag*1	This flag indicates that a 2-bit ECC error has been detected during reading from a tag field in the data cache. 0: A 2-bit ECC error has not been detected. 1: A 2-bit ECC error has been detected.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The write value should be 0. Writing 0 will clear the flag.

ESD1 Flag (Data Cache 1-Bit ECC Error Flag)

This flag indicates whether a 1-bit ECC error has been detected during reading from a data field in the data cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

ESD2 Flag (Data Cache 2-Bit ECC Error Flag)

This flag indicates whether a 2-bit ECC error has been detected during reading from a data field in the data cache. The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

ESTC Flag (Data Cache Tag Clean Line Invalidate Flag)

This flag indicates whether a clean line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the data cache.
The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

ESTD Flag (Data Cache Tag Dirty Line Invalidate Flag)

This flag indicates whether a dirty line has been invalidated due to the detection of a 1-bit ECC error during reading from a tag field in the data cache.
The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored. If the ESTD bit is 1, the dirty line is invalidated without writing its contents back to memory.

EST2 Flag (Data Cache Tag 2-Bit ECC Error Flag)

This flag indicates whether a 2-bit ECC error has been detected during reading from a tag field in the data cache.
The flag can be cleared by writing 0 to it. This flag is also cleared by a reset other than a bus error reset or a memory error reset. If an error occurs at the same time as clearing of the flag, the flag is cleared and the error is ignored.

2.9.3.11 Data Cache Test Access Address Register (OACATAA)

Address(es): CACHE.OACATAA 8101 0054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WAY[1:0]		—	—	—	—	—	—	RW	—	—	—	—	TARGET[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ENTRY[7:0]								OFFSET[2:0]			—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b2	OFFSET[2:0]	Address Offset	Specify the value of bits 4 to 2 of the address for test access to the data cache; these bits indicate the value of the three high-order bits of the cache offset for use in testing. This setting is valid for the reading and writing of cache data or ECC codes.	R/W
b12 to b5	ENTRY[7:0]	Address Entry	Specify the value of bits 12 to 5 of the address for test access to the data cache; these bits indicate the cache entry for use in testing.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	TARGET[2:0]	Target of Test Access	Specify the target of test access to the data cache. The type of access can be selected by the RW bit. b18 b16 0 0 0: Reading or writing of cache data 0 0 1: Reading or writing of ECC codes in the DATA field 0 1 0: Reading or writing of the tag field, D and V bits 0 1 1: Reading or writing of the LRU 1 0 0: Reading or writing of ECC code for the address in the tag field Settings other than above are prohibited.	R/W
b22 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	RW	Read/Write Select	Select the type of access (read or write) to the target of testing. 0: Read 1: Write	R/W
b29 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31, b30	WAY[1:0]	Address Way	Specify the value of bits 31 and 30 of the address for test access to the data cache; these bits indicate the cache way for use in testing. This setting is valid for the reading and writing of cache data, ECC codes, or tag values. The setting is ignored in reading and writing of the LRU value.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register should be accessed in 32-bit units.

When the type of test access is “write”, write to the OACATAD register before writing to the OACATAA register. When the type of test access is “read”, read the OACATAD register after writing to the OACATAA register. Use the procedure in Table 2.24 for reading and the procedure in Table 2.25 for writing.

Table 2.24 Procedure of Reading

1	Write 0 (reading) to the OACATAA.RW bit and set the other bits to values that suit the test.
2	Read the OACATAA register and check the value that is read.
3	Read the OACATAD register.

Table 2.25 Procedure of Writing

1	Write a value to the OACATAD register.
2	Write 1 (writing) to the OACATAA.RW bit and set the other bits to values that suit the test.

OFFSET[2:0] Bits (Address Offset)
ENTRY[7:0] Bits (Address Entry)
WAY[1:0] Bits (Address Way)

Specify the address for test access to the data cache.

Table 2.26 Relationship between the TARGET[2:0], WAY[1:0], ENTRY[7:0], and OFFSET[2:0] Bits

TARGET[2:0]	WAY[1:0]	ENTRY[7:0]	OFFSET[2:0]
000b (Cache data)	Valid	Valid	Valid
001b (ECC code)	Valid	Valid	Valid
010b (TAG, V, and D bits)	Valid	Valid	Invalid
011b (LRU)	Invalid	Valid	Invalid
100b (TAG ECC code)	Valid	Valid	Invalid

TARGET[2:0] Bits (Target of Test Access)

Specify the target of test access to the data cache.

RW Bit (Read/Write Select)

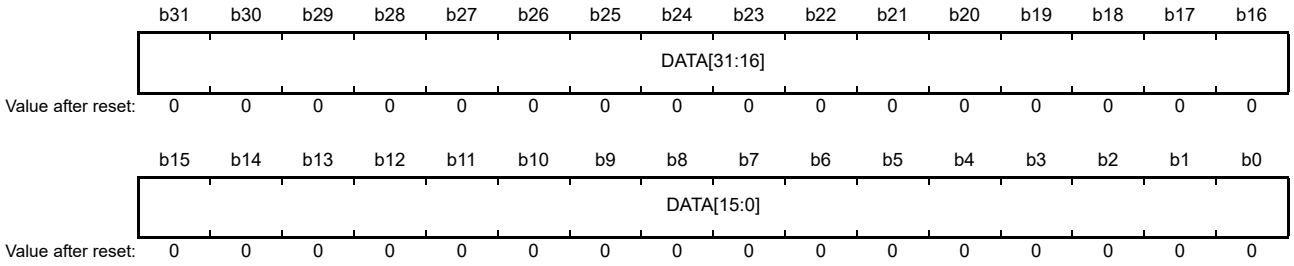
Select the type of access (read or write) to the target of testing.

2.9.3.12 Data Cache Test Access Data Register (OACATAD_DATA, OACATAD_ECC, OACATAD_TAG, OACATAD_LRU, OACATAD_TAGECC)

The valid bits and their functions depend on the setting of the OACATAA.TARGET[2:0] bits.

- OACATAA.TARGET[2:0] bits = 000b (Cache data)

Address(es): CACHE.OACATAD_DATA 8101 0058h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DATA[31:0]	Cache Data	These bits hold the cache data that have been read or are to be written in test access to the data cache.	R/W

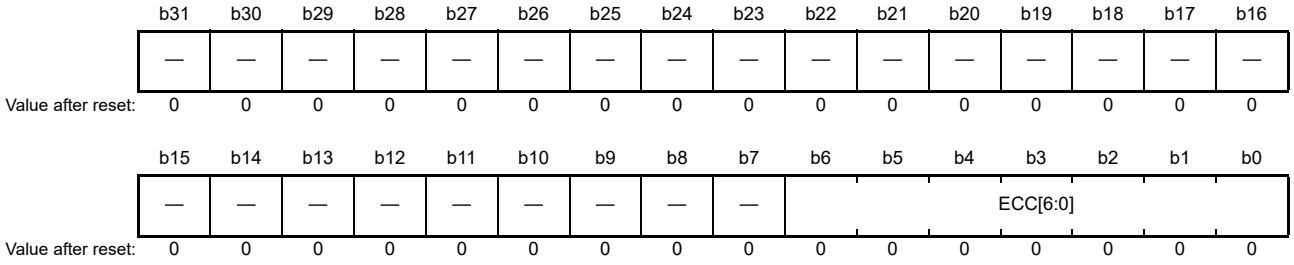
Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

DATA[31:0] Bits (Cache Data)

These bits hold the cache data that have been read or are to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- OACATAA.TARGET[2:0] bits = 001b (ECC code)

Address(es): CACHE.OACATAD_ECC 8101 0058h



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	ECC[6:0]	ECC Code	These bits hold the ECC code that has been read or is to be written in test access to the data cache.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

ECC[6:0] Bits (ECC Code)

These bits hold the ECC code that has been read or is to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read

access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- OACATAA.TARGET[2:0] bits = 010b (TAG, V, D bit)

Address(es): CACHE.OACATAD_TAG 8101 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG[18:3]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG[2:0]			—	—	—	—	—	—	—	—	—	—	—	V	D
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	D	Dirty	This bit holds the value of the dirty bit that has been read or is to be written in test access to the data cache.	R/W
b1	V	Valid	This bit holds the value of the valid bit that has been read or is to be written in test access to the data cache.	R/W
b12 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b13	TAG[18:0]	TAG Data	These bits hold the value of the TAG[31:13] bits that has been read or is to be written in test access to the data cache.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

D Bit (Dirty)

This bit holds the value of the dirty bit that has been read or is to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

V Bit (Valid)

This bit holds the value of the valid bit that has been read or is to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

TAG[18:0] Bits (TAG Data)

These bits hold the value of the TAG[31:13] bits that has been read or is to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- OACATAA.TARGET[2:0] bits = 011b (LRU)

Address(es): CACHE.OACATAD_LRU 8101 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	LRU[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	LRU[4:0]	LRU	These bits hold the value of the LRU that has been read or is to be written in test access to the data cache.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

LRU[4:0] Bits (LRU)

These bits hold the value of the LRU that has been read or is to be written in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

- OACATAA.TARGET[2:0] bits = 100b (TAG ECC code)

Address(es): CACHE.OACATAD_TAGECC 8101 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TAG ECC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	TAG ECC[6:0]	TAG ECC Code	These bits hold the ECC code for the tag that has been read or is to be written in test access to the data cache.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

TAG ECC[6:0] Bits (TAG ECC Code)

These bits hold the ECC code for the address in the tag field in test access to the data cache. Before access to this register, disable the data cache by setting the OACACTL.EN bit to 0. If updating of the OACATAD register by test read access and writing to the register occur simultaneously, updating of the register is ignored and writing to the register proceeds.

2.9.3.13 Operation Settings After Cache Error Detection Register (CAOAD)

Address(es): CACHE.CAOAD 8101 0200h, CACHE_NS.CAOAD 0101 0200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	E1STSEN	ECCMOD1	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OAD	Setting of Operation after Cache Error Detection	0: Interrupt 1: Reset	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	ECCMOD1	ECC Error Correction Enable	0: ECC error correction is disabled. 1: ECC error correction is enabled.	R/W
b4	E1STSEN	Non-fatal ECC Error Information Updating Enable	0: The flags to indicate non-fatal ECC errors are not updated. 1: The flags to indicate non-fatal ECC errors are updated.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

OAD Bit (Setting of Operation after Cache Error Detection)

This bit is used to specify the operation after the detection of an ECC error.

ECCMOD1 Bit (ECC Error Correction Enable)

This bit is used to enable the detection and correction of ECC errors. Only modify this setting while cache operations are stopped.

E1STSEN Bit (Non-fatal ECC Error Information Updating Enable)

This bit is used to enable or disable updating of the status flags for non-fatal ECC errors (IFCAEDST.ESD1, IFCAEDST.ESTC, OACAEDST.ESD1, and OACAEDST.ESTC flags). While this bit is 0, the error status flags are not updated and the operation specified to follow error detection does not proceed regardless of the setting of the OAD bit in this register.

2.9.3.14 Cache Protection Register (CAPRCR)

Address(es): CACHE.CAPRCR 8101 0204h, CACHE_NS.CAPRCR 0101 0204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	KW[6:0]							PRCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRCR	CAOAD Register Write Control	0: Writing to the CAOAD register is disabled. 1: Writing to the CAOAD register is enabled.	R/W
b7 to b1	KW[6:0]	Write Keyword	Modification of the CAOAD register setting is enabled or disabled. Before modifying the CAOAD register setting, write 1111000b to these bits.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

While the KW[6:0] bits are 1111000b, writing 1 to the PRCR bit is enabled. The KW[6:0] bits are always read as 0000000b. The CAOAD register is the only target of write-protection by this means. Once the PRCR bit is set to 1, writing to the CAOAD register is enabled until the PRCR bit is set to 0.

2.9.4 Branch Instruction Processing Accelerator Registers

The branch instruction processing accelerator has the following control registers.

- Branch instruction processing accelerator security attribute register (IFUSAR)
- Branch instruction processing accelerator enable register (CPUBAEN)
- Branch instruction processing accelerator invalidation setting register (CPUBAINV)

2.9.4.1 Branch Instruction Processing Accelerator Security Attribute Register (IFUSAR)

Address(es): CPSCU.IFUSAR 8701 8144h, CPSCU_NS.IFUSAR 0701 8144h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFUSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IFUSA	Security Attribute	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to configure the security attribute of the CPUBAEN and CPUBAINV registers. This register is write-protected by the PRCR_S.PRC4 bit.

2.9.4.2 Branch Instruction Processing Accelerator Enable Register (CPUBAEN)

Address(es): CPU_CTRL.CPUBAEN 8100 0100h, CPU_CTRL_NS.CPUBAEN 0100 0100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BAEN	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	BAEN	Enable	0: The branch instruction processing accelerator is disabled. 1: The branch instruction processing accelerator is enabled (initial value)	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register is used to enable both the entries for branch instructions and the entry for the RTS instruction in the branch instruction accelerator. Writing 0 to the BAEN bit stops the branch instruction processing accelerator.

2.9.4.3 Branch Instruction Processing Accelerator Invalidation Setting Register (CPUBAINV)

Address(es): CPU_CTRL.CPUBAINV 8100 0104h, CPU_CTRL_NS.CPUBAINV 0100 0104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BAIV	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	BAIV	Invalidation	0: Invalidation processing does not proceed. 1: The contents of the branch instruction processing accelerator are invalidated.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register is used to invalidate the contents of both the entries for branch instructions and the entry for the RTS instruction in the branch instruction processing accelerator. The entries are invalid immediately after a reset. Modifying the settings of the SAU or MPU also automatically invalidates the entries, so invalidation by using this register is not required. However, the entries are not automatically invalidated after overwriting an instruction in the MRAM or LRAM or modifying security settings other than those in the SAU, so use this register to invalidate the entries in such cases.

2.9.5 Local SRAM (LRAM) Control Registers

The LRAM has the following seven control registers.

- LRAM security attribute setting register (LRMSAR)
- LRAMn boundary address setting register (LRMSABARn)
- LRAM protection control register (LRMPRCR)*1
- LRAMn control register (LRMCRn)
- LRAM error flag register (LRMESR)
- LRAM error flag clear register (LRMESCLR)
- LRAMn error address register m (LRMEARnm)

Note 1. This register is actually implemented as two registers to support the Protect Zone function. Where this is required, the two registers are distinguished by postfixes appended to the register names, specifically _S for the secure world register and _NS for the non-secure world register.

2.9.5.1 LRAM Security Attribute Setting Register (LRMSAR)

Address(es): CPSCU.LRMSAR 8701 8504h, CPSCU_NS.LRMSAR 0701 8504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LRMSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LRMSA	Security Attribute	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is write-protected by the PRCR_S.PRC4 bit.

This register controls two types of register: the LRMCRn registers and LRMESCLR register. When secure programs or secure data are to be allocated to the LRAM, set the LRMSA bit to 0 to select secure attribute.

2.9.5.2 LRAMn Boundary Address Setting Register (LRMSABARn) (n = 0, 1)

Address(es): CPSCU.LRMSABAR0 8701 8508h, CPSCU.LRMSABAR1 8701 850Ch,
CPSCU_NS.LRMSABAR0 0701 8508h, CPSCU_NS.LRMSABAR1 0701 850Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	LRMSABA[5:3]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LRMSABA[2:0]			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b13	LRMSABA[5:0]	Security Attribute Boundary Address	First address of non-secure area	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is write-protected by the PRCR_S.PRC4 bit.

This register is used to specify the address of the boundary between secure area and non-secure area in each LRAM unit. Specify the address where non-secure area starts in the LRMSABA[5:0] bits.

When the boundary address LRMSABA[5:0] = 000000b, the entire LRAMn becomes a non-secure area. When LRMSABA[5:0] ≥ 010000b, the entire LRAMn becomes a secure area. Since the initial value of LRMSABA[5:0] is 111111b, the entire LRAMn after a reset becomes a secure area.

Table 2.27 Address Map and Accessibility in LRAM0

Master	Address as Seen from the Master	Secure Setting in the LRAM	Access
CPU (secure)	0000 0000h to 0000 0000h + LRMSABAR0 –1	Secure	Access violation
	0000 0000h + LRMSABAR0 to 0001 FFFFh	Non-secure	Available
	8000 0000h to 8000 0000h + LRMSABAR0 –1	Secure	Available
	8000 0000h + LRMSABAR0 to 8001 FFFFh	Non-secure	Access violation
CPU (non-secure)	0000 0000h to 0000 0000h + LRMSABAR0 –1	Secure	Access violation
	0000 0000h + LRMSABAR0 to 0001 FFFFh	Non-secure	Available
	8000 0000h to 8000 0000h + LRMSABAR0 –1	Secure	Access violation
	8000 0000h + LRMSABAR0 to 8001 FFFFh	Non-secure	Access violation
Bus master (secure)	0280 0000h to 0280 0000h + LRMSABAR0 –1	Secure	Access violation
	0280 0000h + LRMSABAR0 to 0281 FFFFh	Non-secure	Available
	8280 0000h to 8280 0000h + LRMSABAR0 –1	Secure	Available
	8280 0000h + LRMSABAR0 to 8281 FFFFh	Non-secure	Access violation
Bus master (non-secure)	0280 0000h to 0280 0000h + LRMSABAR0 –1	Secure	Access violation
	0280 0000h + LRMSABAR0 to 0281 FFFFh	Non-secure	Available
	8280 0000h to 8280 0000h + LRMSABAR0 –1	Secure	Access violation
	8280 0000h + LRMSABAR0 to 8281 FFFFh	Non-secure	Access violation

Table 2.28 Address Map and Accessibility in LRAM1

Master	Address as Seen from the Master	Secure Setting in the LRAM	Access
CPU (secure)	0002 0000h to 0002 0000h + LRMSABAR1 –1	Secure	Access violation
	0002 0000h + LRMSABAR1 to 0003 FFFFh	Non-secure	Available
	8002 0000h to 8002 0000h + LRMSABAR1 –1	Secure	Available
	8002 0000h + LRMSABAR1 to 8003 FFFFh	Non-secure	Access violation
CPU (non-secure)	0002 0000h to 0002 0000h + LRMSABAR1 –1	Secure	Access violation
	0002 0000h + LRMSABAR1 to 0003 FFFFh	Non-secure	Available
	8002 0000h to 8002 0000h + LRMSABAR1 –1	Secure	Access violation
	8002 0000h + LRMSABAR1 to 8003 FFFFh	Non-secure	Access violation
Bus master (secure)	0282 0000h to 0282 0000h + LRMSABAR1 –1	Secure	Access violation
	0282 0000h + LRMSABAR1 to 0283 FFFFh	Non-secure	Available
	8282 0000h to 8282 0000h + LRMSABAR1 –1	Secure	Available
	8282 0000h + LRMSABAR1 to 8283 FFFFh	Non-secure	Access violation
Bus master (non-secure)	0282 0000h to 0282 0000h + LRMSABAR1 –1	Secure	Access violation
	0282 0000h + LRMSABAR1 to 0283 FFFFh	Non-secure	Available
	8282 0000h to 8282 0000h + LRMSABAR1 –1	Secure	Access violation
	8282 0000h + LRMSABAR1 to 8283 FFFFh	Non-secure	Access violation

2.9.5.3 LRAM Protection Control Register for Secure Applications (LRMPRCR_S), LRAM Protection Control Register for Non-Secure Applications (LRMPRCR_NS)

Address(es): LRAM.LRMPRCR_S 8101 0800h, LRAM_NS.LRMPRCR_NS 0101 0804h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PR	LRMCRn Register Write Control	0: Writing to the LRMCRn register is disabled. 1: Writing to the LRMCRn register is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KW[7:0]	Write Keyword	Keyword bits for use in protection against writing to the PR bit	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be accessed in 32-bit units.

The two LRMPRCR registers are present, each for secure application developers (LRAMPRCR_S) and non-secure application developers (LRAMPRCR_NS). Setting the LRMSAR.LRMSA bit to 0 enables the LRAMPRCR_S register, and setting the LRMSAR.LRMSA bit to 1 enables the LRAMPRCR_NS register. The LRAMPRCR_S register is only accessible by secure transactions, and the LRAMPRCR_NS register is only accessible by non-secure transactions. Attempted access to these registers by using an unauthorized type of transaction causes a Protect Zone access error.
The LRAM control register (LRMCRn) is the only target of write-protection by this means. Once the PR bit is set to 1, writing to the LRMCRn register is enabled until the PR bit is set to 0. When writing to the PR bit, write 10100101b to the KW[7:0] bits at the same time. If another value is written to the KW[7:0] bits, the value of the PR bit will not be changed. The KW[7:0] bits are always read as 00000000b.

2.9.5.4 LRAMn Control Register (LRMCRn) (n = 0, 1)

Address(es): LRAM.LRMCR0 8101 0810h, LRAM.LRMCR1 8101 0814h,
LRAM_NS.LRMCR0 0101 0810h, LRAM_NS.LRMCR1 0101 0814h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TSTBYP	—	—	E1STSEN	ECCMOD[1:0]		—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OAD	Setting of Operation after Error Detection	0: Interrupt 1: Reset	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	ECCMOD[1:0]	ECC Operation Mode Select	b3 b2 0 0: ECC function is disabled. 0 1: Setting prohibited 1 0: ECC function is enabled, but without error checking. 1 1: ECC function is enabled, and with error checking.	R/W
b4	E1STSEN	1-Bit ECC Error Information Update Enable	0: Information on 1-bit ECC errors is not updated. 1: Information on 1-bit ECC errors is updated.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSTBYP	ECC Bypass Enable	0: ECC function is not bypassed. 1: ECC function is bypassed.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to this register while access to the LRAM is in progress.

OAD Bit (Setting of Operation after Error Detection)

This bit is used to specify the operation after the detection of an ECC error.

ECCMOD[1:0] Bits (ECC Operation Mode Select)

These bits are used to select the function for detecting and correcting ECC errors.

E1STSEN Bit (1-Bit ECC Error Information Update Enable)

This bit is used to enable or disable updating of the status flags for 1-bit ECC errors (the ERR01 and ERR11 bits in the LRMESR register). Even if a 1-bit ECC error has been corrected, the operation specified to follow error detection does not proceed regardless of the setting of the OAD bit in this register while the E1STSEN bit is 0.

TSTBYP Bit (ECC Bypass Enable)

While the ECCMOD[1:0] bits are 00b (disabling the ECC function), this bit is used to enable direct access to the ECC codes by bypassing the ECC function. The length of the ECC code for 32 bits of data is seven bits. When the TSTBYP bit is 1, bits 0 to 6 of the data that are read hold the ECC code and bits 7 to 31 hold an undefined value. Bits 0 to 6 of the written data are written to the ECC area at the specified address for writing to the LRAM. Bits 7 to 31 of the written data are ignored.

2.9.5.5 LRAM Error Flag Register (LRMESR)

Address(es): LRAM.LRMESR 8101 0840h, LRAM_NS.LRMESR 0101 0840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ERR12	ERR11	ERR02	ERR01
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERR01	LRAM0 1-bit ECC Error Flag	0: A 1-bit ECC error has not occurred. 1: A 1-bit ECC error has occurred.	R
b1	ERR02	LRAM0 2-bit ECC Error Flag	0: A 2-bit ECC error has not occurred. 1: A 2-bit ECC error has occurred.	R
b2	ERR11	LRAM1 1-bit ECC Error Flag	0: A 1-bit ECC error has not occurred. 1: A 1-bit ECC error has occurred.	R
b3	ERR12	LRAM1 2-bit ECC Error Flag	0: A 2-bit ECC error has not occurred. 1: A 2-bit ECC error has occurred.	R
b31 to b4	—	Reserved	These bits are read as 0	R

ERRn1 Flag (LRAMn 1-bit ECC Error Flag) (n = 0, 1)

This flag indicates whether a 1-bit ECC error has occurred. While the ECC function with error checking is enabled and updating of the information on 1-bit errors is enabled, the ERRn1 bit is set to 1 when a 1-bit error is detected. In addition, a reset or an interrupt request is generated according to the setting of the LRMCRn.OAD bit.

ERRn2 Flag (LRAMn 2-bit ECC Error Flag) (n = 0, 1)

This flag indicates whether a 2-bit ECC error has occurred. While the ECC function with error checking is enabled, the ERRn2 bit is set to 1 when a 2-bit error is detected. In addition, a reset or an interrupt request is generated according to the setting of the LRMCRn.OAD bit.

2.9.5.6 LRAM Error Flag Clear Register (LRMESCLR)

Address(es): LRAM.LRMESCLR 8101 0848h, LRAM_NS.LRMESCLR 0101 0848h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CLR12	CLR11	CLR02	CLR01
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR01	LRAM0 1-bit ECC Error Flag Clear	This bit is read as 0. Writing 1 to this bit clears the LRMESR.ERR01 flag and the LRMEAR01.EAR[14:0] bits. Writing 0 is ignored.	R/W
b1	CLR02	LRAM0 2-bit ECC Error Flag Clear	This bit is read as 0. Writing 1 to this bit clears the LRMESR.ERR02 flag and the LRMEAR02.EAR[14:0] bits. Writing 0 is ignored.	R/W
b2	CLR11	LRAM1 1-bit ECC Error Flag Clear	This bit is read as 0. Writing 1 to this bit clears the LRMESR.ERR11 flag and the LRMEAR11.EAR[14:0] bits. Writing 0 is ignored.	R/W
b3	CLR12	LRAM1 2-bit ECC Error Flag Clear	This bit is read as 0. Writing 1 to this bit clears the LRMESR.ERR12 flag and the LRMEAR12.EAR[14:0] bits. Writing 0 is ignored.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register is used to clear the flags in the LRMESR register and the values in the LRMEARnm registers (n = 0, 1; m = 1, 2). Writing 1 to a bit clears the corresponding flag and register to 0.

2.9.5.7 LRAMn Error Address Register m (LRMEARnm) (n = 0, 1) (m = 1, 2)

Address(es): LRAM.LRMEAR01 8101 0850h, LRAM.LRMEAR02 8101 0854h, LRAM.LRMEAR11 8101 0858h, LRAM.LRMEAR12 8101 085Ch, LRAM_NS.LRMEAR01 0101 0850h, LRAM_NS.LRMEAR02 0101 0854h, LRAM_NS.LRMEAR11 0101 0858h, LRAM_NS.LRMEAR12 0101 085Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EAR[14]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EAR[13:0]														—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16 to b2	EAR[14:0]	Error Address	These are the upper 15 bits of the LRAM offset address, corresponding to bits 16 through 2 of the address. If an ECC occurs, the address of the error is stored in these bits.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The error address to be stored in the LRAM error address register is obtained from the address as seen from the CPU in the ways stated in Table 2.29.

Table 2.29 Calculation of Error Addresses (m = 1, 2)

Bus Master	LRAM0 or LRAM1	Address Calculation Expression
CPU	LRAM0	0000 0000h + LRMEAR0m
		8000 0000h + LRMEAR0m
	LRAM1	0002 0000h + LRMEAR1m
		8002 0000h + LRMEAR1m
Bus master	LRAM0	0280 0000h + LRMEAR0m
		8280 0000h + LRMEAR0m
	LRAM1	0282 0000h + LRMEAR1m
		8282 0000h + LRMEAR1m

LRMEAR01 and LRMEAR11 Registers

The addresses where 1-bit ECC errors were detected are stored in these registers. Each register holds the address of the first error to have been generated.

Table 2.30 Relationship between the 1-bit ECC Error Address Registers and the Bits for Clearing the Registers

LRAM0 or LRAM1	Register	Bit for Clearing the Register
LRAM0	LRMEAR01	LRMESCLR.CLR01 bit
LRAM1	LRMEAR11	LRMESCLR.CLR11 bit

LRMEAR02 and LRMEAR12 Registers

The addresses where 2-bit ECC errors were detected are stored in these registers. Each register holds the address of the first error to have been generated.

Table 2.31 Relationship between the 2-bit ECC Error Address Registers and the Bits for Clearing the Registers

LRAM0 or LRAM1	Register	Bit for Clearing the Register
LRAM0	LRMEAR02	LRMESCLR.CLR02 bit
LRAM1	LRMEAR12	LRMESCLR.CLR12 bit

2.9.6 Operations of Caches

2.9.6.1 Basic Operation of Caches

A cache operates when the following conditions are satisfied.

1. The cache is enabled, that is, the EN bit in the corresponding one of the IFCACTL and OACACTL registers is set to 1.
2. The access from the CPU is to a cacheable area (an area for which caching is enabled by the MPU).

A cache hit or miss is judged through comparison of the address requested by the CPU and the addresses stored in the tag fields.

Read miss	When the address requested by the CPU does not match the address in a tag field, the cache reads one line of data from memory, stores the data in the cache, and returns the requested data to the CPU.
Read hit	When the address requested by the CPU matches the address in a tag field, the cache reads the requested data from the data field in the cache and returns the data to the CPU. Therefore, no cycles of waiting are required when the cache is hit.
Write miss	When the address requested by the CPU does not match the address in a tag field, the cache writes data to memory. If the write allocation mode is specified (the OACAWTA.WA bit is set to 1) and the target area has the write allocation attribute, the cache reads one line of data from memory and stores the data in the cache. For the cache attributes with respect to access by the CPU, refer to the section on the MPU.
Write hit	When the address requested by the CPU matches the address in a tag field, the cache writes the data to itself. If the write-through mode is specified (the OACAWTA.WT bit is set to 1) or the target area has the write-through attribute, the cache also writes the data to memory. If the write-through mode is not specified (the OACAWTA.WT bit is set to 0) and the target area has the write-back attribute, the cache sets the dirty bit of the tag corresponding to the written data field. For the cache attributes with respect to access by the CPU, refer to the section on the MPU.

2.9.6.2 Cache Flush

To flush a cache, write 1 to the FL bit in the corresponding one of the IFCAFCT and OACAFCT registers or the FL bit in the corresponding one of the IFCACTL and OACACTL registers. Flushing a cache clears the valid (V) bit, dirty bit (D) bit, and tag address in each tag field. The ECC code for each tag address is initialized to the code generated for a value of 0. Note that the cache data are not cleared.

In the case of access to the same address but with different cache attributes, the contents of the cache must be written back and flushed before any data or instruction access proceeds with changed cache attributes at the target address. While the OACAWTA.WT bit is 1, however, the write-back operation is not required. Here, “same address” refers to addresses that correspond to the same cache line, based on the cache line size. For the cache attributes with respect to access by the CPU, refer to the section on the MPU.

2.9.6.3 LRU and Replace

The caches use the LRU (replacing the least recently used data) algorithm as the strategy for updating the cache. After CPU access is judged to be a cache hit or miss, the data in the caches are updated in order from that to which reference has not been made for the longest period.

2.9.6.4 Interrupt Source

The detection of an ECC error (local memory error) in a cache serves as a source for interrupt generation. Whether to generate a reset request or an interrupt request can be selected by the CAOAD.OAD bit. In the case of an interrupt, the interrupt is generated when at least one of the error flags in the IFCAEDST or OACAEDST register is set to 1 and continues to be generated until all error flags in the IFCAEDST and OACAEDST register are cleared.

2.9.6.5 Bus Error

Responses to bus errors from the bus slave are as shown in Table 2.32.

Table 2.32 Ways of Responding to Bus Errors

No.	State of Cache	Ways of Responding to Bus Errors
1	Cache off	The cache will return a bus error to the CPU.
2	Non-cacheable access	The cache will return a bus error to the CPU.
3	Cache fill	The cache will return a bus error to the CPU. A line that includes an error will not be registered in the cache memory.
4	Write hit	The cache will not return a bus error to the CPU.
5	Write miss	The cache will not return a bus error to the CPU.

2.9.7 Operations of the Branch Instruction Processing Accelerator

2.9.7.1 Judgment of Hit or Miss and Registration of Data

When the CPU executes a branch to an area that is in the scope of caching, the branch instruction processing accelerator searches for the destination address of the branch in the tag fields of the accelerator entries for branch instructions and the entry for the RTS instruction. If the result of the search is a hit*1, the data for that address are transferred to the CPU. If the result is a miss, the instruction at the branch destination is fetched, the destination address of the branch and the CPU state (the current mode (user or supervisor mode) and current world) are registered in the tag field, and one line (128 bits)*2 of data that include the instruction at the branch destination are registered in the data field. In addition, if the branch instruction is a branch to a subroutine (BSR instruction or JSR instruction), one line (128 bits)*2 of data that include the instruction that follows the branch instruction are registered in the accelerator entry for the RTS instruction regardless of the results of the search.

Note 1. The destination address of the branch and the CPU state (the current mode (user or supervisor mode) and WID) matching the contents of a tag is judged to be a hit.
Note 2. One line consists of two 64-bit data values placed on 64-bit boundaries.

2.9.7.2 Algorithm for Replacement

If all entries contain valid data when data are to be registered in the branch instruction processing accelerator, data are replaced according to the LRU algorithm.

2.9.7.3 Invalidation through the Branch Instruction Processing Accelerator Invalidation Setting Register

Use the branch instruction processing accelerator invalidation setting register (CPUBAINV) to invalidate the contents of the branch instruction processing accelerator. This invalidates all lines in the accelerator.

2.9.8 Operations of LRAM

2.9.8.1 Interrupt Source

The detection of an ECC error (local memory error) in the LRAM serves as a source for interrupt generation. Whether to generate a reset request or an interrupt request can be selected by the LRMCRn.OAD bit (n = 0 or 1). In the case of an interrupt, the interrupt is generated when at least one of the error flags in the LRMESR register is set to 1 and continues to be generated until all error flags in the LRMESR register are cleared. The source of the generated error can be checked by the LRMESR register.

2.9.8.2 Security Function

The registers for the LRAM and each of the LRAM units can be protected against non-secure or secure transactions through the use of the security attribute. If the security attribute of the LRAM registers is secure or that of an LRAM unit is secure, the security function prevents access by non-secure transactions by detecting a security error in response, thus protecting the registers and LRAM from non-secure access. If the security attribute of the LRAM registers is non-secure or that of an LRAM unit is non-secure, the security function prevents access by secure transactions by detecting a security error in response, thus protecting the registers and LRAM from secure access. A security attribute setting for the LRAM registers applies in common to the LRMCRn and LRMESCLR registers. In contrast, separate security attribute settings apply to the LRAM0 and LRAM1 areas, that is, each of the areas can separately be set as secure or non-secure. Table 2.33 lists the details of this. If an attempt at access fails due to the security function, an error response will be generated.

Table 2.33 Correspondence of Security Attributes to Accessibility

Security Attribute of LRAM Registers or LRAM Units	Security Attribute of Transactions	Write Access	Read Access
Secure	Secure	Enabled	Enabled
	Non-secure	Protected (not accessible)	Protected (not accessible)
Non-secure	Secure	Protected (not accessible)	Protected (not accessible)
	Non-secure	Enabled	Enabled

2.9.9 ECC

The cache and local RAM (LRAM) have ECC functionality.

2.9.9.1 ECC Function

The respective ECC functions can be enabled or disabled through the register bits listed in Table 2.34. The ECC function is disabled in the initial state. The type of ECC checking is SEC-DED (single (1-bit) error correction and double (2-bit) error detection) for the cache data and local RAM or SED-DED (single error detection and double error detection) for the cache tags. When the ECC function is enabled, a 7-bit ECC code is generated from each 32 bits of data and written to the cache, cache tag, or LRAM. If an ECC error is detected in access to a cache tag, the line is invalidated and handling of the access is as a cache miss. An error in a clean line can be corrected by filling the cache line again but an error in a dirty line cannot be restored to the correct state because data cannot be written back.

Table 2.34 ECC Function

Target Memory	Target Register Bit	Function
Cache	CAOAD.ECCMOD1 bit	0: ECC function is disabled
		1: ECC function is enabled
LRAM	LRMCRn.ECCMOD[1:0] bits	00: ECC function is disabled
		01: Setting prohibited
		10: ECC function is enabled without error checking
		11: ECC function is enabled with error checking

Table 2.35 shows the processing after an error while the ECC function is enabled. If a 1-bit or 2-bit error is present while the ECC function is disabled, the error is neither detected nor corrected. Before using the ECC function for the LRAM, initial writing to the LRAM area to be used in 32-bit units or initial writing with the ECC function enabled and error checking disabled is required.

Table 2.35 Processing after an Error

Target Memory	Processing after a 1-Bit Error	Processing after a 2-Bit Error
Cache*1	The error is corrected. When the CAOAD.E1STSEN bit is 1, the ESD1 bit in the corresponding one of the IFCAEDST or OACAEDST registers is set to 1.	The error is detected but not corrected. The ESD2 bit of the corresponding one of the IFCAEDST or OACAEDST registers is set to 1.
LRAM	• When the LRMCRn.ECCMOD[1:0] bits is 10b: The error is corrected. The LRMESR.ERRn1 bit is not updated. • When the LRMCRn.ECCMOD[1:0] bits is 11b: The error is corrected. When the LRMCRn.E1STSEN bit is 1, the LRMESR.ERRn1 bit is set to 1.	• When theLRMCRn.ECCMOD[1:0] bits is 10b: The error is neither detected nor corrected. • When the LRMCRn.ECCMOD[1:0] bits is 11b: The error is detected but not corrected. The LRMESR.ERRn2 bit is set to 1.

Note 1. When an ECC error is detected in the data field of a cache, only error correction and error status setting proceed but the cache is not invalidated. This differs from the processing after an ECC error in the tag field.

2.9.9.2 Methods of Testing the ECC Decoder Circuits for the Caches

The ECC circuits for the caches have diagnostic circuits that allow users to detect failures by testing the decoder circuits along the paths for reading cache tags and cache data. For the data cache, the ECC decoder circuits consist of one for each way for the cache tags, two for each way in the path for CPU reading of cache data, and two in the write-back path for cache data, for a total of 14. The instruction cache does not have decoders for the write-back path, so the total number is 12. In the ECC decoder test for cache data, conduct tests along the CPU read path (see section 2.9.9.2, (1)) and along the write-back path (see section 2.9.9.2, (2)). In the ECC decoder test for cache tags, conduct tests on all of ways 0 to 3 for the tags with the same flow of testing as for the CPU read path. The ECC decoders for some cache data are shared among ways 0 to 3, so testing the ECC decoders is possible without conducting tests on all ways.

(1) CPU Read Path Test

Table 2.36 lists the steps in the flow of testing for the ECC decoders and the ECC decoders for tags when the CPU reads data from a cache. There are 12 ECC decoders, so repeat the steps from (1) to (13) listed in Table 2.36 for each of the instruction cache and data cache, in accordance with Table 2.37.

Table 2.36 Flow of Testing when the CPU Reads Data from a Cache (1/2)

Step Number	Test Flow Details
(0)	When a cache is in use, disable the cache according to the procedure in section 2.13.2, Points to Note on Cache Operations before starting on the main flow below.
(1)	Initialize four ways of the target lines in the memory. To initialize one line, write data to the addresses with OFFSET = 00h to 1Fh. To initialize the data for way 1, write to the line at “address 0 + different tag address”. Initialize the data for ways 2 and 3 in the same way as well. Example: Address 0 = 2100 0000h Address 0 + different tag 1 = 2101 0000h Address 0 + different tag 2 = 2102 0000h Address 0 + different tag 3 = 2103 0000h
(2)	Enable the ECC. The ECC is enabled by writing 0000 0018h to the CAOAD register after writing 0000 00F1h to the CAPRCR register.
(3)	Enable the cache and wait for flushing to complete. [For the instruction cache] After writing 0000 0101h to the IFCACTL register, wait until the IFCAFCT register becomes 0. [For the data cache] After writing 0000 0101h to the OACACTL register, wait until the OACAFCT register becomes 0.
(4)	Have the CPU perform read access to fill the lines of way 0 in the cache.
(5)	Have the CPU perform read access to fill the lines of ways 1 to 3 in the cache.
(6)	Disable the cache and ECC. [For the instruction cache] The cache and ECC are disabled by writing 0000 0000h to the IFCACTL register and 0000 0000h to the CAOAD register. [For the data cache] The cache and ECC are disabled by writing 0000 0000h to the OACACTL register and 0000 0000h to the CAOAD register.
(7)	Read data from the cache memory to the test access data register. [For the instruction cache] By writing to the IFCATAA register, read data from the cache memory to the IFCATAD register. Before proceeding to the next step, read the IFCATAA register to confirm that it reflects the written data.*1 [For the data cache] By writing to the OACATAA register, read data from the cache memory to the OACATAD register. Before proceeding to the next step, read the OACATAA register to confirm that it reflects the written data.*1

Table 2.36 Flow of Testing when the CPU Reads Data from a Cache (2/2)

Step Number	Test Flow Details
(8)	Write data that have been changed by one bit to the test access data register. [For the instruction cache] Write data that have been changed by one bit to the IFCATAD register. When testing 2-bit ECC error detection, write the value of IFCATAD with two bits inverted. [For the data cache] Write data that have been changed by one bit to the OACATAD register. When testing 2-bit ECC error detection, write the value of OACATAD with two bits inverted.
(9)	Write the address for test access and data to the cache memory. [For the instruction cache] Write the contents of the IFCATAD register to the cache memory by writing to the IFCATAA register. [For the data cache] Write the contents of the OACATAD register to the cache memory by writing to the OACATAA register.
(10)	Enable the cache and ECC. [For the instruction cache] Write 0000 0018h to the CAOAD register to enable the ECC, and then write 0000 0001h to the IFCACTL register to enable the cache. [For the data cache] Write 0000 0018h to the CAOAD register to enable the ECC, and then write 0000 0001h to the OACACTL register to enable the cache.
(11)	Have the CPU read the target address. When testing way 1, have the CPU read “address 0 + different tag 1”. When testing way 2, have the CPU read “address 0 + different tag 2”. When testing way 3, have the CPU read “address 0 + different tag 3”.
(12)	Read the error detection status register to confirm that the register bits indicate the expected error state. [For the instruction cache] When testing 1-bit ECC error correction, check that the data read have been corrected. In each test, confirm that the value of the IFCAEDST register is the corresponding value in Table 2.38. [For the data cache] When testing 1-bit ECC error correction, check that the data read have been corrected. In each test, confirm that the value of the OACAEDST register is the corresponding value in Table 2.38.
(13)	Clear the error state. [For the instruction cache] The error state can be cleared by writing 0000 0000h to the IFCAEDST register. [For the data cache] The error state can be cleared by writing 0000 0000h to the OACAEDST register.

Note 1. When the CPU writes to the I/O registers, it executes subsequent instructions without waiting for completion of the writing. Regarding the specific method of waiting for reflection of the results of preceding instructions, see (2) Notes on Writing to I/O Registers of section 5, I/O Registers.

Table 2.37 Example Settings of the IFCATAA and OACATAA Registers

No.	Target Decoder for Testing	IFCATAA and OACATAA Registers Settings
1	ECC decoder for tags of way 0	WAY[1:0] = 00b, TARGET[2:0] = 010b or 100b
2	ECC decoder for tags of way 1	WAY[1:0] = 01b, TARGET[2:0] = 010b or 100b
3	ECC decoder for tags of way 2	WAY[1:0] = 10b, TARGET[2:0] = 010b or 100b
4	ECC decoder for tags of way 3	WAY[1:0] = 11b, TARGET[2:0] = 010b or 100b
5	ECC decoder for data of way 0 (offset higher-order part [63:32])	WAY[1:0] = 00b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 001b
6	ECC decoder for data of way 0 (offset lower-order part [31:0])	WAY[1:0] = 00b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 000b
7	ECC decoder for data of way 1 (offset higher-order part [63:32])	WAY[1:0] = 01b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 001b
8	ECC decoder for data of way 1 (offset lower-order part [31:0])	WAY[1:0] = 01b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 000b
9	ECC decoder for data of way 2 (offset higher-order part [63:32])	WAY[1:0] = 10b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 001b
10	ECC decoder for data of way 2 (offset lower-order part [31:0])	WAY[1:0] = 10b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 000b
11	ECC decoder for data of way 3 (offset higher-order part [63:32])	WAY[1:0] = 11b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 001b
12	ECC decoder for data of way 3 (offset lower-order part [31:0])	WAY[1:0] = 11b, TARGET[2:0] = 000b or 001b OFFSET[2:0] = 000b

Note: Set RW = 0 (reading) for step (7) and RW = 1 (writing) for step (9).

Table 2.38 Read Data Confirmation Values for Step (12)

Target	Error	IFCAEDST and OACAEDST Registers Settings
Data	1-bit ECC error	0000 0001h
	2-bit ECC error	0000 0002h
Data ECC	1-bit ECC error	0000 0001h
	2-bit ECC error	0000 0002h
Tag	1-bit ECC error	0000 0004h
	2-bit ECC error	0000 0010h
Tag ECC	1-bit ECC error	0000 0004h
	2-bit ECC error	0000 0010h

(2) Write-back Path Test

Table 2.39 shows the flow for testing of the ECC decoders at the time of write-back. The ECC decoders for the write-back path are only provided for the data cache. There are two decoders for the write-back path, which are the offset higher-order and lower-order decoders along the cache data path, so select way 0 (Nos. 5 and 6) from Table 2.37 in steps (7) and (9) and repeat the steps from (1) to (15) twice.

Table 2.39 Flow of Testing when the CPU Writes to the Data Cache (in the Case of Way 0*1)

Step Number	Test Flow Details
(0)	When a cache is in use, disable the cache according to the procedure in section 2.13.2, Points to Note on Cache Operations before starting on the main flow below.
	The steps from (1) to (9) are the same as those for the CPU read path test.
(10)	Replace the D bit of the cache tag with 1. Step 1: Set the OACATAA register according to the settings in Table 2.40 (RW = 0). Step 2: Read the OACATAA register to confirm that it reflects the written data.*2 Step 3: Obtain the logical OR of the OACATAD register value and 00000001h, write it to the OACATAD register, and confirm that result of writing is reflected. Step 4: Set the OACATAA register according to the settings in Table 2.40 (RW = 1). To conduct a test along a path other than that for way 0, see Nos. 7 to 12 in Table 2.37.
(11)	Make the LRU setting for the cache such that “the oldest is way 0”. Set the OACATAD and OACATAA registers according to the settings in Table 2.41. To conduct a test along a path other than that for way 0, see Table 2.42.
(12)	Enable the cache and ECC. Write 0000 0018h to the CAOAD register to enable the ECC, and then write 0000 0001h to the OACACTL register to enable the cache.
(13)	Have the CPU read from an address not registered in the cache to generate a cache miss. The different tag (not in cache) is a tag address different from the address of the way used in step (1). Example: address 0 + different tag (not in cache) = 2110 0000h
(14)	Check the write-back data. When testing the response to 1-bit ECC errors, check that the data read from address 0 again (the write-back value) has been corrected. When testing the response to 2-bit ECC errors, check that the data read from address 0 again (the write-back value) has not been corrected. Read the error detection status register (OACAEDST) to check that its value is the one for the corresponding type of error as listed in Table 2.43.
(15)	Clear the error state. The error state can be cleared by writing 0000 0000h to the OACAEDST register.

Note 1. The case of way 0 is given as an example. The same testing is also possible along the paths for ways 1 to 3.
Note 2. When the CPU writes to the I/O registers, it executes subsequent instructions without waiting for completion of the writing. Regarding the specific method of waiting for reflection of the results of preceding instructions, see (2) Notes on Writing to I/O Registers of section 5, I/O Registers.

Table 2.40 OACATAA Register Settings (in the Case of Way 0)

No.	Target Decoder for Testing	OACATAA Register Settings
1	ECC decoder for write-back (offset higher-order part)	WAY[1:0] = 00b, TARGET[2:0] = 010b OFFSET[2:0] = 001b
2	ECC decoder for write-back (offset lower-order part)	WAY[1:0] = 00b, TARGET[2:0] = 010b OFFSET[2:0] = 000b

Table 2.41 OACATAD and OACATAA Registers Settings (in the Case of Way 0)

No.	Target Decoder for Testing	OACATAD Register Setting	OACATAA Register Settings
1	ECC decoder for write-back (offset higher-order part)	0000 0014h	WAY[1:0] = 00b, TARGET[2:0] = 011b OFFSET[2:0] = 001b, RW = 1b
2	ECC decoder for write-back (offset lower-order part)	0000 0014h	WAY[1:0] = 00b, TARGET[2:0] = 011b OFFSET[2:0] = 000b, RW = 1b

Table 2.42 OACATAD Register Settings According to the Selected Way

	Way0	Way1	Way2	Way3
OACATAA.WAY	00b	01b	10b	11b
OACATAD	0000 0014h	0000 0009h	0000 001Eh	0000 0013h

Table 2.43 Read Data Confirmation Values for Step (14)

Target	Error	OACAEDST Register Setting
Data	1-bit ECC error	0000 0001h
	2-bit ECC error	0000 0002h
Data ECC	1-bit ECC error	0000 0001h
	2-bit ECC error	0000 0002h

2.9.9.3 Methods of Testing the ECC Decoder Circuits for the LRAM

The decoder circuits in the ECC of the LRAM allow users to detect failures through the flow described in Table 2.44.

Table 2.44 Flow of Testing the ECC Decoder Circuits for the LRAM

Step Number	Test Flow Details
(1)	Confirm that the operation of the preceding instruction is completed.*1
(2)	Write 0000 A501h to the LRMPRCR register to enable writing to the LRAM-related registers.
(3)	Write 0000 0008h to the LRMCRn register to enable the ECC function without error checking and disable the ECC bypass function.
(4)	Confirm that the operation of the preceding instruction is completed.*1
(5)	Write 4 bytes of data to the target address for testing. 7-bit ECC code is automatically updated.
(6)	Confirm that the operation of the preceding instruction is completed.*1
(7)	Write 0000 0080h to the LRMCRn register to disable the ECC function and enable the ECC bypass function.
(8)	Confirm that the operation of the preceding instruction is completed.*1
(9)	Read the target address for testing to obtain a 7-bit ECC code.
(10)	To generate an ECC 1-bit or 2-bit error, invert one or two bits of the data that were read in step (9) and write the data back to the target address.
(11)	Confirm that the operation of the preceding instruction is completed.*1
(12)	Write 0000 001Ch to the LRMCRn register to enable the ECC function with error checking, set 1-bit ECC error information updating to “enabled”, and disable the ECC bypass function.
(13)	Confirm that the operation of the preceding instruction is completed.*1
(14)	Read the target address for testing.
(15)	Confirm that the operation of the preceding instruction is completed.*1
(16)	Confirm the generation of ECC errors with LRMESR.

Note 1. When the CPU writes to the I/O registers, it executes subsequent instructions without waiting for completion of the writing. Regarding the specific method of waiting for reflection of the results of preceding instructions, see (2) Notes on Writing to I/O Registers of section 5, I/O Registers.

2.10 CPU Control Registers

2.10.1 CPU Function Lock Control Register (CPULOCKCR)

Address(es): CPU_CTRL.CPULOCKCR 8100 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCKSAU	LCKSMPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCKSMPU	Secure MPU Registers Write Disable	0: Writing to the target registers is enabled. 1: Writing to the target registers is disabled.	R/W*1
b1	LCKSAU	SAU Registers Write Disable	0: Writing to the target registers is enabled. 1: Writing to the target registers is disabled.	R/W*1
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only writing 1 to this bit is valid. Writing 0 to the bit when it is 1 does not clear it.

LCKSMPU Bit (Secure MPU Registers Write Disable)

Setting this bit to 1 disables writing to the MPU registers in the secure world through the CPU or debugger. This applies to the following MPU registers in the secure world:

MPU.RSPAGEn, MPU.REPAGEn, MPU.MPEN, MPU.MPBAC, MPU.MPECLR, MPU.MPESTS, MPU.MPDEA, MPU.MPSA, MPU.MPOPS, MPU.MPOPI, MPU.MHITI, MPU.MHITD

LCKSAU Bit (SAU Registers Write Disable)

Setting this bit to 1 disables writing to the registers in the security attribute unit (SAU). This applies to the following registers listed in section 56, Security Features:

SAU.AAS, SAU.SESTS, SAU.SECLR, SAU.SEA

2.10.2 CPU Non-secure Function Lock Control Register (CPULOCKCRNS)

Address(es): CPU_CTRL_NS.CPULOCKCRNS 0100 0004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCKNSMPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCKNSMPU	Non-secure MPU Registers Write Disable	0: Writing to the target registers is enabled. 1: Writing to the target registers is disabled.	R/W*1
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only writing 1 to this bit is valid. Writing 0 to the bit when it is 1 does not clear it.

LCKNSMPU Bit (Non-secure MPU Registers Write Disable)

Setting this bit to 1 disables writing to the MPU registers in the non-secure world through the CPU or debugger. This applies to the following MPU registers in the non-secure world:

MPU_NS.RSPAGEn, MPU_NS.REPAGEn, MPU_NS.MPEN, MPU_NS.MPBAC, MPU_NS.MPECLR, MPU_NS.MPESTS, MPU_NS.MPDEA, MPU_NS.MPSA, MPU_NS.MPOPS, MPU_NS.MPOPI, MPU_NS.MHITI, MPU_NS.MHITD

2.11 Operation of Instructions

2.11.1 Restrictions on RMPA and String-Manipulation Instructions

2.11.1.1 Transfer Size and Data Prefetching

The RMPA instruction and the string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) transfer data in longword units to speed up the reading of data from and writing of data to the memory. If the last of the data to be processed is less than a longword, data transfer proceeds with the sizes described below:

- RMPA, SSTR, SUNTIL, and SWHILE instructions: Size specified by the size specifier
- SCMPU, SMOVB, SMOVF, and SMOVU instructions: Byte

Additionally, in the above processing, the RMPA instruction and the string-manipulation instructions other than the SSTR instruction (that is, the SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) prefetch data when reading data from the memory. Data is prefetched from the prefetching start position with three bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

2.11.1.2 Access to the External Address Space

Although the external address space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU,SSTR, SUNTIL, and SWHILE instructions) to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external address space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.11.1.3 Access to I/O Registers

The allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

2.12 Numbers of Cycles

2.12.1 Instructions and Numbers of Cycles

Table 2.45 to Table 2.53 show the numbers of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
As, Ad: Accumulator
CR: Control register
dsp: displacement
pcdsp: displacement

Table 2.45 Numbers of Cycles for Arithmetic/logic Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Arithmetic/logic instructions (register-register, immediate-register)	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB, XOR} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd” • {CMP, TST} “#IMM, Rs”/“Rs, Rs2” • NOP • {ROLC, RORC, SAT} “Rd” • SBB “Rs, Rd” • {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”	1
	• DIV “#IMM, Rd”/“Rs, Rd”	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	2 to 18*1
	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	2
	• SATR	3
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• DIV “[Rs], Rd / dsp[Rs], Rd”	5 to 22*1
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	4 to 20*1
	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• RMPA.B	6+7×floor(n/4)+4×(n%4) n: Number of processed bytes*2
	• RMPA.W	6+5×floor(n/2)+4×(n%2) n: Number of processed words*2
	• RMPA.L	6+4n n: Number of processed longwords

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. floor (x): Maximum integer that is smaller than x.

Table 2.46 Numbers of Cycles for Transfer Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SC*Cnd* “Rd” • {STNZ, STZ} “#IMM, Rd”/ “Rs, Rd”	1
	• XCHG “Rs, Rd”	2
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • MOVLI “[Rs], Rd” • POP “Rd”	Throughput: 1 Latency: 2*1
	• POPC “CR”	Throughput: 3 Latency: 4*1
	• POPM “Rd-Rd2”	Throughput: n Latency: n+1 n: Number of registers*1, *2
Transfer instructions (store operation)	• MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]” • PUSH “Rs” • PUSHC “CR” • SC*Cnd* “[Rd]”/“dsp[Rd]” • MOVCO “Rs, [Rd]”	1
	• PUSHM “Rs-Rs2”	n n: Number of registers*3
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	2
Transfer instructions (memory-memory)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	3
Transfer instructions (bit field)	• {BFMOV, BFMOVZ} “#IMM, #IMM, #IMM, R, R”	1

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Note 2. The POPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the MOV instruction repeated for the number of specified registers.

Note 3. The PUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the MOV instruction repeated for the number of specified registers.

Table 2.47 Numbers of Cycles for Bit Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	1
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]”	3

Table 2.48 Numbers of Cycles for Branch Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Branch taken: 3 Branch not taken: 1
	• {RTE, TRRTE}	6
	• RTFI	3
	• RTS	5
	• RTSD “#IMM”	5
	• RTSD “#IMM, Rd-Rd2”	Throughput: n<5?5:1+n Latency: n<4?5:2+n n: Number of registers*1

?: Conditional operator

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Table 2.49 Numbers of Cycles for Single-Precision Floating-Point Operation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Single-precision floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/ “Rs, Rd”/ “Rs, Rs2, Rd”	2
	• FCMP “#IMM, Rs”/“Rs, Rs2”	1
	• FDIV “#IMM, Rd”/“Rs, Rd”	16
	• FMUL “#IMM, Rd”/ “Rs, Rd” / “Rs, Rs2, Rd”	2
	• FSQRT “Rs, Rd”	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	2
	• {FTOU, UTOF} “Rs, Rd”	2
Single-precision floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	18
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FSQRT “[Rs], Rd”/“dsp[Rs], Rd”	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• {FTOU, UTOF} “[Rs], Rd”/“dsp[Rs], Rd”	4

Table 2.50 Numbers of Cycles for DSP Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
DSP instructions	• {EMULA, EMACA, EMSBA, MULLH, MULHI, MULLO, MACLH, MACHI, MACLO, MSBLH, MSBHI, MSBLO}“Rs, Rs2, Ad” • {MVFACHI, MVFACMI, MVFACLO, MVFACGU} “#IMM, As, Rd” • {MVTACHI, MVTACLO, MVTACGU} “Rs, Ad” • {RDACW, RDACL, RACW, RACL} “#IMM, Ad”	1

Table 2.51 Numbers of Cycles for String Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
String manipulation instructions*1	• SCMPU	2+4×floor(n/4)+4×(n%4) n: Number of compared bytes*2
	• SMOVB	n>3?6+3×floor(n/4)+3×(n%4):2+3n n: Number of transferred bytes*2
	• SMOVF, SMOVU	2+3×floor(n/4)+3×(n%4) n: Number of transferred bytes*2
	• SSTR.B	2+floor(n/4)+n%4 n: Number of transferred bytes*2
	• SSTR.W	2+floor(n/2)+n%2 n: Number of transferred words*2
	• SSTR.L	2+n n: Number of transferred longwords
	• SUNTIL.B, SWHILE.B	3+3×floor(n/4)+3×(n%4) n: Number of compared bytes*2
	• SUNTIL.W, SWHILE.W	3+3×floor(n/2)+3×(n%2) n: Number of compared words*2
	• SUNTIL.L, SWHILE.L	3+3×n n: Number of compared longwords

?: Conditional operator

Note 1. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles if the end condition is satisfied during execution.
Note 2. floor (x): Maximum integer less than or equal to x.

Table 2.52 Numbers of Cycles for System Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
System manipulation instructions	• {CLRPSW, SETPSW}“flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL “#IMM”	1
	• TRMVTC “#IMM, CR”/“Rs, CR”	3
	• TRMVFC “CR, Rd”	2
	• RTE	6
	• RTFI	3

Table 2.53 Numbers of Cycles for Instructions for Register Bank Save Function

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Instructions for register bank save function	• SAVE “#IMM”/“R”	1
	• RSTR “#IMM”/“R”	3 to 6

2.12.2 Instructions and Numbers of Cycles (for the Double-Precision Floating-Point Coprocessor)

Table 2.54 shows the numbers of cycles for the double-precision floating-point related instructions. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
Rs, Rd: General-purpose register
dsp: displacement
DRs, DRs2, DRd, DRd2: Double-precision floating-point data register
DRHs, DRHd: Upper 32 bits in the double-precision floating-point data register
DRLs, DRLd: Lower 32 bits in the double-precision floating-point data register
DCRs, DCRs2, DCRd, DCRd2: Double-precision floating-point control register

Table 2.54 Numbers of Cycles for the Double-Precision Floating-Point Related Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Double-precision floating-point processing instructions	• {DADD, DSUB} “DRs, DRs2, DRd”	4*1
	• {DABS, DNEG} “DRs, DRd”	1
	• DCMPcm “DRs, DRs2”	4*1
	• DDIV “DRs, DRs2, DRd”	33*1
	• DMUL “DRs, DRs2, DRd”	5*1
	• DSQRT “DRs, DRd”	33*1
	• {DTOF, DTOI, DTOU, DROUND} “DRs, DRd”	4*1
	• {FTOD, ITOD, UTOD} “Rs, DRd”	4*1
Double-precision floating-point data transfer instructions	• DMOV “Rs, DRHd”/ “Rs, DRLd” / “DRHs, Rd” / “DRLs, Rd” / “DRs, DRd”/“#IMM, DRHd”/ “#IMM, DRLd”	1
	• DMOV “DRs, [Rd]”/ “DRs, dsp[Rd]”	1
	• DMOV “[Rs], DRd”/ “dsp[Rs], DRd”	Throughput: 1 Latency: 2*2
	• DPUSHM “DRs-DRs2” / “DCRs-DCRs2”	n n: Number of registers*4
	• DPOPM “DRd-DRd2” / “DCRd-DCRd2”	Throughput: n Latency: n + 1 n: Number of registers*2, *3
	• MVFDC “DCRs, Rd”	1
	• MVFDR	1
	• MVTDC “Rs, DCRd”	1

Note 1. The numbers of cycles for reference to the results of arithmetic operations by double-precision floating-point data transfer instructions and the DABS and DNEG instructions are the listed numbers of cycles plus 1.
Note 2. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.
Note 3. The DPOPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the DMOV instruction repeated for the number of specified registers.
Note 4. The DPUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the DMOV instruction repeated for the number of specified registers.

2.12.3 Numbers of Cycles for Response to Interrupts

Table 2.55 lists the numbers of cycles taken by processing for response to interrupts.

Table 2.55 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles (ICLK)	
CPU Numbers of cycles from notification to acceptance of the interrupt request	N cycles (CPUCLK0) (varies with the instruction being executed at the time the interrupt was received)	
CPU Pre-processing by hardware Saving the current PC and PSW values in RAM (or in control registers in the case of the fast interrupt) Reading of the vector Branching to the start of the exception handling routine	4 cycles (CPUCLK0)	6 cycles (CPUCLK0)

Times calculated from the values in Table 2.55 will be applicable when access to memory from the CPU is processed with no waiting. This MCU has a LRAM that allows no-wait access and cache memory. Numbers of cycles for response to interrupts can be minimized by placing program code, vectors and the stack in LRAM. In addition, by setting it to a cacheable area when placing it in MRAM or SRAM, the number of interrupt response cycles will be minimized if there is a cache hit. Place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the numbers of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.45 to Table 2.54.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 14.3.1, Timing of Acceptance and Saved PC Value.

2.12.4 Numbers of Cycles of Waiting

Access to the LRAM usually requires no cycles of waiting. However, the following cases do require waiting.

(1) Cycles of waiting are generated when the LRAM is written in 8- or 16-bit units while the ECC function is enabled (the ECCMOD1 bit in the corresponding LRMCRn register is 1 (n = 0 or 1).

(2) In cases of contention between access by the CPU and access by another bus master, the CPU takes priority and the bus interface of the LRAM causes the other bus master to wait for the number of cycles required for the completion of access by the CPU.

(3) If access by the CPU or another bus master to the LRAM leads to a security violation, the LRAM returns an AHB error and this is accompanied by the insertion of cycles of waiting.

2.13 Usage Notes

2.13.1 Notes on Self-Diagnosis of the RAM for the Register-Saving Banks

The register-saving banks in this MCU are configured of RAM. As the register-saving banks are equipped with a buffer, data may be read from the buffer rather than from the memory cells of the RAM when the same address is to be read by a RSTR instruction after a write operation by the SAVE instruction. When running self-diagnosis of the RAM in the register-saving banks, confirm that the data have actually been written to the memory by following the procedure below so that data will not be read from the buffer.

(1) Write data to the bank targeted for diagnosis with the SAVE instruction.
(2) Write data to a bank different from the bank in the procedure (1) with the SAVE instruction.
(3) Read the data from the bank in the procedure (1) with the RSTR instruction.

2.13.2 Points to Note on Cache Operations

If a cache is disabled by the EN bit of the IFCACTL or OACACTL register or a reset is applied, the contents of the cache are not automatically flushed. If valid data remain in the cache while the cache is disabled, a mismatch will occur between the contents of the cache and memory. Use the following procedures to enable and disable the caches.

(1) Disabling the instruction cache

1. Set the IFCACTL register to 0000 0100h to specify disabling and start flushing of the cache.
2. Wait until the IFCAFCT.FL bit becomes 0 to ensure that flushing has been completed.
   The cache is disabled after this.

Note: Make sure that the setting of the IFCAFCT.FL bit is 0 before starting the above procedure.

(2) Enabling the instruction cache after a reset

1. Set the IFCACTL register to 0000 0101h to specify enabling and start flushing of the cache.
2. Wait until the IFCAFCT.FL bit becomes 0 to ensure that flushing has been completed.
   The cache is enabled after this. The cache is automatically flushed in response to the initial write access to the IFCACTL or IFCAFCT register after a reset.

(3) Enabling the instruction cache after it has been disabled

1. Check that the IFCAFCT.FL bit is 0, that is, flushing has been completed.
2. Set the IFCACTL register to 0000 0001h to enable the cache.

(4) Disabling the data cache

1. Set the OACACTL register to 0000 0100h (write-back is disabled) or 0000 0300h (write-back is enabled) to specify disabling and start flushing of the cache.
2. Wait until the OACAFCT register becomes 0000 0000h to ensure that the write-back and flush operations have been completed. The cache is disabled after this.

Note: Make sure that the setting of the OACAFCT.FL bit is 0 before starting the above procedure.

(5) Enabling the data cache after a reset

1. Set the OACACTL register to 0000 0101h to specify enabling and start flushing of the cache.
2. Wait until the OACAFCT.FL bit becomes 0 to ensure that flushing has been completed.
   The cache is enabled after this. The cache is automatically flushed in response to the initial write access to the OACACTL or OACAFCT register after a reset. Even if the WB bit is set to 1 at the same time as the write access to either register, write-back does not proceed.

(6) Enabling the data cache after it has been disabled

1. Check that the OACAFCT register is 0000 0000h, that is, the write-back and flush operations have been completed.
2. Set the OACACTL register to 0000 0001h to enable the cache.

2.13.3 Note on Cache Coherency

Software is required to guarantee coherency between the caches and external memory or on-chip SRAM. When multiple bus masters (for example, the CPU and DMAC) share an area in the scope of a cache, flush and write back the cache contents as required.

2.13.4 Note on Cache Flush and Write-Back

When using the CPU to start a flush or write-back of the contents of a cache in the CPU, check that the corresponding one among the IACAFCT.FL bit, the OACAFCT.WB bit, and the OACAFCT.FL bit has become 0 to confirm the completion of the flush or write-back operation. Do not use a bus master other than the CPU to start a flush or write-back operation for a cache in the CPU. After the flush or write-back processing has begun, that processing takes priority over access by the CPU.

2.13.5 Note on Testing LRAM Memory Cells

As the LRAM units each have a write buffer, data may be read from the write buffer rather than the LRAM memory cells if a value is written to an address and then read from the same address. When testing LRAM memory cells by writing data and then reading the written data, use the procedure stated in Table 2.56 to avoid reading from the write buffer.

Table 2.56 Procedure of Testing LRAM Memory Cells

(1)	Write any value to the target address of testing.
(2)	In the same LRAM unit, write any value to an address which is at least four locations away from that in (1).
(3)	Read data from the address to which writing proceeded in (1).

2.13.6 Notes on Modifying Instruction Codes

Either of the following procedures is required when modifying instruction codes in memory.

- When caching is enabled and the area corresponding to the modification is cacheable, the procedure is as follows.
  - After completing the write operation, make the data cache write back to memory by writing 1 to the OACAFCT.WB bit.
  - Before executing the modified instruction, invalidate the instruction cache by writing 1 to the IFCAFCT.FL bit.
- When the branch instruction processing accelerator is enabled and the area corresponding to the modification is cacheable, the procedure is as follows.
  - Before executing the modified instruction, invalidate the branch instruction processing accelerator by writing 1 to the CPUBAINV.BAIV bit.

Before executing the modified instruction code, verify that the data have actually been written and can be read correctly by reading back the last of the data to have been written.

3. Operating Modes

3.1 Operating Mode Types and Selection

The operating mode is selectable in either of the following ways.

- Level on the mode-setting pin (MD) on release from a reset (RES# pin reset or power-on reset)
- Control by software following release from the reset state

Table 3.1 shows the relationships between the level on the mode-setting pin and the operating mode to be selected. For details on each of the operating modes, refer to section 3.3, Details of Operating Modes. Operation starts with the on-chip code MRAM enabled, regardless of the mode in which operation started.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pin on Release from the Reset State

Mode-Setting Pin		SYSCR0 Initial State
MD	Operating Mode	ROME
High*1, *2	Single-chip mode, boot mode (JTAG/SWD interface)*4	1 (On-chip code MRAM enabled)
Low*1	Boot mode (SCI/USB interface)	
Low → High*3	Boot mode (FINE interface)	

Note 1. Transitions of the operating mode proceed during the waiting time after RES# release or the power-on reset time. Do not change the input level on the MD pin during any of those periods. For details about the waiting time after RES# release and the power-on reset time, refer to section 67, Electrical Characteristics.
Note 2. After the chip starts in single-chip mode, the MD pin can also be used as a general-purpose I/O port pin or P315.
Note 3. This is controlled by the serial programmer or emulator.
Note 4. When a request to transit to boot mode is input from the JTAG or SWD interface while a reset is being applied through the RES# pin, the chip starts up in boot mode through the JTAG or SWD interface.

Table 3.2 gives a list of the operating mode settings that can be made with system control register 0 (SYSCR0).

Table 3.2 Selection of Operating Modes by Register Setting

SYSCR0	
ROME	Operating Mode
0*1	Single-chip mode (On-chip code MRAM disabled)
1	Single-chip mode (On-chip code MRAM enabled)

Note 1. Once the ROME bit is set to 0, it cannot be reverted to 1.

The endian is selectable in single-chip mode. Endian is selected by the endian selection bits (MDE[2:0]) in the endian select register (MDE). Table 3.3 lists the correspondence between the setting and endian. For details on selection of endian, refer to section 7.2.4, Endian Select Register (MDE). In boot mode, little endian is always selected regardless of the setting of the MDE[2:0] bits.

Table 3.3 Selection of Endian

MDE	
Setting of the MDE[2:0] Bits	Selected Endian
000b	Big endian
111b	Little endian

3.2 Register Descriptions

3.2.1 System Control Register 0 (SYSCR0)

Address(es): SYSC.SYSCR0 8701 9006h, SYSC_NS.SYSCR0 0701 9006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	ROME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ROME	On-Chip Code MRAM Enable	0: The on-chip code MRAM is disabled. 1: The on-chip code MRAM is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	SYSCR0 Key Code	These bits control permission and prohibition of writing to the SYSCR0 register. To modify the SYSCR0 register, write 5Ah to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Values written are not retained.

ROME Bit (On-Chip Code MRAM Enable)

The ROME bit enables or disables the on-chip code MRAM.

Once set to 0, it cannot be reverted to 1.

A 0 should not be written to this bit while a program is being executed from the on-chip code MRAM. After writing a 0 to this bit, be sure to disable the on-chip code MRAM by changing the ROME bit to 0 before proceeding with further processing. When the setting of the ROME bit is 0, access to the CS0 area of the external address space is enabled.

3.2.2 Operating Mode Security Attribution Register (MDSAR)

Address(es): SYSC.MDSAR 8701 93D4h, SYSC_NS.MDSAR 0701 93D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NONSEC8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	NONSEC8	Security Attribution	0: Secure 1: Non-secure	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

NONSEC8 Bit (Security Attribution)

The NONSEC8 bit is used to set the security attributes for the SYSCR0 register.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In single-chip mode, all I/O port pins are available for use as input or output port pins, inputs or outputs for peripheral functions, or as interrupt inputs.
The chip starts up in single-chip mode if the chip is released from the reset state while the signal on the MD pin is fixed to the high level. The on-chip code MRAM is enabled (SYSCR0.ROME bit = 1) at this time. The on-chip code MRAM can be disabled by software (by clearing the SYSCR0.ROME bit to 0), but it cannot be re-enabled (by setting the SYSCR0.ROME bit to 1) once this is done. When the setting of the ROME bit is 0, access to the CS0 area of the external address space is enabled.

3.3.2 Boot Mode (SCI Interface)

The on-chip MRAM modifying program (boot program) stored in a dedicated area within the MCU runs in this mode. On-chip MRAM units (code MRAM and data MRAM) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, refer to section 65, MRAM.
The chip starts up in boot mode (SCI interface) if the chip is released from the reset state while the signal on the MD pin is fixed to the low level.

3.3.3 Boot Mode (USB Interface)

The on-chip MRAM modifying program (boot program) stored in a dedicated area within the MCU runs in this mode. On-chip MRAM units (code MRAM and data MRAM) can be modified from outside the MCU by using the USB. For details, refer to section 65, MRAM.
The chip starts up in boot mode (USB interface) if the chip is released from the reset state while the signal on the MD pin is fixed to the low level.

3.3.4 Boot Mode (JTAG/SWD Interface)

The on-chip MRAM modifying program (boot program) stored in a dedicated area within the MCU runs in this mode. On-chip MRAM units (code MRAM and data MRAM) can be modified from outside the MCU by using the JTAG interface or SWD interface. For details, refer to section 65, MRAM.
When the MD pin is being driven high and a request to transit to boot mode is input from the JTAG or SWD interface while a reset is being applied through the RES# pin, the chip starts up in boot mode through the JTAG or SWD interface.

3.3.5 Boot Mode (FINE Interface)

The on-chip MRAM modifying program (boot program) stored in a dedicated area within the MCU runs in this mode. On-chip MRAM units (code MRAM and data MRAM) can be modified from outside the MCU by using a FINE connection. For details, refer to section 65, MRAM.
The chip starts up in boot mode (FINE interface) if the chip is released from the reset state while the signal on the MD pin is fixed to the low level and then driven high within 20 to 100 ms after that.

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pin

Figure 3.1 shows operating mode transitions determined by the settings of the MD pin.

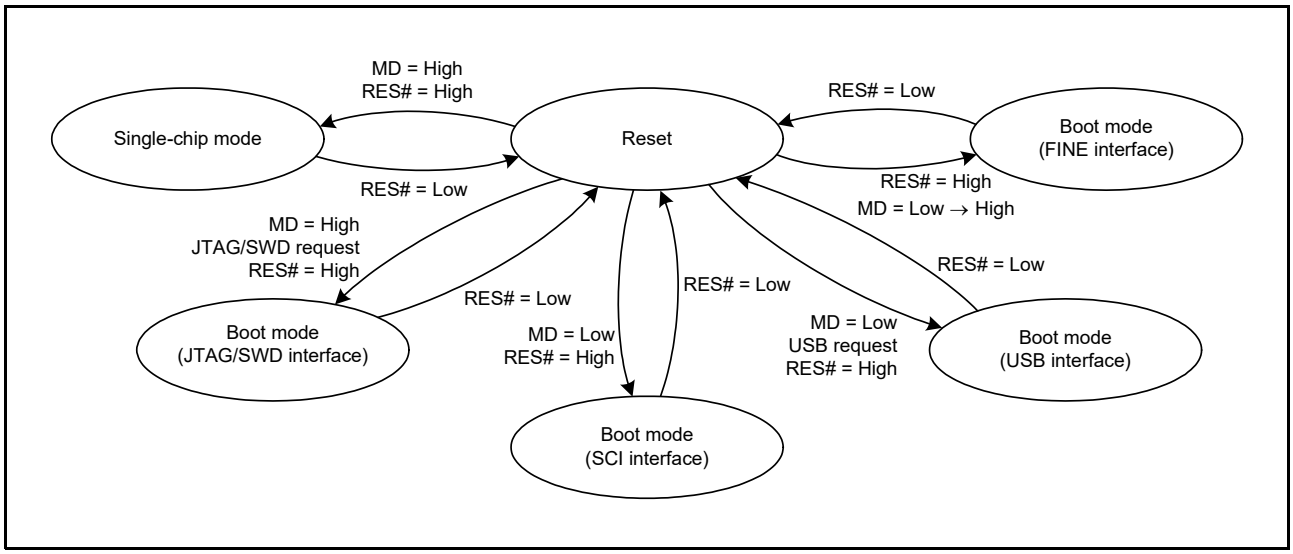


Figure 3.1 Level on the Mode-Setting Pin and Operating Mode

3.4.2 Operating Mode Transitions According to Register Setting

Figure 3.2 shows operating mode transitions according to the setting of the ROME bit in the SYSCR0 register.

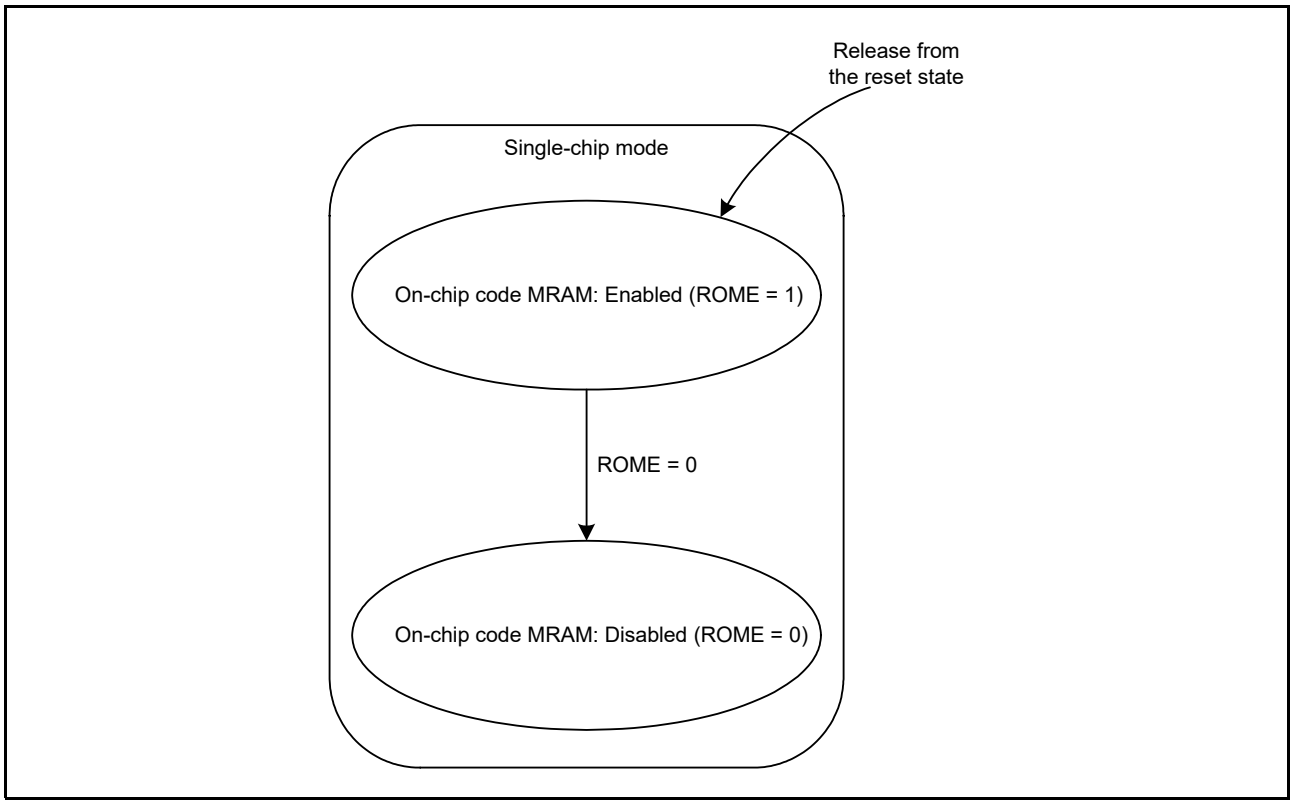


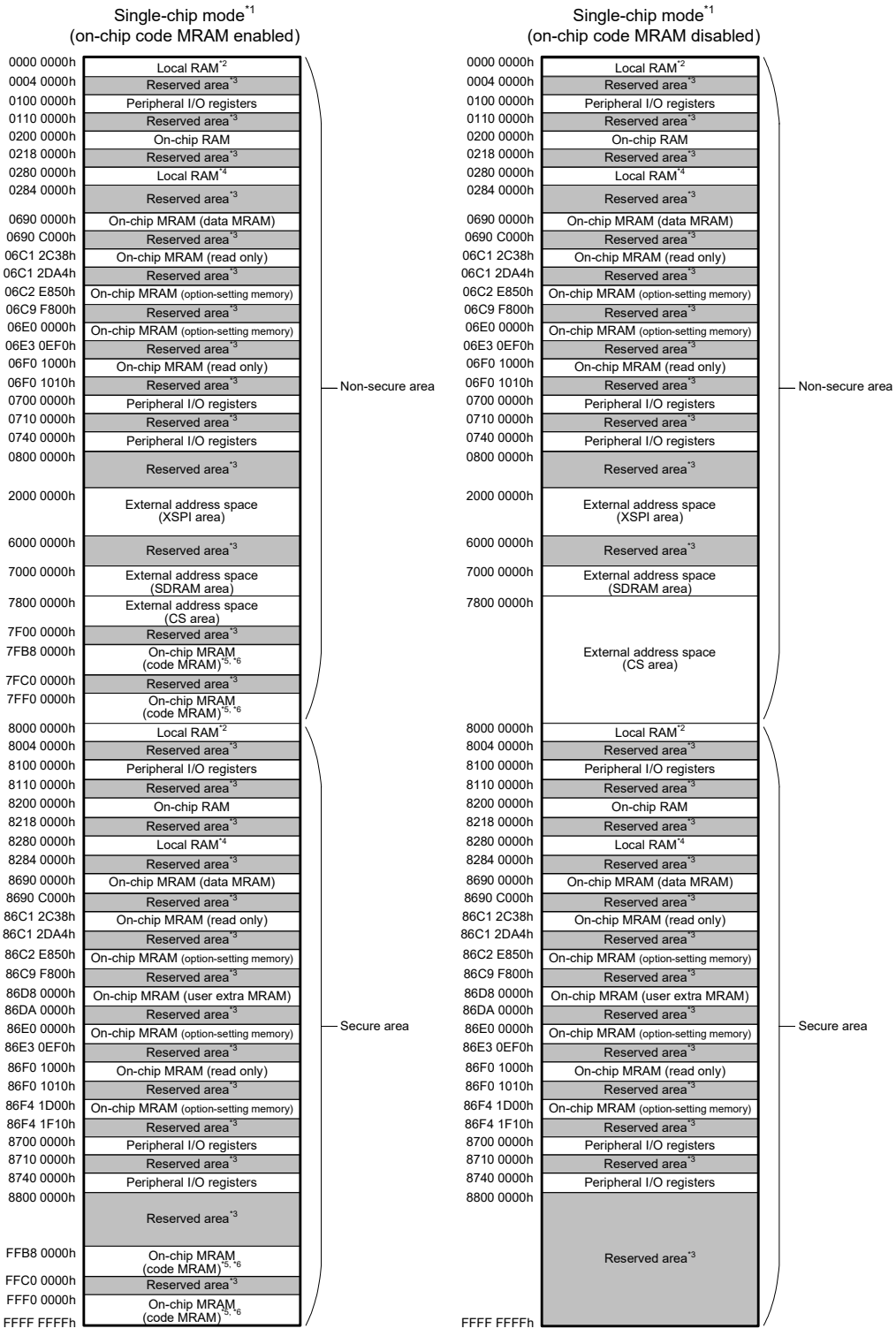
Figure 3.2 Setting of SYSCR0.ROME Bit and Operating Modes

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
The address space is divided into Secure area (addresses 8000 0000h to FFFF FFFFh) and Non-secure area (addresses 0000 0000h to 7FFF FFFFh).
For details on the security attributes, refer to section 56, Security Features.

Figure 4.1 shows the memory maps in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.



Note 1. The address space in boot mode is the same as the address space in single-chip mode.
Note 2. The address to the left applies when access is from the CPU.
Note 3. Reserved areas should not be accessed.
Note 4. The address to the left applies when access is from a bus master other than the CPU.
Note 5. The on-chip MRAM (code MRAM) can be used in linear mode, where the user area forms a single area, or in dual mode, where the user area is divided into two banks. For details, refer to section 65, MRAM.
Note 6. The capacities of the MRAM differ depending on the products. For details, refer to Table 4.1, Capacities of the MRAM (Secure Area) and Table 4.2, Capacities of the MRAM (Non-secure Area).

Figure 4.1 Memory Map in Each Operating Mode

Table 4.1 Capacities of the MRAM (Secure Area)

Code MRAM			Data MRAM		On-Chip RAM	
	Address					
Capacity	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)	Capacity	Address	Capacity	Address
1 Mbyte	FFF0 0000h to FFFF FFFFh	bank 1: FFB8 0000h to FFBF FFFFh	48 Kbytes	8690 0000h to 8690 BFFFh	1.5 Mbytes	8200 0000h to 8217 FFFFh
		bank 0: FFF8 0000h to FFFF FFFFh				
512 Kbytes	FFF8 0000h to FFFF FFFFh	bank 1: FFBC 0000h to FFBF FFFFh	48 Kbytes	8690 0000h to 8690 BFFFh	1.5 Mbytes	8200 0000h to 8217 FFFFh
		bank 0: FFFC 0000h to FFFF FFFFh				

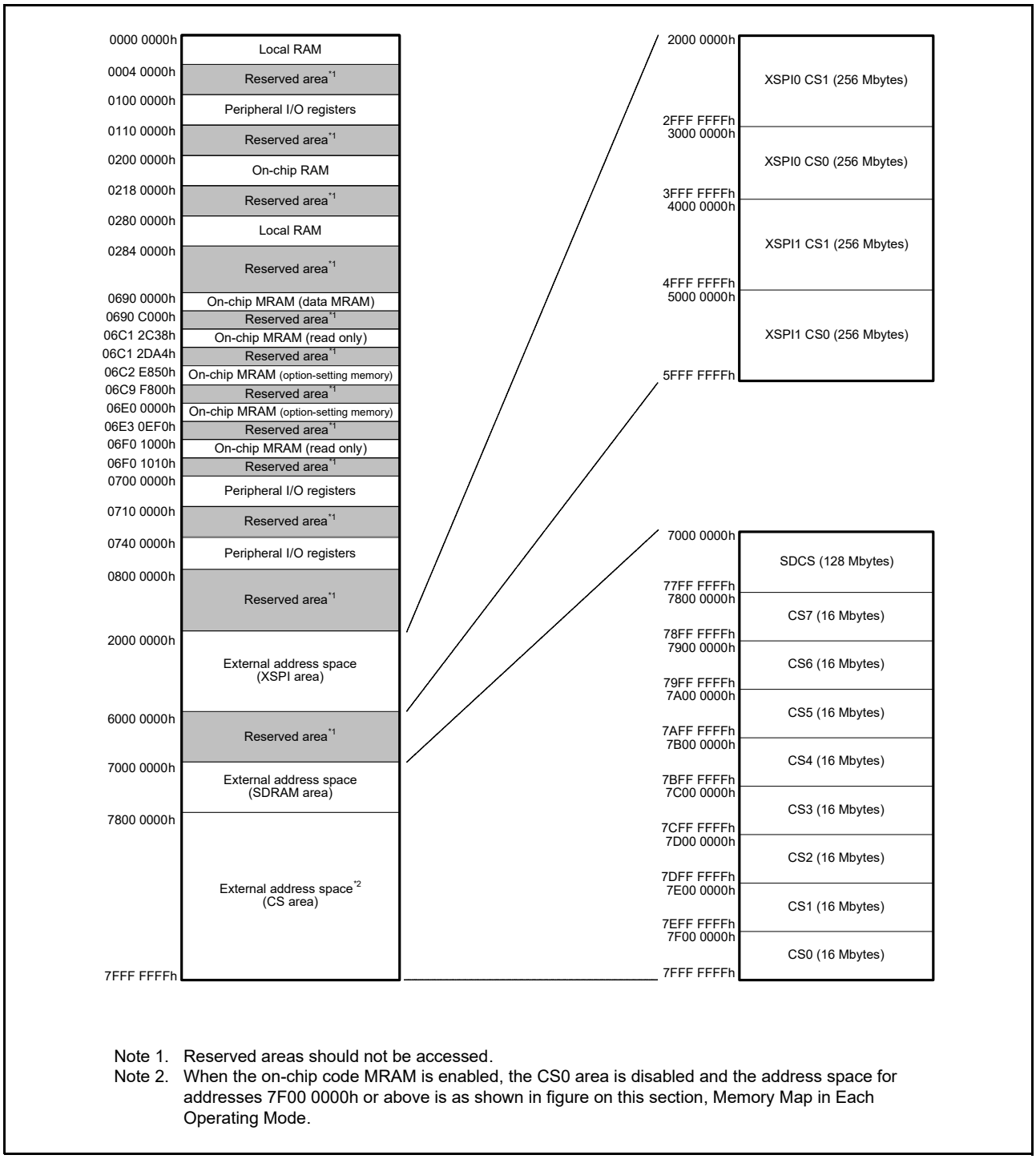
Table 4.2 Capacities of the MRAM (Non-secure Area)

Code MRAM			Data MRAM		On-Chip RAM	
	Address					
Capacity	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)	Capacity	Address	Capacity	Address
1 Mbyte	7FF0 0000h to 7FFF FFFFh	bank 1: 7FB8 0000h to 7FBF FFFFh	48 Kbytes	0690 0000h to 0690 BFFFh	1.5 Mbytes	0200 0000h to 0217 FFFFh
		bank 0: 7FF8 0000h to 7FFF FFFFh				
512 Kbytes	7FF8 0000h to FFFF FFFFh	bank 1: 7FBC 0000h to 7FBF FFFFh	48 Kbytes	0690 0000h to 0690 BFFFh	1.5 Mbytes	0200 0000h to 0217 FFFFh
		bank 0: 7FFC 0000h to 7FFF FFFFh				

4.2 External Address Space

The external address space is divided into CS areas (CS0 to CS7), an SDRAM area (SDCS), and XSPI areas (XSPImCS0, XSPImCS1) (m = 0, 1). The CS areas are divided into up to eight areas (CS0 to CS7), each corresponding to the CSn# signal output from a CSn# (n = 0 to 7) pin. The XSPI area can be divided into up to four XSPI areas (XSPImCS0, XSPImCS1), each corresponding to an XSPIm_CSn# signal output from an XSPIm_CSn# pin (m = 0, 1, n = 0, 1).

Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS7), SDRAM area (SDCS), and XSPI areas (XSPImCS0, XSPImCS1) (m = 0, 1) when the on-chip code MRAM is disabled.



Note 1. Reserved areas should not be accessed.
Note 2. When the on-chip code MRAM is enabled, the CS0 area is disabled and the address space for addresses 7F00 0000h or above is as shown in figure on this section, Memory Map in Each Operating Mode.

Figure 4.2 Correspondence between External Address Spaces and CS Areas (On-Chip Code MRAM Disabled)

5. I/O Registers

The coverage of registers in this section is a summary of information on the on-chip I/O registers that have multiple access sizes. For the other on-chip registers, see the respective sections. The information is given in the form described below. Notes on writing to registers are also given at the end.

(1) I/O Register Addresses (Address Order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Within the internal I/O register area, addresses not stated in the descriptions of registers in each section are reserved. Do not access these addresses; otherwise, the operation in access to the corresponding bits and other subsequent operations cannot be guaranteed.

(2) Notes on Writing to I/O Registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.
As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERm of the ICU (interrupt request enable bit) set to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value from the I/O register to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

[Instruction examples]

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

If multiple registers are written to and a subsequent instruction should be executed after the write operations are entirely completed, only read the I/O register that was last written to and execute the operation using the value; it is not necessary to read or execute operation for all the registers that were written to.

(3) Notes on Sleep Mode and Mode Transitions

Writing to the registers related to system control (indicated by 'SYSC' in the addresses of registers in each section) while in sleep mode, deep sleep mode, or during mode transitions is prohibited.

(4) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
8741 0004h	CRC	CRC Data Input Register	CRCDIR	32	8, 32
8741 0008h	CRC	CRC Data Output Register	CRCDOR	32	8, 16, 32
8741 100Ch	DOC	DOC Data Input Register	DODIR	32	16, 32
8741 1010h	DOC	DOC Data Setting Register 0	DODSR0	32	16, 32
8741 1014h	DOC	DOC Data Setting Register 1	DODSR1	32	16, 32
8745 1014h	USBHS	CFIFO Port Register	CFIFO	32	8, 16, 32
8745 1018h	USBHS	D0FIFO Port Register	D0FIFO	32	8, 16, 32
8745 101Ch	USBHS	D1FIFO Port Register	D1FIFO	32	8, 16, 32
8745 8000h	RSCI0	Receive Data Register	RDR	32	8, 16, 32
8745 8004h	RSCI0	Transmit Data Register	TDR	32	8, 16, 32
8745 8100h	RSCI1	Receive Data Register	RDR	32	8, 16, 32
8745 8104h	RSCI1	Transmit Data Register	TDR	32	8, 16, 32
8745 8200h	RSCI2	Receive Data Register	RDR	32	8, 16, 32
8745 8204h	RSCI2	Transmit Data Register	TDR	32	8, 16, 32
8745 8300h	RSCI3	Receive Data Register	RDR	32	8, 16, 32
8745 8304h	RSCI3	Transmit Data Register	TDR	32	8, 16, 32
8745 8400h	RSCI4	Receive Data Register	RDR	32	8, 16, 32
8745 8404h	RSCI4	Transmit Data Register	TDR	32	8, 16, 32
8745 8500h	RSCI5	Receive Data Register	RDR	32	8, 16, 32
8745 8504h	RSCI5	Transmit Data Register	TDR	32	8, 16, 32
8745 8600h	RSCI6	Receive Data Register	RDR	32	8, 16, 32
8745 8604h	RSCI6	Transmit Data Register	TDR	32	8, 16, 32
8745 8700h	RSCI7	Receive Data Register	RDR	32	8, 16, 32
8745 8704h	RSCI7	Transmit Data Register	TDR	32	8, 16, 32
8745 8800h	RSCI8	Receive Data Register	RDR	32	8, 16, 32
8745 8804h	RSCI8	Transmit Data Register	TDR	32	8, 16, 32
8745 8900h	RSCI9	Receive Data Register	RDR	32	8, 16, 32
8745 8904h	RSCI9	Transmit Data Register	TDR	32	8, 16, 32
8745 8A00h	RSCI10	Receive Data Register	RDR	32	8, 16, 32
8745 8A04h	RSCI10	Transmit Data Register	TDR	32	8, 16, 32
8745 8B00h	RSCI11	Receive Data Register	RDR	32	8, 16, 32
8745 8B04h	RSCI11	Transmit Data Register	TDR	32	8, 16, 32
8745 8C00h	RSCI12	Receive Data Register	RDR	32	8, 16, 32
8745 8C04h	RSCI12	Transmit Data Register	TDR	32	8, 16, 32
8745 C000h	RSPIA0	RSPI Data Register	SPDR	32	8, 16, 32
8745 C100h	RSPIA1	RSPI Data Register	SPDR	32	8, 16, 32
8745 C200h	RSPIA2	RSPI Data Register	SPDR	32	8, 16, 32
8745 F158h	RI3C0	Transmit/Receive Data Register	ICDR	32	8, 32
8745 F558h	RIICHS1	Transmit/Receive Data Register	ICDR	32	8, 32
8745 F958h	RIICHS2	Transmit/Receive Data Register	ICDR	32	8, 32
8747 0520h to 8747 056Bh	CANFD	Receive FIFO 0	RFB0	608	8, 16, 32
8747 056Ch to 8747 05B7h	CANFD	Receive FIFO 1	RFB1	608	8, 16, 32
8747 05B8h to 8747 0603h	CANFD	Common FIFO 0	CFB0	608	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (2/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
8747 0604h to 8747 064Fh	CANFD	Transmit Message Buffer 0	TMB0	608	8, 16, 32
8747 0650h to 8747 069Bh	CANFD	Transmit Message Buffer 1	TMB1	608	8, 16, 32
8747 069Ch to 8747 06E7h	CANFD	Transmit Message Buffer 2	TMB2	608	8, 16, 32
8747 06E8h to 8747 0733h	CANFD	Transmit Message Buffer 3	TMB3	608	8, 16, 32
8747 0920h to 8747 096Bh	CANFD	Receive Message Buffer 0	RMB0	608	8, 16, 32
8747 096Ch to 8747 09B7h	CANFD	Receive Message Buffer 1	RMB1	608	8, 16, 32
8747 09B8h to 8747 0A03h	CANFD	Receive Message Buffer 2	RMB2	608	8, 16, 32
8747 0A04h to 8747 0A4Fh	CANFD	Receive Message Buffer 3	RMB3	608	8, 16, 32
8747 0A50h to 8747 0A9Bh	CANFD	Receive Message Buffer 4	RMB4	608	8, 16, 32
8747 0A9Ch to 8747 0AE7h	CANFD	Receive Message Buffer 5	RMB5	608	8, 16, 32
8747 0AE8h to 8747 0B33h	CANFD	Receive Message Buffer 6	RMB6	608	8, 16, 32
8747 0B34h to 8747 0B7Fh	CANFD	Receive Message Buffer 7	RMB7	608	8, 16, 32
8747 0D20h to 8747 0D6Bh	CANFD	Receive Message Buffer 8	RMB8	608	8, 16, 32
8747 0D6Ch to 8747 0DB7h	CANFD	Receive Message Buffer 9	RMB9	608	8, 16, 32
8747 0DB8h to 8747 0E03h	CANFD	Receive Message Buffer 10	RMB10	608	8, 16, 32
8747 0E04h to 8747 0E4Fh	CANFD	Receive Message Buffer 11	RMB11	608	8, 16, 32
8747 0E50h to 8747 0E9Bh	CANFD	Receive Message Buffer 12	RMB12	608	8, 16, 32
8747 0E9Ch to 8747 0EE7h	CANFD	Receive Message Buffer 13	RMB13	608	8, 16, 32
8747 0EE8h to 8747 0F33h	CANFD	Receive Message Buffer 14	RMB14	608	8, 16, 32
8747 0F34h to 8747 0F7Fh	CANFD	Receive Message Buffer 15	RMB15	608	8, 16, 32
8748 2000h to 8748 204Bh	CANFDG0	Receive Message Buffer 0	RMB0	608	8, 16, 32
8748 2080h to 8748 20CBh	CANFDG0	Receive Message Buffer 1	RMB1	608	8, 16, 32
8748 2100h to 8748 214Bh	CANFDG0	Receive Message Buffer 2	RMB2	608	8, 16, 32
8748 2180h to 8748 21CBh	CANFDG0	Receive Message Buffer 3	RMB3	608	8, 16, 32
8748 2200h to 8748 224Bh	CANFDG0	Receive Message Buffer 4	RMB4	608	8, 16, 32
8748 2280h to 8748 22CBh	CANFDG0	Receive Message Buffer 5	RMB5	608	8, 16, 32
8748 2300h to 8748 234Bh	CANFDG0	Receive Message Buffer 6	RMB6	608	8, 16, 32
8748 2380h to 8748 23CBh	CANFDG0	Receive Message Buffer 7	RMB7	608	8, 16, 32
8748 2400h to 8748 244Bh	CANFDG0	Receive Message Buffer 8	RMB8	608	8, 16, 32
8748 2480h to 8748 24CBh	CANFDG0	Receive Message Buffer 9	RMB9	608	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (3/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
8748 2500h to 8748 254Bh	CANFDG0	Receive Message Buffer 10	RMB10	608	8, 16, 32
8748 2580h to 8748 25CBh	CANFDG0	Receive Message Buffer 11	RMB11	608	8, 16, 32
8748 2600h to 8748 264Bh	CANFDG0	Receive Message Buffer 12	RMB12	608	8, 16, 32
8748 2680h to 8748 26CBh	CANFDG0	Receive Message Buffer 13	RMB13	608	8, 16, 32
8748 2700h to 8748 274Bh	CANFDG0	Receive Message Buffer 14	RMB14	608	8, 16, 32
8748 2780h to 8748 27CBh	CANFDG0	Receive Message Buffer 15	RMB15	608	8, 16, 32
8748 2800h to 8748 284Bh	CANFDG0	Receive Message Buffer 16	RMB16	608	8, 16, 32
8748 2880h to 8748 28CBh	CANFDG0	Receive Message Buffer 17	RMB17	608	8, 16, 32
8748 2900h to 8748 294Bh	CANFDG0	Receive Message Buffer 18	RMB18	608	8, 16, 32
8748 2980h to 8748 29CBh	CANFDG0	Receive Message Buffer 19	RMB19	608	8, 16, 32
8748 2A00h to 8748 2A4Bh	CANFDG0	Receive Message Buffer 20	RMB20	608	8, 16, 32
8748 2A80h to 8748 2ACBh	CANFDG0	Receive Message Buffer 21	RMB21	608	8, 16, 32
8748 2B00h to 8748 2B4Bh	CANFDG0	Receive Message Buffer 22	RMB22	608	8, 16, 32
8748 2B80h to 8748 2BCBh	CANFDG0	Receive Message Buffer 23	RMB23	608	8, 16, 32
8748 2C00h to 8748 2C4Bh	CANFDG0	Receive Message Buffer 24	RMB24	608	8, 16, 32
8748 2C80h to 8748 2CCBh	CANFDG0	Receive Message Buffer 25	RMB25	608	8, 16, 32
8748 2D00h to 8748 2D4Bh	CANFDG0	Receive Message Buffer 26	RMB26	608	8, 16, 32
8748 2D80h to 8748 2DCBh	CANFDG0	Receive Message Buffer 27	RMB27	608	8, 16, 32
8748 2E00h to 8748 2E4Bh	CANFDG0	Receive Message Buffer 28	RMB28	608	8, 16, 32
8748 2E80h to 8748 2ECBh	CANFDG0	Receive Message Buffer 29	RMB29	608	8, 16, 32
8748 2F00h to 8748 2F4Bh	CANFDG0	Receive Message Buffer 30	RMB30	608	8, 16, 32
8748 2F80h to 8748 2FCBh	CANFDG0	Receive Message Buffer 31	RMB31	608	8, 16, 32
8748 6000h to 8748 604Bh	CANFDG0	Receive FIFO 0	RFB0	608	8, 16, 32
8748 6080h to 8748 60CBh	CANFDG0	Receive FIFO 1	RFB1	608	8, 16, 32
8748 6100h to 8748 614Bh	CANFDG0	Receive FIFO 2	RFB2	608	8, 16, 32
8748 6180h to 8748 61CBh	CANFDG0	Receive FIFO 3	RFB3	608	8, 16, 32
8748 6200h to 8748 624Bh	CANFDG0	Receive FIFO 4	RFB4	608	8, 16, 32
8748 6280h to 8748 62CBh	CANFDG0	Receive FIFO 5	RFB5	608	8, 16, 32
8748 6300h to 8748 634Bh	CANFDG0	Receive FIFO 6	RFB6	608	8, 16, 32
8748 6380h to 8748 63CBh	CANFDG0	Receive FIFO 7	RFB7	608	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (4/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
8748 6400h to 8748 644Bh	CANFDG0	Common FIFO 0	CFB0	608	8, 16, 32
8748 6480h to 8748 64CBh	CANFDG0	Common FIFO 1	CFB1	608	8, 16, 32
8748 6500h to 8748 654Bh	CANFDG0	Common FIFO 2	CFB2	608	8, 16, 32
8748 6580h to 8748 65CBh	CANFDG0	Common FIFO 3	CFB3	608	8, 16, 32
8748 6600h to 8748 664Bh	CANFDG0	Common FIFO 4	CFB4	608	8, 16, 32
8748 6680h to 8748 66CBh	CANFDG0	Common FIFO 5	CFB5	608	8, 16, 32
8749 0000h to 8749 004Bh	CANFDG0	Transmit Message Buffer 0	TMB0	608	8, 16, 32
8749 0080h to 8749 00CBh	CANFDG0	Transmit Message Buffer 1	TMB1	608	8, 16, 32
8749 0100h to 8749 014Bh	CANFDG0	Transmit Message Buffer 2	TMB2	608	8, 16, 32
8749 0180h to 8749 01CBh	CANFDG0	Transmit Message Buffer 3	TMB3	608	8, 16, 32
8749 0200h to 8749 024Bh	CANFDG0	Transmit Message Buffer 4	TMB4	608	8, 16, 32
8749 0280h to 8749 02CBh	CANFDG0	Transmit Message Buffer 5	TMB5	608	8, 16, 32
8749 0300h to 8749 034Bh	CANFDG0	Transmit Message Buffer 6	TMB6	608	8, 16, 32
8749 0380h to 8749 03CBh	CANFDG0	Transmit Message Buffer 7	TMB7	608	8, 16, 32
8749 1000h to 8749 104Bh	CANFDG0	Transmit Message Buffer 32	TMB32	608	8, 16, 32
8749 1080h to 8749 10CBh	CANFDG0	Transmit Message Buffer 33	TMB33	608	8, 16, 32
8749 1100h to 8749 114Bh	CANFDG0	Transmit Message Buffer 34	TMB34	608	8, 16, 32
8749 1180h to 8749 11CBh	CANFDG0	Transmit Message Buffer 35	TMB35	608	8, 16, 32
8749 1200h to 8749 124Bh	CANFDG0	Transmit Message Buffer 36	TMB36	608	8, 16, 32
8749 1280h to 8749 12CBh	CANFDG0	Transmit Message Buffer 37	TMB37	608	8, 16, 32
8749 1300h to 8749 134Bh	CANFDG0	Transmit Message Buffer 38	TMB38	608	8, 16, 32
8749 1380h to 8749 13CBh	CANFDG0	Transmit Message Buffer 39	TMB39	608	8, 16, 32
8749 2000h to 8749 204Bh	CANFDG0	Transmit Message Buffer 64	TMB64	608	8, 16, 32
8749 2080h to 8749 20CBh	CANFDG0	Transmit Message Buffer 65	TMB65	608	8, 16, 32
8749 2100h to 8749 214Bh	CANFDG0	Transmit Message Buffer 66	TMB66	608	8, 16, 32
8749 2180h to 8749 21CBh	CANFDG0	Transmit Message Buffer 67	TMB67	608	8, 16, 32
8749 2200h to 8749 224Bh	CANFDG0	Transmit Message Buffer 68	TMB68	608	8, 16, 32
8749 2280h to 8749 22CBh	CANFDG0	Transmit Message Buffer 69	TMB69	608	8, 16, 32
8749 2300h to 8749 234Bh	CANFDG0	Transmit Message Buffer 70	TMB70	608	8, 16, 32
8749 2380h to 8749 23CBh	CANFDG0	Transmit Message Buffer 71	TMB71	608	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (5/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
8749 3000h to 8749 304Bh	CANFDG0	Transmit Message Buffer 96	TMB96	608	8, 16, 32
8749 3080h to 8749 30CBh	CANFDG0	Transmit Message Buffer 97	TMB97	608	8, 16, 32
8749 3100h to 8749 314Bh	CANFDG0	Transmit Message Buffer 98	TMB98	608	8, 16, 32
8749 3180h to 8749 31CBh	CANFDG0	Transmit Message Buffer 99	TMB99	608	8, 16, 32
8749 3200h to 8749 324Bh	CANFDG0	Transmit Message Buffer 100	TMB100	608	8, 16, 32
8749 3280h to 8749 32CBh	CANFDG0	Transmit Message Buffer 101	TMB101	608	8, 16, 32
8749 3300h to 8749 334Bh	CANFDG0	Transmit Message Buffer 102	TMB102	608	8, 16, 32
8749 3380h to 8749 33CBh	CANFDG0	Transmit Message Buffer 103	TMB103	608	8, 16, 32
8785 0014h	USB0	CFIFO Port Register	CFIFO	16	8, 16
8785 0018h	USB0	D0FIFO Port Register	D0FIFO	16	8, 16
8785 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	8, 16
8785 D018h	SSIE0	Transmit FIFO Data Register	SSIFTDR	32	8, 16, 32
8785 D01Ch	SSIE0	Receive FIFO Data Register	SSIFRDR	32	8, 16, 32
8785 D118h	SSIE1	Transmit FIFO Data Register	SSIFTDR	32	8, 16, 32
8785 D11Ch	SSIE1	Receive FIFO Data Register	SSIFRDR	32	8, 16, 32
87C0 0000h	PORT0	Port Direction Register	PDR	16	8, 16
87C0 0002h	PORT0	Port Output Data Register	PODR	16	8, 16
87C0 0004h	PORT0	Port Input Data Register	PIDR	16	8, 16
87C0 0008h	PORT0	Port Output Set Register	POSR	16	8, 16
87C0 000Ah	PORT0	Port Output Reset Register	PORR	16	8, 16
87C0 0020h	PORT1	Port Direction Register	PDR	16	8, 16
87C0 0022h	PORT1	Port Output Data Register	PODR	16	8, 16
87C0 0024h	PORT1	Port Input Data Register	PIDR	16	8, 16
87C0 0028h	PORT1	Port Output Set Register	POSR	16	8, 16
87C0 002Ah	PORT1	Port Output Reset Register	PORR	16	8, 16
87C0 0040h	PORT2	Port Direction Register	PDR	16	8, 16
87C0 0042h	PORT2	Port Output Data Register	PODR	16	8, 16
87C0 0044h	PORT2	Port Input Data Register	PIDR	16	8, 16
87C0 0048h	PORT2	Port Output Set Register	POSR	16	8, 16
87C0 004Ah	PORT2	Port Output Reset Register	PORR	16	8, 16
87C0 0060h	PORT3	Port Direction Register	PDR	16	8, 16
87C0 0062h	PORT3	Port Output Data Register	PODR	16	8, 16
87C0 0064h	PORT3	Port Input Data Register	PIDR	16	8, 16
87C0 0068h	PORT3	Port Output Set Register	POSR	16	8, 16
87C0 006Ah	PORT3	Port Output Reset Register	PORR	16	8, 16
87C0 0080h	PORT4	Port Direction Register	PDR	16	8, 16
87C0 0082h	PORT4	Port Output Data Register	PODR	16	8, 16
87C0 0084h	PORT4	Port Input Data Register	PIDR	16	8, 16
87C0 0088h	PORT4	Port Output Set Register	POSR	16	8, 16
87C0 008Ah	PORT4	Port Output Reset Register	PORR	16	8, 16
87C0 00A0h	PORT5	Port Direction Register	PDR	16	8, 16
87C0 00A2h	PORT5	Port Output Data Register	PODR	16	8, 16
87C0 00A4h	PORT5	Port Input Data Register	PIDR	16	8, 16
87C0 00A8h	PORT5	Port Output Set Register	POSR	16	8, 16

Table 5.1 List of I/O Registers (Address Order) (6/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 00AAh	PORT5	Port Output Reset Register	PORR	16	8, 16
87C0 00C0h	PORT6	Port Direction Register	PDR	16	8, 16
87C0 00C2h	PORT6	Port Output Data Register	PODR	16	8, 16
87C0 00C4h	PORT6	Port Input Data Register	PIDR	16	8, 16
87C0 00C8h	PORT6	Port Output Set Register	POSR	16	8, 16
87C0 00CAh	PORT6	Port Output Reset Register	PORR	16	8, 16
87C0 00E0h	PORT7	Port Direction Register	PDR	16	8, 16
87C0 00E2h	PORT7	Port Output Data Register	PODR	16	8, 16
87C0 00E4h	PORT7	Port Input Data Register	PIDR	16	8, 16
87C0 00E8h	PORT7	Port Output Set Register	POSR	16	8, 16
87C0 00EAh	PORT7	Port Output Reset Register	PORR	16	8, 16
87C0 0100h	PORT8	Port Direction Register	PDR	16	8, 16
87C0 0102h	PORT8	Port Output Data Register	PODR	16	8, 16
87C0 0104h	PORT8	Port Input Data Register	PIDR	16	8, 16
87C0 0108h	PORT8	Port Output Set Register	POSR	16	8, 16
87C0 010Ah	PORT8	Port Output Reset Register	PORR	16	8, 16
87C0 0120h	PORT9	Port Direction Register	PDR	16	8, 16
87C0 0122h	PORT9	Port Output Data Register	PODR	16	8, 16
87C0 0124h	PORT9	Port Input Data Register	PIDR	16	8, 16
87C0 0128h	PORT9	Port Output Set Register	POSR	16	8, 16
87C0 012Ah	PORT9	Port Output Reset Register	PORR	16	8, 16
87C0 0140h	PORTA	Port Direction Register	PDR	16	8, 16
87C0 0142h	PORTA	Port Output Data Register	PODR	16	8, 16
87C0 0144h	PORTA	Port Input Data Register	PIDR	16	8, 16
87C0 0148h	PORTA	Port Output Set Register	POSR	16	8, 16
87C0 014Ah	PORTA	Port Output Reset Register	PORR	16	8, 16
87C0 0160h	PORTB	Port Direction Register	PDR	16	8, 16
87C0 0162h	PORTB	Port Output Data Register	PODR	16	8, 16
87C0 0164h	PORTB	Port Input Data Register	PIDR	16	8, 16
87C0 0166h	PORTB	Port Event Input Data Register	EIDR	16	8, 16
87C0 0168h	PORTB	Port Output Set Register	POSR	16	8, 16
87C0 016Ah	PORTB	Port Output Reset Register	PORR	16	8, 16
87C0 016Ch	PORTB	Event Output Set Register	EOSR	16	8, 16
87C0 016Eh	PORTB	Event Output Reset Register	EORR	16	8, 16
87C0 0180h	PORTC	Port Direction Register	PDR	16	8, 16
87C0 0182h	PORTC	Port Output Data Register	PODR	16	8, 16
87C0 0184h	PORTC	Port Input Data Register	PIDR	16	8, 16
87C0 0186h	PORTC	Port Event Input Data Register	EIDR	16	8, 16
87C0 0188h	PORTC	Port Output Set Register	POSR	16	8, 16
87C0 018Ah	PORTC	Port Output Reset Register	PORR	16	8, 16
87C0 018Ch	PORTC	Event Output Set Register	EOSR	16	8, 16
87C0 018Eh	PORTC	Event Output Reset Register	EORR	16	8, 16
87C0 01A0h	PORTD	Port Direction Register	PDR	16	8, 16
87C0 01A2h	PORTD	Port Output Data Register	PODR	16	8, 16
87C0 01A4h	PORTD	Port Input Data Register	PIDR	16	8, 16
87C0 01A6h	PORTD	Port Event Input Data Register	EIDR	16	8, 16
87C0 01A8h	PORTD	Port Output Set Register	POSR	16	8, 16
87C0 01AAh	PORTD	Port Output Reset Register	PORR	16	8, 16

Table 5.1 List of I/O Registers (Address Order) (7/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 01ACh	PORTD	Event Output Set Register	EOSR	16	8, 16
87C0 01AEh	PORTD	Event Output Reset Register	EORR	16	8, 16
87C0 01C0h	PORTE	Port Direction Register	PDR	16	8, 16
87C0 01C2h	PORTE	Port Output Data Register	PODR	16	8, 16
87C0 01C4h	PORTE	Port Input Data Register	PIDR	16	8, 16
87C0 01C6h	PORTE	Port Event Input Data Register	EIDR	16	8, 16
87C0 01C8h	PORTE	Port Output Set Register	POSR	16	8, 16
87C0 01CAh	PORTE	Port Output Reset Register	PORR	16	8, 16
87C0 01CCh	PORTE	Event Output Set Register	EOSR	16	8, 16
87C0 01CEh	PORTE	Event Output Reset Register	EORR	16	8, 16
87C0 01E0h	PORTF	Port Direction Register	PDR	16	8, 16
87C0 01E2h	PORTF	Port Output Data Register	PODR	16	8, 16
87C0 01E4h	PORTF	Port Input Data Register	PIDR	16	8, 16
87C0 01E8h	PORTF	Port Output Set Register	POSR	16	8, 16
87C0 01EAh	PORTF	Port Output Reset Register	PORR	16	8, 16
87C0 0200h	PORTG	Port Direction Register	PDR	16	8, 16
87C0 0202h	PORTG	Port Output Data Register	PODR	16	8, 16
87C0 0204h	PORTG	Port Input Data Register	PIDR	16	8, 16
87C0 0208h	PORTG	Port Output Set Register	POSR	16	8, 16
87C0 020Ah	PORTG	Port Output Reset Register	PORR	16	8, 16
87C0 0800h	PFS	P000 Pin Function Selection Register	P000PFS	32	8, 16, 32
87C0 0804h	PFS	P001 Pin Function Selection Register	P001PFS	32	8, 16, 32
87C0 0808h	PFS	P002 Pin Function Selection Register	P002PFS	32	8, 16, 32
87C0 080Ch	PFS	P003 Pin Function Selection Register	P003PFS	32	8, 16, 32
87C0 0810h	PFS	P004 Pin Function Selection Register	P004PFS	32	8, 16, 32
87C0 0814h	PFS	P005 Pin Function Selection Register	P005PFS	32	8, 16, 32
87C0 0818h	PFS	P006 Pin Function Selection Register	P006PFS	32	8, 16, 32
87C0 081Ch	PFS	P007 Pin Function Selection Register	P007PFS	32	8, 16, 32
87C0 0820h	PFS	P008 Pin Function Selection Register	P008PFS	32	8, 16, 32
87C0 0824h	PFS	P009 Pin Function Selection Register	P009PFS	32	8, 16, 32
87C0 0828h	PFS	P010 Pin Function Selection Register	P010PFS	32	8, 16, 32
87C0 082Ch	PFS	P011 Pin Function Selection Register	P011PFS	32	8, 16, 32
87C0 0830h	PFS	P012 Pin Function Selection Register	P012PFS	32	8, 16, 32
87C0 0834h	PFS	P013 Pin Function Selection Register	P013PFS	32	8, 16, 32
87C0 0838h	PFS	P014 Pin Function Selection Register	P014PFS	32	8, 16, 32
87C0 083Ch	PFS	P015 Pin Function Selection Register	P015PFS	32	8, 16, 32
87C0 0840h	PFS	P100 Pin Function Selection Register	P100PFS	32	8, 16, 32
87C0 0844h	PFS	P101 Pin Function Selection Register	P101PFS	32	8, 16, 32
87C0 0848h	PFS	P102 Pin Function Selection Register	P102PFS	32	8, 16, 32
87C0 084Ch	PFS	P103 Pin Function Selection Register	P103PFS	32	8, 16, 32
87C0 0850h	PFS	P104 Pin Function Selection Register	P104PFS	32	8, 16, 32
87C0 0854h	PFS	P105 Pin Function Selection Register	P105PFS	32	8, 16, 32
87C0 0858h	PFS	P106 Pin Function Selection Register	P106PFS	32	8, 16, 32
87C0 085Ch	PFS	P107 Pin Function Selection Register	P107PFS	32	8, 16, 32
87C0 0860h	PFS	P108 Pin Function Selection Register	P108PFS	32	8, 16, 32
87C0 0864h	PFS	P109 Pin Function Selection Register	P109PFS	32	8, 16, 32
87C0 0868h	PFS	P110 Pin Function Selection Register	P110PFS	32	8, 16, 32
87C0 086Ch	PFS	P111 Pin Function Selection Register	P111PFS	32	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (8/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 0870h	PFS	P112 Pin Function Selection Register	P112PFS	32	8, 16, 32
87C0 0874h	PFS	P113 Pin Function Selection Register	P113PFS	32	8, 16, 32
87C0 0878h	PFS	P114 Pin Function Selection Register	P114PFS	32	8, 16, 32
87C0 087Ch	PFS	P115 Pin Function Selection Register	P115PFS	32	8, 16, 32
87C0 0880h	PFS	P200 Pin Function Selection Register	P200PFS	32	8, 16, 32
87C0 0884h	PFS	P201 Pin Function Selection Register	P201PFS	32	8, 16, 32
87C0 0888h	PFS	P202 Pin Function Selection Register	P202PFS	32	8, 16, 32
87C0 088Ch	PFS	P203 Pin Function Selection Register	P203PFS	32	8, 16, 32
87C0 0890h	PFS	P204 Pin Function Selection Register	P204PFS	32	8, 16, 32
87C0 0894h	PFS	P205 Pin Function Selection Register	P205PFS	32	8, 16, 32
87C0 0898h	PFS	P206 Pin Function Selection Register	P206PFS	32	8, 16, 32
87C0 089Ch	PFS	P207 Pin Function Selection Register	P207PFS	32	8, 16, 32
87C0 08A0h	PFS	P208 Pin Function Selection Register	P208PFS	32	8, 16, 32
87C0 08A4h	PFS	P209 Pin Function Selection Register	P209PFS	32	8, 16, 32
87C0 08A8h	PFS	P210 Pin Function Selection Register	P210PFS	32	8, 16, 32
87C0 08ACh	PFS	P211 Pin Function Selection Register	P211PFS	32	8, 16, 32
87C0 08B0h	PFS	P212 Pin Function Selection Register	P212PFS	32	8, 16, 32
87C0 08B4h	PFS	P213 Pin Function Selection Register	P213PFS	32	8, 16, 32
87C0 08B8h	PFS	P214 Pin Function Selection Register	P214PFS	32	8, 16, 32
87C0 08BCh	PFS	P215 Pin Function Selection Register	P215PFS	32	8, 16, 32
87C0 08C0h	PFS	P300 Pin Function Selection Register	P300PFS	32	8, 16, 32
87C0 08C4h	PFS	P301 Pin Function Selection Register	P301PFS	32	8, 16, 32
87C0 08C8h	PFS	P302 Pin Function Selection Register	P302PFS	32	8, 16, 32
87C0 08CCh	PFS	P303 Pin Function Selection Register	P303PFS	32	8, 16, 32
87C0 08D0h	PFS	P304 Pin Function Selection Register	P304PFS	32	8, 16, 32
87C0 08D4h	PFS	P305 Pin Function Selection Register	P305PFS	32	8, 16, 32
87C0 08D8h	PFS	P306 Pin Function Selection Register	P306PFS	32	8, 16, 32
87C0 08DCh	PFS	P307 Pin Function Selection Register	P307PFS	32	8, 16, 32
87C0 08E0h	PFS	P308 Pin Function Selection Register	P308PFS	32	8, 16, 32
87C0 08E4h	PFS	P309 Pin Function Selection Register	P309PFS	32	8, 16, 32
87C0 08E8h	PFS	P310 Pin Function Selection Register	P310PFS	32	8, 16, 32
87C0 08ECh	PFS	P311 Pin Function Selection Register	P311PFS	32	8, 16, 32
87C0 08F0h	PFS	P312 Pin Function Selection Register	P312PFS	32	8, 16, 32
87C0 08FCh	PFS	P315 Pin Function Selection Register	P315PFS	32	8, 16, 32
87C0 0900h	PFS	P400 Pin Function Selection Register	P400PFS	32	8, 16, 32
87C0 0904h	PFS	P401 Pin Function Selection Register	P401PFS	32	8, 16, 32
87C0 0908h	PFS	P402 Pin Function Selection Register	P402PFS	32	8, 16, 32
87C0 090Ch	PFS	P403 Pin Function Selection Register	P403PFS	32	8, 16, 32
87C0 0910h	PFS	P404 Pin Function Selection Register	P404PFS	32	8, 16, 32
87C0 0914h	PFS	P405 Pin Function Selection Register	P405PFS	32	8, 16, 32
87C0 0918h	PFS	P406 Pin Function Selection Register	P406PFS	32	8, 16, 32
87C0 091Ch	PFS	P407 Pin Function Selection Register	P407PFS	32	8, 16, 32
87C0 0920h	PFS	P408 Pin Function Selection Register	P408PFS	32	8, 16, 32
87C0 0924h	PFS	P409 Pin Function Selection Register	P409PFS	32	8, 16, 32
87C0 0928h	PFS	P410 Pin Function Selection Register	P410PFS	32	8, 16, 32
87C0 092Ch	PFS	P411 Pin Function Selection Register	P411PFS	32	8, 16, 32
87C0 0930h	PFS	P412 Pin Function Selection Register	P412PFS	32	8, 16, 32
87C0 0934h	PFS	P413 Pin Function Selection Register	P413PFS	32	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (9/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 0938h	PFS	P414 Pin Function Selection Register	P414PFS	32	8, 16, 32
87C0 093Ch	PFS	P415 Pin Function Selection Register	P415PFS	32	8, 16, 32
87C0 0940h	PFS	P500 Pin Function Selection Register	P500PFS	32	8, 16, 32
87C0 0944h	PFS	P501 Pin Function Selection Register	P501PFS	32	8, 16, 32
87C0 0948h	PFS	P502 Pin Function Selection Register	P502PFS	32	8, 16, 32
87C0 094Ch	PFS	P503 Pin Function Selection Register	P503PFS	32	8, 16, 32
87C0 0950h	PFS	P504 Pin Function Selection Register	P504PFS	32	8, 16, 32
87C0 0954h	PFS	P505 Pin Function Selection Register	P505PFS	32	8, 16, 32
87C0 0958h	PFS	P506 Pin Function Selection Register	P506PFS	32	8, 16, 32
87C0 095Ch	PFS	P507 Pin Function Selection Register	P507PFS	32	8, 16, 32
87C0 0960h	PFS	P508 Pin Function Selection Register	P508PFS	32	8, 16, 32
87C0 0964h	PFS	P509 Pin Function Selection Register	P509PFS	32	8, 16, 32
87C0 0968h	PFS	P510 Pin Function Selection Register	P510PFS	32	8, 16, 32
87C0 096Ch	PFS	P511 Pin Function Selection Register	P511PFS	32	8, 16, 32
87C0 0970h	PFS	P512 Pin Function Selection Register	P512PFS	32	8, 16, 32
87C0 0974h	PFS	P513 Pin Function Selection Register	P513PFS	32	8, 16, 32
87C0 0978h	PFS	P514 Pin Function Selection Register	P514PFS	32	8, 16, 32
87C0 097Ch	PFS	P515 Pin Function Selection Register	P515PFS	32	8, 16, 32
87C0 0980h	PFS	P600 Pin Function Selection Register	P600PFS	32	8, 16, 32
87C0 0984h	PFS	P601 Pin Function Selection Register	P601PFS	32	8, 16, 32
87C0 0988h	PFS	P602 Pin Function Selection Register	P602PFS	32	8, 16, 32
87C0 098Ch	PFS	P603 Pin Function Selection Register	P603PFS	32	8, 16, 32
87C0 0990h	PFS	P604 Pin Function Selection Register	P604PFS	32	8, 16, 32
87C0 0994h	PFS	P605 Pin Function Selection Register	P605PFS	32	8, 16, 32
87C0 0998h	PFS	P606 Pin Function Selection Register	P606PFS	32	8, 16, 32
87C0 099Ch	PFS	P607 Pin Function Selection Register	P607PFS	32	8, 16, 32
87C0 09C0h	PFS	P700 Pin Function Selection Register	P700PFS	32	8, 16, 32
87C0 09C4h	PFS	P701 Pin Function Selection Register	P701PFS	32	8, 16, 32
87C0 09C8h	PFS	P702 Pin Function Selection Register	P702PFS	32	8, 16, 32
87C0 09CCh	PFS	P703 Pin Function Selection Register	P703PFS	32	8, 16, 32
87C0 09D0h	PFS	P704 Pin Function Selection Register	P704PFS	32	8, 16, 32
87C0 09D4h	PFS	P705 Pin Function Selection Register	P705PFS	32	8, 16, 32
87C0 09D8h	PFS	P706 Pin Function Selection Register	P706PFS	32	8, 16, 32
87C0 09DCh	PFS	P707 Pin Function Selection Register	P707PFS	32	8, 16, 32
87C0 09E0h	PFS	P708 Pin Function Selection Register	P708PFS	32	8, 16, 32
87C0 09E4h	PFS	P709 Pin Function Selection Register	P709PFS	32	8, 16, 32
87C0 09E8h	PFS	P710 Pin Function Selection Register	P710PFS	32	8, 16, 32
87C0 09ECh	PFS	P711 Pin Function Selection Register	P711PFS	32	8, 16, 32
87C0 09F0h	PFS	P712 Pin Function Selection Register	P712PFS	32	8, 16, 32
87C0 09F4h	PFS	P713 Pin Function Selection Register	P713PFS	32	8, 16, 32
87C0 09F8h	PFS	P714 Pin Function Selection Register	P714PFS	32	8, 16, 32
87C0 09FCh	PFS	P715 Pin Function Selection Register	P715PFS	32	8, 16, 32
87C0 0A00h	PFS	P800 Pin Function Selection Register	P800PFS	32	8, 16, 32
87C0 0A04h	PFS	P801 Pin Function Selection Register	P801PFS	32	8, 16, 32
87C0 0A08h	PFS	P802 Pin Function Selection Register	P802PFS	32	8, 16, 32
87C0 0A0Ch	PFS	P803 Pin Function Selection Register	P803PFS	32	8, 16, 32
87C0 0A10h	PFS	P804 Pin Function Selection Register	P804PFS	32	8, 16, 32
87C0 0A14h	PFS	P805 Pin Function Selection Register	P805PFS	32	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (10/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 0A18h	PFS	P806 Pin Function Selection Register	P806PFS	32	8, 16, 32
87C0 0A1Ch	PFS	P807 Pin Function Selection Register	P807PFS	32	8, 16, 32
87C0 0A40h	PFS	P900 Pin Function Selection Register	P900PFS	32	8, 16, 32
87C0 0A44h	PFS	P901 Pin Function Selection Register	P901PFS	32	8, 16, 32
87C0 0A48h	PFS	P902 Pin Function Selection Register	P902PFS	32	8, 16, 32
87C0 0A4Ch	PFS	P903 Pin Function Selection Register	P903PFS	32	8, 16, 32
87C0 0A50h	PFS	P904 Pin Function Selection Register	P904PFS	32	8, 16, 32
87C0 0A54h	PFS	P905 Pin Function Selection Register	P905PFS	32	8, 16, 32
87C0 0A58h	PFS	P906 Pin Function Selection Register	P906PFS	32	8, 16, 32
87C0 0A5Ch	PFS	P907 Pin Function Selection Register	P907PFS	32	8, 16, 32
87C0 0A80h	PFS	PA00 Pin Function Selection Register	PA00PFS	32	8, 16, 32
87C0 0A84h	PFS	PA01 Pin Function Selection Register	PA01PFS	32	8, 16, 32
87C0 0A88h	PFS	PA02 Pin Function Selection Register	PA02PFS	32	8, 16, 32
87C0 0A8Ch	PFS	PA03 Pin Function Selection Register	PA03PFS	32	8, 16, 32
87C0 0A90h	PFS	PA04 Pin Function Selection Register	PA04PFS	32	8, 16, 32
87C0 0A94h	PFS	PA05 Pin Function Selection Register	PA05PFS	32	8, 16, 32
87C0 0A98h	PFS	PA06 Pin Function Selection Register	PA06PFS	32	8, 16, 32
87C0 0A9Ch	PFS	PA07 Pin Function Selection Register	PA07PFS	32	8, 16, 32
87C0 0AC0h	PFS	PB00 Pin Function Selection Register	PB00PFS	32	8, 16, 32
87C0 0AC4h	PFS	PB01 Pin Function Selection Register	PB01PFS	32	8, 16, 32
87C0 0AC8h	PFS	PB02 Pin Function Selection Register	PB02PFS	32	8, 16, 32
87C0 0ACCh	PFS	PB03 Pin Function Selection Register	PB03PFS	32	8, 16, 32
87C0 0AD0h	PFS	PB04 Pin Function Selection Register	PB04PFS	32	8, 16, 32
87C0 0AD4h	PFS	PB05 Pin Function Selection Register	PB05PFS	32	8, 16, 32
87C0 0AD8h	PFS	PB06 Pin Function Selection Register	PB06PFS	32	8, 16, 32
87C0 0ADCh	PFS	PB07 Pin Function Selection Register	PB07PFS	32	8, 16, 32
87C0 0B00h	PFS	PC00 Pin Function Selection Register	PC00PFS	32	8, 16, 32
87C0 0B04h	PFS	PC01 Pin Function Selection Register	PC01PFS	32	8, 16, 32
87C0 0B08h	PFS	PC02 Pin Function Selection Register	PC02PFS	32	8, 16, 32
87C0 0B0Ch	PFS	PC03 Pin Function Selection Register	PC03PFS	32	8, 16, 32
87C0 0B10h	PFS	PC04 Pin Function Selection Register	PC04PFS	32	8, 16, 32
87C0 0B14h	PFS	PC05 Pin Function Selection Register	PC05PFS	32	8, 16, 32
87C0 0B18h	PFS	PC06 Pin Function Selection Register	PC06PFS	32	8, 16, 32
87C0 0B1Ch	PFS	PC07 Pin Function Selection Register	PC07PFS	32	8, 16, 32
87C0 0B40h	PFS	PD00 Pin Function Selection Register	PD00PFS	32	8, 16, 32
87C0 0B44h	PFS	PD01 Pin Function Selection Register	PD01PFS	32	8, 16, 32
87C0 0B48h	PFS	PD02 Pin Function Selection Register	PD02PFS	32	8, 16, 32
87C0 0B4Ch	PFS	PD03 Pin Function Selection Register	PD03PFS	32	8, 16, 32
87C0 0B50h	PFS	PD04 Pin Function Selection Register	PD04PFS	32	8, 16, 32
87C0 0B54h	PFS	PD05 Pin Function Selection Register	PD05PFS	32	8, 16, 32
87C0 0B58h	PFS	PD06 Pin Function Selection Register	PD06PFS	32	8, 16, 32
87C0 0B5Ch	PFS	PD07 Pin Function Selection Register	PD07PFS	32	8, 16, 32
87C0 0B80h	PFS	PE00 Pin Function Selection Register	PE00PFS	32	8, 16, 32
87C0 0B84h	PFS	PE01 Pin Function Selection Register	PE01PFS	32	8, 16, 32
87C0 0B88h	PFS	PE02 Pin Function Selection Register	PE02PFS	32	8, 16, 32
87C0 0B8Ch	PFS	PE03 Pin Function Selection Register	PE03PFS	32	8, 16, 32
87C0 0B90h	PFS	PE04 Pin Function Selection Register	PE04PFS	32	8, 16, 32
87C0 0B94h	PFS	PE05 Pin Function Selection Register	PE05PFS	32	8, 16, 32

Table 5.1 List of I/O Registers (Address Order) (11/11)

Secure Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size*1
87C0 0B98h	PFS	PE06 Pin Function Selection Register	PE06PFS	32	8, 16, 32
87C0 0B9Ch	PFS	PE07 Pin Function Selection Register	PE07PFS	32	8, 16, 32
87C0 0BC0h	PFS	PF00 Pin Function Selection Register	PF00PFS	32	8, 16, 32
87C0 0BC4h	PFS	PF01 Pin Function Selection Register	PF01PFS	32	8, 16, 32
87C0 0BC8h	PFS	PF02 Pin Function Selection Register	PF02PFS	32	8, 16, 32
87C0 0BCCh	PFS	PF03 Pin Function Selection Register	PF03PFS	32	8, 16, 32
87C0 0BD0h	PFS	PF04 Pin Function Selection Register	PF04PFS	32	8, 16, 32
87C0 0BD4h	PFS	PF05 Pin Function Selection Register	PF05PFS	32	8, 16, 32
87C0 0BD8h	PFS	PF06 Pin Function Selection Register	PF06PFS	32	8, 16, 32
87C0 0BDCh	PFS	PF07 Pin Function Selection Register	PF07PFS	32	8, 16, 32
87C0 0BE0h	PFS	PF08 Pin Function Selection Register	PF08PFS	32	8, 16, 32
87C0 0BE4h	PFS	PF09 Pin Function Selection Register	PF09PFS	32	8, 16, 32
87C0 0C00h	PFS	PG00 Pin Function Selection Register	PG00PFS	32	8, 16, 32
87C0 0C04h	PFS	PG01 Pin Function Selection Register	PG01PFS	32	8, 16, 32
87C0 0C08h	PFS	PG02 Pin Function Selection Register	PG02PFS	32	8, 16, 32
87C0 0C0Ch	PFS	PG03 Pin Function Selection Register	PG03PFS	32	8, 16, 32
87C0 0C10h	PFS	PG04 Pin Function Selection Register	PG04PFS	32	8, 16, 32
87C0 0C14h	PFS	PG05 Pin Function Selection Register	PG05PFS	32	8, 16, 32
87C0 0C18h	PFS	PG06 Pin Function Selection Register	PG06PFS	32	8, 16, 32
87C0 0C1Ch	PFS	PG07 Pin Function Selection Register	PG07PFS	32	8, 16, 32
87C0 0C20h	PFS	PG08 Pin Function Selection Register	PG08PFS	32	8, 16, 32
87C0 0C24h	PFS	PG09 Pin Function Selection Register	PG09PFS	32	8, 16, 32
87C0 0C28h	PFS	PG10 Pin Function Selection Register	PG10PFS	32	8, 16, 32
87C0 0C2Ch	PFS	PG11 Pin Function Selection Register	PG11PFS	32	8, 16, 32
87C0 0C30h	PFS	PG12 Pin Function Selection Register	PG12PFS	32	8, 16, 32
87C0 0C34h	PFS	PG13 Pin Function Selection Register	PG13PFS	32	8, 16, 32
87C0 0C38h	PFS	PG14 Pin Function Selection Register	PG14PFS	32	8, 16, 32
87C0 0C3Ch	PFS	PG15 Pin Function Selection Register	PG15PFS	32	8, 16, 32
87C0 0D30h	PFS	Port Security Attribute Setting Register	P0SAR	16	8, 16
87C0 0D34h	PFS	Port Security Attribute Setting Register	P1SAR	16	8, 16
87C0 0D38h	PFS	Port Security Attribute Setting Register	P2SAR	16	8, 16
87C0 0D3Ch	PFS	Port Security Attribute Setting Register	P3SAR	16	8, 16
87C0 0D40h	PFS	Port Security Attribute Setting Register	P4SAR	16	8, 16
87C0 0D44h	PFS	Port Security Attribute Setting Register	P5SAR	16	8, 16
87C0 0D48h	PFS	Port Security Attribute Setting Register	P6SAR	16	8, 16
87C0 0D4Ch	PFS	Port Security Attribute Setting Register	P7SAR	16	8, 16
87C0 0D50h	PFS	Port Security Attribute Setting Register	P8SAR	16	8, 16
87C0 0D54h	PFS	Port Security Attribute Setting Register	P9SAR	16	8, 16
87C0 0D58h	PFS	Port Security Attribute Setting Register	PASAR	16	8, 16
87C0 0D5Ch	PFS	Port Security Attribute Setting Register	PBSAR	16	8, 16
87C0 0D60h	PFS	Port Security Attribute Setting Register	PCSAR	16	8, 16
87C0 0D64h	PFS	Port Security Attribute Setting Register	PDSAR	16	8, 16
87C0 0D68h	PFS	Port Security Attribute Setting Register	PESAR	16	8, 16
87C0 0D6Ch	PFS	Port Security Attribute Setting Register	PFSAR	16	8, 16
87C0 0D70h	PFS	Port Security Attribute Setting Register	PGSAR	16	8, 16

Note 1. The table above only lists the addresses of registers with the Secure attribute and multiple access sizes. Note that registers with the Non-secure attribute have different addresses from those listed in the table, but the same access sizes.

6. Resets

6.1 Overview

This MCU supports the following 16 types of reset.
It also has functionality for controlling (enabling or disabling) reset generation by using registers which are only accessible by a secure program.
Table 6.1 lists the names and sources of the resets.

Table 6.1 Names and Sources of Resets

Reset Name	Source
RES# pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	VCC rises (voltage used in monitoring: VPOR)*1
Voltage monitoring 0 reset	VCC falls (voltage used in monitoring: Vdet0)*1
Voltage monitoring 1 reset	VCC falls (voltage used in monitoring: Vdet1)*1 VCC rises (voltage used in monitoring: Vdet1)*1
Voltage monitoring 2 reset	VCC falls (voltage used in monitoring: Vdet2)*1 VCC rises (voltage used in monitoring: Vdet2)*1
Voltage monitoring 4 reset	VCC falls (voltage used in monitoring: Vdet4)*1 VCC rises (voltage used in monitoring: Vdet4)*1
Voltage monitoring 5 reset	VCC falls (voltage used in monitoring: Vdet5)*1 VCC rises (voltage used in monitoring: Vdet5)*1
Core voltage monitoring reset	VDD falls (voltage used in monitoring: Vdet_VDDL)*1 VDD rises (voltage used in monitoring: Vdet_VDDH)*1
Temperature monitoring reset	Detection of a temperature beyond the specified range
Independent watchdog timer reset	Independent watchdog timer underflow or refresh error
Watchdog timer reset	Watchdog timer underflow or refresh error
Bus error reset	Bus error
Common memory error reset	ECC error in the RAM
Local memory error reset	ECC error in the CPU cache or local RAM
Deep software standby reset	Release from deep software standby mode by an interrupt
Software reset	Register setting

Note 1. For details on the voltages for use in monitoring (VPOR, Vdet0, Vdet1, Vdet2, Vdet4, Vdet5, Vdet_VDDL, and Vdet_VDDH), see section 8, Programmable Voltage Detection Circuit (PVD) and section 67, Electrical Characteristics.

When the OFS1.RODIS bit is set to 0, an internal reset signal can be output on the RES# pin. When this output occurs, the MCU is also initialized by an RES# pin reset. This function is called the reset output function. For details, see section 6.3.15, Reset Output Function.
Table 6.2 lists the reset sources for which reset output is possible.

Table 6.2 Reset Sources for which Reset Output is Possible

Reset Source	Reset Output
RES# pin reset	Not possible
Power-on reset	Not possible
Voltage monitoring 0 reset	Possible
Voltage monitoring 1 reset	Possible
Voltage monitoring 2 reset	Possible
Voltage monitoring 4 reset	Possible
Voltage monitoring 5 reset	Possible
Core voltage monitoring reset	— *1
Temperature monitoring reset	Possible
Independent watchdog timer reset	Possible
Watchdog timer reset	Possible
Bus error reset	Possible
Common memory error reset	Possible
Local memory error reset	Possible
Deep software standby reset	Not possible
Software reset	Possible

Note 1. When a core voltage monitoring reset occurs, it is propagated within the MCU so the MCU will be initialized, but the reset output is not guaranteed. The reset output not having proceeded can be identified by the RSTSR3.CVMRF flag being set, while the reset output having proceeded can be identified by the ROUTSR2.ROCVMRF flag being set.

The internal state is initialized by a reset, which also sets the pins to their initial states. Table 6.3, Table 6.4, Table 6.5, and Table 6.6 list the targets for initialization according to the type of reset.

Table 6.3 Targets for Initialization According to Type of Reset (Reset Detect Flags 1) (1/2)

Target for Initialization	Reset Source								
	RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Core Voltage Monitoring Reset	Voltage Monitoring 4 Reset	Voltage Monitoring 5 Reset	Temperature Monitoring Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset
Power-on reset detect flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.PVD0RF)	✓	✓	—	—	—	—	—	—	—
Core voltage monitoring reset detect flag (RSTSR3.CVMRF)	✓	✓	✓	—	—	—	—	—	—
Temperature monitoring reset detect flag (RSTSR3.TEMPRF)	✓	✓	✓	—	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR1.IWDTRF)	✓	✓	✓	✓	✓	✓	✓	—	—
Watchdog timer reset detect flag (RSTSR1.WDT0RF)	✓	✓	✓	✓	✓	✓	✓	—	—
Voltage monitoring 1 reset detect flag (RSTSR0.PVD1RF)	✓	✓	✓	✓	✓	✓	✓	—	—
Voltage monitoring 2 reset detect flag (RSTSR0.PVD2RF)	✓	✓	✓	✓	✓	✓	✓	—	—
Voltage monitoring 4 reset detect flag (RSTSR0.PVD4RF)	✓	✓	✓	—	—	—	—	—	—
Voltage monitoring 5 reset detect flag (RSTSR0.PVD5RF)	✓	✓	✓	—	—	—	—	—	—
Software reset detect flag (RSTSR1.SWRF)	✓	✓	✓	✓	✓	✓	✓	—	—
Bus error reset detect flag (RSTSR1.BUSRF)	✓	✓	✓	✓	✓	✓	✓	—	—
Common memory error reset detect flag (RSTSR1.CMRF)	✓	✓	✓	✓	✓	✓	✓	—	—
Local memory error reset detect flag (RSTSR1.LM0RF)	✓	✓	✓	✓	✓	✓	✓	—	—
Deep software standby reset detect flag (RSTSR0.DPSRSTF)	✓	✓	✓	✓	✓	✓	✓	—	—
Cold start/warm start determination flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (ROUTSR0.ROPVD0RF)	—	✓	—	—	—	—	—	—	—
Core voltage monitoring reset detect flag (ROUTSR2.ROCVMRF)	—	✓	✓	—	—	—	—	—	—
Temperature monitoring reset detect flag (ROUTSR2.ROTEMPRF)	—	✓	✓	—	—	—	—	—	—
Independent watchdog timer reset detect flag (ROUTSR1.ROIWDTRF)	—	✓	✓	✓	✓	✓	✓	—	—
Watchdog timer reset detect flag (ROUTSR1.ROWDT0RF)	—	✓	✓	✓	✓	✓	✓	—	—

Table 6.3 Targets for Initialization According to Type of Reset (Reset Detect Flags 1) (2/2)

Target for Initialization	Reset Source								
	RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Core Voltage Monitoring Reset	Voltage Monitoring 4 Reset	Voltage Monitoring 5 Reset	Temperature Monitoring Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset
Voltage monitoring 1 reset detect flag (ROUTSR0.ROPVD1RF)	—	✓	✓	✓	✓	✓	✓	—	—
Voltage monitoring 2 reset detect flag (ROUTSR0.ROPVD2RF)	—	✓	✓	✓	✓	✓	✓	—	—
Voltage monitoring 4 reset detect flag (ROUTSR0.ROPVD4RF)	—	✓	✓	—	—	—	—	—	—
Voltage monitoring 5 reset detect flag (ROUTSR0.ROPVD5RF)	—	✓	✓	—	—	—	—	—	—
Software reset detect flag (ROUTSR1.ROSWRF)	—	✓	✓	✓	✓	✓	✓	—	—
Bus error reset detect flag (ROUTSR1.ROBUSRF)	—	✓	✓	✓	✓	✓	✓	—	—
Common memory error reset detect flag (ROUTSR1.ROCMRF)	—	✓	✓	✓	✓	✓	✓	—	—
Local memory error reset detect flag (ROUTSR1.ROLM0RF)	—	✓	✓	✓	✓	✓	✓	—	—

✓: Initialized to 0, —: Not changed

Table 6.4 Targets for Initialization According to Type of Reset (Reset Detect Flags 2) (1/2)

Target for Initialization	Reset Source								
							Deep Software Standby Reset		
	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset	Bus Error Reset	Common Memory Error Reset	Local Memory Error Reset	Deep Software Standby Mode 1	Deep Software Standby Mode 2	Deep Software Standby Mode 3
Power-on reset detect flag (RSTSR0.PORF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.PVD0RF)	—	—	—	—	—	—	—	—	—
Core voltage monitoring reset detect flag (RSTSR3.CVMRF)	—	—	—	—	—	—	—	—	—
Temperature monitoring reset detect flag (RSTSR3.TEMPRF)	—	—	—	—	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR1.IWDTRF)	—	—	—	—	—	—	—	✓	✓
Watchdog timer reset detect flag (RSTSR1.WDT0RF)	—	—	—	—	—	—	—	✓	✓
Voltage monitoring 1 reset detect flag (RSTSR0.PVD1RF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 2 reset detect flag (RSTSR0.PVD2RF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 4 reset detect flag (RSTSR0.PVD4RF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 5 reset detect flag (RSTSR0.PVD5RF)	—	—	—	—	—	—	—	—	—
Software reset detect flag (RSTSR1.SWRF)	—	—	—	—	—	—	—	✓	✓
Bus error reset detect flag (RSTSR1.BUSRF)	—	—	—	—	—	—	—	✓	✓
Common memory error reset detect flag (RSTSR1.CMRF)	—	—	—	—	—	—	—	✓	✓
Local memory error reset detect flag (RSTSR1.LM0RF)	—	—	—	—	—	—	—	✓	✓
Deep software standby reset detect flag (RSTSR0.DPSRSTF)	—	—	—	—	—	—	—	—	—
Cold start/warm start determination flag (RSTSR2.CWSF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (ROUTSR0.ROPVD0RF)	—	—	—	—	—	—	—	—	—
Core voltage monitoring reset detect flag (ROUTSR2.ROCVMRF)	—	—	—	—	—	—	—	—	—
Temperature monitoring reset detect flag (ROUTSR2.ROTEMPRF)	—	—	—	—	—	—	—	—	—
Independent watchdog timer reset detect flag (ROUTSR1.ROIWDTRF)	—	—	—	—	—	—	—	✓	✓
Watchdog timer reset detect flag (ROUTSR1.ROWDT0RF)	—	—	—	—	—	—	—	✓	✓
Voltage monitoring 1 reset detect flag (ROUTSR0.ROPVD1RF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 2 reset detect flag (ROUTSR0.ROPVD2RF)	—	—	—	—	—	—	—	—	—
Voltage monitoring 4 reset detect flag (ROUTSR0.ROPVD4RF)	—	—	—	—	—	—	—	—	—

Table 6.4 Targets for Initialization According to Type of Reset (Reset Detect Flags 2) (2/2)

Target for Initialization	Reset Source								
							Deep Software Standby Reset		
	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset	Bus Error Reset	Common Memory Error Reset	Local Memory Error Reset	Deep Software Standby Mode 1	Deep Software Standby Mode 2	Deep Software Standby Mode 3
Voltage monitoring 5 reset detect flag (ROUTSR0.ROPVD5RF)	—	—	—	—	—	—	—	—	—
Software reset detect flag (ROUTSR1.ROSWRF)	—	—	—	—	—	—	—	✓	✓
Bus error reset detect flag (ROUTSR1.ROBUSRF)	—	—	—	—	—	—	—	✓	✓
Common memory error reset detect flag (ROUTSR1.ROCMRF)	—	—	—	—	—	—	—	✓	✓
Local memory error reset detect flag (ROUTSR1.ROLM0RF)	—	—	—	—	—	—	—	✓	✓

✓: Initialized to 0, —: Not changed

Table 6.5 Targets for Initialization According to Type of Reset (Registers Related to Each Module 1) (1/2)

Target for Initialization		Reset Source								
		RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Core Voltage Monitoring Reset	Voltage Monitoring 4 Reset	Voltage Monitoring 5 Reset	Temperature Monitoring Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset
Voltage monitoring function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PVD1CR1, PVD1SR	✓	✓	✓	✓	✓	✓	✓	✓	✓
Voltage monitoring function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PVD2CR1, PVD2SR	✓	✓	✓	✓	✓	✓	✓	✓	✓
Voltage monitoring function 4 registers	PVD4CR0, PVD4CMPCR, PVD4FCR	✓	✓	✓	—	—	—	—	—	—
Voltage monitoring function 5 registers	PVD5CR0, PVD5CMPCR, PVD5FCR	✓	✓	✓	—	—	—	—	—	—
Temperature monitoring reset related registers	TEMPRCR, TEMPRLR	✓	✓	✓	—	—	—	—	—	—
Sub-clock oscillator related registers*2	SOSCCR, SOSTDCR, SOSTDSR, SOMCR	—	—	—	—	—	—	—	—	—
Low-speed on-chip oscillator related register	LOCOUTCR	—	✓	✓	✓	✓	✓	✓	—	—
Main clock oscillator related register	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓	✓
High-speed on-chip oscillator related register	HOCOCR2	—	✓	✓	✓	✓	✓	✓	—	—
Pin states (except for XCIN/XCOUT)		✓	✓	✓	✓	✓	✓	✓	✓	✓
Pin states (XCIN/XCOUT)*2		—	—	—	—	—	—	—	—	—
IO capture and tamper detection (VBATT) (sampling timing for RTC) (RTCIC0 -2)*2	TAMPICR1, TAMPICR2, TAMPSR, TAMPCR, TCECR, TAMPCR2, TAMPICR3	—	—	—	—	—	—	—	—	—
Battery backup function control Register 1	VBATTCR1	✓	✓	✓	✓	✓	✓	✓	✓	✓
Battery backup function control register 2*2	VBATTCR2	—	—	—	—	—	—	—	—	—
Backup domain power status register*2	BKPSR	—	—	—	—	—	—	—	—	—

Table 6.5 Targets for Initialization According to Type of Reset (Registers Related to Each Module 1) (2/2)

Target for Initialization		Reset Source								
		RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Core Voltage Monitoring Reset	Voltage Monitoring 4 Reset	Voltage Monitoring 5 Reset	Temperature Monitoring Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset
Backup register control register	BKRCR	—	✓	—	—	—	—	—	—	—
Backup register*2	BKRn	—	—	—	—	—	—	—	—	—
Independent watchdog timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR	✓	✓	✓	✓	✓	✓	✓	✓	✓
Realtime clock*1 register		—	—	—	—	—	—	—	—	—
USB registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓	✓	✓
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset flags	BUSERRSTATCPU0IF, BUSERRSTATCPU0OA, BUSERRSTATDMAC0, BUSERRSTATDMACX, BUSERRSTATEDMAC, BUSERRSTATXSPI0, BUSERRSTATXSPI1, MBWERRSTAT, SRAMESR, SRAMEARnm (n = 0, 1, m = 0, 1) LRMESR, LRMEARxm (x = 0, 1, m = 1, 2), IFCAEDST, OACAEDST	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset flags		See Table 6.3 and Table 6.4								
Low-power-consumption function related registers	DPSBYCR, DPSIER0 to DPSIER5, DPSIFR0 to DPSIFR5, DPSIEGR0 to DPSIEGR4 LPSCR, SSCR1, SVSCR, LVOCR	✓	✓	✓	✓	✓	✓	✓	✓	✓
Low-power-consumption function related register	SYOCDCR	—	✓	✓	✓	✓	✓	✓	—	—
System reset mask control registers	SYRSTMSK0, SYRSTMSK2	✓	✓	✓	✓	✓	✓	✓	—	—
Power shut-off control registers	PDRAMSCR0, PDRAMSCR2	—	✓	✓	✓	✓	✓	✓	—	—
Other than the above		✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Initialized to 0, —: Not changed

Note 1. Some control bits are initialized by any reset. For details on the target control bits, see section 30, Realtime Clock (RTCBb).

Note 2. Only initialized by a backup area reset. For details, see section 12.3.2, Power-Down Detector.

Table 6.6 Targets for Initialization According to Type of Reset (Registers Related to Each Module 2) (1/2)

Target for Initialization		Reset Source								
								Deep Software Standby Reset		
		Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset	Bus Error Reset	Common Memory Error Reset	Local Memory Error Reset	Deep Software Standby Mode 1	Deep Software Standby Mode 2	Deep Software Standby Mode 3
Voltage monitoring function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	—	—	—	—	—	—	—	—	—
	PVD1CR1, PVD1SR	—	—	—	—	—	—	✓	✓	✓
Voltage monitoring function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	—	—	—	—	—	—	—	—	—
	PVD2CR1, PVD2SR	—	—	—	—	—	—	✓	✓	✓
Voltage monitoring function 4 registers	PVD4CR0, PVD4CMPCR, PVD4FCR	—	—	—	—	—	—	—	—	—
Voltage monitoring function 5 registers	PVD5CR0, PVD5CMPCR, PVD5FCR	—	—	—	—	—	—	—	—	—
Temperature monitoring reset related registers	TEMPRCR, TEMPRLR	—	—	—	—	—	—	—	—	—
Sub-clock oscillator related registers*3	SOSCCR, SOSTDCR, SOSTDSR, SOMCR	—	—	—	—	—	—	—	—	—
Low-speed on-chip oscillator related register	LOCOUTCR	—	—	—	—	—	—	—	✓	✓
Main clock oscillator related register	MOMCR	✓	✓	✓	✓	✓	✓	—	—	—
High-speed on-chip oscillator related register	HOCOCR2	—	—	—	—	—	—	✓	✓	✓
Pin states (except for XCIN/XCOUT)		✓	✓	✓	✓	✓	✓	*2	*2	*2
Pin states (XCIN/XCOUT)*3		—	—	—	—	—	—	—	—	—
IO capture and tamper detection (VBATT) (sampling timing for RTC) (RTCIC0 -2)*3	TAMPICR1, TAMPICR2, TAMPSR, TAMPCR, TCECR, TAMPCR2, TAMPICR3	—	—	—	—	—	—	—	—	—
Battery backup function control Register 1	VBATTCR1	✓	✓	✓	✓	✓	✓	—	—	—
Battery backup function control register 2*3	VBATTCR2	—	—	—	—	—	—	—	—	—
Backup domain power status register*3	BKPSR	—	—	—	—	—	—	—	—	—
Backup register control register	BKRCR	—	—	—	—	—	—	—	—	—
Backup register*3	BKRn	—	—	—	—	—	—	—	—	—

Table 6.6 Targets for Initialization According to Type of Reset (Registers Related to Each Module 2) (2/2)

Target for Initialization		Reset Source								
								Deep Software Standby Reset		
		Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset	Bus Error Reset	Common Memory Error Reset	Local Memory Error Reset	Deep Software Standby Mode 1	Deep Software Standby Mode 2	Deep Software Standby Mode 3
Independent watchdog timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRC, IWDTCSTPR	✓	✓	✓	✓	✓	✓	—	✓	✓
Realtime clock*1 register		—	—	—	—	—	—	—	—	—
USB registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	—	✓	✓
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	—	✓	✓
Reset flags	BUSERRSTATCPU0IF, BUSERRSTATCPU0OA, BUSERRSTATDMAC0, BUSERRSTATDMACX, BUSERRSTATEDMAC, BUSERRSTATXSPI0, BUSERRSTATXSPI1, MBWERRSTAT, SRAMESR, SRAMEARnm (n = 0, 1, m = 0, 1) LRMESR, LRMEARxm (x = 0, 1, m = 1, 2), IFCAEDST, OACAEDST	✓	✓	✓	—	—	—	✓	✓	✓
Reset flags		See Table 6.3 and Table 6.4								
Low-power-consumption function related registers	DPSBYCR, DPSIER0 to DPSIER5, DPSIFR0 to DPSIFR5, DPSIEGR0 to DPSIEGR4 LPSCR, SSCR1, SVSCR, LVOCR	✓	✓	✓	✓	✓	✓	—	—	—
Low-power-consumption function related register	SYOCDCR	—	—	—	—	—	—	—	—	—
System reset mask control registers	SYRSTMSK0, SYRSTMSK2	—	—	—	—	—	—	—	—	—
Power shut-off control registers	PDRAMSCR0, PDRAMSCR2	—	—	—	—	—	—	✓	✓	✓
Other than the above		✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Initialized to 0, —: Not changed

Note 1. Some control bits are initialized by any reset. For details on the target control bits, see section 30, Realtime Clock (RTCBb).
Note 2. Depends on the setting of DPSBYCR.IOKEEP.
Note 3. Only initialized by a backup area reset. For details, see section 12.3.2, Power-Down Detector.

The sub-clock oscillator and LOCO are selectable as the clock source for the RTC. Table 6.7 and Table 6.8 list the states of the sub-clock oscillator and LOCO when a reset occurs.

Table 6.7 States of Sub-Clock Oscillator when Reset Occurs

		Backup Area Reset	All Resets
Sub-clock oscillator	Enabled or disabled	Initialized to “disabled”	Maintains the state that was selected before a reset
	Driving ability	Initialized to the standard driving ability	

Table 6.8 States of LOCO when Reset Occurs

		Reset Source	
		Power-On Reset, Voltage Monitoring 0/4/5 Reset, Core Voltage Monitoring Reset, Temperature Monitoring Reset, Deep Software Standby 2/3 Reset	Others
LOCO	Enabled or disabled	Initialized to “enabled”	
	Oscillation accuracy*1	Initialized (to the accuracy of ±15% before adjustment by the LOCOUTCR register)	Maintains the accuracy adjusted by the LOCOUTCR register

Note 1. Note that if the LOCO adjusted by the LOCOUTCR register is selected as the RTC clock source, the LOCO oscillation accuracy is initialized by a power-on reset, voltage monitoring 0/4/5 reset, core voltage monitoring reset, temperature monitoring reset, or deep software standby 2/3 reset.

On release from the reset state, reset exception handling will start. For the reset exception handling, see section 14, Exception Handling.
Table 6.9 lists the I/O pins related to the reset.

Table 6.9 I/O Pins Related to Reset

Pin Name	I/O	Function
RES# / RSTOUT#	I/O	Reset pin

6.2 Register Descriptions

6.2.1 Reset Security Attribute Register (RSTSAR)

Address(es): SYSC.RSTSAR 8701 93C4h, SYSC_NS.RSTSAR 0701 93C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	NONSEC5	—	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NONSEC0	Non-Secure Attribute 0	Target registers: RSTSR0, ROUTSR0 0: Secure 1: Non-secure	R/W
b1	NONSEC1	Non-Secure Attribute 1	Target registers: RSTSR1, ROUTSR1 0: Secure 1: Non-secure	R/W
b2	NONSEC2	Non-Secure Attribute 2	Target register: RSTSR2 0: Secure 1: Non-secure	R/W
b3	NONSEC3	Non-Secure Attribute 3	Target registers: RSTSR3, ROUTSR2 0: Secure 1: Non-secure	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	NONSEC5	Non-Secure Attribute 5	Target register: SWRR 0: Secure 1: Non-secure	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is only writable from the secure world. An attempt at writing to it from the non-secure world will be ignored, but does not cause a protect zone access error. The register can be read from either world.

NONSEC0 Bit (Non-Secure Attribute 0)

This bit controls the security attribute of the RSTSR0 and ROUTSR0 registers.

NONSEC1 Bit (Non-Secure Attribute 1)

This bit controls the security attribute of the RSTSR1 and ROUTSR1 registers.

NONSEC2 Bit (Non-Secure Attribute 2)

This bit controls the security attribute of the RSTSR2 register.

NONSEC3 Bit (Non-Secure Attribute 3)

This bit controls the security attribute of the RSTSR3 and ROUTSR2 registers.

NONSEC5 Bit (Non-Secure Attribute 5)

This bit controls the security attribute of the SWRR register.

6.2.2 Reset Status Register 0 (RSTSR0)

Address(es): SYSC.RSTSR0 8701 9A40h, SYSC_NS.RSTSR0 0701 9A40h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSRSTF	PVD5RF	PVD4RF	—	PVD2RF	PVD1RF	PVD0RF	PORF
Value after reset:	x*1	x*1	x*1	0	x*1	x*1	x*1	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected 1: Power-on reset detected	R/W*2
b1	PVD0RF	Voltage Monitoring 0 Reset Detect Flag	0: Voltage monitoring 0 reset not detected 1: Voltage monitoring 0 reset detected	R/W*2
b2	PVD1RF	Voltage Monitoring 1 Reset Detect Flag	0: Voltage monitoring 1 reset not detected 1: Voltage monitoring 1 reset detected	R/W*2
b3	PVD2RF	Voltage Monitoring 2 Reset Detect Flag	0: Voltage monitoring 2 reset not detected 1: Voltage monitoring 2 reset detected	R/W*2
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	PVD4RF	Voltage Monitoring 4 Reset Detect Flag	0: Voltage monitoring 4 reset not detected 1: Voltage monitoring 4 reset detected	R/W*2
b6	PVD5RF	Voltage Monitoring 5 Reset Detect Flag	0: Voltage monitoring 5 reset not detected 1: Voltage monitoring 5 reset detected	R/W*2
b7	DPSRSTF	Deep Software Standby Reset Detect Flag	0: Request for release from deep software standby mode by an interrupt or reset*3 not generated 1: Request for release from deep software standby mode by an interrupt or reset*3 generated	R/W*2

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.
Note 3. Independent watchdog timer reset, voltage monitoring 1 reset, or voltage monitoring 2 reset

If the reset output function is enabled by OFS, read b1 to b6 of the ROUTSR0 register instead of b1 to b6 of the RSTSR0 register to identify the source of the reset.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.

[Setting condition]

- When a power-on reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PORF flag after it was read as 1

PVD0RF Flag (Voltage Monitoring 0 Reset Detect Flag)

The PVD0RF flag indicates that a voltage monitoring 0 reset has occurred.

[Setting condition]

- When a voltage monitoring 0 reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PVD0RF flag after it was read as 1

PVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The PVD1RF flag indicates that a voltage monitoring 1 reset has occurred.

[Setting condition]

- When a voltage monitoring 1 reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PVD1RF flag after it was read as 1

PVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The PVD2RF flag indicates that a voltage monitoring 2 reset has occurred.

[Setting condition]

- When a voltage monitoring 2 reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PVD2RF flag after it was read as 1

PVD4RF Flag (Voltage Monitoring 4 Reset Detect Flag)

The PVD4RF flag indicates that a voltage monitoring 4 reset has occurred.

[Setting condition]

- When a voltage monitoring 4 reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PVD4RF flag after it was read as 1

PVD5RF Flag (Voltage Monitoring 5 Reset Detect Flag)

The PVD5RF flag indicates that a voltage monitoring 5 reset has occurred.

[Setting condition]

- When a voltage monitoring 5 reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the PVD5RF flag after it was read as 1

DPSRSTF Flag (Deep Software Standby Reset Detect Flag)

The DPSRSTF flag indicates that deep software standby mode has been released by an external or internal interrupt, and that an internal reset (deep software standby reset) has occurred at the time of exception generation from deep software standby mode.

[Setting condition]

- When deep software standby mode is released by an external or an internal interrupt. For details, see section 11, Low Power Consumption.

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the DPSRSTF flag after it was read as 1

6.2.3 Reset Status Register 1 (RSTSR1)

Address(es): SYSC.RSTSR1 8701 90C0h, SYSC_NS.RSTSR1 0701 90C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMRF	—	—	—	BUSRF	—	—	—	—	LM0RF	—	—	SWRF	WDT0RF	IWDTRF
Value after reset:	0	x*1	0	0	0	x*1	0	0	0	0	x*1	0	0	x*1	x*1	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
b1	WDT0RF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W*2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected 1: Software reset detected	R/W*2
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LM0RF	Local Memory Error Reset Detect Flag	0: Local memory error reset not detected 1: Local memory error reset detected	R/W*2
b9 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	BUSRF	Bus Error Reset Detect Flag	0: Bus error reset not detected 1: Bus error reset detected	R/W*2
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CMRF	Common Memory Error Reset Detect Flag	0: Common memory error reset not detected 1: Common memory error reset detected	R/W*2
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.

If the reset output function is enabled by OFS, read the ROUTSR1 register instead of the RSTSR1 register to identify the source of the reset.

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.
[Setting condition]

- When an independent watchdog timer reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the IWDTRF flag after it was read as 1

WDT0RF Flag (Watchdog Timer Reset Detect Flag)

The WDT0RF flag indicates that a watchdog timer reset has occurred.
[Setting condition]

- When a watchdog timer reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the WDT0RF flag after it was read as 1

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.
[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the SWRF flag after it was read as 1

LM0RF Flag (Local Memory Error Reset Detect Flag)

The LM0RF flag indicates that a local memory error reset (ECC error in the CPU cache or local RAM) has occurred.
[Setting condition]

- When a local memory error reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the LM0RF flag after it was read as 1

BUSRF Flag (Bus Error Reset Detect Flag)

The BUSRF flag indicates that a bus error reset (master security attribute unit error, master MPU error, illegal address error, protect zone filter error, slave bus error, or bufferable write error) has occurred.
[Setting condition]

- When a bus error reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the BUSRF flag after it was read as 1

CMRF Flag (Common Memory Error Reset Detect Flag)

The CMRF flag indicates that a common memory error reset (ECC error in the SRAM) has occurred.
[Setting condition]

- When a common memory error reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the CMRF flag after it was read as 1

6.2.4 Reset Status Register 2 (RSTSR2)

Address(es): SYSC.RSTSR2 8701 9A44h, SYSC_NS.RSTSR2 0701 9A44h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	R/W*2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 1 can be written to clear the flag.

The RSTSR2 register determines whether a power-on reset has caused the reset processing (cold start) or the reset signal being input during operation has caused the reset processing (warm start).

CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing (cold start or warm start). The CWSF flag is initialized by a power-on reset. It is not initialized by a RES# pin reset.

[Setting condition]

- When 1 is written by software. Writing 0 to the CWSF flag does not clear it to 0.

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs

6.2.5 Reset Status Register 3 (RSTSR3)

Address(es): SYSC.RSTSR3 8701 9A48h, SYSC_NS.RSTSR3 0701 9A48h

	b7	b6	b5	b4	b3	b2	b1	b0
	TEMPRF	—	—	—	—	—	—	CVMRF
Value after reset:	x*1	0	0	0	0	0	0	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	CVMRF	Core Voltage Monitoring Reset Detect Flag	0: Core voltage monitoring reset not detected 1: Core voltage monitoring reset detected	R/W*2
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TEMPRF	Temperature Monitoring Reset Detect Flag	0: Temperature monitoring reset not detected 1: Temperature monitoring reset detected	R/W*2

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.

If the reset output function is enabled by OFS, read the ROUTSR2 register instead of the RSTSR3 register to identify the source of the reset.

CVMRF Flag (Core Voltage Monitoring Reset Detect Flag)

The CVMRF flag indicates that a core voltage monitoring reset has occurred.
[Setting condition]

- When a core voltage monitoring reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the CVMRF flag after it was read as 1

TEMPRF Flag (Temperature Monitoring Reset Detect Flag)

The TEMPRF flag indicates that a temperature monitoring reset has occurred.
[Setting condition]

- When a temperature monitoring reset occurs

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the TEMPRF flag after it was read as 1

6.2.6 System Reset Mask Control Register 0 (SYRSTMSK0)

Address(es): SYSC.SYRSTMSK0 8701 9AD0h

	b7	b6	b5	b4	b3	b2	b1	b0
	BUSMASK	CMMASK	LM0MASK	—	—	SWMASK	WDT0MASK	IWDTMASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTMASK	Independent Watchdog Timer Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b1	WDT0MASK	Watchdog Timer Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b2	SWMASK	Software Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LM0MASK	Local Memory Error Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b6	CMMASK	Common Memory Error Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b7	BUSMASK	Bus Error Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W

Note: This register is only accessible from the secure world. An attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
Note: Set the PRCR_S.PRC5 bit to 1 (enabling writing) before changing the value of this register.

The SYRSTMSK0 register controls generation of a reset.
The value of the IWDTMASK bit cannot be changed while the independent watchdog timer is operating. The value of the WDT0MASK bit cannot be changed while the watchdog timer is operating.

6.2.7 System Reset Mask Control Register 2 (SYRSTMSK2)

Address(es): SYSC.SYRSTMSK2 8701 9AD8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PVD2MASK	PVD1MASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PVD1MASK	Voltage Monitoring 1 Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b1	PVD2MASK	Voltage Monitoring 2 Reset Mask	0: Reset generation enabled 1: Reset generation disabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is only accessible from the secure world. An attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.

Note: Set the PRCR_S.PRC5 bit to 1 (enabling writing) before changing the value of this register.

The SYRSTMSK2 register controls generation of a reset.

6.2.8 Temperature Monitoring Reset Control Register (TEMPRCR)

Address(es): SYSC.TEMPRCR 8701 9ADCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	TSNKEEP	CMPEN	TSNEN	TEMPREN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TEMPREN	Temperature Monitoring Reset Enable	This bit selects enabling or disabling in combination with the TSNKEEP bit. TSNKEEP TEMPREN 0 0: Temperature monitoring reset disabled 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Temperature monitoring reset enabled	R/W
b1	TSNEN	Temperature Sensor Enable	0: Temperature sensor disabled 1: Temperature sensor enabled	R/W
b2	CMPEN	Comparator Enable	0: Comparator disabled 1: Comparator enabled	R/W
b3	TSNKEEP	Temperature Sensor Setting	This bit selects enabling or disabling in combination with the TEMPREN bit. For details of the function, see the description of the TEMPREN bit.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is only accessible from the secure world. An attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.

Note: Set the PRCR_S.PRC5 bit to 1 (enabling writing) before changing the value of this register.

6.2.9 Temperature Monitoring Reset Lock Register (TEMPRLR)

Address(es): SYSC.TEMPRLR 8701 9AE0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LOCK
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	LOCK	Temperature Monitoring Reset Control Register Lock	0: The TEMPRCR register can be written. 1: The TEMPRCR register cannot be written.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is only accessible from the secure world. An attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
Note: Set the PRCR_S.PRC5 bit to 1 (enabling writing) before changing the value of this register.
Note: This register only accepts write access twice following a reset. If you need to write to the TEMPRLR register three or more times, reset the device before writing to the register again.

6.2.10 Reset Output Status Register 0 (ROUTSR0)

Address(es): SYSC.ROUTSR0 8701 9B78h, SYSC_NS.ROUTSR0 0701 9B78h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ROPVD5RF	ROPVD4RF	—	ROPVD2RF	ROPVD1RF	ROPVD0RF	—
Value after reset:	0	x*1	x*1	0	x*1	x*1	x*1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ROPVD0RF	Voltage Monitoring 0 Reset Detect Flag	0: Voltage monitoring 0 reset not detected 1: Voltage monitoring 0 reset detected	R/W*2
b2	ROPVD1RF	Voltage Monitoring 1 Reset Detect Flag	0: Voltage monitoring 1 reset not detected 1: Voltage monitoring 1 reset detected	R/W*2
b3	ROPVD2RF	Voltage Monitoring 2 Reset Detect Flag	0: Voltage monitoring 2 reset not detected 1: Voltage monitoring 2 reset detected	R/W*2
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ROPVD4RF	Voltage Monitoring 4 Reset Detect Flag	0: Voltage monitoring 4 reset not detected 1: Voltage monitoring 4 reset detected	R/W*2
b6	ROPVD5RF	Voltage Monitoring 5 Reset Detect Flag	0: Voltage monitoring 5 reset not detected 1: Voltage monitoring 5 reset detected	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.

If the reset output function is enabled by OFS, read the RSTSR0 register instead of the ROUTSR0 register to identify the source of the reset.

ROPVD0RF Flag (Voltage Monitoring 0 Reset Detect Flag)

The ROPVD0RF flag indicates that a voltage monitoring 0 reset has occurred.
[Setting condition]

- When a voltage monitoring 0 reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROPVD0RF flag after it was read as 1

ROPVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The ROPVD1RF flag indicates that a voltage monitoring 1 reset has occurred.
[Setting condition]

- When a voltage monitoring 1 reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROPVD1RF flag after it was read as 1

ROPVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The ROPVD2RF flag indicates that a voltage monitoring 2 reset has occurred.
[Setting condition]

- When a voltage monitoring 2 reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROPVD2RF flag after it was read as 1

ROPVD4RF Flag (Voltage Monitoring 4 Reset Detect Flag)

The ROPVD4RF flag indicates that a voltage monitoring 4 reset has occurred.
[Setting condition]

- When a voltage monitoring 4 reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROPVD4RF flag after it was read as 1

ROPVD5RF Flag (Voltage Monitoring 5 Reset Detect Flag)

The ROPVD5RF flag indicates that a voltage monitoring 5 reset has occurred.
[Setting condition]

- When a voltage monitoring 5 reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROPVD5RF flag after it was read as 1

6.2.11 Reset Output Status Register 1 (ROUTSR1)

Address(es): SYSC.ROUTSR1 8701 9B7Ch, SYSC_NS.ROUTSR1 0701 9B7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ROCMRF	—	—	—	ROBUSRF	—	—	—	—	ROLM0RF	—	—	ROSWRF	ROWDT0RF	ROIWDTRF
Value after reset:	0	x*1	0	0	0	x*1	0	0	0	0	x*1	0	0	x*1	x*1	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	ROIWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
b1	ROWDT0RF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W*2
b2	ROSWRF	Software Reset Detect Flag	0: Software reset not detected 1: Software reset detected	R/W*2
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ROLM0RF	Local Memory Error Reset Detect Flag	0: Local memory error reset not detected 1: Local memory error reset detected	R/W*2
b9 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	ROBUSRF	Bus Error Reset Detect Flag	0: Bus error reset not detected 1: Bus error reset detected	R/W*2
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	ROCMRF	Common Memory Error Reset Detect Flag	0: Common memory error reset not detected 1: Common memory error reset detected	R/W*2
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.

Note 1. The value after reset depends on the source of the reset.

Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.

If the reset output function is enabled by OFS, read the RSTSR1 register instead of the ROUTSR1 register to identify the source of the reset.

ROIWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The ROIWDTRF flag indicates that an independent watchdog timer reset has occurred.

[Setting condition]

- When an independent watchdog timer reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROIWDTRF flag after it was read as 1

ROWDT0RF Flag (Watchdog Timer Reset Detect Flag)

The ROWDT0RF flag indicates that a watchdog timer reset has occurred.
[Setting condition]

- When a watchdog timer reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROWDT0RF flag after it was read as 1

ROSWRF Flag (Software Reset Detect Flag)

The ROSWRF flag indicates that a software reset has occurred.
[Setting condition]

- When a software reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROSWRF flag after it was read as 1

ROLM0RF Flag (Local Memory Error Reset Detect Flag)

The ROLM0RF flag indicates that a local memory error reset (ECC error in the CPU cache or local RAM) has occurred.
[Setting condition]

- When a local memory error reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROLM0RF flag after it was read as 1

ROBUSRF Flag (Bus Error Reset Detect Flag)

The ROBUSRF flag indicates that a bus error reset (master security attribute unit error, master MPU error, illegal address error, protect zone filter error, slave bus error, or bufferable write error) has occurred.
[Setting condition]

- When a bus error reset occurs while the reset output function is enabled (OFS1.RODIS = 0).

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROBUSRF flag after it was read as 1

ROCMRF Flag (Common Memory Error Reset Detect Flag)

The ROCMRF flag indicates that a common memory error reset (ECC error in the SRAM) has occurred.
[Setting condition]

- When a common memory error reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROCMRF flag after it was read as 1

6.2.12 Reset Output Status Register 2 (ROUTSR2)

Address(es): SYSC.ROUTSR2 8701 9B80h, SYSC_NS.ROUTSR2 0701 9B80h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROTEMPRF	—	—	—	—	—	—	ROCVMRF
Value after reset:	x*1	0	0	0	0	0	0	x*1

Bit	Symbol	Bit Name	Description	R/W
b0	ROCVMRF	Core Voltage Monitoring Reset Detect Flag	0: Core voltage monitoring reset not detected 1: Core voltage monitoring reset detected	R/W*2
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROTEMPRF	Temperature Monitoring Reset Detect Flag	0: Temperature monitoring reset not detected 1: Temperature monitoring reset detected	R/W*2

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.
Note 1. The value after reset depends on the source of the reset.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 to it after reading 1.

If the reset output function is enabled by OFS, read the RSTSR3 register instead of the ROUTSR2 register to identify the source of the reset.

ROCVMRF Flag (Core Voltage Monitoring Reset Detect Flag)

The ROCVMRF flag indicates that a core voltage monitoring reset has occurred.
[Setting condition]

- When a core voltage monitoring reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROCVMRF flag after it was read as 1

ROTEMPRF Flag (Temperature Monitoring Reset Detect Flag)

The ROTEMPRF flag indicates that a temperature monitoring reset has occurred.
[Setting condition]

- When a temperature monitoring reset occurs while the reset output function is enabled (OFS1.RODIS = 0)

[Clearing conditions]

- When a reset listed in Table 6.3 and Table 6.4 occurs
- When 0 is written to the ROTEMPRF flag after it was read as 1

6.2.13 Software Reset Register (SWRR)

Address(es): SYSC.SWRR 8701 90B0h, SYSC_NS.SWRR 0701 90B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SWRR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SWRR[15:0]	Software Reset	Writing A501h to these bits resets the MCU. When read, they are read as 0000h.	R/W

Note: If this register is allocated to the secure world, it is only accessible from the secure world. In this case, an attempt at access to it from the non-secure world will be ignored, and will cause a protect zone access error.
If this register is allocated to the non-secure world, it is only accessible from the non-secure world. In this case, an attempt at access to it from the secure world will be ignored, and will cause a protect zone access error.

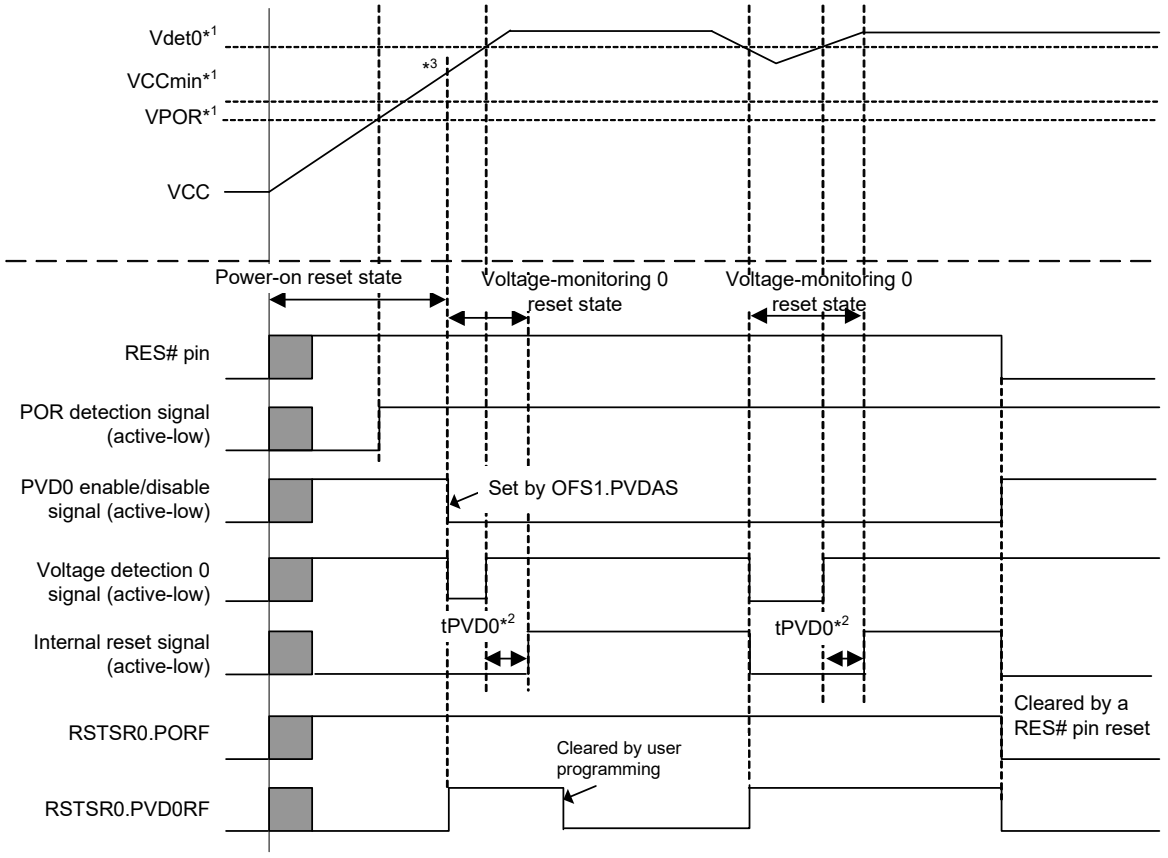
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin. When the RES# pin is driven to the low level, all the processing in progress is aborted and the MCU enters a reset state. In order to unfailingly reset the MCU, the RES# pin should be held low for the specified power supply stabilization time at power-on.
After the RES# pin is driven to the high from the low level, the device is released from the internal reset state after the wait time following RES# de-assertion (tRESWT) has elapsed, and the CPU starts reset exception handling.
For details, see section 67, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

A power-on reset is an internal reset generated by the power-on reset circuit. If the RES# pin is at the high level when power is supplied, a power-on reset is generated.
When VCC exceeds VPOR, the device is released from the internal reset state after the specified period (power-on reset time) has elapsed, and the CPU starts reset exception handling. The power-on reset time is a period for stabilization of power supply and stable operation of the MCU.
When a power-on reset occurs, the RSTSR0.PORF flag is set to 1. The PORF flag is initialized by a RES# pin reset.
Power-on resets are generated by two types of voltage detection circuit, respectively referred to as power-on reset circuits 1 and 2 (POR1 and POR2). The VCC ramp-up rate after power is supplied determines the timing of release from a power-on reset. Specifically, if VCC rises slowly following the supply of power, POR1 determines the timing of release from the power-on reset state. On the other hand, if VCC rises rapidly, POR2 is the determinant. See Figure 67.138, Power-on Reset Timing in section 67, Electrical Characteristics, for details.
A voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 circuit start bit (PVDAS) of option function select register 1 (OFS1) is 0 (enabling the voltage monitoring 0 reset after a reset) and VCC falls to or below Vdet0, the RSTSR0.PVD0RF flag becomes 1 and the voltage detection circuit generates a voltage monitoring 0 reset. When a voltage monitoring 0 reset is to be used, set the OFS1.PVDAS bit to 0.
When VCC exceeds Vdet0, the device is released from the internal reset state after the voltage monitoring 0 reset time (tPVD0) has elapsed, and the CPU starts reset exception handling. The voltage detection level for Vdet0 can be changed by the setting of the VDSEL[2:0] bits in option function select register 1 (OFS1).
Figure 6.1 shows an example of the operation of the power-on reset and voltage monitoring 0 reset.
For details on the voltage monitoring 0 reset, see section 8, Programmable Voltage Detection Circuit (PVD).



Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdet0 is the detection level for the voltage monitoring 0 reset, VPOR is the detection level for the power-on reset, and VCCmin is the minimum guaranteed voltage of the MCU.
Note 2. tPVD0 is the period of the voltage monitoring 0 reset.
Note 3. At power-on, VCC should rise to or above the minimum guaranteed voltage before release from the power-on reset state.

Figure 6.1 Example of Operation at the Time of Power-On Reset and Voltage Monitoring 0 Reset

6.3.3 Voltage Monitoring 1, 2, 4 and 5 Resets

These are internal resets generated by the voltage monitoring circuit.

(1) When PVDiFCR.RHSEL = 0 and PVDjFCR.RHSEL = 0 (i = 1, 2; j = 4, 5)

If VCC falls to or below Vdet1 while the voltage monitoring 1 interrupt/reset enable bit (RIE) of voltage monitoring 1 circuit control register 0 (PVD1CR0) is 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (RI) is 1 (selecting generation of a reset in response to detection of a low voltage), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 1 reset. Likewise, if VCC falls to or below Vdet2 while the voltage monitoring 2 interrupt/reset enable bit (RIE) of voltage monitoring 2 circuit control register 0 (PVD2CR0) is 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (RI) is 1 (selecting generation of a reset in response to detection of a low voltage), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset.

The timing of release from the voltage monitoring 1 reset state is selectable by using the voltage monitoring 1 reset negate select bit (RN) of the PVD1CR0 register. When the PVD1CR0.RN bit is 0, the device is released from the internal reset state following the voltage monitoring 1 reset time (tPVD1) having elapsed after VCC has fallen to or below Vdet1 and then exceeded Vdet1, and the CPU starts reset exception handling.

When the PVD1CR0.RN bit is 1, the device is released from the internal reset state following the voltage monitoring 1 reset time (tPVD1) having elapsed after VCC has fallen to or below Vdet1, and the CPU starts reset exception handling. Likewise, the timing of release from the voltage monitoring 2 reset state is selectable by the setting of the voltage monitoring 2 reset negate select bit (RN) in the PVD2CR0 register.

If VCC falls to or below Vdet4 while the voltage monitoring 4 reset enable bit (RE) of voltage monitoring 4 circuit control register 0 (PVD4CR0) is 1 (enabling generation of a reset by the voltage detection circuit), the RSTSR0.PVD4RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 4 reset.

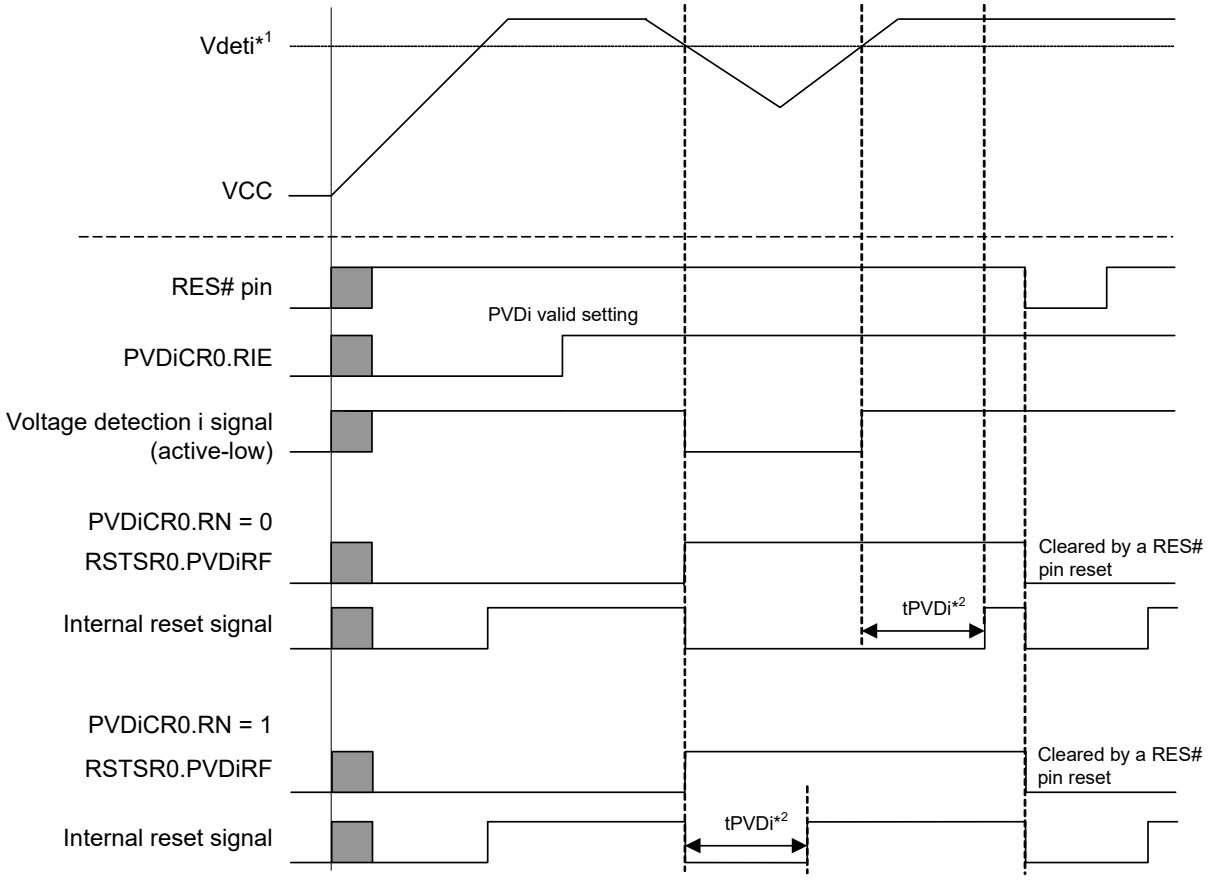
The device is released from the internal reset state following the voltage monitoring 1 reset time (tPVD4) having elapsed after VCC has fallen to or below Vdet4 and then exceeded Vdet4, and the CPU starts reset exception handling.

If VCC falls to or below Vdet5 while the voltage monitoring 5 reset enable bit (RE) of voltage monitoring 5 circuit control register 0 (PVD5CR0) is 1 (enabling generation of a reset by the voltage detection circuit), the RSTSR0.PVD5RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 5 reset.

The device is released from the internal reset state following the voltage monitoring 5 reset time (tPVD5) having elapsed after VCC has fallen to or below Vdet5 and then exceeded Vdet5, and the CPU starts reset exception handling.

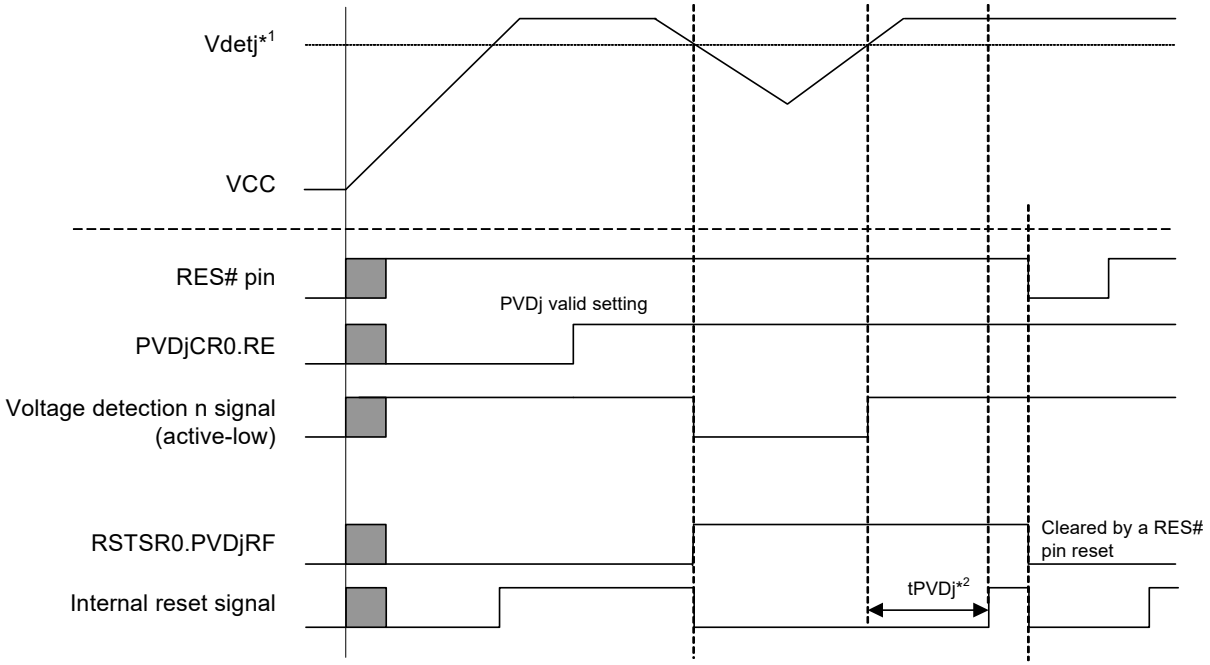
Figure 6.2 shows an example of the operation of the voltage monitoring 1 and 2 resets. Figure 6.3 shows an example of the operation of the voltage monitoring 4 and 5 resets.

For details on the voltage monitoring 1, 2, 4, and 5 resets, see section 8, Programmable Voltage Detection Circuit (PVD).



Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdeti is the detection level for the voltage monitoring 1 or 2 reset. (i = 1, 2)
Note 2. tPVDi is the period of the voltage monitoring 1 or 2 reset. (i = 1, 2)

Figure 6.2 Example of Operation of Voltage Monitoring 1 and 2 Resets (RHSEL = 0)



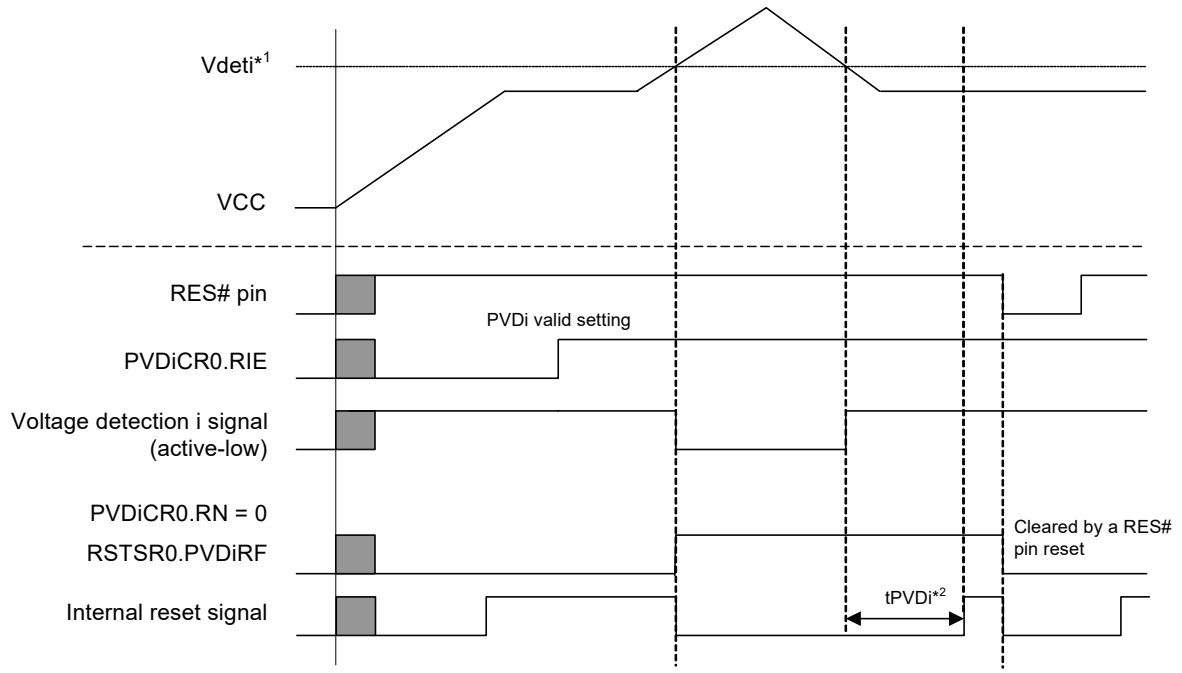
Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdetj is the detection level for the voltage monitoring 4 or 5 reset. (j = 4, 5)
Note 2. tPVDj is the period of the voltage monitoring 4 or 5 reset. (j = 4, 5)

Figure 6.3 Example of Operation of Voltage Monitoring 4 and 5 Resets (RHSEL = 0)

(2) When PVDiFCR.RHSEL = 1 and PVDjFCR.RHSEL = 1 (i = 1, 2; j = 4, 5)

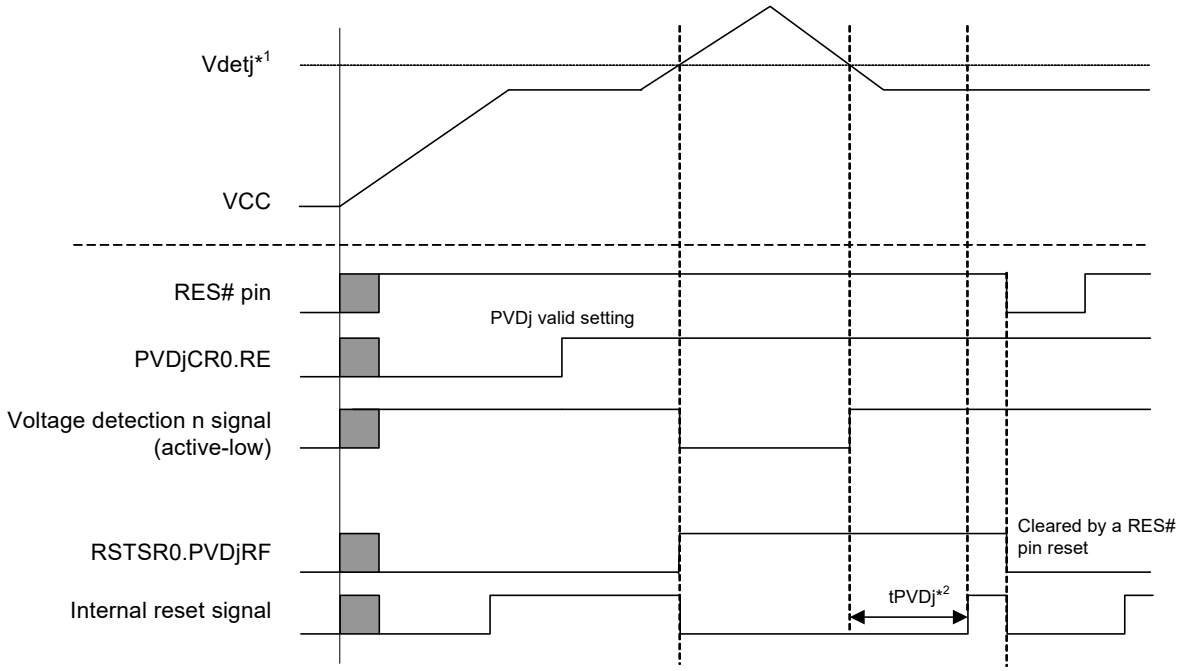
If VCC rises to or above Vdet1 while the voltage monitoring 1 interrupt/reset enable bit (RIE) of voltage monitoring 1 circuit control register 0 (PVD1CR0) is 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (RI) is 1 (selecting generation of a reset in response to detection of a low voltage), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 1 reset.
The device is released from the internal reset state following the voltage monitoring 1 reset time (tPVD1) having elapsed after VCC has risen to or above Vdet1 and then fallen below Vdet1, and the CPU starts reset exception handling.
If VCC rises to or above Vdet2 while the voltage monitoring 2 interrupt/reset enable bit (RIE) of voltage monitoring 2 circuit control register 0 (PVD2CR0) is 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (RI) is 1 (selecting generation of a reset in response to detection of a low voltage), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset.
The device is released from the internal reset state following the voltage monitoring 2 reset time (tPVD2) having elapsed after VCC has risen to or above Vdet2 and then fallen below Vdet2, and the CPU starts reset exception handling.
If VCC rises to or above Vdet4 while the voltage monitoring 4 reset enable bit (RE) of voltage monitoring 4 circuit control register 0 (PVD4CR0) is 1 (enabling generation of a reset by the voltage detection circuit), the RSTSR0.PVD4RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 4 reset.
The device is released from the internal reset state following the voltage monitoring 4 reset time (tPVD4) having elapsed after VCC has risen to or above Vdet4 and then fallen below Vdet4, and the CPU starts reset exception handling.
If VCC rises to or above Vdet5 while the voltage monitoring 5 reset enable bit (RE) of voltage monitoring 5 circuit control register 0 (PVD5CR0) is 1 (enabling generation of a reset by the voltage detection circuit), the RSTSR0.PVD5RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 5 reset.
The device is released from the internal reset state following the voltage monitoring 5 reset time (tPVD5) having elapsed after VCC has risen to or above Vdet5 and then fallen below Vdet5, and the CPU starts reset exception handling.

Figure 6.4 shows an example of the operation of the voltage monitoring 1 and 2 resets. Figure 6.5 shows an example of the operation of the voltage monitoring 4 and 5 resets.



Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdeti is the detection level for the voltage monitoring 1 or 2 reset. (i = 1, 2)
Note 2. tPVDi is the period of the voltage monitoring 1 or 2 reset. (i = 1, 2)

Figure 6.4 Example of Operation of Voltage Monitoring 1 and 2 Resets (RHSEL = 1)

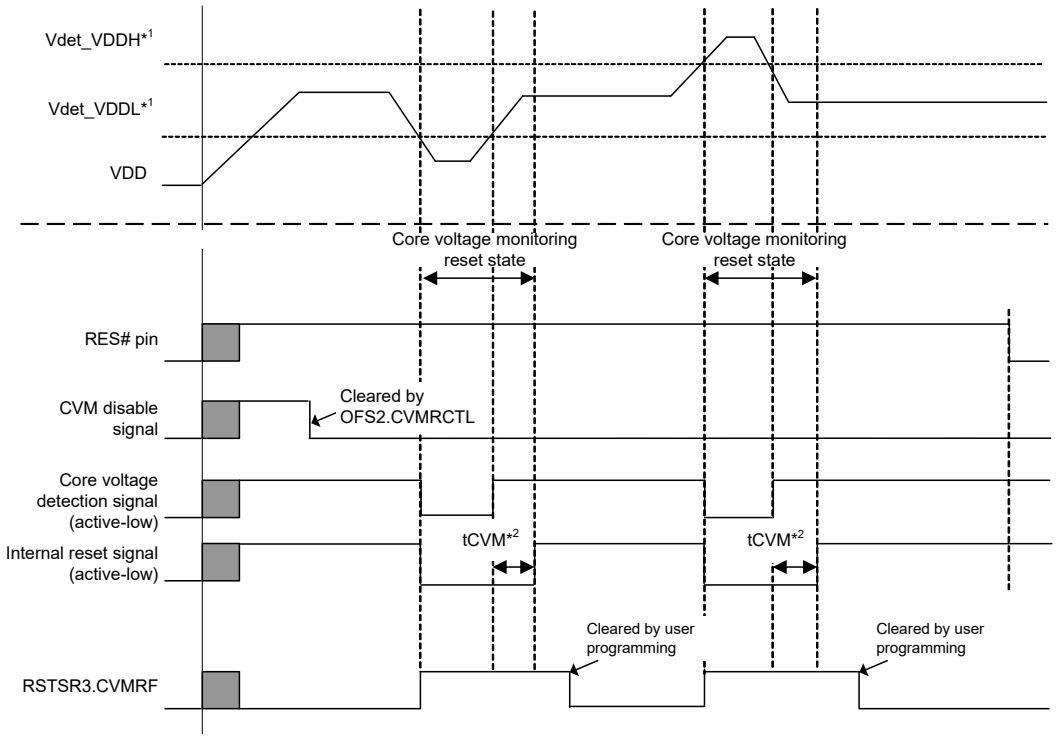


Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdetj is the detection level for the voltage monitoring 4 or 5 reset. (j = 4, 5)
Note 2. tPVDj is the period of the voltage monitoring 4 or 5 reset. (j = 4, 5)

Figure 6.5 Example of Operation of Voltage Monitoring 4 and 5 Resets (RHSEL = 1)

6.3.4 Core Voltage Monitoring Reset

A core voltage monitoring reset is an internal reset generated by the core voltage monitoring circuit.
If VDD falls below Vdet_VDDL or exceeds Vdet_VDDH while the OFS2.CVMRCTL bit is 0 (enabling the core voltage monitoring reset after a reset), the RSTSR3.CVMRF flag is set to 1 and the core voltage detection circuit generates a core voltage monitoring reset.
After that, if VDD falls between Vdet_VDDL and Vdet_VDDH, the device is released from the internal reset state following the core voltage monitoring reset time (tCVM) having elapsed, and the CPU starts reset exception handling.
The core voltage monitoring reset function does not operate during deep software standby mode.
If the RES# pin reset signal is asserted while a core voltage monitoring reset is being issued, the core voltage monitoring reset is not negated.
If a core voltage monitoring reset occurs, the values of the IO registers, SRAM, and backup registers are not guaranteed; whether they are retained depends on the response time until generation of the reset.
Figure 6.6 shows an example of the operation of the core voltage monitoring reset.



Note 1. For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 2. tCVM is the period of the core voltage monitoring reset.

Figure 6.6 Example of Operation of Core Voltage Monitoring Reset

6.3.5 Deep Software Standby Reset

This is an internal reset generated on release from deep software standby mode by an interrupt.
When a trigger for release from deep software standby mode is generated, a deep software standby reset is generated. This then causes release from the deep software standby reset state after the recovery time following release from deep software standby mode (tDSBY) has elapsed. Release from deep software standby mode also proceeds at the same time.
After release from deep software standby mode, the device is released from the internal reset state after the wait time following release from deep software standby mode (tDSBYWT) has elapsed, and the CPU starts reset exception handling.
For details on the deep software standby reset, see section 11, Low Power Consumption.

6.3.6 Independent Watchdog Timer Reset

This is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by the setting of the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written when refresh operation is disabled. After generation of an independent watchdog timer reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.
For details on the independent watchdog timer reset, see section 32, Independent Watchdog Timer (IWDTa).

6.3.7 Watchdog Timer Reset

This is an internal reset generated by the watchdog timer.
Output of the watchdog timer reset from the watchdog timer can be selected by the setting of the WDT reset control register (WDTRCR) or option function select register 0 (OFS0).
When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written while refresh operation is disabled. After generation of a watchdog timer reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.
For details on the watchdog timer reset, see section 31, Watchdog Timer (WDTA).

6.3.8 Software Reset

This is an internal reset generated by the software reset circuit.
When A501h is written to the SWRR register, a software reset is generated. After generation of a software reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.

6.3.9 Bus Error Reset

This is an internal reset generated by buses.
It is a consolidated reset generated by buses and is configured of the following reset:

- Bus error reset (master security attribute unit error, master MPU error, Illegal address error, protect zone filter error, slave bus error, bufferable write error)

Output of the bus error reset can be selected by the MSAOAD.OAD, MMPUOAD.OAD, BUSOAD.ILERROAD, BUSOAD.SLERROAD, or BUSOAD.BWERROAD bit.
After generation of a bus error reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.
For details on the bus error reset, see section 16, Buses. Determination of a bus error as the source of reset generation is similar to determination of a bus error as the source of interrupt generation. See section 15, Interrupt Controller (ICUM).

6.3.10 Common Memory Error Reset

This is an internal reset generated by SRAMs.
It is a consolidated reset generated by SRAMs and is configured of the following reset:

- Common memory error reset (ECC error in the SRAM)

Output of the memory error reset (ECC error in the SRAM) can be selected by the SRAMCR0.OAD or SRAMCR1.OAD bit.
After generation of a common memory error reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.
For details on the common memory error reset, see section 64, SRAM. Determination of a common memory error as the source of reset generation is similar to determination of a common memory error as the source of interrupt generation. See section 15, Interrupt Controller (ICUM).

6.3.11 Local Memory Error Reset

This is an internal reset generated by CPU caches or local RAMs.
It is a consolidated reset generated by CPU caches or local RAMs and is configured of the following reset:

- Local memory error reset (ECC error in the CPU cache or local RAM)

Output of the local memory error reset can be selected by the CAOAD.OAD, LRMCR0.OAD, or LRMCR1.OAD bit.
After generation of a local memory error reset, the device is released from the internal reset state after the internal reset time (tRESW2) has elapsed, and the CPU starts reset exception handling.
For details on the local memory error reset, see section 2, CPU. Determination of a local memory error as the source of reset generation is similar to determination of a local memory error as the source of interrupt generation. See section 15, Interrupt Controller (ICUM).

6.3.12 Temperature Monitoring Reset

This is an internal reset generated by the temperature sensor.

The temperature sensor detects the ambient temperature of the surrounding environment of the MCU. If it detects a temperature outside the specified range, a temperature monitoring reset is generated.

Figure 6.7 shows the procedure for setting the temperature sensor to enable the temperature monitoring reset function. If the temperature sensor is operating and the temperature monitoring reset function is enabled, temperature monitoring (abnormal temperature detection) proceeds. Figure 6.8 and Figure 6.9 show examples of the operation of the temperature monitoring reset for abnormally low and high temperatures, respectively.

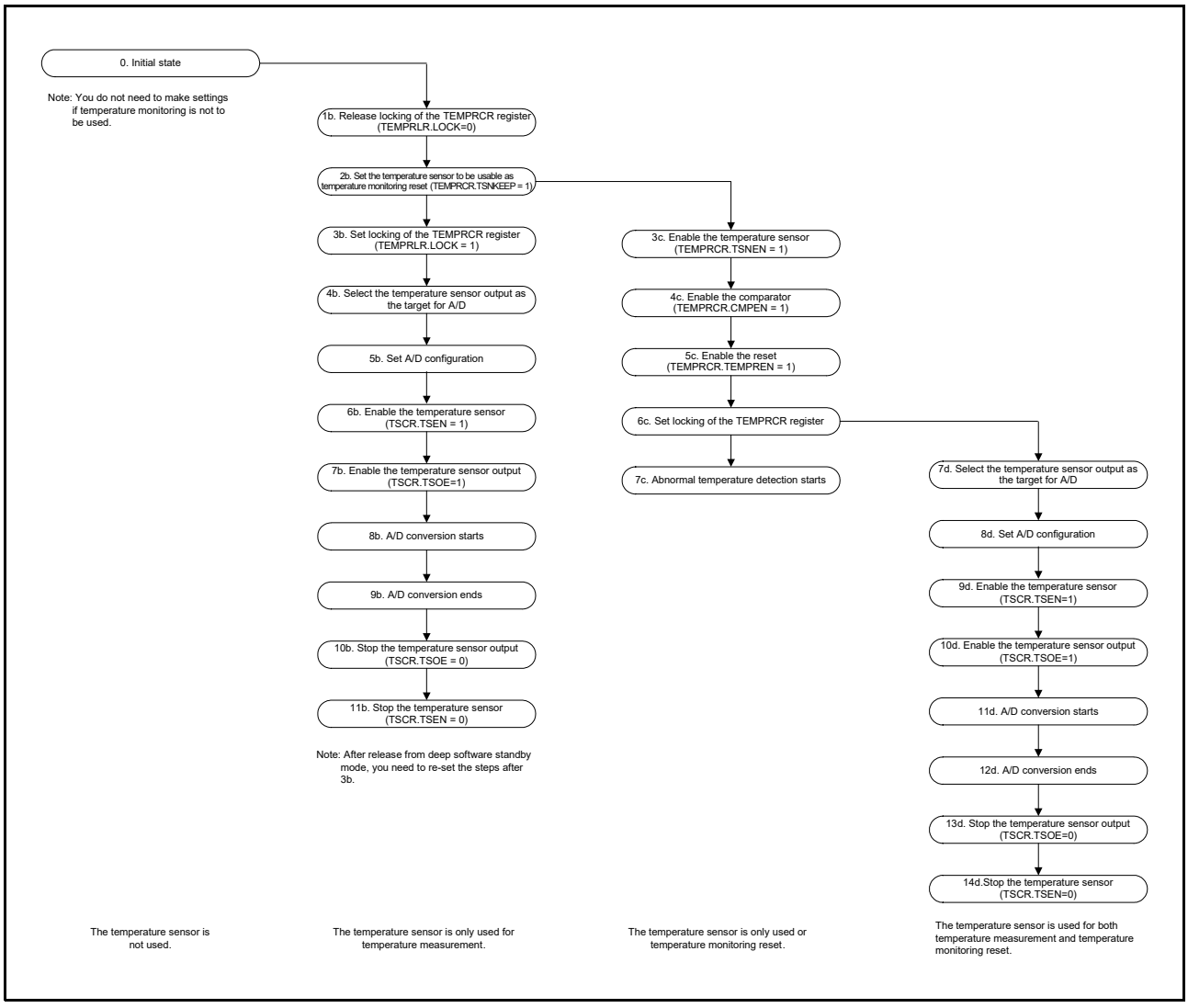
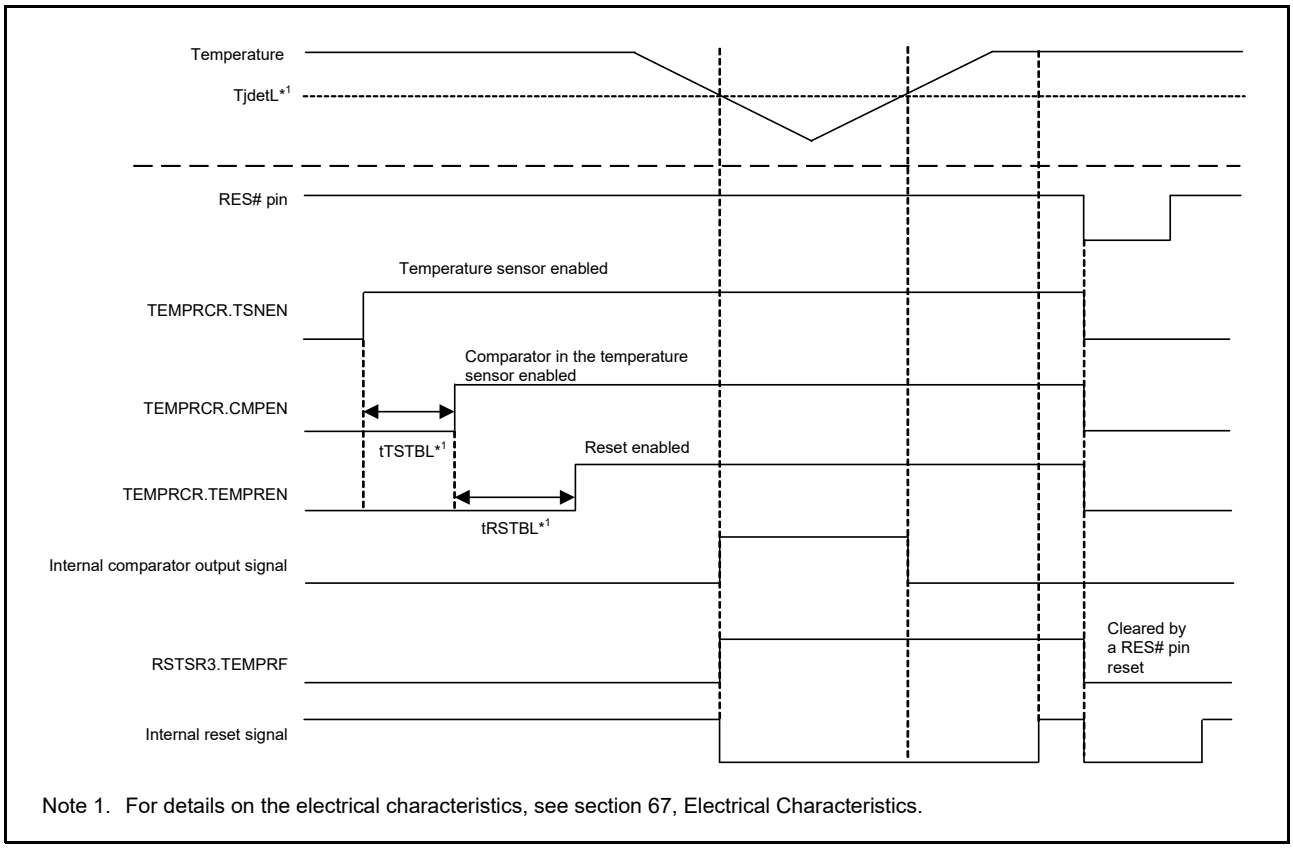
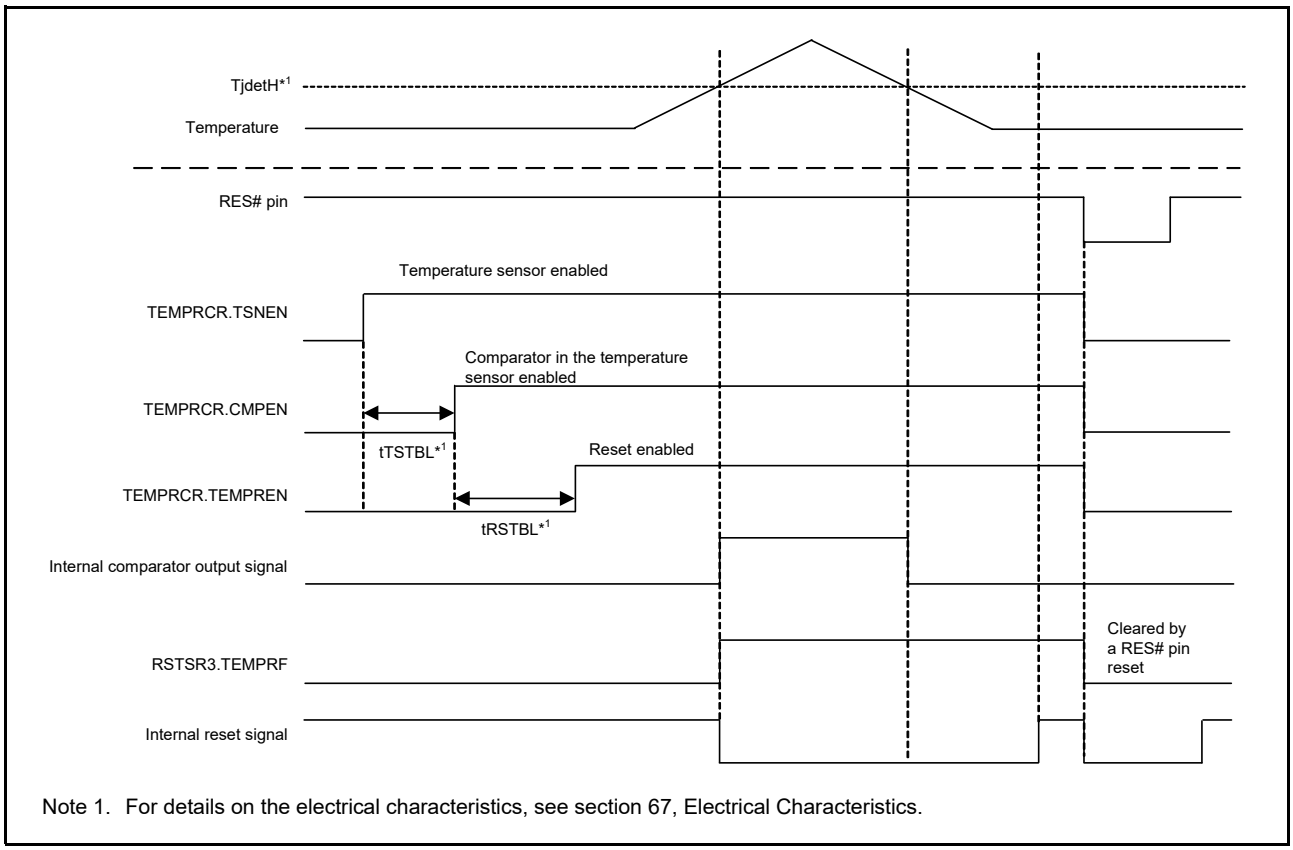


Figure 6.7 Temperature Sensor Setting Procedure



Note 1. For details on the electrical characteristics, see section 67, Electrical Characteristics.

Figure 6.8 Example of Operation of Temperature Monitoring Reset (Abnormally Low Temperature Detection)



Note 1. For details on the electrical characteristics, see section 67, Electrical Characteristics.

Figure 6.9 Example of Operation of Temperature Monitoring Reset (Abnormally High Temperature Detection)

6.3.13 Determination of Cold/Warm Start

The RSTSR2.CWSF flag allows determining whether a power-on reset has caused the reset processing (cold start) or the reset signal being input during operation has caused the reset processing (warm start).
The RSTSR2.CWSF flag is set to 0 when a power-on reset occurs (cold start). Otherwise, the flag is not set to 0. The flag is set to 1 when 1 is written to it. The flag does not change when 0 is written to it.
Figure 6.10 shows an example of the operation of cold/warm start determination.

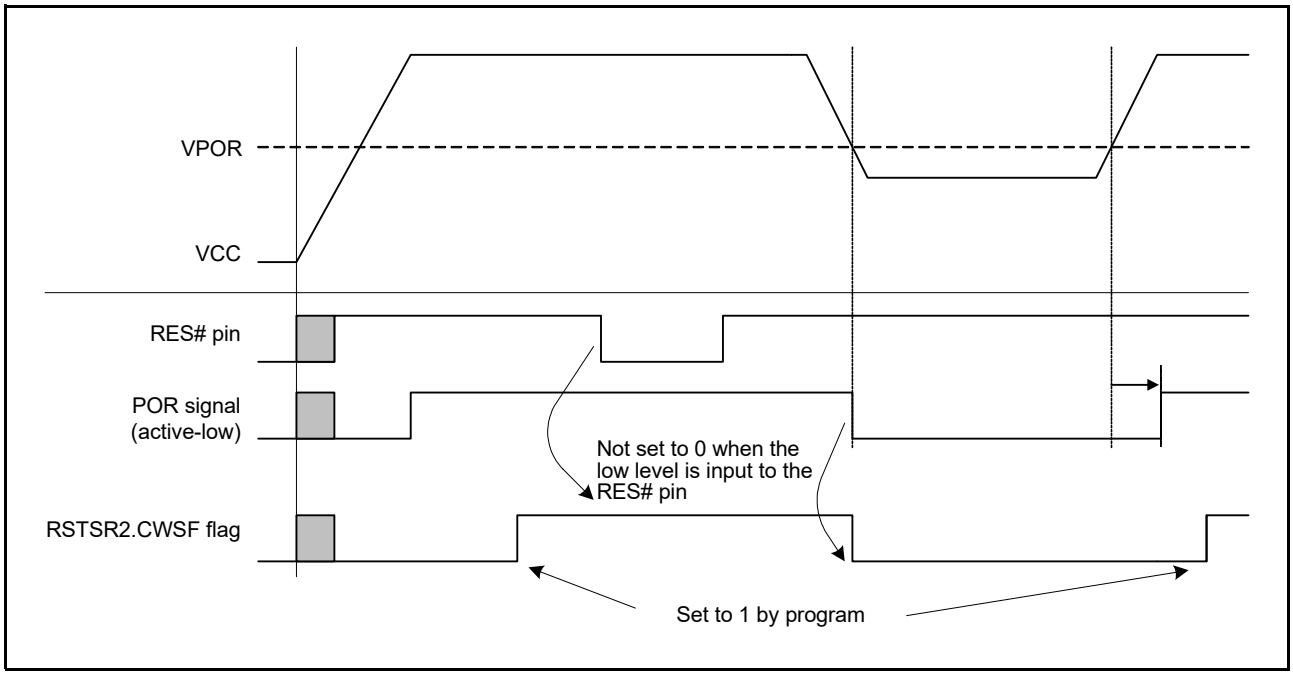
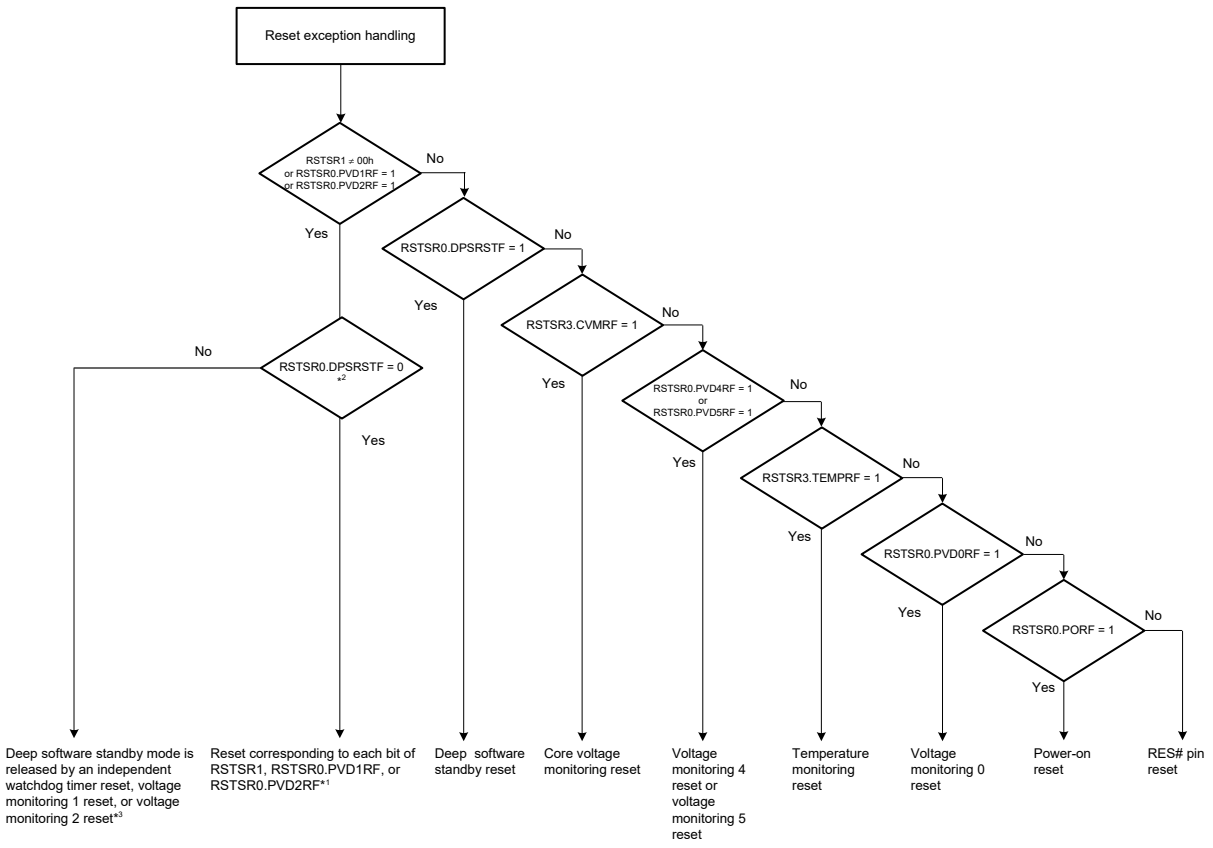


Figure 6.10 Example of Operation of Cold/Warm Start Determination

6.3.14 Determining the Source of Reset Generation

Reading of the RSTSR0, RSTSR1, and RSTSR3 registers can identify the type of reset which led to the reset exception handling.

Figure 6.11 shows an example flowchart to determine the source of reset generation.



Note 1. If any combination of resets corresponding to bits among the RSTSR1, RSTSR0.PVD1RF, and RSTSR0.PVD2RF flags occurred simultaneously, all reset flags for which the reset source condition occurred become 1.

Note 2. After checking that RSTSR1 ≠ 00h, RSTSR0.PVD1RF = 1, or RSTSR0.PVD2RF = 1, check RSTSR0.DPSRSTF to determine whether release from deep software standby mode occurred.

Note 3. Clear the reset status register (RSTSR0, RSTSR1, RSTSR2, RSTSR3) before a transition to deep software standby mode. If the reset flag is not cleared, the recovery source cannot be correctly identified.

Figure 6.11 Example Flowchart to Determine the Source of Reset Generation

6.3.15 Reset Output Function

Reset output is a function for notifying external devices of the internal reset state. The output pin function (RSTOUT#) is multiplexed with the RES# input pin function, and the low level is output at the time of a reset when the reset output function is enabled.

The output of an internal reset to the RSTOUT# pin as a reset output is itself recognized as if it were a reset input on the RES# pin. At this time, the reset control registers and reset source are initialized by the backflow from the reset output being handled as if it were a reset input on the RES# pin, so the internal reset cannot continue to be issued even if the MCU still satisfies the condition for having issued the reset. However, the core voltage monitoring reset is an exception because it can continue to be issued.

Disabling of the reset output function is selectable by the setting of the RODIS bit of option function select register 1 (OFS1). The width of the reset output pulse is selectable by the setting of the ROWSEL[2:0] bits of option function select register 1 (OFS1).

OFS1.ROWSEL[2:0] = 000b: 0.25 ms or above
OFS1.ROWSEL[2:0] = 100b: 0.5 ms or above
OFS1.ROWSEL[2:0] = 101b: 1 ms or above
OFS1.ROWSEL[2:0] = 110b: 5 ms or above
OFS1.ROWSEL[2:0] = 111b: 10 ms or above

When an internal reset is output as a reset output, the reset flags of the RSTSR0 register other than the RSTSR0.PORF and RSTSR0.DPSRSTF flags, and the RSTSR1 register and RSTSR3 register are initialized by the reset output. Therefore, to determine the reset source when the reset output function is enabled, read the RSTSR0.PORF and RSTSR0.DPSRSTF flags and the ROUTSR0, ROUTSR1, and ROUTSR2 registers.

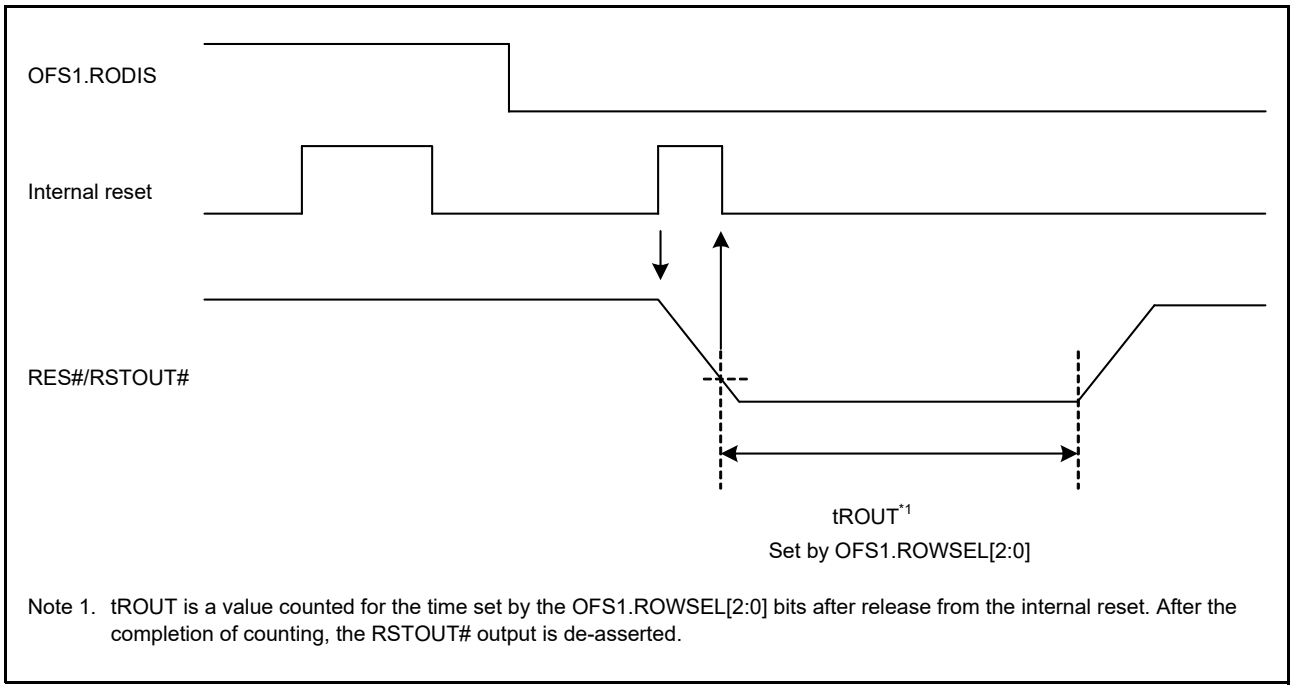
After having determined the reset source, clear all bits of the RSTSR0, RSTSR1, RSTSR2, RSTSR3, ROUTSR0, ROUTSR1, and ROUTSR2 registers in preparation for generation of the next reset.

Table 6.10 shows the relationship between the reset output function settings and the reset source determination flags.

Table 6.10 Relationship between Reset Output Function Settings and Reset Source Determination Flags

		Reset Output Determination Flag	
		Reset Output Function Enabled (OFS1.RODIS = 0)	Reset Output Function Disabled (OFS1.RODIS = 1)
Reset source	Power-on reset	—	RSTSR0.PORF
	Voltage monitoring 0 reset	ROUTSR0.ROPVD0RF	RSTSR0.PVD0RF
	Voltage monitoring 1 reset	ROUTSR0.ROPVD1RF	RSTSR0.PVD1RF
	Voltage monitoring 2 reset	ROUTSR0.ROPVD2RF	RSTSR0.PVD2RF
	Voltage monitoring 4 reset	ROUTSR0.ROPVD4RF	RSTSR0.PVD4RF
	Voltage monitoring 5 reset	ROUTSR0.ROPVD5RF	RSTSR0.PVD5RF
	Deep software standby reset	—	RSTSR0.DPSRSTF
	Independent watchdog timer reset	ROUTSR1.ROIWDTRF	RSTSR1.IWDTRF
	Watchdog timer reset	ROUTSR1.ROWDT0RF	RSTSR1.WDT0RF
	Software reset	ROUTSR1.ROSWRF	RSTSR1.SWRF
	Local memory error reset	ROUTSR1.ROLM0RF	RSTSR1.LM0RF
	Bus error reset	ROUTSR1.ROBUSRF	RSTSR1.BUSRF
	Common memory error reset	ROUTSR1.ROCMRF	RSTSR1.CMRF
	Core voltage monitoring reset	ROUTSR2.ROCVMRF	RSTSR3.CVMRF
	Temperature monitoring reset	ROUTSR2.ROTEMPRF	RSTSR3.TEMPRF

Figure 6.12 shows an example of the operation of the reset output. For details of tROUT, see section 67, Electrical Characteristics.



Note 1. tROUT is a value counted for the time set by the OFS1.ROWSEL[2:0] bits after release from the internal reset. After the completion of counting, the RSTOUT# output is de-asserted.

Figure 6.12 Example of Operation of Reset Output

Figure 6.13 is a block diagram of the reset output.

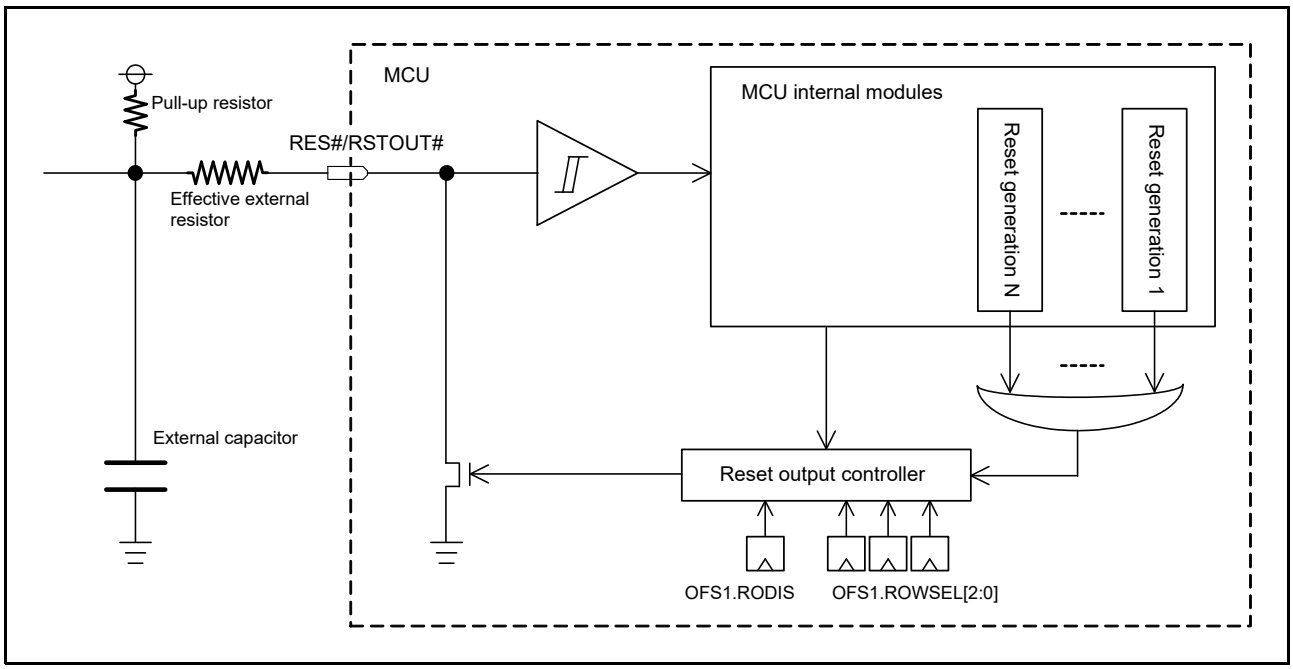


Figure 6.13 Block Diagram of Reset Output

6.4 Protection from Reset Control by a Non-Secure Program

Generation of resets from some of the reset sources is controllable. The settings of the bits of the SYRSTMSK0 and SYRSTMSK2 registers control (enable or disable) generation of resets from those sources. These registers are only accessible from a secure program.

A reset from a source that is disabled is still generated but is not propagated to the MCU. Also, the corresponding reset flags in the RSTSR0 and RSTSR1 registers are not set.

The controllable reset sources are as follows.

- Independent watchdog timer reset
- Watchdog timer reset
- Common memory error reset
- Local memory error reset
- Bus error reset
- Voltage monitoring 1 reset
- Voltage monitoring 2 reset
- Software reset

7. Option-Setting Memory (OFSM)

7.1 Overview

The option-setting memory is a set of registers used to determine the state of the MCU after a reset. The option-setting memory is allocated to the configuration setting area in the extra MRAM area.
Table 7.1 and Table 7.2 show the registers in the option-setting memory area.

Table 7.1 Option-Setting Memory (Extra MRAM Area)

Address	Register Name	Security Attribute*1
06C4 EC60h	Bank Select Register for Non-Secure Application (BANKSEL_NS)	Non-secure
06C4 EC70h	Block Select Register for Non-Secure Application (BLSEL_NS)	Non-secure
06C9 F480h	Data MRAM Block Protect Register 0 for Non-Secure Application (BPSD0_NS)	Non-secure
06C9 F500h to 06C9 F50Ch	Block Swap Setting Register n for Non-Secure Application (BLKSWPn_NS) (n = 0 to 3)	Non-secure
06C9 F600h to 06C9 F63Ch, 06C9 F700h to 06C9 F71Ch	Block Protect Register n for Non-Secure Application (BPSn_NS) (n = 0 to 15, 64 to 71)	Non-secure
86C2 E850h	Bank Mode Select Register (DUALSEL)	Secure
86C2 E870h	Start-Up Area Setting Register (SAS)	Secure
86C2 E874h	Start-Up Area Setting Register 2 (SAS2)	Secure
86C2 E9B0h	Bank Select Register for Secure Application (BANKSEL_S)	Secure
86C2 E9C0h	Block Select Register for Secure Application (BLSEL_S)	Secure
86C2 E9E0h	BANKSEL Attribute Select Register (BANKSEL_SEL)	Secure
86C2 E9F0h	BLSEL Attribute Select Register (BLSEL_SEL)	Secure
86C9 F050h	MRAM Security Attribute Boundary Register (MRAMSABR)	Secure
86C9 F080h	Data MRAM Block Protect Register 0 for Secure Application (BPSD0_S)	Secure
86C9 F100h to 86C9 F10Ch	Block Swap Setting Register n for Secure Application (BLKSWPn_S) (n = 0 to 3)	Secure
86C9 F200h to 86C9 F23Ch, 86C9 F300h to 86C9 F31Ch	Block Protect Register n for Secure Application (BPSn_S) (n = 0 to 15, 64 to 71)	Secure

Note 1. Only secure application developers can set values in registers with the security attribute secure.

Table 7.2 Option-Setting Memory (OTP Area in Extra MRAM) (1/2)

Address	Register Name	Memory Type	Security Attribute*1
06E2 0810h	Non-Secure Switchable OTP Memory Type Setting Register (SOTPMS_NS)	OTP	Non-secure
06E2 0850h	Option Function Select Register 1 for Non-Secure Application (OFS1_NS)	Switchable OTP	Non-secure
06E2 0C00h to 06E2 0C0Ch, 06E2 0C40h, 06E2 0C44h	Permanent Block Protect Register m for Non-Secure Application (PBPSm_NS) (m = 0 to 3, 16, 17)	OTP	Non-secure
06E2 0C80h	Data MRAM Permanent Block Protect Register 0 for Non-Secure Application (PBPSD0_NS)	OTP	Non-secure
06E3 0E80h	Non-secure Switchable OTP Option Setting Memory Protect Register 0 (SOFSPS0_NS)	OTP	Non-secure
06E3 0EE0h	Non-secure MRAM Option Setting Memory Protect Register 0 (MOFSPS0_NS)	OTP	Non-secure
86E0 0000h to 86E0 002Ch	OEM_ROOT_PK0 Hash Register [i] (HOEMRTPK0[i]) (i = 0 to 11)	OTP	Secure
86E0 0030h to 86E0 005Ch	OEM_ROOT_PK1 Hash Register [i] (HOEMRTPK1[i]) (i = 0 to 11)	OTP	Secure
86E0 0060h to 86E0 008Ch	OEM_ROOT_PK2 Hash Register [i] (HOEMRTPK2[i]) (i = 0 to 11)	OTP	Secure
86E0 0090h to 86E0 00BCh	OEM_ROOT_PK3 Hash Register [i] (HOEMRTPK3[i]) (i = 0 to 11)	OTP	Secure
86E0 0400h	FSBL Control Register 0 (FSBLCTRL0)	OTP	Secure

Table 7.2 Option-Setting Memory (OTP Area in Extra MRAM) (2/2)

Address	Register Name	Memory Type	Security Attribute*1
86E0 0404h	FSBL Control Register 1 (FSBLCTRL1)	OTP	Secure
86E0 0408h	FSBL Control Register 2 (FSBLCTRL2)	OTP	Secure
86E0 0414h	Measurement Report Start Address Register (SAMR)	OTP	Secure
86E0 0420h	Code Certificate 0 Start Address Register 0 (SACC00)	OTP	Secure
86E0 0430h	Code Certificate 1 Start Address Register 0 (SACC10)	OTP	Secure
86E0 0440h	Code Certificate 0 Start Address Register 1 (SACC01)	OTP	Secure
86E0 0450h	Code Certificate 1 Start Address Register 1 (SACC11)	OTP	Secure
86E0 0460h	Code Certificate 0 Start Address Register 2 (SACC02)	OTP	Secure
86E0 0470h	Code Certificate 1 Start Address Register 2 (SACC12)	OTP	Secure
86E0 0480h	Code Certificate 0 Start Address Register 3 (SACC03)	OTP	Secure
86E0 0490h	Code Certificate 1 Start Address Register 3 (SACC13)	OTP	Secure
86E1 0610h	Secure Switchable OTP Memory Type Setting Register (SOTPMS_S)	OTP	Secure
86E1 0640h	Option Function Select Register 0 (OFS0)	Switchable OTP	Secure
86E1 0644h	Option Function Select Register 2 (OFS2)	Switchable OTP	Secure
86E1 064Ch	Endian Select Register (MDE)	Switchable OTP	Secure
86E1 07A0h	Option Function Select Register 1 for Secure Application (OFS1_S)	Switchable OTP	Secure
86E1 07D0h	OFS1 Attribute Select Register (OFS1_SEL)	Switchable OTP	Secure
86E1 0A00h to 86E1 0A0Ch, 86E1 0A40h, 86E1 0A44h	Permanent Block Protect Register m for Secure Application (PBPSm_S) (m = 0 to 3, 16, 17)	OTP	Secure
86E1 0A80h	Data MRAM Permanent Block Protect Register 0 for Secure Application (PBPSD0_S)	OTP	Secure
86E3 0E00h to 86E3 0E0Ch	Secure Switchable OTP Option Setting Memory Protect Register n (SOFSPSn_S) (n = 0 to 3)	OTP	Secure
86E3 0E10h	OEM_ROOT_PK Revoke Register (REVOKE)	OTP	Secure
86E3 0E20h	HUK Zeroization Register (ZHUK)	OTP	Secure
86E3 0E30h	Anti-Rollback Counter Setting Register (ARCCS)	OTP	Secure
86E3 0E50h	User Extra MRAM Configuration Register (UMRECR)	OTP	Secure
86E3 0E60h to 86E3 0E6Ch	Secure MRAM Option Setting Memory Protect Register n (MOFSPSn_S) (n = 0 to 3)	OTP	Secure
86F4 1D00h to 86F4 1D7Ch	Anti-Rollback Counter for Secure Application [i] (ARCS[i]) (i = 0 to 31)	OTP	Secure
86F4 1E00h to 86F4 1E1Ch	Anti-Rollback Counter 0 for Non-Secure Application [i] (ARCNS0[i]) (i = 0 to 7)	OTP	Secure
86F4 1E20h to 86F4 1E3Ch	Anti-Rollback Counter 1 for Non-Secure Application [i] (ARCNS1[i]) (i = 0 to 7)	OTP	Secure
86F4 1E40h to 86F4 1E5Ch	Anti-Rollback Counter 2 for Non-Secure Application [i] (ARCNS2[i]) (i = 0 to 7)	OTP	Secure
86F4 1E60h to 86F4 1E7Ch	Anti-Rollback Counter 3 for Non-Secure Application [i] (ARCNS3[i]) (i = 0 to 7)	OTP	Secure
86F4 1E00h to 86F4 1E7Ch	Anti-Rollback Counter for Non-Secure Application [i] (ARCNS[i]) (i = 0 to 31)	OTP	Secure
86F4 1F00h, 86F4 1F04h	Anti-Rollback Counter for OEM Boot Loader [i] (ARCOEMBL[i]) (i = 0, 1)	OTP	Secure

Note 1. Only secure application developers can set values in registers with the security attribute secure.

7.2 Register Descriptions

7.2.1 Secure Switchable OTP Memory Type Setting Register (SOTPMS_S), Non-Secure Switchable OTP Memory Type Setting Register (SOTPMS_NS)

Address(es): OFSM.SOTPMS_S 86E1 0610h, OFSM_NS.SOTPMS_NS 06E2 0810h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTPMS
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	SOTPMS	Switchable OTP Memory Type Select	0: Memory type of the switchable OTP is OTP. 1: Memory type of the switchable OTP is MRAM.	R
b31 to b1	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

SOTPMS Bit (Switchable OTP Memory Type Select)

This bit specifies the memory type for the registers located in the switchable OTP area (06E2 0820h to 06E2 087Fh, 86E1 0620h to 86E1 06EFh, and 86E1 0710h to 86E1 07FFh).

After setting this bit to 0, set the values for the registers in the switchable OTP area again.

7.2.2 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 86E1 0640h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	WDT0SLCSTP	—	WDT0RSTIRQS	WDT0RPSS[1:0]		WDT0RPES[1:0]		WDT0CKS[3:0]				WDT0TOPS[1:0]		WDT0STRT	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IWDTSLCSTP	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically started after a reset (auto-start mode). 1: IWDT is halted after a reset (register start mode).	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT Clock Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request or maskable interrupt request is output. 1: Reset signal is output.	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counter continues counting even in sleep mode, deep sleep mode, software standby mode, or deep software standby mode 1. 1: Counter stops when entering sleep mode, deep sleep mode, software standby mode, or deep software standby mode 1.	R
b16, b15	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b17	WDT0STRT	WDT Start Mode Select	0: WDT is automatically started after a reset (auto-start mode) 1: WDT is halted after a reset (register start mode)	R

Bit	Symbol	Bit Name	Description	R/W
b19, b18	WDT0TOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b23 to b20	WDT0CKS[3:0]	WDT Clock Division Ratio Select	b23 b20 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Settings other than above are prohibited.	R
b25, b24	WDT0RPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R
b27, b26	WDT0RPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R
b28	WDT0RSTIRQS	WDT Reset Interrupt Request Select	0: Non-maskable interrupt request or maskable interrupt request is output. 1: Reset signal is output.	R
b29	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b30	WDT0SLCSTP	WDT Sleep Mode Count Stop Control	0: Counter continues counting even in sleep mode or deep sleep mode. 1: Counter stops when entering sleep mode or deep sleep mode.	R
b31	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value. Once the memory type is set to OTP, it is not possible to change a bit that was set to 0 back to 1.

The OFS0 register specifies the operation of the independent watchdog timer (IWDT) and the watchdog timer (WDT) after a reset.

IWDTSTRT Bit (IWDT Start Mode Select)

This bit selects the IWDT start mode after a reset (register start mode or auto-start mode). When auto-start mode is selected, the OFS0 register setting is reflected in each IWDT register.

IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)

These bits specify the timeout period (time until the counter underflows) in cycles of the divided clock set by the IWDTCKS[3:0] bits. 128, 512, 1024, or 2048 cycles can be selected. The time until underflow after a refresh (number of IWDTCLK cycles) is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.
For details, refer to section 32, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT Clock Division Ratio Select)

These bits select the division ratio of the prescaler that divides IWDTCLK from among divided-by 1, 16, 32, 64, 128, and 256. In combination with the setting of the IWDTTOPS[1:0] bits, the IWDT count period can be set within the range of 128 to 524,288 cycles of the IWDTCLK.
For details, refer to section 32, Independent Watchdog Timer (IWDTa).

IWDTRPES[1:0] Bits (IWDT Window End Position Select)

These bits select the window end position of the counter from 75%, 50%, 25%, or 0% of the count period. It is 100% at when the count starts, and 0% when an underflow occurs. Select a value smaller than the window start position for the window end position. If a value larger than the window start position is selected, the setting of this bit becomes invalid (= 0%).

The counter values for the window start and end positions set by the IWDTRPSS[1:0] and IWDTRPES[1:0] bits depend on the setting of the IWDTTOPS[1:0] bits.

For details, refer to section 32, Independent Watchdog Timer (IWDTa).

IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)

These bits select the window start position of the counter from 100%, 75%, 50%, or 25% of the count period. It is 100% at when the count starts, and 0% when an underflow occurs. The period from the window start position to the window end position is the refresh-permitted period and the other periods are the refresh-prohibited periods.

For details, refer to section 32, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

This bit selects the action to be taken when an underflow or refresh error occurs. An independent watchdog timer reset or an interrupt request (non-maskable or maskable) can be selected.

For details, refer to section 32, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit specifies whether to stop the IWDT counting when entering sleep mode, deep sleep mode, software standby mode, or deep software standby mode 1.

For details, refer to section 32, Independent Watchdog Timer (IWDTa).

WDT0STRT Bit (WDT Start Mode Select)

This bit selects the WDT start mode after a reset (register start mode or auto-start mode). When auto-start mode is selected, the OFS0 register setting is reflected in each WDT register.

WDT0TOPS[1:0] Bits (WDT Timeout Period Select)

These bits specify the timeout period (time until the counter underflows) in cycles of the divided clock set by the WDT0CKS[3:0] bits. 1024, 4096, 8192, or 16384 cycles can be selected. The time until underflow after a refresh (number of PCLKB cycles) is determined by the combination of the WDT0CKS[3:0] and WDT0TOPS[1:0] bits.

For details, refer to section 31, Watchdog Timer (WDTA).

WDT0CKS[3:0] Bits (WDT Clock Division Ratio Select)

These bits select the division ratio of the prescaler that divides PCLKB from among divided-by 4, 64, 128, 512, 2048, and 8192. In combination with the setting of the WDT0TOPS[1:0] bits, the WDT count period can be set within the range of 4096 to 134,217,728 cycles of the PCLKB.

For details, refer to section 31, Watchdog Timer (WDTA).

WDT0RPES[1:0] Bits (WDT Window End Position Select)

These bits select the window end position of the counter from 75%, 50%, 25%, or 0% of the count period. It is 100% at when the count starts, and 0% when an underflow occurs. Select a value smaller than the window start position for the window end position. If a value larger than the window start position is selected, the setting of this bit becomes invalid (= 0%).

The counter values for the window start and end positions set by the WDT0RPSS[1:0] and WDT0RPES[1:0] bits depend on the setting of the WDT0TOPS[1:0] bits.

For details, refer to section 31, Watchdog Timer (WDTA).

WDT0RPSS[1:0] Bits (WDT Window Start Position Select)

These bits select the window start position of the counter from 100%, 75%, 50%, or 25% of the count period. It is 100% at when the count starts, and 0% when an underflow occurs. The period from the window start position to the window end position is the refresh-permitted period and the other periods are the refresh-prohibited periods.
For details, refer to section 31, Watchdog Timer (WDTA).

WDT0RSTIRQS Bit (WDT Reset Interrupt Request Select)

This bit selects the action to be taken when an underflow or refresh error occurs. A watchdog timer reset or an interrupt request (non-maskable or maskable) can be selected.
For details, refer to section 31, Watchdog Timer (WDTA).

WDT0SLCSTP Bit (WDT Sleep Mode Count Stop Control)

This bit specifies whether to stop the WDT counting when entering sleep mode or deep sleep mode.
For details, refer to section 31, Watchdog Timer (WDTA).

7.2.3 Option Function Select Register 2 (OFS2)

Address(es): OFSM.OFS2 86E1 0644h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVMRCTL	—
	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	CVMRCTL	Core Voltage Monitoring Reset Control	0: Core voltage monitoring reset is enabled after a reset. 1: Core voltage monitoring reset is disabled after a reset.	R
b31 to b2	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value. Once the memory type is set to OTP, it is not possible to change a bit that was set to 0 back to 1.

The OFS2 register controls the internal core voltage of the MCU.

CVMRCTL Bit (Core Voltage Monitoring Reset Control)

This bit selects whether the core voltage monitoring reset is enabled or disabled after a reset.

7.2.4 Endian Select Register (MDE)

Address(es): OFSM.MDE 86E1 064Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value. Once the memory type is set to OTP, it is not possible to change a bit that was set to 0 back to 1.

The MDE register selects the endian for the CPU.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.

7.2.5 Bank Mode Select Register (DUALSEL)

Address(es): OFSM.DUALSEL 86C2 E850h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKMD[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BANKMD[2:0]	Bank Mode Select	b2 b0 0 0 0: Dual mode 1 1 1: Linear mode Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The DUALSEL register selects the bank mode in the dual bank function of the code MRAM.

BANKMD[2:0] Bits (Bank Mode Select)

These bits select the bank mode in the dual bank function of the code MRAM.

7.2.6 Bank Select Register for Secure Application (BANKSEL_S), Bank Select Register for Non-Secure Application (BANKSEL_NS)

Address(es): OFSM.BANKSEL_S 86C2 E9B0h, OFSM_NS.BANKSEL_NS 06C4 EC60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BANKSWP[2:0]	Startup Bank Switch	b2 b0 0 0 0: Set Bank 1 as the startup bank.*2 1 1 1: Set Bank 0 as the startup bank.*2 Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.
Note 2. A startup bank is a bank located at address 7FFF FFFFh in the non-secure area and at address FFFF FFFFh in the secure area.

The BANKSEL register selects the program startup bank when the code MRAM is in dual mode. There are two registers: one for secure application developers (BANKSEL_S) and one for non-secure application developers (BANKSEL_NS). Which setting is used is specified by the BANKSEL_SEL register.

BANKSWP[2:0] Bits (Startup Bank Switch)

These bits select whether to start the program from Bank 0 or Bank 1 of the code MRAM when the code MRAM is in dual mode.
The setting of these bits is invalid in linear mode.

7.2.7 BANKSEL Attribute Select Register (BANKSEL_SEL)

Address(es): OFSM.BANKSEL_SEL 86C2 E9E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP_SEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BANKSWP_SEL[2:0]	BANKSWP Attribute Select	b2 b0 0 0 0: The value set in the BANKSEL_S.BANKSWP[2:0] bits is selected. 1 1 1: The value set in the BANKSEL_NS.BANKSWP[2:0] bits is selected. Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 0.	R

Note 1. The value for a blank product is 0000 0000h. After setting a value, this register will retain the set value.

The BANKSEL_SEL register selects whether the settings in the BANKSEL_S register or the BANKSEL_NS register are enabled.

BANKSWP_SEL[2:0] Bits (BANKSWP Attribute Select)

These bits are used to select whether the startup bank switch is configured by the non-secure application developer or the secure application developer.

7.2.8 Block Select Register for Secure Application (BLSEL_S), Block Select Register for Non-Secure Application (BLSEL_NS)

Address(es): OFSM.BLSEL_S 86C2 E9C0h, OFSM_NS.BLSEL_NS 06C4 EC70h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BLSWAREA
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	BLSWAREA	Block Swap Area Select	0: Enable block swapping for half of the entire code MRAM area. 1: Enable block swapping for a quarter of the entire code MRAM area.	R
b31 to b1	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The BLSEL register specifies the target area for the block swap function when the code MRAM is in linear mode. There are two registers: one for secure application developers (BLSEL_S) and one for non-secure application developers (BLSEL_NS). Which setting is used is specified by the BLSEL_SEL register.

BLSWAREA Bit (Block Swap Area Select)

This bit selects whether to enable block swapping of the code MRAM at half or a quarter of the total capacity. Table 7.3 lists the addresses subject to block swapping. Depending on the setting of this bit, 8-Kbytes blocks can be swapped between swap area 0 and swap area 1 on a block-by-block basis.
The setting of this bit is invalid in dual mode.

Table 7.3 Block Swap Area

Code MRAM Size	BLSWAREA bit	Block Swap Area (Offset Address)	Swap Area 0 (Offset Address)	Swap Area 1 (Offset Address)
1 Mbyte	0	F0 0000h to F7 FFFFh	F0 0000h to F3 FFFFh	F4 0000h to F7 FFFFh
	1	F0 0000h to F3 FFFFh	F0 0000h to F1 FFFFh	F2 0000h to F3 FFFFh
512 Kbytes	0	F8 0000h to FB FFFFh	F8 0000h to F9 FFFFh	FA 0000h to FB FFFFh
	1	F8 0000h to F9 FFFFh	F8 0000h to F8 FFFFh	F9 0000h to F9 FFFFh

7.2.9 BLSEL Attribute Select Register (BLSEL_SEL)

Address(es): OFSM.BLSEL_SEL 86C2 E9F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BLSWAREA_SEL
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	BLSWAREA_SEL	BLSWAREA Attribute Select	0: The value set in the BLSEL_S.BLSWAREA bit is selected. 1: The value set in the BLSEL_NS.BLSWAREA bit is selected.	R
b31 to b1	—	Reserved	When reading, these bits return the value written by the user. The write value should be 0.	R

Note 1. The value for a blank product is 0000 0000h. After setting a value, this register will retain the set value.

The BLSEL_SEL register selects whether the BLSEL_S register setting or the BLSEL_NS register setting is to be used.

BLSWAREA_SEL Bit (BLSWAREA Attribute Select)

This bit is used to select whether the size of the block swap area is configured by the non-secure application developer or the secure application developer.

7.2.10 Block Swap Setting Register n for Secure Application (BLKSWPn_S) (n = 0 to 3), Block Swap Setting Register n for Non-Secure Application (BLKSWPn_NS) (n = 0 to 3)

Address(es): OFSM.BLKSWP0_S 86C9 F100h to OFSM.BLKSWP3_S 86C9 F10Ch,
OFSM_NS.BLKSWP0_NS 06C9 F500h to OFSM_NS.BLKSWP3_NS 06C9 F50Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SWP7[3:0]				SWP6[3:0]				SWP5[3:0]				SWP4[3:0]			
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SWP3[3:0]				SWP2[3:0]				SWP1[3:0]				SWP0[3:0]			
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

This register is used to enable or disable block swapping.

SWPi[3:0] Bits (Swap i) (i = 0 to 7)

These bits specify whether to swap the corresponding address blocks. Setting them to 0000b swaps the corresponding address blocks, while setting them to 1111b reverts them to their original state. No other values are permitted.
The address blocks corresponding to each set of bits are shown in Table 7.4 through Table 7.7.

Table 7.4 Corresponding Address Blocks for Each Set of Bits (Code MRAM Size: 1 Mbyte, the BLSEL.BLSWAREA bit is 0)

Register	Bits	Address Blocks to Be Swapped (Offset Address)
BLKSWP0	SWP0[3:0]	F7 E000h to F7 FFFFh and F3 E000h to F3 FFFFh
	SWP1[3:0]	F7 C000h to F7 DFFFh and F3 C000h to F3 DFFFh
	:	:
	SWP7[3:0]	F7 0000h to F7 1FFFh and F3 0000h to F3 1FFFh
BLKSWP1	SWP0[3:0]	F6 E000h to F6 FFFFh and F2 E000h to F2 FFFFh
	:	:
BLKSWP2	SWP0[3:0]	F5 E000h to F5 FFFFh and F1 E000h to F1 FFFFh
	:	:
BLKSWP3	SWP0[3:0]	F4 E000h to F4 FFFFh and F0 E000h to F0 FFFFh
	:	:
	SWP7[3:0]	F4 0000h to F4 1FFFh and F0 0000h to F0 1FFFh

Table 7.5 Corresponding Address Blocks for Each Set of Bits (Code MRAM Size: 1 Mbyte, the BLSEL.BLSWAREA bit is 1)

Register	Bits	Address Blocks to Be Swapped (Offset Address)
BLKSWP0	SWP0[3:0]	F3 E000h to F3 FFFFh and F1 E000h to F1 FFFFh
	SWP1[3:0]	F3 C000h to F3 DFFFh and F1 C000h to F1 DFFFh
	:	:
	SWP7[3:0]	F3 0000h to F3 1FFFh and F1 0000h to F1 1FFFh
BLKSWP1	SWP0[3:0]	F2 E000h to F2 FFFFh and F0 E000h to F0 FFFFh
	:	:
	SWP7[3:0]	F2 0000h to F2 1FFFh and F0 0000h to F0 1FFFh

Table 7.6 Corresponding Address Blocks for Each Set of Bits (Code MRAM Size: 512 Kbytes, the BLSEL.BLSWAREA bit is 0)

Register	Bits	Address Blocks to Be Swapped (Offset Address)
BLKSWP0	SWP0[3:0]	FB E000h to FB FFFFh and F9 E000h to F9 FFFFh
	SWP1[3:0]	FB C000h to FB DFFFh and F9 C000h to F9 DFFFh
	:	:
	SWP7[3:0]	FB 0000h to FB 1FFFh and F9 0000h to F9 1FFFh
BLKSWP1	SWP0[3:0]	FA E000h to FA FFFFh and F8 E000h to F8 FFFFh
	:	:
	SWP7[3:0]	FA 0000h to FA 1FFFh and F8 0000h to F8 1FFFh

Table 7.7 Corresponding Address Blocks for Each Set of Bits (Code MRAM Size: 512 Kbytes, the BLSEL.BLSWAREA bit is 1)

Register	Bits	Address Blocks to Be Swapped (Offset Address)
BLKSWP0	SWP0[3:0]	F9 E000h to F9 FFFFh and F8 E000h to F8 FFFFh
	SWP1[3:0]	F9 C000h to F9 DFFFh and F8 C000h to F8 DFFFh
	:	:
	SWP7[3:0]	F9 0000h to F9 1FFFh and F8 0000h to F8 1FFFh

7.2.11 Start-Up Area Setting Register (SAS)

Address(es): OFSM.SAS 86C2 E870h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTSIZE[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b31, b30	BTSIZE[1:0]	Start-Up Area Size Setting	b31 b30 0 0: Setting prohibited. 0 1: 32 Kbytes 1 0: 16 Kbytes 1 1: 8 Kbytes	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The SAS register specifies the size of start-up area 0 and start-up area 1, which are subject to the start-up program protection function.

BTSIZE[1:0] Bits (Start-Up Area Size Setting)

These bits specify the size of the area subject to the start-up program protection function.
The setting value of these bits and the address of each start-up area are shown in Table 7.8.

Table 7.8 Addresses of Start-Up Area 0 and 1

BTSIZE[1:0] Bits	SAS2.BTFLG Bit	Start-Up Area 0	Start-Up Area 1
11b	1	FFFF E000h to FFFF FFFFh	FFFF C000h to FFFF DFFFh
	0	FFFF C000h to FFFF DFFFh	FFFF E000h to FFFF FFFFh
10b	1	FFFF C000h to FFFF FFFFh	FFFF 8000h to FFFF BFFFh
	0	FFFF 8000h to FFFF BFFFh	FFFF C000h to FFFF FFFFh
01b	1	FFFF 8000h to FFFF FFFFh	FFFF 0000h to FFFF 7FFFh
	0	FFFF 0000h to FFFF 7FFFh	FFFF 8000h to FFFF FFFFh

7.2.12 Start-Up Area Setting Register 2 (SAS2)

Address(es): OFSM.SAS2 86C2 E874h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b31	BTFLG	Start-Up Area Select	0: Start-up area 1 is placed on the FFFF FFFFh side (start-up area is swapped). 1: Start-up area 0 is placed on the FFFF FFFFh side (start-up area is not swapped).	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The SAS2 register selects the start-up area used by the start-up program protection function.

BTFLG Bit (Start-Up Area Select)

This bit sets whether or not to switch the start-up area using the start-up program protection function. Program this bit to 1 when using dual mode (DUALSEL.BANKMD[2:0] = 000b)

Refer to section 65.9.3, Startup Program Protection for details.

7.2.13 Option Function Select Register 1 for Secure Application (OFS1_S), Option Function Select Register 1 for Non-Secure Application (OFS1_NS)

Address(es): OFSM.OFS1_S 86E1 07A0h, OFSM_NS.OFS1_NS 06E2 0850h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RODIS	ROWSEL[2:0]			—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HOCOFRQ0[2:0]			HOCO EN	—	—	DSBYL PSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	VDSEL[2:0]	Voltage Detection 0 Level Select	b2 b0 0 0 0: 2.85 V 0 0 1: 2.58 V 0 1 0: 2.15 V 0 1 1: 2.00 V 1 0 0: 1.90 V 1 0 1: 1.80 V 1 1 0: 1.67 V Settings other than above are prohibited when the PVDAS bit is set to 0.	R
b3	PVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitoring 0 reset is enabled after a reset. 1: Voltage monitoring 0 reset is disabled after a reset.	R
b4	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b5	DSBYLPSEL	Deep Software Standby Mode Power Reduction	Setting this bit to 0 further reduces power consumption during deep software standby modes 1 and 2.	R
b7, b6	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset. 1: HOCO oscillation is disabled after a reset.	R
b11 to b9	HOCOFRQ0[2:0]	HOCO Frequency Setting	b11 b9 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Settings other than above are prohibited.	R
b27 to b12	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b30 to b28	ROWSEL[2:0]	Reset Output Pulse Width Select	b30 b28 0 0 0: 0.25 ms 1 0 0: 0.5 ms 1 0 1: 1 ms 1 1 0: 5 ms 1 1 1: 10 ms Settings other than above are prohibited.?	R
b31	RODIS	Reset Output Enable	0: Reset output function is enabled. 1: Reset output function is disabled.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value. Once the memory type is set to OTP, it is not possible to change a bit that was set to 0 back to 1.

The OFS1 register specifies the operation of the voltage detection 0 circuit, the high-speed on-chip oscillator, and the reset output function. There are two registers: one for secure application developers (OFS1_S) and one for non-secure

application developers (OFS1_NS). Which setting is used is specified by the OFS1_SEL register.

VDSEL[2:0] Bits (Voltage Detection 0 Level Select)

These bits select the voltage detection level of the voltage detection 0 circuit.

PVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitoring 0 reset is enabled or disabled after a reset.

DSBYLPSEL Bit (Deep Software Standby Mode Power Reduction)

This bit controls power consumption during deep software standby mode 1 and 2.
When this bit is set to 0, PVD0 is used for voltage detection in the battery backup function during deep software standby mode 1 or 2. When this bit is set to 1, a dedicated voltage detector is used for voltage detection in the battery backup function during deep software standby mode 1 or 2.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation is enabled or disabled after a reset.
Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operating, reducing the oscillation stabilization wait time.
Note that even if this bit is set to 0, the system clock source is not switched to HOCO. To switch the system clock source, set the clock source select bits (SCKSCR.CKSEL[2:0]).

HOCOFRQ0[2:0] Bits (HOCO Frequency Setting)

These bits select the oscillation frequency of HOCO after a reset.

ROWSEL[2:0] Bits (Reset Output Pulse Width Select)

These bits select the minimum value of the pulse width when reset is output.

RODIS Bit (Reset Output Enable)

This bit selects whether to enable or disable the reset output function. When this bit is set to 0, the reset output function is enabled and the internal reset signal of the MCU is output from the RES# pin.

7.2.14 OFS1 Attribute Select Register (OFS1_SEL)

Address(es): OFSM.OFS1_SEL 86E1 07D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RODIS_SEL	ROWSEL_SEL[2:0]			—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HOCOFRQ0_SEL[2:0]			HOCOEN_SEL	—	—	DSBYLPSEL_SEL	—	PVDAS_SEL	VDSEL_SEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	VDSEL_SEL[2:0]	VDSEL Attribute Select	b2 b0 0 0 0: The value set in the OFS1_S.VDSEL[2:0] bits is selected. 1 1 1: The value set in the OFS1_NS.VDSEL[2:0] bits is selected. Settings other than above are prohibited.	R
b3	PVDAS_SEL	PVDAS Attribute Select	0: The value set in the OFS1_S.PVDAS bit is selected. 1: The value set in the OFS1_NS.PVDAS bit is selected.	R
b4	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b5	DSBYLPSEL_SEL	DSBYLPSEL Attribute Select	0: The value set in the OFS1_S.DSBYLPSEL bit is selected. 1: The value set in the OFS1_NS.DSBYLPSEL bit is selected.	R
b7, b6	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN_SEL	HOCOEN Attribute Select	0: The value set in the OFS1_S.HOCOEN bit is selected. 1: The value set in the OFS1_NS.HOCOEN bit is selected.	R
b11 to b9	HOCOFRQ0_SEL[2:0]	HOCOFREQ0 Attribute Select	b11 b9 0 0 0: The value set in the OFS1_S.HOCOFRQ0[2:0] bits is selected. 1 1 1: The value set in the OFS1_NS.HOCOFRQ0[2:0] bits is selected. Settings other than above are prohibited.	R
b27 to b12	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b30 to b28	ROWSEL_SEL[2:0]	ROWSEL Attribute Select	0: The value set in the OFS1_S.ROWSEL[2:0] bits is selected. 1: The value set in the OFS1_NS.ROWSEL[2:0] bits is selected.	R
b31	RODIS_SEL	RODIS Attribute Select	0: The value set in the OFS1_S.RODIS bit is selected. 1: The value set in the OFS1_NS.RODIS bit is selected.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value. Once the memory type is set to OTP, it is not possible to change a bit that was set to 0 back to 1.

The OFS1_SEL register selects whether the OFS1_S register setting or the OFS1_NS register setting is to be used. When a bit is set to 0 in this register, the setting of the corresponding bit in the OFS1_S register is used, and when a bit is set to 1 in this register, the setting of the corresponding bit in the OFS1_NS register is used.

7.2.15 MRAM Security Attribute Boundary Register (MRAMSABR)

Address(es): OFSM.MRAMSABR 86C9 F050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MRDSAB[4:0]					—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MRCSAB[8:0]									—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b13 to b5	MRCSAB[8:0]	Code MRAM Security Attribute Boundary Setting	• In linear mode When the setting value is n, the address (100 0000h – n × 8000h) (offset address) serves as the boundary of the security attributes. The area starting from this address is the secure area, and the area preceding this address is the non-secure area. Example: 000h: Address 100 0000h is the security attribute boundary (the entire area is non-secure). 001h: Address FF 8000h is the security attribute boundary. 002h: Address FF 0000h is the security attribute boundary. 010h: Address F8 0000h is the security attribute boundary (the entire area is secure in 512-Kbyte products). 01Fh: Address F0 8000h is the security attribute boundary. 020h: Address F0 0000h is the security attribute boundary (the entire area is secure in 1-Mbyte products). 1FFh: The entire area is secure. • In dual mode In addition to the above, the address (C0 0000h – n × 8000h) (offset address) also serves as the boundary for security attributes.	R
b23 to b14	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b28 to b24	MRDSAB[4:0]	Data MRAM Security Attribute Boundary Setting	When the setting value is n, the address (90 0000h + n × 800h) (offset address) serves as the boundary of the security attributes. The area preceding this address is the secure area, and the area following this address is the non-secure area. Example: 00h: Address 90 0000h is the security attribute boundary (the entire area is non-secure). 01h: Address 90 0800h is the security attribute boundary. 17h: Address 90 B800h is the security attribute boundary. 18h: Address 90 C000h is the security attribute boundary (the entire area is secure). 1Fh: The entire area is secure.	R
b31 to b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

This register is used to configure the security attribute boundary for MRAM.

MRCSAB[8:0] Bit (Code MRAM Security Attribute Boundary Setting)

These bits are used to configure the security attribute boundary of the code MRAM in 32-Kbyte units.

When the set value is n, the offset address “100 0000h – n × 8000h” serves as the boundary between the secure and non-secure areas.
For example, in the product with 1 Mbyte of code MRAM and in linear mode:
When n = 001h, addresses FFFF 8000h to FFFF FFFFh are the secure area, and addresses 7FF0 0000h to 7FFF 7FFFh are the non-secure area.
When n = 01Fh, addresses FFF0 8000h to FFFF FFFFh are the secure area, and addresses 7FF0 0000h to 7FF0 7FFFh are the non-secure area.
When n = 000h, the entire area is non-secure, and when n ≥ 020h, the entire area is secure.
In dual mode, the boundary moves by 32 Kbytes for both bank 0 and bank 1.

MRDSAB[4:0] Bit (Data MRAM Security Attribute Boundary Setting)

These bits are used to configure the security attribute boundary of the data MRAM in 2-Kbyte units.
When the set value is n, the offset address “0690 0000h + n × 800h” serves as the boundary between the secure and non-secure areas.
For example:
When n = 01h, addresses F690 0000h to F690 07FFh are the secure area, and addresses 7690 0800h to 7690 BFFFh are the non-secure area.
When n = 17h, addresses F690 0000h to F690 B7FFh are the secure area, and addresses 7690 B800h to 7690 BFFFh are the non-secure area.
When n = 00h, the entire area is non-secure, and when n ≥ 18h, the entire area is secure.

Note: Note that the relationship between setting value and increase or decrease of address differs between code MRAM and data MRAM.

7.2.16 Block Protect Register n for Secure Application (BPSn_S) (n = 0 to 15, 64 to 71), Block Protect Register n for Non-Secure Application (BPSn_NS) (n = 0 to 15, 64 to 71)

Address(es): OFSM.BPS0_S 86C9 F200h to OFSM.BPS15_S 86C9 F23Ch, OFSM.BPS64_S 86C9 F300h to OFSM.BPS71_S 86C9 F31Ch, OFSM_NS.BPS0_NS 06C9 F600h to OFSM_NS.BPS15_NS 06C9 F63Ch, OFSM_NS.BPS64_NS 06C9 F700h to OFSM_NS.BPS71_NS 06C9 F71Ch

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BP7[3:0]				BP6[3:0]				BP5[3:0]				BP4[3:0]			
Value after reset:	The value set by the user*1														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BP3[3:0]				BP2[3:0]				BP1[3:0]				BP0[3:0]			
Value after reset:	The value set by the user*1														

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BP0[3:0]	Block Protect 0	0: The corresponding address block is protected. 1: The corresponding address block is unprotected.	R
b7 to b4	BP1[3:0]	Block Protect 1		R
b11 to b8	BP2[3:0]	Block Protect 2		R
b15 to b12	BP3[3:0]	Block Protect 3		R
b19 to b16	BP4[3:0]	Block Protect 4		R
b23 to b20	BP5[3:0]	Block Protect 5		R
b27 to b24	BP6[3:0]	Block Protect 6		R
b31 to b28	BP7[3:0]	Block Protect 7		R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The BPSn register enables or disables block protection for code MRAM. There are 24 registers each for secure application developers (BPSn_S) and non-secure application developers (BPSn_NS).
Block protection can protect code MRAM in 8-Kbyte units. Use the BPSn_S registers to protect address blocks placed in the secure area of code MRAM, and use the BPSn_NS registers to protect address blocks placed in the non-secure area.

BPi[3:0] Bits (Block Protect i)

Each set of bits corresponds to an 8-Kbyte address block in code MRAM. The address block (offset address) corresponding to the bits is calculated using the following formula.

100 0000h – (n × 8 + i +1) × 2000h to FF FFFFh – (n × 8 + i) × 2000h

Setting these bits to 0000b protects the corresponding address block, preventing it from programming. Setting them to 1111b removes the protection. No other values are permitted.
To disable block protection using this register, set the MRAM.MRCBPROTNS.BPCN bit or the MRAM.MRCBPROTS.BPCN bit to 1.
Refer to section 65, MRAM for details.

Table 7.9 Corresponding Address Block for Each Set of Bits

Register	Bits	Corresponding Address Block (Offset Address)	Register	Bits	Corresponding Address Block (Offset Address)
BPS0	BP0[3:0]	FF E000h to FF FFFFh	BPS64	BP0[3:0]	BF E000h to BF FFFFh
	BP1[3:0]	FF C000h to FF DFFFh		BP1[3:0]	BF C000h to BF DFFFh
	:	:		:	:
	BP7[3:0]	FF 0000h to FF 1FFFh		BP7[3:0]	BF 0000h to BF 1FFFh
:	:	:	:	:	:
BPS7	BP7[3:0]	F8 0000h to F8 1FFFh	BPS71	BP0[3:0]	B8 E000h to B8 FFFFh
:	:	:		:	:
BPS15	BP7[3:0]	F0 0000h to F0 1FFFh		BP7[3:0]	B8 0000h to B8 1FFFh

7.2.17 Data MRAM Block Protect Register 0 for Secure Application (BPSD0_S), Data MRAM Block Protect Register 0 for Non-Secure Application (BPSD0_NS)

Address(es): OFSM.BPSD0_S 86C9 F080h, OFSM_NS.BPSD0_NS 06C9 F480h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	BP23	BP22	BP21	BP20	BP19	BP18	BP17	BP16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BP15	BP14	BP13	BP12	BP11	BP10	BP9	BP8	BP7	BP6	BP5	BP4	BP3	BP2	BP1	BP0
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	BP0	Block Protect 0	0: The corresponding address block is protected. 1: The corresponding address block is unprotected.	R
b1	BP1	Block Protect 1		R
b2	BP2	Block Protect 2		R
b3	BP3	Block Protect 3		R
b4	BP4	Block Protect 4		R
b5	BP5	Block Protect 5		R
b6	BP6	Block Protect 6		R
b7	BP7	Block Protect 7		R
b8	BP8	Block Protect 8		R
b9	BP9	Block Protect 9		R
b10	BP10	Block Protect 10		R
b11	BP11	Block Protect 11		R
b12	BP12	Block Protect 12		R
b13	BP13	Block Protect 13		R
b14	BP14	Block Protect 14		R
b15	BP15	Block Protect 15		R
b16	BP16	Block Protect 16		R
b17	BP17	Block Protect 17		R
b18	BP18	Block Protect 18		R
b19	BP19	Block Protect 19		R
b20	BP20	Block Protect 20		R
b21	BP21	Block Protect 21		R
b22	BP22	Block Protect 22		R
b23	BP23	Block Protect 23		R
b31 to b24	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. After setting a value, this register will retain the set value.

The BPSD0 register enables or disables block protection for data MRAM. There are registers for secure application developers (BPSD0_S) and non-secure application developers (BPSD0_NS).
Block protection for data MRAM can be performed in 2-Kbyte units. Use the BPSD0_S register to protect address blocks placed in the secure area of data MRAM, and use the BPSD0_NS register to protect address blocks placed in the non-secure area.

BPi[3:0] Bits (Block Protect i)

Each bit corresponds to a 2-Kbyte address block in data MRAM. The address block (offset address) corresponding to the bit is calculated using the following formula.

90 0000h + i × 800h to 90 07FFh + i × 800h

Setting this bit to 0 protects the corresponding address block, preventing it from programming. Setting it to 1 removes the protection.

For details, refer to section 65, MRAM.

7.2.18 Permanent Block Protect Register m for Secure Application (PBPSm_S) (m = 0 to 3, 16, 17), Permanent Block Protect Register m for Non-Secure Application (PBPSm_NS) (m = 0 to 3, 16, 17)

Address(es): OFSM.PBPS0_S 86E1 0A00h to OFSM.PBPS3_S 86E1 0A0Ch, OFSM.PBPS16_S 86E1 0A40h, OFSM.PBPS17_S 86E1 0A44h, OFSM_NS.PBPS0_NS 06E2 0C00h to OFSM_NS.PBPS3_NS 06E2 0C0Ch, OFSM_NS.PBPS16_NS 06E2 0C40h, OFSM_NS.PBPS17_NS 06E2 0C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

PBPSm register is used to prohibit rewriting of the BPSn register. There are six registers each for secure application developers (PBPSm_S) and non-secure application developers (PBPSm_NS). Protect the BPSn_S registers with the PBPSm_S registers, and the BPSn_NS registers with the PBPSm_NS registers.
Each bit of the PBPSm register corresponds to the BPSn.BPi[3:0] bits*1. For example, bit 0 of the PBPS0 register can prohibit rewriting of the BPS0.BP0[3:0] bits, and bit 0 of the PBPS1 register can prohibit rewriting of the BPS4.BP0[3:0] bits.

Note 1. The BPSn.BPi[3:0] bits correspond to bit j of the PBPSm register can be calculated by the following formula:
n = int ((m × 32 + j) / 8) = 4m + int (j / 8)
i = (m × 32 + j) % 8 = j % 8

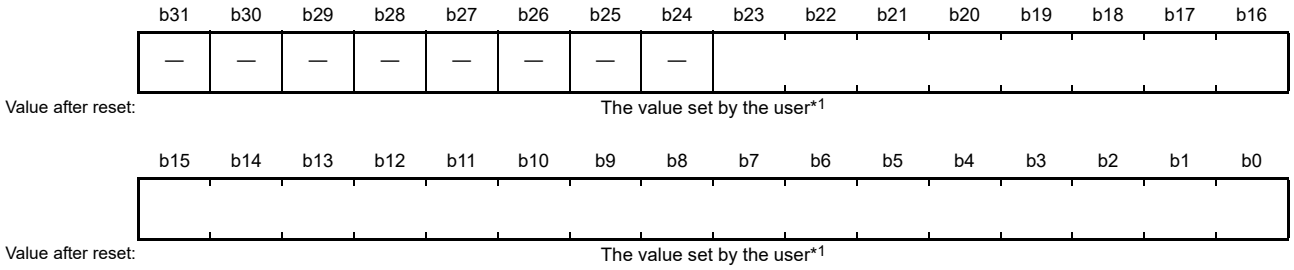
Each bit of the PBPSm register can be set to 0 only when the corresponding BPSn.BPi[3:0] bits are 0000b. Once a bit in the PBPSm register is set to 0, it can never be set back to 1, thus permanently protecting an address block in the code MRAM.
For details, refer to section 65, MRAM.

Table 7.10 Relationship Between the Bits in the PBPSm Register and the BPSn.BPi[3:0] Bits

Bit j of the PBPSm Register	Corresponding BPSn.BPi[3:0] Bits	Description
1	1111b	Program to addresses 100 0000h – (n × 8 + i +1) × 2000h to FF FFFFh – (n × 8 + i) × 2000h is enabled.
	0000b	Program to addresses 100 0000h – (n × 8 + i +1) × 2000h to FF FFFFh – (n × 8 + i) × 2000h is disabled.
0	0000b	Program to addresses 100 0000h – (n × 8 + i +1) × 2000h to FF FFFFh – (n × 8 + i) × 2000h is permanently disabled.

7.2.19 Data MRAM Permanent Block Protect Register 0 for Secure Application (PBPSD0_S), Data MRAM Permanent Block Protect Register 0 for Non-Secure Application (PBPSD0_NS)

Address(es): OFSM.PBPSD0_S 86E1 0A80h, OFSM_NS.PBPSD0_NS 06E2 0C80h



Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

PBPSD0 register is used to prohibit rewriting of the BPSD0 register. There are registers for secure application developers (PBPSD0_S) and non-secure application developers (PBPSD0_NS). Bits 24 to 31 are reserved and should be set to 1. Protect the BPSD0_S registers with the PBPSD0_S registers, and the BPSD0_NS registers with the PBPSD0_NS registers.

Each bit of the PBPSD0 register corresponds one-to-one with a bit of the BPSD0 register.

Each bit of the PBPSD0 register can be set to 0 only when the corresponding bit in the BPSD0 register is 0. Once a bit in the PBPSD0 register is set to 0, it can never be set back to 1, thus permanently protecting an address block in the data MRAM.

For details, refer to section 65, MRAM.

7.2.20 Secure MRAM Option Setting Memory Protect Register n (MOFSPSn_S) (n = 0 to 3)

Address(es): OFSM.MOFSPS0_S 86E3 0E60h to OFSM.MOFSPS3_S 86E3 0E6Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The MOFSPSn_S register disables programming of registers with the secure attribute within the option setting memory listed in Table 7.1.
Each bit of the MOFSPSn_S register corresponds to each register (including reserved areas) at addresses 86C2 E820h to 86C2 E9FCh. For example, bit 12 of the MOFSPS0_S register corresponds to the DUALSEL register, bit 20 of the MOFSPS0_S register corresponds to the SAS register, and bit 20 of the MOFSPS3_S register corresponds to the BLSEL_SEL register. Bits 24 to 31 of the MOFSPS3_S register are reserved bits.
When each bit is set to 0, the corresponding register cannot be programmed permanently.
For details, refer to section 65, MRAM.

7.2.21 Non-secure MRAM Option Setting Memory Protect Register 0 (MOFSPS0_NS)

Address(es): OFSM_NS.MOFSPS0_NS 06E3 0EE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The MOFSPS0_NS register disables programming of registers with the non-secure attribute within the option setting memory listed in Table 7.1.
Each bit of the MOFSPS0_NS register corresponds to each register (including reserved areas) at addresses 06C4 EC20h to 06C4 EC7Ch. Bit 16 of the MOFSPS0_NS register corresponds to the BANKSEL_NS register, and bit 20 of the MOFSPS0_NS register corresponds to the BLSEL_NS register. Bits 24 to 31 of the MOFSPS0_NS register are reserved bits.
When each bit is set to 0, the corresponding register cannot be programmed permanently.
For details, refer to section 65, MRAM.

7.2.22 Secure Switchable OTP Option Setting Memory Protect Register n (SOFSPSn_S) (n = 0 to 3)

Address(es): OFSM.SOFSPS0_S 86E3 0E00h to OFSM.SOFSPS3_S 86E3 0E0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The SOFSPSn_S register disables programming of switchable-OTP registers with the secure attribute within the option setting memory listed in Table 7.2.
Each bit of the SOFSPSn_S register corresponds to each register (including reserved areas) at addresses 86E1 0620h to 86E1 07FCh. For example, bit 8 of the SOFSPS0_S register corresponds to the OFS0 register, and bit 0 of the SOFSPS3_S register corresponds to the OFS1_S register. Bits 20 to 27 of the SOFSPS1_S register and bits 24 to 31 of the SOFSPS3_S register are reserved bits.
When each bit is set to 0, the corresponding register cannot be programmed permanently.
For details, refer to section 65, MRAM.

7.2.23 Non-secure Switchable OTP Option Setting Memory Protect Register 0 (SOFSPS0_NS)

Address(es): OFSM_NS.SOFSPS0_NS 06E3 0E80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The SOFSPS0_NS register disables programming of switchable-OTP registers with the non-secure attribute within the option setting memory listed in Table 7.2.
Each bit of the SOFSPS0_NS register corresponds to each register (including reserved areas) at addresses 06E2 0820h to 06E2 087Ch. Bit 12 of the SOFSPS0_NS register corresponds to the OFS1_NS register. Bits 24 to 31 of the SOFSPS0_NS register are reserved bits.
When each bit is set to 0, the corresponding register cannot be programmed permanently.
For details, refer to section 65, MRAM.

7.2.24 FSBL Control Register 0 (FSBLCTRL0)

Address(es): OFSM.FSBLCTRL0 86E0 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	FSBLCLK[2:0]			FSBLSKIPDS[2:0]			FSBLSKIPSW[2:0]			FSBLEN[2:0]		
	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FSBLEN[2:0]	FSBL Enable	b2 b0 0 0 0: FSBL is enabled. 1 1 1: FSBL is disabled. Settings other than above are prohibited.	R
b5 to b3	FSBLSKIPSW[2:0]	FSBL Skip Enable for Software Reset	b5 b3 0 0 0: Skip FSBL at software reset 1 1 1: Reboot from FSBL even at software reset Settings other than above are prohibited.	R
b8 to b6	FSBLSKIPDS[2:0]	FSBL Skip Enable for Deep Software Standby Reset	b8 b6 0 0 0: Skip FSBL at deep software standby reset 1 1 1: Reboot from FSBL even at deep software standby reset Settings other than above are prohibited.	R
b11 to b9	FSBLCLK[2:0]	FSBL Operating Clock Frequency Select	b11 b9 0 0 0: CPUCLK0 is set to 150 MHz. 0 0 1: CPUCLK0 is set to 200 MHz. 0 1 0: CPUCLK0 is set to 300 MHz. Settings other than above are prohibited. Refer to Table 7.11 for the other clock frequencies.	R
b31 to b12	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The FSBLCTRL0 register specifies the operation of the first-stage boot loader (FSBL).

FSBLEN[2:0] Bits (FSBL Enable)

These bits select whether to boot from FSBL after a reset is released.
When these bits are set to 111b, the system boots from the user extra MRAM area or the code MRAM.

FSBLSKIPSW[2:0] Bits (FSBL Skip Enable for Software Reset)

These bits select whether to reboot from FSBL or skip the FSBL after a software reset. These bits are valid only when the FSBLEN[2:0] bits are set to 000b.

FSBLSKIPDS[2:0] Bits (FSBL Skip Enable for Deep Software Standby Reset)

These bits select whether to reboot from FSBL or skip the FSBL after a deep software standby reset. These bits are valid only when the FSBLEN[2:0] bits are set to 000b.

FSBLCLK[2:0] Bits (FSBL Operating Clock Frequency Select)

These bits select the operating clock frequency when FSBL is being executed. These bits are valid only when the FSBLEN[2:0] bits are set to 000b. Table 7.11 lists the system clock frequencies for each setting value.

Table 7.11 Relationship Between the FSBLCLK[2:0] Bit Setting and System Clock Frequencies

	FSBLCLK[2:0] Bits		
System Clock	**000b**	**001b**	**010b**
CPUCLK0	150 MHz	200 MHz	300 MHz
ICLK	150 MHz	200 MHz	300 MHz
MRICLK	150 MHz	200 MHz	300 MHz
MRPCLK	75 MHz	100 MHz	150 MHz
PCLKA	150 MHz	100 MHz	150 MHz
PCLKB	75 MHz	50 MHz	75 MHz

7.2.25 FSBL Control Register 1 (FSBLCTRL1)

Address(es): OFSM.FSBLCTRL1 86E0 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSBLEXMD[1:0]	
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FSBLEXMD[1:0]	FSBL Execution Mode	b1 b0 0 0: CRC validation, no measurement report 0 1: CRC validation, measurement report 1 0: HMAC authentication, no measurement report 1 1: HMAC authentication, measurement report	R
b31 to b2	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The FSBLCTRL1 register specifies the operation when the first-stage boot loader (FSBL) verifies the integrity of the OEM boot loader (OEM_BL).

FSBLEXMD[1:0] Bits (FSBL Execution Mode)

These bits select whether FSBL validates the integrity of the OEM_BL by checking the CRC value or by performing HMAC authentication. These bits also specify whether to generate a measurement report during the verification.

7.2.26 FSBL Control Register 2 (FSBLCTRL2)

Address(es): OFSM.FSBLCTRL2 86E0 0408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PORTGN[4:0]					PORTPN[3:0]			
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PORTPN[3:0]	FSBL Error Notification Pin Select*2	b3 b0 0 0 0 0: Pm00 pin 0 0 0 1: Pm01 pin : : 1 1 1 1: Pm15 pin	R
b8 to b4	PORTGN[4:0]	FSBL Error Notification Port Select*3	b8 b4 0 0 0 0 0: PORT0 0 0 0 0 1: PORT1 : : 1 0 0 0 0: PORTG Settings other than above are prohibited.	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.
Note 2. Do not select a non-existent pin.
Note 3. Do not select a non-existent port.

The FSBLCTRL2 register selects the pin to output an error if an error occurs during execution of the first-stage boot loader (FSBL) or if the integrity of the OEM boot loader (OEM_BL) cannot be verified. To use this function, set the selected pin to the general output port.

PORTPN[3:0] Bits (FSBL Error Notification Pin Select)

These bits specify the “n” of the Pmn pin.

PORTGN[4:0] Bits (FSBL Error Notification Port Select)

These bits specify the “m” of the Pmn pin.

7.2.27 Code Certificate 0 Start Address Register n (SACC0n) (n = 0 to 3)

Address(es): OFSM.SACC00 86E0 0420h, OFSM.SACC01 86E0 0440h, OFSM.SACC02 86E0 0460h, OFSM.SACC03 86E0 0480h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The SACC0n register specifies the start address of the area where the code certificate is stored when startup area 0 is used.
This register is in the OTP area and cannot be rewritten. It should be used in order from SACC00 to SACC03. FSBL searches for a non-blank register in order from SACC03 to SACC00 and uses the first one found as a valid register.

7.2.28 Code Certificate 1 Start Address Register n (SACC1n) (n = 0 to 3)

Address(es): OFSM.SACC10 86E0 0430h, OFSM.SACC11 86E0 0450h, OFSM.SACC12 86E0 0470h, OFSM.SACC13 86E0 0490h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

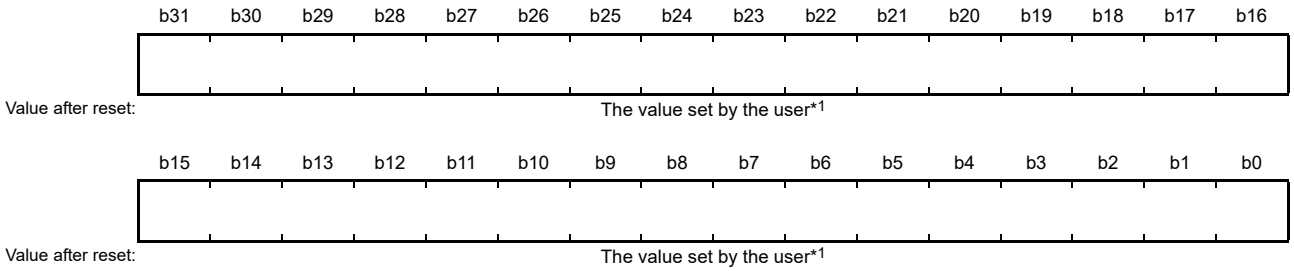
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The SACC1n register specifies the start address of the area where the code certificate is stored when startup area 1 is used.
This register is in the OTP area and cannot be rewritten. It should be used in order from SACC10 to SACC13. FSBL searches for a non-blank register in order from SACC13 to SACC10 and uses the first one found as a valid register.

7.2.29 Measurement Report Start Address Register (SAMR)

Address(es): OFSM.SAMR 86E0 0414h



Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The SAMR register specifies the start address of the area that stores the measurement report generated when the first-stage boot loader (FSBL) is executed.
Specify an address in SRAM for this register. The size of the measurement report is 68 bytes.

7.2.30 OEM_ROOT_PK Revoke Register (REVOKE)

Address(es): OFSM.REVOKE 86E3 0E10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	REVOKE3	REVOKE2	REVOKE1	REVOKE0
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	REVOKE0	OEM_ROOT_PK0 Revoke	0: OEM_ROOT_PK0 is revoked. 1: OEM_ROOT_PK0 is enabled.	R
b1	REVOKE1	OEM_ROOT_PK1 Revoke	0: OEM_ROOT_PK1 is revoked. 1: OEM_ROOT_PK1 is enabled.	R
b2	REVOKE2	OEM_ROOT_PK2 Revoke	0: OEM_ROOT_PK2 is revoked. 1: OEM_ROOT_PK2 is enabled.	R
b3	REVOKE3	OEM_ROOT_PK3 Revoke	0: OEM_ROOT_PK3 is revoked. 1: OEM_ROOT_PK3 is enabled.	R
b31 to b4	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The REVOKE register is used to revoke public root key (OEM_ROOT_PK) for the public key certificate of the OEM boot loader.
When revoking the key, revoke it in order starting from OEM_ROOT_PK3. FSBL searches for non-zero bits starting from the REVOKE3 bit, and uses the key corresponding to the first bit found.

REVOKEn Bit (OEM_ROOT_PKn Revoke) (n = 0 to 3)

This bit is used to revoke the HOEMRTPKn register.
When this bit is set to 0, 0000 0000h can be read from the corresponding HOEMRTPKn[i] registers (i = 0 to 11).

7.2.31 OEM_ROOT_PKn Hash Register [i] (HOEMRTPKn[i]) (n = 0 to 3; i = 0 to 11)

Address(es): OFSM.HOEMRTPK0[0] 86E0 0000h to OFSM.HOEMRTPK0[11] 86E0 002Ch,
OFSM.HOEMRTPK1[0] 86E0 0030h to OFSM.HOEMRTPK1[11] 86E0 005Ch,
OFSM.HOEMRTPK2[0] 86E0 0060h to OFSM.HOEMRTPK2[11] 86E0 008Ch,
OFSM.HOEMRTPK3[0] 86E0 0090h to OFSM.HOEMRTPK3[11] 86E0 00BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The HOEMRTPKn register is a 48-byte register used to store the hash value of the public root key (OEM_ROOT_PK) for the OEM boot loader public key certificate.
This register can only be written to by the boot program.

7.2.32 HUK Zeroization Register (ZHUK)

Address(es): OFSM.ZHUK 86E3 0E20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ZHUKEN3	ZHUKEN2	ZHUKEN1	ZHUKEN0
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	ZHUKEN0	Tamper 0 Zeroization Enable*2	0: Zeroize HUK when tamper 0 event is detected 1: Do not zeroize HUK when tamper 0 event is detected	R
b1	ZHUKEN1	Tamper 1 Zeroization Enable*2	0: Zeroize HUK when tamper 1 event is detected 1: Do not zeroize HUK when tamper 1 event is detected	R
b2	ZHUKEN2	Tamper 2 Zeroization Enable*2	0: Zeroize HUK when tamper 2 event is detected 1: Do not zeroize HUK when tamper 2 event is detected	R
b3	ZHUKEN3	MREZC Zeroization Enable	0: Zeroization of HUK by the MREZC register is enabled. 1: Zeroization of HUK by the MREZC register is disabled.	R
b31 to b4	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

Note 2. Set the ZHUKENn bit to the inverted value of the SYSC.TAMPCR2.ZHUKEn bit (n = 0 to 2) (Refer to section 12, Battery Backup Function (VBATTBa)). For example, when the ZHUKEN2 to ZHUKEN0 bits are 101b, set the SYSC.TAMPCR2.ZHUKE2 to ZHUKE0 bits to 010b.

The ZHUK register is used to specify whether to zeroize the hardware unique key (HUK) when a tamper event is detected or when zeroization is requested by the MREZC register.

ZHUKENn Bit (Tamper n Zeroization Enable) (n = 0 to 2)

This bit selects whether to zeroize the hardware unique key (HUK) when a tamper event is detected.
When this bit is set to 0, set the SYSC.TAMPCR2.ZHUKEn bit to 1 (zeroize HUK on tamper n event).

ZHUKEN3 Bit (MREZC Zeroization Enable)

This bit enables/disables zeroization of the hardware unique key (HUK) by writing 3C07h to the MRAM.MREZC register.

7.2.33 Anti-Rollback Counter Setting Register (ARCCS)

Address(es): OFSM.ARCCS 86E3 0E30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	ARCMD[1:0]		—	—	—	—	—	—	CNFARCNS[1:0]	
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17, b16	CNFARCNS[1:0]	ARCNS Counter Configuration Select	b17 b16 0 0: The ARCNS counter is configured as four 256-bit counters. 0 1: The ARCNS counter is configured as single 1024-bit counter. 1 0: Setting prohibited. 1 1: The ARCNS counter is disabled.	R
b23 to b18	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b25, b24	ARCMD[1:0]	ARC Usage Mode Select	b25 b24 0 0: Mode 2 (ARCS usage mode) 0 1: Mode 1 (ARCOEMBL usage mode) 1 0: Setting prohibited. 1 1: All anti-rollback counters are not used.	R
b31 to b26	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

The ARCCS register is used to set the anti-rollback counter (ARC).

CNFARCNS[1:0] Bits (ARCNS Counter Configuration Select)

These bits are used to enable/disable the anti-rollback counter for non-secure application (ARCNS) and to select the counter configuration.

When these bit are set to 00b, the ARCNS is divided into four 256-bit counters (ARCNS0 to ARCNS3). When these bits are set to 01b, the ARCNS operates as a 1024-bit counter.

When these bit are set to 11b, the ARCNS is disabled. Do not issue the increment counter command or read counter command to the ARCNS.

ARCMD[1:0] Bits (ARC Usage Mode Select)

These bits are used to select the anti-rollback counter for use in version management by the boot program.

Set these bits to 01b when the OEM boot loader is to be used to validate the application program. Set them to 00b when the first-stage boot loader (FSBL) is to be used to validate the application program.

If they are set to 11b, the anti-rollback counters are not incremented even if the increment counter command is issued.

7.2.34 Anti-Rollback Counter for Secure Application [i] (ARCS[i]) (i = 0 to 31)

Address(es): OFSM.ARCS[0] 86F4 1D00h to OFSM.ARCS[31] 86F4 1D7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is 0000 0000h. When an increment counter command is issued, the least significant bit set to 0 becomes 1. Once a bit is set to 1, it cannot be set back to 0.

The ARCS register is a 1024-bit anti-rollback counter for secure application. The most significant 32 bits are stored in the ARCS[31] register, and the least significant 32 bits are stored in the ARCS[0] register.
When an increment counter command is issued to the ARCS register, the least significant bit set to 0 is set to 1. After all bits are set to 1, the value does not change even if an increment counter command is issued.

7.2.35 Anti-Rollback Counter n for Non-Secure Application [i] (ARCNSn[i]) (n = 0 to 3, i = 0 to 7)

Address(es): OFSM.ARCNS0[0] 86F4 1E00h to OFSM.ARCNS0[7] 86F4 1E1Ch,
OFSM.ARCNS1[0] 86F4 1E20h to OFSM.ARCNS1[7] 86F4 1E3Ch,
OFSM.ARCNS2[0] 86F4 1E40h to OFSM.ARCNS2[7] 86F4 1E5Ch,
OFSM.ARCNS3[0] 86F4 1E60h to OFSM.ARCNS3[7] 86F4 1E7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

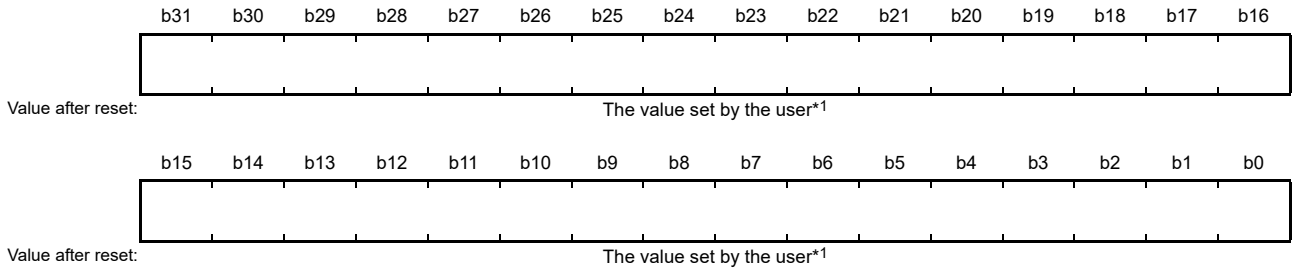
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value for a blank product is 0000 0000h. When an increment counter command is issued, the least significant bit set to 0 becomes 1. Once a bit is set to 1, it cannot be set back to 0.

The ARCNSn registers are 256-bit anti-rollback counters for non-secure applications. This configuration is enabled when the ARCCS.CNFARCNS[1:0] bits are set to 00b. The most significant 32 bits are stored in the ARCNS[7] register, and the least significant 32 bits are stored in the ARCNS[0] register.
When an increment counter command is issued to the ARCNSn register, the least significant bit set to 0 is set to 1. After all bits are set to 1, the value does not change even if an increment counter command is issued.

7.2.36 Anti-Rollback Counter for Non-Secure Application [i] (ARCNS[i]) (i = 0 to 31)

Address(es): OFSM.ARCNS[0] 86F4 1E00h to OFSM.ARCNS[31] 86F4 1E7Ch

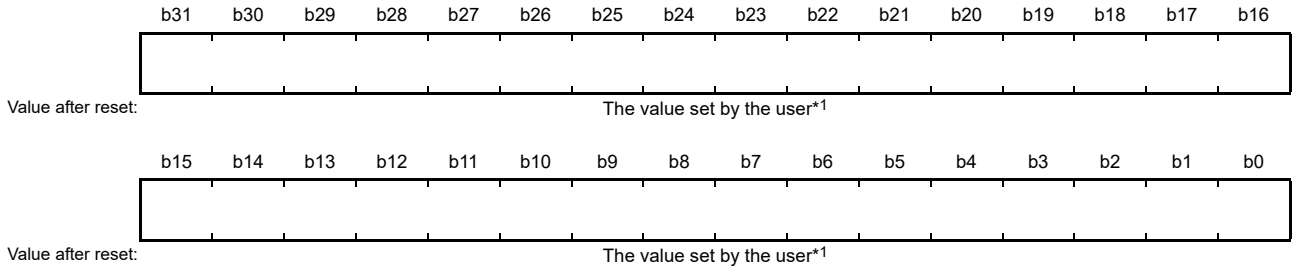


Note 1. The value for a blank product is 0000 0000h. When an increment counter command is issued, the least significant bit set to 0 becomes 1. Once a bit is set to 1, it cannot be set back to 0.

The ARCNS registers are 1024-bit anti-rollback counters for non-secure applications. This configuration is enabled when the ARCCS.CNFARCNS[1:0] bits are set to 01b. The most significant 32 bits are stored in the ARCNS[31] register, and the least significant 32 bits are stored in the ARCNS[0] register.
When an increment counter command is issued to the ARCNS register, the least significant bit set to 0 is set to 1. After all bits are set to 1, the value does not change even if an increment counter command is issued.

7.2.37 Anti-Rollback Counter for OEM Boot Loader [i] (ARCOEMBL[i]) (i = 0, 1)

Address(es): OFSM.ARCOEMBL[0] 86F4 1F00h, OFSM.ARCOEMBL[1] 86F4 1F04h



Note 1. The value for a blank product is 0000 0000h. When an increment counter command is issued, the least significant bit set to 0 becomes 1. Once a bit is set to 1, it cannot be set back to 0.

The ARCOEMBL register is a 64-bit anti-rollback counter for OEM boot loader. The upper 32 bits are stored in the ARCOEMBL[1] register, and the lower 32 bits are stored in the ARCOEMBL[0] register.
When an increment counter command is issued to the ARCOEMBL register, the least significant bit set to 0 is set to 1. After all bits are set to 1, the value does not change even if an increment counter command is issued.

7.2.38 User Extra MRAM Configuration Register (UMRECR)

Address(es): OFSM.UMRECR 86E3 0E50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCDCCTL
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	UMREEN	—	—	—	—	—	—	—	UMREMS
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	UMREMS	User Extra MRAM Area Memory Type Select	0: Memory type of the user extra MRAM area is OTP. 1: Memory type of the user extra MRAM area is MRAM.	R
b7 to b1	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	UMREEN	User Extra MRAM Enable	0: Execute a program in the user extra MRAM area 1: Do not execute a program in the user extra MRAM area	R
b15 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b16	DCDCCTL	DC-DC Converter Control	0: DC-DC converter is disabled after a reset (External VDD mode). 1: DC-DC converter is enabled after a reset.	R
b31 to b17	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value for a blank product is FFFF FFFFh. A bit programmed to 0 becomes 0. Once a bit is set to 0, it cannot be set back to 1.

UMREMS Bit (User Extra MRAM Area Memory Type Select)

This bit selects the memory type of the user extra MRAM area from either MRAM or OTP. When 1 (MRAM) is selected, the user extra MRAM area can be rewritten any number of times within the range specified in the electrical characteristics. When 0 (OTP) is selected, the user extra MRAM area can be programmed only once and cannot be rewritten after it has been written once.

UMREEN Bit (User Extra MRAM Enable)

This bit selects whether to execute the program in the user extra MRAM area when a reset is released.
When the FSBLEN[2:0] bits are 111b and this bit is 0, the system boots from the user extra MRAM. When the FSBLEN[2:0] bits are 111b and this bit is 1, the system boots from the code MRAM.

DCDCCTL Bit (DC-DC Converter Control)

This bit selects whether to operate the DC-DC converter after a reset.
When this bit is set to 1, the MCU starts up in DC-DC mode after a reset. When this bit is set to 0 the MCU starts up in external VDD mode after a reset.

7.3 Setting the Option-Setting Memory

7.3.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Table 7.1 and Table 7.2. The method of data allocation (source code syntax) differs depending on the compiler. Refer to the compiler manual for details.

7.3.2 Programming to Option-Setting Memory

To write the data allocated above into the option-setting memory, use one of the following methods.

(1) Rewriting by Self-Programming

Use the configuration setting command to write data to the option-setting memory in the extra MRAM area (refer to Table 7.1). Use the program command to write data to registers except the HOEMRTPKn register (n = 0 to 3) and the anti-rollback counter in the option-setting memory of the OTP area (refer to Table 7.2).
The anti-rollback counter can only be updated by the increment counter command.
For details of the configuration setting command, program command, and increment counter command, refer to section 65, MRAM.

(2) Rewriting by Serial Programmer or Emulator

The specific procedure varies depending on the tool used. For details, refer to the instruction manual of the tool used.
This MCU provides the following two setting procedures.

- The data allocated as described in section 7.3.1 above is taken from the object file or Motorola S-format file generated by the compiler and programmed into the MCU.
- Set values directly into each register using the tool's GUI interface.

7.3.3 Security Attributes of Option-Setting Memory

The option-setting memory contains registers with secure attribute and registers with non-secure attribute (refer to Table 7.1 and Table 7.2).
Registers with secure attribute can only be read from a secure application. Attempting to read from a non-secure application will result in a Protect Zone access error. Registers with non-secure attribute can be read from any application.
The conditions for rewriting the option setting memory vary depending on the rewriting method.

(1) Rewriting by Self-Programming

Registers with secure attribute can only be rewritten from a secure application. Issuing a commands from a non-secure application will result in a Protect Zone access error. Registers with the non-secure attribute can be rewritten from any application.

(2) Rewriting by Serial Programmer or Emulator

Rewriting registers with secure attribute requires Secure Application Developer privileges (Authentication Level 2 (AL2)); rewriting registers with non-secure attribute requires Non-secure Application Developer or higher privileges (Authentication Level 1 or 2 (AL1 or AL2)).

7.3.4 Registers with Selectable Security Attributes for Each Bit

There are three sets of registers in the option-setting memory, consisting of three registers for non-secure, secure, and attribute selection, as shown below.

- Bank select register (BANKSEL_NS, BANKSEL_S, BANKSEL_SEL)
- Block select register (BLSEL_NS, BLSEL_S, BLSEL_SEL)
- Option function select register 1 (OFS1_NS, OFS1_S, OFS1_SEL)

These registers can be selected to use the setting for non-secure or secure for each bit.
As shown in the example in Figure 7.1, when a bit in the attribute select register is set to 0, the corresponding bit in the secure register is used, and when set to 1, the corresponding bit in the non-secure register is used.
For example, in the case of the OFS1 register, if the reset output and HOCO settings are to be controlled from the non-secure application and the PVD0 setting from the secure application, the OFS1_SEL register must be set to F000 0F00h.

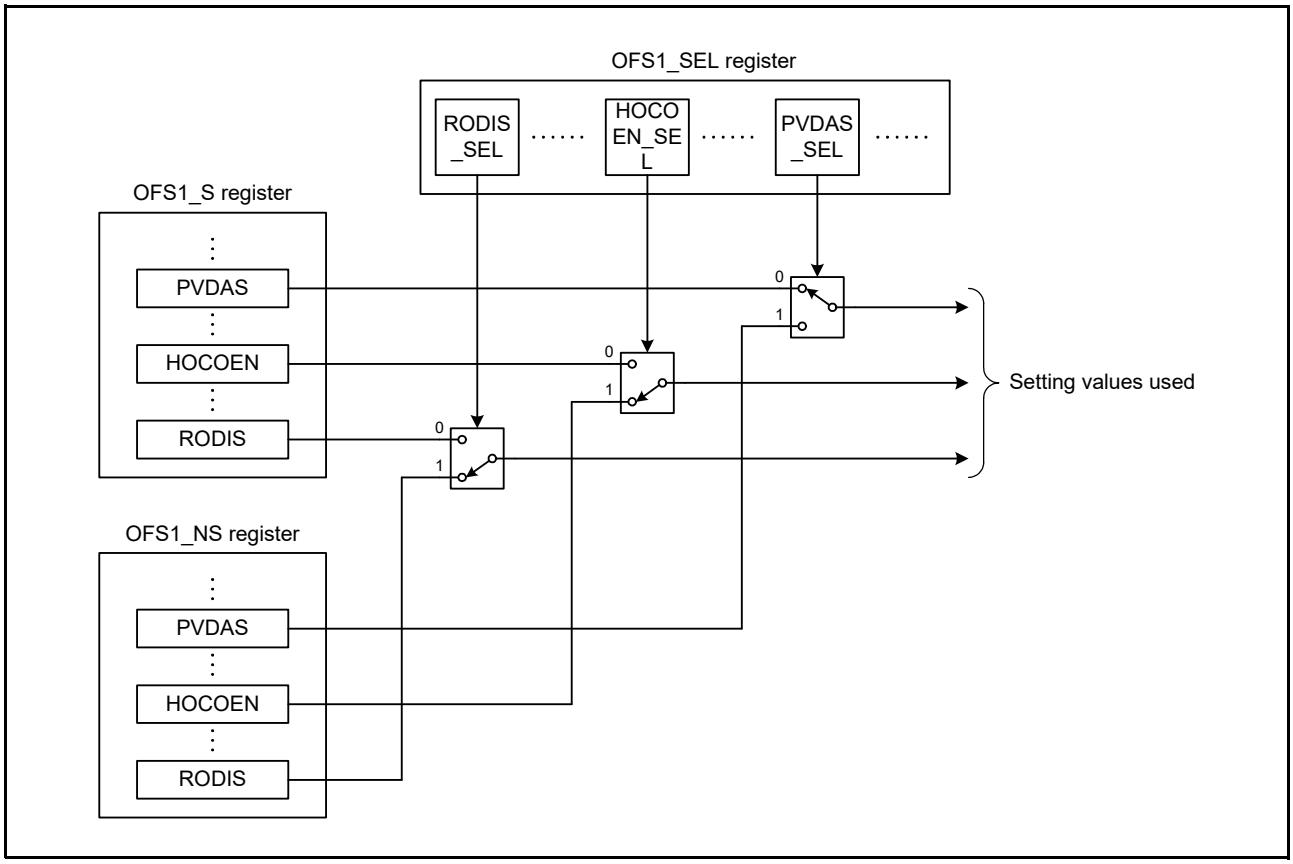


Figure 7.1 Selection of applied value

7.3.5 Reflection Timing of the Setting Value

When the SAS, BPS_NS, BPS_S, PBPS_NS, and PBPS_S registers are rewritten, the set values are immediately reflected in each circuit. When other registers are rewritten, the set values are reflected after the MCU is reset.

7.4 Usage Note

7.4.1 Set Value for Reserved Area

(1) Attribute select register area (86C2 E9E0h to 86C2 E9FFh)

When issuing a program command to the reserved registers or reserved bits in this area, write 0000 0000h or 0, respectively. If 1 is programmed, subsequent operation is not guaranteed.

(2) Areas other than the above

When issuing a program command to the reserved registers or reserved bits in this area, write FFFF FFFFh or 1, respectively. If 0 is programmed, subsequent operation is not guaranteed.

7.4.2 Notes on the First-Stage Boot Loader (FSBL)

Setting the FSBLCTRL0.FSBLEN[2:0] bits to 000b causes FSBL to be executed after the system reset is released. At this time, setting the FSBLSKIPSW[2:0] and the FSBLSKIPDS[2:0] bits to 000b allows the FSBL to skip execution after software reset and deep software standby reset, respectively.
However, after a software reset or deep software standby reset, if any reset detection flag other than the software reset detection flag and the deep software standby reset detection flag is set to 1, the FSBL is executed without being skipped.
When using this function, be sure to clear the RSTSR0, RSTSR1, and RSTSR3 registers before executing a software reset or before transitioning to deep software standby mode.

8. Programmable Voltage Detection Circuit (PVD)

8.1 Overview

The programmable voltage detection (PVD) module monitors the voltage input to the VCC pin. The detection levels are selectable by software.
The PVD module consists of several voltage detectors which measure the voltage being input to the VCC pin. Detection of a change in the input voltage on the VCC pin can be set with various voltage thresholds by registers of the PVD module.
Each of the voltage detectors has an associated voltage monitoring circuit. The voltage monitoring registers can set the PVD to trigger an interrupt, event link output, or reset when the threshold is exceeded.
Table 8.1 lists the specifications of the PVD module. Figure 8.1 is a block diagram of the voltage monitoring 0 reset circuit. Figure 8.2 and Figure 8.3 are block diagrams of the voltage monitoring m interrupt/reset circuit (m = 1, 2) and the voltage monitoring n reset circuit (n = 4, 5), respectively.
Each PVD circuit consists of a low voltage detector (LVD) and a high voltage detector (HVD). The LVD detects VCC falling below the voltage threshold, and the HVD detects VCC rising above the voltage threshold.
The PVDn circuits are implemented for security and their available functions are limited in comparison with those of the PVDm circuits.
The PVDn circuits have a register locking function for security.

A separate voltage detector for the core voltage is also implemented. It monitors the voltage input to VDD and the detection level is predefined in hardware.
The voltage detector for the core voltage has a corresponding voltage monitoring circuit. The voltage monitoring circuit for the core voltage can generate a reset (core voltage monitoring reset) when the core voltage crosses respective thresholds to become less than Vdet_VDDL or to exceed Vdet_VDDH.
The core voltage monitoring reset (CVMR) function is enabled when OFS2.CVMRCTL = 0.

Table 8.1 PVD Specifications (1/2)

Item		Programmable Voltage Monitoring 0	Programmable Voltage Monitoring m (m = 1, 2)	Programmable Voltage Monitoring n (n = 4, 5)
Voltage monitoring	Monitored voltage	VCC	VCC	VCC
	Detection event	Vdet0 > VCC	Voltage rises or falls past Vdetm	Voltage rises or falls past Vdetn
	Detection voltage	Vdet0 is selectable from among 7 different levels by using the OFS1.VDSEL[2:0] bits	Vdetm is selectable from among 12 different levels by using the PVDmCMPCR.PVDLVL[4:0] bits	Vdetn is selectable from among 12 different levels by using the PVDnCMPCR.PVDLVL[4:0] bits
	Monitoring flag	None	PVDmSR.MON flag: Monitors whether voltage is higher or lower than Vdetm	None
			PVDmSR.DET flag: Vdetm crossing detection	

Table 8.1 PVD Specifications (2/2)

Item		Programmable Voltage Monitoring 0	Programmable Voltage Monitoring m (m = 1, 2)	Programmable Voltage Monitoring n (n = 4, 5)
Process upon voltage detection	Reset	Voltage monitoring 0 reset	Voltage monitoring m reset	Voltage monitoring n reset
		Reset when Vdet0 > VCC CPU operation restarts after the specified time with VCC > Vdet0	Reset condition selectable: Reset when Vdetm > VCC (LVD) or Vdetm < VCC (HVD) CPU restart timing selectable: after the specified time with LVD or HVD	Reset condition selectable: Reset when Vdetn > VCC (LVD) or Vdetn < VCC (HVD)
	Interrupt	None	Voltage monitoring m interrupt	None
			Non-maskable interrupt or maskable interrupt selectable	
			Interrupt request issued in response to either or both LVD and HVD	
Digital filter	Enable/Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/s LOCO frequency × 2 (s: 2, 4, 8, 16)	1/s LOCO frequency × 2 (s: 2, 4, 8, 16)
Event linking		None	Output of event signals on detection of Vdetm crossings	None
Register locking function		None	None	PVDLR register

Table 8.2 CVM Specifications

Item		Core Voltage Monitoring
Voltage monitoring	Monitored voltage	VDD
	Detection event	Voltage falls past Vdet_VDDL Voltage rises past Vdet_VDDH
	Detection voltage	Predetermined level*1
Process upon voltage detection	Reset	Core voltage monitoring reset*2
		Reset when Vdet_VDDL > VDD or Vdet_VDDH < VDD CPU operation restarts after the specified time with Vdet_VDDL ≤ VDD ≤ Vdet_VDDH

Note 1. This refers to Vdet_VDDL and Vdet_VDDH.
Note 2. For the core voltage monitoring reset, see section 6, Resets.

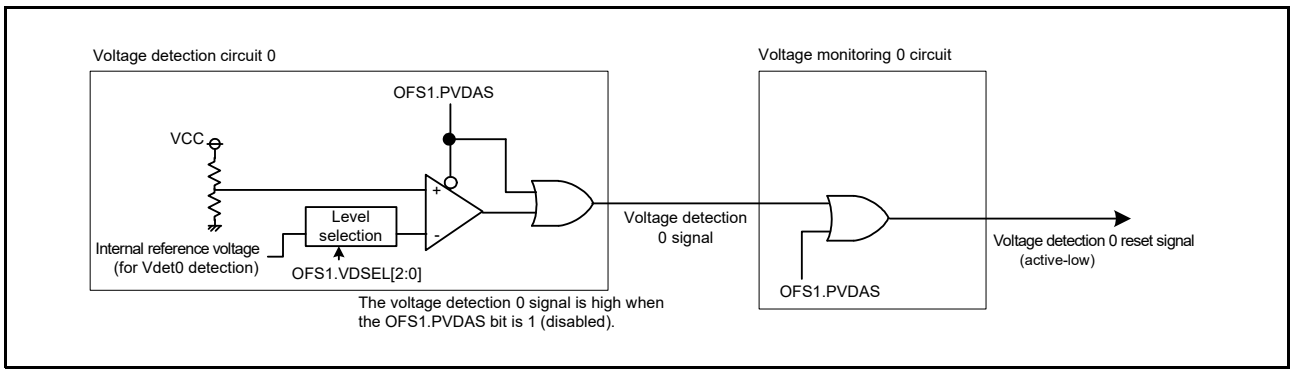


Figure 8.1 Block Diagram of Voltage Monitoring 0 Reset Circuit

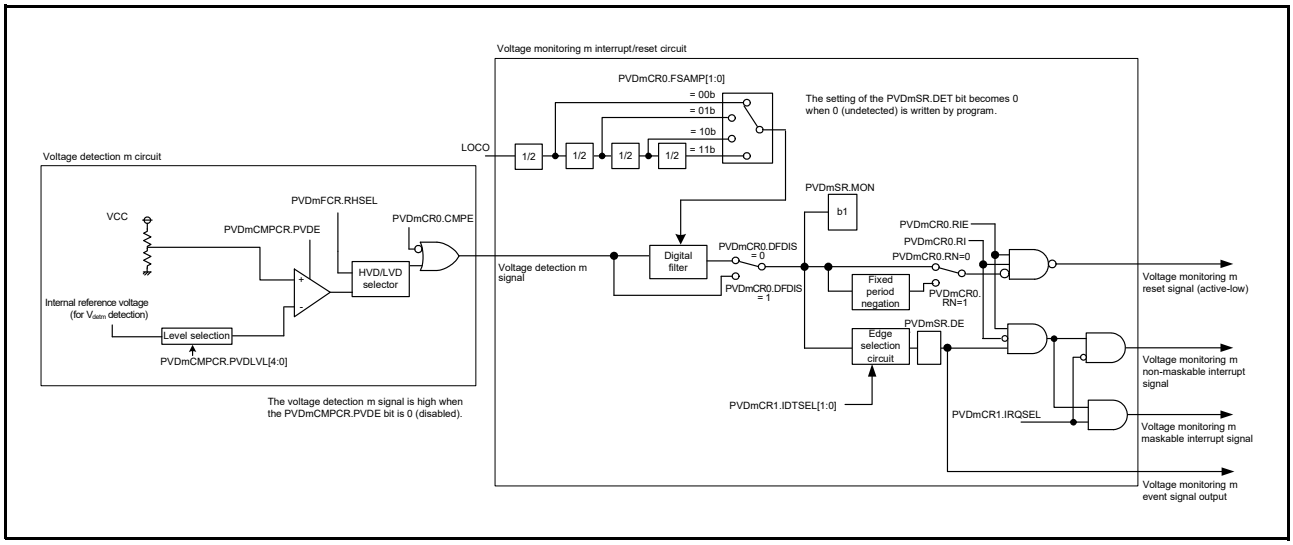


Figure 8.2 Block Diagram of Voltage Monitoring m Interrupt/Reset Circuit (m = 1, 2)

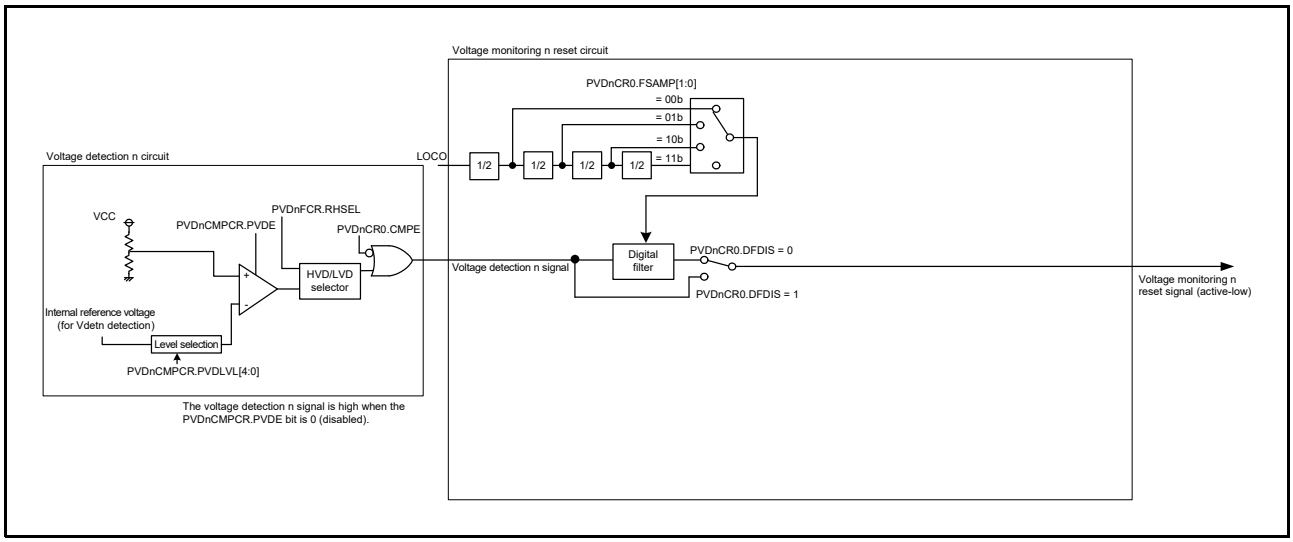


Figure 8.3 Block Diagram of Voltage Monitoring n Reset Circuit (n = 4, 5)

8.2 Register Descriptions

8.2.1 Programmable Voltage Detection Security Attribute Setting Register (PVDSAR)

Address(es): SYSC.PVDSAR 8701 93CCh, SYSC_NS.PVDSAR 0701 93CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NONSEC0	PVD1 Security Attribute Setting	Target registers: Registers for PVD1 0: Secure 1: Non-secure	R/W
b1	NONSEC1	PVD2 Security Attribute Setting	Target registers: Registers for PVD2 0: Secure 1: Non-secure	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: Set the PRCR.PRC4 bit to 1 (enabling writing) before writing to this register.

This register controls the secure attribute of PVD registers.

NONSEC0 Bit (PVD1 Security Attribute Setting)

This bit controls the security attribute of the PVD1CMPCR, PVD1CR0, PVD1CR1, PVD1SR, and PVD1FCR registers.

NONSEC1 Bit (PVD2 Security Attribute Setting)

This bit controls the security attribute of the PVD2CMPCR, PVD2CR0, PVD2CR1, PVD2SR, and PVD2FCR registers.

8.2.2 Voltage Monitoring m Comparator Control Register (PVDmCMPCR) (m = 1, 2)

Address(es): SYSC.PVD1CMPCR 8701 9A58h, SYSC.PVD2CMPCR 8701 9A5Ch,
SYSC_NS.PVD1CMPCR 0701 9A58h, SYSC_NS.PVD2CMPCR 0701 9A5Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	PVDE	—	—	PVDLVL[4:0]				
Value after reset:	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PVDLVL[4:0]	Voltage Detection m Level Select	b4 b0 0 0 1 0 0: 3.14 V (Vdetm_4) 0 0 1 0 1: 3.10 V (Vdetm_5) 0 0 1 1 0: 3.08 V (Vdetm_6) 0 0 1 1 1: 2.85 V (Vdetm_7) 0 1 0 0 0: 2.83 V (Vdetm_8) 0 1 0 0 1: 2.80 V (Vdetm_9) 0 1 0 1 0: 2.62 V (Vdetm_10) 0 1 0 1 1: 2.33 V (Vdetm_11) 0 1 1 0 0: 1.90 V (Vdetm_12) 0 1 1 0 1: 1.86 V (Vdetm_13) 0 1 1 1 0: 1.74 V (Vdetm_14) 0 1 1 1 1: 1.71 V (Vdetm_15) Others: Setting prohibited These are standard voltage levels at voltage drop. For the standard voltage levels at voltage rise, see section 67, Electrical Characteristics.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	PVDE	Voltage Detection m Enable	0: Voltage detection m circuit disabled 1: Voltage detection m circuit enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.

PVDLVL[4:0] Bits (Voltage Detection m Level Select)

These bits can only be changed when all PVDmCMPCR.PVDE bits and all PVDnCMPCR.PVDE bits are 0. Do not set more than one among PVD1, PVD2, PVD4, and PVD5 to the same detection voltage.
Do not change these bits and the PVDmCMPCR.PVDE bit at the same time.

PVDE Bit (Voltage Detection m Enable)

When using the voltage detection m interrupt/reset or PVDmSR.MON flag, set this bit to 1. The voltage detection m circuit starts when td(E-A) has elapsed after the value of this bit has been changed from 0 to 1.
This bit can only be changed when the PVDmCR0.CMPE bit is 0.
When using the voltage detection m circuit in deep software standby mode, do not allow a transition to deep software standby mode 3.

8.2.3 Voltage Monitoring n Comparator Control Register (PVDnCMPCR) (n = 4, 5)

Address(es): SYSC.PVD4CMPCR 8701 9A64h, SYSC.PVD5CMPCR 8701 9A68h

	b7	b6	b5	b4	b3	b2	b1	b0
	PVDE	—	—	PVDLVL[4:0]				
Value after reset:	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PVDLVL[4:0]	Voltage Detection n Level Select	b4 b0 0 0 1 0 0: 3.14 V (Vdetm_4) 0 0 1 0 1: 3.10 V (Vdetm_5) 0 0 1 1 0: 3.08 V (Vdetm_6) 0 0 1 1 1: 2.85 V (Vdetm_7) 0 1 0 0 0: 2.83 V (Vdetm_8) 0 1 0 0 1: 2.80 V (Vdetm_9) 0 1 0 1 0: 2.62 V (Vdetm_10) 0 1 0 1 1: 2.33 V (Vdetm_11) 0 1 1 0 0: 1.90 V (Vdetm_12) 0 1 1 0 1: 1.86 V (Vdetm_13) 0 1 1 1 0: 1.74 V (Vdetm_14) 0 1 1 1 1: 1.71 V (Vdetm_15) Others: Setting prohibited These are standard voltage levels at voltage drop. For the standard voltage levels at voltage rise, see section 67, Electrical Characteristics.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	PVDE	Voltage Detection n Enable	0: Voltage detection n circuit disabled 1: Voltage detection n circuit enabled	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.

PVDLVL[4:0] Bits (Voltage Detection n Level Select)

These bits can only be changed when all PVDmCMPCR.PVDE bits and all PVDnCMPCR.PVDE bits are 0. Do not set more than one among PVD1, PVD2, PVD4, and PVD5 to the same detection voltage.
Do not change these bits and the PVDnCMPCR.PVDE bit at the same time.

PVDE Bit (Voltage Detection n Enable)

When using the voltage detection n reset, set this bit to 1. The voltage detection n circuit starts when td(E-A) has elapsed after the value of this bit has been changed from 0 to 1.
This bit can only be changed when the PVDnCR0.CMPE bit is 0.
When using the voltage detection n circuit in deep software standby mode, do not allow a transition to deep software standby mode 3.

8.2.4 Voltage Monitoring m Circuit Control Register 0 (PVDmCR0) (m = 1, 2)

Address(es): SYSC.PVD1CR0 8701 9A70h, SYSC.PVD2CR0 8701 9A74h,
SYSC_NS.PVD1CR0 0701 9A70h, SYSC_NS.PVD2CR0 0701 9A74h

	b7	b6	b5	b4	b3	b2	b1	b0
	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RIE	Voltage Monitoring m Interrupt/ Reset Enable	0: Disabled 1: Enabled	R/W
b1	DFDIS	Voltage Monitoring m Digital Filter Disable	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	CMPE	Voltage Monitoring m Circuit Comparison Result Output Enable	0: Voltage monitoring m circuit comparison result output disabled 1: Voltage monitoring m circuit comparison result output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5, b4	FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	RI	Voltage Monitoring m Circuit Mode Select	(1) When PVDmFCR.RHSEL = 0 0: Voltage monitoring m interrupt is generated whenever the voltage crosses Vdetm. 1: Voltage monitoring m reset is generated when the voltage crosses Vdetm while falling. (2) When PVDmFCR.RHSEL = 1 0: Setting prohibited 1: Voltage monitoring m reset is generated whenever the voltage crosses Vdetm while rising	R/W
b7	RN	Voltage Monitoring m Reset Negate Select	(1) When PVDmFCR.RHSEL = 0 0: Negation follows a stabilization time (tPVDm) after VCC > Vdetm is detected. 1: Negation follows a stabilization time (tPVDm) after assertion of the PVDm reset. (2) When PVDmFCR.RHSEL = 1 0: Negation follows a stabilization time (tPVDm) after VCC < Vdetm is detected. 1: Setting prohibited	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.

RIE Bit (Voltage Monitoring m Interrupt/Reset Enable)

The RIE bit enables or disables the corresponding voltage monitoring interrupt and voltage monitoring m reset. Ensure that neither the voltage monitoring m interrupt nor voltage monitoring m reset is generated during programming of the MRAM.

DFDIS Bit (Voltage Monitoring m Digital Filter Disable)

The DFDIS bit enables or disables the digital filter. When the DFDIS bit is 0 (digital filter enabled), set the LOCOCR.LCSTP bit to 0 (the LOCO is operating). When using the voltage monitoring m circuit in software standby or

deep software standby mode, set the DFDIS bit to 1 (digital filter disabled).

FSAMP[1:0] Bits (Sampling Clock Select)

The FSAMP[1:0] bits can only be modified when the PVDmCR0.DFDIS bit is 1 (digital filter disabled).
Do not modify the FSAMP[1:0] bits when the PVDmCR0.DFDIS bit is 0 (digital filter enabled).

RI Bit (Voltage Monitoring m Circuit Mode Select)

When the PVDmCR0.RI bit is 1 (voltage monitoring m reset selected), a transition to deep software standby mode 2 or 3 is not possible. In this case, the device makes the transition to deep software standby mode 1. To allow a transition to deep software standby mode 2 or 3, set the PVDmCR0.RI bit to 0 (voltage monitoring m interrupt selected).

RN Bit (Voltage Monitoring m Reset Negate Select)

- When PVDmFCR.RHSEL = 0

When setting the RN bit to 1 (negation follows a stabilization time after assertion of the PVDm reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO is operating). Furthermore, if a transition to software standby or deep software standby mode is to be made, the only possible value for the RN bit is 0 (negation follows a stabilization time after VCC > Vdetm is detected). Do not set the RN bit to 1 when this is the case.

- When PVDmFCR.RHSEL = 1

Do not set the RN bit to 1.

8.2.5 Voltage Monitoring n Circuit Control Register 0 (PVDnCR0) (n = 4, 5)

Address(es): SYSC.PVD4CR0 8701 9A7Ch, SYSC.PVD5CR0 8701 9A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FSAMP[1:0]		—	CMPE	DFDIS	RE
Value after reset:	0	1	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RE	Voltage Monitoring n Reset Enable	0: Disabled 1: Enabled	R/W
b1	DFDIS	Voltage Monitoring n Digital Filter Disable	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	CMPE	Voltage Monitoring n Circuit Comparison Result Output Enable	0: Voltage monitoring n circuit comparison result output disabled 1: Voltage monitoring n circuit comparison result output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5, b4	FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.

RE Bit (Voltage Monitoring n Reset Enable)

The RE bit enables or disables the corresponding voltage monitoring n reset. Ensure that the voltage monitoring n reset is not generated during programming of the MRAM.

DFDIS Bit (Voltage Monitoring n Digital Filter Disable)

The DFDIS bit enables or disables the digital filter. When the DFDIS bit is 0 (digital filter enabled), set the LOCOCR.LCSTP bit to 0 (the LOCO is operating). When using the voltage monitoring n circuit in software standby or deep software standby mode, set the DFDIS bit to 1 (digital filter disabled).

FSAMP[1:0] Bits (Sampling Clock Select)

The FSAMP[1:0] bits can only be modified when the PVDnCR0.DFDIS bit is 1 (digital filter disabled).
Do not modify the FSAMP[1:0] bits when the PVDnCR0.DFDIS bit is 0 (digital filter enabled).

8.2.6 Voltage Monitoring m Circuit Control Register 1 (PVDmCR1) (m = 1, 2)

Address(es): SYSC.PVD1CR1 8701 90E0h, SYSC.PVD2CR1 8701 90E2h, SYSC_NS.PVD1CR1 0701 90E0h, SYSC_NS.PVD2CR1 0701 90E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitoring m Interrupt Generation Condition Select	(1) When PVDmFCR.RHSEL = 0 0 0: An interrupt is generated when VCC ≥ Vdetm (rise) is detected. 0 1: An interrupt is generated when VCC < Vdetm (drop) is detected. 1 0: An interrupt is generated when a voltage drop or rise is detected. 1 1: Settings prohibited (2) When PVDmFCR.RHSEL = 1 These bits are not applied.	R/W
b2	IRQSEL	Voltage Monitoring m Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.PVDmEN bit in the ICU from the reset state.

8.2.7 Voltage Monitoring m Circuit Status Register (PVDmSR) (m = 1, 2)

Address(es): SYSC.PVD1SR 8701 90E1h, SYSC.PVD2SR 8701 90E3h, SYSC_NS.PVD1SR 0701 90E1h, SYSC_NS.PVD2SR 0701 90E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	DET	Voltage Monitoring m Voltage Change Detection Flag*1	(1) When PVDmFCR.RHSEL = 0 0: Undetected 1: Vdetm crossing is detected. (2) When PVDmFCR.RHSEL = 1 This bit is not applied.	R/W
b1	MON	Voltage Monitoring m Signal Monitor Flag	(1) When PVDmFCR.RHSEL = 0 0: VCC ≤ Vdetm 1: VCC > Vdetm or MON is disabled. (2) When PVDmFCR.RHSEL = 1 This bit is not applied.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC3 bit to 1 (enabling writing) before writing to this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, two cycles of the system clock (ICLK) are taken until the bit is read as 0.

DET Flag (Voltage Monitoring m Voltage Change Detection Flag)

The DET flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitoring m circuit comparison result output enabled).
When setting the DET flag to 0, make sure that the PVDmCR0.RIE bit is set to 0 (voltage monitoring m interrupt/reset disabled).
When setting the RIE bit to 1 again after setting it to 0, wait for at least two cycles of PCLKB. Depending on the number of cycles of PCLKB defined for read access to I/O registers, at least two cycles of PCLKB may have to be secured as waiting time.

MON Flag (Voltage Monitoring m Signal Monitor Flag)

The MON flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitoring m circuit comparison result output enabled).

8.2.8 Voltage Monitoring m Function Control Register (PVDmFCR) (m = 1, 2)

Address(es): SYSC.PVD1FCR 8701 9B20h, SYSC.PVD2FCR 8701 9B24h,
SYSC_NS.PVD1FCR 0701 9B20h, SYSC_NS.PVD2FCR 0701 9B24h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RHSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RHSEL	Rising Hysteresis Select	0: Hysteresis level for VCC drop detection selected 1: Hysteresis level for VCC rise detection selected	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The PVDmFCR.RHSEL bit can only be changed when all PVDmCMPCR.PVDE bits and all PVDnCMPCR.PVDE bits are 0.

RHSEL Bit (Rising Hysteresis Select)

When RHSEL = 0 for VCC drop detection, the hysteresis level is set above the detection voltage set by the PVDLVL[4:0] bits. When RHSEL = 1 for VCC rise detection, the hysteresis level is set under the detection voltage set by the PVDLVL[4:0] bits. For details of the hysteresis level of the PVD, see section 67, Electrical Characteristics.
If a PVD reset generated by VCC drop detection is required, set the RHSEL bit to 0.
If a PVD reset generated by VCC rise detection is required, set the RHSEL bit to 1.
When PVDmCR0.RI = 0, do not set the RHSEL bit to 1.
When PVDmCR0.RN = 1, do not set the RHSEL bit to 1.

8.2.9 Voltage Monitoring n Function Control Register (PVDnFCR) (n = 4, 5)

Address(es): SYSC.PVD4FCR 8701 9B2Ch, SYSC.PVD5FCR 8701 9B30h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RHSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RHSEL	Rising Hysteresis Select	0: Hysteresis level for VCC drop detection selected 1: Hysteresis level for VCC rise detection selected	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The PVDnFCR.RHSEL bit can only be changed when all PVDmCMPCR.PVDE bits and all PVDnCMPCR.PVDE bits are 0.

RHSEL Bit (Rising Hysteresis Select)

When RHSEL = 0 for VCC drop detection, the hysteresis level is set above the detection voltage set by the PVDLVL[4:0] bits. When RHSEL = 1 for VCC rise detection, the hysteresis level is set under the detection voltage set by the PVDLVL[4:0] bits. For details of the hysteresis level of the PVD, see section 67, Electrical Characteristics.
If a PVD reset generated by VCC drop detection is required, set the RHSEL bit to 0.
If a PVD reset generated by VCC rise detection is required, set the RHSEL bit to 1.

8.2.10 Voltage Monitoring Lock Register (PVDLR)

Address(es): SYSC.PVDLR 8701 9B34h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LOCK
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	LOCK	Lock Control	0: The control registers of PVD4 and PVD5 are writable 1: The control registers of PVD4 and PVD5 are not writable	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

LOCK Bit (Lock Control)

The initial value of the LOCK bit is 1. When the LOCK bit is 1, the control registers of PVD4 and PVD5 are not writable. 0 can be written to the LOCK bit only once after the LOCK bit has been initialized by a reset that initializes the LOCK bit (POR, RES# pin reset, voltage monitoring 0 reset (PVD0 reset)). The control registers of PVD4 and PVD5 are only writable in this state.

After that, writing either value to the LOCK bit causes its value to be fixed to 1 and the control registers of PVD4 and PVD5 become unwritable. The control registers of PVD4 and PVD5 are PVD4CMPCR, PVD5CMPCR, PVD4CR0, PVD5CR0, PVD4FCR, and PVD5FCR.

8.3 VCC Input Voltage Monitoring

8.3.1 Monitoring Vdet0

The results of comparison by voltage monitoring 0 cannot be read.

8.3.2 Monitoring Vdetm

Table 8.3 lists the procedures for setting up monitoring against Vdetm. After the settings are completed, the results of comparison by voltage monitoring m can be monitored by using the PVDmSR.MON flag.

Table 8.3 Procedures for Setting up Monitoring against Vdetm

Step		Monitoring the Results of Comparison by Voltage Monitoring m
Setting the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit). (Disable the voltage detection m circuit before writing to the PVDmCMPCR.PVDLVL[4:0] bits.)
	2	Select the detection voltage by setting the PVDmCMPCR.PVDLVL[4:0] bits.
	3	Set PVDmCMPCR.PVDE = 1 (enabling the voltage detection m circuit).
	4	Wait for at least td(E-A).*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter by setting the PVDmCR0.FSAMP[1:0] bits.
	6	Set PVDmCR0.DFDIS = 0 (enabling the digital filter).
	7	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).
Enabling output	8	Set PVDmCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitoring m).

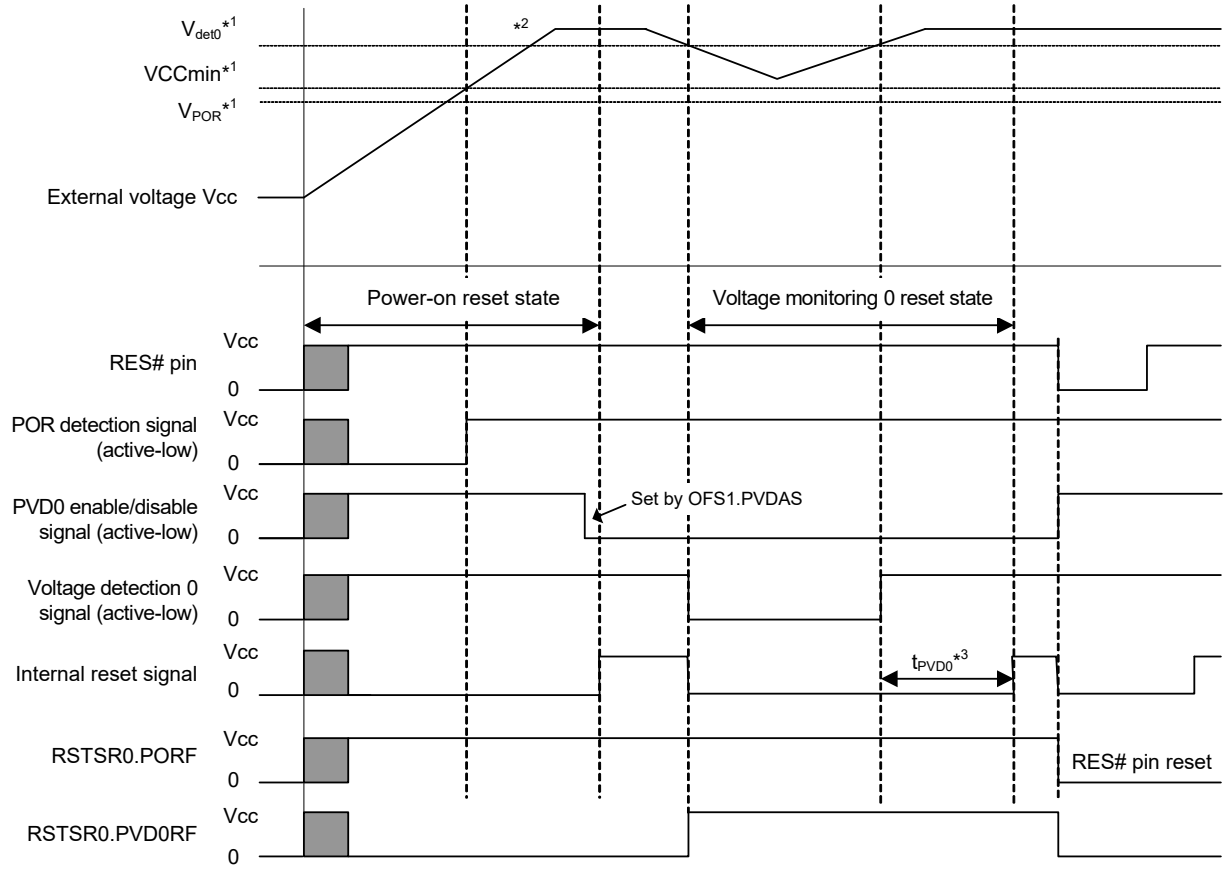
Note 1. Steps 5 to 7 can be performed during the waiting time of step 4.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

When changing the settings of PVDm again after setting PVDm according to the procedure in Table 8.3, set the PVDmCR0.CMPE bit to 0 first, and then set PVDm again according to the procedure in Table 8.3.
When changing the PVDmCMPCR.PVDLVL[4:0] bits, follow the description in section 8.2.2, Voltage Monitoring m Comparator Control Register (PVDmCMPCR) (m = 1, 2).

8.4 Voltage Monitoring 0 Reset

When using a voltage monitoring 0 reset, clear the OFS1.PVDAS bit to 0 (enabling the voltage monitoring 0 reset after a reset).

Figure 8.4 shows an example of the operation of the voltage monitoring 0 reset.



Note . For details on the electrical characteristics, see section 67, Electrical Characteristics.
Note 1. Vdet0 is the detection level for the voltage monitoring 0 reset, VCCmin is the minimum guaranteed voltage of the MCU, and VPOR is the detection level for the power-on reset.
Note 2. At power-on, VCC should rise to or above the minimum guaranteed voltage before release from the POR reset state.
Note 3. tPVD0 is the period until release from the voltage monitoring 0 reset.

Figure 8.4 Example of Operation of Voltage Monitoring 0 Reset

8.5 Voltage Monitoring m Interrupt and Voltage Monitoring m Reset (m = 1, 2)

An interrupt or reset can be generated in response to the results of comparison by the voltage detection m circuit. Table 8.4 lists the procedures for setting bits related to the voltage monitoring m interrupt and voltage monitoring m reset to run voltage monitoring. Table 8.5 lists the procedure for setting bits related to the voltage monitoring m interrupt and voltage monitoring m reset to stop voltage monitoring. Figure 8.5 shows an example of the operation of the voltage monitoring m interrupt. For the operation of the voltage monitoring m reset, see Figure 6.2 in section 6, Resets. Furthermore, when using the voltage monitoring m circuit in software standby or deep software standby mode, make settings for the voltage monitoring m circuit according to the following procedures.

(1) Setting in software standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- When PVDmFCR.RHSEL = 0
  After VCC > Vdetm is detected, negate the voltage monitoring m reset signal (PVDmCR0.RN = 0) following a stabilization time.

(2) Settings in deep software standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- Enable the voltage monitoring m interrupt (PVDmCR0.RI = 0). When the voltage monitoring m reset is enabled (PVDmCR0.RI = 1), a transition to deep software standby mode 2 or 3 is not possible, and the device makes the transition to software standby mode 1.
- During deep software standby mode 3, the voltage monitoring m circuit is stopped. When using the voltage monitoring m circuit in deep software standby mode, do not allow a transition to deep software standby mode 3.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring m Interrupt and Voltage Monitoring m Reset to Run Voltage Monitoring

Step		Voltage Monitoring m Interrupt (Voltage Monitoring m ELC Event Output)	Voltage Monitoring m Reset
Setting the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit). (Disable the voltage detection m circuit before writing to the PVDmCMPCR.PVDLVL[4:0] and PVDmFCR.RHSEL bits.)	
	2	Select the detection voltage by setting the PVDmCMPCR.PVDLVL[4:0] bits.	
	3	—	Select the type of reset condition by setting the PVDmFCR.RHSEL bit.
	4	Set PVDmCMPCR.PVDE = 1 (enabling the voltage detection m circuit).	
	5	Wait for at least td(E-A).*1	
Setting the digital filter*2	6	Select the sampling clock for the digital filter by setting the PVDmCR0.FSAMP[1:0] bits.	
	7	Set PVDmCR0.DFDIS = 0 (enabling the digital filter).	
	8	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s)*4	
Setting the voltage monitoring m interrupt or reset	9	Set PVDmCR0.RI = 0 (voltage monitoring m interrupt).	• Set PVDmCR0.RI = 1 (voltage monitoring m reset). • Select the type of reset negation by setting the PVDmCR0.RN bit.
	10	• Select the timing of interrupt requests by setting the PVDmCR1.IDTSEL[1:0] bits. • Select the type of interrupt by setting the PVDmCR1.IRQSEL bit.	—
Enabling output	11	Set PVDmSR.DET = 0.	—
	12	Set PVDmCR0.RIE = 1 (enabling the voltage monitoring m interrupt or reset).*3	
	13	Set PVDmCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitoring m).	

Note 1. Steps 6 to 12 can be performed during the waiting time of step 5.
Note 2. Steps 6 to 8 are not required if the digital filter is not in use.
Note 3. Step 12 is not required if only the ELC event signal is to be output.
Note 4. Steps 9 to 12 can be performed during the waiting time of step 8.

When changing the settings of PVDm again after setting PVDm according to the procedure in Table 8.4, stop PVDm according to the procedure in Table 8.5, and then set PVDm again according to the procedure in Table 8.4.
When changing the PVDmCMPCR.PVDLVL[4:0] bits, follow the description in section 8.2.2, Voltage Monitoring m Comparator Control Register (PVDmCMPCR) (m = 1, 2).
When changing the PVDmFCR.RHSEL bit, follow the description in section 8.2.8, Voltage Monitoring m Function Control Register (PVDmFCR) (m = 1, 2).

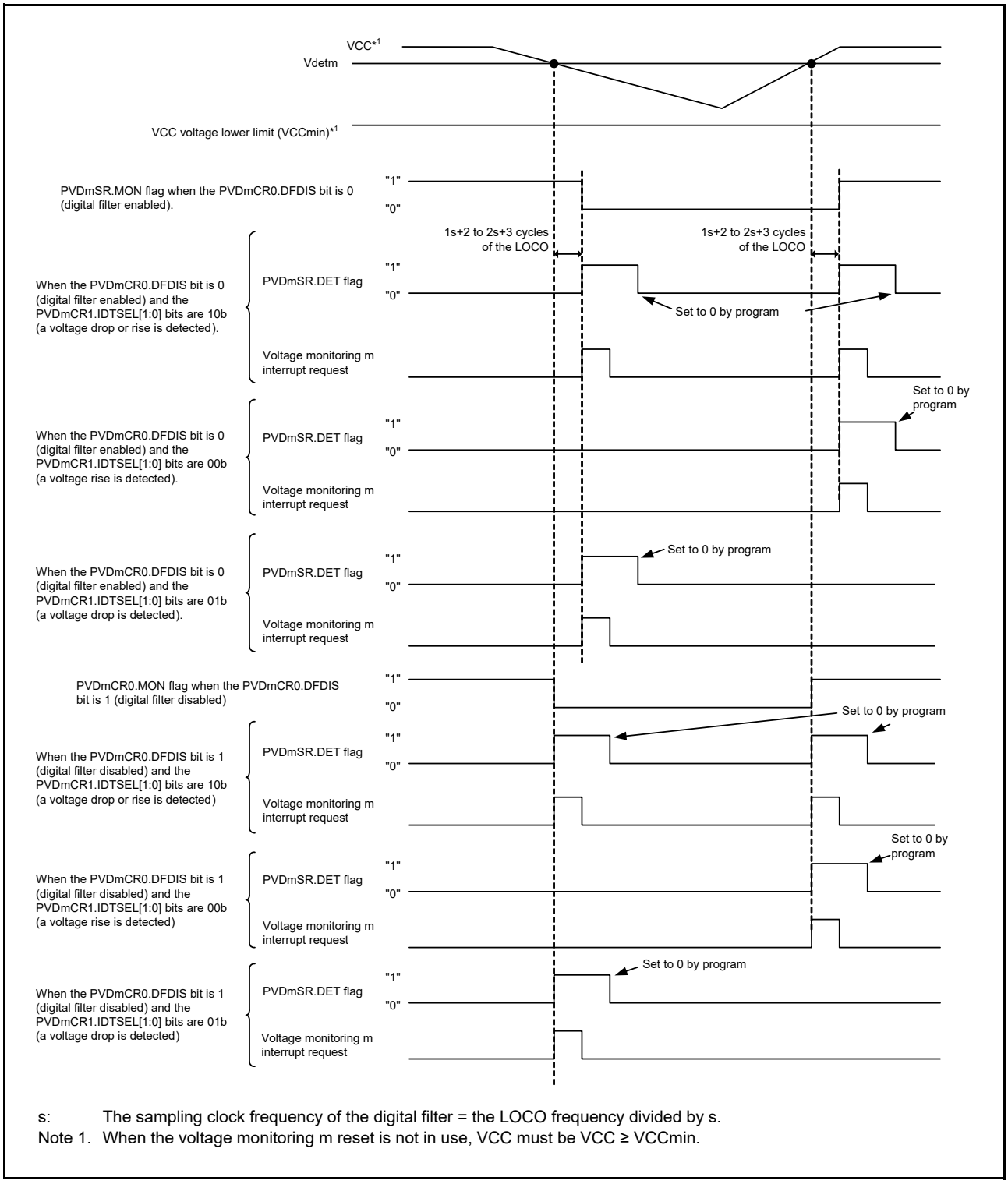
Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring m Interrupt and Voltage Monitoring m Reset to Stop Voltage Monitoring

Step		Voltage Monitoring m Interrupt (Voltage Monitoring m ELC Event Output), Voltage Monitoring m Reset
Stopping enabling of output	1	Set PVDmCR0.CMPE = 0 (disabling output of the results of comparison by voltage monitoring m).
	2	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).*1
	3	Set PVDmCR0.RIE = 0 (disabling the voltage monitoring m interrupt or reset).*3
Stopping the digital filter	4	Set PVDmCR0.DFDIS = 1 (disabling the digital filter).*1, *2
Stopping the voltage detection m circuit	5	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.
Note 3. Step 3 is not required if only the ELC event signal is to be output.

If the voltage monitoring m interrupt or voltage monitoring m reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted depending on the condition.

- Setting or stopping the voltage detection m circuit is not required if the setting for the voltage detection m circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring m interrupt or voltage monitoring m reset is not required if the setting for the voltage monitoring m interrupt or reset is not to be changed.



s: The sampling clock frequency of the digital filter = the LOCO frequency divided by s.

Note 1. When the voltage monitoring m reset is not in use, VCC must be VCC ≥ VCCmin.

Figure 8.5 Example of Operation of Voltage Monitoring m Interrupt

8.6 Voltage Monitoring n Reset (n = 4, 5)

A reset can be generated in response to the results of comparison by the voltage detection n circuit.
Table 8.6 lists the procedures for setting bits related to the voltage monitoring n reset to run voltage monitoring, and Table 8.7 lists the procedure for setting bits related to the voltage monitoring n reset to stop voltage monitoring. For the operation of the voltage monitoring n reset, see Figure 6.3 in section 6, Resets.
Furthermore, when using the voltage monitoring n circuit in software standby or deep software standby mode, make settings for the voltage monitoring n circuit according to the following procedures.

(1) Setting in software standby mode

- Disable the digital filter (PVDnCR0.DFDIS = 1).

(2) Settings in deep software standby mode

- Disable the digital filter (PVDnCR0.DFDIS = 1).
- During deep software standby mode 3, the voltage monitoring n circuit is stopped. When using the voltage monitoring n circuit in deep software standby mode, do not allow a transition to deep software standby mode 3.

Table 8.6 Procedures for Setting Bits Related to the Voltage Monitoring n Reset to Run Voltage Monitoring

Step		Voltage Monitoring n Reset
Setting the voltage detection n circuit	1	Set PVDnCMPCR.PVDE = 0 (disabling the voltage detection n circuit). (Disable the voltage detection n circuit before writing to the PVDnCMPCR.PVDLVL[4:0] and PVDnFCR.RHSEL bits.)
	2	Select the detection voltage by setting the PVDnCMPCR.PVDLVL[4:0] bits.
	3	Select the type of reset condition by setting the PVDnFCR.RHSEL bit.
	4	Set PVDnCMPCR.PVDE = 1 (enabling the voltage detection n circuit).
	5	Wait for at least td(E-A).*1
Setting the digital filter*2	6	Select the sampling clock for the digital filter by setting the PVDnCR0.FSAMP[1:0] bits.
	7	Set PVDnCR0.DFDIS = 0 (enabling the digital filter).
	8	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s)*3
Enabling output	9	Set PVDnCR0.RE = 1 (enabling the voltage monitoring n reset).*3
	10	Set PVDnCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitoring n).

Note 1. Steps 6 to 10 can be performed during the waiting time of step 5.
Note 2. Steps 6 to 8 are not required if the digital filter is not in use.
Note 3. Step 9 can be performed during the waiting time of step 8.

When changing the settings of PVDn again after setting PVDn according to the procedure in Table 8.6, stop PVDn according to the procedure in Table 8.7, and then set PVDn again according to the procedure in Table 8.6.
When changing the PVDnCMPCR.PVDLVL[4:0] bits, follow the description in section 8.2.3, Voltage Monitoring n Comparator Control Register (PVDnCMPCR) (n = 4, 5).
When changing the PVDnFCR.RHSEL bit, follow the description in section 8.2.9, Voltage Monitoring n Function Control Register (PVDnFCR) (n = 4, 5).

Table 8.7 Procedures for Setting Bits Related to the Voltage Monitoring n Reset to Stop Voltage Monitoring

Step		Voltage Monitoring n Reset
Stopping enabling of output	1	Set PVDnCR0.CMPE = 0 (disabling output of the results of comparison by voltage monitoring n).
	2	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).*1
	3	Set PVDnCR0.RE = 0 (disabling the voltage monitoring n reset).
Stopping the digital filter	4	Set PVDnCR0.DFDIS = 1 (disabling the digital filter).*1, *2
Stopping the voltage detection m circuit	5	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage monitoring n reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted depending on the condition.

- Setting or stopping the voltage detection n circuit is not required if the setting for the voltage detection n circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring n reset is not required if the setting for the voltage monitoring n reset is not to be changed.

8.7 Event Link Output

The PVD can output event signals to the event link controller (ELC).

- Vdetm crossing detection event output

The PVD outputs an event signal upon detection of a Vdetm crossing while a voltage detection m circuit is enabled and output of the results of comparison by the voltage monitoring m circuit is enabled (m = 1, 2).

To enable the event link output function of the PVD, enable the PVD event link function on the ELC side after making the setting to enable the PVD. To stop the event link output function of the PVD, disable the PVD event link function on the ELC side before making the setting to stop the PVD.

8.7.1 Interrupt Handling and Event Linking

Each PVD has an enable bit that controls enabling or disabling of the voltage monitoring 1 and 2 interrupts, respectively. When the corresponding interrupt source condition is generated, the interrupt signal is output to the CPU if the interrupt enable bit is set to “enable”.
When handled as an event link output signal, on the other hand, the signal is output to the other module via the ELC as an event signal upon generation of the interrupt source condition regardless of the value of the interrupt enable bit.
The voltage monitoring m interrupt signal can be output in software standby mode, deep software standby mode 1, and deep software standby mode 2. In these modes, the event signals for the ELC are output as follows:

- If a Vdetm crossing is detected while the device is in software standby mode, an event signal for the ELC is not output because a clock signal is not supplied in software standby mode. However, the Vdetm crossing detection flag is retained, so when the clock supply is resumed on return from software standby mode, the event signal for the ELC is output according to the state of the Vdetm crossing detection flag.

- If a Vdetm crossing is detected while the device is in deep software standby mode, an event signal for the ELC is not output.

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.
Table 9.1 and Table 9.2 list the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit and Table 9.3 lists the I/O pins.

Table 9.1 Specifications of the Clock Generation Circuit (Clock Sources)

Clock Source	Description	Specifications
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 48 MHz
	External clock input frequency	Up to 48 MHz
	Connectable resonator or additional circuit: Ceramic resonator or crystal resonator Pins for connection: EXTAL and XTAL	Available
	Driving capacity switching	Available
	Oscillation stop detection function	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	Connectable resonator or additional circuit: Crystal resonator Pins for connection: XCIN and XCOUT	Available
	External clock input	Available
	Driving capacity switching	Available
	Oscillation stop detection function	Available
PLL1 circuit PLL2 circuit	Input clock source	MOSC or HOCO
	Input clock frequency division ratio	Selectable from division by 1 to division by 3
	Input clock frequency	8 MHz to 48 MHz
	Input clock frequency (after input frequency division)	8 MHz to 24 MHz
	Frequency multiplication ratio	Selectable from 40 to 300 (Fractional part: 0, 0.33, 0.50, or 0.66)
	VCO frequency	960 MHz to 2400 MHz
	Number of output clocks	Three different clocks are output.
	PLL output clock P	60 to 600 MHz (PLL1)*1, 60 to 500 MHz (PLL2)*2 (Output frequency division ratio: 2, 3, 4, 6, 8, or 16)
	PLL output clock Q	106.6 to 500 MHz*3 (Output frequency division ratio: 2, 3, 4, 5, 6, 8, or 9)
	PLL output clock R	106.6 to 500 MHz (Output frequency division ratio: 2, 3, 4, 5, 6, 8, or 9)
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16, 18, 20, 32, or 48 MHz
	Frequency locked loop (FLL) function	Available
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

Note 1. The specification of the maximum frequency varies depending on the part number.

- R5K574NxA4xx and R5K574NxB4xx: 60 to 600 MHz
- R5K574NxC4xx and R5K574NxD4xx: 60 to 480 MHz
  In DCDC mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.
- R5K574NxE4xx and R5K574NxF4xx: 60 to 300 MHz
  In DCDC mode or LDO mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.

Note 2. The specification of the maximum frequency varies depending on the part number.
- R5K574NxA4xx, R5K574NxB4xx, R5K574NxC4xx, and R5K574NxD4xx: 60 to 500 MHz
- R5K574NxE4xx and R5K574NxF4xx: 60 to 300 MHz

Note 3. The specification of the maximum frequency varies depending on the part number.
- R5K574NxA4xx, R5K574NxB4xx, R5K574NxC4xx, and R5K574NxD4xx: 106.6 MHz to 500 MHz
- R5K574NxE4xx and R5K574NxF4xx: 106.6 MHz to 300 MHz

Table 9.2 Specifications of the Clock Generation Circuit (Internal Clocks) (1/3)

Item	Clock Source	Maximum Frequency of Clock Source	Destination of Clock Supply	Specifications
CPU clock (CPUCLK0)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	CPU	Up to 600 MHz*1 Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
MRAM code bus clock (MRICLK)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Code MRAM and MRAM code bus	Up to 300 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
MRAM peripheral bus clock (MRPCLK)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Extra MRAM and MRAM peripheral bus	Up to 150 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
System clock (ICLK)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	DMAC, EXDMAC, DTC, SRAM, system bus, I/O ports, and ICU	Up to 300 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Peripheral module clock A (PCLKA)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	High-speed peripheral bus (ESWM, USBHS, RSCI, RI3C, RIICHS, CANFDG, CANFD, RSPIA, CRC, DOC, ADC16H, RSIP-E50D, GPTW, HRPWM, and POEG) and DOTF operating clock	Up to 150 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Peripheral module clock B (PCLKB)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Low-speed peripheral bus (CAC, ELC, RTC, WDT, IWDT, USB, SSIE, SDHI, TEMPS, XSPI, DOTF, DAC12R, PDMIF, PDC, PCIF, CMT, CMTW, TMR, and PIO)	Up to 75 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Peripheral module clock C (PCLKC)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Not used	Up to 75 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Peripheral module clock D (PCLKD)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Peripheral module (GPTW count clock)	Up to 150 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Peripheral module clock E (PCLKE)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	Peripheral module (CANFDG message buffer RAM, CANFDG acceptance filter RAM, CANFD message buffer RAM)	Up to 300 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
External bus clock (BCLK)	MOSC, SOSC, HOCO, MOCO, or PLL1P	600 MHz*2	External bus	Up to 150 MHz Division ratio: 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, or 64
Asynchronous external bus clock (BCLKA)	MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	External bus	Up to 133 MHz Division ratio: 1, 2, 4, 6, 8, 10, 16, or 32
EBCLK pin output (EBCLK)	BCLK, 1/2 BCLK, or BCLKA	—	EBCLK pin	Up to 100 MHz Division ratio: 1 or 2
SDRAM clock (SDCLK)	BCLK or BCLKA	—	SDCLK pin, external bus for SDRAM	Up to 133 MHz

Table 9.2 Specifications of the Clock Generation Circuit (Internal Clocks) (2/3)

Item	Clock Source	Maximum Frequency of Clock Source	Destination of Clock Supply	Specifications
SCI clock (SCICLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	RSCI	Up to 120 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
SPI clock (SPICLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	RSPIA	Up to 100 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
XSPI clock (XSPICLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	XSPI	Up to 266 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
Divided XSPI clock (XSPIDIVCLK)	1/2 XSPICLK	—	XSPI	Up to 133 MHz
CANFD core clock (CANFDCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	CANFD and CANFDG	Up to 80 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
ADC clock (ADCCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	ADC16H	Up to 60 MHz Division ratio: 1, 2, 4, 6, 8, 10, 16, or 32
GPT clock (GPTCLK)	MOSC, HOCO, MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	GPTW, ADC16H, HRPWM, and PCIF	Up to 200 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
USB clock (USBCLK)	MOSC, HOCO, MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	USB and USBHS	48 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
USBHS-PHY clock (USB60CLK)	MOSC, HOCO, MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	USBHS	60 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
I3C clock (I3CCLK)	MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	RI3C and RIICHS	Up to 100 MHz Division ratio: 1, 2, 3, 4, 5, 6, 8, 10, 16, or 32
ESWM clock (ESWCLK)	MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz	ESWM	Up to 250 MHz Division ratio: 1, 2, 4, 6, 8, 10, 16, or 32
ESWM-PHY clock (ESWPHYCLK)	MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz	ESWM	100, 200, or 500 MHz Division ratio: 1, 2, 4, 6, 8, 10, 16, or 32
Ethernet-PHY clock (ETHPHYCLK)	MOSC, MOCO, PLL1P, PLL1Q, PLL1R, PLL2P, PLL2Q, or PLL2R	500 MHz*3	ETHPHYCLK pin	25 or 50 MHz Division ratio: 1, 2, 4, 6, 8, 10, 16, or 32
Clock or buzzer output (CLKOUT)	MOSC, SOSC, HOCO, MOCO, LOCO, ICLK, PLL1P, PLL1Q, PLL2P, or PLL2Q	300 MHz	CLKOUT pin	Up to 48 MHz Division ratio: 1, 2, 4, 8, 16, 32, 64, or 128
MOSC clock for USBHS (USBMCLK)	MOSC	—	USBHS	12, 20, 24, or 48 MHz Not divided
CANFD main clock (CANFDMCLK)	MOSC	—	CANFD and CANFDG	8 MHz to 48 MHz Not divided

Table 9.2 Specifications of the Clock Generation Circuit (Internal Clocks) (3/3)

Item	Clock Source	Maximum Frequency of Clock Source	Destination of Clock Supply	Specifications
PDMIF clock (PDMIFCLK)	MOCO	—	PDMIF	8 MHz Not divided
CAC main clock (CACMCLK)	MOSC	—	CAC	Up to 48 MHz Not divided
CAC sub-clock (CACSCLK)	SOSC	—	CAC	32.768 kHz Not divided
CAC HOCO clock (CACHCLK)	HOCO	—	CAC	16, 18, 20, 32, or 48 MHz Not divided
CAC MOCO clock (CACMOCLK)	MOCO	—	CAC	8 MHz Not divided
CAC LOCO clock (CACLCLK)	LOCO	—	CAC	32.768 kHz Not divided
CAC PLL1P clock (CACPL1PCLK)	PLL1P	—	CAC	Up to 37.5 MHz Divided by 16
CAC PLL1Q clock (CACPL1QCLK)	PLL1Q	—	CAC	Up to 31.25 MHz Divided by 16
CACPLL1R clock (CACPL1RCLK)	PLL1R	—	CAC	Up to 31.25 MHz Divided by 16
CACPLL2P clock (CACPL2PCLK)	PLL2P	—	CAC	Up to 31.25 MHz Divided by 16
CACPLL2Q clock (CACPL2QCLK)	PLL2Q	—	CAC	Up to 31.25 MHz Divided by 16
CACPLL2R clock (CACPL2RCLK)	PLL2R	—	CAC	Up to 31.25 MHz Divided by 16
RTC LOCO clock (RTCLCLK)	LOCO	—	RTC	32.768 kHz Not divided
RTC sub-clock (RTCSCLK)	SOSC	—	RTC	32.768 kHz Not divided
IWDT clock (IWDTCLK)	LOCO divided by 2	—	IWDT	16.384 kHz Not divided
JTAG clock (JTAGTCK)	TCK	—	JTAG	Up to 25 MHz Not divided
Serial wire clock (SWCLK)	TCK	—	OCD	Up to 25 MHz Not divided

Note: Restrictions on clock frequency settings:
CPUCLK0 ≥ ICLK, MRICLK ≥ ICLK, CPUCLK0 ≥ MRICLK, ICLK ≥ MRPCLK, ICLK ≥ PLKA ≥ PCLKB, ICLK ≥ BCLK, GPTCLK ≥ PCLKA, PCLKD ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratios (N: integer up to 64):
CPUCLK0:ICLK = N:1, MRICLK:ICLK = N:1, CPUCLK0:MRICLK = N:1, ICLK:MRPCLK = N:1, ICLK:BCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N, ICLK:PCLKE = N:1 or 1:N
When one clock is set to division by 3, 6, 12, or 24, do not set the other clocks to division by 2, 4, 8, 16, 32, or 64 by the SCKDIVCR and SCKDIVCR2 registers.

Note: Set each clock within the allowable frequency range shown in the Specifications column in the table.
The guaranteed frequency range for the HOCO differs between when the FLL function is used and when it is not used (see section 67, Electrical Characteristics).
Take special care when the PLL clock source is set to the HOCO. Set the PLL multiplication ratio so that the resultant frequency is in the allowable frequency range even if the maximum value of the guaranteed frequency range for the HOCO is reached.
In addition, when the PLL clock source is set to the HOCO, select the SRAM and MRAM wait cycles carefully (see section 64, SRAM, and section 65, MRAM).
For SRAM, determine the setting value based on whether the SRAM access frequency exceeds 150 MHz when the maximum value of the guaranteed frequency range for the HOCO is reached.
For MRAM, specify the MRAM access frequency value when the maximum value of the guaranteed frequency range for the

HOCO is reached.

Note 1. The specification of the maximum frequency varies depending on the part number.
- R5K574NxA4xx and R5K574NxB4xx: 600 MHz
- R5K574NxC4xx and R5K574NxD4xx: 480 MHz
  In DCDC mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.
- R5K574NxE4xx and R5K574NxF4xx: 300 MHz
  In DCDC mode or LDO mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.

Note 2. The maximum frequency of clock source varies depending on the part number.
- R5K574NxA4xx and R5K574NxB4xx: 600 MHz
- R5K574NxC4xx and R5K574NxD4xx: 480 MHz
  In DCDC mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.
- R5K574NxE4xx and R5K574NxF4xx: 300 MHz
  In DCDC mode or LDO mode, set the VSCR.VSCM[2:0] bits to select VSCR_2. In external VDD mode, use this device within voltage range 2.

Note 3. The maximum frequency of clock source varies depending on the part number.
- R5K574NxA4xx, R5K574NxB4xx, R5K574NxC4xx, and R5K574NxD4xx: 500 MHz
- R5K574NxE4xx and R5K574NxF4xx: 300 MHz

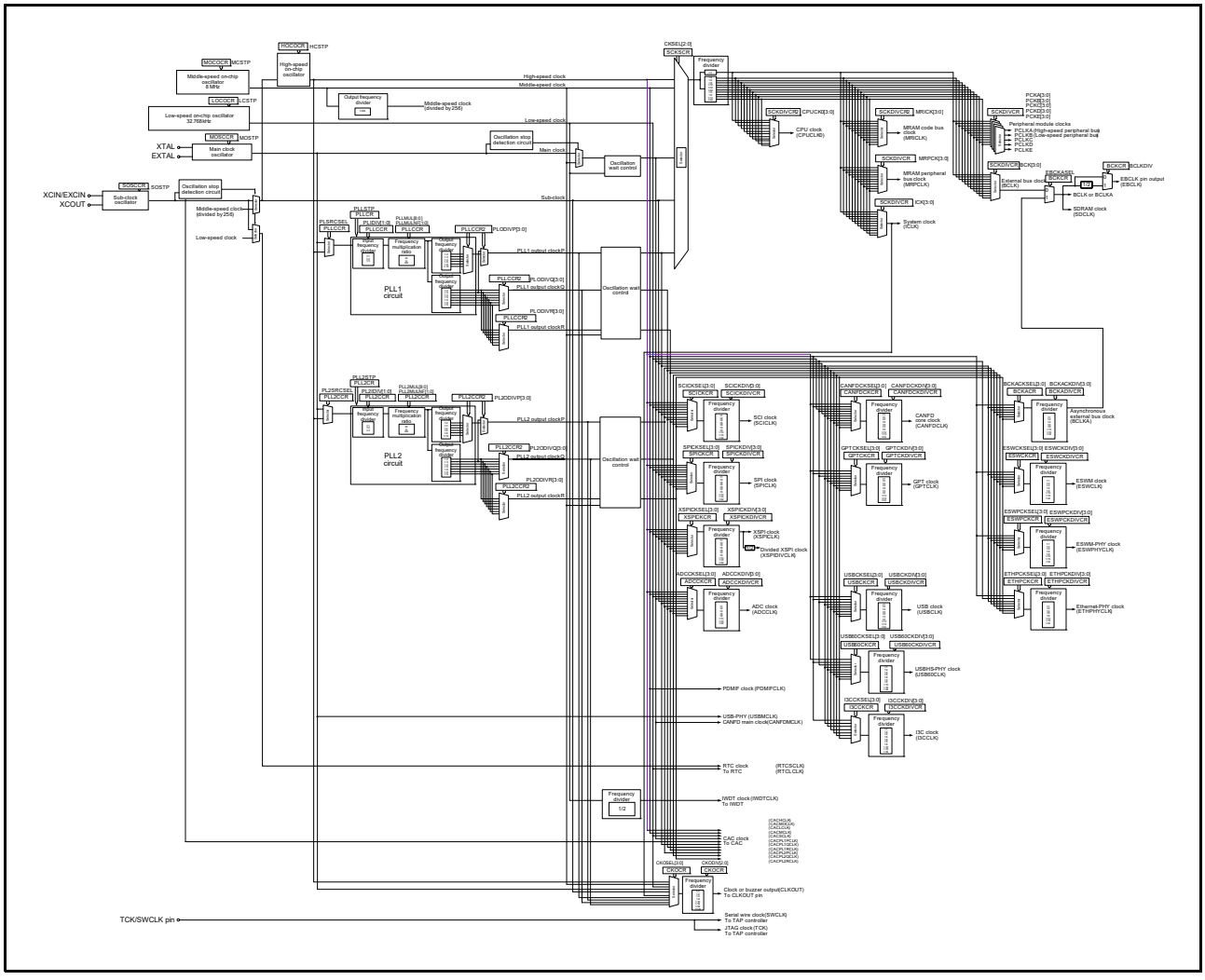


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.3 lists the input/output pins of the clock generation circuit.

Table 9.3 Input/Output Pins of the Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a ceramic resonator or a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 9.3.2, Inputting an External Clock.
EXTAL	Input	
XCIN/EXCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator. The XCIN pin can also be used to input an external clock. For details, see section 9.4.3, Input of an External Clock.
XCOUT	Output	
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG and SWD.
EBCLK	Output	This pin is used to supply external devices with the external bus clock (EBCLK).
SDCLK	Output	This pin is used to supply external devices with the SDRAM clock (SDCLK).
CLKOUT	Output	This pin is used to output the CLKOUT clock or buzzer signal.

9.2 Register Descriptions

9.2.1 Clock Generation Function Security Attribute Register (CGFSAR)

Address(es): SYSC.CGFSAR 8701 93C0h, SYSC_NS.CGFSAR 0701 93C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	NONSEC28	NONSEC27	—	NONSEC25	NONSEC24	—	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC09	NONSEC08	NONSEC07	NONSEC06	NONSEC05	NONSEC04	NONSEC03	NONSEC02	—	NONSEC00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NONSEC00	System Clock Attribute Setting	Target: SCKDIVCR, SCKDIVCR2, SCKSCR 0: Secure 1: Non-secure	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	NONSEC02	HOCO Attribute Setting	Target: HOCOCR, HOCOCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR 0: Secure 1: Non-secure	R/W
b3	NONSEC03	MOCO Attribute Setting	Target: MOCOCR, MOCOUTCR, MOCOSCR 0: Secure 1: Non-secure	R/W
b4	NONSEC04	LOCO Attribute Setting	Target: LOCOCR, LOCOUTCR 0: Secure 1: Non-secure	R/W
b5	NONSEC05	Main Clock Attribute Setting	Target: MOSCCR, MOSCWTCR, MOMCR, MOSCSCR 0: Secure 1: Non-secure	R/W
b6	NONSEC06	Oscillation Stop Detection-Related Register Attribute Setting	Target: OSTDCR, OSTDSR 0: Secure 1: Non-secure	R/W
b7	NONSEC07	Sub-Clock Attribute Setting	Target: SOSCCR, SOMCR 0: Secure 1: Non-secure	R/W
b8	NONSEC08	PLL1 Clock Attribute Setting	Target: PLLCCR, PLLCCR2, PLLCR 0: Secure 1: Non-secure	R/W
b9	NONSEC09	PLL2 Clock Attribute Setting	Target: PLL2CCR, PLL2CCR2, PLL2CR 0: Secure 1: Non-secure	R/W
b10	NONSEC10	Sub-Clock Stop Detection-Related Register Attribute Setting	Target: SOSTDCR, SOSTDSR 0: Secure 1: Non-secure	R/W
b11	NONSEC11	Clock Output Attribute Setting	Target: CKOCR 0: Secure 1: Non-secure	R/W
b12	NONSEC12	External Bus Clock Attribute Setting	Target: BCKCR, EBCKOCR 0: Secure 1: Non-secure	R/W
b13	NONSEC13	SDRAM Clock Attribute Setting	Target: SDCKOCR 0: Secure 1: Non-secure	R/W

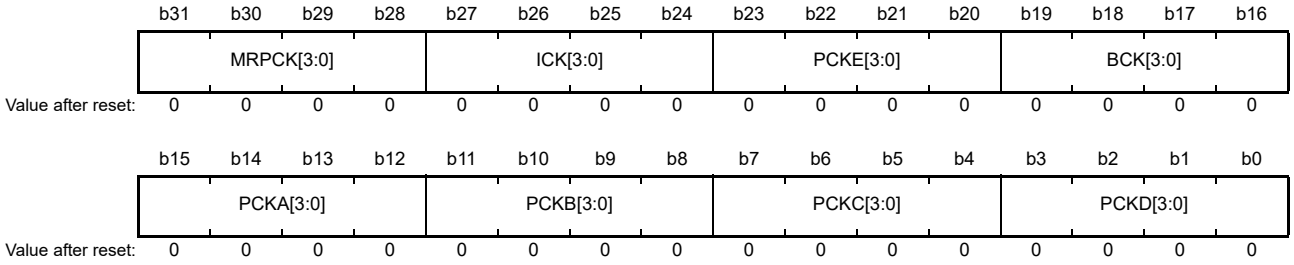
Bit	Symbol	Bit Name	Description	R/W
b14	NONSEC14	Asynchronous External Bus Clock Attribute Setting	Target: BCKADIVCR, BCKACR 0: Secure 1: Non-secure	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	NONSEC16	USB Clock Attribute Setting	Target: USBCKDIVCR, USBCKCR 0: Secure 1: Non-secure	R/W
b17	NONSEC17	XSPI Clock Attribute Setting	Target: XSPICKDIVCR, XSPICKCR 0: Secure 1: Non-secure	R/W
b18	NONSEC18	CANFD Clock Attribute Setting	Target: CANFDCKDIVCR, CANFDCKCR 0: Secure 1: Non-secure	R/W
b19	NONSEC19	USBHS-PHY Clock Attribute Setting	Target: USB60CKDIVCR, USB60CKCR 0: Secure 1: Non-secure	R/W
b20	NONSEC20	I3C Clock Attribute Setting	Target: I3CCKDIVCR, I3CCKCR 0: Secure 1: Non-secure	R/W
b21	NONSEC21	SCI Clock Attribute Setting	Target: SCICKDIVCR, SCICKCR 0: Secure 1: Non-secure	R/W
b22	NONSEC22	SPI Clock Attribute Setting	Target: SPICKDIVCR, SPICKCR 0: Secure 1: Non-secure	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b24	NONSEC24	ADC Clock Attribute Setting	Target: ADCCKDIVCR, ADCCKCR 0: Secure 1: Non-secure	R/W
b25	NONSEC25	GPT Clock Attribute Setting	Target: GPTCKDIVCR, GPTCKCR 0: Secure 1: Non-secure	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	NONSEC27	ESWM-PHY Clock Attribute Setting	Target: ESWCKDIVCR, ESWCKCR, ESWPCKDIVCR, ESWPCKCR 0: Secure 1: Non-secure	R/W
b28	NONSEC28	Ethernet-PHY Clock Attribute Setting	Target: ETHPCKDIVCR, ETHPCKCR 0: Secure 1: Non-secure	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before changing the value of this register.

The CGFSAR register is used to control the security attribute of the clock generation function registers.

9.2.2 System Clock Frequency Division Control Register (SCKDIVCR)

Address(es): SYSC.SCKDIVCR 8701 9020h,
SYSC_NS.SCKDIVCR 0701 9020h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select*1	b3 b0 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b7 to b4	PCKC[3:0]	Peripheral Module Clock C (PCLKC) Select*2	b7 b4 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select*3	b11 b8 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W

Bit	Symbol	Bit Name	Description	R/W
b15 to b12	PCKA[3:0]	Peripheral Module Clock A (PCLKA) Select*3, *4, *5	b15 b12 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b19 to b16	BCK[3:0]	External Bus Clock (BCLK) Select*6	b19 b16 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b23 to b20	PCKE[3:0]	Peripheral Module Clock E (PCLKE) Select*1, *5	b23 b20 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select *1, *3, *4, *6, *7, *8, *9	b27 b24 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b31 to b28	MRPCK[3:0]	MRAM Peripheral Bus Clock (MRPCLK) Select*7, *10	b31 b28 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note: Restrictions on clock frequency settings:
CPUCLK0 ≥ ICLK, MRICLK ≥ ICLK, CPUCLK0 ≥ MRICLK, ICLK ≥ MRPCLK, ICLK ≥ PCLKA ≥ PCLKB, ICLK ≥ BCLK, GPTCLK ≥ PCLKA, PCLKD ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratios (N: integer up to 64):
CPUCLK0:ICLK = N:1, MRICLK:ICLK = N:1, CPUCLK0:MRICLK = N:1, ICLK:MRPCLK = N:1, ICLK:BCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N, ICLK:PCLKE = N:1 or 1:N
When one clock is set to division by 3, 6, 12, or 24, do not set the other clocks to division by 2, 4, 8, 16, 32, or 64 by the SCKDIVCR and SCKDIVCR2 registers.

Note 1. The following condition should be satisfied between the frequencies of the system clock (ICLK) and peripheral module clocks (PCLKD, PCLKE).
ICLK:PCLKD or PCLKE = N:1 (N: integer)

Note 2. Set the PCKC[3:0] bits to the same value as the PCKB[3:0] bits.

Note 3. The following condition should be satisfied between the frequencies of the system clock (ICLK) and peripheral module clocks (PCLKA, PCLKB).
ICLK:PCLKA or PCLKB = N:1(N: integer)

Note 4. When using ESWM, set the PCLKA frequency to be less than the ICLK frequency.

Note 5. When using CANFD or CANFDG, set the PCLKE frequency to twice the PCLKA frequency.

Note 6. The following condition should be satisfied between the frequencies of the system clock (ICLK) and external bus clock (BCLK).
ICLK:BCLK = N:1 (N: integer)

Note 7. The following condition should be satisfied between the frequencies of the system clock (ICLK) and MRAM peripheral bus clock (MRPCLK).
ICLK:MRPCLK = N:1 (N: integer)

Note 8. The following condition should be satisfied between the frequencies of the CPU clock (CPUCLK0) and system clock (ICLK).
CPUCLK0:ICLK = N:1 (N: integer)

Note 9. The following conditions should be satisfied between the frequencies of the MRAM code bus clock (MRICLK), CPU clock (CPUCLK0), and system clock (ICLK).
MRICLK:ICLK = N:1 (N: integer), CPUCLK0:MRICLK = N:1 (N: integer)

Note 10. The frequency of the MRAM peripheral bus clock (MRPCLK) is limited by the setting of the extra MRAM frequency notifications register (MREFREQ). See section 65, MRAM.

The SCKDIVCR register is used to select the frequencies of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE), MRAM peripheral bus clock (MRPCLK), and external bus clock (BCLK).

9.2.3 System Clock Frequency Division Control Register 2 (SCKDIVCR2)

Address(es): SYSC.SCKDIVCR2 8701 9024h,
SYSC_NS.SCKDIVCR2 0701 9024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRICK[3:0]				—	—	—	—	—	—	—	—	CPUCK0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CPUCK0[3:0]	CPU Clock (CPUCLK0) Select*1	b3 b0 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W
b11 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	MRICK[3:0]	MRAM Code Bus Clock (MRICLK) Select*2, *3	b15 b12 0 0 0 0: Division by 1 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 8 0 1 0 0: Division by 16 0 1 0 1: Division by 32 0 1 1 0: Division by 64 1 0 0 0: Division by 3 1 0 0 1: Division by 6 1 0 1 0: Division by 12 1 0 1 1: Division by 24 Settings other than those listed above are prohibited.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Restrictions on clock frequency settings:
CPUCLK0 ≥ ICLK, MRICLK ≥ ICLK, CPUCLK0 ≥ MRICLK, ICLK ≥ MRPCLK, ICLK ≥ PCLKA ≥ PCLKB, ICLK ≥ BCLK, GPTCLK ≥ PCLKA, PCLKD ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratios (N: integer up to 64):
CPUCLK0:ICLK = N:1, MRICLK:ICLK = N:1, CPUCLK0:MRICLK = N:1, ICLK:MRPCLK = N:1, ICLK:BCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N, ICLK:PCLKE = N:1 or 1:N
When one clock is set to division by 3, 6, 12, or 24, do not set the other clocks to division by 2, 4, 8, 16, 32, or 64 by the SCKDIVCR and SCKDIVCR2 registers.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note 1. The following condition should be satisfied between the frequencies of the CPU clock (CPUCLK0) and system clock (ICLK).
CPUCLK0:ICLK = N:1 (N: integer)

Note 2. The following conditions should be satisfied between the frequencies of the MRAM code bus clock (MRICLK), CPU clock (CPUCLK0), and system clock (ICLK).
MRICLK:ICLK = N:1 (N: integer), CPUCLK0:MRICLK = N:1 (N: integer)

Note 3. The frequency of the MRAM code bus clock (MRICLK) is limited by the setting of the code MRAM frequency notifications register (MRCFREQ). See section 65, MRAM.

The SCKDIVCR2 register is used to select the frequencies of the CPU clock (CPUCLK0) and MRAM code bus clock (MRICLK).

9.2.4 System Register Access Status Register (SYRACCR)

Address(es): SYSC.SYRACCR 8701 90CCh,
SYSC_NS.SYRACCR 0701 90CCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BUSY
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BUSY	Access Ready State Monitoring Flag	0: Ready for read/write access 1: Writing in progress	R
b7 to b1	—	Reserved	These bits are read as 0.	R

The SYRACCR register is used to monitor whether the target registers can be accessed.
The target registers are LOCOCR and LOCOUTCR.

BUSY Flag (Access Ready State Monitoring Flag)

When this flag is 0, the target registers can be accessed.
Confirm that this flag is 0 before accessing the target registers.
When this flag is 1, the write access to one of the target registers is in progress. In this state, further writing to one of the target registers is ignored and the values read from the target registers are not guaranteed.

9.2.5 System Clock Source Control Register (SCKSCR)

Address(es): SYSC.SCKSCR 8701 9026h,
SYSC_NS.SCKSCR 0701 9026h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKSEL[2:0]	Clock Source Select	b2 b0 0 0 0: High-speed on-chip oscillator (HOCO) 0 0 1: Middle-speed on-chip oscillator (MOCO) (initial value) 0 1 0: Setting prohibited 0 1 1: Main clock oscillator (MOSC) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: PLL1 output clock P (PLL1P) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The SCKSCR register is used to select the clock source for the system clock.

CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the following clocks.

- CPU clock (CPUCLK0)
- MRAM code bus clock (MRICLK)
- MRAM peripheral bus clock (MRPCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE)
- External bus clock (BCLK)
- SDRAM clock (SDCLK)

Switching to a stopped clock source is prohibited.
The operating state of each clock source is controlled not only by the clock oscillator enable settings but also by the operating states of other functions and the operating mode. Some clock sources might be forcibly stopped depending on the operating mode.
Check the operating and stop conditions of each clock source and the operation state of clock sources in each operating mode, and do not select the clock source to be stopped in SCKSCR. Switch clock sources only when no interrupts from the WDT/IWDT are occurring.

9.2.6 PLL Clock Control Register (PLLCCR)

Address(es): SYSC.PLLCCR 8701 90ACh,
SYSC_NS.PLLCCR 0701 90ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLMUL[8]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PLLMUL[7:0]								PLLMULNF[1:0]		—	PLSRCSEL	—	—	PLIDIV[1:0]	
Value after reset:	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL1 Input Frequency Division Ratio Select*1	b1 b0 0 0: Division by 1 0 1: Division by 2 1 0: Division by 3 Settings other than those listed above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PLSRCSEL	PLL1 Clock Source Select	0: Main clock oscillator 1: HOCO*2	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	PLLMULNF[1:0]	PLL1 Frequency Multiplication Fractional Ratio Select*3	b7 b6 0 0: 0.00 (initial value) 0 1: 0.33 (1/3) 1 0: 0.66 (2/3) 1 1: 0.50 (1/2)	R/W
b16 to b8	PLLMUL[8:0]	PLL1 Frequency Multiplication Ratio Select*3	27h: × 40 (initial value) 28h: × 41 29h: × 42 : 12Ah: × 299 12Bh: × 300 Settings other than those listed above are prohibited.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note 1. The PLIDIV[1:0] bits set the frequency division ratio of the input clock source for the PLL1 circuit. Set these bits so that the input frequency (after division of the input frequency) is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources).

Note 2. Enable the FLL function when using USBCLK.

Note 3. The PLLMUL[8:0] and PLLMULNF[1:0] bits set the frequency multiplication ratio of the PLL1 circuit. Set these bits so that the VCO frequency is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources).

The PLLCCR register is used to set the operation of the PLL1 circuit.

Writing to this register is prohibited while the PLLCR.PLLSTP bit is 0 (PLL1 is operating).

9.2.7 PLL Clock Control Register 2 (PLLCCR2)

Address(es): SYSC.PLLCCR2 8701 904Ch,
SYSC_NS.PLLCCR2 0701 904Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PLODIVR[3:0]				PLODIVQ[3:0]				PLODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PLODIVP[3:0]	PLL1 Output Frequency Division Ratio Select for Output Clock P*1	b3 b0 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 1 1 1: Division by 16 Settings other than those listed above are prohibited.	R/W
b7 to b4	PLODIVQ[3:0]	PLL1 Output Frequency Division Ratio Select for Output Clock Q*1	b7 b4 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 0: Division by 5 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 0 0 0: Division by 9 Settings other than those listed above are prohibited.	R/W
b11 to b8	PLODIVR[3:0]	PLL1 Output Frequency Division Ratio Select for Output Clock R*1	b11 b8 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 0: Division by 5 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 0 0 0: Division by 9 Settings other than those listed above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note 1. The PLODIVn[3:0] bits set the output frequency division ratio of each output clock n in the PLL1 circuit. Set these bits so that PLL1 output clock n is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources). (n = P, Q, or R)

The PLLCCR2 register is used to set the operation of the PLL1 circuit. Writing to this register is prohibited while the PLLCR.PLLSTP bit is 0 (PLL1 is operating).

Even if only one of the PLL1 output clocks P, Q, and R is used, PLLCCR2 must be set so that the frequency of PLL1 output clock n is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources). (n = P, Q, or R)

9.2.8 PLL Control Register (PLLCR)

Address(es): SYSC.PLLCR 8701 902Ah,
SYSC_NS.PLLCR 0701 902Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLLSTP	PLL1 Stop Control	0: PLL1 is started. 1: PLL1 is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The PLLCR register is used to control the operation of the PLL1 circuit.

PLLSTP Bit (PLL1 Stop Control)

This bit starts or stops the operation of the PLL1 circuit.
The PLL1 clock source is selectable as the main clock oscillator and HOCO.
Selecting the main clock oscillator as the PLL1 clock source by the PLLCCR.PLSRCSEL bit requires setting of the main clock oscillator wait control register (MOSCWTCR).
After the value of the PLLSTP bit has been changed to 0, confirm that the OSCSF.PLLSF flag has been set to 1 before starting to use the PLL1 clock. That is, PLL1 requires a certain amount of time for oscillation to become stable after it starts operation. A certain amount of time is also required for oscillation to stop after the PLL1 operation is stopped.
In addition, note the following when starting or stopping PLL1 by the PLLSTP bit.

- After stopping PLL1, confirm that the OSCSF.PLLSF flag is 0 before restarting PLL1.
- Confirm that PLL1 is operating and that the OSCSF.PLLSF flag is 1 before stopping PLL1.
- Regardless of whether PLL1 is selected as the system clock source, confirm that the OSCSF.PLLSF flag is 1 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the PLL1 operation is started.
- Confirm that the OSCSF.PLLSF flag is 0 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the PLL1 operation is stopped.

Writing 1 to the PLLSTP bit is prohibited while SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1 output clock P (PLL1P)).
Confirm the following conditions before writing 0 to the PLLSTP bit.

- When PLL1 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is operating)
- When PLL1 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is operating)

9.2.9 PLL2 Clock Control Register (PLL2CCR)

Address(es): SYSC.PLL2CCR 8701 90C8h,
SYSC_NS.PLL2CCR 0701 90C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2MUL[8]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PLL2MUL[7:0]								PLL2MULNF[1:0]		—	PL2SRCSEL	—	—	PL2IDIV[1:0]	
Value after reset:	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PL2IDIV[1:0]	PLL2 Input Frequency Division Ratio Select*1	b1 b0 0 0: Division by 1 (initial value) 0 1: Division by 2 1 0: Division by 3 Settings other than those listed above are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PL2SRCSEL	PLL2 Clock Source Select	0: Main clock oscillator 1: HOCO*2	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	PLL2MULNF[1:0]	PLL2 Frequency Multiplication Fractional Ratio Select*3	b7 b6 0 0: 0.00 (initial value) 0 1: 0.33 (1/3) 1 0: 0.66 (2/3) 1 1: 0.50 (1/2)	R/W
b16 to b8	PLL2MUL[8:0]	PLL2 Frequency Multiplication Ratio Select*3	27h: × 40 (initial value) 28h: × 41 29h: × 42 : 12Ah: × 299 12Bh: × 300 Settings other than those listed above are prohibited.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. The PL2IDIV[1:0] bits set the frequency division ratio of the input clock source for the PLL2 circuit. Set these bits so that the input frequency (after division of the input frequency) is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources).
Note 2. Be sure to enable the FLL function when using USBCLK.
Note 3. The PLL2MUL[8:0] and PLL2MULNF[1:0] bits set the frequency multiplication ratio of the PLL2 circuit. Set these bits so that the VCO frequency is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources).

The PLL2CCR register is used to set the operation of the PLL2 circuit.
Writing to this register is prohibited while the PLL2CR.PLL2STP bit is 0 (PLL2 is operating).

9.2.10 PLL2 Clock Control Register 2 (PLL2CCR2)

Address(es): SYSC.PLL2CCR2 8701 904Eh,
SYSC_NS.PLL2CCR2 0701 904Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PL2ODIVR[3:0]				PL2ODIVQ[3:0]				PL2ODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PL2ODIVP[3:0]	PLL2 Output Frequency Division Ratio Select for Output Clock P*1	b3 b0 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 1 1 1: Division by 16 Settings other than those listed above are prohibited.	R/W
b7 to b4	PL2ODIVQ[3:0]	PLL2 Output Frequency Division Ratio Select for Output Clock Q*1	b7 b4 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 0: Division by 5 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 0 0 0: Division by 9 Settings other than those listed above are prohibited.	R/W
b11 to b8	PL2ODIVR[3:0]	PLL2 Output Frequency Division Ratio Select for Output Clock R*1	b11 b8 0 0 0 1: Division by 2 0 0 1 0: Division by 3 0 0 1 1: Division by 4 0 1 0 0: Division by 5 0 1 0 1: Division by 6 (initial value) 0 1 1 1: Division by 8 1 0 0 0: Division by 9 Settings other than those listed above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note 1. The PLODIVn[3:0] bits set the output frequency division ratio of each output clock n in the PLL2 circuit. Set these bits so that PLL2 output clock n is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources). (n = P, Q, or R)

The PLL2CCR2 register is used to set the operation of the PLL2 circuit. Writing to this register is prohibited while the PLL2CR.PLL2STP bit is 0 (PLL2 is operating).

Even if only one of the PLL2 output clocks P, Q, and R is used, PLL2CCR2 must be set so that the frequency of PLL2 output clock n is within the range of the specifications shown in Table 9.1, Specifications of the Clock Generation Circuit (Clock Sources). (n = P, Q, or R)

9.2.11 PLL2 Control Register (PLL2CR)

Address(es): SYSC.PLL2CR 8701 904Ah,
SYSC_NS.PLL2CR 0701 904Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLL2STP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLL2STP	PLL2 Stop Control	0: PLL2 is started. 1: PLL2 is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The PLL2CR register is used to control the operation of the PLL2 circuit.

PLL2STP Bit (PLL2 Stop Control)

This bit starts or stops the operation of the PLL2 circuit.
The PLL2 clock source is selectable as the main clock oscillator and HOCO.
Selecting the main clock oscillator as the PLL2 clock source by the PLL2CCR.PL2SRCSEL bit requires setting of the main clock oscillator wait control register (MOSCWTCR).
After the value of the PLL2STP bit has been changed to 0, confirm that the OSCSF.PLL2SF flag has been set to 1 before starting to use the PLL2 clock. That is, PLL2 requires a certain amount of time for stabilization after it starts operation. A certain amount of time is also required for oscillation to stop after the PLL2 operation is stopped.
In addition, note the following when starting or stopping PLL2 by the PLL2STP bit.

- After stopping PLL2, confirm that the OSCSF.PLL2SF flag is 0 before restarting PLL2.
- Confirm that PLL2 is operating and that the OSCSF.PLL2SF flag is 1 before stopping PLL2.
- Confirm that the OSCSF.PLL2SF flag is 1 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the PLL2 operation is started.
- Confirm that the OSCSF.PLL2SF flag is 0 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the PLL2 operation is stopped.

Confirm the following conditions before writing 0 to the PLL2STP bit.

- When PLL2 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is operating)
- When PLL2 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is operating)

9.2.12 External Bus Clock Control Register (BCKCR)

Address(es): SYSC.BCKCR 8701 9030h,
SYSC_NS.BCKCR 0701 9030h

	b7	b6	b5	b4	b3	b2	b1	b0
	EBCKASEL	—	—	—	—	—	—	BCLKDIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BCLKDIV	EBCLK Pin Output Select	0: The clock selected by the EBCKASEL bit is output without change. 1: The clock selected by the EBCKASEL bit is divided by 2 before it is output.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	EBCKASEL	External Bus Asynchronous Source Select*1	0: Synchronous (BCLK) 1: Asynchronous (BCLKA)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Before changing this bit, set BCKACR.BCKACKSREQ = 1 and then change the value while BCKACR.BCKACKSRDY = 1.

The BCKCR register is used to control the external bus clocks.
When the asynchronous clock (BCLKA) is selected for the external bus clock, the setting of the BCKACR register is also required. For details, see section 9.2.61, Asynchronous External Bus Clock Control Register (BCKACR).

BCLKDIV Bit (EBCLK Pin Output Select)

This bit selects the clock for output from the EBCLK pin.
Either the clock specified by the EBCKASEL bit or that clock with its frequency divided by two can be selected.
Set the BCLKDIV bit to 0 when BCLKA is selected by setting the EBCKASEL bit to 1.

EBCKASEL Bit (External Bus Asynchronous Source Select)

This bit selects the clock source for external bus, EBCLK, and SDCLK. Either BCLK or BCLKA can be selected, with division ratios set by the SCKDIVCR.BCK[2:0] bits and the BCKADIVCR.BCKACKDIV[3:0] bits, respectively.

9.2.13 Main Clock Oscillator Control Register (MOSCCR)

Address(es): SYSC.MOSCCR 8701 9032h,
SYSC_NS.MOSCCR 0701 9032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: The main clock oscillator is started (operating).*1 1: The main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Set the MOMCR register before setting the MOSTP bit to 0.

The MOSCCR register is used to control the main clock oscillator.

MOSTP Bit (Main Clock Oscillator Stop)

This bit starts or stops the operation of the main clock oscillator.
After the value of this bit has been changed, read the bit and check that the value has been updated before executing subsequent instructions.
To use the main clock oscillator, set the main clock oscillator mode oscillation control register (MOMCR) and the main clock oscillator wait control register (MOSCWTCR) before setting the MOSTP bit to 0.
After the value of the MOSTP bit has been changed to 0, confirm that the OSCSF.MOSCSF flag has been set to 1 before starting to use the main clock oscillator. That is, the main clock oscillator requires a certain amount of time for oscillation to become stable after it is set to start operation. A certain amount of time is also required for oscillation to stop after the main clock oscillator operation is stopped. In addition, note the following when starting or stopping operation.

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF flag is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCSF.MOSCSF flag is 1 before stopping the main clock oscillator.
- Regardless of whether the main clock oscillator is selected as the system clock source, confirm that the OSCSF.MOSCSF flag is 1 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode.
- Confirm that the OSCSF.MOSCSF flag is 0 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the main clock oscillator operation is stopped.

Writing 1 to the MOSTP bit is prohibited under any of the following conditions.

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 clock source = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 0 (PLL1 clock source = MOSC) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 0 (PLL2 clock source = MOSC) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

9.2.14 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): SYSC.SOSCCR 8701 9C00h,
SYSC_NS.SOSCCR 0701 9C00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: The sub-clock oscillator is started (operating).*1 1: The sub-clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Set the SOMCR register before setting the SOSTP bit to 0.

The SOSCCR register is used to control the sub-clock oscillator.
This register is only initialized by the backup domain reset.

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit starts or stops the operation of the sub-clock oscillator. After the value of this bit has been changed, read the bit and check that the value has been updated before executing subsequent instructions.
To use the sub-clock oscillator, set the sub-clock oscillator mode control register (SOMCR) before setting the SOSTP bit to 0.
When the external clock input is to be used, set the SOMCR.SOSEL bit to 1 to select the external clock input, wait for 50 µs, and then set the SOSTP bit to 0.
Note the following when starting or stopping operation.

- After stopping the sub-clock oscillator, allow a stop interval of at least five cycles of the SOSC clock before restarting the oscillator.
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization wait time ($t_{SUBOSCWT}$) has elapsed.
- Regardless of whether the sub-clock oscillator is selected as the system clock source, confirm that oscillation of the sub-clock is stable before executing a WAIT instruction to place the MCU in software standby or deep software standby mode.
- Wait for at least three cycles of the SOSC clock before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the sub-clock oscillator operation is stopped.

Writing 1 to the SOSTP bit is prohibited under the following condition.

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

9.2.15 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): SYSC.LOCOCR 8701 9400h,
SYSC_NS.LOCOCR 0701 9400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCSTP	LOCO Stop	0: The LOCO is started (operating). 1: The LOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note: Confirm that the SYRACCR.BUSY bit is 0 before accessing the LOCOCR register. If SYRACCR.BUSY is 1, write access is ignored and the read value is not guaranteed.

The LOCOCR register is used to control the LOCO.

LCSTP Bit (LOCO Stop)

This bit starts or stops the operation of the LOCO.

After the value of this bit has been changed, read the bit and check that the value has been updated before executing subsequent instructions.

There are restrictions on the operation and stopping of the LOCO. For details, see section 9.5, Low-Speed On-Chip Oscillator (LOCO).

9.2.16 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): SYSC.HOCOCR 8701 9036h,
SYSC_NS.HOCOCR 0701 9036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HCSTP	HOCO Stop	0: The HOCO is started (operating).*2, *3 1: The HOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. The value of the HCSTP bit after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.
Note 2. When the HOCO is used (HCSTP = 0), set the OFS1.HOCOFRQ0[2:0] bits to an optimal value.
Note 3. The value of the OFS1.HOCOFRQ0[2:0] bits is automatically transferred to the HOCOCR2.HCFRQ0[2:0] bits after a reset, therefore, the HOCO frequency can be changed by setting the HOCOCR2.HCFRQ0[2:0] bits.

The HOCOCR register is used to control the HOCO.

HCSTP Bit (HOCO Stop)

This bit starts or stops the operation of the HOCO.
After the value of this bit has been changed to 0 to start the HOCO, confirm that the OSCSF.HOCOSF flag has been set to 1 before starting to use the HOCO clock. When the OFS1.HOCOEN bit is 0, confirm that the OSCSF.HOCOSF flag is also set to 1 before using the HOCO clock. That is, the HOCO requires a certain amount of time for oscillation to become stable after it is set to start operation. A certain amount of wait time is also required for oscillation to stop after the HOCO clock is set to stop.
Note the following when starting or stopping operation.

- After stopping the HOCO clock, wait for at least the HOCO stop width time and confirm that the OSCSF.HOCOSF flag is 0 before restarting the HOCO clock. For the HOCO stop width time, see section 67, Electrical Characteristics.
- Confirm that the HOCO clock is operating and that the OSCSF.HOCOSF flag is 1 before stopping the HOCO.
- Regardless of whether the HOCO clock is selected as the system clock, confirm that the HOCO clock is stable before executing a WAIT instruction to place the MCU in software standby or deep software standby mode.
- Confirm that the OSCSF.HOCOSF flag is 0 before executing a WAIT instruction to place the MCU in software standby or deep software standby mode after the HOCO operation is stopped.

Writing 1 to the HCSTP bit is prohibited under any of the following conditions.

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO)
- PLLCCR.PLSRCSEL = 1 (PLL1 clock source = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 1 (PLL1 clock source = HOCO) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 1 (PLL2 clock source = HOCO) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

9.2.17 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): SYSC.HOCOCR2 8701 9037h,
SYSC_NS.HOCOCR2 0701 9037h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HCFRQ0[2:0]		
Value after reset:	0	0	0	0	0	0/1*1	0/1*1	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	HCFRQ0[2:0]	HOCO Frequency Setting 0	b2 b0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. The value after a reset depends on the setting of the OFS1.HOCOFRQ0[2:0] bits.

The HOCOCR2 register is used to control the HOCO clock.
Writing to the HOCOCR2 register is prohibited when the HOCOCR.HCSTP bit is 0 (making the HOCO run).

HCFRQ0[2:0] Bits (HOCO Frequency Setting 0)

These bits select the frequency of the HOCO.

9.2.18 Middle-Speed On-Chip Oscillator Control Register (MOCOCR)

Address(es): SYSC.MOCOCR 8701 9038h,
SYSC_NS.MOCOCR 0701 9038h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCSTP	MOCO Stop	0: The MOCO is started (operating). 1: The MOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The MOCOCR register is used to control the MOCO.

MCSTP Bit (MOCO Stop)

This bit starts or stops the operation of the MOCO.

After the value of this bit has been changed, read the bit and check that the value has been updated before executing subsequent instructions.

There are restrictions on MOCO operation conditions, MOCO stop conditions, and the setting of the MCSTP bit. For details, see section 9.6, Middle-Speed On-Chip Oscillator (MOCO).

9.2.19 FLL Control Register 1 (FLLCR1)

Address(es): SYSC.FLLCR1 8701 9039h,
SYSC_NS.FLLCR1 0701 9039h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FLLEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLLEN	FLL Enable	0: The FLL function is disabled. 1: The FLL function is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Stop the HOCO (HOCOCR.HCSTP = 1) before modifying the FLLCR1.FLLEN bit.
Note: The sub-clock oscillator needs to be operating and its oscillation needs to be stable while FLL is enabled (FLLCR1.FLLEN = 1).
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The FLLCR1 register controls the frequency correction function (frequency locked loop (FLL) function) of the HOCO.

FLLEN Bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

The accuracy of clock frequency is guaranteed after FLL has become stable. Stabilization of FLL can be checked through measurement of the frequency by the clock frequency accuracy measurement circuit (CAC) after stabilization of the HOCO.

The FLL function needs to be disabled by setting the FLLEN bit to 0 before placing the MCU in software standby mode. Table 9.4 shows sample procedures for setting the FLL function in some situations.

Table 9.4 Procedures for Setting the FLL Function (1/2)

Procedure		Operation
After release from the reset state or deep software standby mode	1	Start (After release from the reset state or deep software standby mode)
	2	Set the multiplication ratio for FLL (FLLCR2.FLLCNTL[10:0]).
	3	Enable FLL (FLLCR1.FLLEN = 1).*1
	4	Enable the HOCO (HOCOCR.HCSTP = 0).
	5	Wait until FLL has become stable (tFLLWT).
	6	Check that the HOCO has become stable (OSCSF.HOCOSF = 1).
	7	End (The HOCO can be used.)
Transition to or release from software standby mode	1	Start (FLL is being used.)
	2	Stop the HOCO (HOCOCR.HCSTP = 1).*2
	3	Disable FLL (FLLCR1.FLLEN = 0).
	4	Execute a WAIT instruction.
	5	Software standby mode
	6	Release from software standby mode
	7	Enable FLL (FLLCR1.FLLEN = 1).

Table 9.4 Procedures for Setting the FLL Function (2/2)

Procedure		Operation
Transition to or release from software standby mode	8	Enable the HOCO (HOCOCR.HCSTP = 0).
	9	Wait until FLL has become stable (tFLLWT).
	10	Check that the HOCO has become stable (OSCSF.HOCOSF = 1).
	11	End (The HOCO can be used.)

Note 1. Enable the FLL function while oscillation of the SOSC is stable.
Note 2. If the HOCO is used as the source of the system clock or PLL clocks, change the source to a different clock before stopping the HOCO.

9.2.20 FLL Control Register 2 (FLLCR2)

Address(es): SYSC.FLLCR2 8701 903Ah,
SYSC_NS.FLLCR2 0701 903Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLLCNTL[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FLLCNTL[10:0]	FLL Multiplication Control	Set these bits to 1E8h when the OFS1.HOCOFRQ0[2:0] bits are set to 000b (16 MHz) or 100b (32 MHz). Set to 225h when the OFS1.HOCOFRQ0[2:0] bits are set to 001b (18 MHz). Set to 262h when the OFS1.HOCOFRQ0[2:0] bits are set to 010b (20 MHz). Set to 1E8h when the OFS1.HOCOFRQ0[2:0] bits are set to 111b (48 MHz). Settings other than those listed above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note: The value of the OFS1.HOCOFRQ0[2:0] bits is automatically transferred to the HOCOCR2.HCFRQ0[2:0] bits after a reset, therefore, the HOCO frequency can be changed by setting the HOCOCR2.HCFRQ0[2:0] bits.

The FLLCR2 register controls the FLL function of the HOCO.

FLLCNTL[10:0] Bits (FLL Multiplication Control)

These bits select the frequency multiplication ratio of the FLL reference clock.
Set these bits before enabling the FLL function (FLLCR1.FLLEN = 1).

9.2.21 Oscillation Stabilization Flag Register (OSCSF)

Address(es): SYSC.OSCSF 8701 903Ch,
SYSC_NS.OSCSF 0701 903Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PLL2SF	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOSF	HOCO Clock Oscillation Stabilization Flag	0: The HOCO clock is stopped or oscillation of the clock has not yet become stable. 1: Oscillation of the HOCO clock is stable and the clock is available for use as the system clock source.	R
b2, b1	—	Reserved	These bits are read as 0.	R
b3	MOSCSF	Main Clock Oscillation Stabilization Flag	0: The main clock oscillator is stopped or oscillation of the clock has not yet become stable*2 1: Oscillation of the main clock oscillator is stable and the clock is available for use as the system clock source.	R
b4	—	Reserved	This bit is read as 0.	R
b5	PLLSF	PLL1 Clock Oscillation Stabilization Flag	0: The PLL1 clock is stopped or oscillation of the clock has not yet become stable 1: Oscillation of the PLL1 clock is stable and the clock is available for use as the system clock source.	R
b6	PLL2SF	PLL2 Clock Oscillation Stabilization Flag	0: The PLL2 clock is stopped or oscillation of the clock has not yet become stable 1: Oscillation of the PLL2 clock is stable.	R
b7	—	Reserved	This bit is read as 0.	R

Note 1. The value after a reset depends on the setting of the OFS1.HOCOEN bit.
When OFS1.HOCOEN = 1 (HOCO is stopped), the value of HOCOSF after a reset is 0.
When OFS1.HOCOEN = 0 (HOCO is operating), the HOCOSF bit is set to 0 immediately after release from a reset but its value is set to 1 after the HOCO oscillation stabilization wait time has elapsed.

Note 2. This is true when an appropriate value is set in the main clock oscillator wait control register (MOSCWTCR). If the value (wait time) set in the register is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation has become stable.

This register is not controlled by the CGFSAR register.
The OSCSF register contains flags to indicate the states of operation of the counters in the oscillation stabilization wait circuits for the individual oscillators. The counters measure the wait times until the output clocks of the individual oscillators are supplied to the internal circuits after oscillation starts. Each oscillation stabilization flag being set to 1 indicates that the corresponding clock supply is stable and can be used in the related circuits.

HOCOSF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the wait time for the high-speed on-chip clock oscillator (HOCO).
When the OFS1.HOCOEN bit is set to 0, confirm that the OSCSF.HOCOSF flag is set to 1 before using the HOCO clock.
[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0, the HOCOSF flag is set to 1 after the fluctuation of the HOCO clock has fallen within a range of ±10%.

Note: To use the clock in the frequency fluctuation range specified in the electrical characteristics, wait until the oscillation stabilization wait time specified in the electrical characteristics has elapsed. (For details, see section 67, Electrical Characteristics).

Note: When the HOCOCR.HCSTP bit is set to 0, the HOCO clock is supplied to the MCU internal circuits regardless of the value of the HOCOSF flag.

[Clearing condition]

- The flag is cleared after oscillation of the HOCO has been stopped by setting the HOCOCR.HCSTP bit to 1 while the HOCO is operating.

Note: Even if the OFS1.HOCOEN bit has been set to 0, confirm that the OSCSF.HOCOSF flag is set to 1 before starting the use of the HOCO clock.

MOSCSF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 while the main clock oscillator is stopped, the MOSCSF flag is set after the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register has been counted and supply of the main clock in the MCU has started.

[Clearing condition]

- The flag is cleared when deactivation of the main clock oscillator is completed after the MOSCCR.MOSTP bit is set to 1 while the main clock oscillator is operating.

PLLSF Flag (PLL1 Clock Oscillation Stabilization Flag)

This flag indicates the oscillation stabilization status for the PLL1.

[Setting condition]

- After the PLLCR.PLLSTP bit is set to 0 while PLL1 is stopped, supply of the PLL1 clock begins after the clock has become stable.
  If oscillation of the clock source for PLL1 selected by the PLLCCR.PLSRCSEL bit is not stable when the PLLSTP bit is set to 0, wait until the clock becomes stable after oscillation of the clock source has become stable.

[Clearing condition]

- The flag is cleared when deactivation of PLL1 is completed after the PLLCR.PLLSTP bit is set to 1 while PLL1 is operating.

PLL2SF Flag (PLL2 Clock Oscillation Stabilization Flag)

This flag indicates the oscillation stabilization status for the PLL2.

[Setting condition]

- After the PLL2CR.PLL2STP bit is set to 0 while PLL2 is stopped, supply of the PLL2 clock begins after the clock has become stable.
  If oscillation of the clock source for PLL2 selected by the PLL2CCR.PL2SRCSEL bit is not stable when the PLL2STP bit is set to 0, wait until the clock becomes stable after oscillation of the clock source has become stable.

[Clearing condition]

- The flag is cleared when deactivation of PLL2 is completed after the PLL2CR.PLL2STP bit is set to 1 while PLL2 is operating.

9.2.22 Oscillator Monitor Register (OSCMONR)

Address(es): SYSC.OSCMONR 8701 9043h,
SYSC_NS.OSCMONR 0701 9043h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LOCOMON	MOCOMON	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0.	R
b1	MOCOMON	MOCO Operation Monitoring Flag	0: The MOCO is set to operate. 1: The MOCO is set to stop.	R
b2	LOCOMON	LOCO Operation Monitoring Flag	0: The LOCO is set to operate. 1: The LOCO is set to stop.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

This register is not controlled by the CGFSAR register.

MOCOMON Flag (MOCO Operation Monitoring Flag)

This flag indicates whether the MOCO is set to operate or stop.

[Clearing condition]

- The flag is cleared when the MOCO operating conditions are satisfied. For details, see section 9.6.1, MOCO Operating Conditions.

[Setting condition]

- The flag is set when the MOCO stop conditions are satisfied. For details, see section 9.6.2, MOCO Stop Conditions.

LOCOMON Flag (LOCO Operation Monitoring Flag)

This flag indicates whether the LOCO is set to operate or stop.

[Clearing condition]

- The flag is cleared when the LOCO operating conditions are satisfied. For details, see section 9.5.1, LOCO Operating Conditions.

[Setting condition]

- The flag is set when the LOCO stop conditions are satisfied. For details, see section 9.5.2, LOCO Stop Conditions.

9.2.23 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): SYSC.OSTDCR 8701 9040h,
SYSC_NS.OSTDCR 0701 9040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not reported to the POEG. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is reported to the POEG.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: The oscillation stop detection function is disabled. 1: The oscillation stop detection function is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The OSTDCR register is used to enable the main clock oscillation stop detection function.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables or disables the main clock oscillation stop detection function interrupt. It also controls whether main clock oscillation stop detection is reported to the POEG.
If the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) requires clearing, do this after clearing the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before setting the OSTDIE bit to 1 again. By reading an I/O register whose access time is defined with the number of PCLKB cycles, a wait time of at least two PCLKB cycles can be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function for the main clock.
Setting the OSTDE bit to 1 (enabled) changes the MOCO stop bit (MOCOCR.MCSTP) to 0 and the MOCO starts operation.
The MOCO clock cannot be stopped while the main clock oscillation stop detection function is enabled; writing 1 to the MOCOCR.MCSTP bit to stop the MOCO has no effect.
Writing 0 to the OSTDE bit has no effect while the oscillation stop detection flag of the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (stopping of main clock oscillation has been detected).

9.2.24 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): SYSC.OSTDSR 8701 9041h,
SYSC_NS.OSTDSR 0701 9041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Stopping of main clock oscillation has not been detected. 1: Stopping of main clock oscillation has been detected.	R/W*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Only 0 can be written. This flag can be cleared by reading 1 from it and then writing 0.

The OSTDSR register indicates the detection of stopping of the main clock oscillator.

OSTDF Flag (Oscillation Stop Detection Flag)

This flag indicates the state of the main clock oscillator. When this flag is 1, it indicates that stopping of main clock oscillation has been detected. Once stopping of oscillation is detected, the OSTDF flag is not set to 0 even though main clock oscillation is restarted. The flag can be cleared to 0 by reading 1 from it and then writing 0.
At least three ICLK cycles of wait time are necessary until the OSTDF flag is read as 0 after 0 is written to it.
If the OSTDF flag is cleared to 0 while main clock oscillation is stopped, the flag becomes 0 and then returns to 1.

The OSTDF flag cannot be set to 0 under either of the following conditions.

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 clock source = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)

The OSTDF flag should be set to 0 after the clock source is switched to a source other than the main clock oscillator or PLL1P.
[Setting condition]

- The flag is set when main clock oscillation is stopped while the OSTDCR.OSTDE bit is 1 (oscillation stop detection function is enabled).

[Clearing condition]

- The flag is cleared when 1 is read and then 0 is written while neither of the following conditions is satisfied.
  - SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
  - PLLCCR.PLSRCSEL = 0 (PLL1 clock source = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)

9.2.25 Sub-Clock Oscillation Stop Detection Control Register (SOSTDCR)

Address(es): SYSC.SOSTDCR 8701 9C04h,
SYSC_NS.SOSTDCR 0701 9C04h

	b7	b6	b5	b4	b3	b2	b1	b0
	SOSTDE	—	—	—	—	—	—	SOSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTDIE	Sub-Clock Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. 1: The oscillation stop detection interrupt is enabled.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SOSTDE	Sub-Clock Oscillation Stop Detection Function Enable	0: The oscillation stop detection function is disabled. 1: The oscillation stop detection function is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SOSTDCR register is used to enable the sub-clock oscillation stop detection function.
This register is only initialized by the backup domain reset.

SOSTDIE Bit (Sub-Clock Oscillation Stop Detection Interrupt Enable)

This bit enables or disables the sub-clock oscillation stop detection function interrupt.
If the sub-clock oscillation stop detection flag in the sub-clock oscillation stop detection status register (SOSTDSR.SOSTDF) requires clearing, do this after clearing the SOSTDIE bit to 0. Confirm that the SOSTDF flag has been cleared before setting the SOSTDIE bit to 1 again.

SOSTDE Bit (Sub-Clock Oscillation Stop Detection Function Enable)

This bit enables or disables the sub-clock oscillation stop detection function.
When the SOSTDE bit is set to 1 (enabled), the MOCO stop bit (MOCOCR.MCSTP) and LOCO stop bit (LOCOCR.LCSTP) are set to 0, and the MOCO and LOCO start operating.
Except in deep software standby mode and battery backup mode, the MOCO cannot be stopped while the sub-clock oscillation stop detection function is enabled. The LOCO cannot be stopped while the sub-clock oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit or LOCOCR.LCSTP bit has no effect while the sub-clock oscillation stop detection function is enabled.
Writing 0 to the SOSTDE bit has no effect while the sub-clock oscillation stop detection flag of the sub-clock oscillation stop detection status register (SOSTDSR.SOSTDF) is 1 (stopping of sub-clock oscillation has been detected).
To enable the sub-clock oscillation stop detection function, set the SOSTDCR.SOSTDE bit to 1 and then clear the SOSTDSR.SOSTDF flag to 0. Here, after setting the SOSTDCR.SOSTDE bit to 1, wait for the sub-clock oscillation stop detection circuit start-up wait time (t_{sosdup}) to elapse before clearing the SOSTDSR.SOSTDF flag. For the SOSTDSR register, see section 9.2.26, Sub-Clock Oscillation Stop Detection Status Register (SOSTDSR).
Set the system clock to other clock sources than the sub-clock oscillator before enabling the sub-clock oscillation stop detection function.
Set this bit only once. After it has been set to 1, setting it to 0 is prohibited.

9.2.26 Sub-Clock Oscillation Stop Detection Status Register (SOSTDSR)

Address(es): SYSC.SOSTDSR 8701 9C05h,
SYSC_NS.SOSTDSR 0701 9C05h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTDF	Sub-Clock Oscillation Stop Detection Flag	0: Stopping of sub-clock oscillation has not been detected. 1: Stopping of sub-clock oscillation has been detected.	R/W*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Only 0 can be written. This flag can be cleared by reading 1 from it and then writing 0.

The SOSTDSR register indicates the detection of stopping of the sub-clock oscillator.
This register is only initialized by the backup domain reset.

SOSTDF Flag (Sub-Clock Oscillation Stop Detection Flag)

This flag indicates the state of the sub-clock oscillator. When this flag is 1, it indicates that stopping of sub-clock oscillation has been detected.
Once stopping of oscillation is detected, the SOSTDF flag is not set to 0 even though sub-clock oscillation is restarted.
The flag can be cleared to 0 by reading 1 from it and then writing 0.
At least 3 PCLKB cycles of wait time are necessary until the SOSTDF flag is read as 0 after 0 is written to it.
If the SOSTDF flag is cleared to 0 while the sub-clock oscillation is stopped, the flag is immediately set to 1.
The SOSTDF flag cannot be set to 0 under the following condition.

- SCKSCR.CKSEL[2:0] = 100 (system clock source = SOSC)

[Setting condition]

- The flag is set when sub-clock oscillation is stopped while the SOSTDCR.SOSTDE bit is 1 (oscillation stop detection function is enabled).

[Clearing condition]

- The flag is cleared when 1 is read and then 0 is written while the value of the SCKSCR.CKSEL[2:0] bits is not 100b (system clock source = SOSC).

9.2.27 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): SYSC.MOSCWTCR 8701 90A2h,
SYSC_NS.MOSCWTCR 0701 90A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MSTS[3:0]			
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MSTS[3:0]	Main Clock Oscillator Stabilization Wait Time Setting	b3 b0 0 0 0 0: Wait time = 3 cycles (11.4 µs) 0 0 0 1: Wait time = 35 cycles (133.5 µs) 0 0 1 0: Wait time = 67 cycles (255.6 µs) 0 0 1 1: Wait time = 131 cycles (499.7 µs) 0 1 0 0: Wait time = 259 cycles (988.0 µs) 0 1 0 1: Wait time = 547 cycles (2086.6 µs) (initial value) 0 1 1 0: Wait time = 1059 cycles (4039.8 µs) 0 1 1 1: Wait time = 2147 cycles (8190.2 µs) 1 0 0 0: Wait time = 4291 cycles (16368.9 µs) 1 0 0 1: Wait time = 8163 cycles (31139.4 µs) Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

MSTS[3:0] Bits (Main Clock Oscillator Stabilization Wait Time Setting)

These bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to at least the stabilization time recommended by the manufacturer of the resonator. When an external input clock is used as the main clock, set these bits to 0000b because the oscillation stabilization time is not required.

The wait time set in these bits is counted by using the following equation.

1 cycle (µs) = 1/(fLOCO (MHz) × 8) = 1/(0.032768 × 8) = 3.81 µs (min.)

While waiting for the main clock oscillator to stabilize (when the MOSCCR.MOSTP bit is 0 and the OSCSF.MOSCSF flag is 0), the LOCO clock operates regardless of the value of the LOCOCR.LCSTP bit. After the specified wait time has elapsed, supply of the main clock starts in the MCU and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

9.2.28 Main Clock Oscillator Mode Oscillation Control Register (MOMCR)

Address(es): SYSC.MOMCR 8701 9A50h,
SYSC_NS.MOMCR 0701 9A50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	—	—	MODRV0[2:0]			—
Value after reset:	0	0	0	1	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	MODRV0[2:0]	Main Clock Oscillator Driving Capacity Switching	b3 b1 0 0 0: 8 MHz 0 1 1: 8 MHz to 24 MHz 1 0 1: 8 MHz to 48 MHz (initial value) Settings other than those listed above are prohibited.	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: The EXTAL and XTAL pins can be used as port pins. In the initial state, the pins are set as port pins.
Note: Set the MOSCCR.MOSTP bit to 1 (MOSC is stopped) before changing the value of this register.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

MODRV0[2:0] Bits (Main Clock Oscillator Driving Capacity Switching)

These bits switch the driving capacity of the main clock oscillator. These bits should normally be set to 101b. To reduce power consumption, set these bits to an appropriate value according to the frequency of the oscillator or resonator used. See section 67, Electrical Characteristics for current consumption.

MOSEL Bit (Main Clock Oscillator Switching)

This bit switches the source for the main clock oscillator.
For details, see section 9.3, Main Clock Oscillator.

9.2.29 Sub-Clock Oscillator Mode Control Register (SOMCR)

Address(es): SYSC.SOMCR 8701 9C01h,
SYSC_NS.SOMCR 0701 9C01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SOSEL	—	—	—	—	SODRV[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SODRV[1:0]	Sub-Clock Oscillator Driving Capacity Switching	b1 b0 0 0: Standard (12.5 pF) 0 1: Low-power mode 1 (9 pF) 1 0: Low-power mode 2 (7 pF) 1 1: Setting prohibited	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SOSEL	Sub-Clock Oscillator Switching	0: Resonator 1: External clock input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SOMCR register must be modified when the SOSCCR.SOSTP bit is 1 (SOSC is stopped). This register is only initialized by the backup domain reset.

SODRV[1:0] Bits (Sub-Clock Oscillator Driving Capacity Switching)

These bits switch the driving capacity of the sub-clock oscillator.

SOSEL Bit (Sub-Clock Oscillator Switching)

This bit switches the oscillation source for the sub-clock oscillator.

For details, see section 9.4, Sub-Clock Oscillator.

9.2.30 Clock Output Control Register (CKOCR)

Address(es): SYSC.CKOCR 8701 903Eh,
SYSC_NS.CKOCR 0701 903Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKOEN	CKODIV[2:0]			CKOSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CKOSEL[3:0]	Clock Output Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: MOSC 0 1 0 0: SOSC 0 1 0 1: ICLK 1 0 0 0: PLL1P 1 1 0 1: PLL2P 1 1 1 0: PLL1Q 1 1 1 1: PLL2Q Settings other than those listed above are prohibited.	R/W
b6 to b4	CKODIV[2:0]	Clock Output Frequency Division Ratio	b6 b4 0 0 0: Division by 1 0 0 1: Division by 2 0 1 0: Division by 4 0 1 1: Division by 8 1 0 0: Division by 16 1 0 1: Division by 32 1 1 0: Division by 64 1 1 1: Division by 128	R/W
b7	CKOEN	Clock Output Enable	0: Clock output is disabled. 1: Clock output is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

CKOSEL[3:0] Bits (Clock Output Source Select)

These bits select the source of the clock to be output from the CLKOUT pin. Set the CKOEN bit to 0 before changing the clock source.

CKODIV[2:0] Bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits set the frequency division ratio of the clock selected by the CKOSEL[3:0] bits. Set the CKOEN bit to 0 before changing the division ratio. Set these bits so that the frequency of the clock output from the CLKOUT pin is within the range of the electrical characteristics. For details, see section 67, Electrical Characteristics.

CKOEN Bit (Clock Output Enable)

The CKOEN bit enables or disables output from the CLKOUT pin.

Setting the bit to 1 leads to output of the selected clock. Setting the bit to 0 drives the output to the low level.

Confirm that the source clock for the output clock selected by the CKOSEL[3:0] bits is stable before changing the value of this bit. Otherwise, a glitch may be generated in the output.

To stop output from the CLKOUT pin in software standby mode or deep software standby mode, set this bit to 0 before entering the mode.

9.2.31 External Bus Clock Output Control Register (EBCKOCR)

Address(es): SYSC.EBCKOCR 8701 9052h,
SYSC_NS.EBCKOCR 0701 9052h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EBCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EBCKOEN	EBCLK Pin Output Control	0: Output from the EBCLK pin is disabled and the pin is fixed to the high level. 1: Output from the EBCLK pin is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

9.2.32 SDRAM Clock Output Control Register (SDCKOCR)

Address(es): SYSC.SDCKOCR 8701 9053h,
SYSC_NS.SDCKOCR 0701 9053h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDCKOEN
Value after reset:	0	0	0	0	0	0	0	0

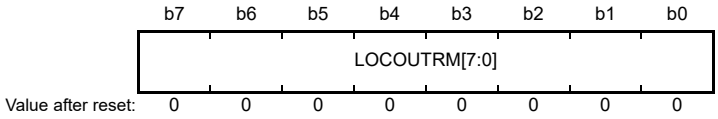
Bit	Symbol	Bit Name	Description	R/W
b0	SDCKOEN	SDCLK Pin Output Control	0: Output from the SDCLK pin is disabled and the pin is fixed to the high level. 1: Output from the SDCLK pin is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

9.2.33 LOCO User Trimming Control Register (LOCOUTCR)

Address(es): SYSC.LOCOUTCR 8701 9402h,
SYSC_NS.LOCOUTCR 0701 9402h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOCOUTRM[7:0]	LOCO User Trimming	80h: –128 81h: –127 : FFh: –1 00h: Center code 01h: +1 : 7Eh: +126 7Fh: +127	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note: Confirm that the SYRACCR.BUSY bit is 0 before accessing the LOCOUTCR register. If SYRACCR.BUSY is 1, write access is ignored and the read value is not guaranteed.

Note: MCU operation is not guaranteed if the LOCOUTCR register is set to a value that causes the LOCO frequency to be outside the specification range.

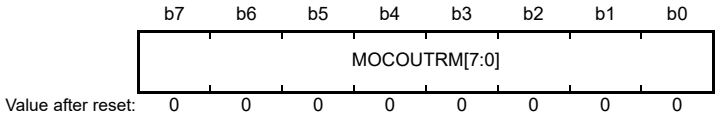
Note: A certain amount of time is required for the frequency to become stable after the value of the LOCOUTCR register is changed. This is the same amount of time as the oscillation stabilization time after operation begins.

The value of the LOCOUTCR register is added to the original LOCO trimming data.

Do not change the LOCOUTCR value while the RTC is operating.

9.2.34 MOCO User Trimming Control Register (MOCOUTCR)

Address(es): SYSC.MOCOUTCR 8701 9061h,
SYSC_NS.MOCOUTCR 0701 9061h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	MOCOUTRM[7:0]	MOCO User Trimming	80h: –128 81h: –127 : FFh: –1 00h: Center code 01h: +1 : 7Eh: +126 7Fh: +127	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

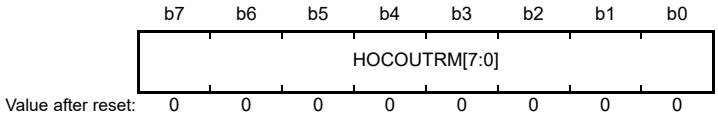
Note: MCU operation is not guaranteed if the MOCOUTCR register is set to a value that causes the MOCO frequency to be outside the specification range.

Note: A certain amount of time is required for the frequency to become stable after the value of the MOCOUTCR register is changed. This is the same amount of time as the oscillation stabilization time after operation begins.

The value of the MOCOUTCR register is added to the original MOCO trimming data.

9.2.35 HOCO User Trimming Control Register (HOCOUTCR)

Address(es): SYSC.HOCOUTCR 8701 9062h,
SYSC_NS.HOCOUTCR 0701 9062h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	HOCOUTRM[7:0]	HOCO User Trimming	80h: –128 81h: –127 : FFh: –1 00h: Center code 01h: +1 : 7Eh: +126 7Fh: +127	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Note: MCU operation is not guaranteed if the HOCOUTCR register is set to a value that causes the HOCO frequency to be outside the specification range.

Note: A certain amount of time is required for the frequency to become stable after the value of the HOCOUTCR register is changed. This is the same amount of time as the oscillation stabilization time after operation begins.

Note: Set this register to 00h when the FLL function is enabled (FLLCR1.FLLEN = 1).

The value of the HOCOUTCR register is added to the original HOCO trimming data.

9.2.36 Main Clock Oscillator Standby Control Register (MOSCSCR)

Address(es): SYSC.MOSCSCR 8701 907Ch,
SYSC_NS.MOSCSCR 0701 907Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSCSOKP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOSCSOKP	Main Clock Oscillator Standby Oscillation Keep Select	0: Disabled 1: Enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

Change the value of the MOSCSCR register while the MOSCCR.MOSTP bit is 1 (MOSC is stopped).

MOSCSOKP Bit (Main Clock Oscillator Standby Oscillation Keep Select)

This bit controls whether the main clock oscillator continues oscillation or stops in software standby mode.

If this bit is 0 (disabled) and the MOSCCR.MOSTP bit is 0 (MOSC is operating), the main clock oscillator stops in software standby mode.

If this bit is 1 (enabled) and the MOSCCR.MOSTP bit is 0 (MOSC is operating), the main clock oscillator keeps oscillating in software standby mode.

If the MOSCCR.MOSTP bit is 1 (MOSC is stopped), the main clock oscillator stops in software standby mode regardless of the setting of the MOSCSOKP bit.

9.2.37 High-Speed On-Chip Oscillator Standby Control Register (HOCOSCR)

Address(es): SYSC.HOCOSCR 8701 907Dh,
SYSC_NS.HOCOSCR 0701 907Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOSOKP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOSOKP	HOCO Standby Oscillation Keep Select	0: Disabled 1: Enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

HOCOSOKP Bit (HOCO Standby Oscillation Keep Select)

This bit controls whether the HOCO continues oscillation or stops in software standby mode.

If this bit is 0 (disabled) and the HOCOCR.HCSTP bit is 0 (HOCO is operating), the HOCO stops in software standby mode.

If this bit is 1 (enabled) and the HOCOCR.HCSTP bit is 0 (HOCO is operating), the HOCO keeps oscillating in software standby mode.

If the HOCOCR.HCSTP bit is 1 (HOCO is stopped), the HOCO stops in software standby mode regardless of the setting of the HOCOSOKP bit.

9.2.38 Middle-Speed On-Chip Oscillator Standby Control Register (MOCOSCR)

Address(es): SYSC.MOCOSCR 8701 9084h,
SYSC_NS.MOCOSCR 0701 9084h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOCOSOKP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOCOSOKP	MOCO Standby Oscillation Keep Select	0: Disabled 1: Enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

MOCOSOKP Bit (MOCO Standby Oscillation Keep Select)

This bit controls the MOCO so that it keeps oscillating in software standby mode.

If this bit is 0 (disabled) and the MOCOCR.MCSTP bit is 0 (MOCO is operating), the MOCO stops in software standby mode.

If this bit is 1 (enabled) and the MOCOCR.MCSTP bit is 0 (MOCO is operating), the MOCO keeps oscillating in software standby mode.

9.2.39 USB Clock Frequency Division Control Register (USBCKDIVCR)

Address(es): SYSC.USBCKDIVCR 8701 906Ch,
SYSC_NS.USBCKDIVCR 0701 906Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	USBCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	USBCKDIV[3:0]	USB Clock (USBCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The USBCKDIVCR register is used to control the USB clock (USBCLK).

USBCKDIV[3:0] Bits (USB Clock (USBCLK) Frequency Division Ratio Select)

These bits select the frequency of USBCLK. Change the value of these bits while the USBCKCR.USBCKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB11 = 1, and MSTPCRB.MSTPB12 = 1 before setting USBCKCR.USBCKSREQ = 1.

9.2.40 XSPI Clock Frequency Division Control Register (XSPICKDIVCR)

Address(es): SYSC.XSPICKDIVCR 8701 906Dh,
SYSC_NS.XSPICKDIVCR 0701 906Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	XSPICKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	XSPICKDIV[3:0]	XSPI Clock (XSPICLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The XSPICKDIVCR register is used to control the XSPI clock (XSPICLK).

XSPICKDIV[3:0] Bits (XSPI Clock (XSPICLK) Frequency Division Ratio Select)

These bits select the frequency of XSPICLK. Change the value of these bits while the XSPICKCR.XSPICKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB16 = 1 and MSTPCRB.MSTPB17 = 1 before setting XSPICKCR.XSPICKSREQ = 1.

9.2.41 CANFD Core Clock Frequency Division Control Register (CANFDCKDIVCR)

Address(es): SYSC.CANFDCKDIVCR 8701 906Eh,
SYSC_NS.CANFDCKDIVCR 0701 906Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CANFDCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CANFDCKDIV[3:0]	CANFD Core Clock (CANFDCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The CANFDCKDIVCR register is used to control the CANFD core clock (CANFDCLK).

CANFDCKDIV[3:0] Bits (CANFD Core Clock (CANFDCLK) Frequency Division Ratio Select)

These bits select the frequency of CANFDCLK. Change the value of these bits while the CANFDCKCR.CANFDCKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC25 = 1 and MSTPCRC.MSTPC27 = 1 before setting CANFDCKCR.CANFDCKSREQ = 1.

9.2.42 USBHS-PHY Clock Frequency Division Control Register (USB60CKDIVCR)

Address(es): SYSC.USB60CKDIVCR 8701 906Fh,
SYSC_NS.USB60CKDIVCR 0701 906Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	USB60CKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	USB60CKDIV[3:0]	USBHS-PHY Clock (USB60CLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The USB60CKDIVCR register is used to control the USBHS-PHY clock (USB60CLK).

USB60CKDIV[3:0] Bits (USBHS-PHY Clock (USB60CLK) Frequency Division Ratio Select)

These bits select the frequency of USB60CLK. Change the value of these bits while the USB60CKCR.USB60CKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB12 = 1 before setting USB60CKCR.USB60CKSREQ = 1.

9.2.43 I3C Clock Frequency Division Control Register (I3CCKDIVCR)

Address(es): SYSC.I3CCKDIVCR 8701 9070h,
SYSC_NS.I3CCKDIVCR 0701 9070h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I3CCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	I3CCKDIV[3:0]	I3C Clock (I3CCLK) Frequency Division Ratio Select*1	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Make sure that the frequency of I3CCLK does not exceed twice the frequency of PCLKA.

The I3CCKDIVCR register is used to control the I3C clock (I3CCLK).

I3CCKDIV[3:0] Bits (I3C Clock (I3CCLK) Frequency Division Ratio Select)

These bits select the frequency of I3CCLK. Change the value of these bits while the I3CCKCR.I3CCKSRDY flag is 1. When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB4 = 1, MSTPCRB.MSTPB7 = 1, and MSTPCRB.MSTPB8 = 1 before setting I3CCKCR.I3CCKSREQ = 1.

9.2.44 USB Clock Control Register (USBCKCR)

Address(es): SYSC.USBCKCR 8701 9074h,
SYSC_NS.USBCKCR 0701 9074h

	b7	b6	b5	b4	b3	b2	b1	b0
	USBCKSRDY	USBCKSREQ	—	—	USBCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	USBCKSEL[3:0]	USB Clock (USBCLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	USBCKSREQ	USB Clock (USBCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	USBCKSRDY	USB Clock (USBCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The USBCKCR register is used to control the USB clock (USBCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching. To change the values of the USBCKDIVCR.USBCKDIV[3:0] and USBCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRB.MSTPB11 and MSTPCRB.MSTPB12 bits only when switching the division ratio setting from 1/n ($n \neq 1$).
2. Wait for two USBCLK only when switching the division ratio setting from 1/n ($n \neq 1$).
3. Write 1 to the USBCKSREQ bit.
4. Poll the USBCKSRDY flag until it is read as 1.
   While the USBCKSRDY flag is 1, USBCLK is not output.
5. Set the USBCKDIVCR.USBCKDIV[3:0] and USBCKSEL[3:0] bits.
6. Write 0 to the USBCKSREQ bit.
7. Poll the USBCKSRDY flag until it is read as 0.
8. When the USBCKSRDY flag becomes 0, clock switching is complete and output of USBCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while USBCKSREQ = 0 and USBCKSRDY = 0.

USBCKSEL[3:0] Bits (USB Clock (USBCLK) Source Select)

These bits select the source of USBCLK. Change the value of these bits while the USBCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

USBCKSREQ Bit (USB Clock (USBCLK) Switching Request)

This bit selects whether to request switching of USBCLK.

USBCKSRDY Flag (USB Clock (USBCLK) Switching Ready State Flag)

This flag indicates whether switching of USBCLK is possible. While this flag is 1, USBCLK is not output.

9.2.45 XSPI Clock Control Register (XSPICKCR)

Address(es): SYSC.XSPICKCR 8701 9075h,
SYSC_NS.XSPICKCR 0701 9075h

	b7	b6	b5	b4	b3	b2	b1	b0
	XSPICKSRDY	XSPICKSREQ	—	—	XSPICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	XSPICKSEL[3:0]	XSPI Clock (XSPICLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	XSPICKSREQ	XSPI Clock (XSPICLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	XSPICKSRDY	XSPI Clock (XSPICLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The XSPICKCR register is used to control the XSPI clock (XSPICLK).

When switching the clock source, stable output of the clock should be ensured before and after switching.

To change the values of the XSPICKDIVCR.XSPICKDIV[3:0] and XSPICKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to MSTPCRB.MSTPB16 and MSTPCRB.MSTPB17 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Check that both the MSTPCRB.MSTPB16 bit and MSTPCRB.MSTPB17 bit are read as 1 and then wait for two XSPICLK only when switching the frequency division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the XSPICKSREQ bit.
4. Poll the XSPICKSRDY flag until it is read as 1.
   While the XSPICKSRDY flag is 1, XSPICLK is not output.
5. Set the XSPICKDIVCR.XSPICKDIV[3:0] and XSPICKSEL[3:0] bits.
6. Write 0 to the XSPICKSREQ bit.
7. Poll the XSPICKSRDY flag until it is read as 0.
8. When the XSPICKSRDY flag becomes 0, clock switching is complete and output of XSPICLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while XSPICKSREQ = 0 and XSPICKSRDY = 0.

XSPICKSEL[3:0] Bits (XSPI Clock (XSPICLK) Source Select)

These bits select the source of XSPICLK. Change the value of these bits while the XSPICKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

XSPICKSREQ Bit (XSPI Clock (XSPICLK) Switching Request)

This bit selects whether to request switching of XSPICLK.

XSPICKSRDY Flag (XSPI Clock (XSPICLK) Switching Ready State Flag)

This flag indicates whether switching of XSPICLK is possible. While this flag is 1, XSPICLK is not output.

9.2.46 CANFD Core Clock Control Register (CANFDCKCR)

Address(es): SYSC.CANFDCKCR 8701 9076h,
SYSC_NS.CANFDCKCR 0701 9076h

	b7	b6	b5	b4	b3	b2	b1	b0
	CANFDCKSRDY	CANFDCKSREQ	—	—	CANFDCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CANFDCKSEL[3:0]	CANFD Core Clock (CANFDCLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CANFDCKSREQ	CANFD Core Clock (CANFDCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	CANFDCKSRDY	CANFD Core Clock (CANFDCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The CANFDCKCR register is used to control the CANFD core clock (CANFDCLK).

When switching the clock source, stable output of the clock should be ensured before and after switching. To change the values of the CANFDCKDIVCR.CANFDCKDIV[3:0] and CANFDCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRC.MSTPC25 and MSTPCRC.MSTPC27 bits only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two CANFDCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the CANFDCKSREQ bit.
4. Poll the CANFDCKSRDY flag until it is read as 1.
   While the CANFDCKSRDY flag is 1, CANFDCLK is not output.
5. Set the CANFDCKDIVCR.CANFDCKDIV[3:0] and CANFDCKSEL[3:0] bits.
6. Write 0 to the CANFDCKSREQ bit.
7. Poll the CANFDCKSRDY flag until it is read as 0.
8. When the CANFDCKSRDY flag becomes 0, clock switching is complete and output of CANFDCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while CANFDCKSREQ = 0 and CANFDCKSRDY = 0.

CANFDCKSEL[3:0] Bits (CANFD Core Clock (CANFDCLK) Source Select)

These bits select the source of CANFDCLK. Change the value of these bits while the CANFDCKSRDY flag is 1. Do not stop the oscillator selected by these bits unless the MOCO is selected.

CANFDCKSREQ Bit (CANFD Core Clock (CANFDCLK) Switching Request)

This bit selects whether to request switching of CANFDCLK.

CANFDCKSRDY Flag (CANFD Core Clock (CANFDCLK) Switching Ready State Flag)

This flag indicates whether switching of CANFDCLK is possible. While this flag is 1, CANFDCLK is not output.

9.2.47 USBHS-PHY Clock Control Register (USB60CKCR)

Address(es): SYSC.USB60CKCR 8701 9077h,
SYSC_NS.USB60CKCR 0701 9077h

	b7	b6	b5	b4	b3	b2	b1	b0
	USB60CKSRDY	USB60CKSREQ	—	—	USB60CKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	USB60CKSEL[3:0]	USBHS-PHY Clock (USB60CLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	USB60CKSREQ	USBHS-PHY Clock (USB60CLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	USB60CKSRDY	USBHS-PHY Clock (USB60CLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The USB60CKCR register is used to control the USBHS-PHY clock (USB60CLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the USB60CKDIVCR.USB60CKDIV[3:0] and USB60CKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRB.MSTPB12 bit only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two USB60CLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the USB60CKSREQ bit.
4. Poll the USB60CKSRDY flag until it is read as 1.
   While the USB60CKSRDY flag is 1, USB60CLK is not output.
5. Set the USB60CKDIVCR.USB60CKDIV[3:0] and USB60CKSEL[3:0] bits.
6. Write 0 to the USB60CKSREQ bit.
7. Poll the USB60CKSRDY flag until it is read as 0.
8. When the USB60CKSRDY flag becomes 0, clock switching is complete and output of USB60CLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while USB60CKSREQ = 0 and USB60CKSRDY = 0.

USB60CKSEL[3:0] Bits (USBHS-PHY Clock (USB60CLK) Source Select)

These bits select the source of USB60CLK. Change the value of these bits while the USB60CKSRDY flag is 1.

Do not stop the oscillator selected by these bits unless the MOCO is selected.

USB60CKSREQ Bit (USBHS-PHY Clock (USB60CLK) Switching Request)

This bit selects whether to request switching of USB60CLK.

USB60CKSRDY Flag (USBHS-PHY Clock (USB60CLK) Switching Ready State Flag)

This flag indicates whether switching of USB60CLK is possible. While this flag is 1, USB60CLK is not output.

9.2.48 I3C Clock Control Register (I3CCKCR)

Address(es): SYSC.I3CCKCR 8701 9078h,
SYSC_NS.I3CCKCR 0701 9078h

	b7	b6	b5	b4	b3	b2	b1	b0
	I3CCK SRDY	I3CCK SREQ	—	—	I3CCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	I3CCKSEL[3:0]	I3C Clock (I3CCLK) Source Select	b3 b0 0 0 0 1: MOCO (initial value) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	I3CCKSREQ	I3C Clock (I3CCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	I3CCKSRDY	I3C Clock (I3CCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The I3CCKCR register is used to control the I3C clock (I3CCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the I3CCKDIVCR.I3CCKDIV[3:0] and I3CCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRB.MSTPB4, MSTPCRB.MSTPB7, and MSTPCRB.MSTPB8 bits only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two I3CCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the I3CCKSREQ bit.
4. Poll the I3CCKSRDY flag until it is read as 1.
   While the I3CCKSRDY flag is 1, I3CCLK is not output.
5. Set the I3CCKDIVCR.I3CCKDIV[3:0] and I3CCKSEL[3:0] bits.
6. Write 0 to the I3CCKSREQ bit.
7. Poll the I3CCKSRDY flag until it is read as 0.
8. When the I3CCKSRDY flag becomes 0, clock switching is complete and output of I3CCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while I3CCKSREQ = 0 and I3CCKSRDY = 0.

I3CCKSEL[3:0] Bits (I3C Clock (I3CCLK) Source Select)

These bits select the source of I3CCLK. Change the value of these bits while the I3CCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

I3CCKSREQ Bit (I3C Clock (I3CCLK) Switching Request)

This bit selects whether to request switching of I3CCLK.

I3CCKSRDY Flag (I3C Clock (I3CCLK) Switching Ready State Flag)

This flag indicates whether switching of I3CCLK is possible. While this flag is 1, I3CCLK is not output.

9.2.49 SCI Clock Frequency Division Control Register (SCICKDIVCR)

Address(es): SYSC.SCICKDIVCR 8701 9054h,
SYSC_NS.SCICKDIVCR 0701 9054h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SCICKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SCICKDIV[3:0]	SCI Clock (SCICLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SCICKDIVCR register is used to control the SCI clock (SCICLK).

SCICKDIV[3:0] Bits (SCI Clock (SCICLK) Frequency Division Ratio Select)

These bits select the frequency of SCICLK. Change the value of these bits while the SCICKCR.CKSRDY flag is 1. When switching the division ratio setting from 1/n (n ≠ 1), set the MSTPCRB.MSTPB0 bit and all of the MSTPCRB.MSTPB20 bit to the MSTPCRB.MSTPB31 bit to 1 before setting SCICKCR.SCICKSREQ = 1.

9.2.50 SPI Clock Frequency Division Control Register (SPICKDIVCR)

Address(es): SYSC.SPICKDIVCR 8701 9056h,
SYSC_NS.SPICKDIVCR 0701 9056h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SPICKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SPICKDIV[3:0]	SPI Clock (SPICLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3 0 1 1 0: Division by 5 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SPICKDIVCR register is used to control the SPI clock (SPICLK).

SPICKDIV[3:0] Bits (SPI Clock (SPICLK) Frequency Division Ratio Select)

These bits select the frequency of SPICLK. Change the value of these bits while the SPICKCR.CKSRDY flag is 1. When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB18 = 1, MSTPCRB.MSTPB19 = 1, and MSTPCRC.MSTPC23 = 1 before setting SPICKCR.SPICKSREQ = 1.

9.2.51 ADC Clock Frequency Division Control Register (ADCCKDIVCR)

Address(es): SYSC.ADCCKDIVCR 8701 905Ah,
SYSC_NS.ADCCKDIVCR 0701 905Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADCCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADCCKDIV[3:0]	ADC Clock (ADCCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ADCCKDIVCR register is used to control the ADC clock (ADCCLK).

ADCCKDIV[3:0] Bits (ADC Clock (ADCCLK) Frequency Division Ratio Select)

These bits select the frequency of ADCCLK. Change the value of these bits while the ADCCKCR.ADCCKSRDY flag is 1.

When switching the division ratio setting from 1/n ($n \neq 1$), set MSTPCRD.MSTPD21 = 1 before setting ADCCKCR.ADCCKSREQ = 1.

9.2.52 GPT Clock Frequency Division Control Register (GPTCKDIVCR)

Address(es): SYSC.GPTCKDIVCR 8701 905Ch,
SYSC_NS.GPTCKDIVCR 0701 905Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	GPTCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	GPTCKDIV[3:0]	GPT Clock (GPTCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 0 1: Division by 3*1 0 1 1 0: Division by 5*1 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. Do not set when GPTCLK is used for ADC16H or HRPWM.

The GPTCKDIVCR register is used to control the GPT clock (GPTCLK).

GPTCKDIV[3:0] Bits (GPT Clock (GPTCLK) Frequency Division Ratio Select)

These bits select the frequency of GPTCLK. Change the value of these bits while the GPTCKCR.GPTCKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRE.MSTPE31 = 1 before setting GPTCKCR.GPTCKSREQ = 1.

9.2.53 SCI Clock Control Register (SCICKCR)

Address(es): SYSC.SCICKCR 8701 9055h,
SYSC_NS.SCICKCR 0701 9055h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCICKSRDY	SCICKSREQ	—	—	SCICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SCICKSEL[3:0]	SCI Clock (SCICLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SCICKSREQ	SCI Clock (SCICLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	SCICKSRDY	SCI Clock (SCICLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SCICKCR register is used to control the SCI clock (SCICLK).

When switching the clock source, stable output of the clock should be ensured before and after switching. To change the values of the SCICKDIVCR.SCICKDIV[3:0] and SCICKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRB.MSTPB0 bit and all of the MSTPCRB.MSTPB20 bit to the MSTPCRB.MSTPB31 bit only when switching the division ratio setting from 1/n ($n \neq 1$).
2. Wait for two SCICLK only when switching the division ratio setting from 1/n ($n \neq 1$).
3. Write 1 to the SCICKSREQ bit.
4. Poll the SCICKSRDY flag until it is read as 1.
   While the SCICKSRDY flag is 1, SCICLK is not output.
5. Set the SCICKDIVCR.SCICKDIV[3:0] and SCICKSEL[3:0] bits.
6. Write 0 to the SCICKSREQ bit.
7. Poll the SCICKSRDY flag until it is read as 0.
8. When the SCICKSRDY flag becomes 0, clock switching is complete and output of SCICLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while SCICKSREQ = 0 and SCICKSRDY = 0.

SCICKSEL[3:0] Bits (SCI Clock (SCICLK) Source Select)

These bits select the source of SCICLK. Change the value of these bits while the SCICKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

SCICKSREQ Bit (SCI Clock (SCICLK) Switching Request)

This bit selects whether to request switching of SCICLK.

SCICKSRDY Flag (SCI Clock (SCICLK) Switching Ready State Flag)

This flag indicates whether switching of SCICLK is possible. While this flag is 1, SCICLK is not output.

9.2.54 SPI Clock Control Register (SPICKCR)

Address(es): SYSC.SPICKCR 8701 9057h,
SYSC_NS.SPICKCR 0701 9057h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPICK SRDY	SPICK SREQ	—	—	SPICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SPICKSEL[3:0]	SPI Clock (SPICLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SPICKSREQ	SPI Clock (SPICLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	SPICKSRDY	SPI Clock (SPICLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The SPICKCR register is used to control the SPI clock (SPICLK).

When switching the clock source, stable output of the clock should be ensured before and after switching. To change the values of the SPICKDIVCR.SPICKDIV[3:0] and SPICKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRB.MSTPB18, MSTPCRB.MSTPB19, and MSTPCRC.MSTPC23 bits only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two SPICLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the SPICKSREQ bit.
4. Poll the SPICKSRDY flag until it is read as 1.
   While the SPICKSRDY flag is 1, SPICLK is not output.
5. Set the SPICKDIVCR.SPICKDIV[3:0] and SPICKSEL[3:0] bits.
6. Write 0 to the SPICKSREQ bit.
7. Poll the SPICKSRDY flag until it is read as 0.
8. When the SPICKSRDY flag becomes 0, clock switching is complete and output of SPICLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while SPICKSREQ = 0 and SPICKSRDY = 0.

SPICKSEL[3:0] Bits (SPI Clock (SPICLK) Source Select)

These bits select the source of SPICLK. Change the value of these bits while the SPICKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

SPICKSREQ Bit (SPI Clock (SPICLK) Switching Request)

This bit selects whether to request switching of SPICLK.

SPICKSRDY Flag (SPI Clock (SPICLK) Switching Ready State Flag)

This flag indicates whether switching of SPICLK is possible.
While the SPICKSRDY flag is 1, SPICLK is not output.

9.2.55 ADC Clock Control Register (ADCCKCR)

Address(es): SYSC.ADCCKCR 8701 905Bh,
SYSC_NS.ADCCKCR 0701 905Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	ADCCKSRDY	ADCCKSREQ	—	—	ADCCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADCCKSEL[3:0]	ADC Clock (ADCCLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ADCCKSREQ	ADC Clock (ADCCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	ADCCKSRDY	ADC Clock (ADCCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ADCCKCR register is used to control the ADC clock (ADCCLK).

When switching the clock source, stable output of the clock should be ensured before and after switching.

To change the values of the ADCCKDIVCR.ADCCKDIV[3:0] and ADCCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRD.MSTPD21 bit only when switching the division ratio setting from 1/n ($n \neq 1$).
2. Wait for two ADCCLK only when switching the division ratio setting from 1/n ($n \neq 1$).
3. Write 1 to the ADCCKSREQ bit.
4. Poll the ADCCKSRDY flag until it is read as 1.
   While the ADCCKSRDY flag is 1, ADCCLK is not output.
5. Set the ADCCKDIVCR.ADCCKDIV[3:0] and ADCCKSEL[3:0] bits.
6. Write 0 to the ADCCKSREQ bit.
7. Poll the ADCCKSRDY flag until it is read as 0.
8. When the ADCCKSRDY flag becomes 0, clock switching is complete and output of ADCCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while ADCCKSREQ = 0 and ADCCKSRDY = 0.

ADCCKSEL[3:0] Bits (ADC Clock (ADCCLK) Source Select)

These bits select the source of ADCCLK.
Change the value of these bits while the ADCCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

ADCCKSREQ Bit (ADC Clock (ADCCLK) Switching Request)

This bit selects whether to request switching of ADCCLK.

ADCCKSRDY Flag (ADC Clock (ADCCLK) Switching Ready State Flag)

This flag indicates whether switching of ADCCLK is possible.
While the ADCCKSRDY flag is 1, ADCCLK is not output.

9.2.56 GPT Clock Control Register (GPTCKCR)

Address(es): SYSC.GPTCKCR 8701 905Dh,
SYSC_NS.GPTCKCR 0701 905Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	GPTCKSRDY	GPTCKSREQ	—	—	GPTCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	GPTCKSEL[3:0]	GPT Clock (GPTCLK) Source Select	b3 b0 0 0 0 0: HOCO 0 0 0 1: MOCO (initial value) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	GPTCKSREQ	GPT Clock (GPTCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	GPTCKSRDY	GPT Clock (GPTCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The GPTCKCR register is used to control the GPT clock (GPTCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the GPTCKDIVCR.GPTCKDIV[3:0] and GPTCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRE.MSTPE31 bit only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two GPTCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the GPTCKSREQ bit.
4. Poll the GPTCKSRDY flag until it is read as 1.
   While the GPTCKSRDY flag is 1, GPTCLK is not output.
5. Set the GPTCKDIVCR.GPTCKDIV[3:0] and GPTCKSEL[3:0] bits.
6. Write 0 to the GPTCKSREQ bit.
7. Poll the GPTCKSRDY flag until it is read as 0.
8. When the GPTCKSRDY flag becomes 0, clock switching is complete and output of GPTCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while GPTCKSREQ = 0 and GPTCKSRDY = 0.

GPTCKSEL[3:0] Bits (GPT Clock (GPTCLK) Source Select)

These bits select the source of GPTCLK. Change the value of these bits while the GPTCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

GPTCKSREQ Bit (GPT Clock (GPTCLK) Switching Request)

This bit selects whether to request switching of GPTCLK.

GPTCKSRDY Flag (GPT Clock (GPTCLK) Switching Ready State Flag)

This flag indicates whether switching of GPTCLK is possible. While this flag is 1, GPTCLK is not output.

9.2.57 Asynchronous External Bus Clock Frequency Division Control Register (BCKADIVCR)

Address(es): SYSC.BCKADIVCR 8701 90D4h,
SYSC_NS.BCKADIVCR 0701 90D4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BCKACKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BCKACKDIV[3:0]	Asynchronous External Bus Clock (BCLKA) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The BCKADIVCR register is used to control the asynchronous external bus clock (BCLKA).

BCKACKDIV[3:0] Bits (Asynchronous External Bus Clock (BCLKA) Frequency Division Ratio Select)

These bits select the frequency of BCLKA. Change the value of these bits while the BCKACR.BCKACKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set EBCKOCR.EBCKOEN = 0 and SDCKOCR.SDCKOEN = 0 before setting BCKACR.BCKACKSREQ = 1. In addition, set BCKCR.EBCKASEL = 0 to select the synchronous (BCLK) side before switching the division ratio.

9.2.58 ESWM Clock Frequency Division Control Register (ESWCKDIVCR)

Address(es): SYSC.ESWCKDIVCR 8701 90D5h,
SYSC_NS.ESWCKDIVCR 0701 90D5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ESWCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ESWCKDIV[3:0]	ESWM Clock (ESWCLK) Frequency Division Ratio Select *1	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.
Note 1. When using ESWM, set the ESWCLK frequency to satisfy the following relationship:
PCLKA < ESWCLK ≤ ICLK × 1.5

The ESWCKDIVCR register is used to control the ESWM clock (ESWCLK).

ESWCKDIV[3:0] Bits (ESWM Clock (ESWCLK) Frequency Division Ratio Select)

These bits select the frequency of ESWCLK. Change the value of these bits while the ESWCKCR.ESWCKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC30 = 1 before setting ESWCKCR.ESWCKSREQ = 1.

9.2.59 ESWM-PHY Clock Frequency Division Control Register (ESWPCKDIVCR)

Address(es): SYSC.ESWPCKDIVCR 8701 90D6h,
SYSC_NS.ESWPCKDIVCR 0701 90D6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ESWPCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ESWPCKDIV[3:0]	ESWM-PHY Clock (ESWPHYCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ESWPCKDIVCR register is used to control the ESWM-PHY clock (ESWPHYCLK).

ESWPCKDIV[3:0] Bits (ESWM-PHY Clock (ESWPHYCLK) Frequency Division Ratio Select)

These bits select the frequency of ESWPHYCLK. Change the value of these bits while the ESWPCKCR.ESWPCKSRDY flag is 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC30 = 1 before setting ESWPCKCR.ESWPCKSREQ = 1.

9.2.60 Ethernet-PHY Clock Frequency Division Control Register (ETHPCKDIVCR)

Address(es): SYSC.ETHPCKDIVCR 8701 90D8h,
SYSC_NS.ETHPCKDIVCR 0701 90D8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ETHPCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ETHPCKDIV[3:0]	Ethernet-PHY Clock (ETHPHYCLK) Frequency Division Ratio Select	b3 b0 0 0 0 0: Division by 1 (initial value) 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 0 1 0 0: Division by 8 0 1 1 1: Division by 10 1 0 0 0: Division by 16 1 0 0 1: Division by 32 Settings other than those listed above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ETHPCKDIVCR register is used to control the Ethernet-PHY clock (ETHPHYCLK).

ETHPCKDIV[3:0] Bits (Ethernet-PHY Clock (ETHPHYCLK) Frequency Division Ratio Select)

These bits select the frequency of ETHPHYCLK. Change the value of these bits while the ETHPCKCR.ETHPCKSRDY flag is 1.

When switching the division ratio setting from 1/n ($n \neq 1$), set MSTPCRC.MSTPC28 = 1 before setting ETHPCKCR.ETHPCKSREQ = 1.

9.2.61 Asynchronous External Bus Clock Control Register (BCKACR)

Address(es): SYSC.BCKACR 8701 90DAh,
SYSC_NS.BCKACR 0701 90DAh

	b7	b6	b5	b4	b3	b2	b1	b0
	BCKACKSRDY	BCKACKSREQ	—	—	BCKACKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BCKACKSEL[3:0]	Asynchronous External Bus Clock (BCLKA) Source Select	b3 b0 0 0 0 1: MOCO (initial value) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	BCKACKSREQ	Asynchronous External Bus Clock (BCLKA) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	BCKACKSRDY	Asynchronous External Bus Clock (BCLKA) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The BCKACR register is used to control the asynchronous external bus clock (BCLKA).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the BCKADIVCR.BCKACKDIV[3:0] and BCKACKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 0 to the EBCKOCR.EBCKOEN and SDCKOCR.SDCKOEN bits only when switching the division ratio setting from 1/n (n ≠ 1).
2. Write 1 to the BCKACKSREQ bit.
3. Poll the BCKACKSRDY flag until it is read as 1.
   While the BCKACKSRDY flag is 1, BCLKA is not output.
4. Write 0 to the BCKCR.EBCKASEL bit only when switching the division ratio setting from 1/n (n ≠ 1).
5. Set the BCKCR.BCLKDIV, BCKCR.EBCKASEL, BCKADIVCR.BCKACKDIV[3:0], and BCKACKSEL[3:0] bits.
6. Write 0 to the BCKACKSREQ bit.
7. Poll the BCKACKSRDY flag until it is read as 0.
8. When the BCKACKSRDY flag becomes 0, clock switching is complete and output of BCLKA begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while BCKACKSREQ = 0 and BCKACKSRDY = 0.

BCKACKSEL[3:0] Bits (Asynchronous External Bus Clock (BCLKA) Source Select)

These bits select the source of BCLKA. Change the value of these bits while the BCKACKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

BCKACKSREQ Bit (Asynchronous External Bus Clock (BCLKA) Switching Request)

This bit selects whether to request switching of BCLKA.

BCKACKSRDY Flag (Asynchronous External Bus Clock (BCLKA) Switching Ready State Flag)

This flag indicates whether switching of BCLKA is possible. While this flag is 1, BCLKA is not output.

9.2.62 ESWM Clock Control Register (ESWCKCR)

Address(es): SYSC.ESWCKCR 8701 90DBh,
SYSC_NS.ESWCKCR 0701 90DBh

	b7	b6	b5	b4	b3	b2	b1	b0
	ESWCKSRDY	ESWCKSREQ	—	—	ESWCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ESWCKSEL[3:0]	ESWM Clock (ESWCLK) Source Select	b3 b0 0 0 0 1: MOCO (initial value) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ESWCKSREQ	ESWM Clock (ESWCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	ESWCKSRDY	ESWM Clock (ESWCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ESWCKCR register is used to control the ESWM clock (ESWCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the ESWCKDIVCR.ESWCKDIV[3:0] and ESWCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRC.MSTPC30 bit only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ESWCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the ESWCKSREQ bit.
4. Poll the ESWCKSRDY flag until it is read as 1.
   While the ESWCKSRDY flag is 1, ESWCLK is not output.
5. Set the ESWCKDIVCR.ESWCKDIV[3:0] and ESWCKSEL[3:0] bits.
6. Write 0 to the ESWCKSREQ bit.
7. Poll the ESWCKSRDY flag until it is read as 0.
8. When the ESWCKSRDY flag becomes 0, clock switching is complete and output of ESWCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while ESWCKSREQ = 0 and ESWCKSRDY = 0.

ESWCKSEL[3:0] Bits (ESWM Clock (ESWCLK) Source Select)

These bits select the source of ESWCLK. Change the value of these bits while the ESWCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

ESWCKSREQ Bit (ESWM Clock (ESWCLK) Switching Request)

This bit selects whether to request switching of ESWCLK.

ESWCKSRDY Flag (ESWM Clock (ESWCLK) Switching Ready State Flag)

This flag indicates whether switching of ESWCLK is possible. While this flag is 1, ESWCLK is not output.

9.2.63 ESWM-PHY Clock Control Register (ESWPCKCR)

Address(es): SYSC.ESWPCKCR 8701 90DCh,
SYSC_NS.ESWPCKCR 0701 90DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	ESWPCKSRDY	ESWPCKSREQ	—	—	ESWPCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ESWPCKSEL[3:0]	ESWM-PHY Clock (ESWPHYCLK) Source Select	b3 b0 0 0 0 1: MOCO (initial value) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ESWPCKSREQ	ESWM-PHY Clock (ESWPHYCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	ESWPCKSRDY	ESWM-PHY Clock (ESWPHYCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ESWPCKCR register is used to control the ESWM-PHY clock (ESWPHYCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the ESWPCKDIVCR.ESWPCKDIV[3:0] and ESWPCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRC.MSTPC30 bit only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ESWPHYCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the ESWPCKSREQ bit.
4. Poll the ESWPCKSRDY flag until it is read as 1.
   While the ESWPCKSRDY flag is 1, ESWPHYCLK is not output.
5. Set the ESWPCKDIVCR.ESWPCKDIV[3:0] and ESWPCKSEL[3:0] bits.
6. Write 0 to the ESWPCKSREQ bit.
7. Poll the ESWPCKSRDY flag until it is read as 0.
8. When the ESWPCKSRDY flag becomes 0, clock switching is complete and output of ESWPHYCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while ESWPCKSREQ = 0 and ESWPCKSRDY = 0.

ESWPCKSEL[3:0] Bits (ESWM-PHY Clock (ESWPHYCLK) Source Select)

These bits select the source of ESWPHYCLK. Change the value of these bits while the ESWPCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

ESWPCKSREQ Bit (ESWM-PHY Clock (ESWPHYCLK) Switching Request)

This bit selects whether to request switching of ESWPHYCLK.

ESWPCKSRDY Flag (ESWM-PHY Clock (ESWPHYCLK) Switching Ready State Flag)

This flag indicates whether switching of ESWPHYCLK is possible. While this flag is 1, ESWPHYCLK is not output.

9.2.64 Ethernet-PHY Clock Control Register (ETHPCKCR)

Address(es): SYSC.ETHPCKCR 8701 90DEh,
SYSC_NS.ETHPCKCR 0701 90DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ETHPCKSRDY	ETHPCKSREQ	—	—	ETHPCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ETHPCKSEL[3:0]	Ethernet-PHY Clock (ETHPHYCLK) Source Select	b3 b0 0 0 0 1: MOCO (initial value) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Settings other than those listed above are prohibited.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ETHPCKSREQ	Ethernet-PHY Clock (ETHPHYCLK) Switching Request	0: No request. 1: Switching is requested.	R/W
b7	ETHPCKSRDY	Ethernet-PHY Clock (ETHPHYCLK) Switching Ready State Flag	0: Switching is not possible. 1: Switching is possible.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before changing the value of this register.

The ETHPCKCR register is used to control the Ethernet-PHY clock (ETHPHYCLK).
When switching the clock source, stable output of the clock should be ensured before and after switching.
To change the values of the ETHPCKDIVCR.ETHPCKDIV[3:0] and ETHPCKSEL[3:0] bits, use the following procedure.

Procedure for switching the clock

1. Write 1 to the MSTPCRC.MSTPC28 bit only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ETHPHYCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to the ETHPCKSREQ bit.
4. Poll the ETHPCKSRDY flag until it is read as 1.
   While the ETHPCKSRDY flag is 1, ETHPHYCLK is not output.
5. Set the ETHPCKDIVCR.ETHPCKDIV[3:0] and ETHPCKSEL[3:0] bits.
6. Write 0 to the ETHPCKSREQ bit.
7. Poll the ETHPCKSRDY flag until it is read as 0.
8. When the ETHPCKSRDY flag becomes 0, clock switching is complete and output of ETHPHYCLK begins.

When the MCU is to be placed in software standby or deep software standby mode, do not execute a WAIT instruction during clock switching. Execute a WAIT instruction while ETHPCKSREQ = 0 and ETHPCKSRDY = 0.

ETHPCKSEL[3:0] Bits (Ethernet-PHY Clock (ETHPHYCLK) Source Select)

These bits select the source of ETHPHYCLK. Change the value of these bits while the ETHPCKSRDY flag is 1.
Do not stop the oscillator selected by these bits unless the MOCO is selected.

ETHPCKSREQ Bit (Ethernet-PHY Clock (ETHPHYCLK) Switching Request)

This bit selects whether to request switching of ETHPHYCLK.

ETHPCKSRDY Flag (Ethernet-PHY Clock (ETHPHYCLK) Switching Ready State Flag)

This flag indicates whether switching of ETHPHYCLK is possible. While this flag is 1, ETHPHYCLK is not output.

9.3 Main Clock Oscillator

To supply a clock signal to the main clock oscillator, use either of the following methods:

- Connect a resonator.
- Connect the input of an external clock signal.

9.3.1 Connecting a Resonator

Figure 9.2 shows an example of connecting a resonator. A damping resistor (Rd) should be inserted, if required.
Since the resistor value varies depending on the resonator and oscillation driving capacity, use a value recommended by the resonator manufacturer. If the addition of an external feedback resistor (Rf) is directed by the resonator manufacturer, follow that direction and insert an Rf resistor between the EXTAL and XTAL pins.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator, as shown in Table 9.1.

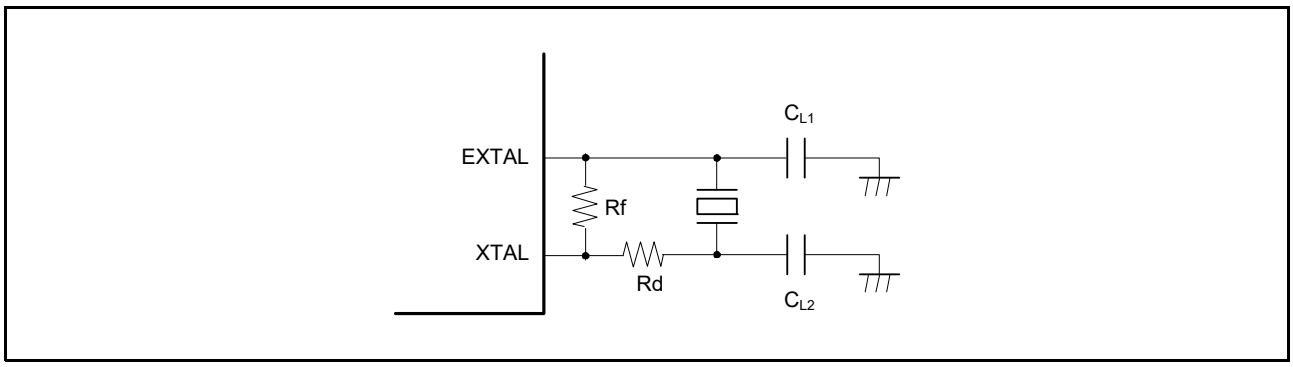


Figure 9.2 Example of Connecting a Resonator

Figure 9.3 shows an equivalent circuit of the resonator.

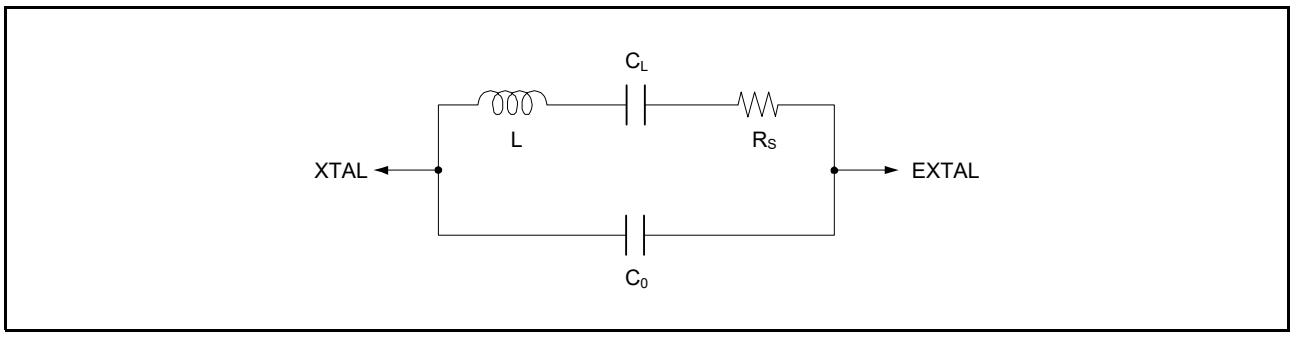


Figure 9.3 Equivalent Circuit of the Resonator

9.3.2 Inputting an External Clock

Figure 9.4 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin is put in the high-impedance state.

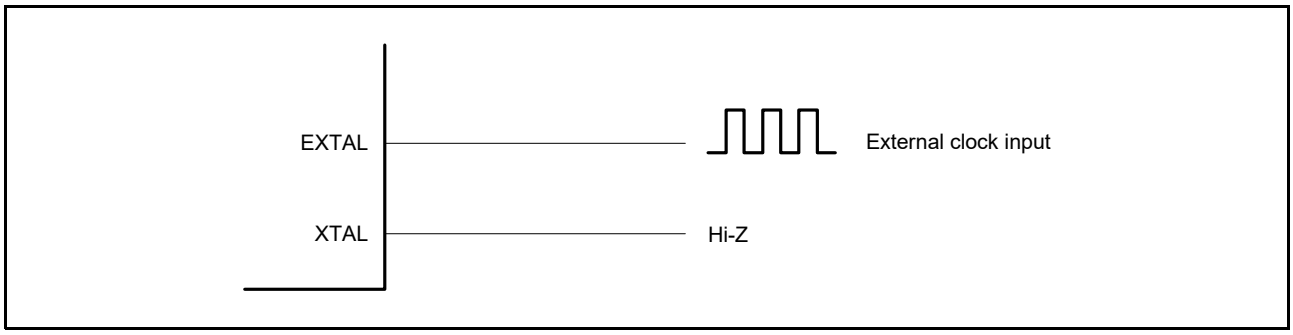


Figure 9.4 Example of Connecting an External Clock

9.3.3 Note on the External Clock Input

The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0.

9.3.4 Notes on Oscillation Keeping in Software Standby Mode

By using the oscillation keep function of the main clock oscillator in software standby mode, the oscillation stabilization wait time of the main clock oscillator after release from software standby mode can be shortened.
If the MCU is placed in software standby mode when the MOSCSCR.MOSCSOKP bit is 1 (enabled) and the main clock oscillator is operating, the main clock oscillator will keep oscillating in software standby mode. When the MCU is released from software standby mode in this state, the main clock can be used without requiring any oscillation stabilization wait time.
When using the oscillation keep function, confirm that the setting value is written to the register before the MCU is placed in software standby mode.
However, even if the oscillation keep function is enabled, the clock supply to the system and peripheral modules is stopped in software standby mode.

9.4 Sub-Clock Oscillator

To supply a clock signal to the sub-clock oscillator, use either of the following methods:

- Connect a resonator.
- Connect the input of an external clock signal.

9.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.5. A damping resistor (Rd) should be inserted, if required. Since the resistor value varies depending on the resonator and oscillation driving capacity, use a value recommended by the resonator manufacturer. If the addition of an external feedback resistor (Rf) is directed by the resonator manufacturer, follow that direction and insert an Rf resistor between the XCIN and XCOUT pins. When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator, as shown in Table 9.1.

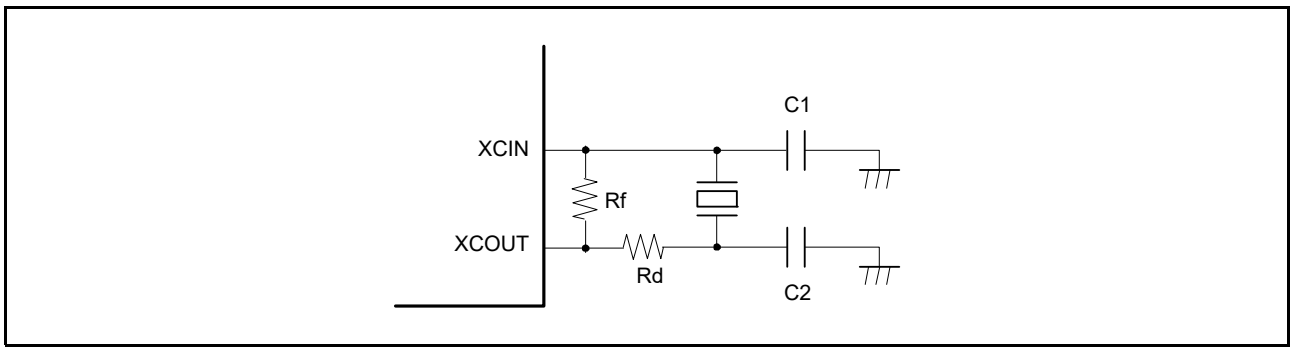


Figure 9.5 Example of Connecting a 32.768-kHz Crystal Resonator

Figure 9.6 shows an equivalent circuit of the 32.768-kHz crystal resonator.

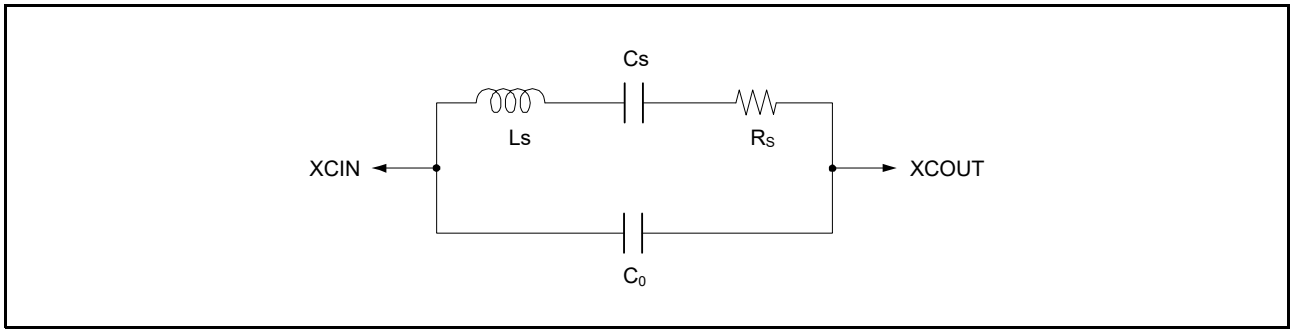


Figure 9.6 Equivalent Circuit of the 32.768-kHz Crystal Resonator

9.4.2 Handling of Pins when the Sub-Clock Oscillator is Not in Use

For handling of pins when the sub-clock is not in use, see section 22.5, Handling of Unused Pins.

9.4.3 Input of an External Clock

In the external clock input mode, apply the external clock to the XCIN pin. See section 22.6.10, Note on Using the Sub-Clock Oscillator.

9.5 Low-Speed On-Chip Oscillator (LOCO)

The low-speed on-chip oscillator (LOCO) is an internal oscillator that supplies a low-speed clock.

9.5.1 LOCO Operating Conditions

There are restrictions on LOCO operation settings. For details, see section 9.5.3, Restrictions on LOCO Operation and Stopping.
The LOCO oscillates when any of the following conditions is met.

[LOCO operating conditions]

- The LOCO is set to start operation in the LOCOCR register (LOCOCR.LCSTP bit is 0).
- The IWDT is set to auto start mode. (OFS0.IWDTSTRT bit is 0).*1
- The IWDT is set to register start mode (OFS0.IWDTSTRT bit is 1) and counting is started.*2
- While waiting for the oscillation stabilization of the main clock oscillator (while the MOSCCR.MOSTP bit is 0 and the OSCSF.MOSCSF bit is 0)*3
- The SOSC oscillation stop detection function is enabled (SOSTDCR.SOSTDE bit is 1).

Note 1. When the IWDT is set to auto start mode, regardless of the setting of the OFS0.IWDTSLCSTP bit, the LOCO continues to oscillate. After that, even if the MCU is placed in CPU sleep mode or CPU deep sleep mode, the LOCO continues to oscillate.
Note 2. When the IWDT is set to register start mode, regardless of the setting of the IWDTCSTPR.SLCSTP bit, the LOCO continues to oscillate once the IWDT starts counting. After that, even if the MCU is placed in CPU sleep mode or CPU deep sleep mode, the LOCO continues to oscillate.
Note 3. Even if the main clock oscillator is set to external clock input mode, the LOCO operates because the oscillation stabilization wait circuit operates when the main clock oscillator starts operating.

9.5.2 LOCO Stop Conditions

There are restrictions on LOCO stop settings. For details, see section 9.5.3, Restrictions on LOCO Operation and Stopping.
The LOCO can be stopped by setting the LCSTP bit of the LOCOCR register to the stop setting (LOCOCR.LCSTP = 1). However, even if the LCSTP bit is set to the stop setting, the LOCO may not stop according to the operating state of the IWDT and oscillator.
The LOCO can be stopped when all of the following conditions are met.

[LOCO stop conditions]

- The LOCO is set to stop operation in the LOCOCR register (LOCOCR.LCSTP bit is 1).
- The IWDT has not started counting or the IWDT was stopped by a transition to software standby mode 1 or 2 or deep software standby mode 1 during counting by the IWDT while the OFS0.IWDTSTPCTL bit is set to 1 or the IWDT.IWDTCSTPR.SLCSTP bit is set to 1.*1
- The main clock oscillator is stopped or main clock oscillation is stable (MOSCCR.MOSTP bit is 1, or MOSCCR.MOSTP bit is 0 and OSCSF.MOSCSF bit is 1).
- When the SOSC oscillation stop detection function is set to disabled (SOSTDCR.SOSTDE bit is 0).

Note 1. The operating state of the IWDT does not affect the LOCO operation in deep software standby mode 2 or 3.

9.5.3 Restrictions on LOCO Operation and Stopping

After setting the LOCOCR.LCSTP bit to 0 to start the LOCO, only use the LOCO clock after the LOCO clock oscillation stabilization wait time (tLOCOWT) has elapsed. A fixed oscillation stabilization wait time is required after setting the LOCO to start operation. A fixed wait time for the oscillation to stop is also required.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO, read the OSCMONR.LOCOMON flag to make sure that the LOCO is stopped, and then set the operation.
- Confirm that LOCO clock oscillation is stable before stopping the LOCO.
- Confirm that LOCO clock oscillation is stable before executing a WAIT instruction to place the MCU in software standby mode or deep software standby mode.
- When placing the MCU in software standby mode or deep software standby mode after setting the LOCO to stop, read the OSCMONR.LOCOMON flag to make sure that the LOCO is stopped, and then execute a WAIT instruction.

The LOCO clock is used to measure the oscillation stabilization wait time of the main clock oscillator. While the main clock oscillator is waiting for oscillation stabilization, the LOCO oscillates regardless of the setting of the LCSTP bit. At this time, it is necessary to prevent conflicts between the LOCO being stopped (or started) by the LCSTP bit and the LOCO oscillating during the oscillation stabilization wait time of the main clock oscillator (or the LOCO clock being stopped after oscillation stabilization of the main clock). Therefore, do not change the LOCOCR.LCSTP bit during the period from three LOCO clock cycles before setting the main clock oscillator to start until confirming stabilization of the main clock oscillator with the oscillation stabilization flag in the OSCSF register.
When using the IWDT in register start mode (OFS0.IWDTSTRT = 1), set the LOCO to operate (LOCOCR.LCSTP = 0) before IWDT counting starts, and use the IWDT after LOCO oscillation has stabilized.
When using the IWDT in register start mode, do not change the LOCOCR.LCSTP bit during the period from three LOCO clock cycles before IWDT counting starts until confirming the start of IWDT counting.
When negation of the voltage-monitoring m reset is selected or digital filtering is used in programmable voltage detection (PVDm) (m = 1, 2, 4, or 5), set the LOCO to start operation (LOCOCR.LCSTP = 0) before enabling PVDm, and use PVDm only after the LOCO oscillation has stabilized. When voltage monitor m reset negation is selected (PVDmCR0.RN = 1) or when the digital filter is used (PVDmCR0.DFDIS = 1), do not write 1 to the LOCOCR.LCSTP bit (LOCO stop) (m = 1, 2, 4, or 5).

9.6 Middle-Speed On-Chip Oscillator (MOCO)

The middle-speed on-chip oscillator (MOCO) is an internal oscillator that supplies a middle-speed clock.

9.6.1 MOCO Operating Conditions

There are restrictions on MOCO operation settings. For details, see section 9.6.3, Restrictions on MOCO Operation and Stopping.
The MOCO oscillates when any of the following conditions is met.

[MOCO operating conditions]

- The MOCO is set to start operation in the MOCOCR register (MOCOCR.MCSTP = 0).
- While waiting for the HOCO oscillation stabilization (while the OFS1.HOCOEN bit is 0 or the HOCOCR.HCSTP bit is 0, and the OSCSF.HOCOSF bit is 0)
- While waiting for the PLL1 oscillation stabilization (while the PLLCR.PLLSTP bit is 0 and the OSCSF.PLLSF bit is 0)
- While waiting for the PLL2 oscillation stabilization (while the PLL2CR.PLL2STP bit is 0 and the OSCSF.PLL2SF bit is 0)
- The MOSC oscillation stop detection function is enabled (OSTDCR.OSTDE bit is 1).
- The SOSC oscillation stop detection function is enabled (SOSTDCR.SOSTDE bit is 1).

Note: The MOCO is stopped in software standby mode or deep software standby mode.

9.6.2 MOCO Stop Conditions

There are restrictions on MOCO stop settings. For details, see section 9.6.3, Restrictions on MOCO Operation and Stopping.
There are two methods to stop the MOCO: stop the MOCO by placing the MCU in a low power consumption mode or stop the MOCO by using the MOCOCR register.

(1) MOCO stop conditions related to low power consumption modes

The MOCO stops when the MCU is placed in either of the following low power consumption modes:

- Software standby mode (MOCOSCR.MOCOSOKP = 0)
- Deep software standby mode

(2) MOCO stop conditions related to the MOCOCR register

When the MOCO is to be stopped using the MOCOCR register, even if the MCSTP bit is set to stop (MOCOCR.MCSTP bit is 1), the MOCO may not stop depending on the operating state of the system.
The MOCO can be stopped when all of the following conditions are met.

[MOCO stop conditions]

- The MOCO is set to stop operation in the MOCOCR register (MOCOCR.MCSTP bit is 1).
- The HOCO is stopped or HOCO oscillation is stable (OFS1.HOCOEN bit is 1 and HOCOCR.HCSTP bit is 1, or OSCSF.HOCOSF bit is 1).
- The PLL1 circuit is stopped or PLL1 oscillation is stable (PLLCR.PLLSTP bit is 1 or OSCSF.PLLSF bit is 1).
- The PLL2 circuit is stopped or PLL2 oscillation is stable (PLL2CR.PLL2STP bit is 1 or OSCSF.PLL2SF bit is 1).

- The MOSC oscillation stop detection function is disabled (the OSTDCR.OSTDE bit is set to 0).
- The SOSC oscillation stop detection function is disabled (the SOSTDCR.SOSTDE bit is set to 0).

9.6.3 Restrictions on MOCO Operation and Stopping

After setting the MOCOCR.MCSTP bit to 0 to start the MOCO, only use the MOCO clock after the MOCO clock oscillation stabilization wait time (tMOCOWT) has elapsed. A fixed oscillation stabilization wait time is required after setting the MOCO to start operation. A fixed wait time for the oscillation to stop is also required.

The following restrictions apply when starting and stopping operation:

- After stopping the MOCO, read the OSCMONR.MOCOMON flag to make sure that the MOCO is stopped, and then set the operation.
- Confirm that MOCO oscillation is stable before stopping the MOCO.
- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing a WAIT instruction to place the MCU in software standby mode or deep software standby mode. If the MOCO is stopped, start operation of the MOCO.
- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing a WAIT instruction to place the MCU in CPU deep sleep mode. If the MOCO is stopped, start operation of the MOCO.
- Confirm that MOCO oscillation is stable before executing power control for each power domain by software. If the MOCO is stopped, start operation of the MOCO.
- Confirm that MOCO oscillation is stable before programming or erasing MRAM. If the MOCO is stopped, start operation of the MOCO.

Writing 1 to the MCSTP bit is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO)
- Main clock oscillation stop detection is enabled (OSTDCR.OSTDE bit is 1).
- While waiting for HOCO oscillation stabilization (while the OFS1.HOCOEN bit is 0 or the HOCOCR.HCSTP bit is 0, and the OSCSF.HOCOSF bit is 0)
- While waiting for PLL1 oscillation stabilization (while the PLLCR.PLLSTP bit is 0 and the OSCSF.PLLSF bit is 0)
- While waiting for PLL2 oscillation stabilization (while the PLL2CR.PLL2STP bit is 0 and the OSCSF.PLL2SF bit is 0)
- While executing power control for each power domain by software (during power gating control)

The MOCO clock is used to measure the oscillation stabilization wait time of the HOCO, PLL1, or PLL2 clock. While waiting for the HOCO, PLL1, or PLL2 clock oscillation to become stable, the MOCO oscillates regardless of the MOCOCR.MCSTP bit setting. At this time, it is necessary to prevent conflicts between the MOCO being stopped (or started) by the MOCOCR.MCSTP bit and the MOCO oscillating during the HOCO, PLL1, or PLL2 clock oscillation stabilization wait time (or the MOCO clock being stopped after HOCO, PLL1, or PLL2 clock oscillation stabilization). Therefore, do not change the MOCOCR.MCSTP bit during the period from three MOCO clock cycles before setting the HOCO, PLL1, or PLL2 to start until confirming stabilization of the HOCO, PLL1, or PLL2 clock oscillation with the HOCOSF, PLLSF, or PLL2SF flag in the OSCSF register.

9.7 High-Speed On-Chip Oscillator (HOCO)

The high-speed on-chip oscillator (HOCO) is an internal oscillator that supplies a high-speed clock.

9.7.1 Frequency Locked Loop Function

The HOCO supports the frequency locked loop (FLL) function that corrects the frequency accuracy of the HOCO clock using the sub-clock oscillator. When the FLL function is enabled, the frequency accuracy of the HOCO clock can be improved compared to when the FLL function is disabled.
When using the FLL function, oscillation of the sub-clock oscillator must be stable. In addition, the sub-clock oscillator must meet the frequency accuracy specified in the electrical characteristics. (For details, see section 67, Electrical Characteristics.)
When enabling the FLL function and using the HOCO, enable the FLL function in advance and enable the HOCO (HOCOCR.HCTSP = 0) after SOSC oscillation has stabilized.
Set the FLLCR2.FLLCNTL[10:0] bits as follows for each HOCO frequency setting. The HOCO clock will reach the desired frequency after the period of tFLLWT has elapsed following the HOCO becoming enabled.

Table 9.5 FLLCNTL Setting for Each HOCO Frequency Setting

FLLCR2.FLLCNTL[10:0]	OFS1.HOCOFRQ0[2:0]	Target Frequency (FHOCO)
1E8h	000b	16 MHz
225h	001b	18 MHz
262h	010b	20 MHz
1E8h	100b	32 MHz
1E8h	111b	48 MHz

9.7.2 Notes on Oscillation Keeping in Software Standby Mode

By using the oscillation keep function of the HOCO in software standby mode, the HOCO oscillation stabilization wait time after release from software standby mode can be shortened.
To keep the HOCO oscillating in software standby mode, make the following settings:

- Enable the HOCO standby oscillation keep function (set the HOCOSCR.HOCOSOKP bit to 1).
- Set power control for the HOCO in the HOCOLDOCR register.

If the MCU is placed in software standby mode while the HOCO is operating with the above settings, the HOCO will keep oscillating in software standby mode. When the MCU is released from software standby mode in this state, the HOCO clock can be used without any oscillator stabilization wait time.
When using the oscillation keep function, confirm that the setting value is written to the register before the MCU is placed in software standby mode.
However, even if the oscillation keep function is enabled, the clock supply to the system and peripheral modules is stopped in software standby mode.

9.8 PLL Circuits

The PLL circuits are capable of multiplying the frequency of the clock input from an oscillator.
Figure 9.7 shows a block diagram of the PLL1 circuit, and Figure 9.8 shows a block diagram of the PLL2 circuit. The PLL1 and PLL2 circuits have the same circuit configuration. The PLL1 circuit is explained below as an example.
The input clock source selection circuit selects the clock source input to the PLL circuit.
The input clock frequency dividing circuit divides the frequency of the input clock and generates the PLL reference clock.
The frequency control circuit controls the VCO clock to be generated from the PLL reference clock according to the setting of the frequency multiplier. The multiplication ratio is determined by the combination of the PLLMUL and PLLMULNF bits. The PLLMUL bits determine the integer part of the multiplication ratio and the PLLMULNF bits determine the fractional part of the multiplication ratio. For example, if the setting of the PLLMUL bits is 26 and the setting of the PLLMULNF bits is 0.33, the multiplication ratio is 26.33.
The PLL circuit has three output clocks. The output clock frequency dividing circuit divides the VCO clock to generate each PLL clock output. The division ratio for generating each output clock can be set independently with the PLODIVP, PLODIVQ, and PLODIVR bits.
The PLL input clock frequency, PLL reference clock frequency, VCO clock frequency, and each output clock frequency must be within the specified ranges in section 67, Electrical Characteristics.
The PLL frequency setting can only be changed when the PLL circuit is stopped. Do not change the PLL frequency setting when the PLL stop control bit (PLLSTP) is 0.

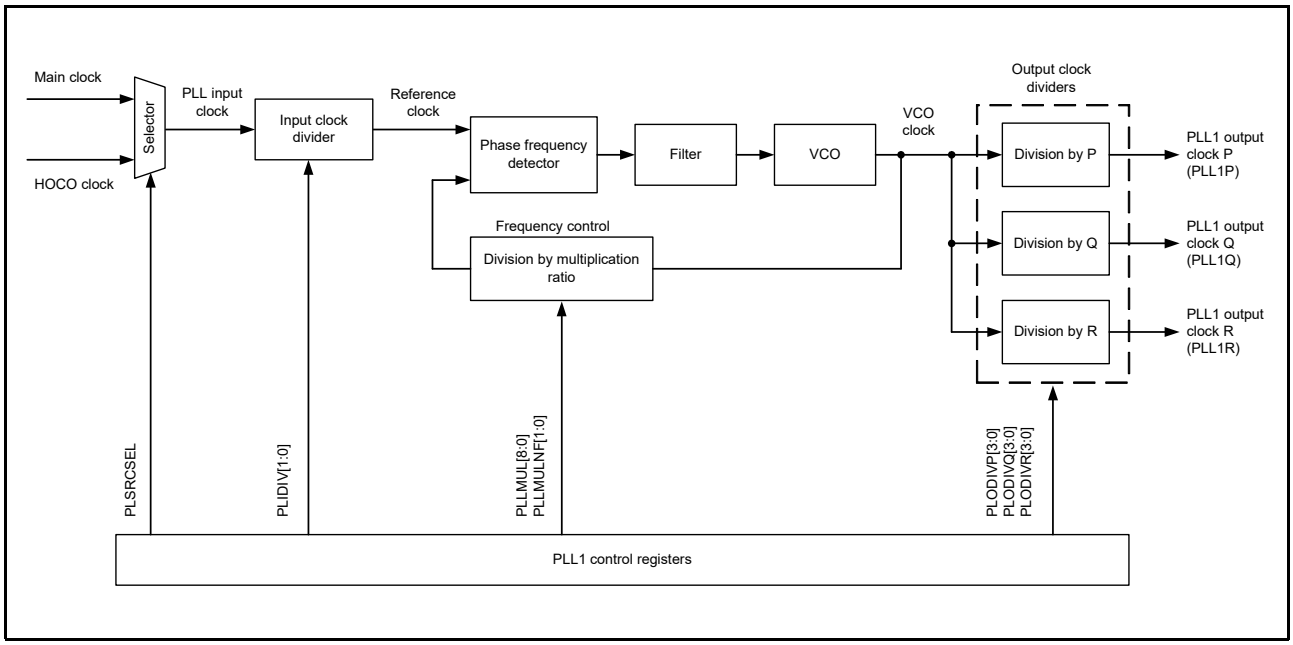


Figure 9.7 Block Diagram of the PLL1 Circuit

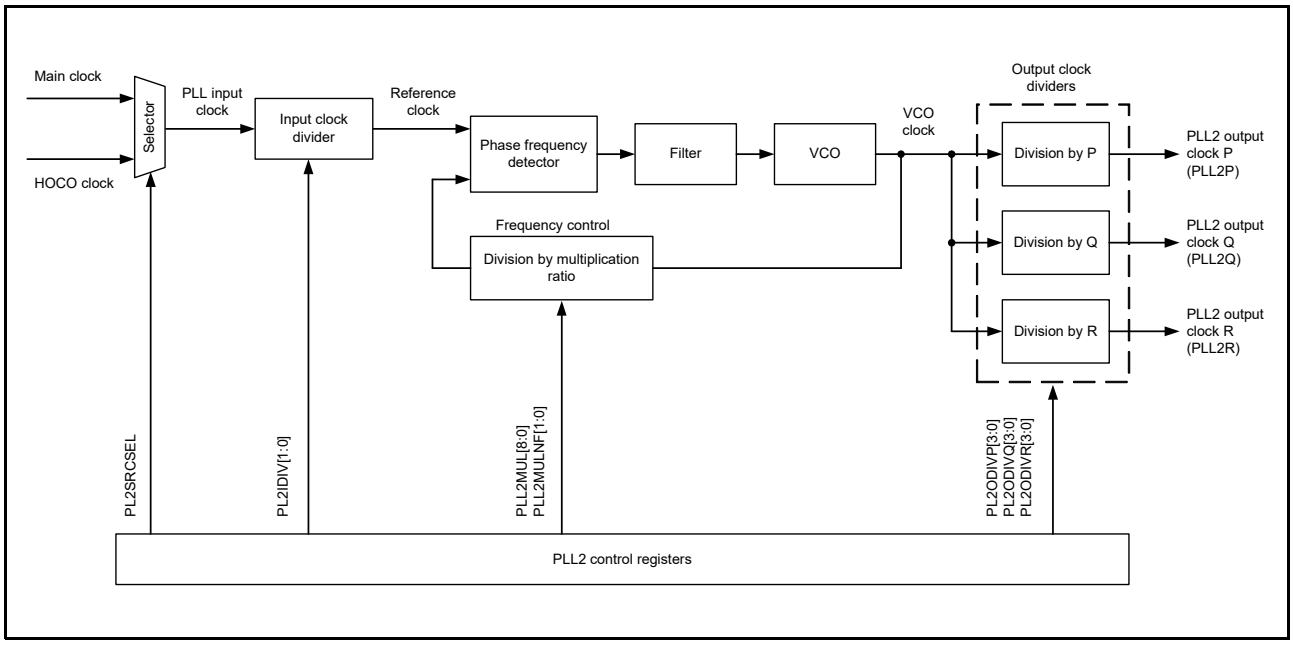


Figure 9.8 Block Diagram of the PLL2 Circuit

9.9 Oscillation Stop Detection Function

9.9.1 Main Clock Oscillation Stop Detection and Operation after Detection

The main clock oscillation stop detection function is used to detect stopping of the main clock oscillator. When an oscillation stop is detected, the system clock source is switched as follows:

- If an oscillation stop is detected when SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source is switched to the MOCO clock.
- If an oscillation stop is detected when PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P), the system clock source remains to be the PLL1 clock. The frequency becomes the free-running value and the setting of the SCKSCR.CKSEL[2:0] bits does not change.

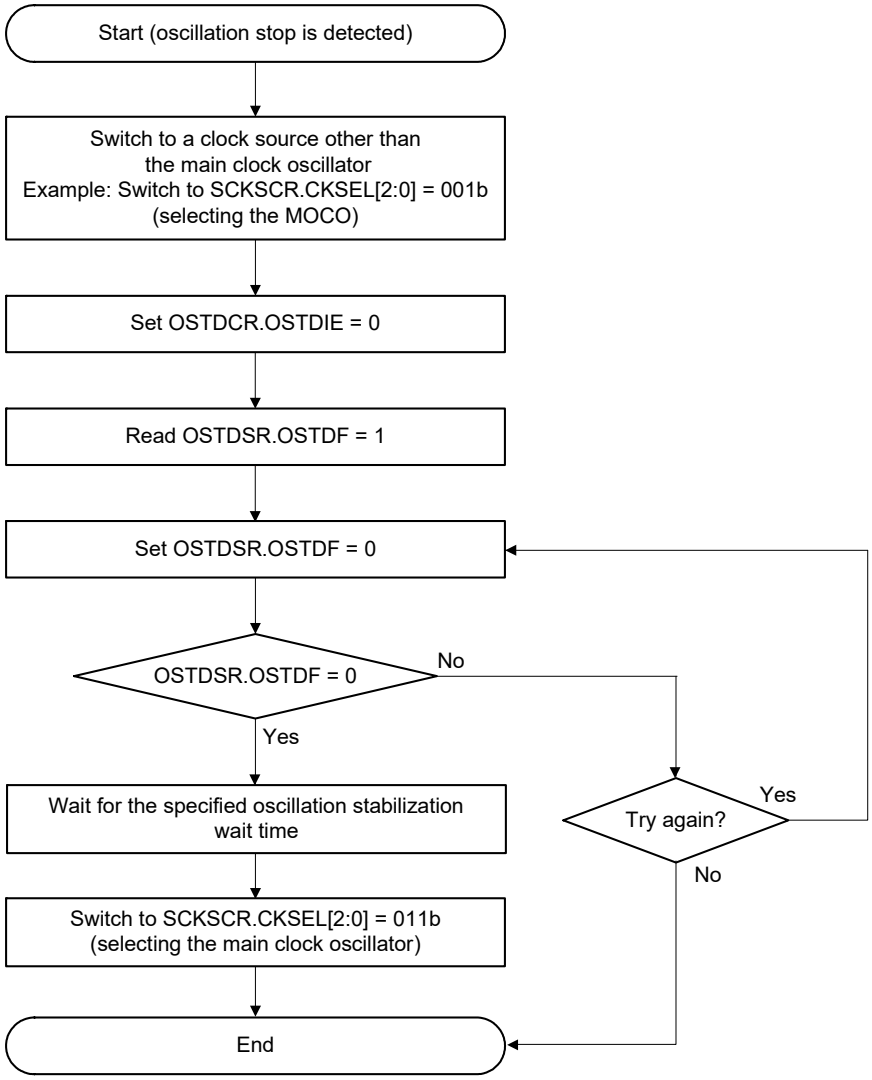
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the general PWM timer (GPTW) output can be forced to the stop state on detection.
Stopping of the main clock oscillator is detected when the input clock of the oscillation stop detection circuit remains low or high for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 67, Electrical Characteristics.
Switching between the main clock and MOCO clock or between the PLL1 clock and PLL1 free-running clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF).
The OSTDF flag controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
  - When the OSTDF flag changes from 0 to 1, the clock source switches to the MOCO clock.
  - When the OSTDF flag changes from 1 to 0, the clock source switches back to MOSC.
- When PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
  - When the OSTDF flag changes from 0 to 1, the clock source switches to the PLL1 free-running oscillation clock.
  - When the OSTDF flag changes from 1 to 0, the clock source switches back to the PLL1 clock.

To switch the clock source to the main clock or PLL1 clock again after detection of an oscillation stop, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL1 clock and clear the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL1 clock after the specified oscillation stabilization wait time has elapsed.
After release from the reset state, the main clock oscillator is stopped and the main clock oscillation stop detection function is disabled. To enable the main clock oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after the specified oscillation stabilization wait time has elapsed.
The main clock oscillation stop detection function detects when the main clock oscillator is stopped by an external cause.
The main clock oscillation stop detection function switches the following clocks to the MOCO clock (when the system clock source is MOSC).

- All clocks that can be selected as the MOSC clock, except for CLKOUT and the clock source of PLL
- The system clock (ICLK) frequency during MOCO operation (when the system clock source is MOSC) is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[3:0]).

Example of returning when setting CKSEL[2:0] = 011b (selecting the main clock oscillator) after an oscillation stop has been detected



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed from the user system to allow oscillation to resume.

Figure 9.9 Flow for Recovery from Detecting Stopping of the Main Clock Oscillator

9.9.2 Main Clock Oscillation Stop Detection Interrupts

A main clock oscillation stop detection interrupt (OSTDI) is generated when the oscillation stop detection flag (OSTDSR.OSTDF) is 1 and the oscillation stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (enabled). At this time, the GPTW port output enable (POEG) is notified that the main clock oscillator has stopped. On accepting the notification, the POEG sets the oscillation stop detection flag in the POEG group x setting register (POEGGx.OSTPF) (x = A to D) to 1.
After the oscillation stop has been detected, wait for at least 10 clock cycles of PCLKB before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after setting the oscillation stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) to 0. Wait for at least two clock cycles of PCLKB before setting the OSTDCR.OSTDIE bit to 1 again. A waiting time longer than two cycles of PCLKB may have to be secured, depending on the number of cycles required to read a given I/O register.
Main clock oscillation stop detection interrupts are non-maskable interrupts. Since non-maskable interrupts are disabled in the initial state after release from the reset state, enable non-maskable interrupts through software before using main clock oscillation stop detection interrupts. For details, see section 15, Interrupt Controller (ICUM).

9.9.3 Sub-Clock Oscillation Stop Detection and Operation after Detection

The sub-clock oscillation stop detection function is used to detect stopping of the sub-clock oscillator. When an oscillation stop is detected, the system clock source is switched as follows:

- If an oscillation stop is detected when SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC), the system clock source is switched to the MOCO/256 clock.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected.
Stopping of the sub-clock oscillator is detected when the input clock of the oscillation stop detection circuit remains low or high for a certain period, for example, when a malfunction occurs in the sub-clock oscillator. See section 67, Electrical Characteristics.
Switching between the sub-clock and MOCO/256 clock is controlled by the sub-clock oscillation stop detection flag (SOSTDSR.SOSTDF).
The SOSTDF flag controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC)
  - When the SOSTDF flag changes from 0 to 1, the clock source switches to the MOCO/256 clock.
  - When the SOSTDF flag changes from 1 to 0, the clock source switches back to SOSC.

To switch the clock source to the sub-clock again after detection of an oscillation stop, set the CKSEL[2:0] bits to a clock source other than the sub-clock and clear the SOSTDF flag to 0. After that, check that the SOSTDF flag is not 1, and then set the CKSEL[2:0] bits to the sub-clock after the specified oscillation stabilization wait time has elapsed.

After release from the backup area reset, the sub-clock oscillator is stopped and the sub-clock oscillation stop detection function is disabled.
To enable the sub-clock oscillation stop detection function, activate the sub-clock oscillator and write 1 to the sub-clock oscillation stop detection function enable bit (SOSTDCR.SOSTDE) after the specified oscillation stabilization wait time has elapsed.
After that, clear the sub-clock oscillation stop detection flag (SOSTDSR.SOSTDF) to 0 after the sub-clock oscillation stop detection circuit operation start wait time (t_{sosdup}) has elapsed.
When selecting the sub-clock for the system clock, complete the above settings before switching.
The sub-clock oscillation stop detection function detects when the sub-clock oscillator is stopped by an external cause.
The sub-clock oscillation stop detection function switches the following clocks to the MOCO/256 clock.

- The input clock of all modules that can select the SOSC clock as the input clock, except for RTCSCLK and CACSCLK

- The system clock (ICLK) frequency during MOCO/256 operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[3:0]).
- When stopping of the sub-clock is detected, RTCSCLK is switched to the LOCO clock.
- Even when stopping of the sub-clock is detected, CACSCLK is not switched to another clock.

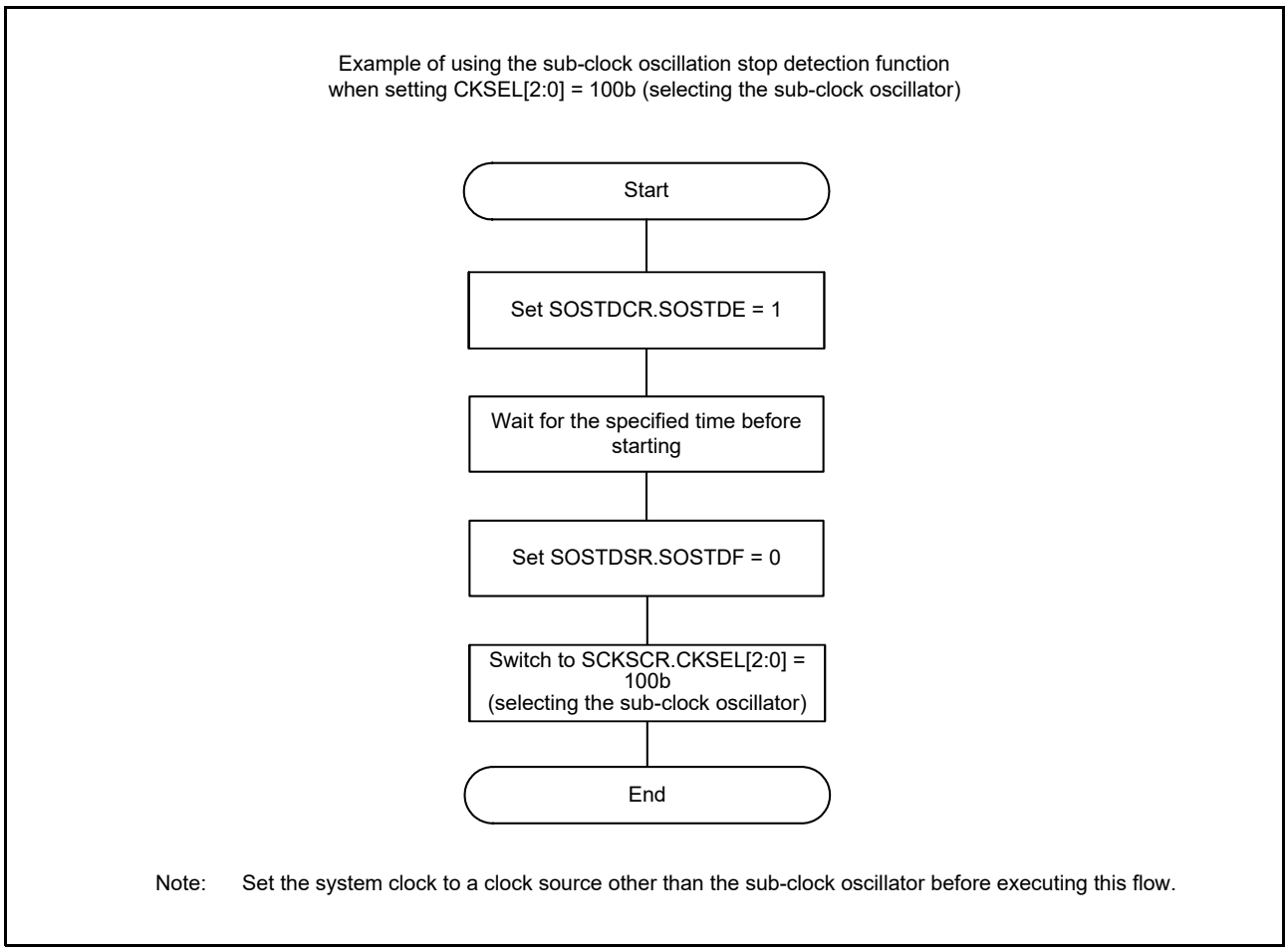
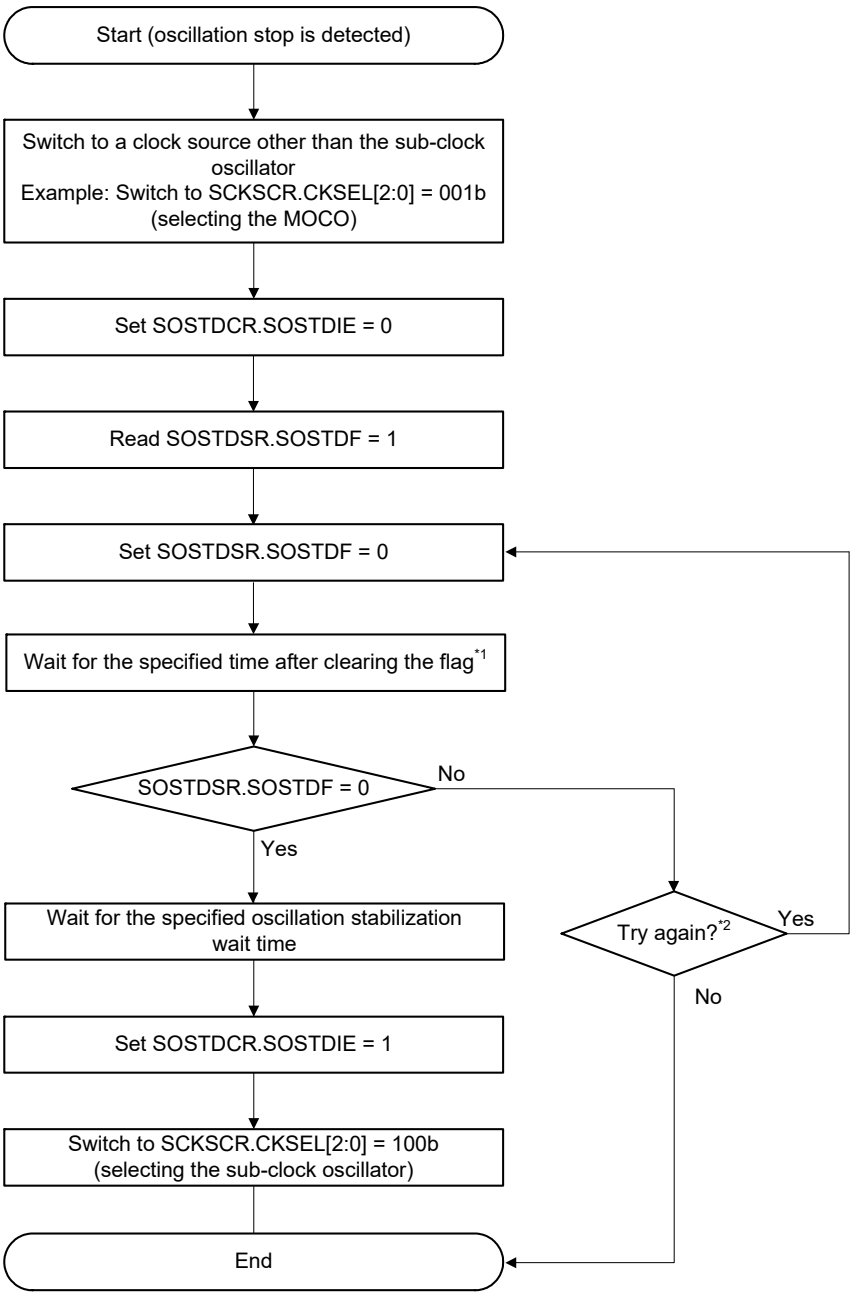


Figure 9.10 Operation Flow of the Sub-Clock Oscillation Stop Detection Function

Example of returning when setting CKSEL[2:0] = 100b (selecting the sub-clock oscillator) after an oscillation stop has been detected



Note: On return from the oscillation-stopped state, the factor responsible for stopping the sub-clock oscillation circuit must be removed from the user system to allow oscillation to resume.
Note 1. If the SOSTDSR.SOSTDF flag is cleared, wait for at least 1.5 µs before reading the next SOSTDSR.SOSTDF flag.
Note 2. Retry at least once.

Figure 9.11 Flow for Recovery from Detecting Stopping of the Sub-Clock Oscillator

9.9.4 Sub-Clock Oscillation Stop Detection Interrupts

A sub-clock oscillation stop detection interrupt (SOSTDI) is generated when the sub-clock oscillation stop detection flag (SOSTDSR.SOSTDF) is 1 and the sub-clock oscillation stop detection interrupt enable bit in the sub-clock oscillation stop detection control register (SOSTDCR.SOSTDIE) is 1 (enabled).

When the SOSTDF flag requires clearing, after setting the sub-clock oscillation stop detection interrupt enable bit in the sub-clock oscillation stop detection control register (SOSTDCR.SOSTDIE) to 0, clear the SOSTDF flag before setting the SOSTDIE bit to 1.

9.10 Internal Clocks

The clock sources for the internal clock signals are as follows:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL1 clock (PLL1P, PLL1Q, PLL1R)
- PLL2 clock (PLL2P, PLL2Q, PLL2R)
- External clock input for JTAG
- External clock input for SWD

The following internal clocks are produced from these clock sources.

- Operating clock for the CPU: CPU clock (CPUCLK0)
- Operating clock for the CPU, DMAC, EXDMAC, DTC, SRAM, system bus, I/O ports, and ICU: System clock (ICLK)
- Operating clocks of the code MRAM and MRAM code bus: MRAM code bus clock (MRICLK)
- Operating clocks of peripheral modules: Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE)
- Operating clocks of the extra MRAM and MRAM peripheral bus: MRAM peripheral bus clock (MRPCLK)
- Clocks for the external bus controller and external pin output: External bus clock (BCLK, EBCLK)
- Clock for the external bus controller and external pin output for the SDRAM: SDRAM clock (SDCLK)
- Operating clock for the RSCI: SCI clock (SCICLK)
- Operating clock for the RSPIA: SPI clock (SPICLK)
- Operating clock for the XSPI: XSPI clock (XSPICLK, XSPIDIVCLK)
- Operating clock for the CANFD and CANFDG: CANFD core clock (CANFDCLK)
- Operating clock for the USB and USBHS: USB clock (USBCLK)
- Operating clock for the USBHS: USBHS-PHY clock (USB60CLK)
- Operating clock for the ADC16H: ADC clock (ADCCLK)
- Operating clock for the GPTW, PCIF, and ADC16H: GPT clock (GPTCLK)
- Operating clock for the RI3C and RIICHS: I3C clock (I3CCLK)
- Clock for the external bus controller (asynchronous) and clock for output through an external pin: Asynchronous external bus clocks (BCLKA, EBCLK)
- Operating clock for ESWM: ESWM clock (ESWCLK)
- Operating clock for ESWM-PHY: ESWM-PHY clock (ESWPHYCLK)
- Operating clock for the ETHPHYCLK pin output: Ethernet-PHY clock (ETHPHYCLK)
- Clock for external pin output: Clock/Buzzer output (CLKOUT)
- Operating clock for the USBHS: MOSC clock for USBHS (USBMCLK)
- Operating clock for the CANFD and CANFDG: CANFD main clock (CANFDMCLK)
- Operating clock for the CAC: CAC clock (CACCLK)
- Operating clock for the RTC: RTC LOCO clock (RTCLCLK) and RTC sub-clock (RTCSCLK)
- Operating clock for the PDMIF: PDMIF clock (PDMIFCLK)
- Operating clock for the IWDT: IWDT clock (IWDTCLK)
- Operating clock for the JTAG: JTAG clock (JTAGTCK)

For details on the registers used to set the frequencies of the internal clocks, see section 9.10.4, System Clock (ICLK) to section 9.10.28, JTAG Clock.
If the value of any of these bits is changed, subsequent operation is performed at the frequency determined by the new value.

9.10.1 CPU Clock (CPUCLK0)

The CPU clock (CPUCLK0) is the operating clock for the CPU. The frequency is specified by the following bits.

- CPUCK0[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.2 MRAM Code Bus Clock (MRICLK)

The MRAM code bus clock (MRICLK) is the operating clock for the code MRAM and the MRAM code bus. The frequency is specified by the following bits.

- MRICK[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.3 MRAM Peripheral Bus Clock (MRPCLK)

The MRAM peripheral bus clock (MRPCLK) is the operating clock for extra MRAM and MRAM peripheral bus. The frequency is specified by the following bits.

- MRPCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.4 System Clock (ICLK)

The system clock (ICLK) is the operating clock for the DMAC, EXDMAC, DTC, MRAM, SRAM, system bus, I/O ports, and ICU. The frequency is specified by the following bits.

- ICK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer while the clock sources are switched. See Figure 9.12 and Figure 9.13.

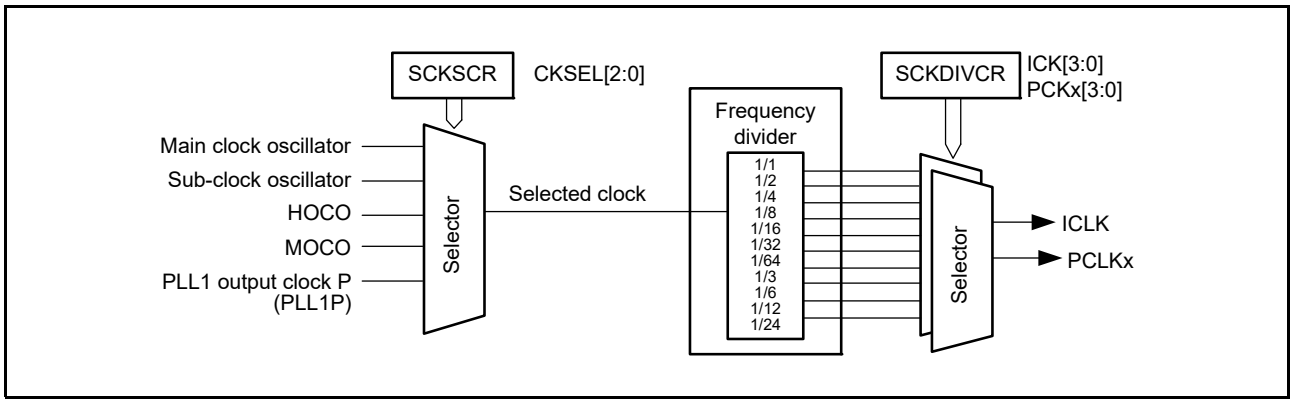


Figure 9.12 Block Diagram of Clock Source Selectors

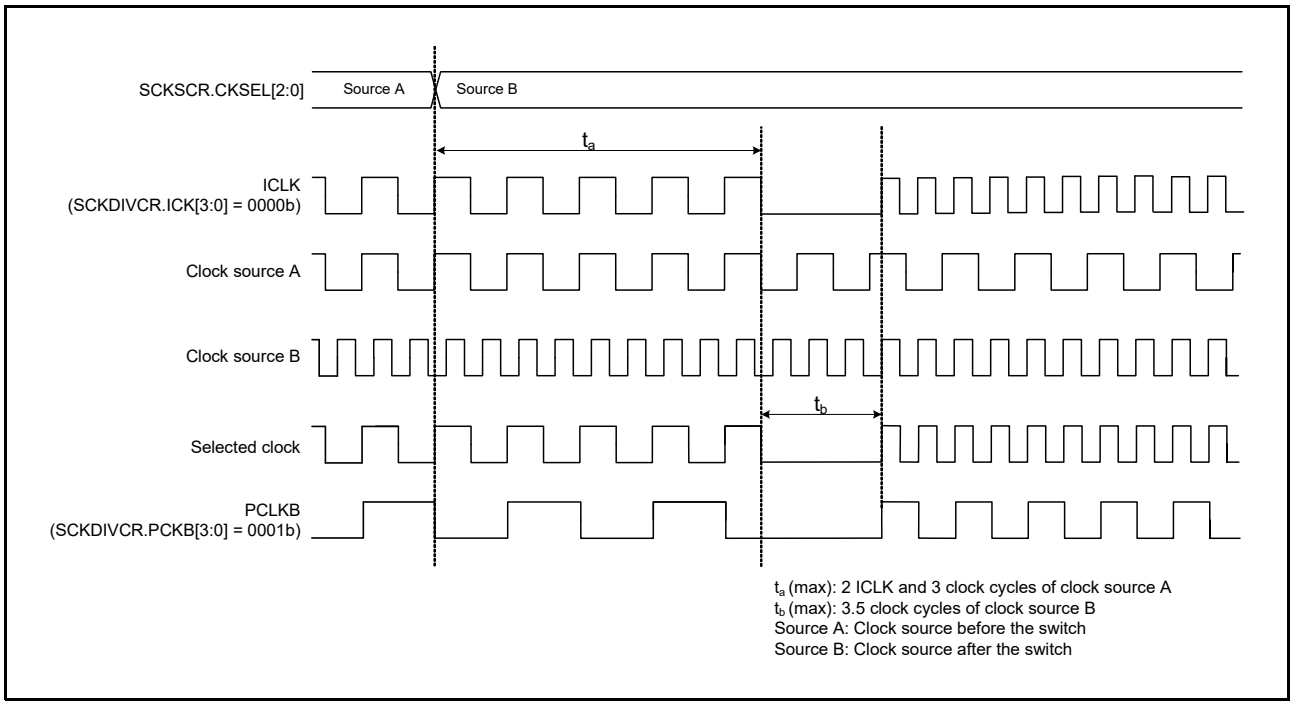


Figure 9.13 Timing of Clock Source Switching

9.10.5 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE)

The peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE) are the operating clocks for use by peripheral modules.
The frequency of the given clock is specified by the following bits.

- PCKA[3:0], PCKB[3:0], PCKC[3:0], PCKD[3:0], and PCKE[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer while the clock sources are switched. See Figure 9.12 and Figure 9.13.

9.10.6 External Bus Clock (BCLK, EBCLK)

The external bus clock (BCLK) is the operating clock for the external bus controller. It is also output externally from the EBCLK pin to the external bus.
BCLK can be output from the EBCLK pin by setting the EBCKOCR.EBCKOEN bit to 1 and the P801PFS.PSEL[6:0] bits to 0111101b. The value of the P801PFS.PSEL[6:0] bits can be changed to 0111101b only when the EBCKOCR.EBCKOEN bit is 0.
BCLK can be selected by setting the BCKCR.EBCKASEL bit to 0.
When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the EBCLK pin. The frequency is specified by the following bits.

- BCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

A frequency higher than the system clock (ICLK) should not be set for BCLK.

9.10.7 SDRAM Clock (SDCLK)

The SDRAM clock (SDCLK) is the operating clock for the external bus controller. It is output externally from the SDCLK pin to SDRAM that is connected to the external bus.
To output SDCLK from the SDCLK pin, set the SDCKOCR.SDCKOEN bit to 1 and the P700PFS.PSEL[6:0] bits to 0111101b (SDCLK output is enabled). The value of the P700PFS.PSEL[6:0] bits can be changed to 0111101b only when the SDCKOCR.SDCKOEN bit is 0. The frequency is specified by the following bits.

- BCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR

- PLODIVP[3:0] bits in PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

A frequency higher than the system clock (ICLK) should not be set for SDCLK.

9.10.8 SCI Clock (SCICLK)

The SCI clock (SCICLK) is the operating clock for the RSCI. The frequency is specified by the following bits.

- SCICKDIV[3:0] bits in SCICKDIVCR
- SCICKSEL[3:0] bits in SCICKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.9 SPI Clock (SPICLK)

The SPI clock (SPICLK) is the operating clock for the RSPIA. The frequency is specified by the following bits.

- SPICKDIV[3:0] bits in SPICKDIVCR
- SPICKSEL[3:0] bits in SPICKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.10 XSPI Clock (XSPICLK, XSPIDIVCLK)

The XSPI clock (XSPICLK) is the operating clock for the XSPI. The frequency is specified by the following bits.

- XSPICKDIV[3:0] bits in XSPICKDIVCR
- XSPICKSEL[3:0] bits in XSPICKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

XSPICLK divided by 2 is always output for XSPIDIVCLK.

9.10.11 CANFD Core Clock (CANFDCLK)

The CANFD core clock (CANFDCLK) is the operating clock for the CANFD and CANFDG. The frequency is specified by the following bits.

- CANFDCKDIV[3:0] bits in CANFDCKDIVCR
- CANFDCKSEL[3:0] bits in CANFDCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.12 USB Clock (USBCLK)

The USB clock (USBCLK) is the operating clock for the USB and USBHS.
A 48-MHz clock must be supplied when using the USB module or when using the USBHS module in classic-only mode (CL-only mode). USBCLK does not need to be supplied except for when using the USBHS in CL-only mode.
The frequency is specified by the following bits.

- USBCKDIV[3:0] bits in USBCKDIVCR
- USBCKSEL[3:0] bits in USBCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.13 USBHS-PHY Clock (USB60CLK)

The USBHS-PHY clock (USB60CLK) is the operating clock for the USBHS. A 60-MHz clock must be supplied when using the USBHS in CL-only mode. USB60CLK does not need to be supplied except for when using the USBHS in CL-only mode.
The frequency is specified by the following bits.

- USB60CKDIV[3:0] bits in USB60CKDIVCR
- USB60CKSEL[3:0] bits in USB60CKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2

- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.14 MOSC Clock for USBHS (USBMCLK)

The MOSC clock for USBHS (USBMCLK) is the operating clock for the USBHS module when it is used in modes other than CL-only mode. USBMCLK does not need to be supplied when using the USBHS module in CL-only mode.
The USBMCLK frequency is 12, 20, 24, or 48 MHz, and it is supplied from the main clock oscillator.

9.10.15 ADC Clock (ADCCLK)

The ADC clock (ADCCLK) is the operating clock for the ADC16H. The frequency is specified by the following bits.

- ADCCKDIV[3:0] bits in ADCCKDIVCR
- ADCCKSEL[3:0] bits in ADCCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.16 GPT Clock (GPTCLK)

The GPT clock (GPTCLK) is the operating clock for the GPTW, PCIF, and ADC16H. The frequency is specified by the following bits.

- GPTCKDIV[3:0] bits in GPTCKDIVCR
- GPTCKSEL[3:0] bits in GPTCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.17 I3C Clock (I3CCLK)

The I3C clock (I3CCLK) is the operating clock for the RI3C and RIICHS. The frequency is specified by the follwing bits.

- I3CCKDIV[3:0] bits in I3CCKDIVCR
- I3CCKSEL[3:0] bits in I3CCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.18 Asynchronous External Bus Clock (BCLKA, EBCLK)

The asynchronous external bus clock (BCLKA) is used as the operating clock for the external bus controller. It is also output externally from the EBCLK pin on the external bus.
BCLKA can be output from the EBCLK pin by setting the EBCKOCR.EBCKOEN bit to 1 and the P801PFS.PSEL[6:0] bits to 0111101b. Change the value of the P801PFS.PSEL[6:0] bits to 0111101b only while the EBCKOCR.EBCKOEN bit is 0.
BCLKA can be selected by setting the BCKCR.EBCKASEL bit to 1. When BCLKA is selected, set the BCKCR.BCLKDIV bit to 0. The frequency is specified by the following bits.

- BCKACKDIV[3:0] bits in BCKADIVCR
- BCKACKSEL[3:0] bits in BCKACR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.19 ESWM Clock (ESWCLK)

The ESWM clock (ESWCLK) is the operating clock for the ESWM. The frequency is specified by the following bits.

- ESWCKDIV[3:0] bits in ESWCKDIVCR
- ESWCKSEL[3:0] bits in ESWCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset,

therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.20 ESWM-PHY Clock (ESWPHYCLK)

The ESWM-PHY clock (ESWPHYCLK) is the operating clock for the ESWM. The frequency is specified by the following bits.

- ESWPCKDIV[3:0] bits in ESWPCKDIVCR
- ESWPCKSEL[3:0] bits in ESWPCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.21 Ethernet-PHY Clock (ETHPHYCLK)

The Ethernet-PHY clock (ETHPHYCLK) is the clock for the ETHPHYCLK pin output. The frequency is specified by the following bits.

- ETHPCKDIV[3:0] bits in ETHPCKDIVCR
- ETHPCKSEL[3:0] bits in ETHPCKCR
- PLLMUL[8:0], PLLMULNF[1:0], and PLIDIV[1:0] bits in PLLCCR
- PLODIVP[3:0], PLODIVQ[3:0], and PLODIVR[3:0] bits in PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], and PL2IDIV[1:0] bits in PLL2CCR
- PL2ODIVP[3:0], PL2ODIVQ[3:0], and PL2ODIVR[3:0] bits in PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.22 CANFD Main Clock (CANFDMCLK)

The CANFD main clock (CANFDMCLK) is the operating clock for the CANFD and CANFDG. CANFDMCLK is generated by the main clock oscillator.

9.10.23 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC.CACCLK includes the following clocks.

- CAC main clock (CACMCLK) that is generated by the main clock oscillator
- CAC sub-clock (CACSCLK) that is generated by the sub-clock oscillator
- CAC HOCO clock (CACHCLK) that is generated by the HOCO
- CAC MOCO clock (CACMOCLK) that is generated by the MOCO
- CAC LOCO clock (CACLCLK) that is generated by the LOCO
- CAC PLL1P clock (CACPL1PCLK) that is generated from the PLL1P clock of PLL1
- CAC PLL1Q clock (CACPL1QCLK) that is generated from the PLL1Q clock of PLL1

- CAC PLL1R clock (CACPL1RCLK) that is generated from the PLL1R clock of PLL1
- CAC PLL2P clock (CACPL2PCLK) that is generated from the PLL2P clock of PLL2
- CAC PLL2Q clock (CACPL2QCLK) that is generated from the PLL2Q clock of PLL2
- CAC PLL2R clock (CACPL2RCLK) that is generated from the PLL2R clock of PLL2

9.10.24 RTC Clock (RTCSCLK, RTCLCLK)

The RTC clock (RTCSCLK, RTCLCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO.

9.10.25 IWDT Clock (IWDTCLK)

The IWDT clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the LOCO which always divides the clock by 2.

9.10.26 PDMIF Clock (PDMIFCLK)

The PDMIF clock (PDMIFCLK) is the operating clock for the PDMIF.
PDMIFCLK is generated by the MOCO.

9.10.27 External Pin Output Clock (CLKOUT)

CLKOUT is output externally from the CLKOUT pin for the clock output or buzzer output. CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. The value of the CKODIV[2:0] or CKOSEL[3:0] bits can be changed only when the CKOCR.CKOEN bit is 0.
The frequency is specified by the following bits.

- CKODIV[2:0] and CKOSEL[3:0] bits in CKOCR
- HOCOFRQ0[2:0] bits in OFS1 and HCFRQ0[2:0] bits in HOCOCR2*1

Note 1. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.28 JTAG Clock

The JTAG clock (JTAGTCK) is the clock for the JTAG. JTAGTCK is generated from the JTAG external clock (TCK).

9.11 Clock Setting

9.11.1 Setting the System Clock

9.11.1.1 Initial Settings of the System Clock

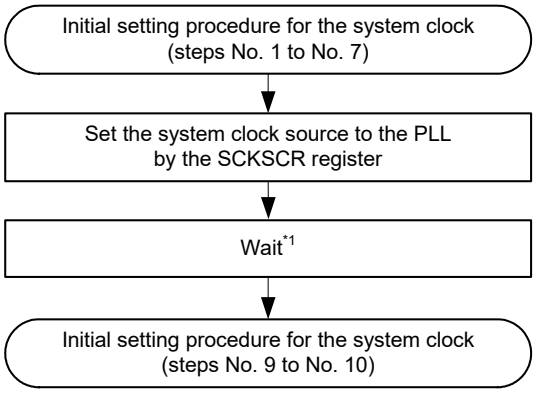
Table 9.6 shows an example of the procedure for making initial settings of the system clock.

Table 9.6 Example of the Procedure for Making Initial Settings of the System Clock (After Release from the Reset State or Deep Software Standby Mode)

No.	Step	Description
1	Start	After release from the reset state or deep software standby mode (MOCO is selected as the system clock source)
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the clock source to oscillate	Set the clock source that is switched to be the system clock source to start oscillation.
4	Set wait cycles for MRAM and SRAM	Set the number of wait cycles for access to MRAM and SRAM according to the operating frequency after the system clock (ICLK), MRICLK, and MRPCLK have been set. For details, see section 64, SRAM and section 65, MRAM.
5	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Consumption.
6	Set the internal clock frequency division ratios	Set the internal clock frequency division ratios with the SCKDIVCR and SCKDIVCR2 registers.
7	Set the external bus clock output or SDRAM clock output*1	Set the EBCLK pin output with the BCKCR register. Set the SDCLK pin output with the SDCKOCR register.
8	Switch the system clock source	Switch the system clock source with the SCKSCR register. Before switching the system clock source, make sure that oscillation of the clock source is stable.
9	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
10	End	Setup of the system clock is complete.

Note 1. This setting is not required when the external bus or SDRAM is not used.

When PLL1 is selected as the system clock source, an additional step is required at step No. 8. This additional step is shown in Figure 9.14. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, wait again after return from the interrupt.



Note 1. 30 µs is in DCDC mode, 10 µs in External VDD mode. No wait time is required in LDO mode.

Figure 9.14 Additional Flow of Initial System Clock Setting When PLL1 is Selected for the System Clock Source (Assuming All Module Stop Bits are Initial Value)

9.11.1.2 Setting the System Clock to Increase the Frequency

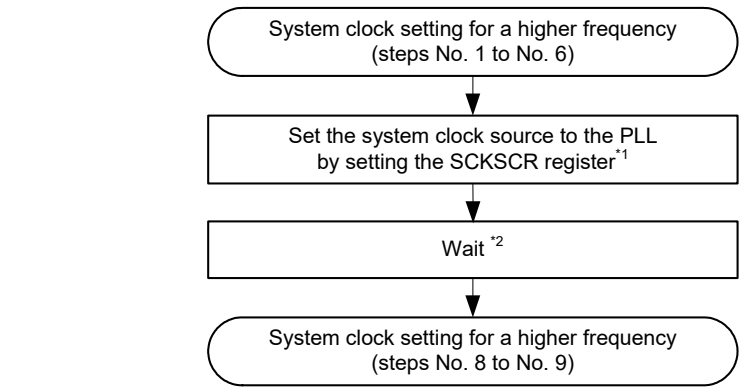
Table 9.7 shows an example of the procedure for making settings to change the current system clock frequency to a higher frequency.

Table 9.7 Example of the Procedure for Making Settings to Increase the Current System Clock Frequency

No.	Step	Description
1	Start	
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the clock source to oscillate	Set the clock source that is switched to be the system clock source to start oscillation.*1
4	Set wait cycles for MRAM and SRAM	Set the number of wait cycles for access to MRAM and SRAM according to the operating frequency after the system clock (ICLK), MRICLK, and MRPCLK have been set. For details, see section 64, SRAM and section 65, MRAM.
5	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Consumption.
6	Set the internal clock frequency division ratios	Set the internal clock frequency division ratios with the SCKDIVCR and SCKDIVCR2 registers.
7	Switch the system clock source*2	Switch the system clock source with the SCKSCR register. Before switching the system clock source, make sure that oscillation of the clock source is stable.
8	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
9	End	The frequency of the system clock becomes higher.

Note 1. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.
Note 2. This setting is not required when the system clock source is not changed.

When PLL1 is selected as the system clock source, an additional step is required at step No. 7. This additional step is shown in Figure 9.15. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, wait again after return from the interrupt.



Note 1. Do not set the SCKSCR register when the PLL is already selected.
Note 2. 30 µs is in DCDC mode, 10 µs in External VDD mode. No wait time is required in LDO mode.

Figure 9.15 Additional Flow for Setting the System Clock (When Using a PLL for the System Clock Source and CPUCLK0 and Increasing the Frequency)

9.11.1.3 Setting the System Clock to Decrease the Frequency

Table 9.8 shows an example of the procedure for making settings to change the current system clock frequency to a lower frequency.

Table 9.8 Example of the Procedure for Making Settings to Decrease the Current System Clock Frequency

No.	Step	Description
1	Start	
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the clock source to oscillate*1	Set the clock source that is switched to be the system clock source to start oscillation.*2
4	Set the internal clock frequency division ratios	Make the following settings so that the frequency does not exceed the range of the electrical characteristics when the system clock source is switched in the next step. • Set the internal clock frequency division ratios to be greater than the current division ratio. • Set the internal clock frequency division ratios so that the frequency after the switchover of the system clock source is lower than the current frequency.*1
5	Switch the system clock source*1	Switch the system clock source with the SCKSCR register. Before switching the system clock source, make sure that oscillation of the clock source is stable.
6	Re-set the internal clock frequency division ratios	Re-set the internal clock frequency division ratios with the SCKDIVCR and SCKDIVCR2 registers, according to the frequency after the system clock source has been switched.
7	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Consumption.
8	Set wait cycles for MRAM and SRAM	Set an access wait cycle count for MRAM and SRAM according to the frequency at which it operates after the system clock (ICLK), MRICLK, and MRPCLK have been set. For details, see section 64, SRAM and section 65, MRAM.
9	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
10	End	The frequency of the system clock becomes lower.

Note 1. This setting is not required when the system clock source is not changed.
Note 2. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source before switching the system clock source, additional procedure is required before procedure No.4. This additional step is shown in Figure 9.16. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, wait again after return from the interrupt.

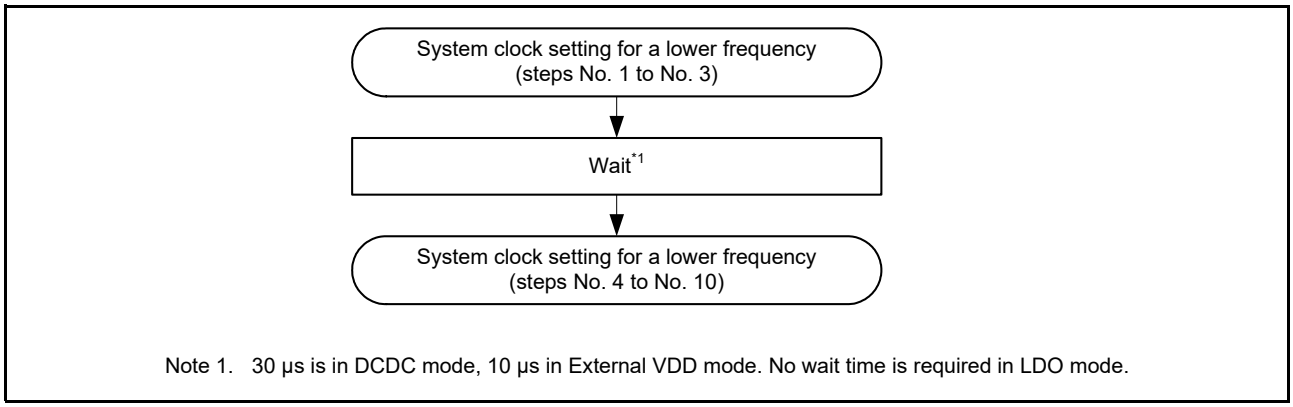


Figure 9.16 Additional Flow for Setting the System Clock (When Using a PLL for the System Clock Source and CPUCLK0 and Decreasing the Frequency)

9.11.2 Setting Peripheral Module-Dedicated Clocks

The procedures for setting peripheral module-dedicated clocks are described. The target clocks are SCICLK, SPICLK, XSPICLK, CANFDCLK, USBCLK, USB60CLK, ADCCLK, GPTCLK, I3CCLK, BCLKA, ESWCLK, ESWPHYCLK, and ETHPHYCLK.

9.11.2.1 Initial Settings of a Peripheral Module-Dedicated Clock

Table 9.9 shows an example of the procedure for making initial settings of a peripheral module-dedicated clock.

Table 9.9 Example of the Procedure for Making Initial Settings of a Peripheral Module-Dedicated Clock (After Release from the Reset State or Deep Software Standby Mode)

No.	Step	Description
1	Start	After release from the reset state or deep software standby mode (MOCO is selected as the system clock source)
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the CKSREQ bit to 1 and wait for the CKSRDY flag to be set	Write 1 to the CKSREQ bit in each CKCR register. Carry out polling until the CKSRDY flag in each CKCR register is read as 1. When the CKSRDY flag becomes 1, the clock supply is stopped.
4	Set the clock source to oscillate	Set the clock source that is switched to be the system clock source to start oscillation.
5	Set the clock frequency division ratio and switch the clock source	Write the setting value to the CKDIV[3:0] bits in each CKDIVCR register. Write the setting value to the CKSEL[3:0] bits in each CKCR register.
6	Set the CKSREQ bit to 0 and wait for the CKSRDY flag to be cleared	Write 0 to the CKSREQ bit in each CKCR register. Carry out polling until the CKSRDY flag in each CKCR register is read as 0. When the CKSRDY flag becomes 0, the clock supply is started.
7	Cancel the module stop control	Cancel the module stop control in the MSTPCRn register (n = A to F) and supply the clock to the peripheral module.
8	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
9	End	Setup of the clock is complete. After that, set the peripheral module.

9.11.2.2 Changing the Clock Source of a Peripheral Module-Dedicated Clock

Table 9.10 shows an example of the procedure for changing the settings when switching the clock source of a peripheral module-dedicated clock.

Table 9.10 Example of the Procedure for Changing the Settings When Switching the Clock Source of a Peripheral Module-Dedicated Clock

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module whose operating clock is to be changed.
3	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
4	Set the CKSREQ bit to 1 and wait for the CKSRDY flag to be set	Write 1 to the CKSREQ bit in each CKCR register. Carry out polling until the CKSRDY flag in each CKCR register is read as 1. When the CKSRDY flag becomes 1, the clock supply is stopped.
5	Set the clock source to oscillate	Set the clock source that is switched to be the system clock source to start oscillation.
6	Set the clock frequency division ratio and switch the clock source	Write the setting value to the CKDIV[3:0] bits in each CKDIVCR register. Write the setting value to the CKSEL[3:0] bits in each CKCR register.
7	Set the CKSREQ bit to 0 and wait for the CKSRDY flag to be cleared	Write 0 to the CKSREQ bit in each CKCR register. Carry out polling until the CKSRDY flag in each CKCR register is read as 0. When the CKSRDY flag becomes 0, the clock supply is started.
8	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
9	Restart the peripheral module	After the settings of the operating clock have been changed, in accordance with those settings, change the settings of the peripheral module. After that, restart the operation of the peripheral module.
10	End	Setup of the clock is complete.

9.11.2.3 Changing the Division Ratio of a Peripheral Module-Dedicated Clock

Table 9.11 shows an example of the procedure for changing the settings when switching the division ratio of a peripheral module-dedicated clock.

Table 9.11 Example of the Procedure for Changing the Settings When Switching the Division Ratio of a Peripheral Module-Dedicated Clock (No Change in the Clock Source)

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module whose operating clock is to be changed.
3	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
4	Set the clock frequency division ratio	Write the setting value to the CKDIV[3:0] bits in each CKDIVCR register.
5	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
6	Restart the peripheral module	After the settings of the operating clock have been changed, in accordance with those settings, change the settings of the peripheral module. After that, restart the operation of the peripheral module.
7	End	Setup of the clock is complete.

9.11.3 Setting the Main Clock Oscillator

Table 9.12 shows an example of the procedure for making initial settings of the main clock oscillator.

Table 9.12 Example of the Procedure for Making Initial Settings of the Main Clock Oscillator (After Release from the Reset State or Deep Software Standby Mode)

No.	Step	Description
1	Start	The main clock oscillator is stopped after release from the reset state or deep software standby mode.
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the mode of the main clock oscillator	Set the following main clock oscillator mode with the MOMCR register. • Resonator or clock input • Driving capacity
4	Set the oscillation stabilization wait time	Set the oscillation stabilization wait time of the main clock oscillator in the MOSCWTCR register.
5	Set oscillation keeping in software standby mode	Set the MOSCSCR register if the main clock oscillator keeps oscillating in software standby mode.
6	Set the main clock oscillator to operate	Set the main clock oscillator to start oscillating with the MOSCCR register.
7	Wait for main clock oscillation to become stable	Carry out polling until the MOSCSF flag in the OSCSF register is read as 1 (oscillation is stable).
8	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
9	End	Setup of the main clock oscillator is complete. The main clock is available.

9.11.4 Setting the Sub-Clock Oscillator

Table 9.13 shows an example of the procedure for making initial settings of the sub-clock oscillator.

Table 9.13 Example of the Procedure for Making Initial Settings of the Sub-Clock Oscillator (After Release from the Backup Domain Reset State)

No.	Step	Description
1	Start	The sub-clock oscillator is stopped after release from the backup domain reset state.
2	Cancel write protection for registers	Set the PRC0 bit in the PRCR register to 1.
3	Set the mode of the sub-clock oscillator	Set the following sub-clock oscillator mode with the SOMCR register. • Resonator or clock input • Driving capacity
4	Set the sub-clock oscillator to operate	Set the sub-clock oscillator to start oscillating with the SOSCCR register.
5	Wait for sub-clock oscillation to become stable	Wait for the oscillation stabilization wait time of the sub-clock (tSUBOSCWT) to elapse. For the oscillation stabilization wait time of the sub-clock, see section 67, Electrical Characteristics.
6	Set write protection for registers	Set the PRC0 bit in the PRCR register to 0.
7	End	Setup of the sub-clock oscillator is complete. The sub-clock is available.

9.11.5 Setting the High-Speed On-Chip Oscillator

9.11.5.1 Initial Settings of the HOCO Clock

Table 9.14 shows an example of the procedure for making initial settings of the HOCO.

Table 9.14 Example of the Procedure for Making Initial Settings of the HOCO (After Release from the Reset State or Deep Software Standby Mode) (OFS1.HOCOEN = 1, without the FLL Function)

No.	Step	Description
1	Start	The HOCO is stopped after release from the reset state or deep software standby mode when the OFS1.HOCOEN bit is 1.
2	Cancel write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 1.
3	Check the HOCO power supply*1	Confirm that the following bit in the HOCOLDOCR register is set as shown. • LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with the HOCOCR2 register.
5	Set oscillation keeping in software standby mode	Set the following if the HOCO keeps oscillating in software standby mode. • HOCOSOKP bit in the HOCOSCR register • SKEEP bit in the HOCOLDOCR register
6	Set the HOCO to operate	Set the HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to become stable	Carry out polling until the HOCOSF flag in the OSCSF register is read as 1 (oscillation is stable).
8	Set write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 0.
9	End	Setup of the HOCO is complete. The HOCO clock is available.

Note 1. When the HOCOLDOCR.LDOSTP bit is 1, do not operate the HOCO. If the HOCOLDOCR.LDOSTP bit is switched from 1 to 0, it is necessary to wait until the HOCO power supply has stabilized. For the wait time required until the HOCO power supply becomes stable, see section 67, Electrical Characteristics.

9.11.5.2 Setting the HOCO by the FLL Function

Table 9.15 shows an example of the procedure for making initial settings of the HOCO using the FLL function.
Table 9.16 shows the flow for setting the FLL function when placing the MCU in software standby mode and releasing the MCU from software standby mode.

Table 9.15 Example of the Procedure for Making Initial Settings of the HOCO Using the FLL Function (After Release from the Reset State or Deep Software Standby Mode) (OFS1.HOCOEN = 1)

No.	Step	Description
1	Start	The HOCO is stopped after release from the reset state or deep software standby mode when the OFS1.HOCOEN bit is 1.
2	Cancel write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 1.
3	Check the HOCO power supply*1	Confirm that the following bit in the HOCOLDOCR register is set as shown. • LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with the HOCOCR2 register.
5	Enable the FLL function*2	Set FLL multiplication control with the FLLCR2 register. Set the FLL function to be enabled with the FLLCR1 register.
6	Set the HOCO to operate	Set the HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to become stable	Carry out polling until the HOCOSF flag in the OSCSF register is read as 1 (oscillation is stable).
8	Wait for FLL stabilization	Wait for the FLL stabilization wait time (tFLLWT) to elapse or wait until the frequency accuracy being stable can be confirmed by measuring the HOCO clock.
9	Check the HOCO stabilization	Check that the HOCOSF flag in the OSCSF register is read as 1.
10	Set write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 0.
11	End	Setup of the HOCO is complete. The HOCO clock is available.

Note 1. When the HOCOLDOCR.LDOSTP bit is 1, do not operate the HOCO. If the HOCOLDOCR.LDOSTP bit is switched from 1 to 0, it is necessary to wait until the HOCO power supply has stabilized. For the wait time required until the HOCO power supply becomes stable, see section 67, Electrical Characteristics.
Note 2. The sub-clock oscillator must be oscillating stably.

Table 9.16 Flow for Setting the FLL Function at a Transition to Software Standby Mode and Release from Software Standby Mode

No.	Step	Description
1	Start	The HOCO is operating with the FLL function enabled.
2	Cancel write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 1.
3	Stop the HOCO*1	Stop the HOCO with the HOCOCR register.
4	Disable the FLL function	Set the FLL function to be disabled with the FLLCR1 register.
5	WAIT instruction*2	The MCU is placed in software standby mode by the WAIT instruction.
6	Software standby mode*2	The MCU is in software standby mode.
7	Release from software standby mode*2	Release the MCU from software standby mode.
8	Enable the FLL function*3	Set FLL multiplication control with the FLLCR2 register. Set the FLL function to be enabled with the FLLCR1 register.
9	Set the HOCO to operate	Set the HOCO to start oscillating with the HOCOCR register.
10	Wait for FLL stabilization	Wait for the FLL stabilization wait time (tFLLWT) to elapse or wait until the frequency accuracy being stable can be confirmed by measuring the HOCO clock at CAC.
11	Check the HOCO stabilization	Check that the HOCOSF flag in the OSCSF register is read as 1.
12	Set write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 0.
13	End	The HOCO clock is available.

Note 1. If the HOCO is used as the system clock source or the clock source of PLL, these clock sources must be changed to another clock before the HOCO is stopped.
Note 2. For details on software standby mode, see section 11, Low Power Consumption.
Note 3. The sub-clock oscillator must be oscillating stably.

9.11.6 Setting PLL Circuits

9.11.6.1 Initial Settings of a PLL Clock

Table 9.17 shows an example of the procedure for making initial settings of a PLL.

Table 9.17 Example of the Procedure for Making Initial Settings of a PLL (After Release from the Reset State or Deep Software Standby Mode) (PLL1 or PLL2)

No.	Step	Description
1	Start	PLL1 is stopped after release from the reset state or deep software standby mode. PLL2 is stopped after release from the reset state or deep software standby mode.
2	Cancel write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 1.
3	Check the PLL power supply PLL*1	[PLL1] Confirm that the following bit in the PLL1LDOCR register is set as shown. • LDOSTP bit is 0 (LDO is enabled) [PLL2] Confirm that the following bit in the PLL2LDOCR register is set as shown. • LDOSTP bit is 0 (LDO is enabled)
4	Set the PLL	[PLL1] Make the following PLL1 settings with the PLLCCR and PLLCCR2 registers. [PLL2] Make the following PLL2 settings with the PLL2CCR and PLL2CCR2 registers. [Setting items] • PLL input frequency division ratio • PLL clock source • Frequency multiplication ratio • Frequency multiplication fractional ratio • PLL output frequency division ratio (P/Q/R)
5	Set the PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
6	Wait for PLL clock oscillation to become stable	[PLL1] Carry out polling until the PLLSF flag in the OSCSF register is read as 1 (oscillation is stable). [PLL2] Carry out polling until the PLL2SF flag in the OSCSF register is read as 1 (oscillation is stable).
7	Set write protection for registers	Set the PRC0 and PRC1 bits in the PRCR register to 0.
8	End	Setup of the PLL1 or PLL2 is complete. [PLL1] The PLL1P, PLL1Q, or PLL1R clock is available. [PLL2] The PLL2P, PLL2Q, or PLL2R clock is available.

Note 1. When the PLL1LDOCR.LDOSTP bit is 1, do not operate PLL1. If the PLL1LDOCR.LDOSTP bit is switched from 1 to 0, it is necessary to wait until the PLL1 power supply has stabilized. For the wait time required until the PLL1 power supply becomes stable, see section 67, Electrical Characteristics.
When the PLL2LDOCR.LDOSTP bit is 1, do not operate PLL2. If the PLL2LDOCR.LDOSTP bit is switched from 1 to 0, it is necessary to wait until the PLL2 power supply has stabilized. For the wait time required until the PLL2 power supply becomes stable, see section 67, Electrical Characteristics.

9.11.6.2 Changing the PLL Settings

Table 9.18 shows an example of the procedure for changing the PLL settings.

Table 9.18 Example of the Procedure for Changing the PLL Settings (PLL1 or PLL2)

No.	Step	Description
1	Start	PLL1 is oscillating and the PLL1 clocks (PLL1P, PLL1Q, and PLL1R) are being used. PLL2 is oscillating and the PLL2 clocks (PLL2P, PLL2Q, and PLL2R) are being used.
2	Cancel write protection for registers	Set the PRC0 and PRC1*1 bits in the PRCR register to 1.
3	Stop the PLL*1	[PLL1] Stop PLL1 with the PLLCR register. [PLL2] Stop PLL2 with the PLL2CR register.
4	Change the PLL settings	[PLL1] Make the following PLL1 settings with the PLLCCR and PLLCCR2 registers. [PLL2] Make the following PLL2 settings with the PLL2CCR and PLL2CCR2 registers. [Setting items] • PLL input frequency division ratio • PLL clock source • Frequency multiplication ratio • Frequency multiplication fractional ratio • PLL output frequency division ratio (P/Q/R)
5	Set the PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
6	Wait for PLL clock oscillation to become stable	[PLL1] Carry out polling until the PLLSF flag in the OSCSF register is read as 1 (oscillation is stable). [PLL2] Carry out polling until the PLL2SF flag in the OSCSF register is read as 1 (oscillation is stable)
7	Set write protection for registers	Set the PRC0 and PRC1*1 bits in the PRCR register to 0.
8	End	Setup of the PLL1 or PLL2 is complete. [PLL1] The PLL1P, PLL1Q, or PLL1R clock is available. [PLL2] The PLL2P, PLL2Q, or PLL2R clock is available.

Note 1. If the PLL1P clock is used as the system clock source, the system clock source must be changed to another clock source before PLL1 is stopped.

9.12 Usage Notes

9.12.1 Notes on the Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the settings of the SCKDIVCR and SCKDIVCR2 registers.

- CPU clock (CPUCLK0)
- MRAM code bus clock (MRICLK)
- System clock (ICLK)
- Peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE)
- MRAM peripheral bus clock (MRPCLK)
- External bus clock (BCLK)

Each frequency must meet the following conditions.

- Each frequency must be selected within the operation guaranteed range of the operating frequency (f) that is specified in the AC characteristics. See section 67, Electrical Characteristics.
- Each clock must be set according to Table 9.2.
- Do not change the clock frequency during external bus accesses. Furthermore, when an external bus access is to start after the clock frequency has been changed, make sure to start the external bus access after confirming that the frequency change has been completed.

To ensure correct processing after changing the clock frequency, first write to the relevant clock control register to change the frequency, then read the register value to confirm the change, and finally execute the subsequent processing.

9.12.2 Restrictions on a Resonator

Since resonator characteristics relate closely to the user's board design, adequate evaluation is required before use. For an example of connecting a resonator, see Figure 9.5. The circuit constants for a resonator vary depending on the resonator to be used and the stray capacitance of the circuit on which it is to be mounted. Therefore, the circuit constants should always be determined in consultation with the resonator manufacturer. The voltage to be applied to resonator connection pins must not exceed the absolute maximum rating.

9.12.3 Notes on Board Design

When using a crystal resonator, place the resonator and load capacitors as close to the XTAL and EXTAL pins as possible. Signal lines should be routed away from the oscillation circuit as shown in Figure 9.17 to prevent electromagnetic induction from causing abnormal oscillation. Figure 9.17 shows a case in which the main clock oscillator is used. For a case in which the sub-clock oscillator is used, the figure will be the same as Figure 9.17.

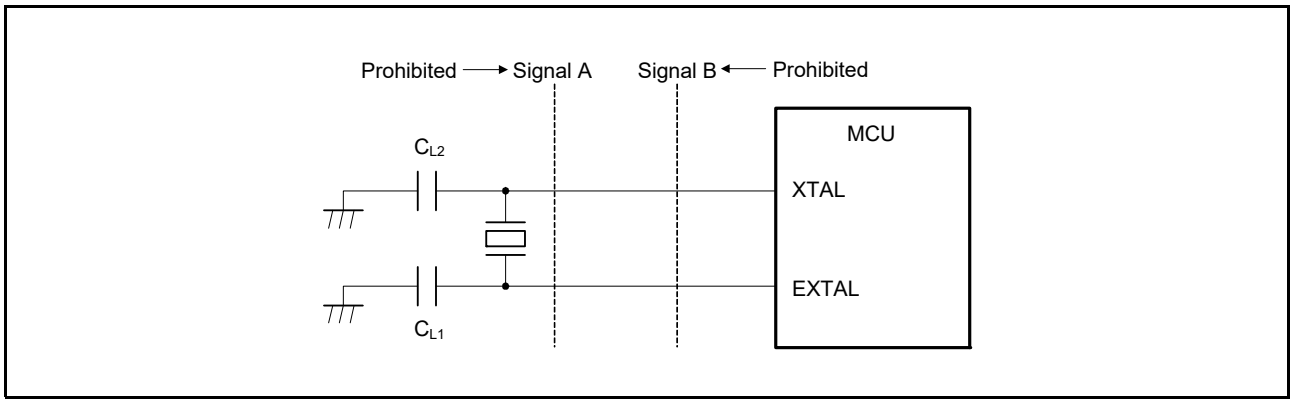


Figure 9.17 Selecting Signal Paths in Oscillation Circuit Board Design

9.12.4 Notes on Resonator Connection Pins

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).
When the sub-clock is not used, the XCIN and XCOUT pins can be used as general ports. When these pins are used as general ports, the sub-clock must be stopped (SOSCCR.SOSTP bit should be set to 1).

9.12.5 Notes on Using the Sub-Clock Oscillator

The output of the P306 (EXTAL) and P307 (XTAL) pins may affect the oscillation of the sub-clock oscillator.
If the sub-clock oscillator is used, design the board so as not to affect oscillation. When using the P306 (EXTAL) and P307 (XTAL) pins as output pins while using the sub-clock oscillator, setting the P306PFS.DSCR[3:0] and P307PFS.DSCR[3:0] bits to 0000b or 0001b is strongly recommended.
In addition, when using the P306 (EXTAL) and P307 (XTAL) pins as output pins while using the sub-clock oscillator in a low-power mode (SOMCR.SODRV[1:0] = 01b or 10b), setting the P306PFS.DSCR[3:0] and P307PFS.DSCR[3:0] bits to 0000b is recommended.

9.12.6 Notes on the Reset Conditions of the SOSC and LOCO

The RTC is not initialized by any reset source.
The SOSC and LOCO can be selected as the clock source of the RTC. Table 9.19 and Table 9.20 show the states of the SOSC and LOCO when a reset occurs.

Table 9.19 State of the SOSC at a Reset

		Reset Source	
		Backup Area Reset	**Others**
SOSC	Enabled/Disabled	Initialized	The state before the reset is retained.
	Driving capacity	Initialized	The state before the reset is retained.

Table 9.20 State of the LOCO at a Reset

		Reset Source	
		Power-On Reset, Voltage-Monitoring 0, 4, or 5 Reset, Core Voltage Monitoring Reset, or Deep Software Standby 2 or 3 Reset	**Others**
LOCO	Enabled/Disabled	Initialized	
	Oscillation accuracy*1	Initialized to the accuracy before trimming by the LOCOUTCR register (accuracy: ±15%)	The accuracy that was trimmed by the LOCOUTCR register is kept.

Note 1. If the LOCO frequency trimmed by the LOCOUTCR register is selected as the RTC clock source, note that the oscillation accuracy of the LOCO will be initialized by a power-on reset, voltage-monitoring 0, 4, or 5 reset, core voltage monitoring reset, or deep software standby 2 or 3 reset.

After release from the reset state, the reset exception handling starts.
For details on the reset exception handling, see section 6, Resets.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

This MCU incorporates three units of clock frequency accuracy measurement circuits (CAC0 to CAC2).
The CAC counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

10.1 Overview

Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

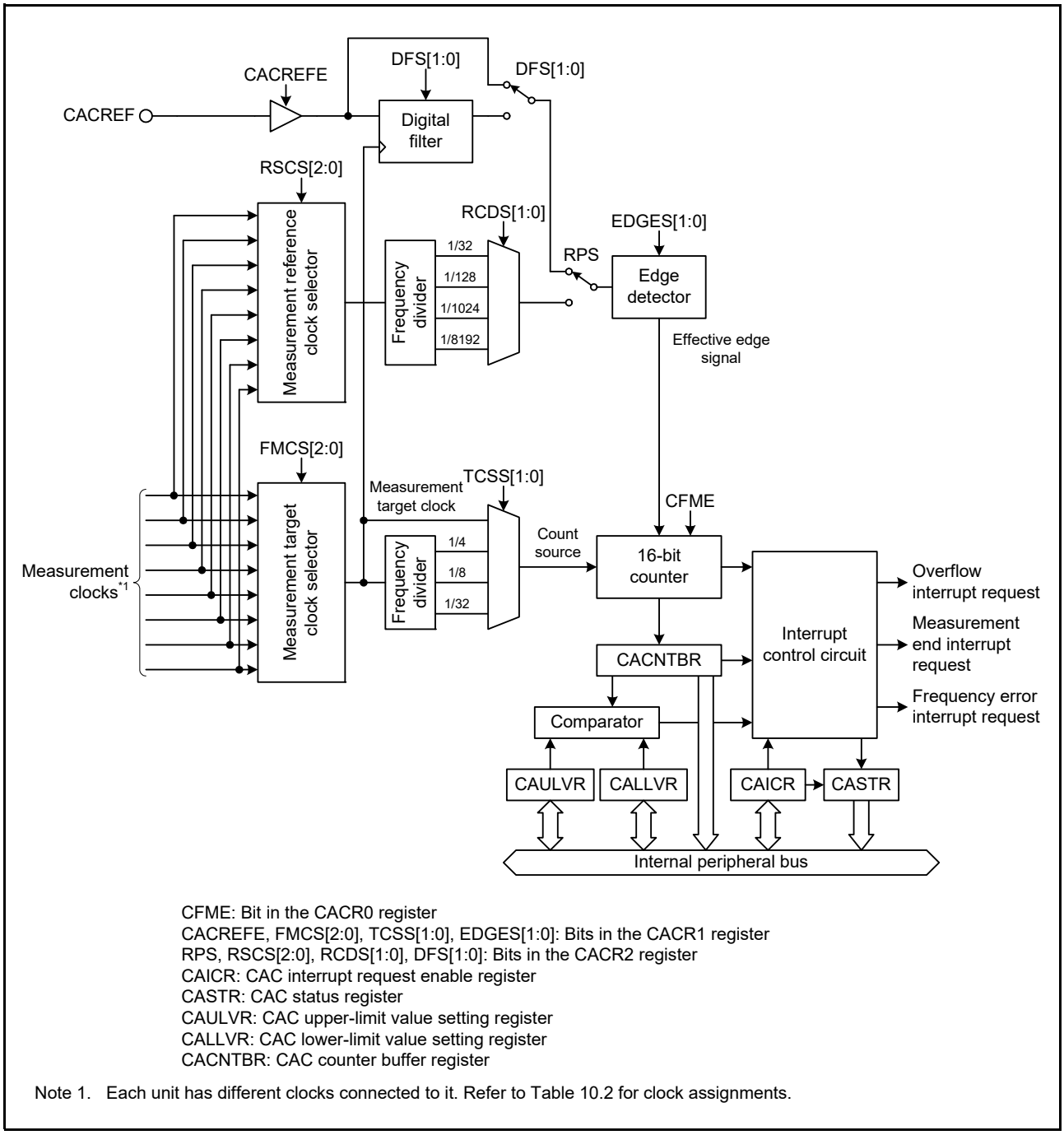
Table 10.1 CAC Specifications

Item	Description
Measurement target clocks*1	The frequency of the following clocks can be measured. • Main clock • Sub-clock • HOCO clock • MOCO clock • LOCO clock • Peripheral module clock B (PCLKB) • PLL1 output clock P/Q/R (divided by 16) • PLL2 output clock P/Q/R (divided by 16)
Measurement reference clocks*1	• External clock input to the CACREF pin • Main clock • Sub-clock • HOCO clock • MOCO clock • LOCO clock • Peripheral module clock B (PCLKB) • PLL1 output clock P/Q/R (divided by 16) • PLL2 output clock P/Q/R (divided by 16)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attribute can be set.*2

Note 1. Each unit has different clocks connected to it. Refer to Table 10.2 for clock assignments.
Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 10.2 Clocks Connected to Each Unit

Unit 0	Unit 1	Unit 2
Main clock	Main clock	Main clock
Sub-clock	HOCO clock	HOCO clock
HOCO clock	MOCO clock	MOCO clock
MOCO clock	Peripheral module clock B (PCLKB)	Peripheral module clock B (PCLKB)
LOCO clock	PLL1 output clock P (divided by 16)	PLL2 output clock P (divided by 16)
Peripheral module clock B (PCLKB)	PLL1 output clock Q (divided by 16)	PLL2 output clock Q (divided by 16)
	PLL1 output clock R (divided by 16)	PLL2 output clock R (divided by 16)



CFME: Bit in the CACR0 register
CACREFE, FMCS[2:0], TCSS[1:0], EDGES[1:0]: Bits in the CACR1 register
RPS, RSCS[2:0], RCDS[1:0], DFS[1:0]: Bits in the CACR2 register
CAICR: CAC interrupt request enable register
CASTR: CAC status register
CAULVR: CAC upper-limit value setting register
CALLVR: CAC lower-limit value setting register
CACNTBR: CAC counter buffer register

Note 1. Each unit has different clocks connected to it. Refer to Table 10.2 for clock assignments.

Figure 10.1 CAC Block Diagram

Table 10.3 shows the pin configuration of the CAC.

Table 10.3 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC0.CACR0 8780 2400h, CAC1.CACR0 8780 2480h, CAC2.CACR0 8780 2500h, CAC0_NS.CACR0 0780 2400h, CAC1_NS.CACR0 0780 2480h, CAC2_NS.CACR0 0780 2500h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

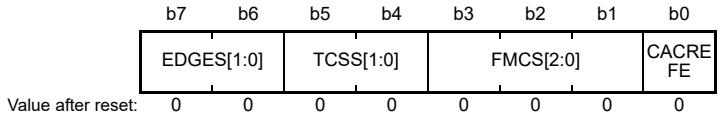
CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC0.CACR1 8780 2401h, CAC1.CACR1 8780 2481h, CAC2.CACR1 8780 2501h, CAC0_NS.CACR1 0780 2401h, CAC1_NS.CACR1 0780 2481h, CAC2_NS.CACR1 0780 2501h



Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	• Unit 0 b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited. • Unit 1 b3 b1 0 0 0: Main clock 0 0 1: PLL1 output clock P (divided by 16) 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: PLL1 output clock Q (divided by 16) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: PLL1 output clock R (divided by 16) 1 1 1: Setting prohibited. • Unit 2 b3 b1 0 0 0: Main clock 0 0 1: PLL2 output clock P (divided by 16) 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: PLL2 output clock Q (divided by 16) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: PLL2 output clock R (divided by 16) 1 1 1: Setting prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Source Select)

These bits select the count source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC0.CACR2 8780 2402h, CAC1.CACR2 8780 2482h, CAC2.CACR2 8780 2502h, CAC0_NS.CACR2 0780 2402h, CAC1_NS.CACR2 0780 2482h, CAC2_NS.CACR2 0780 2502h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	• Unit 0 b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited. • Unit 1 b3 b1 0 0 0: Main clock 0 0 1: PLL1 output clock P (divided by 16) 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: PLL1 output clock Q (divided by 16) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: PLL1 output clock R (divided by 16) 1 1 1: Setting prohibited. • Unit 2 b3 b1 0 0 0: Main clock 0 0 1: PLL2 output clock P (divided by 16) 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: PLL2 output clock Q (divided by 16) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: PLL2 output clock R (divided by 16) 1 1 1: Setting prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): CAC0.CAICR 8780 2403h, CAC1.CAICR 8780 2483h, CAC2.CAICR 8780 2503h, CAC0_NS.CAICR 0780 2403h, CAC1_NS.CAICR 0780 2483h, CAC2_NS.CAICR 0780 2503h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC0.CASTR 8780 2404h, CAC1.CASTR 8780 2484h, CAC2.CASTR 8780 2504h, CAC0_NS.CASTR 0780 2404h, CAC1_NS.CASTR 0780 2484h, CAC2_NS.CASTR 0780 2504h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC0.CAULVR 8780 2406h, CAC1.CAULVR 8780 2486h, CAC2.CAULVR 8780 2506h, CAC0_NS.CAULVR 0780 2406h, CAC1_NS.CAULVR 0780 2486h, CAC2_NS.CAULVR 0780 2506h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CAULVR register is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC0.CALLVR 8780 2408h, CAC1.CALLVR 8780 2488h, CAC2.CALLVR 8780 2508h, CAC0_NS.CALLVR 0780 2408h, CAC1_NS.CALLVR 0780 2488h, CAC2_NS.CALLVR 0780 2508h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CALLVR register is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC0.CACNTBR 8780 240Ah, CAC1.CACNTBR 8780 248Ah, CAC2.CACNTBR 8780 250Ah, CAC0_NS.CACNTBR 0780 240Ah, CAC1_NS.CACNTBR 0780 248Ah, CAC2_NS.CACNTBR 0780 250Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CACNTBR register is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

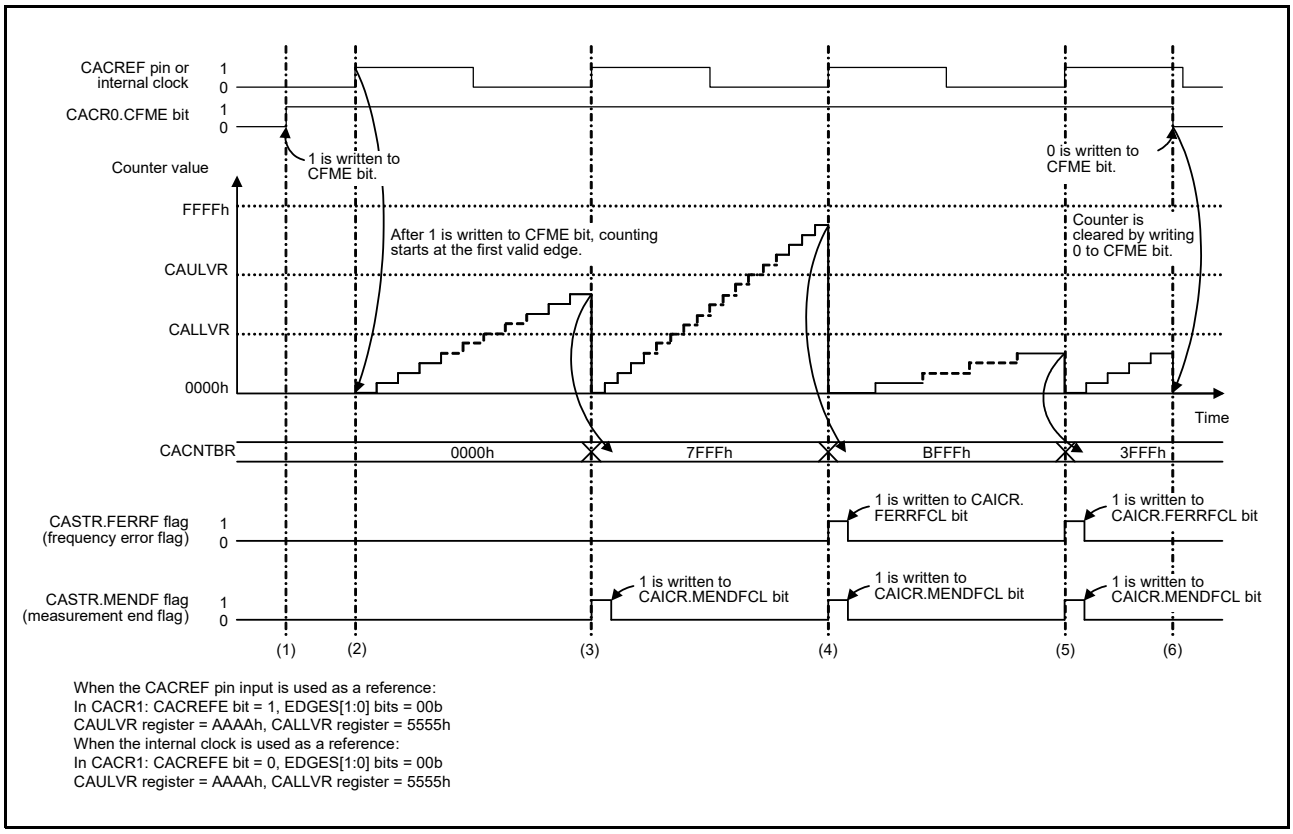


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (the CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (the CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. If the formula CALLVR ≤ CACNTBR ≤ CAULVR is satisfied, only the CASTR.MENDF flag is set to 1 because the clock frequency is correct. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR > CAULVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is

generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR < CALLVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(6) While the CACR0.CFME bit is 1, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers every time a valid edge is input. Writing 0 to the CACR0.CFME bit clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in the CACNTBR register may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.4 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.4 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, EBCLK output control, SDCLK output control, stopping modules, power gating control, and transitions to low power consumption modes.

Table 11.1 lists the specifications of low power consumption functions. Table 11.2 to Table 11.4 list the conditions to shift to low power consumption modes, states of the CPU and peripheral modules, and the method for release from each mode. After a reset, this MCU enters the normal program execution state, but modules except for the DMAC, DTC, and RAM do not operate.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the CPU clock (CPUCLK0), system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKD, PCLKE), external bus clock (BCLK), MRAM code bus clock (MRICLK), and MRAM peripheral bus clock (MRPCLK).*1
EBCLK output control function	BCLK output or high-level output can be selected.*1
SDCLK output control function	SDCLK output or high-level output can be selected.*1
Module-stop function	Functions can be stopped independently for each peripheral module.
Power gating control function	The power state can be controlled for each power domain. • Control the ON/OFF state of each power domain. • Control the retention of specific circuits during power gating.
Transition to the low power consumption state	• Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Processor low power consumption modes	• CPU sleep mode • CPU deep sleep mode
Low power consumption modes	• Software standby mode 1, 2*2 • Deep software standby mode 1, 2, 3*2

Note 1. For details, see section 9, Clock Generation Circuit.
Note 2. This mode is not supported in external VDD mode.

Table 11.2 Entering and Exiting Processor Low Power Consumption Modes and Operating States in Each Mode

Item	CPU Sleep Mode	CPU Deep Sleep Mode
Transition condition	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 0h and the CLPCR.DEEPSLEEP bit is 0: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to CPU sleep mode is complete after execution of the WAIT instruction).	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 0h and the CLPCR.DEEPSLEEP bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to CPU sleep mode is complete after execution of the WAIT instruction).
Method for release from each mode	Interrupt	Interrupt
	Reset	Reset
State after release by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after release by a reset	Reset state	Reset state
CPU	Stopped (retained)	Stopped (retained)*2
TFU	Operation possible	Stopped (retained)*2
DMA controller (DMAC)	Operation possible	Operation possible
Data transfer controller (DTC)	Operation possible	Operation possible
Watchdog timer (WDT)	Operation possible*3	Operation possible*3
Independent watchdog timer (IWDT)	Operation possible*4	Operation possible*4
Other peripheral modules	Operation possible	Operation possible

Note: “Operation possible” means that operation or stopping can be controlled by setting the control register.
“Stopped (retained)” means that internal register values are retained and internal operations are suspended.

Note 1. “Valid interrupt request” refers to any interrupt or exception that is not masked by the priority level of a current exception or the priority level specified by the IPRr register.

Note 2. The power is turned off.

Note 3. In WDT auto-start mode, operation or stopping can be selected by setting the WDT sleep mode count stop control bit (OFS0.WDT0SLCSTP) in option function select register 0. In WDT register start mode, operation or stopping can be selected by setting the sleep mode count stop control bit (WDTCSTPR.SLCSTP) in the WDT count stop control register.

Note 4. In IWDT auto-start mode, operation or stopping can be selected by setting the IWDT sleep mode count stop control bit (OFS0.IWDTSLCSTP) in option function select register 0. In IWDT register start mode, operation or stopping can be selected by setting the sleep mode count stop control bit (IWDTCSTPR.SLCSTP) in the IWDT count stop control register.

Table 11.3 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode (1/2)

Item	Software Standby Mode (SSTBY)		Deep Software Standby Mode (DSTBY)		
	SSTBY 1	**SSTBY 2**	**DSTBY 1**	**DSTBY 2**	**DSTBY 3**
Transition condition	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 4h and the CLPCR.DEEPSLEEP bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to software standby mode 1 is complete after execution of the WAIT instruction).	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 5h and the CLPCR.DEEPSLEEP bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to software standby mode 2 is complete after execution of the WAIT instruction).	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 8h, the CLPCR.DEEPSLEEP bit is 1, and the DCCR.DSTBYCON bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to deep software standby mode 1 is complete after execution of the WAIT instruction).	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are 9h, the CLPCR.DEEPSLEEP bit is 1, and the DCCR.DSTBYCON bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to deep software standby mode 2 is complete after execution of the WAIT instruction).	When the following conditions are satisfied while the LPSCR.LPMD[3:0] bits are Ah, the CLPCR.DEEPSLEEP bit is 1, and the DCCR.DSTBYCON bit is 1: • The WAIT instruction is executed • The CPU is unable to accept a valid interrupt request*1 (including a request before a transition to deep software standby mode 3 is complete after execution of the WAIT instruction).
Method for release from each mode	Interrupts shown in Table 11.4		Interrupts shown in Table 11.4		
	Reset		Reset		
State after release by an interrupt	Program execution state		Reset state		
State after release by a reset	Reset state		Reset state		
Main clock oscillator	Operation possible*2		Stopped		
Sub-clock oscillator	Operation possible		Operation possible		
High-speed on-chip oscillator	Operation possible*3		Stopped		
Middle-speed on-chip oscillator	Operation possible*4		Stopped		
Low-speed on-chip oscillator	Operation possible*5, *6		Operation possible *5, *6	Operation possible*6	
PLL1	Stopped		Stopped		
PLL2	Stopped		Stopped		
Main clock oscillator stop detection function	Operation possible*7		Stopped		
Sub-clock oscillator stop detection function	Operation possible		Operation possible		
Clock/buzzer output function	Operation possible*8		Stopped (undefined)		
External bus (EBCLK)	Stopped (retained)		Stopped (retained)		
CPU	Stopped (retained)		Stopped (undefined)		
Local RAM (SRAM)	Stopped (retained/undefined)*9		Stopped (undefined)		
User SRAM	Stopped (retained/undefined)*10		Stopped (undefined)		
Backup register	Stopped (retained)		Stopped (retained)		
MRAM	Stopped (retained)		Stopped (retained)		
Independent watchdog timer (IWDT)	Operation possible*11		Operation possible *11	Stopped (undefined)	

Table 11.3 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode (2/2)

Item	Software Standby Mode (SSTBY)		Deep Software Standby Mode (DSTBY)		
	SSTBY 1	SSTBY 2	DSTBY 1	DSTBY 2	DSTBY 3
Ethernet module (ESWM)	Stopped (retained)*12	Stopped (undefined)	Stopped (undefined)		
USB 2.0 full-speed host/function module (USB)	Stopped (retained)*13		Stopped (retained) *13	Stopped (undefined)	
USB 2.0 high-speed host/function module (USBHS)	Stopped (retained)*14		Stopped (retained) *14	Stopped (undefined)	
Realtime clock (RTC)	Operation possible		Operation possible		
CAN FD module with gateway functionality (CANFDG)	Stopped (retained)*15	Stopped (undefined)	Stopped (undefined)		
CAN FD module (CANFD)	Stopped (retained)*16	Stopped (undefined)	Stopped (undefined)		
SD/MMC host interface (SDHI)	Stopped (retained)*17	Stopped (undefined)	Stopped (undefined)		
High-speed I^2C bus interface (RIICHS)	Operation possible*18		Stopped (undefined)		
I3C bus interface (RI3C)	Operation possible*19		Stopped (undefined)		
12-bit D/A Converter (DAC12R)	Operation possible*20		Operation possible*20		
Temperature sensor (TEMPS)	Operation possible*21		Operation possible*21		
Pulse-density modulation interface (PDMIF)	Operation possible*22		Stopped (undefined)		
IRQn (n = 0 to 31) pin interrupt	Operation possible		Stopped (undefined)		
NMI, IRQn-DS (n = 0 to 31) Pin interrupt	Operation possible		Operation possible		
Programmable voltage detection circuit (PVD)	Operation possible		Operation possible*23	Operation possible*23	Stopped (undefined)*23
VCC voltage drop detection circuit dedicated to VBATT	Operation possible		Operation possible*24	Operation possible*24	Stopped (undefined)
Power-on reset circuit	Enabled		Enabled	Enabled	Enabled*25
CVM reset circuit	Operation possible		Stopped	Stopped	Stopped
Other peripheral modules	Stopped (retained)		Stopped (undefined)		
I/O ports	Retained*26		Retained*26		

Note: “Operation possible” means that operation or stopping can be controlled by setting the control register.
“Stopped (retained)” means that operation is suspended and after recovery from a low power consumption mode, the internal state before the transition to that mode is retained.
“Stopped (undefined)” means that power to the internal circuitry is turned off and after recovery from a low power consumption mode, the internal state is initialized.

Note 1. “Valid interrupt request” refers to any interrupt or exception that is not masked by the priority level of a current exception or the priority level specified by the IPRr register.

Note 2. If MOSCSCR.MOSCSOKP is set to 1 while the MOSC is oscillating, the MOSC continues to oscillate during software standby mode.

Note 3. If MOSCSCR.MOSCSOKP is set to 1 while the HOCO is oscillating, the HOCO continues to oscillate during software standby mode.

Note 4. Operation or stopping can be selected by setting the MOCOSCR.MOCOSOKP bit.

Note 5. This oscillator can be made to operate or stop by the setting of the LOCOCR.LCSTP bit. However, the oscillator continues to operate regardless of the LCSTP setting in either of the following cases.

- When the IWDT is set for counting operation in IWDT auto-start mode and the IWDT sleep mode count stop control bit in option function select register 0 (OFS0.IWDTSLCSTP) is set to 0
- When the IWDT is set for counting operation in IWDT register start mode and the sleep mode count stop control bit in the IWDT count stop control register (IWDTCSTPR.SLCSTP) is set to 0

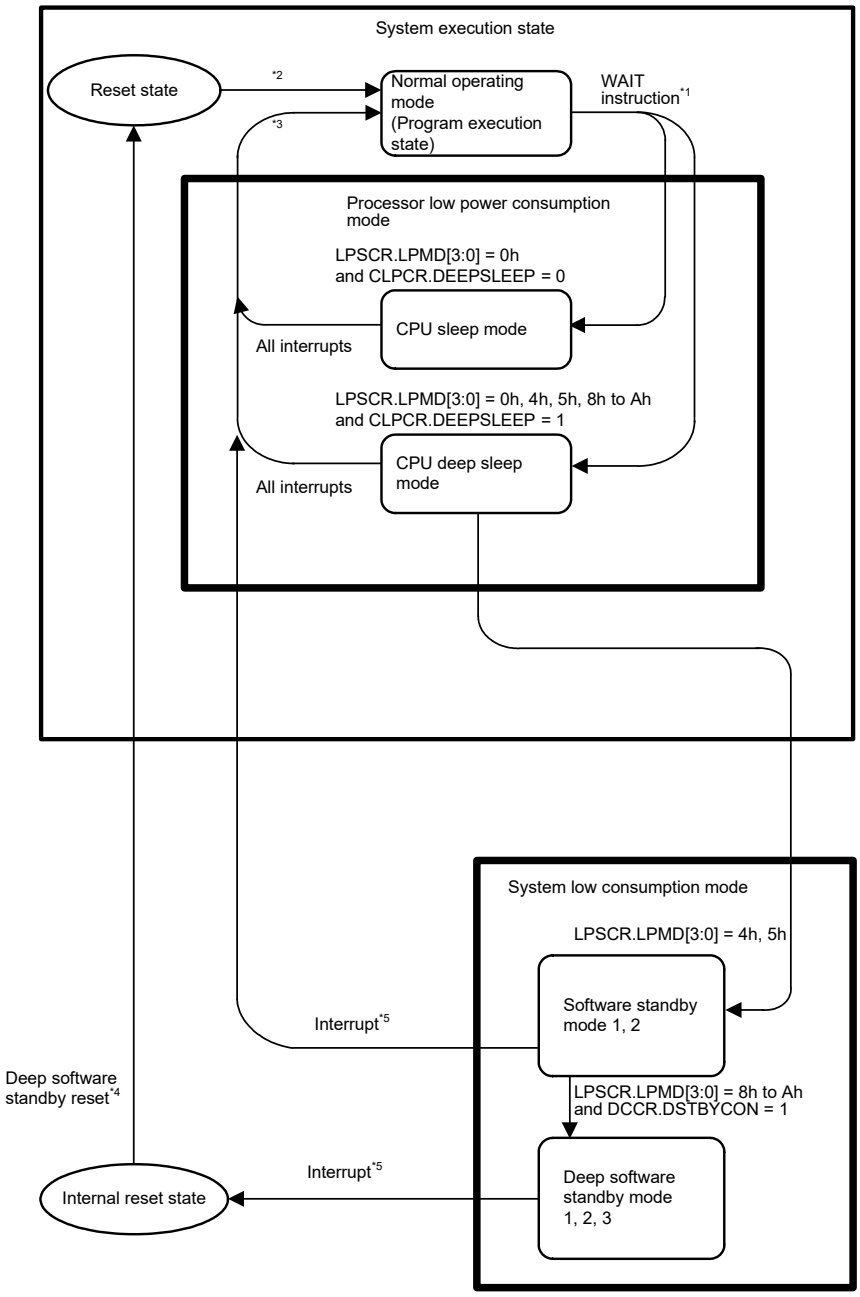
Note 6. This oscillator operates when the sub-clock oscillation stop detection function is enabled (SOSTDCR.SOSTDE = 1).
Note 7. Operation or stopping can be selected by setting the MOSCSCR.MOSCSOKP bit.
Note 8. Operation is possible only when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to 010b (LOCO) or 100b (SOSC). Stopped when the bits are set to a value other than 010b (LOCO) and 100b (SOSC).
Note 9. A retained or undefined state can be selected by setting the PDRAMSCR2.RKEEP0 bit.
Note 10. A retained or undefined state can be selected by setting the PDRAMSCR0.RKEEPn (n = 0 to 11) bit.
Note 11. In IWDT auto-start mode, operation or stopping can be selected by setting the IWDT sleep mode count stop control bit (OFS0.IWDTSLCSTP) in option function select register 0. In IWDT register start mode, operation or stopping can be selected by setting the sleep mode count stop control bit (IWDTCSTPR.SLCSTP) in the IWDT count stop control register.
Note 12. The state of RAM for ESWM is undefined.
Note 13. Only the resume detection is available. The contents of the FIFO buffer are undefined.
Note 14. Only the resume detection is available. The contents of the FIFO buffer are undefined.
Note 15. The contents of the acceptance filter list RAM and message buffer RAM are undefined.
Note 16. The contents of the message buffer RAM are undefined.
Note 17. The state of the SD buffer is undefined.
Note 18. Only the wakeup function is available.
Note 19. Only the wakeup function is available.
Note 20. The D/A output is retained if D/A conversion is enabled.
Note 21. Only the temperature monitoring reset is available.
Note 22. Only the audio detection interrupt is available.
Note 23. When using the PVD, do not set the LPSCR.LPMD[3:0] bits to Ah (deep software standby mode 3).
Note 24. When the MCU transitions to deep software standby mode 1 or 2 with OFS1.PVDAS = 0 and OFS1.DSBYLPSEL = 0, the battery power switch is controlled by voltage monitor 0, which consumes less power than the VCC voltage drop detection circuit dedicated to VBATT. When the MCU transitions to deep software standby mode 1 or 2 with OFS1.DSBYLPSEL = 1, the battery power switch is controlled by the VCC voltage drop detection circuit dedicated to VBATT. For details, see section 12, Battery Backup Function (VBATTBa) and section 67, Electrical Characteristics.
Note 25. In deep software standby mode 3, the PVD circuit is stopped.
Note 26. Whether the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, ALE, BACK#, CKE, SDCS#, RAS#, CAS#, WE#, DQM0 to DQM3) are placed in the high-impedance or the output state is retained can be specified by the SBYCR.OPE bit.

Table 11.4 Interrupts That Can Cause Release from Software Standby Mode and Deep Software Standby Mode

Source of interrupt request	Interrupt Source	Name	Software Standby Mode		Deep Software Standby Mode		
			SSTBY1	SSTBY2	DSTBY1	DSTBY2	DSTBY3
ICU	NMI pin	NMI	Possible	Possible	Possible	Possible	Possible
ICU	External pin	IRQn (n = 0 to 31)	Possible	Possible	Impossible	Impossible	Impossible
		IRQn-DS (n = 0 to 31)	Possible	Possible	Possible	Possible	Possible
PVD	Voltage monitoring m (m = 1, 2)	PVDm (m = 1, 2)	Possible	Possible	Possible	Possible	Impossible
IWDT	IWDT underflow	IWUNI	Possible	Possible	Possible	Impossible	Impossible
USB0	USB0 suspend/resume	USBR*1	Possible	Possible	Possible*2	Impossible	Impossible
USBHS	USBHS suspend/resume	USBHSR*3	Possible	Possible	Possible*2	Impossible	Impossible
RTC	RTC alarm	ALM	Possible	Possible	Possible	Possible	Possible
	RTC periodic	PRD	Possible	Possible	Possible	Possible	Possible
RIICHS1	Wake-up	WUI	Possible	Possible	Impossible	Impossible	Impossible
RIICHS2	Wake-up	WUI	Possible	Possible	Impossible	Impossible	Impossible
RI3C0	Wake-up	WUI	Possible	Possible	Impossible	Impossible	Impossible
PDMIF	Audio detection	SDET	Possible	Possible	Impossible	Impossible	Impossible
Sub-clock oscillator stop detection function (SOST)	Sub-clock oscillator stop detection	SOSTDI	Possible	Possible	Possible	Possible	Possible
Battery backup function (VBATT)	Tamper detection	TAMPDI	Possible	Possible	Possible	Possible	Possible

Note 1. For details on USBR, see Table 40.14, USB Interrupts and section 40.3.1.5, Release from Deep Software Standby Mode 1 Due to USB Suspend/Resume Interrupts.
Note 2. To comply with USB 2.0 when using resume interrupts, it is necessary to consider the oscillator stabilization time.
Note 3. For details on USBHSR, see Table 41.17, USBHS Interrupt Sources and section 41.3.18, Release from Deep Software Standby Mode 1 in Response to the USBHS Resume Interrupt.

Figure 11.1 shows the transitions between low power consumption modes.
For details, see section 11.6, Low Power Consumption Modes.



Note 1. If an interrupt that causes release of the MCU from CPU sleep mode or CPU deep sleep mode is accepted during the transition to CPU sleep mode or CPU deep sleep mode, the MCU will return to normal operating mode instead of entering CPU sleep mode or CPU deep sleep mode.
If an interrupt that causes release of the MCU from CPU sleep mode or CPU deep sleep mode is received during the transition to software standby mode, the MCU will return to normal operating mode instead of entering software standby mode.
Note 2. The MOCO is the source of the operating clock following a transition from the reset state to normal operating mode.
Note 3. When a transition from CPU sleep mode, CPU deep sleep mode or software standby mode to normal mode occurs because of an interrupt, the clock source is the same as before entering each low power consumption mode.
Note 4. If, after transitioning to deep software standby mode, an interrupt request that acts as a releasing source is generated, an internal reset (deep software standby reset) occurs for a certain period of time after the mode transition is complete. The system then transitions to normal mode upon release from the internal reset state, and executes the reset exception processing with the MOCO clock as the operating clock source.
Note 5. For interrupts that act as a releasing source for each low power consumption mode, see Table 11.4.

Figure 11.1 Transitions between Low Power Consumption Modes

11.2 Register Descriptions

11.2.1 Low Power Consumption Mode Security Attribution Register (LPMSAR)

Address(es): SYSC.LPMSAR 8701 93C8h, SYSC_NS.LPMSAR 0701 93C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	NONSEC21	—	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NONSEC8	—	—	—	NONSEC4	—	NONSEC2	NONSEC1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	NONSEC1	Security Attribution 1	Target register: LPSCR 0: Secure 1: Non-secure	R/W
b2	NONSEC2	Security Attribution 2	Target register: SBYCR, SSCR1, SVSCR 0: Secure 1: Non-secure	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	NONSEC4	Security Attribution 4	Target register: VSCR 0: Secure 1: Non-secure	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	NONSEC8	Security Attribution 8	Target register: DPSBYCR, DCCR 0: Secure 1: Non-secure	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	NONSEC16	Security Attribution 16	Target register: LDOECR 0: Secure 1: Non-secure	R/W
b17	NONSEC17	Security Attribution 17	Target register: HOCOLDOCR 0: Secure 1: Non-secure	R/W
b18	NONSEC18	Security Attribution 18	Target register: PLL1LDOCR 0: Secure 1: Non-secure	R/W
b19	NONSEC19	Security Attribution 19	Target register: PLL2LDOCR 0: Secure 1: Non-secure	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b21	NONSEC21	Security Attribution 21	Target register: LVOCR, MWMCR 0: Secure 1: Non-secure	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The LPMSAR register is used to set the security attribute of each low power consumption mode register.

NONSEC1 Bit (Security Attribution 1)

This bit sets the security attribute of the LPSCR register.

NONSEC2 Bit (Security Attribution 2)

This bit sets the security attribute of the SBYCR, SSCR1, and SVSCR registers.

NONSEC4 Bit (Security Attribution 4)

This bit sets the security attribute of the VSCR register.

NONSEC8 Bit (Security Attribution 8)

This bit sets the security attribute of the DPSBYCR and DCCR registers.

NONSEC16 Bit (Security Attribution 16)

This bit sets the security attribute of the LDOECR register.

NONSEC17 Bit (Security Attribution 17)

This bit sets the security attribute of the HOCOLDOCR register.

NONSEC18 Bit (Security Attribution 18)

This bit sets the security attribute of the PLL1LDOCR register.

NONSEC19 Bit (Security Attribution 19)

This bit sets the security attribute of the PLL2LDOCR register.

NONSEC21 Bit (Security Attribution 21)

This bit sets the security attribute of the LVOCR and MWMCR registers.
For details about the LVOCR register, see section 22.3.17, Low-Voltage Operating Characteristics Control Register (LVOCR). For details about the MWMCR register, see section 65.5.52, MRAM-OTP Write Mode Control Register (MWMCR).

11.2.2 Power Gating Security Attribution Register (PGCSAR)

Address(es): SYSC.PGCSAR 8701 93D8h, SYSC_NS.PGCSAR 0701 93D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	NONSEC6	NONSEC5	—	NONSEC3	—	—	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NONSEC0	Security Attribution 0	Target register: PGSCR 0: Secure 1: Non-secure	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	NONSEC3	Security Attribution 3	Target register: PSSTCR2 0: Secure 1: Non-secure	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	NONSEC5	Security Attribution 5	Target register: PSSTCR4 0: Secure 1: Non-secure	R/W
b6	NONSEC6	Security Attribution 6	Target register: PSSTCR5, PDCTRESWM 0: Secure 1: Non-secure	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PGCSAR register is used to set the security attribute of each power gating register.

NONSEC0 Bit (Security Attribution 0)

This bit sets the security attribute of the PGSCR register.

NONSEC3 Bit (Security Attribution 3)

This bit sets the security attribute of the PSSTCR2 register.

NONSEC5 Bit (Security Attribution 5)

This bit sets the security attribute of the PSSTCR4 register.

NONSEC6 Bit (Security Attribution 6)

This bit sets the security attribute of the PSSTCR5 and PDCTRESWM registers.

11.2.3 Deep Software Standby Interrupt Source Security Attribution Register (DPFSAR)

Address(es): SYSC.DPFSAR 8701 93E0h, SYSC_NS.DPFSAR 0701 93E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DPFSA31	DPFSA30	DPFSA29	—	—	—	DPFSA25	DPFSA24	—	—	—	DPFSA20	DPFSA19	DPFSA18	DPFSA17	DPFSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPFSA15	DPFSA14	DPFSA13	DPFSA12	DPFSA11	DPFSA10	DPFSA9	DPFSA8	DPFSA7	DPFSA6	DPFSA5	DPFSA4	DPFSA3	DPFSA2	DPFSA1	DPFSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPFSA0	Deep Software Standby Interrupt Source Security Attribution0	Target register: DPSIER0.DIRQ0E, DPSIFR0.DIRQ0F, DPSIEGR0.DIRQ0EG Target source: IRQ0-DS pin 0: Secure 1: Non-secure	R/W
b1	DPFSA1	Deep Software Standby Interrupt Source Security Attribution 1	Target register: DPSIER0.DIRQ1E, DPSIFR0.DIRQ1F, DPSIEGR0.DIRQ1EG Target source: IRQ1-DS pin 0: Secure 1: Non-secure	R/W
b2	DPFSA2	Deep Software Standby Interrupt Source Security Attribution 2	Target register: DPSIER0.DIRQ2E, DPSIFR0.DIRQ2F, DPSIEGR0.DIRQ2EG Target source: IRQ2-DS pin 0: Secure 1: Non-secure	R/W
b3	DPFSA3	Deep Software Standby Interrupt Source Security Attribution 3	Target register: DPSIER0.DIRQ3E, DPSIFR0.DIRQ3F, DPSIEGR0.DIRQ3EG Target source: IRQ3-DS pin 0: Secure 1: Non-secure	R/W
b4	DPFSA4	Deep Software Standby Interrupt Source Security Attribution 4	Target register: DPSIER0.DIRQ4E, DPSIFR0.DIRQ4F, DPSIEGR0.DIRQ4EG Target source: IRQ4-DS pin 0: Secure 1: Non-secure	R/W
b5	DPFSA5	Deep Software Standby Interrupt Source Security Attribution 5	Target register: DPSIER0.DIRQ5E, DPSIFR0.DIRQ5F, DPSIEGR0.DIRQ5EG Target source: IRQ5-DS pin 0: Secure 1: Non-secure	R/W
b6	DPFSA6	Deep Software Standby Interrupt Source Security Attribution 6	Target register: DPSIER0.DIRQ6E, DPSIFR0.DIRQ6F, DPSIEGR0.DIRQ6EG Target source: IRQ6-DS pin 0: Secure 1: Non-secure	R/W
b7	DPFSA7	Deep Software Standby Interrupt Source Security Attribution 7	Target register: DPSIER0.DIRQ7E, DPSIFR0.DIRQ7F, DPSIEGR0.DIRQ7EG Target source: IRQ7-DS pin 0: Secure 1: Non-secure	R/W

Bit	Symbol	Bit Name	Description	R/W
b8	DPFSA8	Deep Software Standby Interrupt Source Security Attribution8	Target register: DPSIER1.DIRQ8E, DPSIFR1.DIRQ8F, DPSIEGR1.DIRQ8EG Target source: IRQ8-DS pin 0: Secure 1: Non-secure	R/W
b9	DPFSA9	Deep Software Standby Interrupt Source Security Attribution 9	Target register: DPSIER1.DIRQ9E, DPSIFR1.DIRQ9F, DPSIEGR1.DIRQ9EG Target source: IRQ9-DS pin 0: Secure 1: Non-secure	R/W
b10	DPFSA10	Deep Software Standby Interrupt Source Security Attribution 10	Target register: DPSIER1.DIRQ10E, DPSIFR1.DIRQ10F, DPSIEGR1.DIRQ10EG Target source: IRQ10-DS pin 0: Secure 1: Non-secure	R/W
b11	DPFSA11	Deep Software Standby Interrupt Source Security Attribution 11	Target register: DPSIER1.DIRQ11E, DPSIFR1.DIRQ11F, DPSIEGR1.DIRQ11EG Target source: IRQ11-DS pin 0: Secure 1: Non-secure	R/W
b12	DPFSA12	Deep Software Standby Interrupt Source Security Attribution 12	Target register: DPSIER1.DIRQ12E, DPSIFR1.DIRQ12F, DPSIEGR1.DIRQ12EG Target source: IRQ12-DS pin 0: Secure 1: Non-secure	R/W
b13	DPFSA13	Deep Software Standby Interrupt Source Security Attribution 13	Target register: DPSIER1.DIRQ13E, DPSIFR1.DIRQ13F, DPSIEGR1.DIRQ13EG Target source: IRQ13-DS pin 0: Secure 1: Non-secure	R/W
b14	DPFSA14	Deep Software Standby Interrupt Source Security Attribution 14	Target register: DPSIER1.DIRQ14E, DPSIFR1.DIRQ14F, DPSIEGR1.DIRQ14EG Target source: IRQ14-DS pin 0: Secure 1: Non-secure	R/W
b15	DPFSA15	Deep Software Standby Interrupt Source Security Attribution 15	Target register: DPSIER1.DIRQ15E, DPSIFR1.DIRQ15F, DPSIEGR1.DIRQ15EG Target source: IRQ15-DS pin 0: Secure 1: Non-secure	R/W
b16	DPFSA16	Deep Software Standby Interrupt Source Security Attribution 16	Target register: DPSIER2.DPVD1IE, DPSIFR2.DPVD1IF, DPSIEGR2.DPVD1EG Target source: PVD1 0: Secure 1: Non-secure	R/W
b17	DPFSA17	Deep Software Standby Interrupt Source Security Attribution 17	Target register: DPSIER2.DPVD2IE, DPSIFR2.DPVD2IF, DPSIEGR2.DPVD2EG Target source: PVD2 0: Secure 1: Non-secure	R/W
b18	DPFSA18	Deep Software Standby Interrupt Source Security Attribution 18	Target register: DPSIER2.DRTCIIE, DPSIFR2.DRTCIIF Target source: RTC interval 0: Secure 1: Non-secure	R/W
b19	DPFSA19	Deep Software Standby Interrupt Source Security Attribution 19	Target register: DPSIER2.DRTCAIE, DPSIFR2.DRTCAIF Target source: RTC alarm 0: Secure 1: Non-secure	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	DPFSA20	Deep Software Standby Interrupt Source Security Attribution 20	Target register: DPSIER2.DNMIE, DPSIFR2.DNMIF, DPSIEGR2.DNMIEG Target source: NMI pin 0: Secure 1: Non-secure	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DPFSA24	Deep Software Standby Interrupt Source Security Attribution 24	Target register: DPSIER3.DUSBFSIE, DPSIFR3.DUSBFSIF Target source: USB0 suspend/resume 0: Secure 1: Non-secure	R/W
b25	DPFSA25	Deep Software Standby Interrupt Source Security Attribution 25	Target register: DPSIER3.DUSBHSIE, DPSIFR3.DUSBHSIF Target source: USBHS suspend/resume 0: Secure 1: Non-secure	R/W
b28 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	DPFSA29	Deep Software Standby Interrupt Source Security Attribution 29	Target register: DPSIER3.DIWDTIE, DPSIFR3.DIWDTIF Target source: IWDT underflow 0: Secure 1: Non-secure	R/W
b30	DPFSA30	Deep Software Standby Interrupt Source Security Attribution 30	Target register: DPSIER3.DSOSTDIE, DPSIFR3.DSOSTDIF Target source: Sub-clock oscillator stop detection 0: Secure 1: Non-secure	R/W
b31	DPFSA31	Deep Software Standby Interrupt Source Security Attribution 31	Target register: DPSIER3.DVBATTADIE, DPSIFR3.DVBATTADIF Target source: Tamper detection 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The DPFSAR register is used to set the security attribute of each deep software standby interrupt source control register.

DPFSAn Bit (Deep Software Standby Interrupt Source Security Attribution n) (n = 0 to 7)

This bit sets the security attribute of the DPSIER0.DIRQnE, DPSIFR0.DIRQnF, and DPSIEGR0.DIRQnEG bits.
The target source is the IRQn-DS pin.

DPFSAn Bit (Deep Software Standby Interrupt Source Security Attribution n) (n = 8 to 15)

This bit sets the security attribute of the DPSIER1.DIRQnE, DPSIFR1.DIRQnF, and DPSIEGR1.DIRQnEG bits.
The target source is the IRQn-DS pin.

DPFSA16 Bit (Deep Software Standby Interrupt Source Security Attribution 16)

This bit sets the security attribute of the DPSIER2.DPVD1IE, DPSIFR2.DPVD1IF, and DPSIEGR2.DPVD1EG bits.
The target source is PVD1.

DPFSA17 Bit (Deep Software Standby Interrupt Source Security Attribution 17)

This bit sets the security attribute of the DPSIER2.DPVD2IE, DPSIFR2.DPVD2IF, and DPSIEGR2.DPVD2EG bits.
The target source is PVD2.

DPFSA18 Bit (Deep Software Standby Interrupt Source Security Attribution 18)

This bit sets the security attribute of the DPSIER2.DRTCIIE and DPSIFR2.DRTCIIF bits.
The target source is the RTC interval.

DPFSA19 Bit (Deep Software Standby Interrupt Source Security Attribution 19)

This bit sets the security attribute of the DPSIER2.DRTCAIE and DPSIFR2.DRTCAIF bits.
The target source is the RTC alarm.

DPFSA20 Bit (Deep Software Standby Interrupt Source Security Attribution 20)

This bit sets the security attribute of the DPSIER2.DNMIE, DPSIFR2.DNMIF, and DPSIEGR2.DNMIEG bits.
The target source is the NMI pin.

DPFSA24 Bit (Deep Software Standby Interrupt Source Security Attribution 24)

This bit sets the security attribute of the DPSIER3.DUSBFSIE and DPSIFR3.DUSBFSIF bits.
The target source is USB0 suspend/resume.

DPFSA25 Bit (Deep Software Standby Interrupt Source Security Attribution 25)

This bit sets the security attribute of the DPSIER3.DUSBHSIE and DPSIFR3.DUSBHSIF bits.
The target source is USBHS suspend/resume.

DPFSA29 Bit (Deep Software Standby Interrupt Source Security Attribution 29)

This bit sets the security attribute of the DPSIER3.DIWDTIE and DPSIFR3.DIWDTIF bits.
The target source is IWDT underflow.

DPFSA30 Bit (Deep Software Standby Interrupt Source Security Attribution 30)

This bit sets the security attribute of the DPSIER3.DSOSTDIE and DPSIFR3.DSOSTDIF bits.
The target source is oscillation stop detection in the sub-clock oscillator.

DPFSA31 Bit (Deep Software Standby Interrupt Source Security Attribution 31)

This bit sets the security attribute of the DPSIER3.DVBATTADIE and DPSIFR3.DVBATTADIF bits.
The target source is tamper detection.

11.2.4 Deep Software Standby Interrupt Source Security Attribution Register 1 (DPFSAR1)

Address(es): SYSC.DPFSAR1 8701 93E8h, SYSC_NS.DPFSAR1 0701 93E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPFSA15	DPFSA14	DPFSA13	DPFSA12	DPFSA11	DPFSA10	DPFSA9	DPFSA8	DPFSA7	DPFSA6	DPFSA5	DPFSA4	DPFSA3	DPFSA2	DPFSA1	DPFSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPFSA0	Deep Software Standby Interrupt Source Security Attribution 0	Target register: DPSIER4.DIRQ16E, DPSIFR4.DIRQ16F, DPSIEGR3.DIRQ16EG Target source: IRQ16-DS pin 0: Secure 1: Non-secure	R/W
b1	DPFSA1	Deep Software Standby Interrupt Source Security Attribution 1	Target register: DPSIER4.DIRQ17E, DPSIFR4.DIRQ17F, DPSIEGR3.DIRQ17EG Target source: IRQ17-DS pin 0: Secure 1: Non-secure	R/W
b2	DPFSA2	Deep Software Standby Interrupt Source Security Attribution 2	Target register: DPSIER4.DIRQ18E, DPSIFR4.DIRQ18F, DPSIEGR3.DIRQ18EG Target source: IRQ18-DS pin 0: Secure 1: Non-secure	R/W
b3	DPFSA3	Deep Software Standby Interrupt Source Security Attribution 3	Target register: DPSIER4.DIRQ19E, DPSIFR4.DIRQ19F, DPSIEGR3.DIRQ19EG Target source: IRQ19-DS pin 0: Secure 1: Non-secure	R/W
b4	DPFSA4	Deep Software Standby Interrupt Source Security Attribution 4	Target register: DPSIER4.DIRQ20E, DPSIFR4.DIRQ20F, DPSIEGR3.DIRQ20EG Target source: IRQ20-DS pin 0: Secure 1: Non-secure	R/W
b5	DPFSA5	Deep Software Standby Interrupt Source Security Attribution 5	Target register: DPSIER4.DIRQ21E, DPSIFR4.DIRQ21F, DPSIEGR3.DIRQ21EG Target source: IRQ21-DS pin 0: Secure 1: Non-secure	R/W
b6	DPFSA6	Deep Software Standby Interrupt Source Security Attribution 6	Target register: DPSIER4.DIRQ22E, DPSIFR4.DIRQ22F, DPSIEGR3.DIRQ22EG Target source: IRQ22-DS pin 0: Secure 1: Non-secure	R/W
b7	DPFSA7	Deep Software Standby Interrupt Source Security Attribution 7	Target register: DPSIER4.DIRQ23E, DPSIFR4.DIRQ23F, DPSIEGR3.DIRQ23EG Target source: IRQ23-DS pin 0: Secure 1: Non-secure	R/W

Bit	Symbol	Bit Name	Description	R/W
b8	DPFSA8	Deep Software Standby Interrupt Source Security Attribution 8	Target register: DPSIER5.DIRQ24E, DPSIFR5.DIRQ24F, DPSIEGR4.DIRQ24EG Target source: IRQ24-DS pin 0: Secure 1: Non-secure	R/W
b9	DPFSA9	Deep Software Standby Interrupt Source Security Attribution 9	Target register: DPSIER5.DIRQ25E, DPSIFR5.DIRQ25F, DPSIEGR4.DIRQ25EG Target source: IRQ25-DS pin 0: Secure 1: Non-secure	R/W
b10	DPFSA10	Deep Software Standby Interrupt Source Security Attribution 10	Target register: DPSIER5.DIRQ26E, DPSIFR5.DIRQ26F, DPSIEGR4.DIRQ26EG Target source: IRQ26-DS pin 0: Secure 1: Non-secure	R/W
b11	DPFSA11	Deep Software Standby Interrupt Source Security Attribution 11	Target register: DPSIER5.DIRQ27E, DPSIFR5.DIRQ27F, DPSIEGR4.DIRQ27EG Target source: IRQ27-DS pin 0: Secure 1: Non-secure	R/W
b12	DPFSA12	Deep Software Standby Interrupt Source Security Attribution 12	Target register: DPSIER5.DIRQ28E, DPSIFR5.DIRQ28F, DPSIEGR4.DIRQ28EG Target source: IRQ28-DS pin 0: Secure 1: Non-secure	R/W
b13	DPFSA13	Deep Software Standby Interrupt Source Security Attribution 13	Target register: DPSIER5.DIRQ29E, DPSIFR5.DIRQ29F, DPSIEGR4.DIRQ29EG Target source: IRQ29-DS pin 0: Secure 1: Non-secure	R/W
b14	DPFSA14	Deep Software Standby Interrupt Source Security Attribution 14	Target register: DPSIER5.DIRQ30E, DPSIFR5.DIRQ30F, DPSIEGR4.DIRQ30EG Target source: IRQ30-DS pin 0: Secure 1: Non-secure	R/W
b15	DPFSA15	Deep Software Standby Interrupt Source Security Attribution 15	Target register: DPSIER5.DIRQ31E, DPSIFR5.DIRQ31F, DPSIEGR4.DIRQ31EG Target source: IRQ31-DS pin 0: Secure 1: Non-secure	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The DPFSAR1 register is used to set the security attribute of each deep software standby interrupt source control register.

DPFSAn Bit (Deep Software Standby Interrupt Source Security Attribution n) (n = 0 to 7)

This bit sets the security attribute of the DPSIER4.DIRQmE, DPSIFR4.DIRQmF, and DPSIEGR3.DIRQmEG bits. The target source is the IRQm-DS pin. (m = 16 to 23)

DPFSAn Bit (Deep Software Standby Interrupt Source Security Attribution n) (n = 8 to 15)

This bit sets the security attribute of the DPSIER5.DIRQmE, DPSIFR5.DIRQmF, and DPSIEGR4.DIRQmEG bits. The target source is the IRQm-DS pin. (m = 24 to 31)

11.2.5 RAM Standby Control Security Attribution Register (RSCSAR)

Address(es): SYSC.RSCSAR 8701 93E4h, SYSC_NS.RSCSAR 0701 93E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	RSCSA24	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RSCSA11	RSCSA10	RSCSA9	RSCSA8	RSCSA7	RSCSA6	RSCSA5	RSCSA4	RSCSA3	RSCSA2	RSCSA1	RSCSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RSCSA0	RAM Standby Control Security Attribution 0	Target register: PDRAMSCR0.RKEEP0 0: Secure 1: Non-secure	R/W
b1	RSCSA1	RAM Standby Control Security Attribution 1	Target register: PDRAMSCR0.RKEEP1 0: Secure 1: Non-secure	R/W
b2	RSCSA2	RAM Standby Control Security Attribution 2	Target register: PDRAMSCR0.RKEEP2 0: Secure 1: Non-secure	R/W
b3	RSCSA3	RAM Standby Control Security Attribution 3	Target register: PDRAMSCR0.RKEEP3 0: Secure 1: Non-secure	R/W
b4	RSCSA4	RAM Standby Control Security Attribution 4	Target register: PDRAMSCR0.RKEEP4 0: Secure 1: Non-secure	R/W
b5	RSCSA5	RAM Standby Control Security Attribution 5	Target register: PDRAMSCR0.RKEEP5 0: Secure 1: Non-secure	R/W
b6	RSCSA6	RAM Standby Control Security Attribution 6	Target register: PDRAMSCR0.RKEEP6 0: Secure 1: Non-secure	R/W
b7	RSCSA7	RAM Standby Control Security Attribution 7	Target register: PDRAMSCR0.RKEEP7 0: Secure 1: Non-secure	R/W
b8	RSCSA8	RAM Standby Control Security Attribution 8	Target register: PDRAMSCR0.RKEEP8 0: Secure 1: Non-secure	R/W
b9	RSCSA9	RAM Standby Control Security Attribution 9	Target register: PDRAMSCR0.RKEEP9 0: Secure 1: Non-secure	R/W
b10	RSCSA10	RAM Standby Control Security Attribution 10	Target register: PDRAMSCR0.RKEEP10 0: Secure 1: Non-secure	R/W
b11	RSCSA11	RAM Standby Control Security Attribution 11	Target register: PDRAMSCR0.RKEEP11 0: Secure 1: Non-secure	R/W
b23 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	RSCSA24	RAM Standby Control Security Attribution 24	Target register: PDRAMSCR2.RKEEP0 0: Secure 1: Non-secure	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The RSCSAR register is used to set the security attribute of each standby control register for the SRAM power domain.

RSCSAn Bit (RAM Standby Control Security Attribution n) (n = 0 to 11)

This bit sets the security attribute of the PDRAMSCR0.RKEEPn bit.

RSCSA24 Bit (RAM Standby Control Security Attribution 24)

This bit sets the security attribute of the PDRAMSCR2.RKEEP0 bit.

11.2.6 Module Stop Control Register A (MSTPCRA)

Address(es): MSTP.MSTPCRA 8780 3000h, MSTP_NS.MSTPCRA 0780 3000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	MSTPA29	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPA1	MSTPA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPA0	SRAM0 Module Stop*1	Target module: SRAM0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b1	MSTPA1	SRAM1 Module Stop*1	Target module: SRAM1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b21 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPA22	DMA Controller and Data Transfer Controller Module Stop*2	Target module: DMAC, DTC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b28 to b23	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b29	MSTPA29	EXDMA Controller Module Stop*3	Target module: EXDMAC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b31 to b30	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. After changing the value of this bit, confirm that the value has actually been updated before executing subsequent instructions.

Note 2. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.

Note 3. When rewriting the MSTPA29 bit from 0 to 1, disable the EXDMAC before setting the MSTPA29 bit.

11.2.7 Module Stop Control Register B (MSTPCRB)

Address(es): MSTP.MSTPCRB 8780 3004h, MSTP_NS.MSTPCRB 0780 3004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	MSTPB26	MSTPB25	MSTPB24	MSTPB23	MSTPB22	MSTPB21	MSTPB20	MSTPB19	MSTPB18	MSTPB17	MSTPB16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MSTPB12	MSTPB11	—	—	MSTPB8	MSTPB7	—	—	MSTPB4	—	—	—	MSTPB0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0	Serial Communication Interface 12 Module Stop*1	Target module: RSCI12 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b3 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	I3C Bus Interface 0 Module Stop*2	Target module: RI3C0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	MSTPB7	High-Speed I2C Bus Interface 2 Module Stop*2	Target module: RIICHS2 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b8	MSTPB8	High-Speed I2C Bus Interface 1 Module Stop*2	Target module: RIICHS1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b10, b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b11	MSTPB11	USB 2.0 Full-Speed Host/Function Module Stop*3	Target module: USB0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b12	MSTPB12	USB 2.0 High-Speed Host/Function Module Stop*3	Target module: USBHS 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b15 to b13	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b16	MSTPB16	Expanded Serial Peripheral Interface 0 and Decryption On The Fly Module 0 Stop*4	Target module: XSPI0 and DOTF0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b17	MSTPB17	Expanded Serial Peripheral Interface 1 and Decryption On The Fly Module 1 Stop*4	Target module: XSPI1 and DOTF1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b18	MSTPB18	Serial Peripheral Interface 1 Module Stop*5	Target module: RSPIA1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b19	MSTPB19	Serial Peripheral Interface 0 Module Stop*5	Target module: RSPIA0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b20	MSTPB20	Serial Communication Interface 11 Module Stop*1	Target module: RSCI11 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b21	MSTPB21	Serial Communication Interface 10 Module Stop*1	Target module: RSCI10 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Bit	Symbol	Bit Name	Description	R/W
b22	MSTPB22	Serial Communication Interface 9 Module Stop*1	Target module: RSCI9 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b23	MSTPB23	Serial Communication Interface 8 Module Stop*1	Target module: RSCI8 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop*1	Target module: RSCI7 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop*1	Target module: RSCI6 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop*1	Target module: RSCI5 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop*1	Target module: RSCI4 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop*1	Target module: RSCI3 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop*1	Target module: RSCI2 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop*1	Target module: RSCI1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop*1	Target module: RSCI0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. The MSTPBi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the SCI clock (SCICLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 0, 20 to 31).

Note 2. The MSTPBi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the I3C clock (I3CCLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 4, 7, 8).

Note 3. The MSTPBi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the USB clock (USBCLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 11, 12).

Note 4. The MSTPBi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the XSPI clock (XSPICLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 16, 17). After changing the value of this bit, confirm that the value has actually been updated before executing subsequent instructions.

Note 5. The MSTPBi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the SPI clock (SPICLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 18, 19).

11.2.8 Module Stop Control Register C (MSTPCRC)

Address(es): MSTP.MSTPCRC 8780 3008h, MSTP_NS.MSTPCRC 0780 3008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPC31	MSTPC30	—	MSTPC28	MSTPC27	—	MSTPC25	MSTPC24	MSTPC23	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MSTPC14	MSTPC13	MSTPC12	—	—	—	MSTPC8	MSTPC7	—	—	—	—	MSTPC2	MSTPC1	MSTPC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop *1	Target module: CAC0/1/2 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b1	MSTPC1	CRC Calculator Module Stop	Target module: CRC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b2	MSTPC2	Parallel Data Capture Unit Module Stop	Target module: PDC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b6 to b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	MSTPC7	Serial Sound Interface 1 Module Stop	Target module: SSIE1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b8	MSTPC8	Serial Sound Interface 0 Module Stop	Target module: SSIE0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b11 to b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b12	MSTPC12	SD/MMC Host Interface Module Stop	Target module: SDHI 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b13	MSTPC13	Data Operation Circuit Module Stop	Target module: DOC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b14	MSTPC14	Event Link Controller Module Stop	Target module: ELC 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b22 to b15	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23	MSTPC23	Serial Peripheral Interface 2 Module Stop*2	Target module: RSPIA2 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b24	MSTPC24	Pulse-Density Modulation Interface Module Stop*3	Target module: PDMIF 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b25	MSTPC25	CAN FD Module with Gateway Functionality Module Stop*4	Target module: CANFDG 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b26	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b27	MSTPC27	CAN FD Module Stop*4	Target module: CANFD 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	MSTPC28	Ethernet-PHY Clock Module Stop*5	Target module: ETHPHYCLK 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b29	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b30	MSTPC30	Ethernet Module Stop*6	Target module: ESWM 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b31	MSTPC31	Renesas Secure IP Module Stop	Target module: RSIP-E50D 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. The MSTPC0 bit must be rewritten while the oscillation of the clock to be controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to this bit, wait for 2 cycles of the slowest clock among the clocks output by the oscillators to elapse after writing the new value, and then execute the WAIT instruction.

Note 2. The MSTPC23 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to the MSTPC23 bit, wait for two cycles of the SPI clock (SPICLK) to elapse after writing the new value, and then execute the WAIT instruction.

Note 3. The MSTPC24 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to the MSTPC24 bit, wait for two cycles of the PDMIF clock (PDMIFCLK) to elapse after writing the new value, and then execute the WAIT instruction.

Note 4. The MSTPCi bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to the MSTPCi bit, wait for two cycles of the CANFD clock (CANFDCLK) to elapse after writing the new value, and then execute the WAIT instruction (i = 25, 27).

Note 5. The MSTPC28 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to the MSTPC28 bit, wait for two cycles of the Ethernet-PHY clock (ETHPHYCLK) to elapse after writing the new value, and then execute the WAIT instruction.

Note 6. The MSTPC30 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. To place the MCU in software standby mode after writing a new value to the MSTPC30 bit, wait for two cycles of the ESWM clock (ESWCLK) and two cycles of the ESWM-PHY clock (ESWPHYCLK) to elapse after writing the new value and then execute the WAIT instruction.

11.2.9 Module Stop Control Register D (MSTPCRD)

Address(es): MSTP.MSTPCRD 8780 300Ch, MSTP_NS.MSTPCRD 0780 300Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	MSTPD22	MSTPD21	MSTPD20	MSTPD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MSTPD14	MSTPD13	MSTPD12	MSTPD11	—	—	—	—	MSTPD6	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	MSTPD6	High Resolution PWM Waveform Generation Circuit Module Stop	Target module: HRPWM0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b10 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b11	MSTPD11	Port Output Enable for GPTW Group D Module Stop	Target module: POEG group D 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b12	MSTPD12	Port Output Enable for GPTW Group C Module Stop	Target module: POEG group C 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b13	MSTPD13	Port Output Enable for GPTW Group B Module Stop	Target module: POEG group B 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b14	MSTPD14	Port Output Enable for GPTW Group A Module Stop	Target module: POEG group A 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b18 to b15	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPD19	12-bit D/A Converter 1 Module Stop	Target module: DAC12R1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b20	MSTPD20	12-bit D/A Converter 0 Module Stop	Target module: DAC12R0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b21	MSTPD21	16-bit A/D Converter Module Stop*1	Target module: ADC16H 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b22	MSTPD22	Temperature Sensor Module Stop	Target module: TEMPS 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b31 to b23	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. The MSTPD21 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPD21 bit, wait for two cycles of the ADC clock (ADCCLK) to elapse after writing the new value, and then execute the WAIT instruction.

11.2.10 Module Stop Control Register E (MSTPCRE)

Address(es): MSTP.MSTPCRE 8780 3010h, MSTP_NS.MSTPCRE 0780 3010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPE31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPE31	General PWM Timer Module Stop*1	Target module: GPTW 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. The MSTPE31 bit must be rewritten while the oscillation of the clock controlled by this bit is stabilized. For entering software standby mode after writing a new value to the MSTPE31 bit, wait for two cycles of the GPT clock (GPTCLK) to elapse after writing the new value, and then execute the WAIT instruction.

11.2.11 Module Stop Control Register F (MSTPCRF)

Address(es): MSTP.MSTPCRF 8780 3014h, MSTP_NS.MSTPCRF 0780 3014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPF31	MSTPF30	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPF16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPF15	MSTPF14	MSTPF13	MSTPF12	—	—	—	—	MSTPF7	MSTPF6	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	MSTPF6	8-Bit Timer 3, 2 (Unit 1) Module Stop	Target module: TMR2, TMR3 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b7	MSTPF7	8-Bit Timer 1, 0 (Unit 0) Module Stop	Target module: TMR0, TMR1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b11 to b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b12	MSTPF12	Compare Match Timer W (Unit 1) Module Stop	Target module: CMTW1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b13	MSTPF13	Compare Match Timer W (Unit 0) Module Stop	Target module: CMTW0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b14	MSTPF14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b15	MSTPF15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b16	MSTPF16	Software Configurable 32-Bit Programmable I/O Port Module Stop	Target module: PIO 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b29 to b17	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	MSTPF30	Phase Counting Interface 1 Module Stop	Target module: PCIF1 0: Release from the module-stop state 1: Transition to the module-stop state	R/W
b31	MSTPF31	Phase Counting Interface 0 Module Stop	Target module: PCIF0 0: Release from the module-stop state 1: Transition to the module-stop state	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

11.2.12 Power Gating Shift Control Register (PGSCR)

Address(es): SYSC.PGSCR 8701 9104h, SYSC_NS.PGSCR 0701 9104h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PGS
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Power Gating Shift Timing	0: Power gating control for all domains is started simultaneously. 1: Power gating control for the individual domains is started sequentially.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

PGS Bit (Power Gating Shift Timing)

The PGS bit is used to specify the timing to start power gating for the domains. Set this bit to 0 before placing the MCU in software standby mode or deep software standby mode.

11.2.13 Power Switch Control Start Time Control Register n (PSSTCRn) (n = 2, 4, 5)

Address(es): SYSC.PSSTCR2 8701 9214h, SYSC.PSSTCR4 8701 9218h, SYSC.PSSTCR5 8701 921Ah,
SYSC_NS.PSSTCR2 0701 9214h, SYSC_NS.PSSTCR4 0701 9218h, SYSC_NS.PSSTCR5 0701 921Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PSST[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	PSST[12:0]	Power Switch Control Start Time	Set the time to start power switch control while PGSCR.PGS = 0. Set these bits to 00B0h. Other settings are prohibited.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

11.2.14 ESWM Power Domain Control Register (PDCTRESWM)

Address(es): SYSC.PDCTRESWM 8701 9118h, SYSC_NS.PDCTRESWM 0701 9118h

	b7	b6	b5	b4	b3	b2	b1	b0
	PDPGSF	PDCSF	—	—	—	—	—	PDDE
Value after reset:	1	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PDDE	Power Control Enable	0: The target domain is turned on 1: The target domain is turned off	R/W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PDCSF	Power Transition State Flag	0: Transition is completed. 1: Transition is in progress.	R
b7	PDPGSF	Power Status Flag	0: The target power domain is turned on and not gated. 1: The target power domain is turned off and gated.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The PDCTRESWM register is used to control the power to the ESWM power domain.
For constraints related to this register, see not only this section but also section 11.5.1, Power Gating Control Using the Power Domain Control Register.

PDDE Bit (Power Control Enable)

The PDDE bit controls the power to the target power domain. Before changing the PDDE bit from 0 to 1, confirm that both the PDCSF and PDPGSF bits are 0.
Also, before changing the PDDE bit from 1 to 0, confirm that the PDCSF bit is 0 and the PDPGSF bit is 1.

PDCSF Flag (Power Transition State Flag)

The PDCSF flag indicates whether a change to the power state of the target power domain is in progress.

PDPGSF Flag (Power Status Flag)

The PDPGSF flag indicates whether the power to the target power domain is on or off.

11.2.15 SRAM Power Domain Standby Control Register 0 (PDRAMSCR0)

Address(es): SYSC.PDRAMSCR0 8701 9140h, SYSC_NS.PDRAMSCR0 0701 9140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RKEEP11	RKEEP10	RKEEP9	RKEEP8	RKEEP7	RKEEP6	RKEEP5	RKEEP4	RKEEP3	RKEEP2	RKEEP1	RKEEP0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RKEEP0	RAM Retention	0: When entering the software standby mode, the contents of the target RAM are not retained 1: When entering the software standby mode, the contents of the target RAM are retained	R/W
b1	RKEEP1			R/W
b2	RKEEP2			R/W
b3	RKEEP3			R/W
b4	RKEEP4			R/W
b5	RKEEP5			R/W
b6	RKEEP6			R/W
b7	RKEEP7			R/W
b8	RKEEP8			R/W
b9	RKEEP9			R/W
b10	RKEEP10			R/W
b11	RKEEP11			R/W
b14 to b12	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The PDRAMSCR0 register is used to control the target memory in software standby mode.

RKEEPn Bit (RAM Retention) (n = 0 to 11)

The RKEEPn bit controls whether to retain the contents of the target memory in software standby mode.
Table 11.5 shows the relationship between each bit and the target memory.

Table 11.5 Relationship Between Each Bit and the Target Memory (1/2)

Bit in the PDRAMSCR0 Register	Target Memory	
RKEEP0	SRAM0	8200 0000h to 8201 FFFFh (Secure area) 0200 0000h to 0201 FFFFh (Non-secure area)
RKEEP1	SRAM0	8202 0000h to 8203 FFFFh (Secure area) 0202 0000h to 0203 FFFFh (Non-secure area)
RKEEP2	SRAM0	8204 0000h to 8205 FFFFh (Secure area) 0204 0000h to 0205 FFFFh (Non-secure area)
RKEEP3	SRAM0	8206 0000h to 8207 FFFFh (Secure area) 0206 0000h to 0207 FFFFh (Non-secure area)
RKEEP4	SRAM0	8208 0000h to 8209 FFFFh (Secure area) 0208 0000h to 0209 FFFFh (Non-secure area)
RKEEP5	SRAM0	820A 0000h to 820B FFFFh (Secure area) 020A 0000h to 020B FFFFh (Non-secure area)

Table 11.5 Relationship Between Each Bit and the Target Memory (2/2)

Bit in the PDRAMSCR0 Register	Target Memory	
RKEEP6	SRAM0	820C 0000h to 820D FFFFh (Secure area) 020C 0000h to 020D FFFFh (Non-secure area)
RKEEP7	SRAM0	820E 0000h to 820F FFFFh (Secure area) 020E 0000h to 020F FFFFh (Non-secure area)
RKEEP8	SRAM1	8210 0000h to 8211 FFFFh (Secure area) 0210 0000h to 0211 FFFFh (Non-secure area)
RKEEP9	SRAM1	8212 0000 to 8213 FFFFh (Secure area) 0212 0000 to 0213 FFFFh (Non-secure area)
RKEEP10	SRAM1	8214 0000 to 8215 FFFFh (Secure area) 0214 0000 to 0215 FFFFh (Non-secure area)
RKEEP11	SRAM1	8216 0000h to 8217 FFFFh (Secure area) 0216 0000h to 0217 FFFFh (Non-secure area)

11.2.16 SRAM Power Domain Standby Control Register 2 (PDRAMSCR2)

Address(es): SYSC.PDRAMSCR2 8701 9143h, SYSC_NS.PDRAMSCR2 0701 9143h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RKEEP0
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RKEEP0	RAM Retention(Local RAM)	0: When entering the software standby mode, the contents of the target RAM are not retained 1: When entering the software standby mode, the contents of the target RAM are retained	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The PDRAMSCR2 register is used to control the target memory in software standby mode.

RKEEP0 Bit (RAM Retention)

The RKEEP0 bit controls whether to retain the contents of the local RAM in software standby mode.

11.2.17 Standby Control Register (SBYCR)

Address(es): SYSC.SBYCR 8701 900Ch, SYSC_NS.SBYCR 0701 900Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OPE	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The SBYCR register is used to control the bus signal output in software standby mode and deep software standby mode.

OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, ALE, BACK#, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0 to DQM3), or to set the output to the high-impedance state in software standby mode or deep software standby mode.

11.2.18 Software Standby Control Register 1 (SSCR1)

Address(es): SYSC.SSCR1 8701 9A98h, SYSC_NS.SSCR1 0701 9A98h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SS2LP[1:0]		—	SS2FR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SS2FR	Software Standby 2 Fast Return	0: When returning from software standby mode 2, the fast return function is disabled 1: When returning from software standby mode 2, fast return function is enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	SS2LP[1:0]	Software Standby 2 Low Power Select	b3 b2 0 0: SS2LP_0 (default mode) 0 1: SS2LP_1 (low power setting 1) Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

SS2FR Bit (Software Standby 2 Fast Return)

The SS2FR bit controls the time required to return from software standby mode 2 when the SS2LP[0] bit is 1.
When the SS2LP[1:0] bits are set to 01b (SS2LP_1), setting the SS2FR bit to 1 shortens the time required to return from software standby mode 2. Setting this bit to 1 is recommended. The only effect of setting this bit to 0 is extending the time to return from software standby mode 2. For details on the time to return, see section 67, Electrical Characteristics.
When the SS2LP[1:0] bits are set to 00b, the SS2FR bit setting is invalid.

SS2LP[1:0] Bit (Software Standby 2 Low Power Select)

The SS2LP[1:0] bits control the current consumption in software standby mode 2. Setting these bits to 01b (SS2LP_1) reduces the current consumption in software standby mode 2 relative to that when the bits are set to 00b (SS2LP_0). Since the time to return from software standby mode 2 becomes longer, setting the SS2FR bit to 1 is recommended. For details on the current consumption and time to return, see section 67, Electrical Characteristics.

11.2.19 Low Power Consumption State Control Register (LPSCR)

Address(es): SYSC.LPSCR 8701 9A90h, SYSC_NS.LPSCR 0701 9A90h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LPMD[3:0]			
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LPMD[3:0]	Low Power Consumption Mode Setting	0h: Software standby mode/deep software standby mode is disabled 4h: Software standby mode 1 5h: Software standby mode 2 8h: Deep software standby mode 1 9h: Deep software standby mode 2 Ah: Deep software standby mode 3 Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The LPSCR register is used to control the transition destination of low power consumption mode.
The LPSCR register is not initialized by the internal reset signal (deep software standby reset signal) that is the source for release from deep software standby mode. For details, see section 6, Resets.

LPMD[3:0] Bits (Low Power Consumption Mode Setting)

The LPMD[3:0] bits indicate the destination of low power consumption mode. Actual low power consumption mode transition is executed by the WAIT instruction in the CPU.
The transition of software standby mode and deep software standby mode does not occur unless each WAIT instruction is executed in the CPU.
When an interrupt has initiated release from the low power consumption mode, the value of the LPMD[3:0] bits remains the same as before the transition. Writing 0 to this bit clears the bit itself.
Depending on the settings of bits in related registers, the transition to the low power consumption mode specified by the LPMD[3:0] bits may not proceed. For details, see Table 11.7.
While the MENTRYR.MENTRY bit is 1, the selection of software standby mode or deep software standby mode is ineffective. Even if the value of the LPMD[3:0] bits is any of 4h, 5h, or 8h to Ah, the MCU is placed in CPU deep sleep mode in response to the execution of a WAIT instruction.
While the OFS0.IWDTSLCSTP bit is 0 (counting continues), the selection of deep software standby mode 2 or 3 is ineffective. Even if the value of the LPMD[3:0] bits is any of 9h or Ah, the MCU is placed in deep software standby mode 1 in response to the execution of the WAIT instruction.
While the IWDTCSTPR.SLCSTP bit is 0 (counting continues), the selection of deep software standby mode 2 or 3 is ineffective. Even if the value of the LPMD[3:0] bits is any of 9h or Ah, the MCU is placed in deep software standby mode 1 in response to the execution of the WAIT instruction.
While the voltage monitor n (n = 1, 2) reset is enabled (PVDnCR0.RI = 1), the selection of deep software standby mode 2 or 3 is ineffective. Even if the value of the LPMD[3:0] bits is any of 9h or Ah, the MCU is placed in deep software standby mode 1 in response to the execution of the WAIT instruction.
If at least one of the PVDnCR0.RE (n = 4, 5) bis is 1, both of the PVDnCR0.RI (n = 1, 2) bits are 0, and stopping the counting operation of the IWDT in a low power consumption mode is specified (the OFS0.IWDTSLCSTP bit is 1 in auto-start mode or the IWDTCSTPR.SLCSTP bit is 1 in register start mode), the selection of deep software standby

mode 3 will be ineffective. Even if the value of the LPMD[3:0] bits is Ah, the MCU is placed in deep software standby mode 2 in response to the execution of the WAIT instruction.

11.2.20 Deep Software Standby Control Register (DPSBYCR)

Address(es): SYSC.DPSBYCR 8701 9A00h, SYSC_NS.DPSBYCR 0701 9A00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	IOKEEP	—	—	DCSSMODE[1:0]		LDOSSMODE[1:0]	
Value after reset:	0	0	0	1	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LDOSSMODE[1:0]	LDO Soft Start Mode	Select the return time from deep software standby mode in LDO mode b1 b0 0 0: Setting prohibited 0 1: LDOSSMODE_1 1 0: LDOSSMODE_2 1 1: LDOSSMODE_3	R/W
b3, b2	DCSSMODE[1:0]	DCDC Soft Start Mode	Select the return time from deep software standby mode in DCDC mode b3 b2 0 0: Setting prohibited 0 1: DCSSMODE_1 1 0: DCSSMODE_2 1 1: DCSSMODE_3	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Upon release from deep software standby mode, the I/O ports, and D/A output are in the reset state 1: Upon release from deep software standby mode, the I/O ports, and D/A output are in the same state as in the deep software standby mode	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DPSBYCR register is used to control the deep software standby mode.
The DPSBYCR register is not initialized by the internal reset signal (deep software standby reset signal) that is the source for release from deep software standby mode. For details, see section 6, Resets.

LDOSSMODE[1:0] Bits (LDO Soft Start Mode)

The LDOSSMODE[1:0] bits are used to select the time required to return from deep software standby mode in LDO mode. Increasing the time to return reduces the inrush current. For details on the time to return and inrush current, see section 67, Electrical Characteristics.

DCSSMODE[1:0] Bits (DCDC Soft Start Mode)

The DCSSMODE[1:0] bits are used to select the time required to return from deep software standby mode in DCDC mode. Increasing the time to return reduces the inrush current. For details on the time to return and inrush current, see section 67, Electrical Characteristics.

IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports, and D/A output keep the same states as in the software standby mode. The IOKEEP bit determines whether the state of the I/O ports, and D/A output should be reset after release from deep software standby mode.

11.2.21 Deep Software Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): SYSC.DPSIER0 8701 9A08h, SYSC_NS.DPSIER0 0701 9A08h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Release from deep software standby mode by the IRQ0-DS pin is disabled 1: Release from deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Release from deep software standby mode by the IRQ1-DS pin is disabled 1: Release from deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Release from deep software standby mode by the IRQ2-DS pin is disabled 1: Release from deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Release from deep software standby mode by the IRQ3-DS pin is disabled 1: Release from deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Release from deep software standby mode by the IRQ4-DS pin is disabled 1: Release from deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Release from deep software standby mode by the IRQ5-DS pin is disabled 1: Release from deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Release from deep software standby mode by the IRQ6-DS pin is disabled 1: Release from deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Release from deep software standby mode by the IRQ7-DS pin is disabled 1: Release from deep software standby mode by the IRQ7-DS pin is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIER0 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER0 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR0 register being set to 1. Therefore, the DPSIFR0 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.22 Deep Software Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): SYSC.DPSIER1 8701 9A0Ch, SYSC_NS.DPSIER1 0701 9A0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	0: Release from deep software standby mode by the IRQ8-DS pin is disabled 1: Release from deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Release from deep software standby mode by the IRQ9-DS pin is disabled 1: Release from deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Release from deep software standby mode by the IRQ10-DS pin is disabled 1: Release from deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Release from deep software standby mode by the IRQ11-DS pin is disabled 1: Release from deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Release from deep software standby mode by the IRQ12-DS pin is disabled 1: Release from deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Release from deep software standby mode by the IRQ13-DS pin is disabled 1: Release from deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Release from deep software standby mode by the IRQ14-DS pin is disabled 1: Release from deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Release from deep software standby mode by the IRQ15-DS pin is disabled 1: Release from deep software standby mode by the IRQ15-DS pin is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIER1 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER1 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR1 register being set to 1. Therefore, the DPSIFR1 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.23 Deep Software Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): SYSC.DPSIER2 8701 9A10h, SYSC_NS.DPSIER2 0701 9A10h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIE	DRTCAIE	DRTCIIE	DPVD2IE	DPVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPVD1IE	PVD1 Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by PVD1 is disabled 1: Release from deep software standby mode by PVD1 is enabled	R/W
b1	DPVD2IE	PVD2 Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by PVD2 is disabled 1: Release from deep software standby mode by PVD2 is enabled	R/W
b2	DRTCIIE	RTC Periodic interrupt Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by the RTC periodic interrupt signal is disabled 1: Release from deep software standby mode by the RTC periodic interrupt signal is enabled	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by the RTC alarm interrupt signal is disabled 1: Release from deep software standby mode by the RTC alarm interrupt signal is enabled	R/W
b4	DNMIE	NMI Pin Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by the NMI pin is disabled 1: Release from deep software standby mode by the NMI pin is enabled	R/W*1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

The DPSIER2 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER2 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR2 register being set to 1. Therefore, the DPSIFR2 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.24 Deep Software Standby Interrupt Enable Register 3 (DPSIER3)

Address(es): SYSC.DPSIER3 8701 9A14h, SYSC_NS.DPSIER3 0701 9A14h

	b7	b6	b5	b4	b3	b2	b1	b0
	DVBATTADIE	DSOSTDIE	DIWDTIE	—	—	—	DUSBHSIE	DUSBFSIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DUSBFSIE	USB0 Suspend/Resume Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by the USBFS suspend/resume is disabled 1: Release from deep software standby mode by the USBFS suspend/resume is enabled	R/W
b1	DUSBHSIE	USBHS Suspend/Resume Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by the USBHS suspend/resume is disabled 1: Release from deep software standby mode by the USBHS suspend/resume is enabled	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	DIWDTIE	IWDT Underflow Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by an IWDT underflow is disabled 1: Release from deep software standby mode by an IWDT underflow is enabled	R/W
b6	DSOSTDIE	Sub-Clock Oscillation Stop Detection Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by sub-clock oscillation stop detection is disabled 1: Release from deep software standby mode by sub-clock oscillation stop detection is enabled	R/W
b7	DVBATTADIE	VBATT Tamper Detection Deep Software Standby Release Signal Enable	0: Release from deep software standby mode by VBATT tamper detection is disabled 1: Release from deep software standby mode by VBATT tamper detection is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIER3 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER3 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR3 register being set to 1. Therefore, the DPSIFR3 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.25 Deep Software Standby Interrupt Enable Register 4 (DPSIER4)

Address(es): SYSC.DPSIER4 8701 9B40h, SYSC_NS.DPSIER4 0701 9B40h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ23E	DIRQ22E	DIRQ21E	DIRQ20E	DIRQ19E	DIRQ18E	DIRQ17E	DIRQ16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ16E	IRQ16-DS Pin Enable	0: Release from deep software standby mode by the IRQ16-DS pin is disabled 1: Release from deep software standby mode by the IRQ16-DS pin is enabled	R/W
b1	DIRQ17E	IRQ17-DS Pin Enable	0: Release from deep software standby mode by the IRQ17-DS pin is disabled 1: Release from deep software standby mode by the IRQ17-DS pin is enabled	R/W
b2	DIRQ18E	IRQ18-DS Pin Enable	0: Release from deep software standby mode by the IRQ18-DS pin is disabled 1: Release from deep software standby mode by the IRQ18-DS pin is enabled	R/W
b3	DIRQ19E	IRQ19-DS Pin Enable	0: Release from deep software standby mode by the IRQ19-DS pin is disabled 1: Release from deep software standby mode by the IRQ19-DS pin is enabled	R/W
b4	DIRQ20E	IRQ20-DS Pin Enable	0: Release from deep software standby mode by the IRQ20-DS pin is disabled 1: Release from deep software standby mode by the IRQ20-DS pin is enabled	R/W
b5	DIRQ21E	IRQ21-DS Pin Enable	0: Release from deep software standby mode by the IRQ21-DS pin is disabled 1: Release from deep software standby mode by the IRQ21-DS pin is enabled	R/W
b6	DIRQ22E	IRQ22-DS Pin Enable	0: Release from deep software standby mode by the IRQ22-DS pin is disabled 1: Release from deep software standby mode by the IRQ22-DS pin is enabled	R/W
b7	DIRQ23E	IRQ23-DS Pin Enable	0: Release from deep software standby mode by the IRQ23-DS pin is disabled 1: Release from deep software standby mode by the IRQ23-DS pin is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIER4 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER4 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR4 register being set to 1. Therefore, the DPSIFR4 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.26 Deep Software Standby Interrupt Enable Register 5 (DPSIER5)

Address(es): SYSC.DPSIER5 8701 9B44h, SYSC_NS.DPSIER5 0701 9B44h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ31E	DIRQ30E	DIRQ29E	DIRQ28E	DIRQ27E	DIRQ26E	DIRQ25E	DIRQ24E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ24E	IRQ24-DS Pin Enable	0: Release from deep software standby mode by the IRQ24-DS pin is disabled 1: Release from deep software standby mode by the IRQ24-DS pin is enabled	R/W
b1	DIRQ25E	IRQ25-DS Pin Enable	0: Release from deep software standby mode by the IRQ25-DS pin is disabled 1: Release from deep software standby mode by the IRQ25-DS pin is enabled	R/W
b2	DIRQ26E	IRQ26-DS Pin Enable	0: Release from deep software standby mode by the IRQ26-DS pin is disabled 1: Release from deep software standby mode by the IRQ26-DS pin is enabled	R/W
b3	DIRQ27E	IRQ27-DS Pin Enable	0: Release from deep software standby mode by the IRQ27-DS pin is disabled 1: Release from deep software standby mode by the IRQ27-DS pin is enabled	R/W
b4	DIRQ28E	IRQ28-DS Pin Enable	0: Release from deep software standby mode by the IRQ28-DS pin is disabled 1: Release from deep software standby mode by the IRQ28-DS pin is enabled	R/W
b5	DIRQ29E	IRQ29-DS Pin Enable	0: Release from deep software standby mode by the IRQ29-DS pin is disabled 1: Release from deep software standby mode by the IRQ29-DS pin is enabled	R/W
b6	DIRQ30E	IRQ30-DS Pin Enable	0: Release from deep software standby mode by the IRQ30-DS pin is disabled 1: Release from deep software standby mode by the IRQ30-DS pin is enabled	R/W
b7	DIRQ31E	IRQ31-DS Pin Enable	0: Release from deep software standby mode by the IRQ31-DS pin is disabled 1: Release from deep software standby mode by the IRQ31-DS pin is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIER5 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

After the setting of the DPSIER5 register is modified, an edge may be internally generated depending on the state of the pin, resulting in the DPSIFR5 register being set to 1. Therefore, the DPSIFR5 register should be set to 0 before a transition to deep software standby mode.

For the sources of interrupts that cause release of the MCU from each type of deep software standby mode, see Table 11.4.

11.2.27 Deep Software Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): SYSC.DPSIFR0 8701 9A18h, SYSC_NS.DPSIFR0 0701 9A18h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ0-DS pin 1: Request for release is generated on the IRQ0-DS pin	R/W*1
b1	DIRQ1F	IRQ1-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ1-DS pin 1: Request for release is generated on the IRQ1-DS pin	R/W*1
b2	DIRQ2F	IRQ2-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ2-DS pin 1: Request for release is generated on the IRQ2-DS pin	R/W*1
b3	DIRQ3F	IRQ3-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ3-DS pin 1: Request for release is generated on the IRQ3-DS pin	R/W*1
b4	DIRQ4F	IRQ4-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ4-DS pin 1: Request for release is generated on the IRQ4-DS pin	R/W*1
b5	DIRQ5F	IRQ5-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ5-DS pin 1: Request for release is generated on the IRQ5-DS pin	R/W*1
b6	DIRQ6F	IRQ6-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ6-DS pin 1: Request for release is generated on the IRQ6-DS pin	R/W*1
b7	DIRQ7F	IRQ7-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ7-DS pin 1: Request for release is generated on the IRQ7-DS pin	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a request for release specified by the DPSIEGR0 register is generated.
This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER0 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR0 register is cleared to 00h.
To clear the DPSIFR0 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIFR0 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER0 register.
The DPSIFR0 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DIRQnF Flag (IRQn-DS Pin Deep Software Standby Release Flag) (n = 0 to 7)

The DIRQnF flag indicates that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR0 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

11.2.28 Deep Software Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): SYSC.DPSIFR1 8701 9A1Ch, SYSC_NS.DPSIFR1 0701 9A1Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8F	IRQ8-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ8-DS pin 1: Request for release is generated on the IRQ8-DS pin	R/W*1
b1	DIRQ9F	IRQ9-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ9-DS pin 1: Request for release is generated on the IRQ9-DS pin	R/W*1
b2	DIRQ10F	IRQ10-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ10-DS pin 1: Request for release is generated on the IRQ10-DS pin	R/W*1
b3	DIRQ11F	IRQ11-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ11-DS pin 1: Request for release is generated on the IRQ11-DS pin	R/W*1
b4	DIRQ12F	IRQ12-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ12-DS pin 1: Request for release is generated on the IRQ12-DS pin	R/W*1
b5	DIRQ13F	IRQ13-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ13-DS pin 1: Request for release is generated on the IRQ13-DS pin	R/W*1
b6	DIRQ14F	IRQ14-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ14-DS pin 1: Request for release is generated on the IRQ14-DS pin	R/W*1
b7	DIRQ15F	IRQ15-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ15-DS pin 1: Request for release is generated on the IRQ15-DS pin	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a request for release specified by the DPSIEGR1 register is generated.
This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER1 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR1 register is cleared to 00h.
To clear the DPSIFR1 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIFR1 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER1 register.
The DPSIFR1 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DIRQnF Flag (IRQn-DS Pin Deep Software Standby Release Flag) (n = 8 to 15)

The DIRQnF flag indicates that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR1 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

11.2.29 Deep Software Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): SYSC.DPSIFR2 8701 9A20h, SYSC_NS.DPSIFR2 0701 9A20h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIF	DRTCAIF	DRTCIIF	DPVD2IF	DPVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPVD1IF	PVD1 Deep Software Standby Release Flag	0: Request for release by PVD1 is not generated 1: Request for release by PVD1 is generated	R/W*1
b1	DPVD2IF	PVD2 Deep Software Standby Release Flag	0: Request for release by PVD2 is not generated 1: Request for release by PVD2 is generated	R/W*1
b2	DRTCIIF	RTC Periodic Interrupt Deep Software Standby Release Flag	0: Request for release by the RTC periodic interrupt signal is not generated 1: Request for release by the RTC periodic interrupt signal is generated	R/W*1
b3	DRTCAIF	RTC Alarm Interrupt Deep Software Standby Release Flag	0: Request for release by the RTC alarm interrupt signal is not generated 1: Request for release by the RTC alarm interrupt signal is generated	R/W*1
b4	DNMIF	NMI Pin Deep Software Standby Release Flag	0: Request for release is not generated on the NMI pin 1: Request for release is generated on the NMI pin	R/W*1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

The PVDm deep software standby release flag (m = 1, 2) and NMI pin deep software standby release flag are set to 1 when the request for release corresponding to that specified in the DPSIEGR2 register is generated. The RTC periodic interrupt deep software standby release flag and RTC alarm interrupt deep software standby release flag are also set to 1 when the corresponding request for release is generated.

This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER2 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR2 register is cleared to 00h.

To clear the DPSIFR2 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIFR2 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER2 register.

The DPSIFR2 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DPVDmIF Flag (PVDm Deep Software Standby Release Flag) (m = 1, 2)

The DPVDmIF flag indicates that a request for release by the voltage monitor m signal has been generated.

[Setting condition]

- A request for release is generated by the voltage monitor m signal that is selected in the DPSIEGR2 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

DRTCIIF Flag (RTC Periodic Interrupt Deep Software Standby Release Flag)

This flag indicates that a request for release by the RTC interval interrupt signal has been generated.

[Setting condition]

- A request for release by the RTC periodic interrupt signal is generated

[Clearing condition]

- Writing 0 to each flag after 1 is read

DRTCAIF Flag (RTC Alarm Interrupt Deep Software Standby Release Flag)

This flag indicates that a request for release by the RTC alarm interrupt signal has been generated.

[Setting condition]

- A request for release by the RTC alarm interrupt signal is generated

[Clearing condition]

- Writing 0 to each flag after 1 is read

DNMIF Flag (NMI Pin Deep Software Standby Release Flag)

This flag indicates that a request for release has been generated on the NMI pin.

[Setting condition]

- A request for release is generated on the NMI pin specified by the DPSIEGR2 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

11.2.30 Deep Software Standby Interrupt Flag Register 3 (DPSIFR3)

Address(es): SYSC.DPSIFR3 8701 9A24h, SYSC_NS.DPSIFR3 0701 9A24h

	b7	b6	b5	b4	b3	b2	b1	b0
	DVBATTADIF	DSOSTDIF	DIWDTIF	—	—	—	DUSBHSIF	DUSBFSIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DUSBFSIF	USB0 Suspend/Resume Deep Software Standby Release Flag	0: Request for release by the USB suspend/resume is not generated 1: Request for release by the USB suspend/resume is generated	R/W*1
b1	DUSBHSIF	USBHS Suspend/Resume Deep Software Standby Release Flag	0: Request for release by the USBHS suspend/resume is not generated 1: Request for release by the USBHS suspend/resume is generated	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	DIWDTIF	IWDT Underflow Deep Software Standby Release Flag	0: Request for release by an IWDT underflow is not generated 1: Request for release by an IWDT underflow is generated	R/W*1
b6	DSOSTDIF	Sub-Clock Oscillation Stop Detection Deep Software Standby Release Flag	0: Request for release by sub-clock oscillation stop detection is not generated 1: Request for release by sub-clock oscillation stop detection is generated	R/W*1
b7	DVBATTADIF	VBATT Tamper Detection Deep Software Standby Release Flag	0: Request for release by VBATT tamper detection is not generated 1: Request for release by VBATT tamper detection is generated	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when the corresponding request for release is generated.
This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER3 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR3 register is cleared to 00h.
To clear the DPSIER3 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIER3 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER3 register.
The DPSIFR3 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DUSBFSIF Flag (USB0 Suspend/Resume Deep Software Standby Release Flag)

This flag indicates that a request for release by the USB0 suspend/resume has been generated.
The DUSBFSIF flag is a flag for USB0.
[Setting condition]

- A request for release by the USB0 suspend/resume is generated

[Clearing condition]

- Writing 0 after 1 is read

DUSBHSIF Flag (USBHS Suspend/Resume Deep Software Standby Release Flag)

This flag indicates that a request for release by the USBHS suspend/resume has been generated.
The DUSBHSIF flag is a flag for USBHS.
[Setting condition]

- A request for release by the USBHS suspend/resume is generated

[Clearing condition]

- Writing 0 after 1 is read

DIWDTIF Flag (IWDT Underflow Deep Software Standby Release Flag)

This flag indicates that a request for release by an IWDT underflow has been generated.
[Setting condition]

- A request for release by the IWDT is generated

[Clearing condition]

- Writing 0 after 1 is read

DSOSTDIF Flag (Sub-Clock Oscillation Stop Detection Deep Software Standby Release Flag)

This flag indicates that a request for release by sub-clock oscillation stop detection has been generated.
[Setting condition]

- A request for release by the SOSTD is generated

[Clearing condition]

- Writing 0 after 1 is read

DVBATTADIF Flag (VBATT Tamper Detection Deep Software Standby Release Flag)

This flag indicates that a request for release by the VBATT tamper detection has been generated.
[Setting condition]

- A request for release by the VBATT tamper detection is generated

[Clearing condition]

- Writing 0 after 1 is read

11.2.31 Deep Software Standby Interrupt Flag Register 4 (DPSIFR4)

Address(es): SYSC.DPSIFR4 8701 9B48h, SYSC_NS.DPSIFR4 0701 9B48h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ23F	DIRQ22F	DIRQ21F	DIRQ20F	DIRQ19F	DIRQ18F	DIRQ17F	DIRQ16F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ16F	IRQ16-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ16-DS pin 1: Request for release is generated on the IRQ16-DS pin	R/W*1
b1	DIRQ17F	IRQ17-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ17-DS pin 1: Request for release is generated on the IRQ17-DS pin	R/W*1
b2	DIRQ18F	IRQ18-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ18-DS pin 1: Request for release is generated on the IRQ18-DS pin	R/W*1
b3	DIRQ19F	IRQ19-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ19-DS pin 1: Request for release is generated on the IRQ19-DS pin	R/W*1
b4	DIRQ20F	IRQ20-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ20-DS pin 1: Request for release is generated on the IRQ20-DS pin	R/W*1
b5	DIRQ21F	IRQ21-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ21-DS pin 1: Request for release is generated on the IRQ21-DS pin	R/W*1
b6	DIRQ22F	IRQ22-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ22-DS pin 1: Request for release is generated on the IRQ22-DS pin	R/W*1
b7	DIRQ23F	IRQ23-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ23-DS pin 1: Request for release is generated on the IRQ23-DS pin	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a request for release specified by the DPSIEGR3 register is generated.
This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER4 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR4 register is cleared to 00h.
To clear the DPSIER4 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIFR4 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER4 register.
The DPSIFR4 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DIRQnF Flag (IRQn-DS Pin Deep Software Standby Release Flag) (n = 16 to 23)

The DIRQnF flag indicates that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR3 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

11.2.32 Deep Software Standby Interrupt Flag Register 5 (DPSIFR5)

Address(es): SYSC.DPSIFR5 8701 9B4Ch, SYSC_NS.DPSIFR5 0701 9B4Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ31F	DIRQ30F	DIRQ29F	DIRQ28F	DIRQ27F	DIRQ26F	DIRQ25F	DIRQ24F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ24F	IRQ24-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ24-DS pin 1: Request for release is generated on the IRQ24-DS pin	R/W*1
b1	DIRQ25F	IRQ25-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ25-DS pin 1: Request for release is generated on the IRQ25-DS pin	R/W*1
b2	DIRQ26F	IRQ26-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ26-DS pin 1: Request for release is generated on the IRQ26-DS pin	R/W*1
b3	DIRQ27F	IRQ27-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ27-DS pin 1: Request for release is generated on the IRQ27-DS pin	R/W*1
b4	DIRQ28F	IRQ28-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ28-DS pin 1: Request for release is generated on the IRQ28-DS pin	R/W*1
b5	DIRQ29F	IRQ29-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ29-DS pin 1: Request for release is generated on the IRQ29-DS pin	R/W*1
b6	DIRQ30F	IRQ30-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ30-DS pin 1: Request for release is generated on the IRQ30-DS pin	R/W*1
b7	DIRQ31F	IRQ31-DS Pin Deep Software Standby Release Flag	0: Request for release is not generated on the IRQ31-DS pin 1: Request for release is generated on the IRQ31-DS pin	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a request for release specified by the DPSIEGR4 register is generated.
This flag may be set to 1 when a request for release is generated in any mode (not only deep software standby mode). It may also be set to 1 by modifying the setting of the DPSIER5 register. Therefore, a transition to deep software standby mode should be made after the DPSIFR5 register is cleared to 00h.
To clear the DPSIER5 register to 00h after modifying it, wait for at least six PCLKB cycles, read the DPSIER5 register, and then write 0 to it. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER5 register.
The DPSIFR5 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

DIRQnF Flag (IRQn-DS Pin Deep Software Standby Release Flag) (n = 24 to 31)

The DIRQnF flag indicates that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR4 register

[Clearing condition]

- Writing 0 to each flag after 1 is read

11.2.33 Deep Software Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): SYSC.DPSIEGR0 8701 9A28h, SYSC_NS.DPSIEGR0 0701 9A28h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIEGR0 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

11.2.34 Deep Software Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): SYSC.DPSIEGR1 8701 9A2Ch, SYSC_NS.DPSIEGR1 0701 9A2Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIEGR1 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

11.2.35 Deep Software Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): SYSC.DPSIEGR2 8701 9A30h, SYSC_NS.DPSIEGR2 0701 9A30h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIEG	—	—	DPVD2EG	DPVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPVD1EG	PVD1 Edge Select	0: A request for release is generated when VCC < Vdet1 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DPVD2EG	PVD2 Edge Select	0: A request for release is generated when VCC < Vdet2 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet2 (rise) is detected	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIEGR2 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

11.2.36 Deep Software Standby Interrupt Edge Register 3 (DPSIEGR3)

Address(es): SYSC.DPSIEGR3 8701 9A34h, SYSC_NS.DPSIEGR3 0701 9A34h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ23EG	DIRQ22EG	DIRQ21EG	DIRQ20EG	DIRQ19EG	DIRQ18EG	DIRQ17EG	DIRQ16EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ16EG	IRQ16-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ17EG	IRQ17-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ18EG	IRQ18-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ19EG	IRQ19-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ20EG	IRQ20-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ21EG	IRQ21-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ22EG	IRQ22-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ23EG	IRQ23-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIEGR3 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

11.2.37 Deep Software Standby Interrupt Edge Register 4 (DPSIEGR4)

Address(es): SYSC.DPSIEGR4 8701 9B50h, SYSC_NS.DPSIEGR4 0701 9B50h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ31EG	DIRQ30EG	DIRQ29EG	DIRQ28EG	DIRQ27EG	DIRQ26EG	DIRQ25EG	DIRQ24EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ24EG	IRQ24-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ25EG	IRQ25-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ26EG	IRQ26-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ27EG	IRQ27-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ28EG	IRQ28-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ29EG	IRQ29-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ30EG	IRQ30-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ31EG	IRQ31-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

The DPSIEGR4 register is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see section 6, Resets.

11.2.38 PLL1-LDO Control Register (PLL1LDOCR)

Address(es): SYSC.PLL1LDOCR 8701 9B04h, SYSC_NS.PLL1LDOCR 0701 9B04h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LDOSTP	LDO Stop	0: PLL1-LDO is enabled 1: PLL1-LDO is stopped	R/W
b1	SKEEP	STBY Retention	0: PLL1-LDO is stopped during software standby mode 2 1: During software standby mode 2, the PLL1-LDO state before software standby mode 2 is retained	R/W
b2	—	Reserved	The read value is undefined.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The PLL1LDOCR register is used to control the dedicated LDO for PLL1 (PLL1-LDO).

LDOSTP Bit (LDO Stop)

This bit controls switching of the PLL1-LDO on and off. After setting this bit to 1, do not set it to 0 for at least 1 µs. This bit can only be set to 0 after 1 µs has elapsed since it was set to 1.
If this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.
Allow the power supply to become stable before making this PLL1 setting.

SKEEP Bit (STBY Retention)

This bit controls the PLL1-LDO state during software standby mode 2. When PLL1-LDO operation stops in software standby mode 2, power consumption in software standby mode 2 can be reduced.

11.2.39 PLL2-LDO Control Register (PLL2LDOCR)

Address(es): SYSC.PLL2LDOCR 8701 9B08h, SYSC_NS.PLL2LDOCR 0701 9B08h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LDOSTP	LDO Stop	0: PLL2-LDO is enabled 1: PLL2-LDO is stopped	R/W
b1	SKEEP	STBY Retention	0: PLL2-LDO is stopped during software standby mode 2 1: During software standby mode 2, the PLL2-LDO state before software standby mode 2 is retained	R/W
b2	—	Reserved	The read value is undefined.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The PLL2LDOCR register is used to control the dedicated LDO for PLL2 (PLL2-LDO).

LDOSTP Bit (LDO Stop)

This bit controls switching of the PLL2-LDO on and off. After setting this bit to 1, do not set it to 0 for at least 1 µs. This bit can only be set to 0 after 1 µs has elapsed since it was set to 1.
If this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.
Allow the power supply to become stable before making this PLL2 setting.

SKEEP Bit (STBY Retention)

This bit controls the PLL2-LDO state during software standby mode. If the PLL2-LDO operation is stopped during software standby mode, the power consumption during software standby mode can be reduced.

11.2.40 HOCO/LDO Control Register (HOCOLDOCR)

Address(es): SYSC.HOCOLDOCR 8701 9B0Ch, SYSC_NS.HOCOLDOCR 0701 9B0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LDOSTP	LDO Stop	0: HOCO-LDO is enabled 1: HOCO-LDO is stopped	R/W
b1	SKEEP	STBY Retention	0: HOCO-LDO is stopped during software standby mode 2 1: During software standby mode 2, the HOCO-LDO state before software standby mode 2 is retained	R/W
b2	—	Reserved	The read value is undefined.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The HOCOLDOCR register is used to control the dedicated LDO for HOCO (HOCO-LDO).

LDOSTP Bit (LDO Stop)

This bit controls switching of the HOCO-LDO on and off. After setting this bit to 1, do not set it to 0 for at least 2 µs.
This bit can only be set to 0 after 2 µs has elapsed since it was set to 1.
If this bit is changed from 1 to 0, it takes 5 µs for the power supply to stabilize.
Allow the power supply to become stable before making this HOCO setting.

SKEEP Bit (STBY Retention)

This bit controls the HOCO-LDO state during software standby mode 2. When HOCO-LDO operation stops in software standby mode 2, power consumption in software standby mode 2 can be reduced.

11.2.41 Voltage Scaling Control Register (VSCR)

Address(es): SYSC.VSCR 8701 9014h, SYSC_NS.VSCR 0701 9014h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VSCMTSF	—	VSCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	VSCM[2:0]	Voltage Scaling Control Mode	Set the VDD target voltage. b2 b0 0 0 1: VSCR_1 0 1 0: VSCR_2 (initial value) Settings other than the above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	VSCMTSF	Voltage Scaling Control Mode Transition Status Flag	• Read 0: Transition completed 1: During transition • Write Write 0	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The VSCR register is used to control voltage scaling processing.

VSCM[2:0] Bits (Voltage Scaling Control Mode)

The VSCM bits set the VDD target voltage for voltage scaling processing. See section 67, Electrical Characteristics, for details on the VDD target voltage.

VSCMTSF Flag (Voltage Scaling Control Mode Transition Status Flag)

The VSCMTSF flag indicates the status when voltage scaling control processing is being executed. When a write attempt is made to change the voltage scaling mode, the VSCMTSF flag is set to 1. This flag is set to 0 when the voltage scaling control mode change is complete. Before performing the subsequent operation, make sure that the VSCMTSF flag is set to 0 (voltage scaling control mode transition processing is complete).

11.2.42 SSTBY Voltage Scaling Control Register (SVSCR)

Address(es): SYSC.SVSCR 8701 9A9Ch, SYSC_NS.SVSCR 0701 9A9Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVSCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SVSCM[2:0]	SSTBY Voltage Scaling Control Mode	Set the VDD target voltage in SSTBY2. b2 b0 0 0 1: SVSCR_1 0 1 0: SVSCR_2 (initial value) 0 1 1: SVSCR_3 1 0 0: SVSCR_4 1 0 1: SVSCR_5 Settings other than the above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The SVSCR register is used to control the voltage scaling processing in software standby mode 2.

SVSCM[2:0] Bits (SSTBY Voltage Scaling Control Mode)

The SVSCM bits set the VDD target voltage for voltage scaling processing in SSTBY2. See section 67, Electrical Characteristics, for details on the VDD target voltage.

When the SVSCM bit is set to SVSCR_1 and the value of the SSCR1.SS2LP[1:0] bits is 01b, transition to SSTBY2 mode is prohibited. When SVSCR_3, SVSCR_4, or SVSCR_5 is set, interrupt sources that can cause a return from SSTBY2 are limited to NMI and IRQ.

When using XSPI, setting SVSCR_4, or SVSCR_5 and transitioning to SSTBY2 is prohibited.

11.2.43 LDO Efficiency Control Register (LDOECR)

Address(es): SYSC.LDOECR 8701 9AF4h, SYSC_NS.LDOECR 0701 9AF4h

	b7	b6	b5	b4	b3	b2	b1	b0
	EFUPTSF	—	—	—	—	—	—	EFUPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EFUPE	LDO Efficiency Improvement Enable	0: LDO efficiency improvement function is disabled 1: LDO efficiency improvement function is enabled	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	EFUPTSF	LDO Efficiency Improvement Transition Status Flag	0: Transition completed 1: During transition	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The LDOECR register is used to control enabling or disabling of the LDO efficiency improvement function in LDO mode.

EFUPE Bit (LDO Efficiency Improvement Enable)

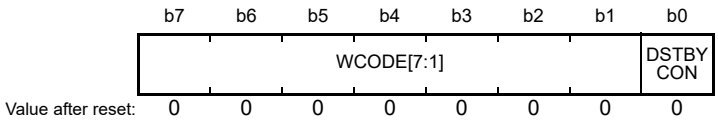
This bit is used to improve the power efficiency of the LDO. In general, setting this bit to 1 is recommended. However, the power efficiency deteriorates when the VCC voltage is low. Consult the graph of the voltage dependence of the current consumption when the LDO efficiency improvement function is enabled in section 67, Electrical Characteristics, to determine whether or not to use this function.
Do not place the MCU in software standby mode or deep software standby mode when the EFUPE bit is 1. Set the EFUPE bit to 0 and wait for the EFUPTSF flag to become 0 before proceeding to place the MCU in software standby mode or deep software standby mode.

EFUPTSF Flag (LDO Efficiency Improvement Transition Status Flag)

The EFUPTSF flag is used to indicate the status of the LDO efficiency improvement function enable/disable processing. Do not write to the EFUPE bit when this flag is set to 1.

11.2.44 DSTBY Condition Control Register (DCCR)

Address(es): SYSC.DCCR 8701 9AB8h, SYSC_NS.DCCR 0701 9AB8h



Bit	Symbol	Bit Name	Description	R/W
b0	DSTBYCON	DSTBY Control	0: Transition to deep software standby mode is disabled 1: Transition to deep software standby mode is enabled	R/W
b7 to b1	WCODE[7:1]	Write Code	1010010: Writing 1 to DSTBYCON is enabled Other than above: Writing 0 to DSTBYCON is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

To set the DSTBYCON bit to 1, write 10100101b to the WCODE[7:1] bits.
If any value other than 10100101b is written to the WCODE[7:1] bits, the DSTBYCON bit will be set to 0.

DSTBYCON Bit (DSTBY Control)

After setting the LPSCR.LPMD[3:0] bits, set the DSTBYCON bit to 1 immediately before executing the deep software standby transition with the WAIT instruction.
Note that the DSTBYCON bit is automatically set to 0 when returning from deep software standby mode. For this reason, when transitioning back to deep software standby mode, set the DSTBYCON bit to 1 immediately before executing the WAIT instruction.

11.2.45 CPU Low Power Control Register (CLPCR)

Address(es): LPM.CLPCR 8700 6BFCh, LPM_NS.CLPCR 0700 6BFCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DEEPSLEEP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DEEPSLEEP	Deep Sleep Select	0: Transition to CPU sleep mode with WAIT instruction 1: Transition to CPU deep sleep mode with WAIT instruction	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

The security attribute of this register is set with the ICUSARC.SACLPCR0 bit.

DEEPSLEEP Bit (Deep Sleep Select)

This bit selects the low power mode for the CPU.

When 0, the MCU enters CPU sleep mode with the WAIT instruction.

When 1, the MCU enters CPU deep sleep mode with the WAIT instruction.

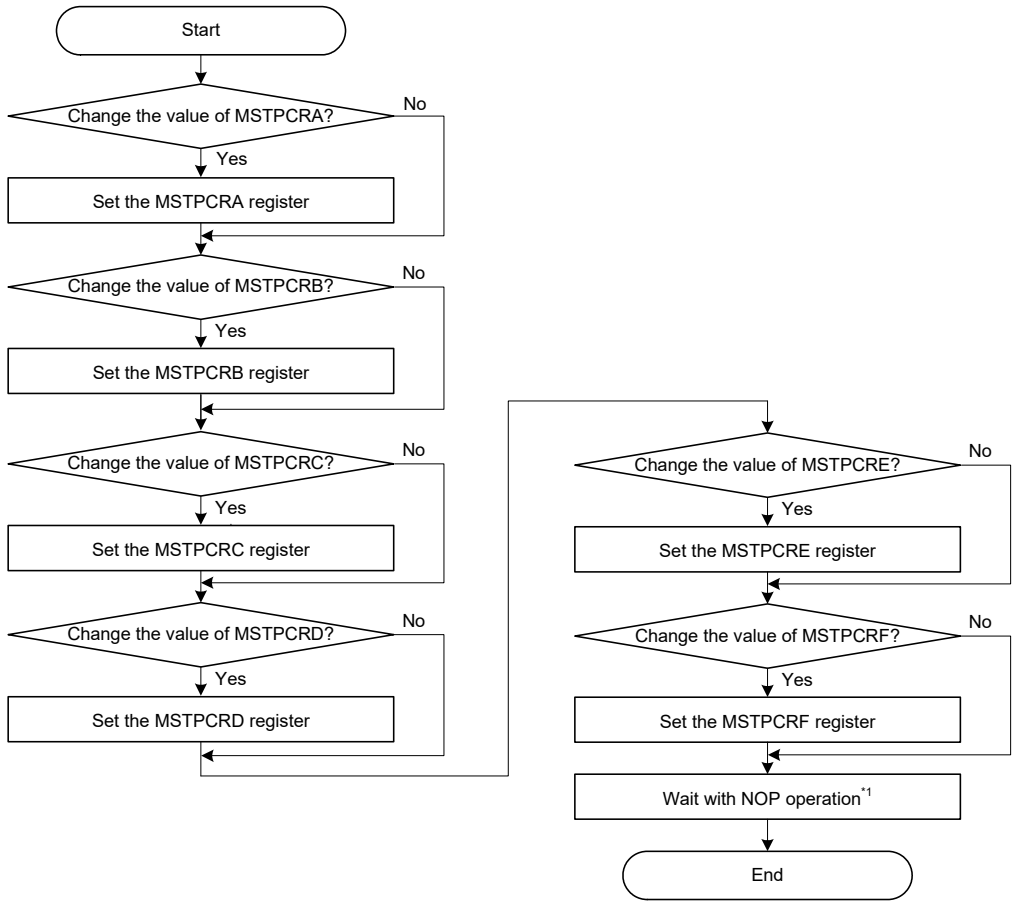
When transitioning to software standby mode or deep software standby mode, set this bit to 1.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKDIVCR and SCKDIVCR2 registers are set, the clock frequency changes.
For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to F; i = 31 to 0) in registers MSTPCRA to MSTPCRF is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module is released from the module-stop state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After release from a reset, all modules except for the DMAC, DTC, EXDMAC, and RAM are placed in the module-stop state. Do not access a module while the corresponding MSTPmi bit is 1. Read data, write data, and module operation are not guaranteed. Also, do not set the MSTPmi bit to 1 while accessing the corresponding module.
If the CPUCLK0 is faster than the ICLK maximum frequency in the electrical characteristics, after the MSTPmi bit is changed, a wait time should be provided according to the flow shown in Figure 11.2. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. This is required to take up 30 µs in DCDC mode and 10 µs in external VDD mode. No waiting time is required in LDO mode.

Figure 11.2 Setting Flow of Module Stop Control Registers in Case of CPUCLK0 is Higher than the ICLK Maximum Frequency in Electrical Characteristics

11.5 Power Gating Control

The MCU supports the power gating function of the power domains.
Users can individually control the power on/off of divided power domains.
In the power-gated state, power to the domain is switched off and the domain is placed in the same state as indicated by the items listed under SSTBY2 in Table 11.3.

11.5.1 Power Gating Control Using the Power Domain Control Register

Table 11.6 shows the relationship between the PDCTR register and the target power domain.
When the PDCTR register is set, power gating control is executed. Be sure to set the MOCOCR.MCSTP bit to 0 (MOCO operation) to use the power gating function.
Do not place the MCU in CPU deep sleep mode, software standby mode, or deep software standby mode while power domain control through the PDCTRESWM register is in use (PDCTRESWM.PDCSF = 1).

Table 11.6 Relationship Between the PDCTR Register and the Target Power Domain

Control Register				Target Power Domain Name	Target Module
Register Name	PDDE	PDCSF	PDPGSF		
PDCTRESWM	Available	Available	Available	ESWM Domain	ESWM

Example of power gating control procedure
(1) Power Off with the PDCTRESWM.PDDE Bit*1
Step 1: Make sure that the target domain is not undergoing a power gating process.
- Confirm that PDCTRESWM.PDCSF = 0.
Step 2: Set the module stop control register so that the module stops.
- Write 1 to the MSTPCRC.MSTPC30 bit.
Step 3: Set the PDCTRESWM.PDDE bit and start power control.
- Write 1 to the PDCTRESWM.PDDE bit.
Step 4: Confirm that the power gating control is complete and that the target power domain is powered off.
- Confirm that PDCTRESWM.PDCSF = 0 and PDCTRESWM.PDPGSF = 1.

(2) Power On with the PDCTRESWM.PDDE Bit*1
Step 1: Make sure that the target domain is not undergoing a power gating process.
- Confirm that PDCTRESWM.PDCSF = 0.
Step 2: Set the PDCTRESWM.PDDE bit and start power control.
- Write 1 to the PDCTRESWM.PDDE bit.
Step 3: Confirm that the power gating control is complete and that the target power domain is powered on.
- Confirm that PDCTRESWM.PDCSF = 0 and PDCTRESWM.PDPGSF = 0.
Step 4: Set the module stop control register so that the module operates.
- Write 0 to the MSTPCRC.MSTPC30 bit.

Note 1. Do not change the PDCTRESWM register when the PDCTRESWM.PDCSF flag is set to 1 (power gating control processing in progress).

11.6 Low Power Consumption Modes

11.6.1 CPU Sleep Mode/CPU Deep Sleep Mode

This MCU has two low power consumption modes for the CPU: CPU sleep mode and CPU deep sleep mode. The mode is selected by the CLPCR.DEEPSLEEP bit. In CPU deep sleep mode, power to the CPU is switched off, lengthening the time to return from the mode but reducing power consumption more sharply. For details, see section 67, Electrical Characteristics. Aside from those points, operations in relation to CPU sleep mode and CPU deep sleep mode are the same.

11.6.1.1 Transition to CPU Sleep Mode or CPU Deep Sleep Mode

When the WAIT instruction is executed with the LPSCR.LPMD[3:0] bits set to 0h and the CLPCR.DEEPSLEEP bit set to 0, the MCU enters CPU sleep mode. When the WAIT instruction is executed with the LPSCR.LPMD[3:0] bits set to 0h and the CLPCR.DEEPSLEEP bit set to 1, the MCU enters CPU deep sleep mode. Set the MOCOCR.MCSTP bit to 0 (MOCO operation) before switching to either mode. In these modes, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.

Counting by the IWDT stops if the MCU enters CPU sleep mode or CPU deep sleep mode while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to CPU sleep mode or CPU deep sleep mode).
In the same way, counting by the IWDT stops if the MCU enters CPU sleep mode or CPU deep sleep mode while the IWDT is being used in register start mode and the SLCSTP bit in the IWDTCSTPR register is 1 (stopping counting by the IWDT at transitions to CPU sleep mode or CPU deep sleep mode).

Counting by the IWDT continues even if the MCU enters CPU sleep mode or CPU deep sleep mode while the IWDT is used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to CPU deep sleep mode or CPU deep sleep mode).
In the same way, counting by the IWDT continues even if the MCU enters CPU sleep mode or CPU deep sleep mode while the IWDT is being used in register start mode and the SLCSTP bit in the IWDTCSTPR register is 0 (counting by IWDT continues through transitions to CPU sleep mode or CPU deep sleep mode).

Counting by the WDT stops if the MCU enters CPU sleep mode or CPU deep sleep mode while the WDT is being used in auto-start mode and the OFS0.WDT0SLCSTP bit is 1 (stopping counting by the WDT at transitions to CPU sleep mode or CPU deep sleep mode). In the same way, counting by the WDT stops if the MCU enters CPU sleep mode or CPU deep sleep mode while the WDT is being used in register start mode and the SLCSTP bit in the WDTCSTPR register is 1 (stopping counting by the WDT at transitions to CPU sleep mode or CPU deep sleep mode).

Counting by the WDT continues even if the MCU enters CPU sleep mode or CPU deep sleep mode while the WDT is used in auto-start mode and the OFS0.WDT0SLCSTP bit is 0 (counting by the WDT continues through transitions to CPU deep sleep mode or CPU deep sleep mode). In the same way, counting by the WDT continues even if the MCU enters CPU sleep mode or CPU deep sleep mode while the WDT is being used in register start mode and the SLCSTP bit in the WDTCSTPR register is 0 (counting by WDT continues through transitions to CPU sleep mode or CPU deep sleep mode).

11.6.1.2 Release from CPU Sleep Mode/CPU Deep Sleep Mode

The MCU is released from CPU sleep mode or CPU deep sleep mode by resets or interrupts due to the following sources:

- Interrupt
- Reset:
  - RES# pin reset
  - Power-on reset
  - Voltage monitoring 0 reset
  - Independent watchdog timer reset
  - Watchdog timer reset
  - Voltage monitoring m reset (m = 1, 2, 4, 5)
  - Common memory error reset
  - Bus error reset
  - Core voltage monitoring reset
  - Temperature-monitoring reset
  - Local memory error reset

When resets or interrupts due to these sources are generated, operation of this MCU is as indicated under the corresponding heading below.

(1) Release triggered by an interrupt signal

Generation of an interrupt request releases the MCU from CPU sleep mode or CPU deep sleep mode and the MCU starts handling for the given interrupt.

(2) Release due to a reset on the RES# pin

When the RES# pin is driven low, the MCU enters the reset state. Keep the RES# pin at the low level for the period specified in section 67, Electrical Characteristics. When the RES# pin is driven high after the specified time period, the CPU starts reset exception handling.

(3) Release due to an IWDT reset

Generation of an internal reset due to an IWDT underflow releases the MCU from CPU sleep mode or CPU deep sleep mode and the MCU starts reset exception handling. However, since the IWDT is stopped in CPU sleep mode and CPU deep sleep mode under the following conditions, it will not generate an internal reset to release the MCU from CPU sleep mode or CPU deep sleep mode.

- OFS0.IWDTSTRT = 0 (auto-start mode) and OFS0.IWDTSLCSTP = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

(4) Release due to a WDT reset

Generation of an internal reset due to a WDT underflow releases the MCU from CPU sleep mode or CPU deep sleep mode and the MCU starts reset exception handling. However, since the WDT is stopped in CPU sleep mode and CPU deep sleep mode under the following conditions, it will not generate an internal reset to release the MCU from CPU sleep mode or CPU deep sleep mode.

- OFS0.WDT0STRT = 0 (auto-start mode) and OFS0.WDT0SLCSTP = 1
- OFS0.WDT0STRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1

(5) Release due to other resets in CPU sleep mode or CPU deep sleep mode

The MCU is released from CPU sleep mode or CPU deep sleep mode by any other reset and it starts reset exception handling.

Note: For details on the settings of interrupts, see section 15, Interrupt Controller (ICUM).

11.6.2 Software Standby Mode

11.6.2.1 Transitioning to Software Standby Mode

When the WAIT instruction is executed with the LPSCR.LPMD[3:0] bits set to 4h or 5h and the CLPCR.DEEPSLEEP bit set to 1, the MCU enters software standby mode 1 or 2 according to the setting in the LPSCR.LPMD[3:0] bits. Be sure to set the MOCOCR.MCSTP bit to 0 (MOCO operation) before executing the WAIT instruction to place the MCU in this mode.

In software standby mode 1 or 2, the CPU, most of the on-chip peripheral functions and most of the oscillators stop*1. However, the contents of the CPU internal registers, RAM data, and the states of on-chip peripheral functions and the I/O ports are retained.
Software standby mode 1 or 2 allows significant reduction in power consumption because most oscillators stop in this mode. Table 11.3 shows the status of each on-chip peripheral function and oscillator.
In addition, setting the SSCR1.SS2LP[1:0] bits to 01b (low power setting 1) can further reduce power consumption in software standby mode 2. For details, see section 11.2.18, Software Standby Control Register 1 (SSCR1).

Note 1. The state of the oscillators depends on the setting of the control register for each oscillator. For details, see section 9, Clock Generation Circuit.

The status of address bus and bus control signals in software standby mode can be selected by SBYCR.OPE bit.
The SBYCR.OPE bit specifies whether the address bus and bus control signals are placed in the high-impedance state or retain the output in software standby mode.

Set the DMAST.DMST bit, DTCST.DTCST bit, and EDMAST.DMST bit to 0 before executing the WAIT instruction.

Counting by IWDT stops when the MCU enters software standby mode while the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to software standby mode). Similarly, counting by IWDT stops when the MCU enters software standby mode while the IWDT is in register start mode and the SLCSTP bit in the IWDTCSTPR register is 1 (stopping counting by the IWDT at transitions to software standby mode).

Counting by IWDT continues even when the MCU enters software standby mode while the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by IWDT continues through transitions to software standby mode). Similarly, counting by IWDT continues even when the MCU enters software standby mode while the IWDT is in register start mode and the SLCSTP bit in the IWDTCSTPR register is 0 (counting by IWDT continues through transitions to software standby mode).

WDT stops counting when the MCU enters software standby mode 1 or 2 because the PCLK stops.

Do not place the MCU in software standby mode while MRAM is being programmed. In order to switch to software standby mode in this situation, wait until the programming procedure is complete and then execute the WAIT instruction.

Do not place the MCU in software standby mode while MRAM configuration update transfer control is in progress. In order to switch to software standby mode in this situation, wait until the configuration update transfer control is complete and then execute the WAIT instruction.

Do not place the MCU in software standby mode 1 or 2 while the LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1). To set the LDOECR.EFUPE bit to 0 to switch to this mode, wait until the LDOECR.EFUPTSF flag becomes 0 before starting the transition.

In external VDD mode, do not put the MCU into software standby mode 1 or 2.

Depending on the register bit settings, the MCU may not transition to the low power mode set by the LPSCR.LPMD[3:0] bits.
Table 11.7 shows the relationship between the settings of the register bits that affect the transition to low power consumption mode, the settings of the LPSCR.LPMD[3:0] bits, and the low power consumption mode to which the transition occurs.

Table 11.7 Relationship Between the Settings of the Register Bits That Affect the Transition to Low Power Consumption Mode, Settings of the LPSCR.LPMD[3:0] Bits, and Low Power Consumption Mode to Which the Transition Occurs

		LPSCR.LPMD[3:0] Bit Setting (When the WAIT Instruction Is Executed with CLPCR.DEEPSLEEP = 1)				
		4h (SSTBY1)	5h (SSTBY2)	8h (DSTBY1)	9h (DSTBY2)	Ah (DSTBY3)
Bit setting that affects the transition to low power consumption mode	MENTRYR.MENTRY = 1	CPU deep sleep mode	CPU deep sleep mode	CPU deep sleep mode	CPU deep sleep mode	CPU deep sleep mode
	OFS0.IWDTSLCSTP = 0 (in auto-start mode)	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	Deep software standby 1	Deep software standby 1
	IWDTCSTPR.SLCSTP = 0 (in register start mode)	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	Deep software standby 1	Deep software standby 1
	PVD1CR0.RI = 1	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	Deep software standby 1	Deep software standby 1
	PVD2CR0.RI = 1	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	Deep software standby 1	Deep software standby 1
	PVD4CR0.RE = 1	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	No effect (deep software standby 2)	Deep software standby 2
	PVD5CR0.RE = 1	No effect (software standby 1)	No effect (software standby 2)	No effect (deep software standby 1)	No effect (deep software standby 2)	Deep software standby 2

Note: If multiple bit settings that affect low power consumption mode transition are set, the higher priority conditions in Table 11.7 take precedence. For example, if LPSCR.LPMD[3:0] = Ah and PVD1CR0.RI = 1 and PVD4CR0.RE = 1, the low-power mode that is transitioned to is deep software standby 1.

11.6.2.2 Release from Software Standby Mode

The MCU is released from software standby mode 1 or 2 by any of the interrupts listed in Table 11.4, a RES# pin reset, a power-on reset, a voltage monitoring reset, a core voltage monitoring reset, a reset caused by an IWDT underflow, or a temperature monitoring reset.
Upon release from software standby mode, the oscillators that were running before a transition to the mode restart operation. After the operation of all these oscillators has become stable, the MCU returns from software standby to normal operating mode.

(1) Release triggered by an interrupt signal

When an interrupt request (see Table 11.4) is generated, each oscillator that was operating before transitioning to software standby mode resumes operation. After all these oscillators have been stabilized, the MCU returns from software standby mode to normal mode and starts interrupt processing.
When release from software standby mode is initiated by an interrupt, each power domain returns to the same power state as the state before entering the software standby state.

(2) Release due to a reset on the RES# pin

When the RES# pin is set to low, the MCU transitions to the reset state, and the oscillator that was in the operating state by default starts oscillating. Be sure to keep the RES# pin low for the time period specified in section 67, Electrical Characteristics. When the RES# pin is driven high after the specified time period, the CPU starts the reset exception processing.

(3) Release due to a power-on reset

The MCU is released from software standby mode by a power-on reset and it starts the reset exception processing.

(4) Release due to a voltage monitoring reset

The MCU is released from software standby mode by a voltage monitoring reset by the voltage detection circuit and it starts the reset exception processing.

(5) Release due to an IWDT reset

When an internal reset is generated due to an IWDT underflow, the MCU is released from software standby mode and it starts reset exception processing. However, under the following conditions, the IWDT stops in software standby mode, and an internal reset to initiate a release from software standby mode is not generated.

- OFS0.IWDTSTRT = 0 (auto-start mode) and OFS0.IWDTSLCSTP = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

(6) Release due to a core voltage monitoring reset

The MCU is released from software standby mode by a core voltage monitoring reset (CVMR) and it starts the reset exception processing.

(7) Release due to a temperature monitoring reset

The MCU is released from software standby mode by a temperature monitoring reset and it then starts reset exception handling.

11.6.2.3 Example of Software Standby Mode Application

Figure 11.3 shows an example where a transition to software standby mode is made at the falling edge of the IRQn (n = 0 to 31) pin, and release from software standby mode is initiated at the rising edge of the IRQn pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] (i = 0 to 31) bits of the ICU set to 00b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). Furthermore, if MOCO is not operating, MOCOCR.MCSTP must be set to 0 (MOCO operation). Next, after the LPSCR.LPMD[3:0] bits are set to 4h and the CLPCR.DEEPSLEEP bit set to 1, the WAIT instruction is executed.
Thus, a transition to software standby mode 1 is made. After that, release from software standby mode 1 is initiated at the rising edge of the IRQn pin.
To return from software standby mode, settings of the ICU are also necessary. For details, see section 15, Interrupt Controller (ICUM). Figure 11.3 The oscillation stabilization time in is specified in section 67, Electrical Characteristics.

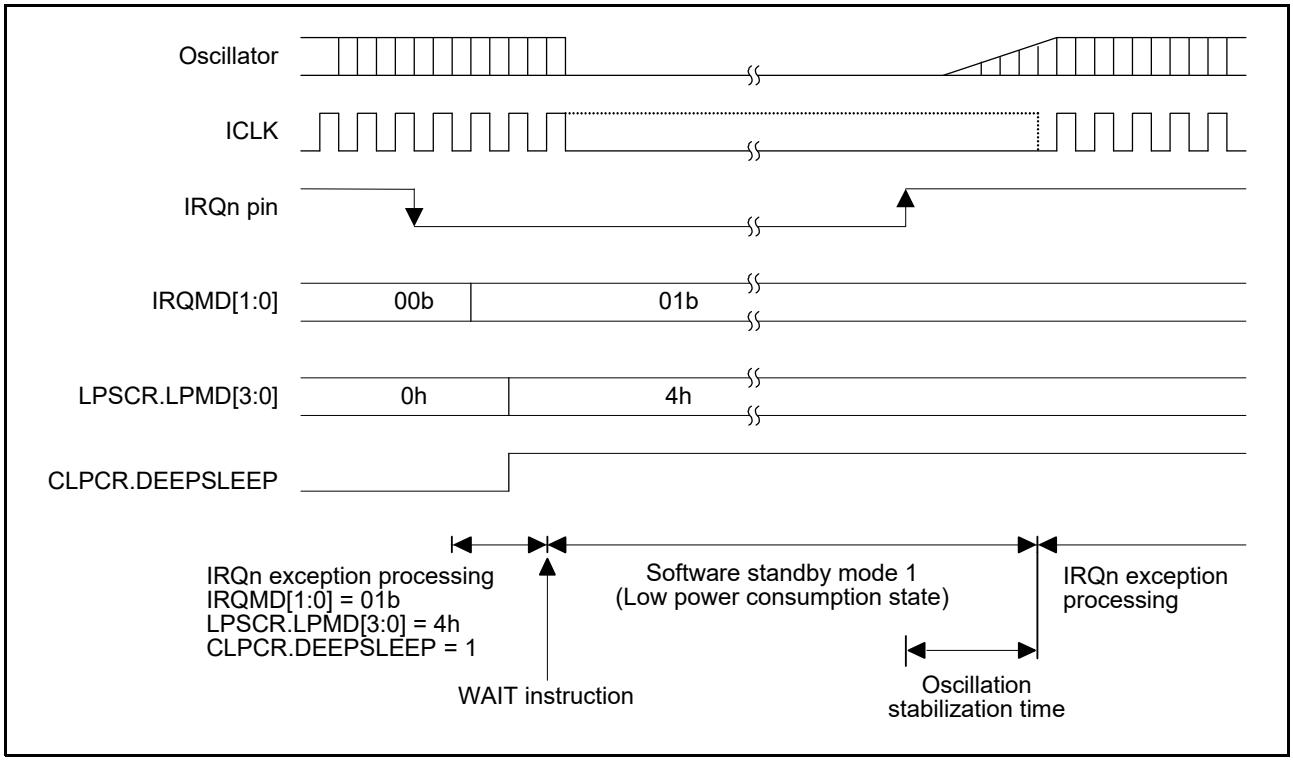


Figure 11.3 Example of Software Standby Mode Application

11.6.3 Deep Software Standby Mode

11.6.3.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the LPSCR.LPMD[3:0] bits set to 8h, 9h or Ah, the CLPCR.DEEPSLEEP bit set to 1 and the DCCR.DSTBYCON bit set to 1, the MCU enters deep software standby mode 1, 2 or 3 according to the setting in the LPSCR.LPMD[3:0] bits. Before executing the WAIT instruction to transition to this mode, the MOCOCR.MCSTP bit must be set to 0 (MOCO operation).

In these modes, almost all of the CPU, on-chip peripheral functions, SRAM, and oscillators are stopped. Furthermore, since the internal power supply to these inactive modules is stopped, power consumption is remarkably reduced. The contents of all the registers of the CPU and specific internal peripheral modules become undefined. For details, see Table 11.3.

When the battery backup function is in use in deep software standby mode 1 or 2, setting OFS1.PVDAS = 0 and OFS1.DSBYLPSEL = 0 further reduces power consumption in the deep software standby mode because the power switch is controlled by the voltage detection 0 circuit (PVD0), which consumes less power than the voltage detection circuit in the battery backup function. In deep software standby mode 2, the supply of internal power to the USB0 resume detection unit, USBHS resume detection unit, and IWDT is also stopped, further reducing power consumption. In deep software standby mode 3, the PVD is also stopped, even further reducing power consumption. For details, see section 67, Electrical Characteristics.

Set the DMAST.DMST bit, DTCST.DTCST bit, and EDMAST.DMST bit to 0 and then execute the WAIT instruction.

Depending on the settings of bits in related registers, the transition to the deep software standby mode specified by the LPSCR.LPMD[3:0] bits may not proceed. Table 11.7 lists the relationships between the register bit settings that may affect transitions to the low power consumption modes, the LPSCR.LPMD[3:0] bit settings, and the low power consumption mode to which the transition proceeds.

When the MCU is placed in deep software standby mode 2 or 3 while the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP bit is 1, the IWDT clock (IWDTCLK) stops and the power supply to the IWDT is turned off. Counting by the IWDT thus also stops.

When the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP bit is 0, the MCU enters deep software standby mode 1 instead of deep software standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD[3:0] bits.

When the IWDT is in register start mode and the IWDTCSTPR.SLCSTP is 0, the MCU enters deep software standby mode 1 instead of deep software standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD[3:0] bits.

If one of the PVDnCR0.RI (n = 1, 2) bits is set to 1 (voltage monitoring n reset selected), the MCU enters deep software standby mode 1 instead of entering deep software standby mode 2 or 3 regardless of the setting of the LPSCR.LPMD[3:0] bits.

If both of the PVDnCR0.RI (n = 1, 2) bits are set to 0, at least one of the PVDnCR0.RE (n = 4, 5) bits is set to 1, and stopping the counting operation of the IWDT in a low power consumption mode is specified (the OFS0.IWDTSLCSTP bit is 1 in auto-start mode or the IWDTCSTPR.SLCSTP bit is 1 in register start mode), the MCU enters deep software standby mode 2 instead of deep software standby mode 3 regardless of the setting of the LPSCR.LPMD[3:0] bits.

The states of the I/O port pins are the same as those in software standby mode.

Do not place the MCU in deep software standby mode while MRAM is being programmed. In order to switch to deep software standby mode in this situation, wait until the programming procedure is complete and then execute the WAIT instruction.

Do not place the MCU in deep software standby mode while MRAM configuration update transfer control is in progress. In order to switch to deep software standby mode in this situation, wait until the configuration update transfer control is complete and then execute the WAIT instruction.

Do not place the MCU in deep software standby mode in LDO mode with LDOECR.EFUPE = 1. To set the LDOECR.EFUPE bit to 0 to switch to this mode, wait until the LDOECR.EFUPTSF flag becomes 0 before starting the transition.

In external VDD mode, do not put the MCU into deep software standby mode.

11.6.3.2 Release from Deep Software Standby Mode

Release from deep software standby mode is initiated by any of the interrupts shown in Table 11.4, an RES# pin reset, a power-on reset, a voltage monitoring reset, a reset caused by an IWDT underflow, or a temperature monitoring reset.

(1) Release triggered by an interrupt signal

Release from deep software standby mode is controlled by registers DPSIERn (n = 0 to 5) and DPSIFRn (n = 0 to 5). When a deep software standby release interrupt is generated, the corresponding flag in the DPSIFRn register is set to 1. At this time, if release from deep software standby mode by the interrupt is enabled in the DPSIERn register, release from deep software standby mode proceeds. Rising edge or falling edge can be selected by the DPSIEGRn register (n = 0 to 4). The interrupts for which an edge can be selected are NMIs, IRQ0-DS to IRQ31-DS, voltage monitoring n (n = 1, 2) interrupts. When a request for release from deep software standby mode is generated, the internal power is supplied and MOCO starts oscillating, and then an internal reset (deep software standby reset) is generated for the entire MCU.
The stable MOCO clock is then supplied to the entire MCU, which is released from the deep software standby reset. Then, the MCU starts reset exception processing.
When release from deep software standby mode is triggered by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Release due to a reset on the RES# pin

When the RES# pin is driven low, the MCU is released from deep software standby mode and enters a reset state. Make sure to keep RES# pin low for the time period specified in the section of Electrical Specifications. When the RES# pin is driven high after the specified time period, the CPU starts the reset exception processing.

(3) Release due to a power-on reset

The MCU is released from deep software standby mode by a power-on reset and it starts the reset exception processing.

(4) Release due to a voltage monitoring 0 reset

The MCU is released from deep software standby mode by a voltage monitoring 0 reset from the voltage detection circuit and it starts the reset exception processing.

(5) Release due to a voltage monitoring n reset (n = 1, 2)

The MCU is released from deep software standby mode 1 or 2 by a voltage monitoring n reset (n = 1, 2) from the voltage detection circuit and it starts the reset exception processing.
A voltage monitoring n reset (n = 1, 2) from the voltage detection circuit causes the MCU to be released from deep software standby mode 1 or 2 and the MCU starts the reset exception processing.

(6) Release due to an IWDT reset

An IWDT reset (internal reset due to an underflow of the IWDT) leads to release from deep software standby mode 1 and the MCU starts the reset exception processing. However, under the following conditions, the IWDT stops in deep software standby mode 1, and an IWDT reset to initiate a release from deep software standby mode 1 is not generated.

- OFS0.IWDTSTRT = 0 (auto-start mode) and OFS0.IWDTSLCSTP = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

(7) Release due to a temperature monitoring reset

The MCU is released from deep software standby mode by a temperature monitoring reset and it then starts reset exception handling.

11.6.3.3 Pin States at the Time of Release from Deep Software Standby Mode

In deep software standby mode, the I/O ports remain in the same state as before transitioning to deep software standby mode. The MCU is initialized by an internal reset generated on release from deep software standby mode. Upon release from deep software standby mode, reset exception processing starts immediately.
Whether to initialize the I/O ports or to retain the I/O port states as before transition to deep software standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0

I/O ports are initialized by an internal reset generated on release from deep software standby mode.

- When the DPSBYCR.IOKEEP bit = 1

Although the MCU is initialized by an internal reset generated on release from deep software standby mode, I/O ports remain in the same state as before transitioning to deep software standby mode. At this time, the I/O port states remain unchanged as before transition to deep software standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by setting the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state.
The DPSBYCR.IOKEEP bit is not initialized by the internal reset signal (deep software standby reset signal) that is the source for release from deep software standby mode.

11.6.3.4 Example of Deep Software Standby Mode Application

Figure 11.4 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and release from deep software standby mode is initiated at the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge). If MOCO is not operating, the MOCOCR.MCSTP bit must be set to 0 (MOCO operation). After that, the DPSIEGRy.DIRQnEG (y = 0, 1, 4, 5; n = 0 to 31) bit is set to 1 (rising edge), the LPSCR.LPMD[3:0] bits are set to 8h, 9h or AH, the CLPCR.DEEPSLEEP bit is set to 1, the DCCR.DSTBYCON bit is set to 1, and then the WAIT instruction is executed. Thus, a transition to deep software standby mode is made. After that, release from deep software standby mode is initiated at the rising edge of the IRQn-DS pin.

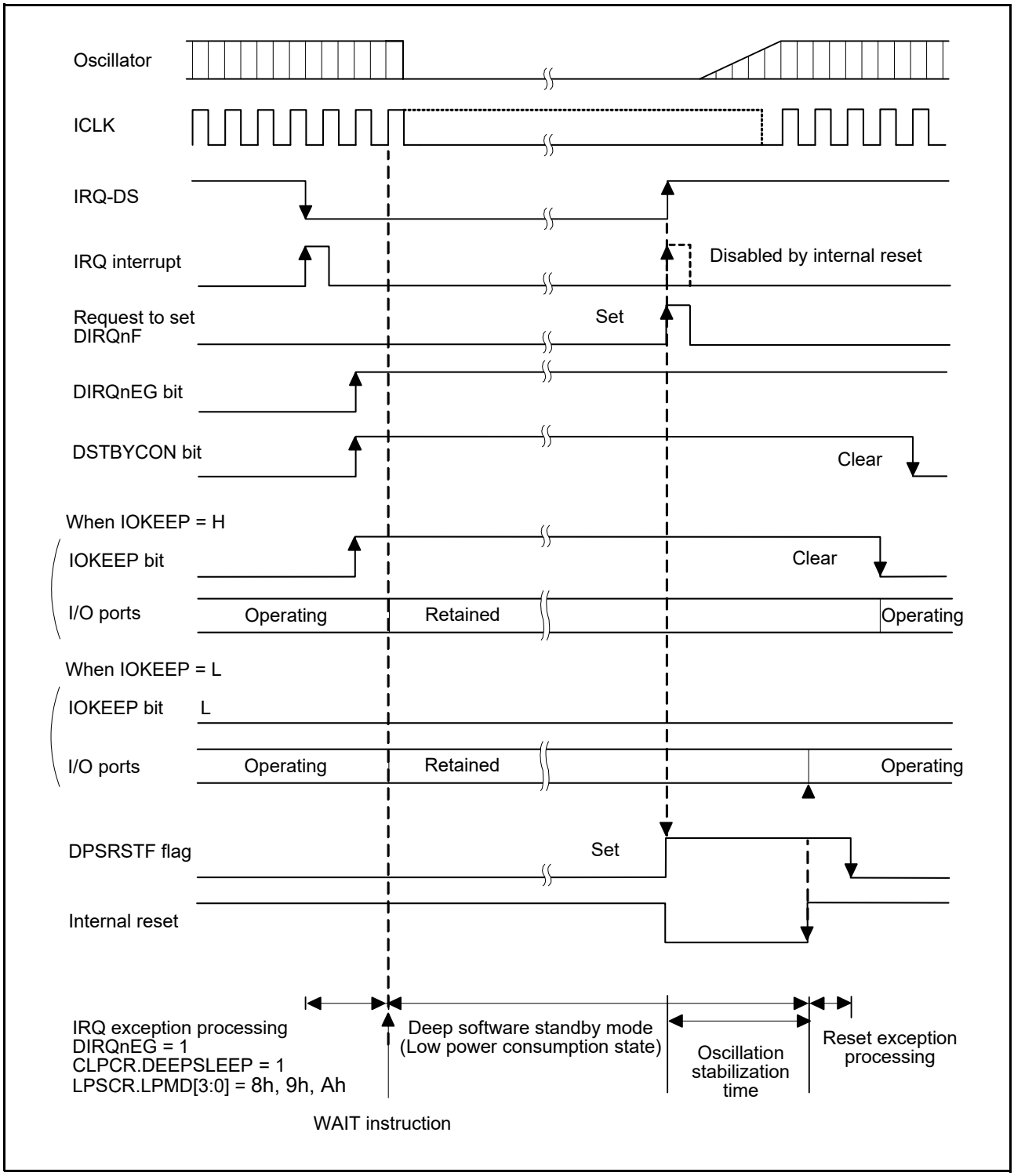


Figure 11.4 Example of Deep Software Standby Mode Application

11.6.3.5 Flowchart to Use Deep Software Standby Mode

Figure 11.5 shows an example of a flowchart to use deep software standby mode. In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception processing to determine whether a reset was generated by the RES# pin or by release from deep software standby mode. In the case of a reset by the RES# pin, a transition to deep software standby mode is made after the required register settings have been made. In the case of a reset by release from deep software standby mode, the DPSBYCR.IOKEEP bit is set to 0 after the I/O port settings have been made.

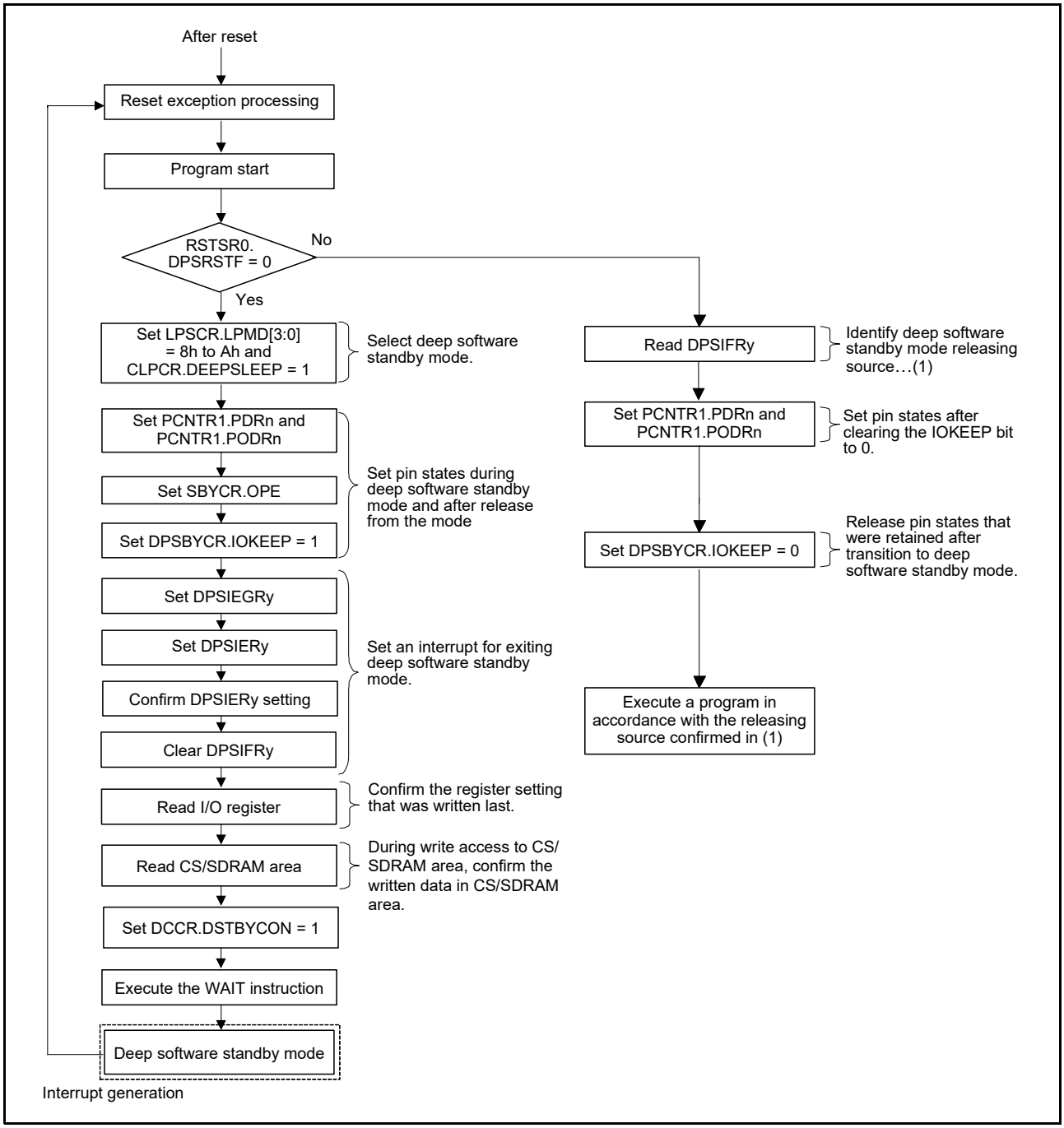


Figure 11.5 Example of Flowchart to Use Deep Software Standby Mode

11.7 Voltage Scaling Control

The MCU supports voltage scaling (DVFS) control for low-power or high-performance operation. Figure 11.6 shows the transitions of DVFS control.

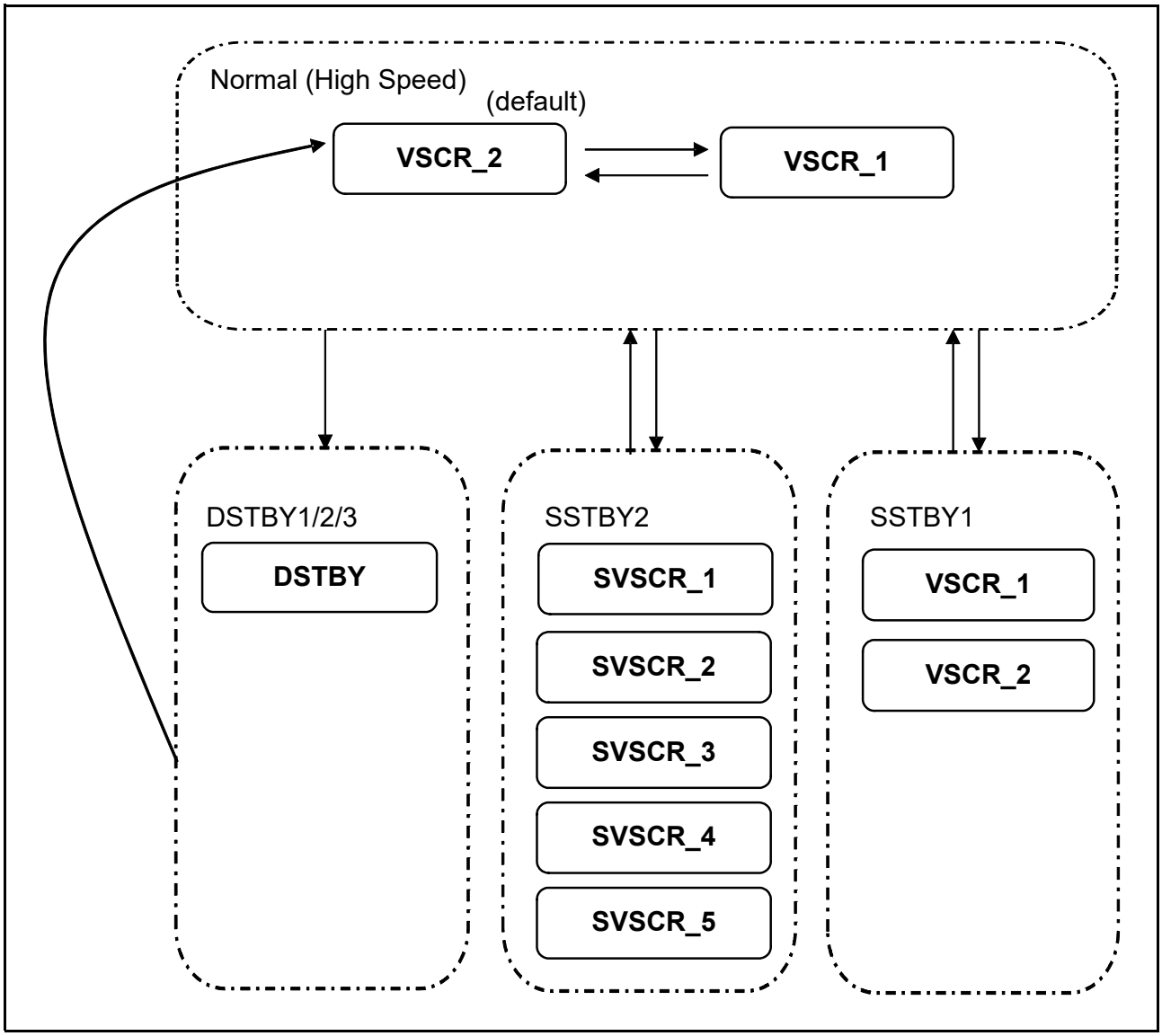


Figure 11.6 Voltage Scaling (DVFS) Control for Low Power Consumption and High Performance Operation

<Normal operating mode>

Make sure that the operating conditions (frequency range) before and after switching VDD in normal mode are always within the specified range. Switch the voltage so that each clock does not exceed the maximum frequency indicated in section 67, Electrical Characteristics.

The following is an example of the VDD switching procedure:

(1) Switching from VSCR_2 to VSCR_1

1. Set the VSCR.VSCM[2:0] bits to 001b.
2. Confirm that the VSCR.VSCMTSF flag is set to 0.
3. Set SCKDIVCR and SCKDIVCR2 to the desired values.

(2) Switching from VSCR_1 to VSCR_2

1. Set SCKDIVCR and SCKDIVCR2 to the desired values.
2. Set the VSCR.VSCM[2:0] bits to 010b.
3. Confirm that the VSCR.VSCMTSF flag is set to 0.

(3) If the PLL Settings Must Be Changed When Changing VDD

1. Change the clock source to one other than PLL.
2. Stop the PLL.
3. Change the PLL settings.
4. Set the VSCR.VSCM[2:0] bits to the desired values.
5. Confirm that the VSCR.VSCMTSF flag is set to 0.
6. Enable the PLL.
7. Wait for the PLL to stabilize.
8. Change the clock source to PLL.

<Software standby mode 1>

- A transition to and return from SSTBY1 are possible even if the VSCR.VSCM[2:0] bits are set to a value other than the default in normal operating mode.
- In SSTBY1, the setting of the VSCR.VSCM[2:0] bits is maintained, and the setting is retained and remains effective upon return from SSTBY1.

<Software standby mode 2>

- A transition to and return from SSTBY2 are possible even if the VSCR.VSCM[2:0] bits are set to a value other than the default in normal operating mode.
  The value of the SVSCR.SVSCM[2:0] bits becomes effective in SSTBY2. Prior setup of these bits is required. The value of the VSCR.VSCM[2:0] bits is retained and remains effective upon return from SSTBY2.
- With the SVSCR.SVSCM[2:0] bits are set to 001b and the SSCR1.SS2LP[1:0] bits are set to 01b, transition to SSTBY2 is prohibited.

<Deep software standby mode 1, 2, 3>

- A transition to and return from DSTBY are possible even if the VSCR.VSCM[2:0] bits are set to a value other than the default in normal operating mode.
- The VSCR.VSCM[2:0] bits are reset to the initial value upon return from DSTBY.

11.8 Usage Notes

11.8.1 Register Access

(1) Do not write to the following registers under any of the following conditions.

<Register>

- All registers whose peripheral function name is SYSC

<Conditions>

- MENTRYR.MENTRY = 1 (MRAM is being programmed)
- VSCR.VSCMTSF = 1 (during transition of voltage scaling control mode)
- Timing in the range from execution of the WAIT instruction until the return to normal operating mode

(2) Do not write to the following registers by the DTC or DMAC.

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, MSTPCRE, MSTPCRF

(3) When the PRCR.PRC1 bit is 0, write access to the following registers is invalid.

- PGSCR, PSSTCRn, PDRAMSCRn, SBYCR, SSCR1, LPSCR, DPSBYCR, DPSIERn, DPSIFRn, DPSIEGRn, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR, VSCR, SVSCR, MWMCR, LDOECR, DCCR

(4) When the PRCR.PRC4 bit is 0, write access to the following registers is invalid.

- LPMSAR, PGCSAR, DPFSAR, DPFSAR1, RSCSAR

11.8.2 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode. Therefore, the current consumption is not reduced while output signals are held high.

11.8.3 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA22 bit to 1, set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCA). Before setting the MSTPCRA.MSTPA29 bit to 1, set the EDMAST.DMST bit of the EXDMAC to 0. For details, see section 19, EXDMA Controller (EXDMACAa).

11.8.4 On-Chip Peripheral Module Interrupts

These interrupts do not operate in the module-stop state. Therefore, if the module-stop state is entered after an interrupt request is generated, a CPU interrupt source or a DMAC or DTC activating source cannot be cleared. For this reason, disable interrupts before entering the module-stop state.

11.8.5 Input Buffer Control by the DIRQnE Bit (n = 0 to 31)

Setting the DPSIERy.DIRQnE (y = 0, 1, 4, 5; n = 0 to 31) bit to 1 enables the input buffer of the IRQ0-DS to IRQ31-DS pins. Note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0, 1, 4, 5; n = 0 to 31) bit, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

11.8.6 Timing of the WAIT Instruction

It is possible for the WAIT instruction to be executed before I/O register and CS/SDRAM area writes are complete, in which case operation may not be as intended. This can happen if the WAIT instruction is executed immediately after a write to an I/O register and CS/SDRAM area. To avoid this problem, it is advisable to read back the register and CS/SDRAM area that were written to confirm that the write has completed.

11.8.7 Writing to the WDT/IWDT Registers by the DMAC or DTC

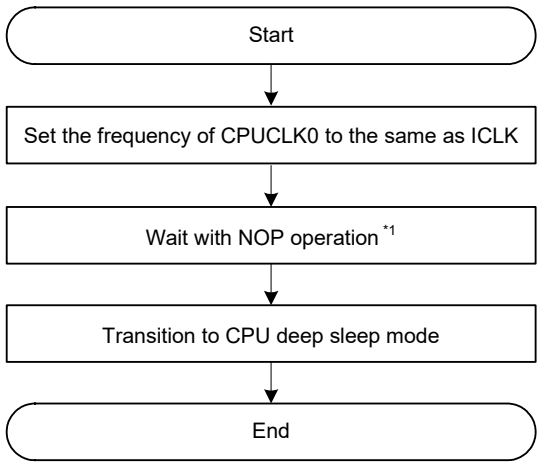
Do not write to the registers in the WDT or IWDT by the DMAC or DTC while the WDT or IWDT is stopped in processor low power consumption mode.

11.8.8 Timing of Writing to the Module Stop Bits

Access to an I/O register of a module may proceed before writing to the module stop bit that controls the module is complete. In such a case, access to the I/O register may lead to unintended behavior. To avoid this problem, read the module stop bit to confirm that the write operation is complete before proceeding with access to an I/O register.

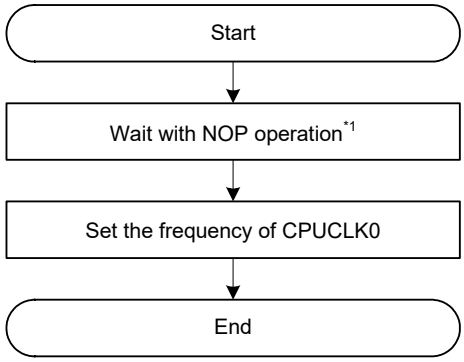
11.8.9 Notes on Transition to and Return from Low Power Modes

A transition to or return from a low power mode requires a specific procedure.
Figure 11.7 to Figure 11.10 show the flows of these procedures. The use of software to measure the stated waiting times is recommended. Consider worst-case scenarios to ensure that the required waiting times have elapsed. If an unavoidable interrupt occurs during a waiting time, measure the waiting time again after execution returns from handling of the interrupt.



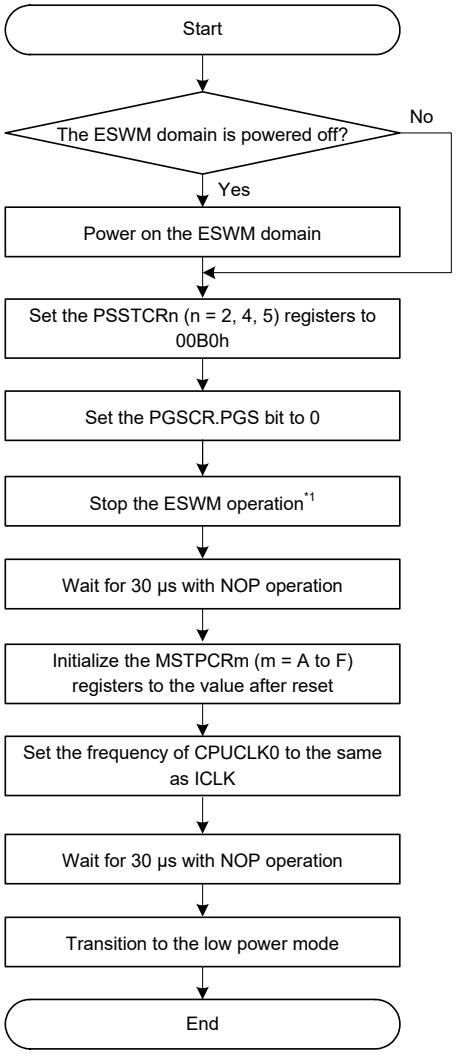
Note 1. This is required to take up 30 µs in DCDC mode and 10 µs in external VDD mode. No waiting time is required in LDO mode.

Figure 11.7 Flow for the Transition to CPU Deep Sleep Mode



Note 1. This is required to take up 30 µs in DCDC mode and 10 µs in external VDD mode. No waiting time is required in LDO mode.

Figure 11.8 Flow for Return from CPU Deep Sleep Mode (Assuming that CPUCLK0 and ICLK are Running at the Same Frequency upon Return).



Note 1. For how to stop the operation of the module, refer to the usage notes in section 33, Ethernet Module.

Figure 11.9 Flow for Transition to Software Standby Mode and Deep Software Standby Mode 1, 2, or 3

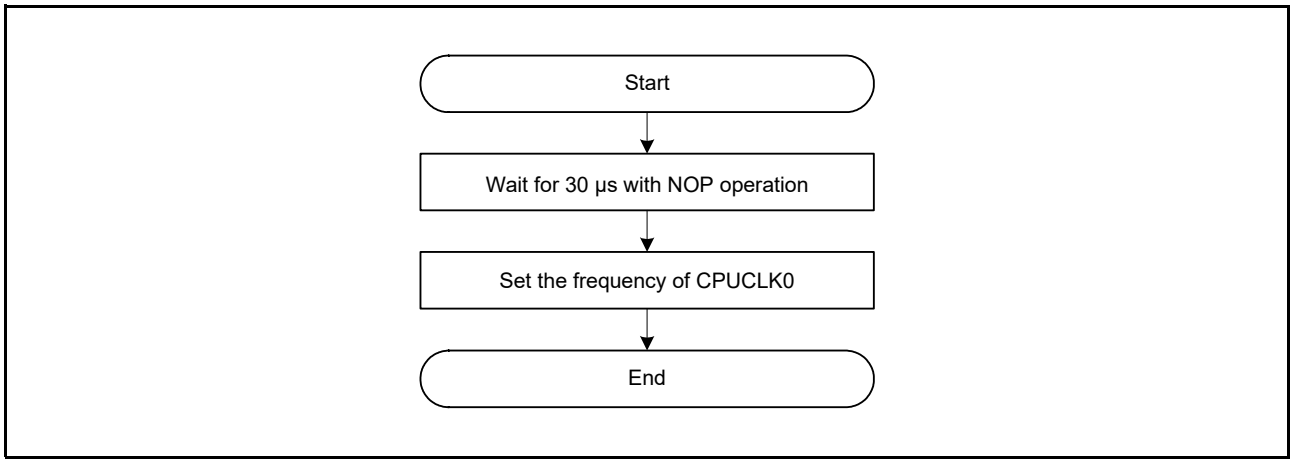


Figure 11.10 Flow for Return from Software Standby Mode (Assuming CPUCLK0 and ICLK are Running at the Same Frequency upon Return).

12. Battery Backup Function (VBATTBa)

12.1 Overview

Backup power supplied by a battery can keep the realtime clock (RTC) operating and retain the important data while the VCC power is turned off. The tamper detection function can be used to erase the backed-up data in cases where an intrusion to the system has been detected.

Table 12.1 lists the specifications of the VBATT module, Figure 12.1 is a block diagram, and Table 12.2 lists the I/O pins of the VBATT module.

Table 12.1 VBATT Specifications

Item	Description
Scope of backup	All modules in the backup domain • Backup registers • Sub-clock oscillator • Low-speed on-chip oscillator (LOCO) • Power-down detector • Tamper detector • Realtime clock (RTC)
Backup registers	128 bytes • The backup registers can be immediately erased in response to the detection of physical tampering.
Battery voltage measurement function	1/6 of the voltage applied to the VBATT pin can be measured by the A/D converter.
Backup domain power-down detection	A backup domain reset signal is generated when the supply voltage for the backup domain has fallen below V_{PDR}.
Tamper event detection	When the tamper detector has detected an intrusion to the system, it indicates this by setting a flag and, if this is selected, generating an interrupt. • Each tamper event can acquire a timestamp upon detection of a tamper event. • Three tamper input pins (TAMPI0 to TAMPI2) • Noise filter (sampling rate: 32.768 kHz or 64 to 1 Hz; passes signal transitions in response to three consecutive samples with the same level) • Capable of triggering wakeup from deep software standby mode • The backup registers can be immediately erased upon detection of a tamper event. • The hardware unique key (HUK) can be zeroized upon detection of a tamper event.
Protect Zone filter	Security attribute can be set for each register.

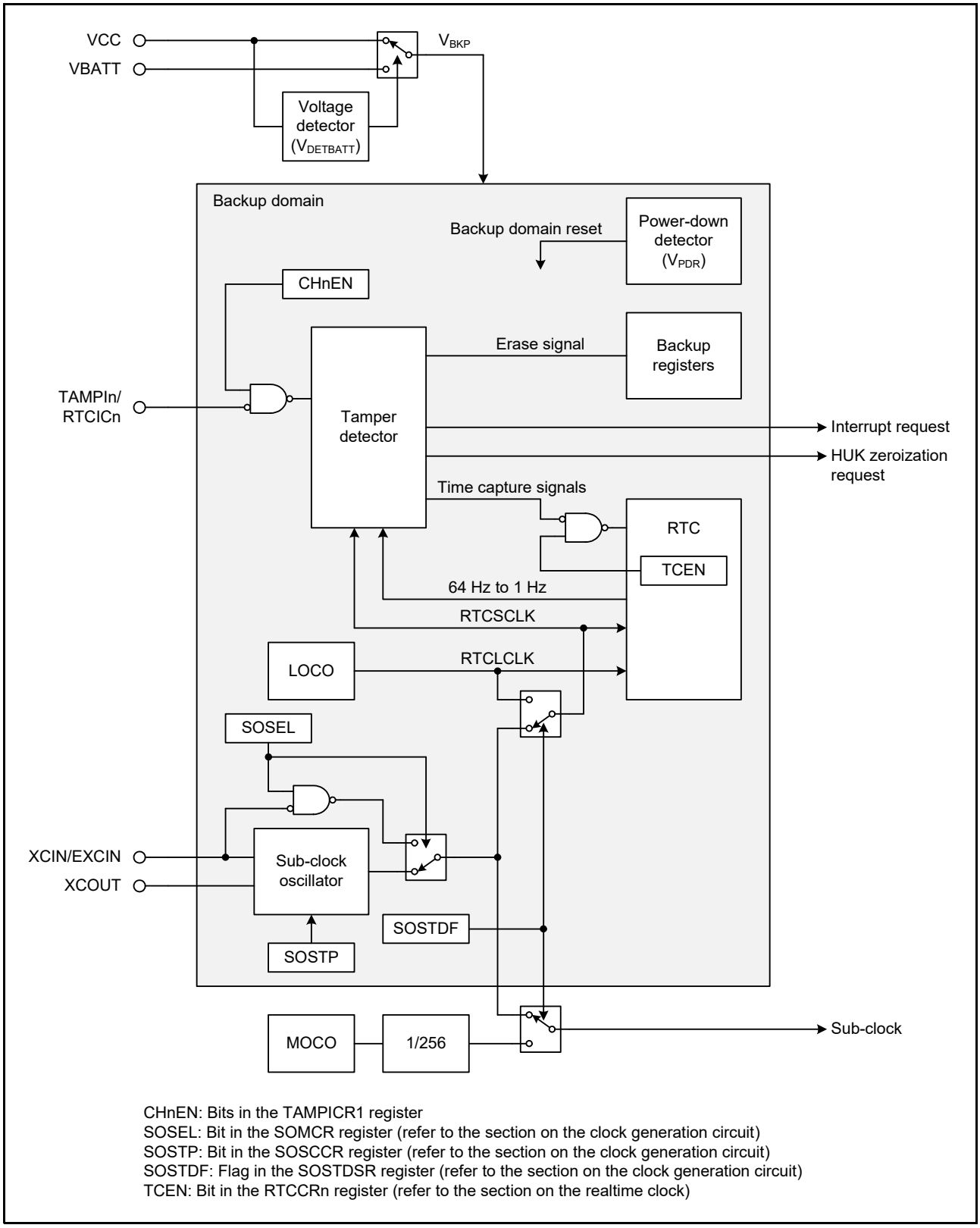


Figure 12.1 Block Diagram of VBATT (n = 0 to 2)

Table 12.2 VBATT Input/Output Pins

Pin Name	I/O	Function
VCC	Input	Power-supply pin
VBATT	Input	Pin for backup battery (power-supply pin)
XCIN	Input	Pins for a crystal for the sub-clock Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
EXCIN	Input	Clock input pin for an external sub-clock oscillator
TAMPI0 to TAMPI2	Input	Tamper input pins
RTCIC0 to RTCIC2	Input	Time capture event input pins for RTC

12.2 Register Descriptions

Registers in the battery backup function (VBATT) can only be initialized by the backup domain reset except for some registers*1. Resets generated by other sources such as the RES# pin or software will not affect these registers.

Note 1. Registers VBSAR, BKRSABR, VBATTMCR, and VBATTCR1

12.2.1 VBATT Security Attribution Register (VBSAR)

Address(es): SYSC.VBSAR 8701 93D0h, SYSC_NS.VBSAR 0701 93D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	SATAM PICR3	SATAM PCR2	SATAM PD	SAVBA TT	SATAM PI	SABKR CR	SAVBA TTMCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAVBATTMCR	VBATTMCR Security Attribution	0: Secure 1: Non-secure	R/W
b1	SABKRCR	BKRCR Security Attribution		R/W
b2	SATAMPI	Tamper Input Security Attribution		R/W
b3	SAVBATT	Battery Backup Security Attribution		R/W
b4	SATAMPD	Tamper Detection Security Attribution		R/W
b5	SATAMPCR2	TAMPCR2 Security Attribution		R/W
b6	SATAMPICR3	TAMPICR3 Security Attribution		R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the SYSC.PRCR_S.PRC4 bit.

This register is used to set the security attributes for the registers in the battery backup function.

SAVBATTMCR Bit (VBATTMCR Security Attribution)

This bit is used to set the security attribute of the VBATTMCR register.

SABKRCR Bit (BKRCR Security Attribution)

This bit is used to set the security attribute of the BKRCR register.

SATAMPI Bit (Tamper Input Security Attribution)

This bit is used to set the security attributes of the tamper/RTCIC input related registers (TAMPICR1, TAMPICR2, TAMPIMR).

SAVBATT Bit (Battery Backup Security Attribution)

This bit is used to set the security attributes of the battery backup related registers (VBATTCR1, VBATTCR2, BKPSR).

SATAMPD Bit (Tamper Detection Security Attribution)

This bit is used to set the security attributes of the tamper detector related registers (TAMPSR, TAMPCR, TCECR).

SATAMPCR2 Bit (TAMPCR2 Security Attribution)

This bit is used to set the security attribute of the TAMPCR2 register.

SATAMPICR3 Bit (TAMPICR3 Security Attribution)

This bit is used to set the security attribute of the TAMPICR3 register.

12.2.2 Backup Register Security Attribute Boundary Register (BKRSABR)

Address(es): SYSC.BKRSABR 8701 93B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR_S.PRC4 bit.

The BKRSABR register is used to specify the boundary for dividing the backup registers (BKRn) into Secure and Non-secure areas.
Set the lower 16 bits of the boundary address in this register. There are five possible values: 9D00h, 9D20h, 9D40h, 9D60h, and 9D80h. The set offset address and the area above it have the Non-secure attribute. For example, if 9D40h is set in this register, addresses 8701 9D00h to 8701 9D3Fh become the Secure area and addresses 0701 9D40h to 0701 9D7Fh become the Non-secure area. If 9D00h is set in this register, the entire area becomes Non-secure area, and if 9D80h is set, the entire area becomes Secure area.

12.2.3 Battery Voltage Measurement Function Control Register (VBATTMCR)

Address(es): SYSC.VBATTMCR 8701 9A84h, SYSC_NS.VBATTMCR 0701 9A84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VBATTME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VBATTME	VBATT Voltage Measurement Enable	0: VBATT voltage measurement is disabled. 1: VBATT voltage measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC3 bit.
Note: This register can be initialized by all resets except the backup domain reset.

The VBATTMCR register is used to control the battery voltage measurement function.

VBATTME Bit (VBATT Voltage Measurement Enable)

This bit is used to measure the voltage input from the VBATT pin with the A/D converter.
After setting this bit to 1, wait for the output voltage to stabilize (t_{MONWT}) before starting A/D conversion. The voltage of VBATT pin is six times of the measurement result.
After completing the measurement, set this bit to 0.

12.2.4 Battery Backup Function Control Register 1 (VBATTCR1)

Address(es): SYSC.VBATTCR1 8701 9A88h, SYSC_NS.VBATTCR1 0701 9A88h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PSWDIS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSWDIS	Power Switch Disable	0: The power switches are enabled. 1: The power switches are disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can be initialized by all resets except the backup domain reset and deep software standby reset.

The VBATTCR1 register is used to control the battery backup function.

PSWDIS Bit (Power Switch Disable)

This bit is used to disable the switches to supply power from the VBATT pin when the main power from the VCC pin is turned off. When this bit is set to 1, the battery backup function is also disabled, and the VBATTCR2.VDETE bit should also be set to 0.

When changing this bit from 1 to 0, make sure that the VDETE bit is set to 0. After setting this bit to 0, wait for the voltage detector to stabilize (t_{DETWT}) before setting the VDETE bit to 1.

12.2.5 Battery Backup Function Control Register 2 (VBATTCR2)

Address(es): SYSC.VBATTCR2 8701 9C45h, SYSC_NS.VBATTCR2 0701 9C45h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VDETE	—	VDETSEL[2:0]		
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	VDETSEL[2:0]	Detection Voltage Select	b2 b0 0 0 0: 2.80 V 0 0 1: 2.53 V 0 1 0: 2.10 V 0 1 1: 1.95 V 1 0 0: 1.85 V 1 0 1: 1.75 V Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	VDETE	Voltage Detection Enable	0: The voltage detector is disabled. 1: The voltage detector is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The VBATTCR2 register is used to control the battery backup function.

VDETSEL[2:0] Bits (Detection Voltage Select)

These bits are used to select the $V_{DETBATT}$ voltage. Set these bits while the VDETE bit is 0.
The $V_{DETBATT}$ voltage should be lower than the voltage detection 0 level (V_{det0}) and higher than the power-on reset voltage (V_{POR}).

VDETE Bit (Voltage Detection Enable)

This bit is used to enable the voltage detector in the battery backup function.
After setting the VDETSEL[2:0] bits, wait for the voltage detector to stabilize (t_{DETWT}) before setting this bit to 1.

12.2.6 Backup Domain Power Status Register (BKPSR)

Address(es): SYSC.BKPSR 8701 9C46h, SYSC_NS.BKPSR 0701 9C46h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSMF	PDRMF	—	—	—	PDDF
Value after reset:	0	0	x	x	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PDDF	Power-Down Detected Flag	0: The voltage for the backup domain has not fallen below V_{PDR}. 1: The voltage for the backup domain has fallen below V_{PDR}.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PDRMF	Backup Domain Reset Monitor Flag	0: The backup domain reset signal is asserted. 1: The backup domain reset signal is negated.	R
b5	PSMF	Power Switch Monitor Flag	0: Power for the backup domain is supplied from the VBATT pin (VCC < $V_{DETBATT}$) 1: Power for the backup domain is supplied from the VCC pin (VCC ≥ $V_{DETBATT}$)	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The BKPSR register indicates the state of the backup domain power supply.

PDDF Flag (Power-Down Detected Flag)

The PDDF flag indicates that the supply voltage for the backup domain has fallen below V_{PDR} and a backup domain reset was generated.

[Setting condition]

- The voltage supplied to the backup domain falling below V_{PDR}.

[Clearing conditions]

- 0 being written to the PDDF flag.

PDRMF Flag (Backup Domain Reset Monitor Flag)

The PDRMF flag indicates the status of the backup domain reset signal. When this flag is 0, the registers in the backup domain are being reset. Wait until this flag becomes 1 before accessing the registers.

PSMF Flag (Power Switch Monitor Flag)

The PSMF flag indicates the state of the power switch (result of comparison between VCC and $V_{DETBATT}$).

12.2.7 Tamper Status Register (TAMPSR)

Address(es): SYSC.TAMPSR 8701 9C48h, SYSC_NS.TAMPSR 0701 9C48h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TAMP2F	TAMP1F	TAMP0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAMP0F	Tamper 0 Event Detection Flag	0: Tamper 0 event has not been detected. 1: Tamper 0 event has been detected.	R/(W) *1
b1	TAMP1F	Tamper 1 Event Detection Flag	0: Tamper 1 event has not been detected. 1: Tamper 1 event has been detected.	R/(W) *1
b2	TAMP2F	Tamper 2 Event Detection Flag	0: Tamper 2 event has not been detected. 1: Tamper 2 event has been detected.	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.
Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that its value is 1 and then write 0 to it.

The TAMPSR register indicates the detection status of the tamper events.

TAMPnF Flag (Tamper n Event Detection Flag) (n = 0 to 2)

Each TAMPnF flag indicates that a trigger (an effective edge) was input to the TAMPIn pin. The trigger is selected by the TAMPICR2.CHnTRG bit.

[Setting condition]

- The trigger set by the TAMPICR2.CHnTRG bit being detected on the TAMPIn pin.

[Clearing conditions]

- 0 being written to the TAMPnF flag after having confirmed that its value is 1.

12.2.8 Tamper Control Register (TAMPCR)

Address(es): SYSC.TAMPCR 8701 9C49h, SYSC_NS.TAMPCR 0701 9C49h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TAMP2EE	TAMP1EE	TAMP0EE	—	TAMP2IE	TAMP1IE	TAMP0IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAMP0IE	Tamper 0 Event Detection Interrupt Enable	0: Tamper 0 event detection interrupt is disabled. 1: Tamper 0 event detection interrupt is enabled.	R/W
b1	TAMP1IE	Tamper 1 Event Detection Interrupt Enable	0: Tamper 1 event detection interrupt is disabled. 1: Tamper 1 event detection interrupt is enabled.	R/W
b2	TAMP2IE	Tamper 2 Event Detection Interrupt Enable	0: Tamper 2 event detection interrupt is disabled. 1: Tamper 2 event detection interrupt is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	TAMP0EE	Tamper 0 Erase Enable	0: Backup registers are not erased in response to a tamper 0 event. 1: Backup registers are erased in response to a tamper 0 event.	R/W
b5	TAMP1EE	Tamper 1 Erase Enable	0: Backup registers are not erased in response to a tamper 1 event. 1: Backup registers are erased in response to a tamper 1 event.	R/W
b6	TAMP2EE	Tamper 2 Erase Enable	0: Backup registers are not erased in response to a tamper 2 event. 1: Backup registers are erased in response to a tamper 2 event.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPCR register controls operations in response to the detection of tampering.

TAMPnIE Bit (Tamper n Event Detection Interrupt Enable) (n = 0 to 2)

The TAMPnIE bit enables or disables the tamper n event detection interrupt.

TAMPnEE Bit (Tamper n Erase Enable) (n = 0 to 2)

The TAMPnEE bit selects whether the backup registers should be or should not be erased in response to a tamper n event.

12.2.9 Tamper Control Register 2 (TAMPCR2)

Address(es): SYSC.TAMPCR2 8701 9C54h, SYSC_NS.TAMPCR2 0701 9C54h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ZHUKE2	ZHUKE1	ZHUKE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ZHUKE0	Tamper 0 Zeroization Request Enable	0: HUK zeroization request triggered by tamper 0 event detection is disabled. 1: HUK zeroization request triggered by tamper 0 event detection is enabled.	R/W
b1	ZHUKE1	Tamper 1 Zeroization Request Enable	0: HUK zeroization request triggered by tamper 1 event detection is disabled. 1: HUK zeroization request triggered by tamper 1 event detection is enabled.	R/W
b2	ZHUKE2	Tamper 2 Zeroization Request Enable	0: HUK zeroization request triggered by tamper 2 event detection is disabled. 1: HUK zeroization request triggered by tamper 2 event detection is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPCR2 register controls operations in response to the detection of tampering.

ZHUKEn Bit (Tamper n Zeroization Request Enable) (n = 0 to 2)

The ZHUKEn bit enables or disables the zeroization request for the hardware unique key (HUK) when a tamper event is detected.

12.2.10 Time Capture Event Control Register (TCECR)

Address(es): SYSC.TCECR 8701 9C4Ah, SYSC_NS.TCECR 0701 9C4Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TCE2S	TCE1S	TCE0S
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCE0S	Time Capture Event 0 Source Select	0: Input signal from the RTCIC0 pin 1: Tamper 0 event	R/W
b1	TCE1S	Time Capture Event 1 Source Select	0: Input signal from the RTCIC1 pin 1: Tamper 1 event	R/W
b2	TCE2S	Time Capture Event 2 Source Select	0: Input signal from the RTCIC2 pin 1: Tamper 2 event	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TCECR register selects the sources for time capture events of the RTC.

TCEnS Bit (Time Capture Event n Source Select) (n = 0 to 2)

The TCEnS bit selects the source of time capture event n for the RTC.
When this bit is set to 0, a signal input to the RTCICn pin is input to the RTC. Selection of this setting is recommended when tamper event detection is not to be used.
When this bit is set to 1, the tamper n event detection signal is input to the RTC. Selection of this setting is recommended when a tamper event is to cause a timestamp to be recorded.

12.2.11 Tamper/RTCIC Input Control Register 1 (TAMPICR1)

Address(es): SYSC.TAMPICR1 8701 9C4Ch, SYSC_NS.TAMPICR1 0701 9C4Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CH2EN	CH1EN	CH0EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH0EN	Channel 0 Input Enable	0: The TAMPI0/RTCIC0 signal input is disabled. 1: The TAMPI0/RTCIC0 signal input is enabled.	R/W
b1	CH1EN	Channel 1 Input Enable	0: The TAMPI1/RTCIC1 signal input is disabled. 1: The TAMPI1/RTCIC1 signal input is enabled.	R/W
b2	CH2EN	Channel 2 Input Enable	0: The TAMPI2/RTCIC2 signal input is disabled. 1: The TAMPI2/RTCIC2 signal input is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPICR1 register enables or disables input from the TAMPIn/RTCICn pins (n = 0 to 2).

CHnEN Bit (Channel n Input Enable) (n = 0 to 2)

The CHnEN bit enables or disables the input from the TAMPIn/RTCICn pin.

Set each bit to 1 not only if tamper events are to be detected but also if the RTC time capture function is to be used. After setting this bit to 1, it requires up to 50 µs for the output from the input buffer to stabilize. Wait at least 50 µs before using the TAMPIn/RTCICn signal.

12.2.12 Tamper/RTCIC Input Control Register 2 (TAMPICR2)

Address(es): SYSC.TAMPICR2 8701 9C4Dh, SYSC_NS.TAMPICR2 0701 9C4Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CH2TRG	CH1TRG	CH0TRG	—	CH2NFE	CH1NFE	CH0NFE
Value after reset:	0	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH0NFE	Channel 0 Noise Filter Enable	0: Noise filter for the TAMPI0/RTCIC0 pin is disabled. 1: Noise filter for the TAMPI0/RTCIC0 pin is enabled.	R/W
b1	CH1NFE	Channel 1 Noise Filter Enable	0: Noise filter for the TAMPI1/RTCIC1 pin is disabled. 1: Noise filter for the TAMPI1/RTCIC1 pin is enabled.	R/W
b2	CH2NFE	Channel 2 Noise Filter Enable	0: Noise filter for the TAMPI2/RTCIC2 pin is disabled. 1: Noise filter for the TAMPI2/RTCIC2 pin is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CH0TRG	Channel 0 Trigger Select	Select the trigger for tamper 0 event detection. 0: A falling edge of the input on the TAMPI0 pin 1: A rising edge of the input on the TAMPI0 pin	R/W
b5	CH1TRG	Channel 1 Trigger Select	Select the trigger for tamper 1 event detection. 0: A falling edge of the input on the TAMPI1 pin 1: A rising edge of the input on the TAMPI1 pin	R/W
b6	CH2TRG	Channel 2 Trigger Select	Select the trigger for tamper 2 event detection. 0: A falling edge of the input on the TAMPI2 pin 1: A rising edge of the input on the TAMPI2 pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPICR2 register is used to enable or disable the noise filters for the TAMPIn/RTCICn pins (n = 0 to 2) and select the triggers for tamper event detection.

CHnNFE Bit (Channel n Noise Filter Enable) (n = 0 to 2)

The CHnNFE bit enables or disables the noise filter for the TAMPIn/RTCICn pin. When tamper n event detection is not to be used, set the bit to 0.

Five cycles of the clock selected by the TAMPICR3.NFCS[2:0] bits are necessary for the output of the noise filter to become stable after this bit has been set to 1.

CHnTRG Bit (Channel n Trigger Select) (n = 0 to 2)

The CHnTRG bit is used to select the effective edge for the input signal from the TAMPIn pin for use as a trigger of tamper event detection.

12.2.13 Tamper/RTCIC Input Control Register 3 (TAMPICR3)

Address(es): SYSC.TAMPICR3 8701 9C50h, SYSC_NS.TAMPICR3 0701 9C50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	b2 b0 0 0 0: 32.768 kHz 0 0 1: 64 Hz 0 1 0: 32 Hz 0 1 1: 16 Hz 1 0 0: 8 Hz 1 0 1: 4 Hz 1 1 0: 2 Hz 1 1 1: 1 Hz	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPICR3 register is used to control the noise filters for the TAMPIn/RTCICn pins (n = 0 to 2).

NFCS[2:0] Bits (Noise Filter Clock Select)

The NFCS[2:0] bits are used to select the sampling frequency of the noise filters. When setting a value other than 000b, set the RTC.RCR2.START bit to 1.
Set these bits while the TAMPICR2.CHnNFE bits are all 0 (n = 0 to 2).

12.2.14 Tamper/RTCIC Input Monitoring Register (TAMPIMR)

Address(es): SYSC.TAMPIMR 8701 9C4Eh, SYSC_NS.TAMPIMR 0701 9C4Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CH2LVL	CH1LVL	CH0LVL
Value after reset:	0	0	0	0	0	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b0	CH0LVL	Channel 0 Level Monitoring Flag	0: The low level is being input on the TAMPI0 pin. 1: The high level is being input on the TAMPI0 pin.	R
b1	CH1LVL	Channel 1 Level Monitoring Flag	0: The low level is being input on the TAMPI1 pin. 1: The high level is being input on the TAMPI1 pin.	R
b2	CH2LVL	Channel 2 Level Monitoring Flag	0: The low level is being input on the TAMPI2 pin. 1: The high level is being input on the TAMPI2 pin.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPIMR register is used to monitor the input levels on the TAMPIn pins (n = 0 to 2).

CHnLVL Flag (Channel n Level Monitoring Flag) (n = 0 to 2)

Each CHnLVL flag indicates the input level on an TAMPIn pin.

12.2.15 Backup Register Control Register (BKRCR)

Address(es): SYSC.BKRCR 8701 9C40h, SYSC_NS.BKRCR 0701 9C40h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BKRAE	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	BKRAE	Backup Register Access Enable	0: Access to the BKRn registers is disabled. 1: Access to the BKRn registers is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register is write-protected by the SYSC.PRCR.PRC1 bit.
Note: This register is only initialized when switching to battery backup mode or by a power-on reset. During battery backup mode, the BKRAE bit is reset to 1, but the internal state retains the value set before entering battery backup mode.

The BKRCR register is used to control access to the backup registers.

BKRAE Bit (Backup Register Access Enable)

This bit enables access to the BKRn registers. When reading from or writing to the BKRn registers, wait at least 500 ns after setting this bit to 1. After access to the BKRn registers is no longer required, set this bit to 0. If the mode is switched to battery backup mode with this bit remains 1, the contents of the BKRn registers are not guaranteed.

12.2.16 Backup Register n (BKRn) (n = 0 to 127)

Address(es): SYSC.BKR0 8701 9D00h to SYSC.BKR127 8701 9D7Fh, SYSC_NS.BKR0 0701 9D00h to SYSC_NS.BKR127 0701 9D7Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Note: When these registers are set to the secure attribute, they are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When these registers are set to the non-secure attribute, they are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: These registers are write-protected by the SYSC.PRCR.PRC1 bit.
Note: These registers can only be initialized by the backup domain reset. They are not affected by the MCU reset.

The contents of the BKRn registers are retained even in the battery backup mode.
The BKRn registers can be erased when a tamper event is detected.

12.3 Operation

12.3.1 Battery Backup Function

This function is used to retain the operations of the modules in the backup domain by supplying power from the VBATT pin (battery backup mode) while the main power supply is turned off.
If the VCC voltage falls below $V_{DETBATT}$, the backup domain power supply is automatically switched to VBATT. When the VCC voltage rises back to $V_{DETBATT}$, the backup domain power supply is automatically switched back to VCC.
When the battery backup function is to be used, enable the voltage monitor 0 reset.
Figure 12.2 shows the structure of the power switches and Figure 12.3 shows switching of the power supply for the backup domain.

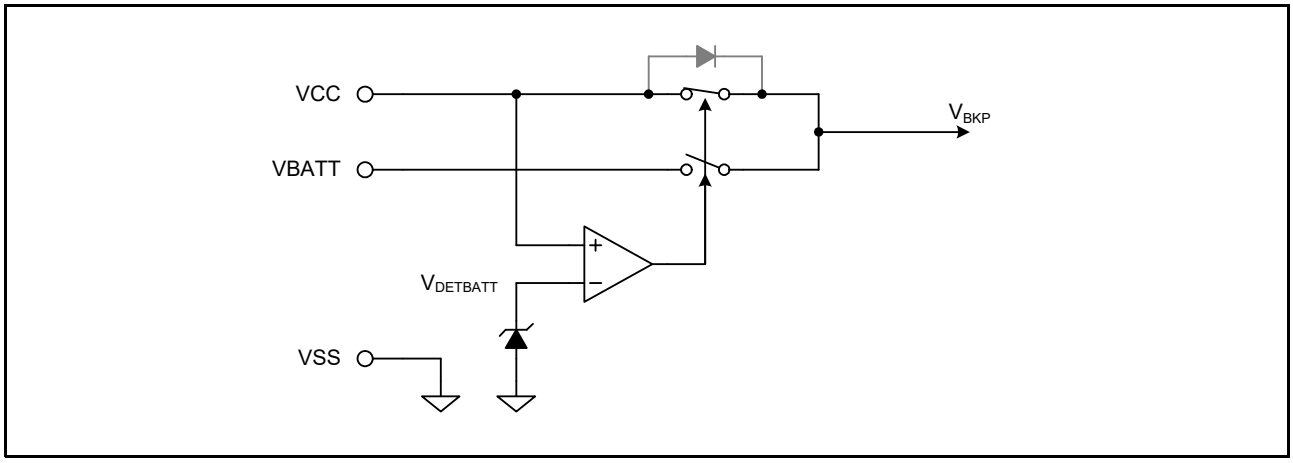


Figure 12.2 Power Switches

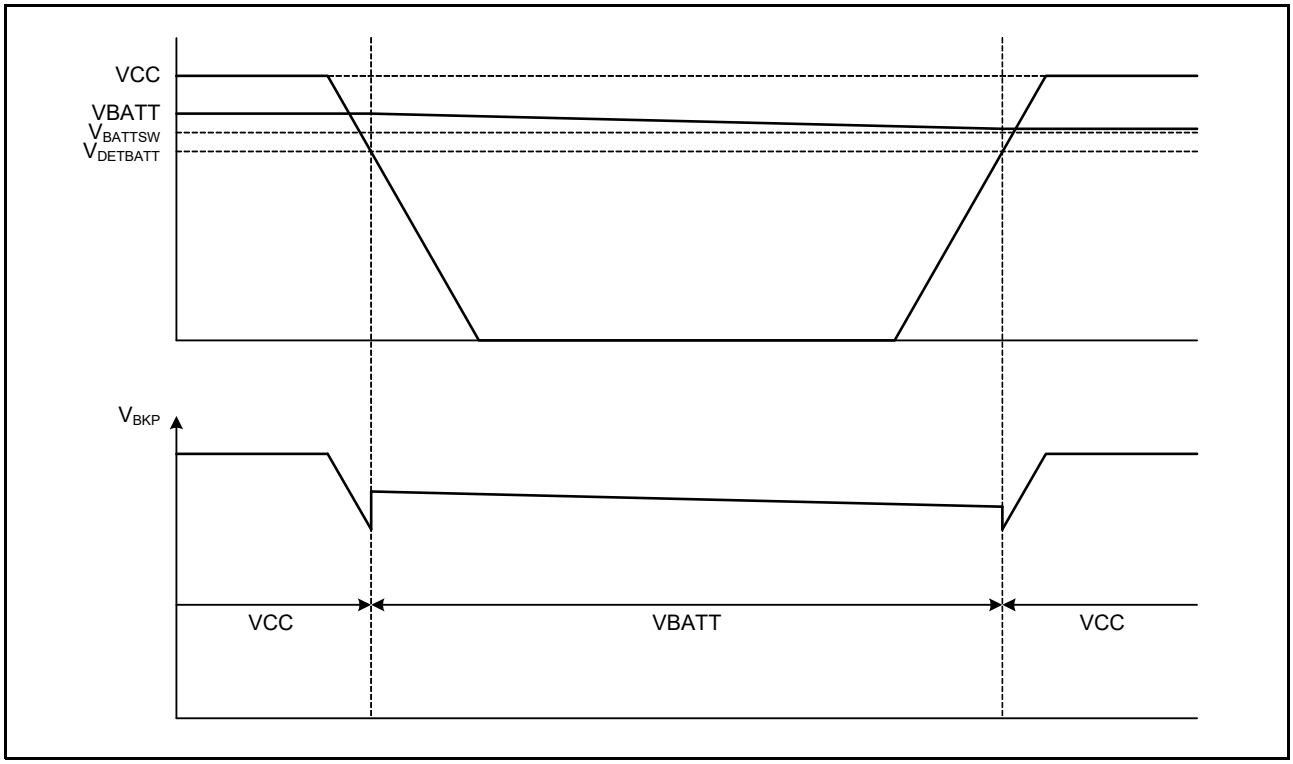


Figure 12.3 Switching of the Power Supply for the Backup Domain

Setting the OFSM.OFS1.DSBYLPSEL bit to 0 can further reduce the power consumption during deep software standby mode 1 or 2. When the DSBYLPSEL bit is set to 0, the voltage detector in the battery backup function does not operate

during deep software standby mode 1 or 2, and instead the voltage detection 0 circuit (PVD0) controls the power supply switch. In this case, the detection voltage changes from $V_{DETBATT}$ to V_{det0}. Set the V_{det0} voltage using the OFSM.OFS1.VDSEL[2:0] bits.

The following modules are in the backup domain.

- Sub-clock oscillator
- Low-speed on-chip oscillator (LOCO)
- Power-down detector
- Backup registers
- Tamper detector
- RTC

The VBATT pin only supplies power for the following pins in the battery backup mode.

- The XCIN and XCOUT pins for a crystal resonator
- The EXCIN pin for an external crystal oscillator
- The TAMPI0, TAMPI1, and TAMPI2 pins for tamper event detection

When using the battery backup function, configure each register according to the procedure shown in Figure 12.4 after a power-on reset.

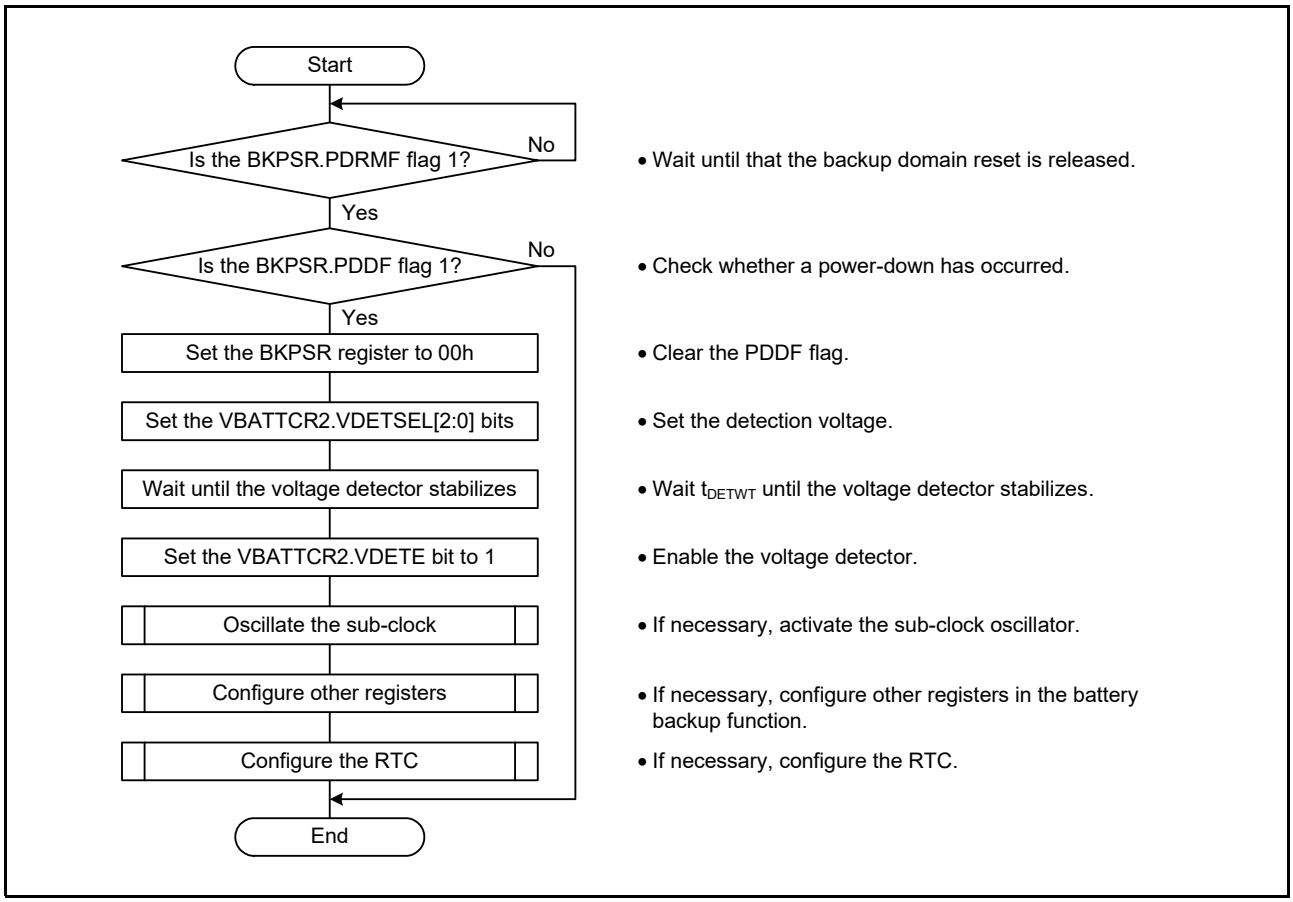


Figure 12.4 Example of the Configuration Flow after Power-On Reset

12.3.2 Power-Down Detector

This module monitors the supply voltage to the backup domain (V_{BKP}) and generates a backup domain reset signal. Specifically, the backup domain reset signal is asserted when the V_{BKP} voltage falls below $V_{PDR(BKP)}$. When the V_{BKP} voltage rises above $V_{PDR(BKP)}$, the reset signal is negated.

A backup domain reset only affects the registers described in this chapter and the SOSCCR, SOSTDCR, SOSTDSR, and SOMCR registers described in the chapter on the clock generation circuit. However, the reset does not affect the registers and counters in the RTC.

The value of the BKPSR.PDDF flag is 1 when a backup domain reset had been generated. If the BKPSR.PDDF flag is 1, set all registers in the backup domain again.

Figure 12.5 shows the structure of the power-down detector and Figure 12.6 shows its operation.

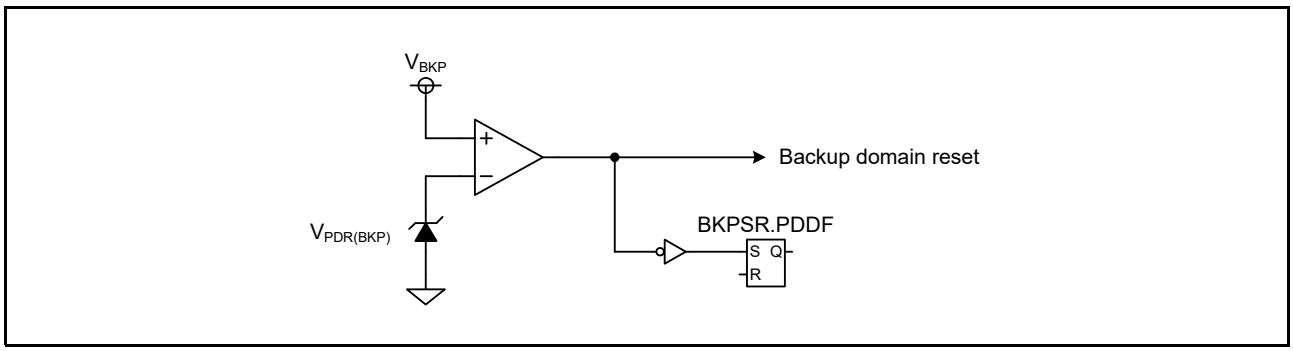


Figure 12.5 Structure of the Power-Down Detector

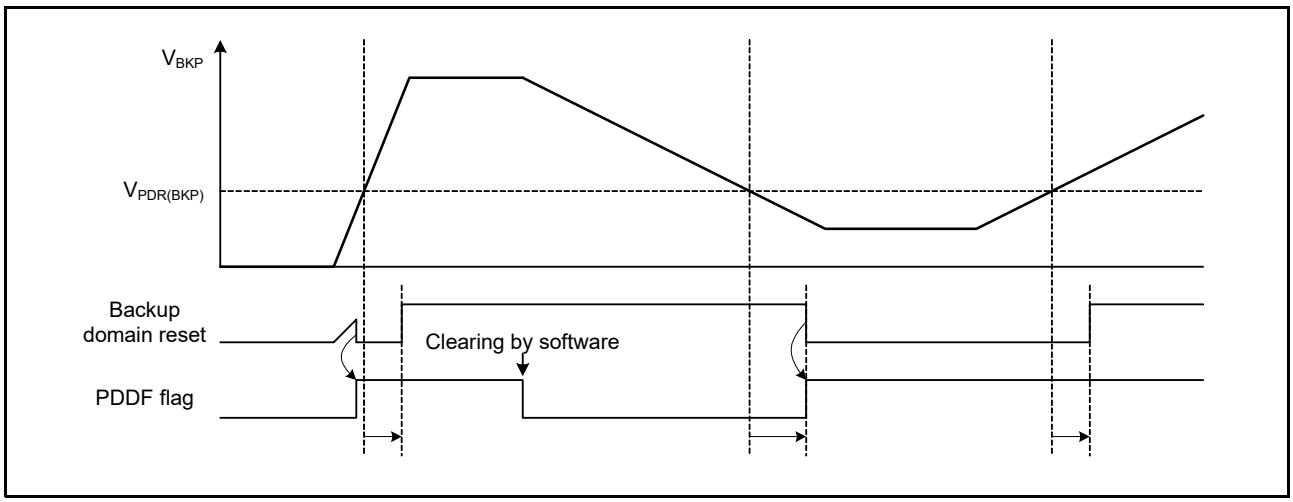


Figure 12.6 Backup Domain Reset Signal

12.3.3 Backup Registers

This MCU has 128 bytes of backup registers (BKRn (n = 0 to 127)) to save data that should not be lost even if the VCC power supply is lost.

Protection for the BKRn registers becomes effective 250 ns after the BKRCR.BKRAE bit is set to 0. To transition to deep software standby mode after saving data to the BKRn registers, set the BKRAE bit to 0, wait at least 250 ns, and then execute the WAIT instruction.

When the TAMPSR.TAMPnF flag is set to 1 while the TAMPCR.TAMPnEE bit is 1 (Backup registers are erased in response to a tamper n event), erasing the backup registers begins (n = 0 to 2). To stop erasing the backup registers, wait at least 100 ns after the TAMPnF flag is set to 1, then set the TAMPnEE bit or the TAMPnF flag to 0. Wait at least 500 ns before accessing the backup registers again.

12.3.4 Tamper Detector

This MCU has three tamper pins (TAMPI0 to TAMPI2) which remain functional in the battery backup mode and can detect physical intrusions.
The tamper inputs can be configured for rising or falling edge detection and with or without noise filtering.
The following operations can be selected in response to the detection of tampering.

- Generating an interrupt request
- Erasing the backup registers
- Zeroization of a hardware unique key (HUK)
- Generating RTC timestamp events

Figure 12.7 is a block diagram of the tamper detector.

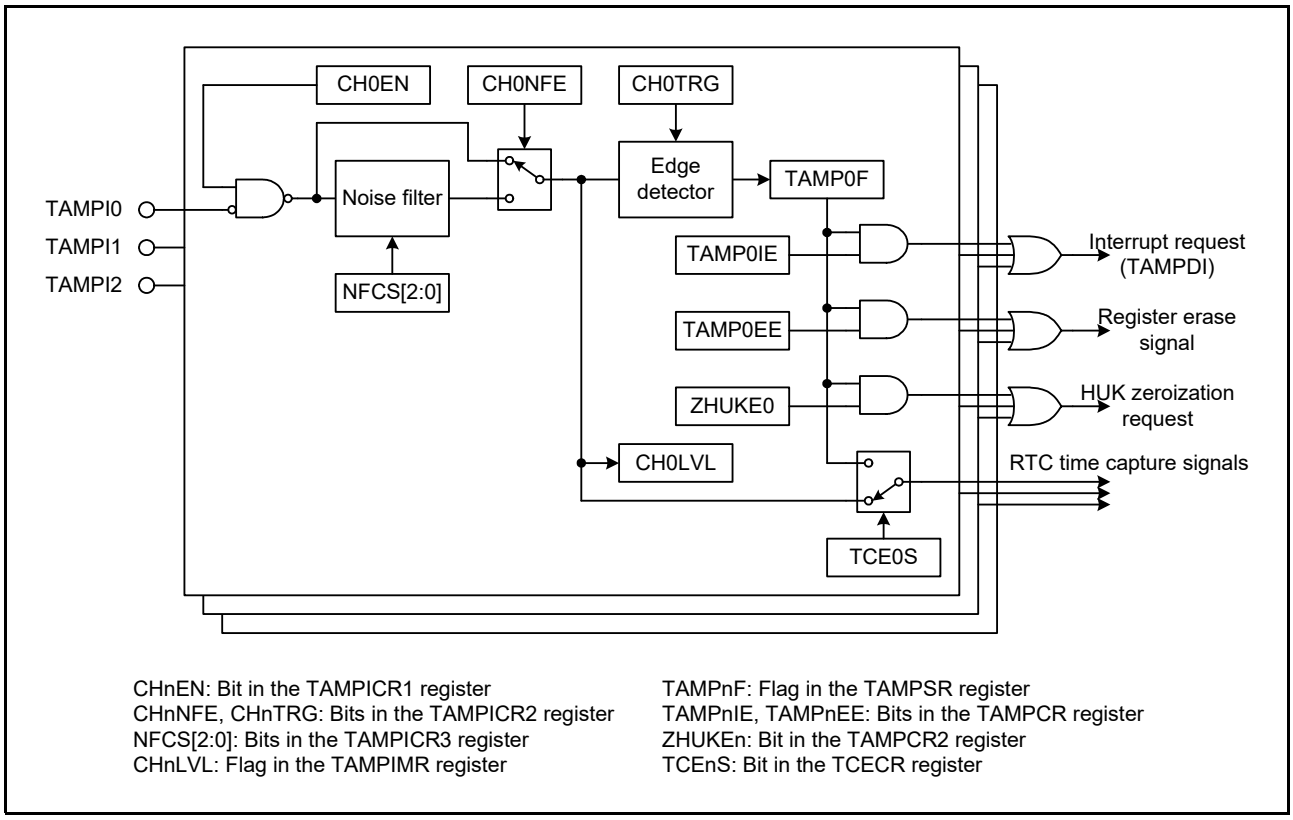


Figure 12.7 Tamper detector (n = 0 to 2)

If the value of the TAMPICR1.CHnEN bit or the CHnNFE or CHnTRG bit in the TAMPICR2 register is changed, a TAMPSR.TAMPnF flag may become 1 (n = 0 to 2). Set the TAMPnIE and TAMPnEE bits in the TAMPCR register and the TAMPCR2.ZHUKEn bit to 0 before setting these bits. Clear the TAMPnF flags after setting these bits.
The TAMPIMR.CHnLVL flags are for monitoring the levels being input on the TAMPIn pins.
When an effective edge is detected in any of the channels, the corresponding TAMPnF flag becomes 1. If the TAMPnIE bit is 1 at this point, the TAMPDI interrupt is generated. If the TAMPnEE bit is 1, the backup registers are erased to 00h. If the ZHUKEn bit is 1, the HUK zeroization request is output.
Setting the TCECR.TCEnS bit to 1 enables the time capture function of the RTC to record a timestamp when a tamper event is detected. For details on the time capture function, refer to section 30, Realtime Clock (RTCBb).

Figure 12.8 shows the setting flow when using the tamper detector.

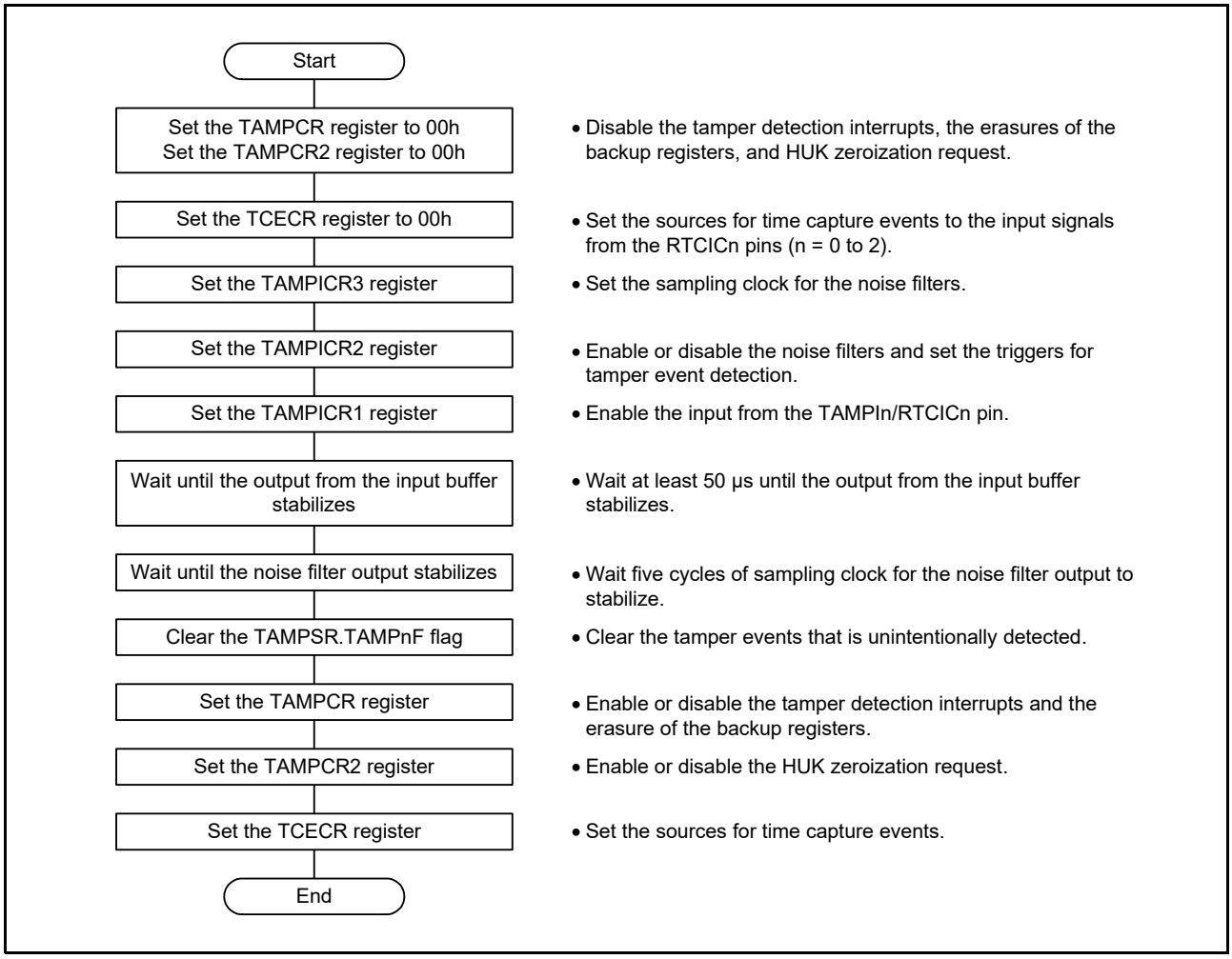


Figure 12.8 Setting Flow for Tamper Detector

12.4 Interrupt

The VBATT has the three interrupt sources listed in Table 12.3.

Table 12.3 VBATT Interrupt Sources

Name	Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit
Tamper event detection interrupt	TAMPDI	Tamper 0 event detection	TAMP0F	TAMP0IE
		Tamper 1 event detection	TAMP1F	TAMP1IE
		Tamper 2 event detection	TAMP2F	TAMP2IE

12.5 Usage Notes

12.5.1 Voltage Monitor 0 Reset

When the battery backup function is to be used, enable the voltage monitor 0 reset (OFSM.OFS1.PVDAS bit = 0).

12.5.2 When the Battery Backup Function is Not to be Used

If the battery backup function is not to be used, externally connect the VBATT pin to the VCC pin. Also, set the VBATTCR1.PSWDIS bit to 1 and the VBATTCR2.VDETE bit to 0.
With this configuration, a backup domain reset may not be generated in cases such as when the VCC voltage drops and then recovers immediately. Initialize the registers using the procedure shown in Figure 12.9.

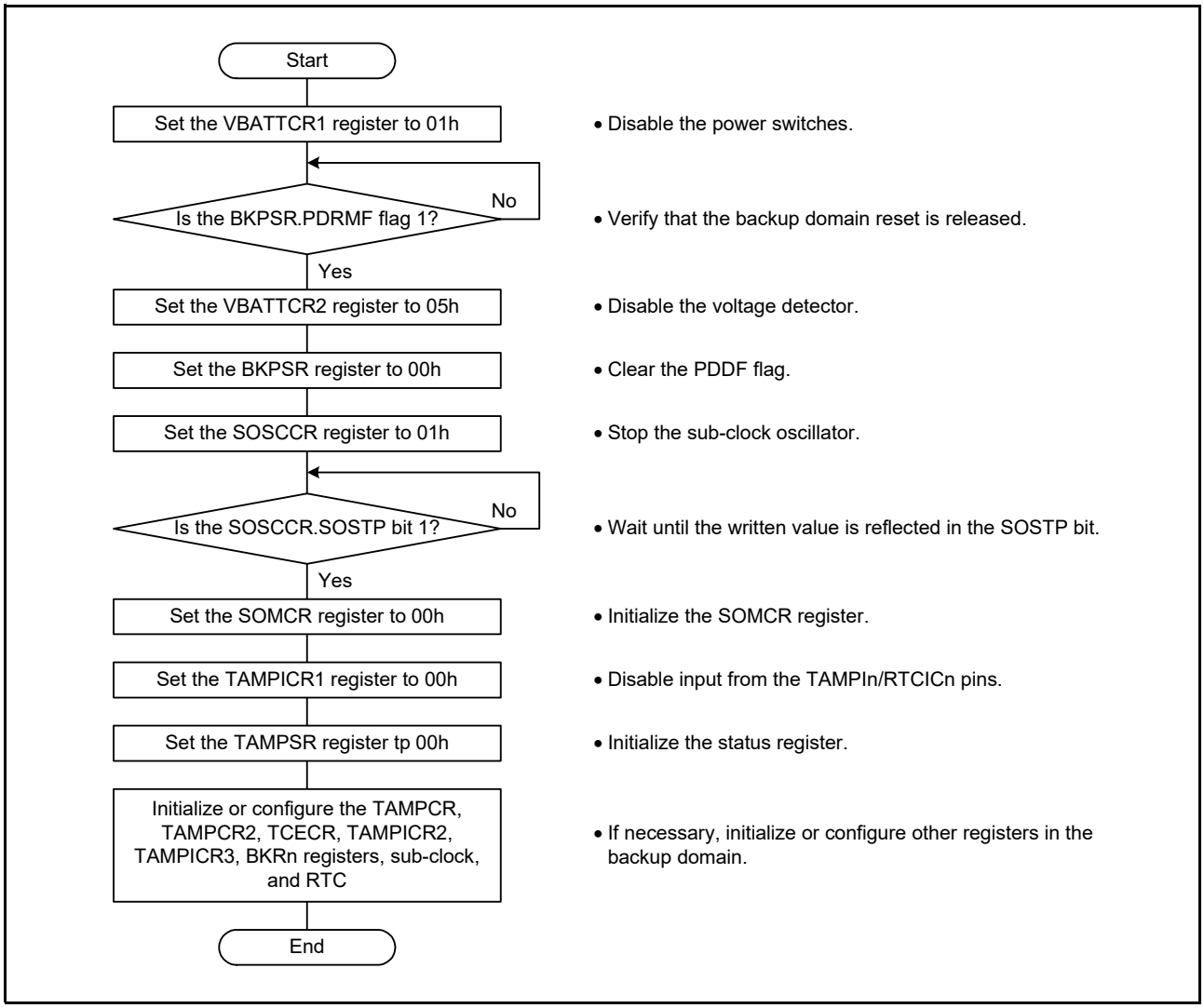


Figure 12.9 Example of Register Initialization Flow

12.5.3 Current Injection Flowing into the VBATT Pin

During operation in the battery backup mode, the voltage of the VCC pin being higher than V_{BATT} may inject a current into the VBATT pin through the parasitic diode connected to the power switch (refer to Figure 12.2). If the power supply or battery connected to the VBATT pin cannot support this current injection, it is recommended that an external low-drop diode be connected between this power supply and the VBATT pin.

13. Register Write Protection Function

The register write protection function protects the specified registers from being overwritten in case a program runs out of control. The registers to be protected can be specified by using the protect registers (PRCR_S and PRCR_NS).
The PRCR_S register is used to protect the registers set as Secure, and the PRCR_NS register is used to protect the registers set as Non-secure. For the sake of descriptions in this section, these two registers are collectively referred to as the PRCR registers.
Table 13.1 lists the association between the PRCR register and the registers to be protected.

Table 13.1 Association between PRCR Register and Registers to be Protected

PRCR	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, HOCOCR2, MOCOCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, SOSTDCR, SOSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, EBCKOCR, SDCKOCR, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, ADCCKDIVCR, ADCCKCR, GPTCKDIVCR, GPTCKCR, BCKADIVCR, ESWCKDIVCR, ESWPCKDIVCR, ETHPCKDIVCR, BCKACR, ESWCKCR, ESWPCKCR, ETHPCKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, XSPICKDIVCR, CANFDCKDIVCR, USB60CKDIVCR, I3CCKDIVCR, USBCKCR, XSPICKCR, CANFDCKCR, USB60CKCR, I3CCKCR, MOSCSCR, HOCOSCR, MOCOSCR, MOSCWTCR, LOCOCR, LOCOUTCR, MOMCR, SOSCCR, SOMCR
PRC1	• Registers related to the operating modes: SYSCR0 • Registers related to the low power consumption functions: PGSCR, PSSTCR2, PSSTCR4, PSSTCR5, PDCTRESWM, PDRAMSCR0, PDRAMSCR2, SBYCR, SSCR1, LPSCR, DPSBYCR, DPSIERn (n = 0 to 5), DPSIFRn (n = 0 to 5), DPSIEGRn (n = 0 to 4), PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR, VSCR, SVSCR, MWMCR, LDOECR, DCCR • Registers related to the battery backup function: BKRCR, TAMPICR1, BKRn (n = 0 to 127), VBATTCR1, VBATTCR2, BKPSR, TAMPSR, TAMPCR, TCECR, TAMPICR2, TAMPCR2, TAMPICR3 • Software reset register: SWRR
PRC3	• Registers related to the PVD: PVD1CMPCR, PVD2CMPCR, PVD4CMPCR, PVD5CMPCR, PVD1CR0, PVD2CR0, PVD4CR0, PVD5CR0, PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1FCR, PVD2FCR, PVD4FCR, PVD5FCR, PVDLR, VBATTMCR
PRC4*1	• Registers related to the security function: ELCSARx (x = A to D), PSARx (x = B to F), MSSAR, PmSAR (m = 0 to 9 or A to G), CACHESAR, IFUSR, LRMSAR, LRMSABARx (x = 0, 1), ICUSARx (x = A to D), ICU0SARx (x = 0 to 7), TEVTRCR, ICUNSACR, PRTZCR, BUSSARx (x = A to C), MMPUSARB, DMACCHSAR, DMACSAR, EXDMACCHSAR, EXDMACSAR, DTCSAR, SRAMSABARx (x = 0, 1), SRAMSAR, MSAR, CGFSAR, RSTSAR, LPMSAR, PVDSAR, VBSAR, DPFSAR, DPFSAR1, RSCSAR, PGCSAR, BKRSABR
PRC5*1	• Registers related to the resets: SYRSTMSK0, SYRSTMSK2, TEMPRCR, TEMPRLR

Note 1. This bit is only present in the PRCR_S register.

13.1 Register Descriptions

13.1.1 Protect Register (PRCR_S)

Address(es): SYSC.PRCR_S 8701 93FAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	PRC5	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption, battery backup, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the PVD. 0: Write disabled 1: Write enabled	R/W
b4	PRC4	Protect Bit 4	Enables writing to the registers related to the security function. 0: Write disabled 1: Write enabled	R/W
b5	PRC5	Protect Bit 5	Enables writing to the registers related to the resets. 0: Write disabled 1: Write enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR_S register. To modify the PRCR_S register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Written values are not retained. These bits are read as 00h.

The PRCR_S register is used to protect the registers set as Secure.

PRCi Bits (Protect Bit i) (i = 0, 1, 3, 4, 5)

These bits enable or disable writing to the corresponding registers to be protected.
Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.
In case of successive write access to the PRCR_S register and to registers controlled by the PRCR_S register, the registers under control of the PRC4 bit may not reflect a change in the PRC4 bit. Avoid successive write access, or after having written a new value to the PRC4 bit, read the PRCR_S register before writing to the registers under control of the PRC4 bit.

13.1.2 Protect Register (PRCR_NS)

Address(es): SYSC_NS.PRCR_NS 0701 93FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption, battery backup, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the PVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR_NS register. To modify the PRCR_NS register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Written values are not retained. These bits are read as 00h.

The PRCR_NS register is used to protect the registers set as Non-secure.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.
Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.

14. Exception Handling

14.1 Exception Events

During the execution of a program by the CPU, the occurrence of a certain event may cause execution of the program to be suspended and execution of another program to be started. Such events are called exception events. The RXv3e CPU supports ten types of exception. The types of exception events are shown in Figure 14.1. The occurrence of an exception causes the processor mode to shift to supervisor mode.

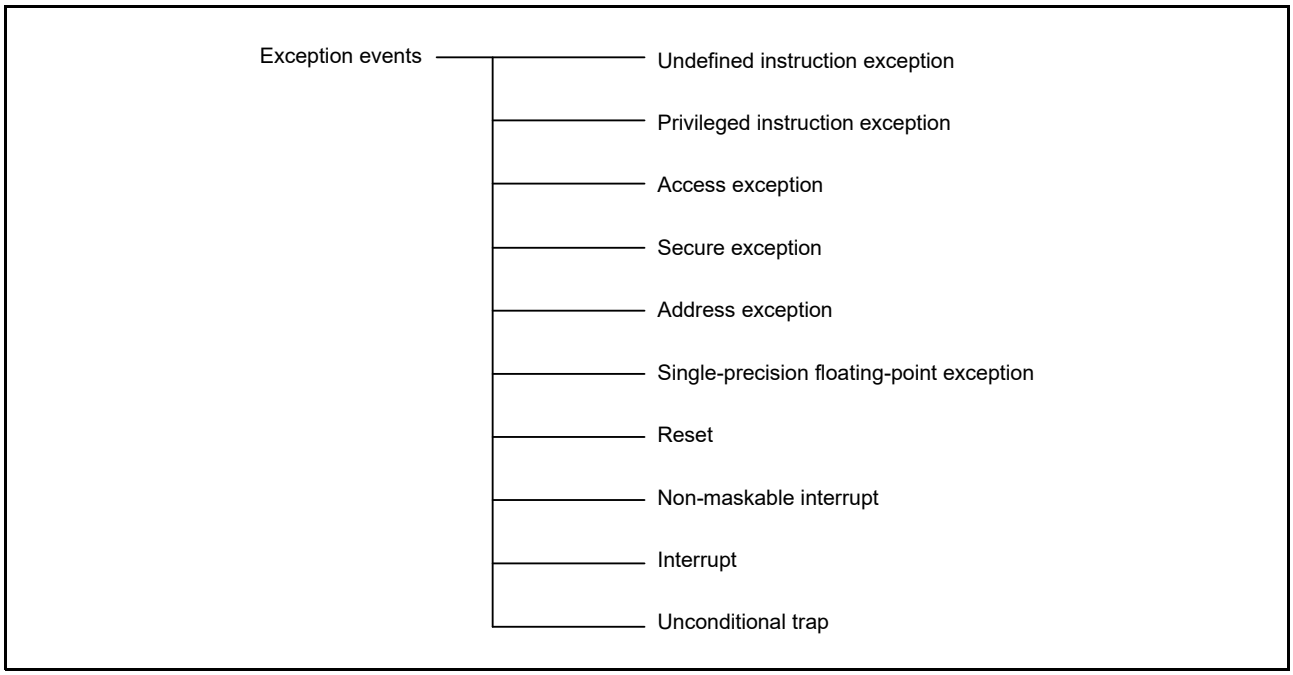


Figure 14.1 Types of Exception Events

14.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected. This exception also occurs in the case of an attempt to execute an instruction that is exclusively for use in the root world (TRMVTC, TRMVFC, or TRRTE) from outside the root world.

14.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

14.1.3 Access Exception

An access exception occurs when the memory protection unit detects an error in memory access by the CPU. If the error is due to instruction fetching, an instruction access exception occurs, and if the error is due to operand access, an operand access exception occurs.

14.1.4 Secure Exception

A secure exception occurs when a security violation is detected in memory access by the CPU. If an error occurs due to an instruction fetch, an instruction secure exception occurs, and if an error occurs due to an operand access, an operand secure exception occurs.
The CPU core is placed in the root world on generation of a secure exception.
Even when a security violation is detected in memory access, the CPU may not generate a secure exception. This depends on the location and timing of the detection.

14.1.5 Address Exception

An address exception is generated in response to 64-bit operand access to an address that is not multiples of four.

14.1.6 Single-Precision Floating-Point Exception

A single-precision floating-point exception is generated when any of the five exceptions specified in the IEEE 754 standard, namely overflow, underflow, inexact, division-by-zero, or invalid operation, or an attempt to use processing that is not implemented, is detected upon execution of a single-precision floating-point operation instruction. Exception handling by the CPU only proceeds when any among the EX, EU, EZ, EO, or EV bits in the FPSW, which corresponding to the five types of exception, is set to 1.

14.1.7 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

14.1.8 Non-Maskable Interrupt

Input of the non-maskable interrupt signal to the CPU generates a non-maskable interrupt. This interrupt is assumed to be used when a fatal problem with the system is judged to have arisen. In such cases, use it under the condition that return to the original program being executed when the exception was generated will not proceed after the exception handling routine. In addition, unless a Flat-Project is in use, returning to the original program being executed when the exception was generated is not possible.
Non-maskable interrupts are accepted independently of the type of the world that is currently in use.
The behavior upon acceptance of a non-maskable interrupt is divided into the following two cases according to the situation at the time of acceptance.

	Situation on Acceptance of a Non-Maskable Interrupt	World Transition
Normal acceptance	The ID of the destination world for a non-maskable interrupt is the same as the ID of the current world.	No world transition
World-transition acceptance	The ID of the destination world for a non-maskable interrupt is different from the ID of the current world.	Transition to the root world occurs

In the case of normal acceptance, a transition from the current world to another world does not occur.
In the case of world-transition acceptance, a transition to the root world occurs. After that, a transition to the destination world of the non-maskable interrupt is produced by software running in the root world.

14.1.9 Interrupt

An interrupt is generated by the input of an interrupt signal to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 31 (the highest).

An interrupt can only be accepted when the conditions in the table below are met. The conditions for acceptance depend on the current world.

Table 14.1 Conditions of Interrupt Acceptance

Current World		
Root	**Secure**	**Non-secure**
• ITW is root and • PSW.I is 1 and • (PSW.IPL < IPR) and • (WPRI.WIPL < IPR)	• PSW.I is 1 and • (PSW.IPL < IPR) and • (WPRI.WIPL < IPR)	• PSW.I is 1 and • (PSW.IPL < IPR) and • (WPRI.WIPL < IPR) However, when the IVC.NSIC bit is 1, an interrupt will even be accepted under the following condition in addition to the above conditions. • (15 < IPR) and • (WPRI.WIPL < IPR)

IPR: The priority level of a requested interrupt
ITW: The destination world ID for a requested interrupt

The behavior upon acceptance of an interrupt is divided into the following two cases according to the situation at the time of acceptance.

	Situation on Acceptance of Interrupt	World Transition
Normal acceptance	The ID of the destination world for an interrupt is the same as the ID of the current world.	No world transition
World-transition acceptance	The ID of the destination world for an interrupt is different from the ID of the current world.	Transition to the root world occurs

In the case of normal acceptance, a transition from the current world to another world does not occur. In the case of world-transition acceptance, a transition to the root world occurs. After that, a transition to the destination world of the interrupt is produced by software running in the root world. Interrupt priority levels 1 to 31 can be used for interrupts having root or secure as the destination world. Interrupt priority levels 1 to 15 can be used for interrupts having non-secure as the destination world. Fast interrupts are only usable in Flat-Projects.

14.1.10 Unconditional Trap

An unconditional trap is generated when the INT, BRK, or TRCALL instruction is executed. The behavior upon acceptance depends on the instruction and is divided into the following two cases.

	Instructions	World Transition
Normal acceptance	INT, BRK	No world transition
World-transition acceptance	TRCALL	Transition to the root world occurs

14.2 Exception Handling Procedure

Exception handling procedures are of two types. In one, the exception event is processed without world transitions. In the other, the exception event is processed with world transitions.
Table 14.2 shows which exception events generate world transitions.

Table 14.2 Exception Events and World Transition

Exception Event	Presence of World Transitions
Undefined instruction exception	No world transitions
Privileged instruction exception	No world transitions
Access exception	No world transitions
Secure exception	World transitions
Address exception	No world transitions
Single-precision floating-point exception	No world transitions
Reset	— (always causes a transition to the root world)
Non-maskable interrupt (with normal acceptance)	No world transitions
Non-maskable interrupt (with world-transition acceptance)	World transitions
Interrupt (with normal acceptance)	No world transitions
Interrupt (with world-transition acceptance)	World transitions
Unconditional trap (with normal acceptance)	No world transitions
Unconditional trap (with world-transition acceptance)	World transitions

14.2.1 Exception Handling Procedure without World Transitions

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 14.2 shows the processing procedure when an exception other than a reset is accepted.

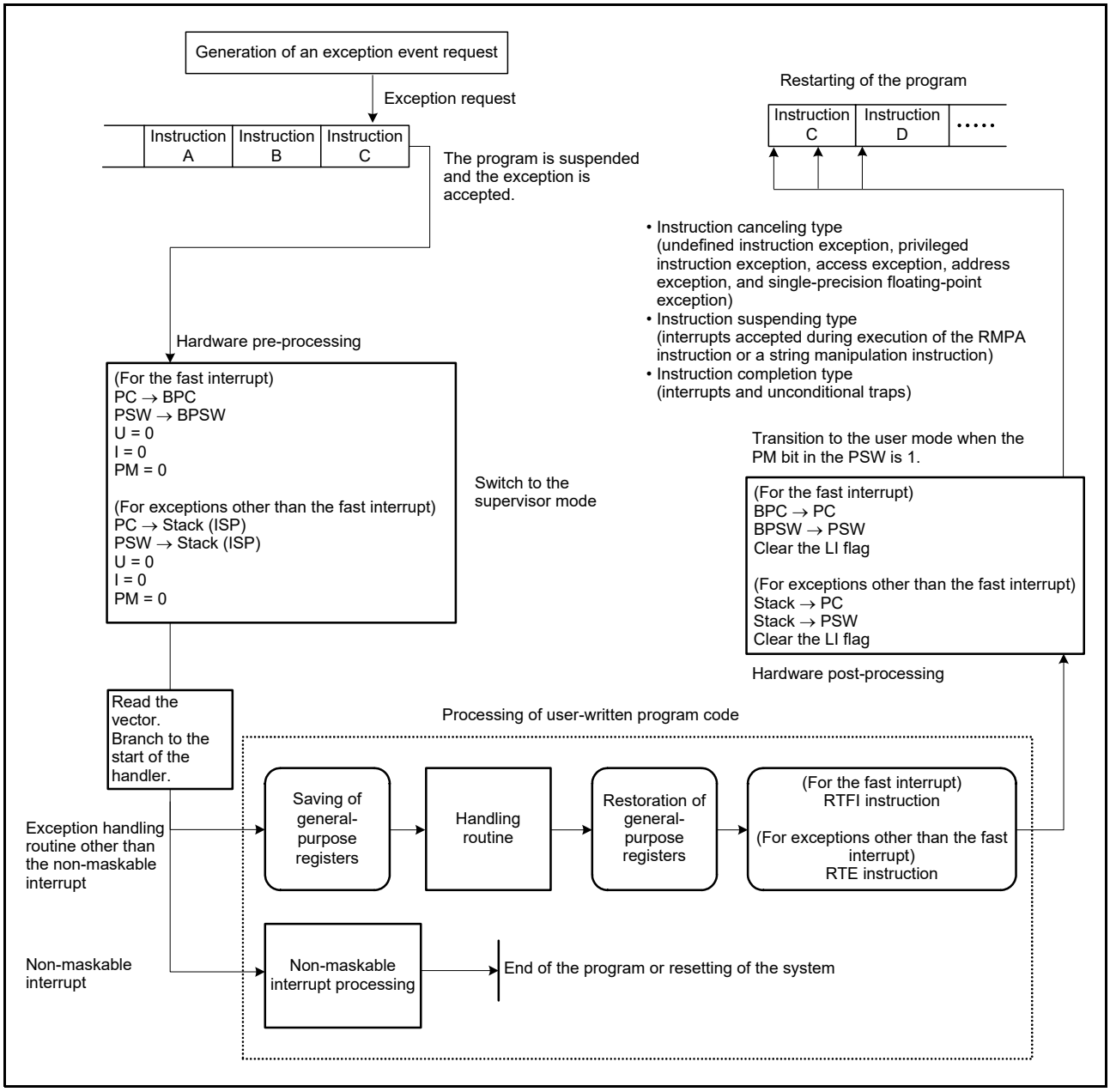


Figure 14.2 Outline of Exception Handling Procedure without World Transitions

When an exception is accepted, hardware processing by the RXv3e CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.

Hardware pre-processing by the RXv3e CPU handles saving of the values of the program counter (PC) and processor status word (PSW). In the case of the fast interrupt, the values of the PC and PSW are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than the fast interrupt, the values of the PC and PSW are saved on the stack. The values of general purpose registers and control registers other than the PC and PSW that are to be used within an exception handling routine must be saved by the user program at the start of the exception

handling routine.
At the end of exception handling routine, after the restoration of registers saved by the user, the RTE instruction is executed to return from the exception handling routine to the original program. For return from the fast interrupt, the RTFI instruction is used instead. In the case of the non-maskable interrupt, end the program or reset the system without returning to the original program.
Hardware post-processing by the RXv3e CPU handles restoration of the contents of the PC and PSW. In the case of the fast interrupt, the values of the BPC and BPSW are restored to the PC and PSW, respectively. In the case of other exceptions, the values are restored from the stack to the PC and PSW.
The stack or the register-saving bank can be used to save and restore the general-purpose and other registers at the start and end of an exception handling routine.
Saving to and restoring from the register-saving bank is executed by using the SAVE and RSTR instructions. To save and restore a register that is not within the scope of saving or restoring by the SAVE and RSTR instructions, use the PUSH and POP instructions for saving to and restoring from the stack.
Using the register-saving bank is usually faster than using the stack, except when the number of registers that require saving and restoring in transitions to and from an exception-handling routine is extremely small.

14.2.2 Exception Handling Procedure with World Transitions

An exception handling procedure has a part that is automatically processed by hardware and parts that are processed by user-created programs (processing programs to be executed in the root world, processing programs to be executed in the destination world of the exception event, and exception handling routines). Figure 14.3 shows an exception handling procedure that includes world transitions.

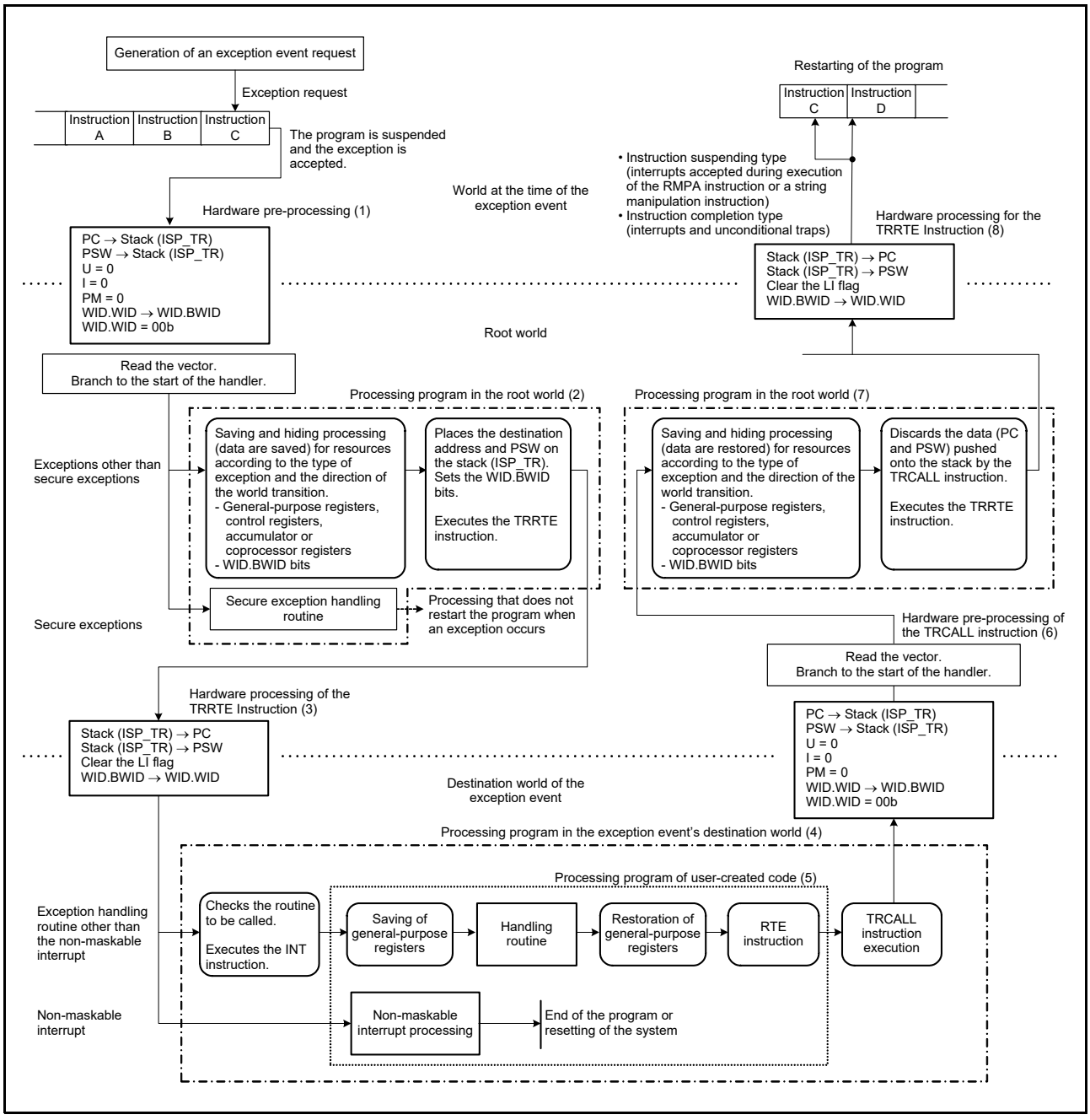


Figure 14.3 Outline of Exception Handling Procedure with World Transitions

When an exception that includes world transitions is accepted, the CPU performs hardware pre-processing (1) and access to the vector table, then obtains the branch destination address. A vector address is assigned to the vector for each exception, and the branch destination address to the processing program in the root world (2) is written to that address. In hardware preprocessing by the CPU (1), the values of the PC and PSW are placed on the stack (ISP_TR). Also, the value of the WID.WID bits is saved in the WID.BWID bits, and a transition is made to the root world with the WID.WID bits set to 0.

In the processing program in the root world (2), the flow of processing depends on whether the exception is secure (in other words, processing of the original program will not resume on return) or not (in other words, processing of the original program in the original world resumes after the exception handling routine in another world has been run via the root world). In the case of a secure exception, after appropriate processing by the secure exception handling routine in the root world, the program is terminated or a jump is made to processing that does not involve resuming processing by the program at the time the exception occurred. In the case of an exception other than a secure exception, consider the type of the exception and the direction of world transitions due to that exception, and perform processing to save or hide resources required by the system (general-purpose registers, control registers, and accumulator or coprocessor registers). After that, set the WID.BWID bits to match the destination world, place the start address and PSW of the processing program in the exception event's destination world (4) on the stack (ISP_TR), and execute the TRRTE instruction.
The hardware processing of the TRRTE instruction (3) causes a transition to the processing program in the exception event's destination world (4).
With the processing program in the exception event's destination world (4), if the interrupt is not non-maskable, the INT instruction is used to cause a transition to processing of the user-created code (5). In processing an interrupt, the value of the IVC register is used as an argument of the INT instruction. Programming in this manner in the destination world of the exception event enables (5) to be used in common with cases that involve no world transitions (this method is shown here as an example of a processing procedure, but can also be used to make a common program except for the processing to save and restore the general-purpose registers). After that, execute the TRCALL instruction to execute the processing program in the root world (7).
In the hardware pre-processing of the TRCALL instruction (6), the values of the PC and PSW are placed on the stack (ISP_TR). Also, the value of the WID.WID bits is saved in the WID.BWID bits, and a transition is made to the root world with the WID.WID bits set to 0.
The processing program in the root world (7), after restoring the resources saved in (2), including the value of the WID.BWID bits, and discarding the PC and PSW that were pushed onto the stack (ISP_TR) by the immediately preceding TRCALL instruction, executes the TRRTE instruction.
The hardware processing of the TRRTE instruction (8) causes a transition to the location for resuming processing by the program at the point where the exception was accepted.

The flow of the processing procedure is explained above, but the processing program in the root world ((2) or (7)), should also run checking according to the current security requirements, such as whether the world from which the TRCALL instruction was called is appropriate, or whether there are any inconsistencies in the number of world transitions going back and forth.

14.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

14.3.1 Timing of Acceptance and Saved PC Value

Table 14.3 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event. For interrupts, non-maskable interrupts, and unconditional traps, the behavior shown in the table below will not change regardless of whether the exception is accepted through world-transition acceptance or normal acceptance.

Table 14.3 Timing of Acceptance and Saved PC Value

Exception		Type of Handling	Timing of Acceptance	Value Saved in the BPC/on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Secure exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Address exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Single-precision floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

14.3.2 Vector and Location for Saving the PC and PSW

The vector for each type of exception and the location for saving the contents of the program counter (PC) and processor status word (PSW) are listed in Table 14.4 and Table 14.5. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Tables.

Table 14.4 Vector and Location for Saving the PC and PSW for an Exception without World Transitions

Exception		Vector	Location for Saving the PC and PSW
Undefined instruction exception		Exception vector table in the current world	Stack in the current world
Privileged instruction exception		Exception vector table in the current world	Stack in the current world
Access exception		Exception vector table in the current world	Stack in the current world
Address exception		Exception vector table in the current world	Stack in the current world
Single-precision floating-point exception		Exception vector table in the current world	Stack in the current world
Reset		FFFF FFFCh	Nowhere
Non-maskable interrupt		Exception vector table in the current world	Stack in the current world
Interrupt	Fast interrupt	FINTV	BPC and BPSW in the root world
	Other than above	Interrupt vector table in the current world	Stack in the current world
Unconditional trap		Interrupt vector table in the current world	Stack in the current world

Table 14.5 Vector and Location for Saving the PC and PSW for an Exception with World Transitions

Exception		Vector	Location for Saving the PC and PSW
Non-maskable interrupt		Exception vector table in the root world (EXTB_TR)	Stack in the root world (ISP_TR)
Interrupts	Fast interrupt	—	—
	Other than above	Vector table for interrupts that cause world transitions (WINTB)	Stack in the root world (ISP_TR)
Secure exception		Exception vector table in the root world (EXTB_TR)	Stack in the root world (ISP_TR)
Unconditional trap		Root call vector table (TRCTB)	Stack in the root world (ISP_TR)

14.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

14.4.1 Hardware Processing for Accepting and Returning from Exceptions Not Involving World Transitions

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The FPSW is not saved by the hardware pre-processing. If single-precision floating-point operation instructions are used within an exception-handling routine, the user must save the FPSW on the stack within the exception-handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

(c) Clearing the LI flag

14.4.2 Hardware Processing for Accepting and Returning from Exceptions Involving World Transitions

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving WID

WID.WID → WID.BWID

(b) Updating the WID

WID.WID: Cleared to 00b

(c) Saving the PSW

PSW → Stack (ISP_TR)

(d) Updating the PM, U, and I bits in the PSW

I: Cleared to 0
U: Cleared to 0
PM: Cleared to 0

(e) Saving the PC

PC → Stack (ISP_TR)

(f) Setting the branch-destination address of the exception handling routine in the PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and branching accordingly.

(2) Hardware post-processing for executing TRRTE instruction

(a) Restoring the WID

WID.BWID → WID.WID

(b) Restoring the PSW

Stack (ISP_TR) → PSW

(c) Restoring the PC

Stack (ISP_TR) → PC

(d) Clearing the LI flag

14.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

14.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the address obtained by the value of EXTB + 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.2 Privileged Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the address obtained by the value of EXTB + 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.3 Access Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the address obtained by the value of EXTB + 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.4 Secure Exception

1. The value of the WID.WID bits is saved to the WID.BWID bits.
2. The WID.WID bits are set to 0 and a transition is made to the root world.
3. The value of the processor status word (PSW) is saved on the stack (ISP_TR).
4. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
5. The value of the program counter (PC) is saved on the stack (ISP_TR).
6. The vector is fetched from the value of EXTB + address 0000 0008h.
7. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

14.5.5 Address Exceptions

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the address obtained by the value of EXTB + 0000 0060h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.6 Single-Precision Floating-Point Exceptions

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the address obtained by the value of EXTB + 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.7 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

14.5.8 Non-Maskable Interrupts (with Normal Acceptance)

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[4:0]) in PSW are set to 0Fh if the current instruction is being executed in the non-secure world; if not, they are set to 1Fh.
5. The vector is fetched from the address obtained by the value of EXTB + 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.9 Non-Maskable Interrupts (with World-Transition Acceptance)

1. The value in WID.WID is saved to WID.BWID.
2. Set WID.WID to 0 and transition to the root world.
3. The destination world ID of the non-maskable interrupt is stored in the WID.BITW bits.
4. The value of the processor status word (PSW) is saved on the stack (ISP_TR).
5. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
6. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP_TR). For other instructions, the PC value of the next instruction is saved.
7. The processor interrupt priority level bits (IPL[4:0]) in the PSW are set to 1Fh.
8. The vector is fetched from the value of EXTB + address 0000 0018h.
9. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

14.5.10 Interrupts (with Normal Acceptance)

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved on the stack (ISP). Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[4:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.11 Interrupts (with World-Transition Acceptance)

1. The value in WID.WID is saved to WID.BWID.
2. Set WID.WID to 0 and transition to the root world.
3. The destination world ID of the interrupt is stored in the WID.BITW bits.
4. The interrupt source number is stored in the IVC.BWINM bits in the root world and in the destination world of the interrupt.
5. The value of the processor status word (PSW) is saved on the stack (ISP_TR).
6. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
7. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP_TR). For other instructions, the PC value of the next instruction is saved.
8. The interrupt priority level of the accepted interrupt is set in the processor interrupt priority level bits (IPL[4:0]) in the PSW.
9. The vector corresponding to the destination world of the interrupt is fetched from the vector table for the interrupt that caused the world transition.
10. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

14.5.12 Unconditional Trap (with Normal Acceptance)

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.13 Unconditional Trap (with World-Transition Acceptance)

1. The value in WID.WID is saved to WID.BWID.
2. Set WID.WID to 0 and transition to the root world.
3. The value of the PSW.I bit is stored in the WID.BWI bit.
4. The value of the processor status word (PSW) is saved on the stack (ISP_TR).
5. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in the PSW are cleared to 0.
6. The value of the program counter (PC) for the next instruction is saved on the stack (ISP_TR).
7. The vector corresponding to the TRCALL instruction number is fetched from the root call vector table.
8. The PC is set to the fetched address and processing branches to the start of the exception handling routine.

14.6 Return from Exception Handling Routine

Executing the instruction listed in Table 14.6 at the end of the corresponding exception handling routine leads to restoration of the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the backup PC (BPC) or the backup PSW (BPSW) by the hardware preprocessing.

Table 14.6 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Secure exception		TRRTE*1
Address exception		RTE
Single-precision floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt (normal acceptance)		RTE*2
Non-maskable interrupt (world-transition acceptance)		TRRTE*2
Interrupts (normal acceptance)	Fast interrupt	RTFI
	Other than above	RTE
Interrupts (world-transition acceptance)	Fast interrupt	—
	Other than above	TRRTE
Unconditional trap (normal acceptance)		RTE
Unconditional trap (world-transition acceptance)		TRRTE

Note 1. Returning to the original program is not possible.
Note 2. Returning to the original program is not recommended. If the destination world of the non-maskable interrupt is secure or non-secure, returning to the original program is in any case not possible.

14.7 Priority of Exception Events

The priority of exception events is listed in Table 14.7. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 14.7 Priority of Exception Events

Priority		Exception Event
High	1	Reset
▲	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Instruction secure exception
	6	Undefined instruction exception Privileged instruction exception
	7	Unconditional trap
	8	Address exception
	9	Operand access exception
	10	Operand secure exception
Low	11	Single-precision floating-point exception

14.8 Exception Generated in Coprocessor

This section describes exceptions generated in the coprocessor and handling of them.

14.8.1 Double-Precision Floating-Point Exceptions

Double-precision floating-point exceptions are generated when any of the five exceptions specified in the IEEE 754 standard, namely overflow, underflow, inexact, division-by-zero, or invalid operation, or an attempt to use processing that is not implemented, is detected upon execution of a double-precision floating-point operation instruction. When a double-precision floating-point exception is generated, an interrupt request is sent to the interrupt controller according to the settings (the DEX, DEU, DEZ, DEO, and DEV bits of the DPSW corresponding to the five exceptions, and the EHM bit of DECNT) without the CPU performing exception handling.
For the vector numbers allocated to the interrupt controller, refer to section 15, Interrupt Controller (ICUM).

15. Interrupt Controller (ICUM)

15.1 Overview

The interrupt controller (ICU) controls various interrupt requests from the peripheral modules and the IRQi pin (i = 0 to 31), and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.
Table 15.1 lists the ICU specifications, and Figure 15.1 shows a block diagram of the interrupt controller.

Table 15.1 ICU Specifications (1/2)

Item		Description
Interrupts	Peripheral interrupts	Interrupts from peripheral modules • Interrupt detection method: Edge detection/level detection (fixed for each interrupt source) • Group interrupt: Multiple interrupt sources are grouped together and treated as an interrupt source.*1 • Group IL0 and Group IL1 interrupts: Interrupt sources of memories and coprocessors that use ICLK as the operating clock • Group BL0 and Group BL1 interrupts: Interrupt sources of peripheral modules that use PCLKB as the operating clock • Group AL0 to Group AL6 interrupts: Interrupt sources of peripheral modules that use PCLKA as the operating clock • Software configurable interrupts: Any of the interrupt sources for peripheral modules can be assigned to interrupt vector numbers 128 to 255.
	External pin interrupt	Interrupt by the input signal to the IRQi pin (i = 0 to 31) • Interrupt detection method: Detection of low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital noise filter can be used to remove noise.
	Software interrupt	• Interrupt request can be generated by writing to a register. • Two interrupt sources
	Interrupt priority	Priority level can be set with interrupt source priority register r (IPRr) (r = 000 to 255).
	Fast interrupt function*2	Fast interrupt response due to the use of internal CPU resources instead of a stack. This function can be used for only one interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*3
	EXDMAC control	Any two of the interrupt sources assigned to software configurable interrupts can be used to start EXDMAC0 and EXDMAC1, respectively.
Non-maskable interrupts *4, *5	NMI pin interrupt	Interrupt by the input signal to the NMI pin • Interrupt detection: Falling edge/rising edge • Digital noise filter can be used to remove noise.
	Main clock oscillation stop detection interrupt	This interrupt occurs when the main clock oscillator stop is detected.
	Sub-clock oscillation stop detection interrupt	This interrupt occurs when the sub-clock oscillator stop is detected.
	WDT underflow/refresh error interrupt	This interrupt occurs when the watchdog timer (WDT) underflows or a refresh error occurs.
	IWDT underflow/refresh error interrupt	This interrupt occurs when the independent watchdog timer (IWDT) underflows or a refresh error occurs.
	Voltage monitoring 1 interrupt	Interrupt from voltage detection circuit 1 (PVD1)
	Voltage monitoring 2 interrupt	Interrupt from voltage detection circuit 2 (PVD2)
	Double-precision floating-point exceptions	Exceptions from double-precision floating-point coprocessor
	Bus error interrupt	This interrupt occurs when a bus error occurs.
	Common memory error interrupt	This interrupt occurs when an ECC error occurs in SRAM0 or SRAM1.
	Local memory error interrupt	This interrupt occurs when an ECC error occurs in the cache memory or LRAM in the CPU.

Table 15.1 ICU Specifications (2/2)

Item		Description
Non-maskable interrupts *4, *5	Code MRAM read access error interrupt	This interrupt occurs when an read access error occurs in the code area of MRAM.
	Extra MRAM read access error interrupt	This interrupt occurs when an read access error occurs in the extra area of MRAM.

Note 1. Groups to which no interrupt source is assigned are reserved. Also, there is no register corresponding to that group.
Note 2. Fast interrupt can be used when the PRTZCR.PRTZC bit is 0.
Note 3. For the DTC and DMAC triggers, refer to Table 15.6, Interrupt Vector Table.
Note 4. Once non-maskable interrupts are enabled, they cannot be disabled.
Note 5. Non-maskable interrupts, except NMI pin interrupt, can be used as maskable interrupts. When using for maskable interrupts, do not change the NMIER register value from the value after reset. To enable the voltage monitoring 1 interrupt, set the PVD1CR1.IRQSEL bit to 1, and to enable the voltage monitoring 2 interrupt, set the PVD2CR1.IRQSEL bit to 1.

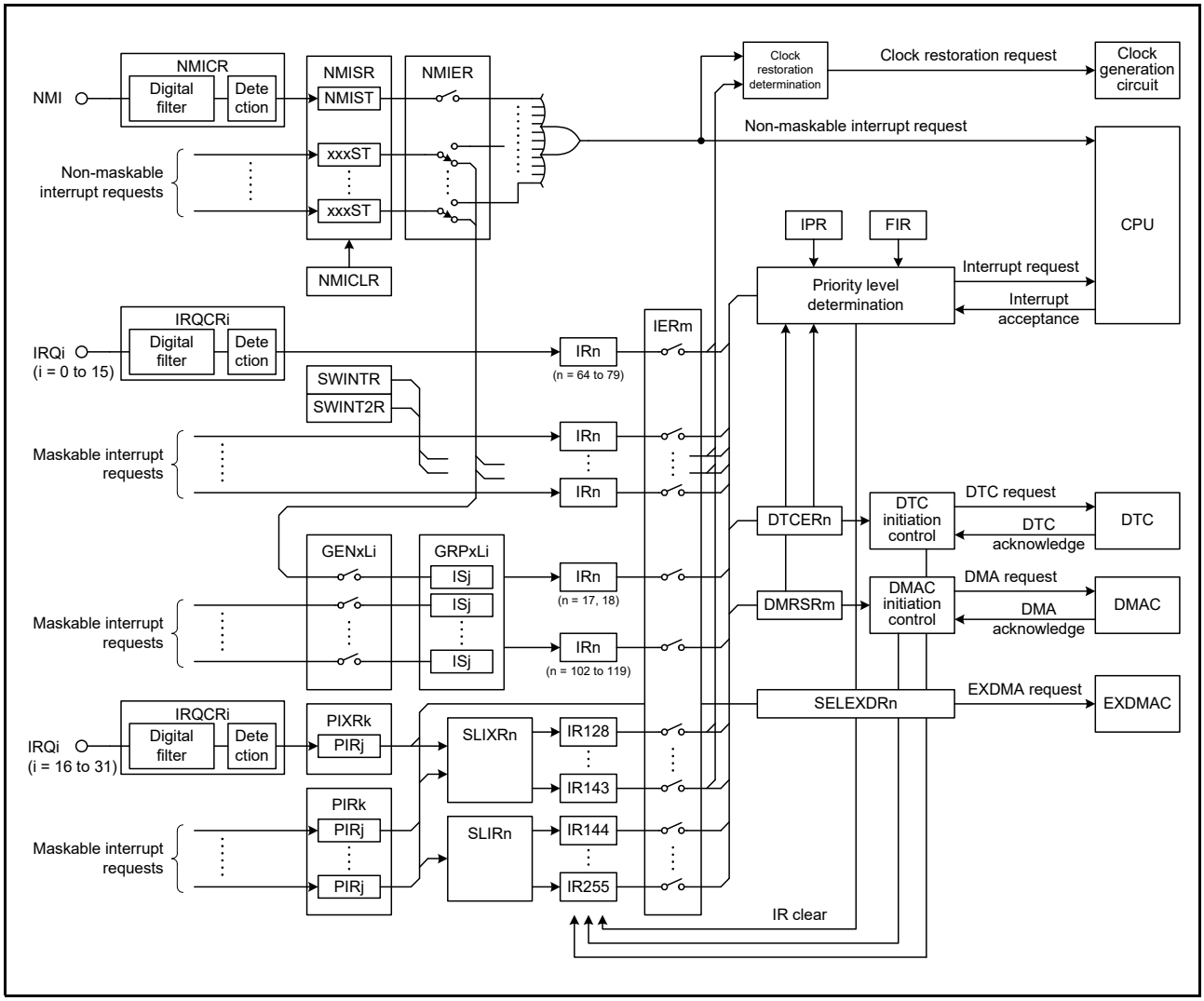


Figure 15.1 Block Diagram of the ICU

Table 15.2 lists I/O pins used for the ICU.

Table 15.2 ICU I/O Pins

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ31	Input	External interrupt request pins

15.2 Register Descriptions

15.2.1 Interrupt Request Register n (IRn) (n = 16 to 255)

Address(es): ICU.IR16 8700 6010h to ICU.IR255 8700 60FFh, ICU_NS.IR16 0700 6010h to ICU_NS.IR255 0700 60FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated. 1: An interrupt request is generated.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. For an edge detection interrupt source, only 0 can be written to this bit; do not write 1.
For a level detection interrupt source, neither 0 nor 1 can be written.

The IRn register indicates whether an interrupt request has been generated.

This register is provided for each interrupt vector number, and n matches the interrupt vector number.

For the correspondence between interrupt sources and interrupt vector numbers, refer to Table 15.6, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

The IR flag is a status flag indicating whether an interrupt request has been generated. This flag becomes 1 when an interrupt request is generated. To detect an interrupt request, set the interrupt enable bit of the peripheral module to enable output of the interrupt request.

An interrupt request can be detected by edge detection or level detection. For interrupts from peripheral modules, the detection method (edge detection or level detection) is determined depending on the source. Refer to Table 15.6, Interrupt Vector Table for details on the detection method for each source. For interrupts from the IRQi pin (i = 0 to 31), edge detection or level detection can be selected by setting the IRQCRi.IRQMD[1:0] bits.

The interrupt status flag for group interrupts is the ISj flag (j = 0 to 31) in the group interrupt request register (GRPIL0, GRPIL1, GRPBL0, GRPBL1, and GRPAL0 to GRPAL6). When any of the ISj flags becomes 1, the IRn.IR flag corresponding to each group interrupt becomes 1. Group interrupts are detected by level detection.

Refer to section 15.4.4, Group Interrupts for details on group interrupts.

(1) Edge detection

This flag becomes 1 under the following condition:

- The IR flag becomes 1 when an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 when the interrupt request destination accepts an interrupt request.
- The IR flag becomes 0 by writing 0 to the IR flag. Note that when the interrupt request destination is the DTC or DMAC, do not write 0 to the IR flag.

(2) Level detection

This flag becomes 1 under any of the following conditions:

- The IR flag is 1 while an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 1 when the ISj flag in the group interrupt request register (GRPIL0, GRPIL1, GRPBL0, GRPBL1, and GRPAL0 to GRPAL6) is 1 (interrupt request is generated) while the ENj bit in the group interrupt request enable register (GENIL0, GENIL1, GENBL0, GENBL1, and GENAL0 to GENAL6) is 1 (enabled) (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 by clearing output of the peripheral module interrupt request. The IR flag does not become 0 when the interrupt request destination accepts the interrupt request. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 0 when the ENj bit in the group interrupt request enable register is 0 (disabled) or when the ISj flag in the group interrupt request register is 0 (interrupt request is not generated).

When level detection is selected for detecting external pin interrupts, set the input level of the IRQi pin to high (i = 0 to 31) to cancel the external pin interrupt that has occurred. When level detection is selected, do not write to the IR flag.

15.2.2 Interrupt Request Enable Register m (IERm) (m = 2 to 31)

Address(es): ICU.IER2 8700 6202h to ICU.IER31 8700 621Fh, ICU_NS.IER2 0700 6202h to ICU_NS.IER31 0700 621Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: The security attributes for this register can be set for each bit.
When a bit is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When a bit is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note: When the interrupt source of the interrupt vector number is reserved, set the corresponding bit to 0. The read value is 0.

The IERm register enables or disables output of the interrupt request to the interrupt request destination.

IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When the IENj bit is 1, an interrupt request is output to the destination. When the IENj bit is 0, an interrupt request is not output to the destination.

The IRn.IR flag (n = 16 to 255) is not affected by the IENj bit setting. Even when the IENj bit is 0, the IR flag changes according to the conditions described in section 15.2.1, Interrupt Request Register n (IRn) (n = 16 to 255).

The IERm.IENj bit is provided for each interrupt vector number.

Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt sources and the IERm.IENj bit.

Note that m and j can be calculated by the following formula:

m = quotient when n divided by 8

j = remainder when n divided by 8

Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for the procedure to set the IERm.IENj bit to select the interrupt request destination.

15.2.3 Interrupt Source Priority Register r (IPRr) (r = 0 to 255)

Address(es): ICU.IPR0 8700 6300h to ICU.IPR255 8700 63FFh, ICU_NS.IPR0 0700 6300h to ICU_NS.IPR255 0700 63FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IPR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IPR[4:0]	Interrupt Priority Level Select	b4 b0 0 0 0 0 0: Level 0 (interrupt disabled)*1 0 0 0 0 1: Level 1 0 0 0 1 0: Level 2 0 0 0 1 1: Level 3 : : 0 1 1 0 1: Level 13 0 1 1 1 0: Level 14 0 1 1 1 1: Level 15 (highest level in Non-secure)*2 1 0 0 0 0: Level 16 1 0 0 0 1: Level 17 1 0 0 1 0: Level 18 : : 1 1 1 0 0: Level 28 1 1 1 0 1: Level 29 1 1 1 1 0: Level 30 1 1 1 1 1: Level 31 (highest level)	R/W
b7 to b5	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. For an interrupt source of a fast interrupt, even when the IPR[4:0] bits are set to level 0, the priority level is level 31.
Note 2. The interrupt sources from the non-secure world should be set in the range of level 0 to level 15.

The IPRr register sets the interrupt priority level of an interrupt source that is assigned to the corresponding interrupt vector number.

IPR[4:0] Bits (Interrupt Priority Level Select)

The IPR[4:0] bits select the interrupt priority level of the corresponding interrupt source.
The priority level selected by the IPR[4:0] bits is used for only determining the priority level of interrupt requests to the CPU. It does not affect transfer requests to the DTC and DMAC.
The CPU accepts only interrupt requests that have the higher priority level than the processor interrupt priority level indicated by the PSW.IPL[4:0] bits.
When multiple interrupt requests are concurrently generated, the priority levels selected by the corresponding IPR[4:0] bits are compared. When multiple interrupt requests that have the same priority level are concurrently generated, the interrupt request that has the smallest interrupt vector number has priority.
Write to this register while the corresponding IERm.IENj bit is 0 (interrupt request is disabled) (m = 2 to 31; j = 0 to 7).
Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vectors and the IPRr register.
Note that r matches the vector number when the interrupt vector number is 32 or greater.

15.2.4 Fast Interrupt Set Register (FIR)

Address(es): ICU.FIR 8700 62F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIEN	—	—	—	—	—	—	—	FVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Set the vector number of an interrupt source that is assigned to a fast interrupt.	R/W
b14 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable*1	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

Note 1. When the PRTZCR.PRTZC bit is 1, the fast interrupt cannot be used. Do not set this bit to 1 when the PRTZCR.PRTZC bit is set to 1.

The FIR register sets an interrupt source that is handled as the fast interrupt.
The fast interrupt is enabled only when the destination is the CPU. When the destination is the DTC or DMAC, the DTC or DMA transfer request is not affected by setting the interrupt vector number as the fast interrupt.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 2 to 31; j = 0 to 7).
Refer to section 15.9, Fast interrupt for details on the fast interrupt.

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits set the interrupt vector number of an interrupt source for the fast interrupt.
Refer to Table 15.6, Interrupt Vector Table for interrupt vector numbers that can be set in the FVCT[7:0] bits. Do not set an interrupt vector number that is reserved.

FIEN Bit (Fast Interrupt Enable)

The FIEN bit enables the fast interrupt to be used.
When the FIEN bit is 1, the interrupt source that is assigned to the interrupt vector number set by the FVCT[7:0] bits is handled as the fast interrupt.
When an interrupt request of the interrupt vector number set by the FVCT[7:0] bits is generated to the CPU while the FIEN bit is 1, an interrupt request is output to the CPU as the fast interrupt regardless of the IPRr register setting (r = 000 to 255). Note that the IPRr register setting is required when using the fast interrupt to exit software standby mode. Refer to section 15.10.2, Exiting Software Standby Mode for details.

15.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): ICU.SWINTR 8700 62E0h, ICU_NS.SWINTR 0700 62E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	The read value is 0. When writing 1 to this bit, a software interrupt request is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The SWINTR register controls generation of a software interrupt request.

SWINT Bit (Software Interrupt Generation)

When the SWINT bit is set to 1, a software interrupt request (SWINT) is generated, and the IR27.IR flag becomes 1. A software interrupt request (SWINT) can be set as a DTC trigger, but it cannot be set as a DMAC trigger.

15.2.6 Software Interrupt 2 Generation Register (SWINT2R)

Address(es): ICU.SWINT2R 8700 62E1h, ICU_NS.SWINT2R 0700 62E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT2	Software Interrupt 2 Generation	The read value is 0. When writing 1 to this bit, a software interrupt request 2 is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The SWINT2R register controls generation of software interrupt request 2.

SWINT2 Bit (Software Interrupt 2 Generation)

When the SWINT2 bit is set to 1, software interrupt request 2 (SWINT2) is generated, and the IR26.IR flag becomes 1. Software interrupt request 2 (SWINT2) can be set as the DTC trigger, but it cannot be set as the DMAC trigger.

15.2.7 DTC Transfer Request Enable Register n (DTCERn) (n = 26 to 255)

Address(es): ICU.DTCER26 8700 611Ah to ICU.DTCER255 8700 61FFh, ICU_NS.DTCER26 0700 611Ah to ICU_NS.DTCER255 0700 61FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is selected as an interrupt request to the CPU or as the DMAC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCERn register selects the interrupt source corresponding to interrupt vector number n as the DTC trigger.
Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt sources and interrupt vector numbers and interrupt sources that can be used as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.
This bit becomes 1 under the following condition:

- 1 is written to the DTCE bit

This bit becomes 0 under any of the following conditions:

- The specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- 0 is written to the DTCE bit

15.2.8 DMAC Trigger Select Register m (DMRSRm) (m = 0 to 7)

Address(es): ICU.DMRSR0 8700 6400h, ICU.DMRSR1 8700 6404h, ICU.DMRSR2 8700 6408h, ICU.DMRSR3 8700 640Ch, ICU.DMRSR4 8700 6410h, ICU.DMRSR5 8700 6414h, ICU.DMRSR6 8700 6418h, ICU.DMRSR7 8700 641Ch, ICU_NS.DMRSR0 0700 6400h, ICU_NS.DMRSR1 0700 6404h, ICU_NS.DMRSR2 0700 6408h, ICU_NS.DMRSR3 0700 640Ch, ICU_NS.DMRSR4 0700 6410h, ICU_NS.DMRSR5 0700 6414h, ICU_NS.DMRSR6 0700 6418h, ICU_NS.DMRSR7 0700 641Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	These bits set the interrupt vector number of the interrupt source as the DMAC trigger.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, the behavior changes depending on the trusted event route setting.
When the trusted event route is disabled, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
When the trusted event route is enabled, it can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The DMRSRm register sets an interrupt source as the DMACm trigger.

Do not set the same vector number for multiple DMRSRm registers. Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Otherwise, the operation is not guaranteed.

Set the interrupt vector number of an interrupt source used as the DMAC trigger in the DMRSRm register. Do not set vector numbers of interrupt sources that cannot be used as the DMAC trigger.

Refer to Table 15.6, Interrupt Vector Table for interrupt vector numbers of interrupt sources.

Write the DMRSRm register while the DMACm.DMCNT.DTE bit is 0.

15.2.9 IRQi Pin Interrupt Control Register (IRQCRi) (i = 0 to 31)

Address(es): ICU.IRQCR0 8700 6C20h to ICU.IRQCR31 8700 6C3Fh, ICU_NS.IRQCR0 0700 6C20h to ICU_NS.IRQCR31 0700 6C3Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN	FCLKSEL[2:0]			—	FLTMD	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IRQMD[1:0]	IRQ Detection Select	b1 b0 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
b2	FLTMD	IRQ Digital Noise Filter Operation Mode Select	0: Detects 3 consecutive levels match 1: Detects 4 consecutive levels match	R/W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b6 to b4	FCLKSEL[2:0]	IRQ Digital Noise Filter Sampling Clock Select	b6 b4 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 0: PCLKB/32 0 1 1: PCLKB/64 1 0 0: PCLKB/128 1 0 1: PCLKB/4 1 1 0: PCLKB/16 1 1 1: Setting prohibited	R/W
b7	FLTEN	IRQ Digital Noise Filter Enable	0: Digital noise filter is disabled. 1: Digital noise filter is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The IRQCRi register selects the detection method and the digital noise filter operation for external pin interrupts.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 2 to 31; j = 0 to 7). After writing to this register, set the IRn.IR flag to 0, and then set the IENj bit to 1 (n = 16 to 255). Note that setting the IR flag to 0 is not required when changing the detection method to level detection.

IRQMD[1:0] Bits (IRQ Detection Select)

The IRQMD[1:0] bits set the detection method for the IRQi pin interrupt (i = 0 to 31).
Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupts.

FLTEN Bit (IRQ Digital Noise Filter Enable)

This bit is used to enable the digital noise filter for the IRQi pin.
When this bit is 1, the signal input to the IRQi pin is sampled with the clock set by the FCLKSEL[2:0] bits. When the sampled signal level does not match for the number of consecutive times specified by the FLTMD bit, the input signal is removed.
Refer to section 15.7.6, Digital Noise Filter for details on the digital noise filter.

15.2.10 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 8700 6580h, ICU_NS.NMISR 0700 6580h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	MRERDST	MRCRDST	DPFPUST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	LMST	CMST	BUSST	—	—	—	—	NMIST	OSTST	SOSTST	—	PVD2ST	PVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested. 1: IWDT underflow/refresh error interrupt is requested.	R
b1	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested. 1: WDT underflow/refresh error interrupt is requested.	R
b2	PVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested. 1: Voltage monitoring 1 interrupt is requested.	R
b3	PVD2ST	Voltage Monitoring 2 Interrupt Status Flag	0: Voltage monitoring 2 interrupt is not requested. 1: Voltage monitoring 2 interrupt is requested.	R
b4	—	Reserved	This bit is read as 0 and cannot be modified.	R
b5	SOSTST	Sub-Clock Oscillation Stop Detection Interrupt Status Flag	0: Sub-clock oscillation stop detection interrupt is not requested. 1: Sub-clock oscillation stop detection interrupt is requested.	R
b6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag	0: Main clock oscillation stop detection interrupt is not requested. 1: Main clock oscillation stop detection interrupt is requested.	R
b7	NMIST	NMI Pin Interrupt Status Flag	0: NMI pin interrupt is not requested. 1: NMI pin interrupt is requested.	R
b11 to b8	—	Reserved	These bits are read as 0 and cannot be modified.	R
b12	BUSST	Bus Error Interrupt Status Flag	0: Bus error interrupt is not requested. 1: Bus error interrupt is requested.	R
b13	CMST	Common Memory Error Interrupt Status Flag	0: Common memory error interrupt is not requested. 1: Common memory error interrupt is requested.	R
b14	LMST	Local Memory Error Interrupt Status Flag	0: Local memory error interrupt is not requested. 1: Local memory error interrupt is requested.	R
b15	—	Reserved	This bit is read as 0 and cannot be modified.	R
b16	DPFPUST	Double-Precision Floating-Point Exception Status Flag	0: Double-precision floating-point exception is not requested. 1: Double-precision floating-point exception is requested.	R
b17	MRCRDST	Code MRAM Read Access Error Interrupt Status Flag	0: Code MRAM read access error interrupt is not requested. 1: Code MRAM read access error interrupt is requested.	R
b18	MRERDST	Extra MRAM Read Access Error Interrupt Status Flag	0: Extra MRAM read access error interrupt is not requested. 1: Extra MRAM read access error interrupt is requested.	R
b31 to b19	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The NMISR register indicates whether a non-maskable interrupt request has occurred.

Each flag in the NMISR register is not affected by the setting of the corresponding bit in the NMIER register.

In the non-maskable interrupt handler, read the NMISR register to check if the other non-maskable interrupt has

occurred. Confirm that all the status flags are 0 before exiting the interrupt handler.

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates whether an IWDT underflow/refresh error interrupt request is generated.

This flag becomes 1 under the following condition:

- An IWDT underflow/refresh error interrupt occurs while the IWDTRCR.RSTIRQS bit is 0.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.IWDTCLR bit.

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates whether an WDT underflow/refresh error interrupt request is generated.

This flag becomes 1 under the following condition:

- A WDT underflow/refresh error interrupt occurs while the WDTRCR.RSTIRQS bit is 0.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.WDTCLR bit.

PVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

The PVD1ST flag indicates whether a voltage monitoring 1 interrupt request is generated.

This flag becomes 1 under the following condition:

- A voltage monitoring 1 interrupt occurs while the PVD1CR1.IRQSEL bit is 0.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.PVD1CLR bit.

PVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

The PVD2ST flag indicates whether a voltage monitoring 2 interrupt request is generated.

This flag becomes 1 under the following condition:

- A voltage monitoring 2 interrupt occurs while the PVD2CR1.IRQSEL bit is 0.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.PVD2CLR bit.

SOSTST Flag (Sub-Clock Oscillation Stop Detection Interrupt Status Flag)

The SOSTST flag indicates whether an oscillation stop detection interrupt request has been generated in the sub-clock oscillator.

This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs in the sub-clock oscillator.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.SOSTCLR bit

OSTST Flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates whether an oscillation stop detection interrupt request has been generated in the main clock oscillator.

This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs in the main clock oscillator.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.OSTCLR bit

NMIST Flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates whether an NMI pin interrupt request has been generated.

This flag becomes 1 under the following condition:

- An edge set in the NMICR.NMIMD bit is input to the NMI pin.

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.NMICLR bit.

BUSST Flag (Bus Error Interrupt Status Flag)

The BUSST flag indicates whether an bus error interrupt request is generated.

This flag becomes 1 under the following condition:

- A bus error interrupt occurs.

This flag becomes 0 under the following condition:

- When all sources which set the BUSST flag to 1 are cleared.

CMST Flag (Common Memory Error Interrupt Status Flag)

The CMST flag indicates whether a common memory error interrupt request is generated.

This flag becomes 1 under the following condition:

- An ECC error occurs in the common memory.

This flag becomes 0 under the following condition:

- When all sources which set the CMST flag to 1 are cleared.

LMST Flag (Local Memory Error Interrupt Status Flag)

The LMST flag indicates whether a local memory error interrupt request is generated.

This flag becomes 1 under the following condition:

- An ECC error occurs in the local memory.

This flag becomes 0 under the following condition:

- When all sources which set the LMST flag to 1 are cleared.

DPFPUST Flag (Double-Precision Floating-Point Exception Status Flag)

The DPFPUST flag indicates whether an double-precision floating-point exception is generated by double-precision floating-point coprocessor.

This flag becomes 1 under the following condition:

- A double-precision floating-point exception occurs.

This flag becomes 0 under the following condition:

- When all sources which set the DPFPUST flag to 1 are cleared.

MRCRDST Flag (Code MRAM Read Access Error Interrupt Status Flag)

The MRCRDST flag indicates whether a read access error interrupt has occurred in the code area of the MRAM.

This flag becomes 1 under the following condition:

- An ECC error has occurred in the code area of the MRAM.

This flag becomes 0 under the following condition:

- When all sources which set the MRCRDST flag to 1 are cleared.

MRERDST Flag (Extra MRAM Read Access Error Interrupt Status Flag)

The MRERDST flag indicates whether a read access error interrupt has occurred in the extra area of the MRAM.

This flag becomes 1 under the following condition:

- An ECC error has occurred in the extra area of the MRAM.

This flag becomes 0 under the following condition:

- When all sources which set the MRERDST flag to 1 are cleared.

15.2.11 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 8700 6584h, ICU_NS.NMIER 0700 6584h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	MRERDEN	MRCRDEN	DPFPUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	LMEN	CMEN	BUSEN	—	—	—	—	NMIEN	OSTEN	SOSTEN	—	PVD2EN	PVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTEN	IWDT Underflow/Refresh Error Enable*1	0: IWDT underflow/refresh error interrupt is disabled. 1: IWDT underflow/refresh error interrupt is enabled.	R/(W) *2
b1	WDTEN	WDT Underflow/Refresh Error Enable*1	0: WDT underflow/refresh error interrupt is disabled. 1: WDT underflow/refresh error interrupt is enabled.	R/(W) *2
b2	PVD1EN	Voltage Monitoring 1 Interrupt Enable*1	0: Voltage monitoring 1 interrupt is disabled. 1: Voltage monitoring 1 interrupt is enabled.	R/(W) *2
b3	PVD2EN	Voltage Monitoring 2 Interrupt Enable*1	0: Voltage monitoring 2 interrupt is disabled. 1: Voltage monitoring 2 interrupt is enabled.	R/(W) *2
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	SOSTEN	Sub-Clock Oscillation Stop Detection Interrupt Enable*1	0: Sub-clock oscillation stop detection interrupt is disabled. 1: Sub-clock oscillation stop detection interrupt is enabled.	R/(W) *2
b6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable*1	0: Main clock oscillation stop detection interrupt is disabled. 1: Main clock oscillation stop detection interrupt is enabled.	R/(W) *2
b7	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled. 1: NMI pin interrupt is enabled.	R/(W) *2
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	BUSEN	Bus Error Interrupt Enable*1	0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.	R/(W) *2
b13	CMEN	Common Memory Error Interrupt Enable*1	0: Common memory error interrupt is disabled. 1: Common memory error interrupt is enabled.	R/(W) *2
b14	LMEN	Local Memory Error Interrupt Enable*1	0: Local memory error interrupt is disabled. 1: Local memory error interrupt is enabled.	R/(W) *2
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	DPFPUEN	Double-Precision Floating-Point Exception Enable*1	0: Double-precision floating-point exception is disabled. 1: Double-precision floating-point exception is enabled.	R/(W) *2
b17	MRCRDEN	Code MRAM Read Access Error Interrupt Enable*1	0: Code MRAM read access error interrupt is disabled. 1: Code MRAM read access error interrupt is enabled.	R/(W) *2
b18	MRERDEN	Local MRAM Read Access Error Interrupt Enable*1	0: Local MRAM read access error interrupt is disabled. 1: Local MRAM read access error interrupt is enabled.	R/(W) *2
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. When this interrupt is to be used as a maskable interrupt, leave this bit set to 0.

Note 2. Once this bit is set to 1, it cannot be set to 0 by software.

The NMIER register enables and disables generation of non-maskable interrupt requests. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

The IWDTEN bit enables and disables generation of a non-maskable interrupt by the IWDT underflow/refresh error interrupt.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

The WDTEN bit enables and disables generation of a non-maskable interrupt by the WDT underflow/refresh error interrupt.

PVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

The PVD1EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 1 interrupt.

PVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

The PVD2EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 2 interrupt.

SOSTEN Bit (Sub-Clock Oscillation Stop Detection Interrupt Enable)

The SOSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt of the sub-clock oscillator.

OSTEN Bit (Main Clock Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt of the main clock oscillator.

NMIEN Bit (NMI Pin Interrupt Enable)

The NMIEN bit enables and disables using the NMI pin interrupt.

BUSEN Bit (Bus Error Interrupt Enable)

The BUSEN bit enables and disables generation of a non-maskable interrupt by the bus error interrupt.

CMEN Bit (Common Memory Error Interrupt Enable)

The CMEN bit enables and disables generation of a non-maskable interrupt by the common memory error interrupt.

LMEN Bit (Local Memory Error Interrupt Enable)

The LMEN bit enables and disables generation of a non-maskable interrupt by the local memory error interrupt.

DPFPUEN Bit (Double-Precision Floating-Point Exception Enable)

The DPFPUEN bit enables and disables generation of a non-maskable interrupt by the double-precision floating-point exception from double-precision floating-point coprocessor.

MRCRDEN Bit (Code MRAM Read Access Error Interrupt Enable)

The MRCRDEN bit enables and disables generation of a non-maskable interrupt by the read access error interrupt of code MRAM.

MRERDEN Bit (Local MRAM Read Access Error Interrupt Enable)

The MRERDEN bit enables and disables generation of a non-maskable interrupt by the read access error interrupt of extra MRAM.

15.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 8700 6588h, ICU_NS.NMICLR 0700 6588h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	NMICLR	OSTCLR	SOSTCLR	—	PVD2CLR	PVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTCLR	IWDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.IWDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b1	WDTCLR	WDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.WDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b2	PVD1CLR	PVD1 Clear	The read value is 0. When 1 is written to this bit, the NMISR.PVD1ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b3	PVD2CLR	PVD2 Clear	The read value is 0. When 1 is written to this bit, the NMISR.PVD2ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b4	—	Reserved	The read value is 0. The write value should be 0.	R/W
b5	SOSTCLR	SOST Clear	The read value is 0. When 1 is written to this bit, the NMISR.SOSTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b6	OSTCLR	OST Clear	The read value is 0. When 1 is written to this bit, the NMISR.OSTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b7	NMICLR	NMI Clear	The read value is 0. When 1 is written to this bit, the NMISR.NMIST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b31 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The NMICLR register clears each flag in the NMISR register.

When writing 1 to a bit in this register, the corresponding status flag becomes 0.

15.2.13 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 8700 6C0Ch, ICU_NS.NMICR 0700 6C0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	NFLTEN	NFCLKSEL[2:0]			—	NFLTMD	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIMD	NMI Detection Select	0: Falling edge 1: Rising edge	R/W
b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b2	NFLTMD	NMI Digital Noise Filter Operation Mode Select	0: Detects 3 consecutive levels match 1: Detects 4 consecutive levels match	R/W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b6 to b4	NFCLKSEL[2:0]	NMI Digital Noise Filter Sampling Clock	b6 b4 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 0: PCLKB/32 0 1 1: PCLKB/64 1 0 0: PCLKB/128 1 0 1: PCLKB/4 1 1 0: PCLKB/16 1 1 1: Setting prohibited	R/W
b7	NFLTEN	NMI Digital Noise Filter Enable	0: Digital noise filter is disabled. 1: Digital noise filter is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The NMICR register selects a detection method for the NMI pin interrupt.
Write to the NMICR register while the NMIER.NMIEN bit is 0.

NMIMD Bit (NMI Detection Select)

This bit sets the detection method for the NMI pin interrupt.

NFLTEN Bit (NMI Digital Noise Filter Enable)

This bit is used to enable the digital noise filter for the NMI pin.
When this bit is 1, the signal input to the NMI pin is sampled with the clock set by the NFCLKSEL[2:0] bits. When the sampled signal level does not match for the number of consecutive times specified by the NFLTMD bit, the input signal is removed.
Refer to section 15.7.6, Digital Noise Filter for details on the digital noise filter.

15.2.14 Group ILi Non-Secure Interrupt Request Register (GRPILi_NS) (i = 0, 1), Group BLi Non-Secure Interrupt Request Register (GRPBLi_NS) (i = 0, 1), Group ALi Non-Secure Interrupt Request Register (GRPALi_NS) (i = 0 to 6)

Address(es): ICU_NS.GRPIL0_NS 0700 6540h, ICU_NS.GRPIL1_NS 0700 6544h, ICU_NS.GRPBL0_NS 0700 6600h, ICU_NS.GRPBL1_NS 0700 6604h, ICU_NS.GRPAL0_NS 0700 6700h, ICU_NS.GRPAL1_NS 0700 6704h, ICU_NS.GRPAL2_NS 0700 6708h, ICU_NS.GRPAL3_NS 0700 670Ch, ICU_NS.GRPAL4_NS 0700 6710h, ICU_NS.GRPAL5_NS 0700 6714h, ICU_NS.GRPAL6_NS 0700 6718h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b16	IS16	Interrupt Status Flag 16		R
b17	IS17	Interrupt Status Flag 17		R
b18	IS18	Interrupt Status Flag 18		R
b19	IS19	Interrupt Status Flag 19		R
b20	IS20	Interrupt Status Flag 20		R
b21	IS21	Interrupt Status Flag 21		R
b22	IS22	Interrupt Status Flag 22		R
b23	IS23	Interrupt Status Flag 23		R
b24	IS24	Interrupt Status Flag 24		R
b25	IS25	Interrupt Status Flag 25		R
b26	IS26	Interrupt Status Flag 26		R
b27	IS27	Interrupt Status Flag 27		R
b28	IS28	Interrupt Status Flag 28		R
b29	IS29	Interrupt Status Flag 29		R
b30	IS30	Interrupt Status Flag 30		R
b31	IS31	Interrupt Status Flag 31		R

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access

error is generated.

These registers indicate the status of each interrupt request from peripherals allocated to the non-secure world among the group interrupt sources.
The GRPILi_NS register (i = 0, 1) contains the statuses of interrupt sources that are detected by level detection and use ICLK as the operating clock.
The GRPBLi_NS register (i = 0, 1) contains the statuses of interrupt sources that are detected by level detection and use PCLKB as the operating clock.
The GRPALi_NS register (i = 0 to 6) contains the statuses of interrupt sources that are detected by level detection and use PCLKA as the operating clock.
These registers are collectively referred to as the “group interrupt request register”.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ISj Flag (Interrupt Status Flag j) (j = 0 to 31)

The ISj flag indicates the status of interrupt sources from peripherals allocated to the non-secure world among the interrupt sources assigned to group interrupts.
The ISj flag becomes 1 only when the corresponding ENj bit in the group interrupt request enable register (GENIL0_NS, GENIL1_NS, GENBL0_NS, GENBL1_NS, and GENAL0_NS to GENAL6_NS) is 1 and the attribute of corresponding interrupt is non-secure. When any of the ISj flags becomes 1, the IRn.IR flag corresponding to the group interrupt becomes 1 (n = 16 to 255).
This flag becomes 1 under the following condition:

- This flag becomes 1 while the corresponding peripheral module interrupt request is generated when the corresponding ENj bit in the group interrupt request enable register is 1.

This flag becomes 0 under any of the following conditions:

- This flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- This flag becomes 0 when the corresponding ENj bit in the group interrupt request enable register is set to 0.

15.2.15 Group ILi Secure Interrupt Request Register (GRPILi_S) (i = 0, 1), Group BLi Secure Interrupt Request Register (GRPBLi_S) (i = 0, 1), Group ALi Secure Interrupt Request Register (GRPALi_S) (i = 0 to 6)

Address(es): ICU.GRPIL0_S 8700 6550h, ICU.GRPIL1_S 8700 6554h, ICU.GRPBL0_S 8700 6640h, ICU.GRPBL1_S 8700 6644h, ICU.GRPAL0_S 8700 6740h, ICU.GRPAL1_S 8700 6744h, ICU.GRPAL2_S 8700 6748h, ICU.GRPAL3_S 8700 674Ch, ICU.GRPAL4_S 8700 6750h, ICU.GRPAL5_S 8700 6754h, ICU.GRPAL6_S 8700 6758h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b16	IS16	Interrupt Status Flag 16		R
b17	IS17	Interrupt Status Flag 17		R
b18	IS18	Interrupt Status Flag 18		R
b19	IS19	Interrupt Status Flag 19		R
b20	IS20	Interrupt Status Flag 20		R
b21	IS21	Interrupt Status Flag 21		R
b22	IS22	Interrupt Status Flag 22		R
b23	IS23	Interrupt Status Flag 23		R
b24	IS24	Interrupt Status Flag 24		R
b25	IS25	Interrupt Status Flag 25		R
b26	IS26	Interrupt Status Flag 26		R
b27	IS27	Interrupt Status Flag 27		R
b28	IS28	Interrupt Status Flag 28		R
b29	IS29	Interrupt Status Flag 29		R
b30	IS30	Interrupt Status Flag 30		R
b31	IS31	Interrupt Status Flag 31		R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access

error is generated.

These registers indicate the status of each interrupt request from peripherals allocated to the secure world among the group interrupt sources.
The GRPILi_S register (i = 0, 1) contains the statuses of interrupt sources that are detected by level detection and use ICLK as the operating clock.
The GRPBLi_S register (i = 0, 1) contains the statuses of interrupt sources that are detected by level detection and use PCLKB as the operating clock.
The GRPALi_S register (i = 0 to 6) contains the statuses of interrupt sources that are detected by level detection and use PCLKA as the operating clock.
These registers are collectively referred to as the “group interrupt request register”.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ISj Flag (Interrupt Status Flag j) (j = 0 to 31)

The ISj flag indicates the status of interrupt sources from peripherals allocated to the secure world among the interrupt sources assigned to group interrupts.
The ISj flag becomes 1 only when the corresponding ENj bit in the group interrupt request enable register (GENIL0_S, GENIL1_S, GENBL0_S, GENBL1_S, and GENAL0_S to GENAL6_S) is 1 and the attribute of corresponding interrupt is secure. When any of the ISj flags becomes 1, the IRn.IR flag corresponding to the group interrupt becomes 1 (n = 16 to 255).
This flag becomes 1 under the following condition:

- This flag becomes 1 while the corresponding peripheral module interrupt request is generated when the corresponding ENj bit in the group interrupt request enable register is 1.

This flag becomes 0 under any of the following conditions:

- This flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- This flag becomes 0 when the corresponding ENj bit in the group interrupt request enable register is set to 0.

15.2.16 Group ILi Non-Secure Interrupt Request Enable Register (GENILi_NS) (i = 0, 1), Group BLi Non-Secure Interrupt Request Enable Register (GENBLi_NS) (i = 0, 1), Group ALi Non-Secure Interrupt Request Enable Register (GENALi_NS) (i = 0 to 6)

Address(es): ICU_NS.GENIL0_NS 0700 6560h, ICU_NS.GENIL1_NS 0700 6564h, ICU_NS.GENBL0_NS 0700 6680h, ICU_NS.GENBL1_NS 0700 6684h, ICU_NS.GENAL0_NS 0700 6780h, ICU_NS.GENAL1_NS 0700 6784h, ICU_NS.GENAL2_NS 0700 6788h, ICU_NS.GENAL3_NS 0700 678Ch, ICU_NS.GENAL4_NS 0700 6790h, ICU_NS.GENAL5_NS 0700 6794h, ICU_NS.GENAL6_NS 0700 6798h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt Request Enable 1		R/W
b2	EN2	Interrupt Request Enable 2		R/W
b3	EN3	Interrupt Request Enable 3		R/W
b4	EN4	Interrupt Request Enable 4		R/W
b5	EN5	Interrupt Request Enable 5		R/W
b6	EN6	Interrupt Request Enable 6		R/W
b7	EN7	Interrupt Request Enable 7		R/W
b8	EN8	Interrupt Request Enable 8		R/W
b9	EN9	Interrupt Request Enable 9		R/W
b10	EN10	Interrupt Request Enable 10		R/W
b11	EN11	Interrupt Request Enable 11		R/W
b12	EN12	Interrupt Request Enable 12		R/W
b13	EN13	Interrupt Request Enable 13		R/W
b14	EN14	Interrupt Request Enable 14		R/W
b15	EN15	Interrupt Request Enable 15		R/W
b16	EN16	Interrupt Request Enable 16		R/W
b17	EN17	Interrupt Request Enable 17		R/W
b18	EN18	Interrupt Request Enable 18		R/W
b19	EN19	Interrupt Request Enable 19		R/W
b20	EN20	Interrupt Request Enable 20		R/W
b21	EN21	Interrupt Request Enable 21		R/W
b22	EN22	Interrupt Request Enable 22		R/W
b23	EN23	Interrupt Request Enable 23		R/W
b24	EN24	Interrupt Request Enable 24		R/W
b25	EN25	Interrupt Request Enable 25		R/W
b26	EN26	Interrupt Request Enable 26		R/W
b27	EN27	Interrupt Request Enable 27		R/W
b28	EN28	Interrupt Request Enable 28		R/W
b29	EN29	Interrupt Request Enable 29		R/W

Bit	Symbol	Bit Name	Description	R/W
b30	EN30	Interrupt Request Enable 30	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b31	EN31	Interrupt Request Enable 31		R/W

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: When a bit has no corresponding interrupt source (bit is reserved), set the bit to 0.

These registers select whether the ISj flag in the group interrupt request register is set to 1 when each interrupt request for group interrupt sources is generated from a peripheral allocated to the non-secure world. These registers are collectively referred to as the “group interrupt request enable register”.
Registers GENIL0_NS, GENIL1_NS, GENBL0_NS, GENBL1_NS, and GENAL0_NS to GENAL6_NS control the ISj flag in registers GRPIL0_NS, GRPIL1_NS, GRPBL0_NS, GRPBL1_NS, and GRPAL0_NS to GRPAL6_NS, respectively.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ENj Bits (Interrupt Request Enable j) (j = 0 to 31)

The ENj bits select whether the corresponding ISj flag in the group interrupt request register is set to 1 when an interrupt request assigned to group interrupts is generated.
When a peripheral module interrupt request is generated while the ENj bit is 1, the corresponding ISj flag in the group interrupt request register (GRPIL0_NS, GRPIL1_NS, GRPBL0_NS, GRPBL1_NS, and GRPAL0_NS to GRPAL6_NS) becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
The setting of this bit is invalid for interrupt sources from peripherals allocated to the secure world.
When the ENj bit is set to 0, the ISj flag becomes 0.

15.2.17 Group ILi Secure Interrupt Request Enable Register (GENILi_S) (i = 0, 1), Group BLi Secure Interrupt Request Enable Register (GENBLi_S) (i = 0, 1), Group ALi Secure Interrupt Request Enable Register (GENALi_S) (i = 0 to 6)

Address(es): ICU.GENIL0_S 8700 6570h, ICU.GENIL1_S 8700 6574h, ICU.GENBL0_S 8700 66C0h, ICU.GENBL1_S 8700 66C4h, ICU.GENAL0_S 8700 67C0h, ICU.GENAL1_S 8700 67C4h, ICU.GENAL2_S 8700 67C8h, ICU.GENAL3_S 8700 67CCh, ICU.GENAL4_S 8700 67D0h, ICU.GENAL5_S 8700 67D4h, ICU.GENAL6_S 8700 67D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt Request Enable 1		R/W
b2	EN2	Interrupt Request Enable 2		R/W
b3	EN3	Interrupt Request Enable 3		R/W
b4	EN4	Interrupt Request Enable 4		R/W
b5	EN5	Interrupt Request Enable 5		R/W
b6	EN6	Interrupt Request Enable 6		R/W
b7	EN7	Interrupt Request Enable 7		R/W
b8	EN8	Interrupt Request Enable 8		R/W
b9	EN9	Interrupt Request Enable 9		R/W
b10	EN10	Interrupt Request Enable 10		R/W
b11	EN11	Interrupt Request Enable 11		R/W
b12	EN12	Interrupt Request Enable 12		R/W
b13	EN13	Interrupt Request Enable 13		R/W
b14	EN14	Interrupt Request Enable 14		R/W
b15	EN15	Interrupt Request Enable 15		R/W
b16	EN16	Interrupt Request Enable 16		R/W
b17	EN17	Interrupt Request Enable 17		R/W
b18	EN18	Interrupt Request Enable 18		R/W
b19	EN19	Interrupt Request Enable 19		R/W
b20	EN20	Interrupt Request Enable 20		R/W
b21	EN21	Interrupt Request Enable 21		R/W
b22	EN22	Interrupt Request Enable 22		R/W
b23	EN23	Interrupt Request Enable 23		R/W
b24	EN24	Interrupt Request Enable 24		R/W
b25	EN25	Interrupt Request Enable 25		R/W
b26	EN26	Interrupt Request Enable 26		R/W
b27	EN27	Interrupt Request Enable 27		R/W
b28	EN28	Interrupt Request Enable 28		R/W
b29	EN29	Interrupt Request Enable 29		R/W
b30	EN30	Interrupt Request Enable 30		R/W
b31	EN31	Interrupt Request Enable 31		R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note: When a bit has no corresponding interrupt source (bit is reserved), set the bit to 0.

These registers select whether the ISj flag in the group interrupt request register is set to 1 when each interrupt request for group interrupt sources is generated from a peripheral allocated to the secure world. These registers are collectively referred to as the “group interrupt request enable register”.
Registers GENIL0_S, GENIL1_S, GENBL0_S, GENBL1_S, and GENAL0_S to GENAL6_S control the ISj flag in registers GRPIL0_S, GRPIL1_S, GRPBL0_S, GRPBL1_S, and GRPAL0_S to GRPAL6_S, respectively.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ENj Bits (Interrupt Request Enable j) (j = 0 to 31)

The ENj bits select whether the corresponding ISj flag in the group interrupt request register is set to 1 when an interrupt request assigned to group interrupts is generated.
When a peripheral module interrupt request is generated while the ENj bit is 1, the corresponding ISj flag in the group interrupt request register (GRPIL0_S, GRPIL1_S, GRPBL0_S, GRPBL1_S, and GRPAL0_S to GRPAL6_S) becomes 1.
When the ENj bit is 0, the ISj flag does not become 1.
The setting of this bit is invalid for interrupt sources from peripherals allocated to the non-secure world.
When the ENj bit is set to 0, the ISj flag becomes 0.

15.2.18 Software Configurable Interrupt Request Register Xk (PIXRk) (k = 0, 1)

Address(es): ICU.PIXR0 8700 6520h, ICU.PIXR1 8700 6521h, ICU_NS.PIXR0 0700 6520h, ICU_NS.PIXR1 0700 6521h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1, *2 0: Ignored. 1: The software configurable interrupt status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt Status Flag 7		R/W

Note: The security attributes for this register can be set for each bit.
When a bit is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When a bit is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Writing to this register is only valid when any of the edge detection is selected by IRQCRi.IRQMD[1:0] bits. When the level detection is selected, writing to this register is ignored.

Note 2. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 only to the flag to be cleared and write 0 to the other flags (write to this register in 8-bit units).

These registers are used for polling of interrupt requests of external pin interrupts (IRQi pin interrupts (i = 16 to 31)) that is assigned to software configurable interrupt by software. The IRQ16 to IRQ23 pins correspond to the PIXR0.PIR0 to PIR7 flags, and the IRQ24 to IRQ31 pins correspond to the PIXR1.PIR0 to PIR7 flags, respectively. When “3FFh” (external pin interrupt) is set in the SLIXRn register, use the corresponding IRn.IR flag for polling (n = 128 to 143).

PIRj Flag (Software Configurable Interrupt Status Flag j) (j = 0 to 7)

When an interrupt request of external pin interrupt assigned to software configurable interrupt is generated, the corresponding PIRj flag becomes 1 regardless of whether “3FFh” is set in the SLIXRn register.
Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling at edge detection, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

- When an interrupt request is generated when edge detection is selected
- When low is input to the IRQi pin when level detection is selected

This flag becomes 0 under the following condition:

- When 1 is written to the PIXRk.PIRj flag when edge detection is selected
- When high is input to the IRQi pin when level detection is selected

15.2.19 Software Configurable Interrupt Request Register k (PIRk) (k = 0 to 127)

Address(es): ICU.PIR0 8700 6800h to ICU.PIR127 8700 687Fh, ICU_NS.PIR0 0700 6800h to ICU_NS.PIR127 0700 687Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The software configurable interrupt B status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt Status Flag 7		R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
To allow access by the non-secure transaction, set the CPSCU.ICUNSACR.NSACE bit to 1.

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 only to the flag to be cleared and write 0 to the other flags (write to this register in 8-bit units).

The PIRk register is used for polling of interrupt requests of interrupt sources that is assigned to software configurable interrupt by software. For an interrupt request of software configurable interrupt set in the SLIXRn or SLIRn register, use the corresponding IRn.IR flag for polling (n = 128 to 255).
Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt.

PIRj Flag (Software Configurable Interrupt Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt is generated, the corresponding PIRk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIXRn or SLIRn register.
Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIRk.PIRj flag

15.2.20 Software Configurable Interrupt Source Select Register Xn (SLIXRn) (n = 128 to 143)

Address(es): ICU.SLIXR128 8700 6900h to ICU.SLIXR143 8700 691Eh, ICU_NS.SLIXR128 0700 6900h to ICU_NS.SLIXR143 0700 691Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	000h: No interrupt source selected. 001h: Interrupt source number 1 002h: Interrupt source number 2 : : 3FDh: Interrupt source number 1021 3FEh: Interrupt source number 1022 3FFh: External pin interrupt (IRQi pin interrupt (i = n – 112)) Settings other than above are prohibited.	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, the behavior changes depending on the trusted event route setting.
When the trusted event route is disabled, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
When the trusted event route is enabled, it can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIXRn register is used to assign interrupt sources assigned to software configurable interrupt, or external pin interrupts (IRQ16 pin interrupt to IRQ31 pin interrupt) to interrupt vector numbers 128 to 143.

Table 15.3, Interrupt Sources for Software Configurable Interrupt lists interrupt sources assigned to software configurable interrupt. Set the SLIXRn register to an interrupt source number that is not reserved. When 000h are set, no interrupt source is assigned to interrupt vector number n. When 3FFh are set, the IRQ16 to IRQ31 pin interrupts are assigned to interrupt vector numbers 128 to 143, respectively.

Do not assign the same interrupt source to multiple registers from registers SLIXRn and SLIRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt for details on which sources can be used to start the DTC or DMAC.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.2.21 Software Configurable Interrupt Source Select Register n (SLIRn) (n = 144 to 255)

Address(es): ICU.SLIR144 8700 6920h to ICU.SLIR255 8700 69FEh, ICU_NS.SLIR144 0700 6920h to ICU_NS.SLIR255 0700 69FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	000h: No interrupt source selected. 001h: Interrupt source number 1 002h: Interrupt source number 2 : : 3FDh: Interrupt source number 1021 3FEh: Interrupt source number 1022 3FFh: No interrupt source selected. Settings other than above are prohibited.	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, the behavior changes depending on the trusted event route setting.
When the trusted event route is disabled, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
When the trusted event route is enabled, it can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIRn register is used to assign interrupt vector numbers 144 to 255 to interrupt sources for software configurable interrupt.

Table 15.3, Interrupt Sources for Software Configurable Interrupt lists interrupt sources for software configurable interrupt. Set the SLIRn register to an interrupt source number that is not reserved. When 000h or 3FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple registers from registers SLIXRn and SLIRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt for details on which sources can be used to start the DTC or DMAC.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 15.3 Interrupt Sources for Software Configurable Interrupt (1/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	None	No interrupt selected (initial value)	N/A	N/A	—
1 to 127	Reserved	—	N/A	N/A	PIR0.PIR1 to PIR15.PIR7
128	CMT2	CMI2 (CMCOR compare match)	✓	✓	PIR16.PIR0
129	CMT3	CMI3 (CMCOR compare match)	✓	✓	PIR16.PIR1
130	TMR0	CMIA0 (TCORA compare match)	✓	N/A	PIR16.PIR2
131		CMIB0 (TCORB compare match)	✓	N/A	PIR16.PIR3
132		OVI0 (TCNT overflow)	N/A	N/A	PIR16.PIR4
133	TMR1	CMIA1 (TCORA compare match)	✓	N/A	PIR16.PIR5
134		CMIB1 (TCORB compare match)	✓	N/A	PIR16.PIR6
135		OVI1 (TCNT overflow)	N/A	N/A	PIR16.PIR7
136	TMR2	CMIA2 (TCORA compare match)	✓	N/A	PIR17.PIR0
137		CMIB2 (TCORB compare match)	✓	N/A	PIR17.PIR1
138		OVI2 (TCNT overflow)	N/A	N/A	PIR17.PIR2
139	TMR3	CMIA3 (TCORA compare match)	✓	N/A	PIR17.PIR3
140		CMIB3 (TCORB compare match)	✓	N/A	PIR17.PIR4
141		OVI3 (TCNT overflow)	N/A	N/A	PIR17.PIR5
142	CMTW0	IC0I0 (input capture of the CMWICR0 register)	✓	✓	PIR17.PIR6
143		IC1I0 (input capture of the CMWICR1 register)	✓	✓	PIR17.PIR7
144		OC0I0 (output compare of the CMWOCR0 register)	✓	✓	PIR18.PIR0
145		OC1I0 (output compare of the CMWOCR1 register)	✓	✓	PIR18.PIR1
146	CMTW1	IC0I1 (input capture of the CMWICR0 register)	✓	✓	PIR18.PIR2
147		IC1I1 (input capture of the CMWICR1 register)	✓	✓	PIR18.PIR3
148		OC0I1 (output compare of the CMWOCR0 register)	✓	✓	PIR18.PIR4
149		OC1I1 (output compare of the CMWOCR1 register)	✓	✓	PIR18.PIR5
150	RTC	CUP (carry interrupt)	N/A	N/A	PIR18.PIR6
151	USB0	USBI0 (status interrupt from 15 sources)	N/A	N/A	PIR18.PIR7
152		D0FIFO (DMA/DTC transfer request 0)	✓	✓	PIR19.PIR0
153		D1FIFO (DMA/DTC transfer request 1)	✓	✓	PIR19.PIR1
154	ELC	ELSR88I (ELC interrupt)	✓	✓	PIR19.PIR2
155		ELSR89I (ELC interrupt)	✓	✓	PIR19.PIR3
156		ELSR90I (ELC interrupt)	✓	✓	PIR19.PIR4
157		ELSR91I (ELC interrupt)	✓	✓	PIR19.PIR5
158	PORTB	EVENT (port B event)	✓	✓	PIR19.PIR6
159	PORTC	EVENT (port C event)	✓	✓	PIR19.PIR7
160	PORTD	EVENT (port D event)	✓	✓	PIR20.PIR0
161	PORTE	EVENT (port E event)	✓	✓	PIR20.PIR1
162	PDMIF	DAT0 (data reception interrupt 0)	✓	✓	PIR20.PIR2
163	Reserved	—	N/A	N/A	PIR20.PIR3
164	Reserved	—	N/A	N/A	PIR20.PIR4
165	SDHI	SBFAI (SD buffer access interrupt)	✓	✓	PIR20.PIR5
166	Reserved	—	N/A	N/A	PIR20.PIR6

Table 15.3 Interrupt Sources for Software Configurable Interrupt (2/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
167	SSIE0	SSITXI0 (transmit data empty)	✓	✓	PIR20.PIR7
168		SSIRXI0 (receive data full)	✓	✓	PIR21.PIR0
169	SSIE1	SSITXI1 (transmit data empty)	✓	✓	PIR21.PIR1
170		SSIRXI1 (receive data full)	✓	✓	PIR21.PIR2
171	PDC	PCDFI	✓	✓	PIR21.PIR3
172 to 255	Reserved	—	N/A	N/A	PIR21.PIR4 to PIR31.PIR7
256	GPTW0	GTCIA0 (GTCCRA input capture/compare match)	✓	✓	PIR32.PIR0
257		GTCIB0 (GTCCRB input capture/compare match)	✓	✓	PIR32.PIR1
258		GTCIC0 (GTCCRC compare match)	✓	✓	PIR32.PIR2
259		GTCID0 (GTCCRD compare match)	✓	✓	PIR32.PIR3
260		GDTE0 (dead time error)	N/A	N/A	PIR32.PIR4
261		GTCIE0 (GTCCRE compare match)	✓	✓	PIR32.PIR5
262		GTCIF0 (GTCCRF compare match)	✓	✓	PIR32.PIR6
263		GTCIV0 (GTCNT overflow (GTPR compare match))	✓	✓	PIR32.PIR7
264		GTCIU0 (GTCNT underflow)	✓	✓	PIR33.PIR0
265		GTCEI0 (cycle count end)	✓	✓	PIR33.PIR1
266	Reserved	—	N/A	N/A	PIR33.PIR2
267	GPTW1	GTCIA1 (GTCCRA input capture/compare match)	✓	✓	PIR33.PIR3
268		GTCIB1 (GTCCRB input capture/compare match)	✓	✓	PIR33.PIR4
269		GTCIC1 (GTCCRC compare match)	✓	✓	PIR33.PIR5
270		GTCID1 (GTCCRD compare match)	✓	✓	PIR33.PIR6
271		GDTE1 (dead time error)	N/A	N/A	PIR33.PIR7
272		GTCIE1 (GTCCRE compare match)	✓	✓	PIR34.PIR0
273		GTCIF1 (GTCCRF compare match)	✓	✓	PIR34.PIR1
274		GTCIV1 (GTCNT overflow (GTPR compare match))	✓	✓	PIR34.PIR2
275		GTCIU1 (GTCNT underflow)	✓	✓	PIR34.PIR3
276		GTCEI1 (cycle count end)	✓	✓	PIR34.PIR4
277	Reserved	—	N/A	N/A	PIR34.PIR5
278	GPTW2	GTCIA2 (GTCCRA input capture/compare match)	✓	✓	PIR34.PIR6
279		GTCIB2 (GTCCRB input capture/compare match)	✓	✓	PIR34.PIR7
280		GTCIC2 (GTCCRC compare match)	✓	✓	PIR35.PIR0
281		GTCID2 (GTCCRD compare match)	✓	✓	PIR35.PIR1
282		GDTE2 (dead time error)	N/A	N/A	PIR35.PIR2
283		GTCIE2 (GTCCRE compare match)	✓	✓	PIR35.PIR3
284		GTCIF2 (GTCCRF compare match)	✓	✓	PIR35.PIR4
285		GTCIV2 (GTCNT overflow (GTPR compare match))	✓	✓	PIR35.PIR5
286		GTCIU2 (GTCNT underflow)	✓	✓	PIR35.PIR6
287		GTCEI2 (cycle count end)	✓	✓	PIR35.PIR7
288	Reserved	—	N/A	N/A	PIR36.PIR0

Table 15.3 Interrupt Sources for Software Configurable Interrupt (3/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
289	GPTW3	GTCIA3 (GTCCRA input capture/compare match)	✓	✓	PIR36.PIR1
290		GTCIB3 (GTCCRB input capture/compare match)	✓	✓	PIR36.PIR2
291		GTCIC3 (GTCCRC compare match)	✓	✓	PIR36.PIR3
292		GTCID3 (GTCCRD compare match)	✓	✓	PIR36.PIR4
293		GDTE3 (dead time error)	N/A	N/A	PIR36.PIR5
294		GTCIE3 (GTCCRE compare match)	✓	✓	PIR36.PIR6
295		GTCIF3 (GTCCRF compare match)	✓	✓	PIR36.PIR7
296		GTCIV3 (GTCNT overflow (GTPR compare match))	✓	✓	PIR37.PIR0
297		GTCIU3 (GTCNT underflow)	✓	✓	PIR37.PIR1
298		GTCEI3 (cycle count end)	✓	✓	PIR37.PIR2
299	Reserved	—	N/A	N/A	PIR37.PIR3
300	GPTW4	GTCIA4 (GTCCRA input capture/compare match)	✓	✓	PIR37.PIR4
301		GTCIB4 (GTCCRB input capture/compare match)	✓	✓	PIR37.PIR5
302		GTCIC4 (GTCCRC compare match)	✓	✓	PIR37.PIR6
303		GTCID4 (GTCCRD compare match)	✓	✓	PIR37.PIR7
304		GDTE4 (dead time error)	N/A	N/A	PIR38.PIR0
305		GTCIE4 (GTCCRE compare match)	✓	✓	PIR38.PIR1
306		GTCIF4 (GTCCRF compare match)	✓	✓	PIR38.PIR2
307		GTCIV4 (GTCNT overflow (GTPR compare match))	✓	✓	PIR38.PIR3
308		GTCIU4 (GTCNT underflow)	✓	✓	PIR38.PIR4
309		GTCEI4 (cycle count end)	✓	✓	PIR38.PIR5
310	Reserved	—	N/A	N/A	PIR38.PIR6
311	GPTW5	GTCIA5 (GTCCRA input capture/compare match)	✓	✓	PIR38.PIR7
312		GTCIB5 (GTCCRB input capture/compare match)	✓	✓	PIR39.PIR0
313		GTCIC5 (GTCCRC compare match)	✓	✓	PIR39.PIR1
314		GTCID5 (GTCCRD compare match)	✓	✓	PIR39.PIR2
315		GDTE5 (dead time error)	N/A	N/A	PIR39.PIR3
316		GTCIE5 (GTCCRE compare match)	✓	✓	PIR39.PIR4
317		GTCIF5 (GTCCRF compare match)	✓	✓	PIR39.PIR5
318		GTCIV5 (GTCNT overflow (GTPR compare match))	✓	✓	PIR39.PIR6
319		GTCIU5 (GTCNT underflow)	✓	✓	PIR39.PIR7
320		GTCEI5 (cycle count end)	✓	✓	PIR40.PIR0
321	Reserved	—	N/A	N/A	PIR40.PIR1
322	GPTW6	GTCIA6 (GTCCRA input capture/compare match)	✓	✓	PIR40.PIR2
323		GTCIB6 (GTCCRB input capture/compare match)	✓	✓	PIR40.PIR3
324		GTCIC6 (GTCCRC compare match)	✓	✓	PIR40.PIR4
325		GTCID6 (GTCCRD compare match)	✓	✓	PIR40.PIR5
326		GDTE6 (dead time error)	N/A	N/A	PIR40.PIR6
327		GTCIE6 (GTCCRE compare match)	✓	✓	PIR40.PIR7
328		GTCIF6 (GTCCRF compare match)	✓	✓	PIR41.PIR0
329		GTCIV6 (GTCNT overflow (GTPR compare match))	✓	✓	PIR41.PIR1
330		GTCIU6 (GTCNT underflow)	✓	✓	PIR41.PIR2
331		GTCEI6 (cycle count end)	✓	✓	PIR41.PIR3
332	Reserved	—	N/A	N/A	PIR41.PIR4

Table 15.3 Interrupt Sources for Software Configurable Interrupt (4/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
333	GPTW7	GTCIA7 (GTCCRA input capture/compare match)	✓	✓	PIR41.PIR5
334		GTCIB7 (GTCCRB input capture/compare match)	✓	✓	PIR41.PIR6
335		GTCIC7 (GTCCRC compare match)	✓	✓	PIR41.PIR7
336		GTCID7 (GTCCRD compare match)	✓	✓	PIR42.PIR0
337		GDTE7 (dead time error)	N/A	N/A	PIR42.PIR1
338		GTCIE7 (GTCCRE compare match)	✓	✓	PIR42.PIR2
339		GTCIF7 (GTCCRF compare match)	✓	✓	PIR42.PIR3
340		GTCIV7 (GTCNT overflow (GTPR compare match))	✓	✓	PIR42.PIR4
341		GTCIU7 (GTCNT underflow)	✓	✓	PIR42.PIR5
342		GTCEI7 (cycle count end)	✓	✓	PIR42.PIR6
343	Reserved	—	N/A	N/A	PIR42.PIR7
344	GPTW8	GTCIA8 (GTCCRA input capture/compare match)	✓	✓	PIR43.PIR0
345		GTCIB8 (GTCCRB input capture/compare match)	✓	✓	PIR43.PIR1
346		GTCIC8 (GTCCRC compare match)	✓	✓	PIR43.PIR2
347		GTCID8 (GTCCRD compare match)	✓	✓	PIR43.PIR3
348		GDTE8 (dead time error)	N/A	N/A	PIR43.PIR4
349		GTCIE8 (GTCCRE compare match)	✓	✓	PIR43.PIR5
350		GTCIF8 (GTCCRF compare match)	✓	✓	PIR43.PIR6
351		GTCIV8 (GTCNT overflow (GTPR compare match))	✓	✓	PIR43.PIR7
352		GTCIU8 (GTCNT underflow)	✓	✓	PIR44.PIR0
353		GTCEI8 (cycle count end)	✓	✓	PIR44.PIR1
354	Reserved	—	N/A	N/A	PIR44.PIR2
355	GPTW9	GTCIA9 (GTCCRA input capture/compare match)	✓	✓	PIR44.PIR3
356		GTCIB9 (GTCCRB input capture/compare match)	✓	✓	PIR44.PIR4
357		GTCIC9 (GTCCRC compare match)	✓	✓	PIR44.PIR5
358		GTCID9 (GTCCRD compare match)	✓	✓	PIR44.PIR6
359		GDTE9 (dead time error)	N/A	N/A	PIR44.PIR7
360		GTCIE9 (GTCCRE compare match)	✓	✓	PIR45.PIR0
361		GTCIF9 (GTCCRF compare match)	✓	✓	PIR45.PIR1
362		GTCIV9 (GTCNT overflow (GTPR compare match))	✓	✓	PIR45.PIR2
363		GTCIU9 (GTCNT underflow)	✓	✓	PIR45.PIR3
364		GTCEI9 (cycle count end)	✓	✓	PIR45.PIR4
365	Reserved	—	N/A	N/A	PIR45.PIR5
366	GPTW10	GTCIA10 (GTCCRA input capture/compare match)	✓	✓	PIR45.PIR6
367		GTCIB10 (GTCCRB input capture/compare match)	✓	✓	PIR45.PIR7
368		GTCIC10 (GTCCRC compare match)	✓	✓	PIR46.PIR0
369		GTCID10 (GTCCRD compare match)	✓	✓	PIR46.PIR1
370		GDTE10 (dead time error)	N/A	N/A	PIR46.PIR2
371		GTCIE10 (GTCCRE compare match)	✓	✓	PIR46.PIR3
372		GTCIF10 (GTCCRF compare match)	✓	✓	PIR46.PIR4
373		GTCIV10 (GTCNT overflow (GTPR compare match))	✓	✓	PIR46.PIR5
374		GTCIU10 (GTCNT underflow)	✓	✓	PIR46.PIR6
375		GTCEI10 (cycle count end)	✓	✓	PIR46.PIR7
376	Reserved	—	N/A	N/A	PIR47.PIR0

Table 15.3 Interrupt Sources for Software Configurable Interrupt (5/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
377	GPTW11	GTCIA11 (GTCCRA input capture/compare match)	✓	✓	PIR47.PIR1
378		GTCIB11 (GTCCRB input capture/compare match)	✓	✓	PIR47.PIR2
379		GTCIC11 (GTCCRC compare match)	✓	✓	PIR47.PIR3
380		GTCID11 (GTCCRD compare match)	✓	✓	PIR47.PIR4
381		GDTE11 (dead time error)	N/A	N/A	PIR47.PIR5
382		GTCIE11 (GTCCRE compare match)	✓	✓	PIR47.PIR6
383		GTCIF11 (GTCCRF compare match)	✓	✓	PIR47.PIR7
384		GTCIV11 (GTCNT overflow (GTPR compare match))	✓	✓	PIR48.PIR0
385		GTCIU11 (GTCNT underflow)	✓	✓	PIR48.PIR1
386		GTCEI11 (cycle count end)	✓	✓	PIR48.PIR2
387	Reserved	—	N/A	N/A	PIR48.PIR3
388	GPTW12	GTCIA12 (GTCCRA input capture/compare match)	✓	✓	PIR48.PIR4
389		GTCIB12 (GTCCRB input capture/compare match)	✓	✓	PIR48.PIR5
390		GTCIC12 (GTCCRC compare match)	✓	✓	PIR48.PIR6
391		GTCID12 (GTCCRD compare match)	✓	✓	PIR48.PIR7
392		GDTE12 (dead time error)	N/A	N/A	PIR49.PIR0
393		GTCIE12 (GTCCRE compare match)	✓	✓	PIR49.PIR1
394		GTCIF12 (GTCCRF compare match)	✓	✓	PIR49.PIR2
395		GTCIV12 (GTCNT overflow (GTPR compare match))	✓	✓	PIR49.PIR3
396		GTCIU12 (GTCNT underflow)	✓	✓	PIR49.PIR4
397		GTCEI12 (cycle count end)	✓	✓	PIR49.PIR5
398	Reserved	—	N/A	N/A	PIR49.PIR6
399	GPTW13	GTCIA13 (GTCCRA input capture/compare match)	✓	✓	PIR49.PIR7
400		GTCIB13 (GTCCRB input capture/compare match)	✓	✓	PIR50.PIR0
401		GTCIC13 (GTCCRC compare match)	✓	✓	PIR50.PIR1
402		GTCID13 (GTCCRD compare match)	✓	✓	PIR50.PIR2
403		GDTE13 (dead time error)	N/A	N/A	PIR50.PIR3
404		GTCIE13 (GTCCRE compare match)	✓	✓	PIR50.PIR4
405		GTCIF13 (GTCCRF compare match)	✓	✓	PIR50.PIR5
406		GTCIV13 (GTCNT overflow (GTPR compare match))	✓	✓	PIR50.PIR6
407		GTCIU13 (GTCNT underflow)	✓	✓	PIR50.PIR7
408		GTCEI13 (cycle count end)	✓	✓	PIR51.PIR0
409	Reserved	—	N/A	N/A	PIR51.PIR1
410	GPTW14	GTCIA14 (GTCCRA input capture/compare match)	✓	✓	PIR51.PIR2
411		GTCIB14 (GTCCRB input capture/compare match)	✓	✓	PIR51.PIR3
412		GTCIC14 (GTCCRC compare match)	✓	✓	PIR51.PIR4
413		GTCID14 (GTCCRD compare match)	✓	✓	PIR51.PIR5
414		GDTE14 (dead time error)	N/A	N/A	PIR51.PIR6
415		GTCIE14 (GTCCRE compare match)	✓	✓	PIR51.PIR7
416		GTCIF14 (GTCCRF compare match)	✓	✓	PIR52.PIR0
417		GTCIV14 (GTCNT overflow (GTPR compare match))	✓	✓	PIR52.PIR1
418		GTCIU14 (GTCNT underflow)	✓	✓	PIR52.PIR2
419		GTCEI14 (cycle count end)	✓	✓	PIR52.PIR3
420	Reserved	—	N/A	N/A	PIR52.PIR4

Table 15.3 Interrupt Sources for Software Configurable Interrupt (6/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
421	GPTW15	GTCIA15 (GTCCRA input capture/compare match)	✓	✓	PIR52.PIR5
422		GTCIB15 (GTCCRB input capture/compare match)	✓	✓	PIR52.PIR6
423		GTCIC15 (GTCCRC compare match)	✓	✓	PIR52.PIR7
424		GTCID15 (GTCCRD compare match)	✓	✓	PIR53.PIR0
425		GDTE15 (dead time error)	N/A	N/A	PIR53.PIR1
426		GTCIE15 (GTCCRE compare match)	✓	✓	PIR53.PIR2
427		GTCIF15 (GTCCRF compare match)	✓	✓	PIR53.PIR3
428		GTCIV15 (GTCNT overflow (GTPR compare match))	✓	✓	PIR53.PIR4
429		GTCIU15 (GTCNT underflow)	✓	✓	PIR53.PIR5
430		GTCEI15 (cycle count end)	✓	✓	PIR53.PIR6
431 to 631	Reserved	—	N/A	N/A	PIR53.PIR7 to PIR78.PIR7
632	GPTP	PTPOUT0 (pulse output 0)	✓	✓	PIR79.PIR0
633		PTPOUT1 (pulse output 1)	✓	✓	PIR79.PIR1
634	Reserved	—	N/A	N/A	PIR79.PIR2
635	Reserved	—	N/A	N/A	PIR79.PIR3
636	GPTP	MATCH0 (media clock 0 recovery)	✓	✓	PIR79.PIR4
637		MATCH1 (media clock 1 recovery)	✓	✓	PIR79.PIR5
638	Reserved	—	N/A	N/A	PIR79.PIR6
639	Reserved	—	N/A	N/A	PIR79.PIR7
640	Reserved	—	N/A	N/A	PIR80.PIR0
641	Reserved	—	N/A	N/A	PIR80.PIR1
642	Reserved	—	N/A	N/A	PIR80.PIR2
643	Reserved	—	N/A	N/A	PIR80.PIR3
644	Reserved	—	N/A	N/A	PIR80.PIR4
645	Reserved	—	N/A	N/A	PIR80.PIR5
646	USBHS	D0FIFO (DMA/DTC transfer request 0)	✓	✓	PIR80.PIR6
647		D1FIFO (DMA/DTC transfer request 1)	✓	✓	PIR80.PIR7
648	RSCI0	RXI (receive data full)	✓	✓	PIR81.PIR0
649		TXI (transmit data empty)	✓	✓	PIR81.PIR1
650		AED (active edge detected)	N/A	N/A	PIR81.PIR2
651	RSCI1	RXI (receive data full)	✓	✓	PIR81.PIR3
652		TXI (transmit data empty)	✓	✓	PIR81.PIR4
653		AED (active edge detected)	N/A	N/A	PIR81.PIR5
654	RSCI2	RXI (receive data full)	✓	✓	PIR81.PIR6
655		TXI (transmit data empty)	✓	✓	PIR81.PIR7
656		AED (active edge detected)	N/A	N/A	PIR82.PIR0
657	RSCI3	RXI (receive data full)	✓	✓	PIR82.PIR1
658		TXI (transmit data empty)	✓	✓	PIR82.PIR2
659		AED (active edge detected)	N/A	N/A	PIR82.PIR3
660	RSCI4	RXI (receive data full)	✓	✓	PIR82.PIR4
661		TXI (transmit data empty)	✓	✓	PIR82.PIR5
662		AED (active edge detected)	N/A	N/A	PIR82.PIR6

Table 15.3 Interrupt Sources for Software Configurable Interrupt (7/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
663	RSCI5	RXI (receive data full)	✓	✓	PIR82.PIR7
664		TXI (transmit data empty)	✓	✓	PIR83.PIR0
665		AED (active edge detected)	N/A	N/A	PIR83.PIR1
666	RSCI6	RXI (receive data full)	✓	✓	PIR83.PIR2
667		TXI (transmit data empty)	✓	✓	PIR83.PIR3
668		AED (active edge detected)	N/A	N/A	PIR83.PIR4
669	RSCI7	RXI (receive data full)	✓	✓	PIR83.PIR5
670		TXI (transmit data empty)	✓	✓	PIR83.PIR6
671		AED (active edge detected)	N/A	N/A	PIR83.PIR7
672	RSCI8	RXI (receive data full)	✓	✓	PIR84.PIR0
673		TXI (transmit data empty)	✓	✓	PIR84.PIR1
674		AED (active edge detected)	N/A	N/A	PIR84.PIR2
675	RSCI9	RXI (receive data full)	✓	✓	PIR84.PIR3
676		TXI (transmit data empty)	✓	✓	PIR84.PIR4
677		AED (active edge detected)	N/A	N/A	PIR84.PIR5
678	RSCI10	RXI (receive data full)	✓	✓	PIR84.PIR6
679		TXI (transmit data empty)	✓	✓	PIR84.PIR7
680		AED (active edge detected)	N/A	N/A	PIR85.PIR0
681	RSCI11	RXI (receive data full)	✓	✓	PIR85.PIR1
682		TXI (transmit data empty)	✓	✓	PIR85.PIR2
683		AED (active edge detected)	N/A	N/A	PIR85.PIR3
684	RSCI12	RXI (receive data full)	✓	✓	PIR85.PIR4
685		TXI (transmit data empty)	✓	✓	PIR85.PIR5
686		AED (active edge detected)	N/A	N/A	PIR85.PIR6
687	Reserved	—	N/A	N/A	PIR85.PIR7
688	Reserved	—	N/A	N/A	PIR86.PIR0
689	RIICHS1	RXI (receive data full)	✓	✓	PIR86.PIR1
690		TXI (transmit data empty)	✓	✓	PIR86.PIR2
691	RIICHS2	RXI (receive data full)	✓	✓	PIR86.PIR3
692		TXI (transmit data empty)	✓	✓	PIR86.PIR4
693	RI3C0	RESPI (response queue full)	✓	✓	PIR86.PIR5
694		CMDI (command queue empty)	✓	✓	PIR86.PIR6
695		IBII (IBI queue empty/full)	✓	✓	PIR86.PIR7
696		RXI (receive data full)	✓	✓	PIR87.PIR0
697		TXI (transmit data empty)	✓	✓	PIR87.PIR1
698		RCVI (receive status queue full)	✓	✓	PIR87.PIR2
699		HRESPI (high-priority response queue full)	✓	✓	PIR87.PIR3
700		HCMDI (high-priority command queue empty)	✓	✓	PIR87.PIR4
701		HRXI (high-priority receive data full)	✓	✓	PIR87.PIR5
702		HTXI (high-priority transmit data empty)	✓	✓	PIR87.PIR6
703 to 722	Reserved	—	N/A	N/A	PIR87.PIR7 to PIR90.PIR2

Table 15.3 Interrupt Sources for Software Configurable Interrupt (8/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
723	RSPIA0	SPRI (receive buffer full)	✓	✓	PIR90.PIR3
724		SPTI (transmit buffer empty)	✓	✓	PIR90.PIR4
725		SPCI (communication end)	N/A	N/A	PIR90.PIR5
726	RSPIA1	SPRI (receive buffer full)	✓	✓	PIR90.PIR6
727		SPTI (transmit buffer empty)	✓	✓	PIR90.PIR7
728		SPCI (communication end)	N/A	N/A	PIR91.PIR0
729	RSPIA2	SPRI (receive buffer full)	✓	✓	PIR91.PIR1
730		SPTI (transmit buffer empty)	✓	✓	PIR91.PIR2
731		SPCI (communication end)	N/A	N/A	PIR91.PIR3
732	CANFDG0	RFDREQ0 (receive FIFO 0 DTC/DMA transfer request)	✓	✓	PIR91.PIR4
733		RFDREQ1 (receive FIFO 1 DTC/DMA transfer request)	✓	✓	PIR91.PIR5
734		RFDREQ2 (receive FIFO 2 DTC/DMA transfer request)	✓	✓	PIR91.PIR6
735		RFDREQ3 (receive FIFO 3 DTC/DMA transfer request)	✓	✓	PIR91.PIR7
736		RFDREQ4 (receive FIFO 4 DTC/DMA transfer request)	✓	✓	PIR92.PIR0
737		RFDREQ5 (receive FIFO 5 DTC/DMA transfer request)	✓	✓	PIR92.PIR1
738		RFDREQ6 (receive FIFO 6 DTC/DMA transfer request)	✓	✓	PIR92.PIR2
739		RFDREQ7 (receive FIFO 7 DTC/DMA transfer request)	✓	✓	PIR92.PIR3
740	CANFD0	CFDREQ (common FIFO DTC/DMA transfer request)	✓	✓	PIR92.PIR4
741	CANFD1	CFDREQ (common FIFO DTC/DMA transfer request)	✓	✓	PIR92.PIR5
742 to 751	Reserved	—	N/A	N/A	PIR92.PIR6 to PIR93.PIR7
752	CANFDG0	AFLRAM0EI (AFLRAM0 ECC error)	N/A	N/A	PIR94.PIR0
753		AFLRAM1EI (AFLRAM1 ECC error)	N/A	N/A	PIR94.PIR1
754		MBRMAEI (MBRAM ECC error)	N/A	N/A	PIR94.PIR2
755	Reserved	—	N/A	N/A	PIR94.PIR3
756	Reserved	—	N/A	N/A	PIR94.PIR4
757	Reserved	—	N/A	N/A	PIR94.PIR5
758	CANFD	RFDREQ0 (receive FIFO 0 DTC/DMA transfer request)	✓	✓	PIR94.PIR6
759		RFDREQ1 (receive FIFO 1 DTC/DMA transfer request)	✓	✓	PIR94.PIR7
760	CANFD2	CFDREQ (common FIFO DTC/DMA transfer request)	✓	✓	PIR95.PIR0
761 to 766	Reserved	—	N/A	N/A	PIR95.PIR1 to PIR95.PIR6
767	CANFD	MBRMAEI (MBRAM ECC error)	N/A	N/A	PIR95.PIR7
768	Reserved	—	N/A	N/A	PIR96.PIR0
769	Reserved	—	N/A	N/A	PIR96.PIR1

Table 15.3 Interrupt Sources for Software Configurable Interrupt (9/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
770	ADC (ADC16H)	ADI0 (group 0 A/D scan end)	✓	✓	PIR96.PIR2
771		ADI1 (group 1 A/D scan end)	✓	✓	PIR96.PIR3
772		ADI2 (group 2 A/D scan end)	✓	✓	PIR96.PIR4
773		ADI3 (group 3 A/D scan end)	✓	✓	PIR96.PIR5
774		ADI4 (group 4 A/D scan end)	✓	✓	PIR96.PIR6
775		ADI5 (group 5 A/D scan end)	✓	✓	PIR96.PIR7
776		ADI6 (group 6 A/D scan end)	✓	✓	PIR97.PIR0
777		ADI7 (group 7 A/D scan end)	✓	✓	PIR97.PIR1
778		ADI8 (group 8 A/D scan end)	✓	✓	PIR97.PIR2
779		ADI9 (group 9 A/D scan end)	✓	✓	PIR97.PIR3
780		ADI10 (group 10 A/D scan end)	✓	✓	PIR97.PIR4
781		ADI11 (group 11 A/D scan end)	✓	✓	PIR97.PIR5
782 to 785	Reserved	—	N/A	N/A	PIR97.PIR6 to PIR98.PIR1
786	ADC (ADC16H)	FIFOREQ0 (FIFO data read request 0)	✓	✓	PIR98.PIR2
787		FIFOREQ1 (FIFO data read request 1)	✓	✓	PIR98.PIR3
788		FIFOREQ2 (FIFO data read request 2)	✓	✓	PIR98.PIR4
789		FIFOREQ3 (FIFO data read request 3)	✓	✓	PIR98.PIR5
790		FIFOREQ4 (FIFO data read request 4)	✓	✓	PIR98.PIR6
791		FIFOREQ5 (FIFO data read request 5)	✓	✓	PIR98.PIR7
792		FIFOREQ6 (FIFO data read request 6)	✓	✓	PIR99.PIR0
793		FIFOREQ7 (FIFO data read request 7)	✓	✓	PIR99.PIR1
794		FIFOREQ8 (FIFO data read request 8)	✓	✓	PIR99.PIR2
795		FIFOREQ9 (FIFO data read request 9)	✓	✓	PIR99.PIR3
796		FIFOREQ10 (FIFO data read request 10)	✓	✓	PIR99.PIR4
797		FIFOREQ11 (FIFO data read request 11)	✓	✓	PIR99.PIR5
798 to 801	Reserved	—	N/A	N/A	PIR99.PIR6 to PIR100.PIR1
802	ADC (ADC16H)	CCMPM0 (composite condition compare match 0)	✓	✓	PIR100.PIR2
803		CCMPM1 (composite condition compare match 1)	✓	✓	PIR100.PIR3
804		CCMPM2 (composite condition compare match 2)	✓	✓	PIR100.PIR4
805		CCMPM3 (composite condition compare match 3)	✓	✓	PIR100.PIR5
806		CCMPM4 (composite condition compare match 4)	✓	✓	PIR100.PIR6
807		CCMPM5 (composite condition compare match 5)	✓	✓	PIR100.PIR7
808		CCMPM6 (composite condition compare match 6)	✓	✓	PIR101.PIR0
809		CCMPM7 (composite condition compare match 7)	✓	✓	PIR101.PIR1
810 to 817	Reserved	—	N/A	N/A	PIR101.PIR2 to PIR102.PIR1

Table 15.3 Interrupt Sources for Software Configurable Interrupt (10/10)

Interrupt Source No.	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
818	ADC (ADC16H)	CCMPUM0 (composite condition compare unmatch 0)	✓	✓	PIR102.PIR2
819		CCMPUM1 (composite condition compare unmatch 1)	✓	✓	PIR102.PIR3
820		CCMPUM2 (composite condition compare unmatch 2)	✓	✓	PIR102.PIR4
821		CCMPUM3 (composite condition compare unmatch 3)	✓	✓	PIR102.PIR5
822		CCMPUM4 (composite condition compare unmatch 4)	✓	✓	PIR102.PIR6
823		CCMPUM5 (composite condition compare unmatch 5)	✓	✓	PIR102.PIR7
824		CCMPUM6 (composite condition compare unmatch 6)	✓	✓	PIR103.PIR0
825		CCMPUM7 (composite condition compare unmatch 7)	✓	✓	PIR103.PIR1
826 to 1022	Reserved	—	N/A	N/A	PIR103.PIR2 to PIR127.PIR6
1023	Reserved	External pin interrupt/No interrupt selected*1	N/A	N/A	PIR127.PIR7

Note 1. “External pin interrupt” for the SLIXRn register, and “no interrupt selected” for the SLIRn register.

15.2.22 EXDMAC Trigger Select Register n (SELEXDRn) (n = 0, 1)

Address(es): ICU.SELEXDR0 8700 6C40h, ICU.SELEXDR1 8700 6C44h, ICU_NS.SELEXDR0 0700 6C40h, ICU_NS.SELEXDR1 0700 6C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SLIRQ[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SLI[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SLI[15:0]	Interrupt Source Select	000h: No interrupt source selected. 001h: Interrupt source number 1 002h: Interrupt source number 2 : : 3FDh: Interrupt source number 1021 3FEh: Interrupt source number 1022 3FFh: External pin interrupt (selected by the SLIRQ[4:0]bit) Settings other than above are prohibited.	R/W
b20 to b16	SLIRQ[4:0]	External Pin Interrupt Select*1	00h: IRQ0 pin interrupt 01h: IRQ1 pin interrupt 02h: IRQ2 pin interrupt : : 1Eh: IRQ30 pin interrupt 1Fh: IRQ31 pin interrupt	R/W
b31 to b21	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Set a value other than 11b in the IRQCRi.IRQMD[1:0] bits corresponding to the selected IRQi pin.

The SELEXDRn register selects an interrupt source assigned to software configurable interrupt, or an external pin interrupt (IRQ0 pin interrupt to IRQ31 pin interrupt) as the EXDMAC trigger.

Table 15.3, Interrupt Sources for Software Configurable Interrupt lists interrupt sources assigned to software configurable interrupt. Set the SELEXDRn register to an interrupt source number that is not reserved. When 000h is set, the EXDMAC cannot be triggered by an interrupt.

15.2.23 Non-Secure Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR_NS)

Address(es): ICU_NS.SLIPRCR_NS 0700 6A00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	WPRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WPRC	Software Configurable Interrupt Source Select Register Write Protect	0: Write enabled. 1: Write disabled.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The SLIPRCR_NS register protects registers that control assignment of software configurable interrupts from being written to. This register protects registers with the non-secure attribute.

WPRC Bit (Software Configurable Interrupt Source Select Register Write Protect)

The WPRC bit disables writing to the SLIXRn and SLIRn registers with the non-secure attribute.

Once this bit is set to 1, it cannot be set to 0 by software.

After assigning software configurable interrupts, confirm that the WPRC bit is 1 before the corresponding interrupt request is generated. Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.2.24 Secure Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR_S)

Address(es): ICU.SLIPRCR_S 8700 6A04h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	WPRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WPRC	Software Configurable Interrupt Source Select Register Write Protect	0: Write enabled. 1: Write disabled.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The SLIPRCR_S register protects registers that control assignment of software configurable interrupts from being written to. This register protects registers with the secure attribute.

WPRC Bit (Software Configurable Interrupt Source Select Register Write Protect)

The WPRC bit disables writing to the SLIXRn and SLIRn registers with the secure attribute.

Once this bit is set to 1, it cannot be set to 0 by software.

After assigning software configurable interrupts, confirm that the WPRC bit is 1 before the corresponding interrupt request is generated. Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.2.25 ICU Security Attribution Register A (ICUSARA)

Address(es): CPSCU.ICUSARA 8701 8040h, CPSCU_NS.ICUSARA 0701 8040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIRQCR31	SAIRQCR30	SAIRQCR29	SAIRQCR28	SAIRQCR27	SAIRQCR26	SAIRQCR25	SAIRQCR24	SAIRQCR23	SAIRQCR22	SAIRQCR21	SAIRQCR20	SAIRQCR19	SAIRQCR18	SAIRQCR17	SAIRQCR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIRQCR15	SAIRQCR14	SAIRQCR13	SAIRQCR12	SAIRQCR11	SAIRQCR10	SAIRQCR9	SAIRQCR8	SAIRQCR7	SAIRQCR6	SAIRQCR5	SAIRQCR4	SAIRQCR3	SAIRQCR2	SAIRQCR1	SAIRQCR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIRQCR0	IRQCR0 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIRQCR1	IRQCR1 Security Attribution		R/W
b2	SAIRQCR2	IRQCR2 Security Attribution		R/W
b3	SAIRQCR3	IRQCR3 Security Attribution		R/W
b4	SAIRQCR4	IRQCR4 Security Attribution		R/W
b5	SAIRQCR5	IRQCR5 Security Attribution		R/W
b6	SAIRQCR6	IRQCR6 Security Attribution		R/W
b7	SAIRQCR7	IRQCR7 Security Attribution		R/W
b8	SAIRQCR8	IRQCR8 Security Attribution		R/W
b9	SAIRQCR9	IRQCR9 Security Attribution		R/W
b10	SAIRQCR10	IRQCR10 Security Attribution		R/W
b11	SAIRQCR11	IRQCR11 Security Attribution		R/W
b12	SAIRQCR12	IRQCR12 Security Attribution		R/W
b13	SAIRQCR13	IRQCR13 Security Attribution		R/W
b14	SAIRQCR14	IRQCR14 Security Attribution		R/W
b15	SAIRQCR15	IRQCR15 Security Attribution		R/W
b16	SAIRQCR16	IRQCR16 and PIXR0.PIR0 Security Attribution		R/W
b17	SAIRQCR17	IRQCR17 and PIXR0.PIR1 Security Attribution		R/W
b18	SAIRQCR18	IRQCR18 and PIXR0.PIR2 Security Attribution		R/W
b19	SAIRQCR19	IRQCR19 and PIXR0.PIR3 Security Attribution		R/W
b20	SAIRQCR20	IRQCR20 and PIXR0.PIR4 Security Attribution		R/W
b21	SAIRQCR21	IRQCR21 and PIXR0.PIR5 Security Attribution		R/W
b22	SAIRQCR22	IRQCR22 and PIXR0.PIR6 Security Attribution		R/W
b23	SAIRQCR23	IRQCR23 and PIXR0.PIR7 Security Attribution		R/W
b24	SAIRQCR24	IRQCR24 and PIXR1.PIR0 Security Attribution		R/W
b25	SAIRQCR25	IRQCR25 and PIXR1.PIR1 Security Attribution		R/W
b26	SAIRQCR26	IRQCR26 and PIXR1.PIR2 Security Attribution		R/W
b27	SAIRQCR27	IRQCR27 and PIXR1.PIR3 Security Attribution		R/W
b28	SAIRQCR28	IRQCR28 and PIXR1.PIR4 Security Attribution		R/W
b29	SAIRQCR29	IRQCR29 and PIXR1.PIR5 Security Attribution		R/W
b30	SAIRQCR30	IRQCR30 and PIXR1.PIR6 Security Attribution		R/W
b31	SAIRQCR31	IRQCR31 and PIXR1.PIR7 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the IRQCR0 to IRQCR31 registers, the PIXR0.PIR0 to PIR7 bits, and the PIXR1.PIR0 to PIR7 bits.

15.2.26 ICU Security Attribution Register B (ICUSARB)

Address(es): CPSCU.ICUSARB 8701 8044h, CPSCU_NS.ICUSARB 0701 8044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SANMI0	SANMICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SANMICR	NMICR Security Attribution	0: Secure 1: Non-secure	R/W
b1	SANMI0	Non-Maskable Interrupt Security Attribution 0		R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the NMICR, NMISR, NMIER, and NMICLR registers.

SANMICR Bit (NMICR Security Attribution)

This bit is used to set the security attribute of the NMICR register.

SANMI0 Bit (Non-Maskable Interrupt Security Attribution 0)

This bit is used to set the security attributes of the NMISR, NMIER, and NMICLR registers.

15.2.27 ICU Security Attribution Register C (ICUSARC)

Address(es): CPSCU.ICUSARC 8701 8048h, CPSCU_NS.ICUSARC 0701 8048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SACLPCR0	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SACLPCR0	CLPCR Security Attribution 0	0: Secure 1: Non-secure	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the CLPCR register.

SACLPCR0 Bit (CLPCR Security Attribution 0)

This bit is used to set the security attribute of the LPM.CLPCR register. Refer to section 11, Low Power Consumption for details on the CLPCR register.

15.2.28 ICU Security Attribution Register D (ICUSARD)

Address(es): CPSCU.ICUSARD 8701 804Ch, CPSCU_NS.ICUSARD 0701 804Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SAMEWE	SAMDWE	SAMCWE	SAMEEE	SAMCEE	—	—	—	—	SAHLDTO	SACM	SALM	SADPFPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SADPFPU	Double-Precision Floating-Point Exception Security Attribution	0: Secure 1: Non-secure	R/W
b1	SALM	Local Memory Error Interrupt Security Attribution		R/W
b2	SACM	Common Memory Error Interrupt Security Attribution		R/W
b3	SAHLDTO	External Bus Hold Timeout Interrupt Security Attribution		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SAMCEE	Code MRAM ECC Error Interrupt Security Attribution	0: Secure 1: Non-secure	R/W
b9	SAMEEE	Extra MRAM ECC Error Interrupt Security Attribution		R/W
b10	SAMCWE	Code MRAM Write Error Interrupt Security Attribution		R/W
b11	SAMDWE	Data MRAM Write Error Interrupt Security Attribution		R/W
b12	SAMEWE	Extra MRAM Write Error Interrupt Security Attribution		R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the interrupt sources assigned to the group IL0.

15.2.29 ICU Security Attribution Register 0 (ICU0SAR0)

Address(es): CPSCU.ICU0SAR0 8701 80A0h, CPSCU_NS.ICU0SAR0 0701 80A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR31	SAIR30	SAIR29	SAIR28	SAIR27	SAIR26	SAIR25	SAIR24	SAIR23	SAIR22	SAIR21	SAIR20	SAIR19	SAIR18	SAIR17	SAIR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR15	SAIR14	SAIR13	SAIR12	SAIR11	SAIR10	SAIR9	SAIR8	SAIR7	SAIR6	SAIR5	SAIR4	SAIR3	SAIR2	SAIR1	SAIR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR0	IR0 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR1	IR1 Security Attribution		R/W
b2	SAIR2	IR2 Security Attribution		R/W
b3	SAIR3	IR3 Security Attribution		R/W
b4	SAIR4	IR4 Security Attribution		R/W
b5	SAIR5	IR5 Security Attribution		R/W
b6	SAIR6	IR6 Security Attribution		R/W
b7	SAIR7	IR7 Security Attribution		R/W
b8	SAIR8	IR8 Security Attribution		R/W
b9	SAIR9	IR9 Security Attribution		R/W
b10	SAIR10	IR10 Security Attribution		R/W
b11	SAIR11	IR11 Security Attribution		R/W
b12	SAIR12	IR12 Security Attribution		R/W
b13	SAIR13	IR13 Security Attribution		R/W
b14	SAIR14	IR14 Security Attribution		R/W
b15	SAIR15	IR15 Security Attribution		R/W
b16	SAIR16	IR16 Security Attribution		R/W
b17	SAIR17	IR17 Security Attribution*1		R/W
b18	SAIR18	IR18 Security Attribution*2		R/W
b19	SAIR19	IR19 Security Attribution*1		R/W
b20	SAIR20	IR20 Security Attribution*2		R/W
b21	SAIR21	IR21 Security Attribution		R/W
b22	SAIR22	IR22 Security Attribution		R/W
b23	SAIR23	IR23 Security Attribution		R/W
b24	SAIR24	IR24 Security Attribution		R/W
b25	SAIR25	IR25 Security Attribution		R/W
b26	SAIR26	IR26 Security Attribution		R/W
b27	SAIR27	IR27 Security Attribution		R/W
b28	SAIR28	IR28 Security Attribution		R/W
b29	SAIR29	IR29 Security Attribution		R/W
b30	SAIR30	IR30 Security Attribution		R/W
b31	SAIR31	IR31 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note: Leave the bits corresponding to the vector numbers marked as “Reserved” in Table 15.6 set to 0.

Note 1. Set this bit to 0.
Note 2. Set this bit to 1.

The ICU0SAR0 register configures the security attributes of registers and bits associated with vector numbers 0 to 31. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers. Table 15.4 lists the targets controlled by each bit.

Table 15.4 Targets controlled by each bit

Bit	Targets to Be Controlled
SAIR0	IPR0 register
SAIR1	IPR1 register
SAIR2	IPR2 register
SAIR3	IPR3 register
SAIR4	IPR4 register
SAIR5	IPR5 register
SAIR6	IPR6 register
SAIR7	IPR7 register
SAIR8	IPR8 register
SAIR9 to SAIR15	None
SAIR16	IR16 register, IER2.IEN0 bit
SAIR17	IR17 register, IER2.IEN1 bit
SAIR18	IR18 register, IER2.IEN2 bit
SAIR19	IR19 register, IER2.IEN3 bit, IPR19 register
SAIR20	IR20 register, IER2.IEN4 bit, IPR20 register
SAIR21 to SAIR25	None
SAIR26	IR26 register, IER3.IEN2 bit, DTCER26 register, SWINT2R register
SAIR27	IR27 register, IER3.IEN3 bit, DTCER27 register, SWINTR register
SAIR28	IR28 register, IER3.IEN4 bit, DTCER28 register
SAIR29	IR29 register, IER3.IEN5 bit, DTCER29 register
SAIR30	IR30 register, IER3.IEN6 bit, DTCER30 register
SAIR31	IR31 register, IER3.IEN7 bit, DTCER31 register

15.2.30 ICU Security Attribution Register 1 (ICU0SAR1)

Address(es): CPSCU.ICU0SAR1 8701 80A4h, CPSCU_NS.ICU0SAR1 0701 80A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR63	SAIR62	SAIR61	SAIR60	SAIR59	SAIR58	SAIR57	SAIR56	SAIR55	SAIR54	SAIR53	SAIR52	SAIR51	SAIR50	SAIR49	SAIR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR47	SAIR46	SAIR45	SAIR44	SAIR43	SAIR42	SAIR41	SAIR40	SAIR39	SAIR38	SAIR37	SAIR36	SAIR35	SAIR34	SAIR33	SAIR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR32	IR32 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR33	IR33 Security Attribution		R/W
b2	SAIR34	IR34 Security Attribution		R/W
b3	SAIR35	IR35 Security Attribution		R/W
b4	SAIR36	IR36 Security Attribution		R/W
b5	SAIR37	IR37 Security Attribution		R/W
b6	SAIR38	IR38 Security Attribution		R/W
b7	SAIR39	IR39 Security Attribution		R/W
b8	SAIR40	IR40 Security Attribution		R/W
b9	SAIR41	IR41 Security Attribution		R/W
b10	SAIR42	IR42 Security Attribution		R/W
b11	SAIR43	IR43 Security Attribution		R/W
b12	SAIR44	IR44 Security Attribution		R/W
b13	SAIR45	IR45 Security Attribution		R/W
b14	SAIR46	IR46 Security Attribution		R/W
b15	SAIR47	IR47 Security Attribution		R/W
b16	SAIR48	IR48 Security Attribution		R/W
b17	SAIR49	IR49 Security Attribution		R/W
b18	SAIR50	IR50 Security Attribution		R/W
b19	SAIR51	IR51 Security Attribution		R/W
b20	SAIR52	IR52 Security Attribution		R/W
b21	SAIR53	IR53 Security Attribution		R/W
b22	SAIR54	IR54 Security Attribution		R/W
b23	SAIR55	IR55 Security Attribution		R/W
b24	SAIR56	IR56 Security Attribution		R/W
b25	SAIR57	IR57 Security Attribution		R/W
b26	SAIR58	IR58 Security Attribution		R/W
b27	SAIR59	IR59 Security Attribution		R/W
b28	SAIR60	IR60 Security Attribution		R/W
b29	SAIR61	IR61 Security Attribution		R/W
b30	SAIR62	IR62 Security Attribution		R/W
b31	SAIR63	IR63 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note: Leave the bits corresponding to the vector numbers marked as “Reserved” in Table 15.6 set to 0.

The ICU0SAR1 register configures the security attributes of registers and bits associated with vector numbers 32 to 63. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 32 to 63)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, and the DTCERn register.

Example:

When “n” is 32: the IR32 register, the IER4.IEN0 bit, the IPR32 register, and the DTCER32 register

When “n” is 63: the IR63 register, the IER7.IEN7 bit, the IPR63 register, and the DTCER63 register

15.2.31 ICU Security Attribution Register 2 (ICU0SAR2)

Address(es): CPSCU.ICU0SAR2 8701 80A8h, CPSCU_NS.ICU0SAR2 0701 80A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR95	SAIR94	SAIR93	SAIR92	SAIR91	SAIR90	SAIR89	SAIR88	SAIR87	SAIR86	SAIR85	SAIR84	SAIR83	SAIR82	SAIR81	SAIR80
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR79	SAIR78	SAIR77	SAIR76	SAIR75	SAIR74	SAIR73	SAIR72	SAIR71	SAIR70	SAIR69	SAIR68	SAIR67	SAIR66	SAIR65	SAIR64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR64	IR64 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR65	IR65 Security Attribution		R/W
b2	SAIR66	IR66 Security Attribution		R/W
b3	SAIR67	IR67 Security Attribution		R/W
b4	SAIR68	IR68 Security Attribution		R/W
b5	SAIR69	IR69 Security Attribution		R/W
b6	SAIR70	IR70 Security Attribution		R/W
b7	SAIR71	IR71 Security Attribution		R/W
b8	SAIR72	IR72 Security Attribution		R/W
b9	SAIR73	IR73 Security Attribution		R/W
b10	SAIR74	IR74 Security Attribution		R/W
b11	SAIR75	IR75 Security Attribution		R/W
b12	SAIR76	IR76 Security Attribution		R/W
b13	SAIR77	IR77 Security Attribution		R/W
b14	SAIR78	IR78 Security Attribution		R/W
b15	SAIR79	IR79 Security Attribution		R/W
b16	SAIR80	IR80 Security Attribution		R/W
b17	SAIR81	IR81 Security Attribution		R/W
b18	SAIR82	IR82 Security Attribution		R/W
b19	SAIR83	IR83 Security Attribution		R/W
b20	SAIR84	IR84 Security Attribution		R/W
b21	SAIR85	IR85 Security Attribution		R/W
b22	SAIR86	IR86 Security Attribution		R/W
b23	SAIR87	IR87 Security Attribution		R/W
b24	SAIR88	IR88 Security Attribution		R/W
b25	SAIR89	IR89 Security Attribution		R/W
b26	SAIR90	IR90 Security Attribution		R/W
b27	SAIR91	IR91 Security Attribution		R/W
b28	SAIR92	IR92 Security Attribution		R/W
b29	SAIR93	IR93 Security Attribution		R/W
b30	SAIR94	IR94 Security Attribution		R/W
b31	SAIR95	IR95 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note: Leave the bits corresponding to the vector numbers marked as “Reserved” in Table 15.6 set to 0.

The ICU0SAR2 register configures the security attributes of registers and bits associated with vector numbers 64 to 95. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 64 to 95)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, and the DTCERn register.

Example:

When “n” is 64: the IR64 register, the IER8.IEN0 bit, the IPR64 register, and the DTCER64 register

When “n” is 95: the IR95 register, the IER11.IEN7 bit, the IPR95 register, and the DTCER95 register

15.2.32 ICU Security Attribution Register 3 (ICU0SAR3)

Address(es): CPSCU.ICU0SAR3 8701 80ACh, CPSCU_NS.ICU0SAR3 0701 80ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR127	SAIR126	SAIR125	SAIR124	SAIR123	SAIR122	SAIR121	SAIR120	SAIR119	SAIR118	SAIR117	SAIR116	SAIR115	SAIR114	SAIR113	SAIR112
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR111	SAIR110	SAIR109	SAIR108	SAIR107	SAIR106	SAIR105	SAIR104	SAIR103	SAIR102	SAIR101	SAIR100	SAIR99	SAIR98	SAIR97	SAIR96
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR96	IR96 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR97	IR97 Security Attribution		R/W
b2	SAIR98	IR98 Security Attribution		R/W
b3	SAIR99	IR99 Security Attribution		R/W
b4	SAIR100	IR100 Security Attribution		R/W
b5	SAIR101	IR101 Security Attribution		R/W
b6	SAIR102	IR102 Security Attribution*1		R/W
b7	SAIR103	IR103 Security Attribution*1		R/W
b8	SAIR104	IR104 Security Attribution*1		R/W
b9	SAIR105	IR105 Security Attribution*1		R/W
b10	SAIR106	IR106 Security Attribution*1		R/W
b11	SAIR107	IR107 Security Attribution*1		R/W
b12	SAIR108	IR108 Security Attribution*1		R/W
b13	SAIR109	IR109 Security Attribution*1		R/W
b14	SAIR110	IR110 Security Attribution*1		R/W
b15	SAIR111	IR111 Security Attribution*2		R/W
b16	SAIR112	IR112 Security Attribution*2		R/W
b17	SAIR113	IR113 Security Attribution*2		R/W
b18	SAIR114	IR114 Security Attribution*2		R/W
b19	SAIR115	IR115 Security Attribution*2		R/W
b20	SAIR116	IR116 Security Attribution*2		R/W
b21	SAIR117	IR117 Security Attribution*2		R/W
b22	SAIR118	IR118 Security Attribution*2		R/W
b23	SAIR119	IR119 Security Attribution*2		R/W
b24	SAIR120	IR120 Security Attribution		R/W
b25	SAIR121	IR121 Security Attribution		R/W
b26	SAIR122	IR122 Security Attribution		R/W
b27	SAIR123	IR123 Security Attribution		R/W
b28	SAIR124	IR124 Security Attribution		R/W
b29	SAIR125	IR125 Security Attribution		R/W
b30	SAIR126	IR126 Security Attribution		R/W
b31	SAIR127	IR127 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note: Leave the bits corresponding to the vector numbers marked as “Reserved” in Table 15.6 set to 0.

Note 1. Set this bit to 0.
Note 2. Set this bit to 1.

The ICU0SAR3 register configures the security attributes of registers and bits associated with vector numbers 96 to 127. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 96 to 127)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, and the DTCERn register.

Example:

When “n” is 96: the IR96 register, the IER12.IEN0 bit, the IPR96 register, and the DTCER96 register

When “n” is 127: the IR127 register, the IER15.IEN7 bit, the IPR127 register, and the DTCER127 register

15.2.33 ICU Security Attribution Register 4 (ICU0SAR4)

Address(es): CPSCU.ICU0SAR4 8701 80B0h, CPSCU_NS.ICU0SAR4 0701 80B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR159	SAIR158	SAIR157	SAIR156	SAIR155	SAIR154	SAIR153	SAIR152	SAIR151	SAIR150	SAIR149	SAIR148	SAIR147	SAIR146	SAIR145	SAIR144
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR143	SAIR142	SAIR141	SAIR140	SAIR139	SAIR138	SAIR137	SAIR136	SAIR135	SAIR134	SAIR133	SAIR132	SAIR131	SAIR130	SAIR129	SAIR128
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR128	IR128 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR129	IR129 Security Attribution		R/W
b2	SAIR130	IR130 Security Attribution		R/W
b3	SAIR131	IR131 Security Attribution		R/W
b4	SAIR132	IR132 Security Attribution		R/W
b5	SAIR133	IR133 Security Attribution		R/W
b6	SAIR134	IR134 Security Attribution		R/W
b7	SAIR135	IR135 Security Attribution		R/W
b8	SAIR136	IR136 Security Attribution		R/W
b9	SAIR137	IR137 Security Attribution		R/W
b10	SAIR138	IR138 Security Attribution		R/W
b11	SAIR139	IR139 Security Attribution		R/W
b12	SAIR140	IR140 Security Attribution		R/W
b13	SAIR141	IR141 Security Attribution		R/W
b14	SAIR142	IR142 Security Attribution		R/W
b15	SAIR143	IR143 Security Attribution		R/W
b16	SAIR144	IR144 Security Attribution		R/W
b17	SAIR145	IR145 Security Attribution		R/W
b18	SAIR146	IR146 Security Attribution		R/W
b19	SAIR147	IR147 Security Attribution		R/W
b20	SAIR148	IR148 Security Attribution		R/W
b21	SAIR149	IR149 Security Attribution		R/W
b22	SAIR150	IR150 Security Attribution		R/W
b23	SAIR151	IR151 Security Attribution		R/W
b24	SAIR152	IR152 Security Attribution		R/W
b25	SAIR153	IR153 Security Attribution		R/W
b26	SAIR154	IR154 Security Attribution		R/W
b27	SAIR155	IR155 Security Attribution		R/W
b28	SAIR156	IR156 Security Attribution		R/W
b29	SAIR157	IR157 Security Attribution		R/W
b30	SAIR158	IR158 Security Attribution		R/W
b31	SAIR159	IR159 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The ICU0SAR4 register configures the security attributes of registers and bits associated with vector numbers 128 to 159. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 128 to 143)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, the DTCERn register, and the SLIXRn register.

Example:

When “n” is 128: the IR128 register, the IER16.IEN0 bit, the IPR128 register, the DTCER128 register, and the SLIXR128 register

When “n” is 143: the IR143 register, the IER17.IEN7 bit, the IPR143 register, the DTCER143 register, and the SLIXR143 register

SAIRn Bit (IRn Security Attribution) (n = 144 to 159)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, the DTCERn register, and the SLIRn register.

Example:

When “n” is 144: the IR144 register, the IER18.IEN0 bit, the IPR144 register, the DTCER144 register, and the SLIR144 register

When “n” is 159: the IR159 register, the IER19.IEN7 bit, the IPR159 register, the DTCER159 register, and the SLIR159 register

15.2.34 ICU Security Attribution Register 5 (ICU0SAR5)

Address(es): CPSCU.ICU0SAR5 8701 80B4h, CPSCU_NS.ICU0SAR5 0701 80B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR191	SAIR190	SAIR189	SAIR188	SAIR187	SAIR186	SAIR185	SAIR184	SAIR183	SAIR182	SAIR181	SAIR180	SAIR179	SAIR178	SAIR177	SAIR176
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR175	SAIR174	SAIR173	SAIR172	SAIR171	SAIR170	SAIR169	SAIR168	SAIR167	SAIR166	SAIR165	SAIR164	SAIR163	SAIR162	SAIR161	SAIR160
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR160	IR160 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR161	IR161 Security Attribution		R/W
b2	SAIR162	IR162 Security Attribution		R/W
b3	SAIR163	IR163 Security Attribution		R/W
b4	SAIR164	IR164 Security Attribution		R/W
b5	SAIR165	IR165 Security Attribution		R/W
b6	SAIR166	IR166 Security Attribution		R/W
b7	SAIR167	IR167 Security Attribution		R/W
b8	SAIR168	IR168 Security Attribution		R/W
b9	SAIR169	IR169 Security Attribution		R/W
b10	SAIR170	IR170 Security Attribution		R/W
b11	SAIR171	IR171 Security Attribution		R/W
b12	SAIR172	IR172 Security Attribution		R/W
b13	SAIR173	IR173 Security Attribution		R/W
b14	SAIR174	IR174 Security Attribution		R/W
b15	SAIR175	IR175 Security Attribution		R/W
b16	SAIR176	IR176 Security Attribution		R/W
b17	SAIR177	IR177 Security Attribution		R/W
b18	SAIR178	IR178 Security Attribution		R/W
b19	SAIR179	IR179 Security Attribution		R/W
b20	SAIR180	IR180 Security Attribution		R/W
b21	SAIR181	IR181 Security Attribution		R/W
b22	SAIR182	IR182 Security Attribution		R/W
b23	SAIR183	IR183 Security Attribution		R/W
b24	SAIR184	IR184 Security Attribution		R/W
b25	SAIR185	IR185 Security Attribution		R/W
b26	SAIR186	IR186 Security Attribution		R/W
b27	SAIR187	IR187 Security Attribution		R/W
b28	SAIR188	IR188 Security Attribution		R/W
b29	SAIR189	IR189 Security Attribution		R/W
b30	SAIR190	IR190 Security Attribution		R/W
b31	SAIR191	IR191 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The ICU0SAR5 register configures the security attributes of registers and bits associated with vector numbers 160 to 191. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 160 to 191)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, the DTCERn register, and the SLIRn register.

Example:

When “n” is 160: the IR160 register, the IER20.IEN0 bit, the IPR160 register, the DTCER160 register, and the SLIR160 register

When “n” is 191: the IR191 register, the IER23.IEN7 bit, the IPR191 register, the DTCER191 register, and the SLIR191 register

15.2.35 ICU Security Attribution Register 6 (ICU0SAR6)

Address(es): CPSCU.ICU0SAR6 8701 80B8h, CPSCU_NS.ICU0SAR6 0701 80B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR223	SAIR222	SAIR221	SAIR220	SAIR219	SAIR218	SAIR217	SAIR216	SAIR215	SAIR214	SAIR213	SAIR212	SAIR211	SAIR210	SAIR209	SAIR208
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR207	SAIR206	SAIR205	SAIR204	SAIR203	SAIR202	SAIR201	SAIR200	SAIR199	SAIR198	SAIR197	SAIR196	SAIR195	SAIR194	SAIR193	SAIR192
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR192	IR192 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR193	IR193 Security Attribution		R/W
b2	SAIR194	IR194 Security Attribution		R/W
b3	SAIR195	IR195 Security Attribution		R/W
b4	SAIR196	IR196 Security Attribution		R/W
b5	SAIR197	IR197 Security Attribution		R/W
b6	SAIR198	IR198 Security Attribution		R/W
b7	SAIR199	IR199 Security Attribution		R/W
b8	SAIR200	IR200 Security Attribution		R/W
b9	SAIR201	IR201 Security Attribution		R/W
b10	SAIR202	IR202 Security Attribution		R/W
b11	SAIR203	IR203 Security Attribution		R/W
b12	SAIR204	IR204 Security Attribution		R/W
b13	SAIR205	IR205 Security Attribution		R/W
b14	SAIR206	IR206 Security Attribution		R/W
b15	SAIR207	IR207 Security Attribution		R/W
b16	SAIR208	IR208 Security Attribution		R/W
b17	SAIR209	IR209 Security Attribution		R/W
b18	SAIR210	IR210 Security Attribution		R/W
b19	SAIR211	IR211 Security Attribution		R/W
b20	SAIR212	IR212 Security Attribution		R/W
b21	SAIR213	IR213 Security Attribution		R/W
b22	SAIR214	IR214 Security Attribution		R/W
b23	SAIR215	IR215 Security Attribution		R/W
b24	SAIR216	IR216 Security Attribution		R/W
b25	SAIR217	IR217 Security Attribution		R/W
b26	SAIR218	IR218 Security Attribution		R/W
b27	SAIR219	IR219 Security Attribution		R/W
b28	SAIR220	IR220 Security Attribution		R/W
b29	SAIR221	IR221 Security Attribution		R/W
b30	SAIR222	IR222 Security Attribution		R/W
b31	SAIR223	IR223 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The ICU0SAR6 register configures the security attributes of registers and bits associated with vector numbers 192 to 223. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 192 to 223)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, the DTCERn register, and the SLIRn register.

Example:

When “n” is 192: the IR192 register, the IER24.IEN0 bit, the IPR192 register, the DTCER192 register, and the SLIR192 register

When “n” is 223: the IR223 register, the IER27.IEN7 bit, the IPR223 register, the DTCER223 register, and the SLIR223 register

15.2.36 ICU Security Attribution Register 7 (ICU0SAR7)

Address(es): CPSCU.ICU0SAR7 8701 80BCh, CPSCU_NS.ICU0SAR7 0701 80BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SAIR255	SAIR254	SAIR253	SAIR252	SAIR251	SAIR250	SAIR249	SAIR248	SAIR247	SAIR246	SAIR245	SAIR244	SAIR243	SAIR242	SAIR241	SAIR240
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SAIR239	SAIR238	SAIR237	SAIR236	SAIR235	SAIR234	SAIR233	SAIR232	SAIR231	SAIR230	SAIR229	SAIR228	SAIR227	SAIR226	SAIR225	SAIR224
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SAIR224	IR224 Security Attribution	0: Secure 1: Non-secure	R/W
b1	SAIR225	IR225 Security Attribution		R/W
b2	SAIR226	IR226 Security Attribution		R/W
b3	SAIR227	IR227 Security Attribution		R/W
b4	SAIR228	IR228 Security Attribution		R/W
b5	SAIR229	IR229 Security Attribution		R/W
b6	SAIR230	IR230 Security Attribution		R/W
b7	SAIR231	IR231 Security Attribution		R/W
b8	SAIR232	IR232 Security Attribution		R/W
b9	SAIR233	IR233 Security Attribution		R/W
b10	SAIR234	IR234 Security Attribution		R/W
b11	SAIR235	IR235 Security Attribution		R/W
b12	SAIR236	IR236 Security Attribution		R/W
b13	SAIR237	IR237 Security Attribution		R/W
b14	SAIR238	IR238 Security Attribution		R/W
b15	SAIR239	IR239 Security Attribution		R/W
b16	SAIR240	IR240 Security Attribution		R/W
b17	SAIR241	IR241 Security Attribution		R/W
b18	SAIR242	IR242 Security Attribution		R/W
b19	SAIR243	IR243 Security Attribution		R/W
b20	SAIR244	IR244 Security Attribution		R/W
b21	SAIR245	IR245 Security Attribution		R/W
b22	SAIR246	IR246 Security Attribution		R/W
b23	SAIR247	IR247 Security Attribution		R/W
b24	SAIR248	IR248 Security Attribution		R/W
b25	SAIR249	IR249 Security Attribution		R/W
b26	SAIR250	IR250 Security Attribution		R/W
b27	SAIR251	IR251 Security Attribution		R/W
b28	SAIR252	IR252 Security Attribution		R/W
b29	SAIR253	IR253 Security Attribution		R/W
b30	SAIR254	IR254 Security Attribution		R/W
b31	SAIR255	IR255 Security Attribution		R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The ICU0SAR7 register configures the security attributes of registers and bits associated with vector numbers 224 to 255. Refer to Table 15.6, Interrupt Vector Table for the correspondence between interrupt vector numbers and registers.

SAIRn Bit (IRn Security Attribution) (n = 224 to 255)

This bit configures the security attribute of the IRn register, the IERm.IENj bit (m = n / 8; j = n % 8), the IPRn register, the DTCERn register, and the SLIRn register.

Example:

When “n” is 224: the IR224 register, the IER28.IEN0 bit, the IPR224 register, the DTCER224 register, and the SLIR224 register

When “n” is 255: the IR255 register, the IER31.IEN7 bit, the IPR255 register, the DTCER255 register, and the SLIR255 register

15.2.37 Trusted Event Route Control Register (TEVTRCR)

Address(es): CPSCU.TEVTRCR 8701 8600h, CPSCU_NS.TEVTRCR 0701 8600h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TEVTEICU0	TEVTE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TEVTE	ELC Trusted Event Route Control	0: Trusted event route is disabled. 1: Trusted event route is enabled.	R/W
b1	TEVTEICU0	ICU Trusted Event Route Control 0		R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

Note: This register is write-protected by the SYSC.PRCR.PRC4 bit.

This register is used to provide special attributes regarding the security attributes for some registers in the ELC and ICU. When the security attribute of the target register is configured as non-secure and the trusted event route is enabled, writing by the non-secure transaction is disabled and reading/writing by the secure transaction is enabled.

TEVTE Bit (ELC Trusted Event Route Control)

This bit controls the trusted event route for the ELC.ELSRn registers.

TEVTEICU0 Bit (ICU Trusted Event Route Control 0)

This bit controls the trusted event route for the DMRSRm, SLIXRn, and SLIRn registers.

15.2.38 ICU Non-Secure Access Control Register (ICUNSACR)

Address(es): CPSCU.ICUNSACR 8701 8604h, CPSCU_NS.ICUNSACR 0701 8604h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NSACE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NSACE	PIRk Register Non-Secure Access Enable	0: Disabled. 1: Enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the SYSC.PRCR.PRC4 bit.

This register is used to provide special attributes for the PIRk registers (k = 0 to 127).

NSACE Bit (PIRk Register Non-Secure Access Enable)

This bit enables reading from or writing to the PIRk registers by the non-secure transaction.

15.2.39 Protect Zone Interrupt Control Register (PRTZCR)

Address(es): CPSCU.PRTZCR 8701 8620h, CPSCU_NS.PRTZCR 0701 8620h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRTZC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRTZC	Protect Zone Control	0: The destination for all interrupts is fixed as the root world. 1: The destination for each interrupt can be selected as either the secure world or the non-secure world.*1	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the SYSC.PRCR.PRC4 bit.
Note 1. The destination for each maskable interrupt can be selected by the corresponding SAIRn bit in the ICU0SARm registers, while the destination for non-maskable interrupts can be selected by the ICUSARB.SANMI0 bit (n = interrupt vector number; m = 0 to 7).

This register controls the operation of the ICU according to the Protect Zone setting.

PRTZC Bit (Protect Zone Control)

This bit is used to direct the ICU whether or not to operate according to the settings of the Protect Zone feature of the CPU. Set this bit to 0 when a program is only to run in the root world, and to 1 when a program that utilizes the concept of worlds within the Protect Zone is to be run.
Table 15.5 lists the setting of this bit and the operation of the ICU.

Table 15.5 PRTZC Bit Setting and ICU Operation

Item	PRTZC = 0	PRTZC = 1	
		Secure interrupt	Non-secure interrupt
Fast interrupt	Enabled	Disabled	
Interrupt destination	Fixed to Root world	Secure world	Non-secure world
Interrupt priority level	0 to 31	0 to 31	0 to 15

15.3 Vector Table

There are two types of exceptions detected by the ICU: maskable interrupts (hereinafter referred to as “interrupts”) and non-maskable interrupts.
When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

15.3.1 Interrupt Vector Table

The vector table that is used for maskable interrupts is called the interrupt vector table.
The interrupt vector table is allocated to a 1024-byte area (4 bytes × 256 sources) beginning with the address set in the INTB register in the CPU. Set the INTB register before enabling interrupts. Set a multiple of 4 in the INTB register.
An unconditional trap is generated when the INT or BRK instruction is executed. Interrupt vectors for unconditional traps use the same area as the interrupt vector table. The BRK instruction is assigned to interrupt vector number 0. The INT instruction is assigned to the interrupt vector number corresponding to the value set as the operand (0 to 255).
Table 15.6 lists details of the interrupt vectors. Details of the headings in Table 15.6 are listed below.

Heading	Description
Interrupt request generated by	Name of the source that generates the interrupt request (module symbol)
Name	Name of the interrupt source (symbol)
Vector no.	Interrupt vector number
Vector address offset	Offset from the address set in the INTB register
Interrupt detection method	“Edge” indicates that the interrupt is detected by edge detection. “Level” indicates that the interrupt is detected by level detection.
CPU interrupt	Interrupt source indicated by “✓” can be used as an interrupt source to the CPU.
Start the DTC	Interrupt source indicated by “✓” can be used as the DTC trigger.
Start the DMAC	Interrupt source indicated by “✓” can be used as the DMAC trigger.
Exit from SSBY	Interrupt source indicated by “✓” can be used as a source to exit software standby mode.
IER	Name of the bit in the IER register corresponding to the interrupt vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Table 15.6 Interrupt Vector Table (1/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	—	—	—
BUS	BUSERR*2	16	0040h	Level	✓	N/A	N/A	N/A	IER2.IEN0	IPR0	—
ICU*1	GROUPIL0_S	17	0044h	Level	✓	N/A	N/A	N/A	IER2.IEN1	IPR1	—
	GROUPIL0_NS	18	0048h	Level	✓	N/A	N/A	N/A	IER2.IEN2	IPR2	—
	GROUPIL1_S	19	004Ch	Level	✓	N/A	N/A	N/A	IER2.IEN3	IPR19	—
	GROUPIL1_NS	20	0050h	Level	✓	N/A	N/A	N/A	IER2.IEN4	IPR20	—
—	Reserved	21	0054h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	23	005Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT2	26	0068h	Edge	✓	✓	N/A	N/A	IER3.IEN2	IPR8	DTCER26
	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	IER3.IEN3	IPR3	DTCER27
CMT0	CMI0 (for OS)	28	0070h	Edge	✓	✓	✓	N/A	IER3.IEN4	IPR4	DTCER28
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	IER3.IEN5	IPR5	DTCER29
CMTW0	CMWI0	30	0078h	Edge	✓	✓	✓	N/A	IER3.IEN6	IPR6	DTCER30
CMTW1	CMWI1	31	007Ch	Edge	✓	✓	✓	N/A	IER3.IEN7	IPR7	DTCER31
MRAM	MRDYI	32	0080h	Edge	✓	N/A	N/A	N/A	IER4.IEN0	IPR32	—
	MRPRRI	33	0084h	Edge	✓	✓	✓	N/A	IER4.IEN1	IPR33	DTCER33
—	Reserved	34	0088h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	36	0090h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	38	0098h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	39	009Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	40	00A0h	—	N/A	N/A	N/A	N/A	—	—	—

Table 15.6 Interrupt Vector Table (2/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
—	Reserved	41	00A4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	42	00A8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	43	00ACh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	44	00B0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	45	00B4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	46	00B8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	47	00BCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	52	00D0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	53	00D4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	54	00D8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	55	00DCh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	56	00E0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	57	00E4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	58	00E8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	59	00ECh	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	60	00F0h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	61	00F4h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	62	00F8h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	63	00FCh	—	N/A	N/A	N/A	N/A	—	—	—
ICU	IRQ0	64	0100h	Edge/ Level	✓	✓	✓	✓	IER8.IEN0	IPR64	DTCER64
	IRQ1	65	0104h	Edge/ Level	✓	✓	✓	✓	IER8.IEN1	IPR65	DTCER65
	IRQ2	66	0108h	Edge/ Level	✓	✓	✓	✓	IER8.IEN2	IPR66	DTCER66
	IRQ3	67	010Ch	Edge/ Level	✓	✓	✓	✓	IER8.IEN3	IPR67	DTCER67
	IRQ4	68	0110h	Edge/ Level	✓	✓	✓	✓	IER8.IEN4	IPR68	DTCER68
	IRQ5	69	0114h	Edge/ Level	✓	✓	✓	✓	IER8.IEN5	IPR69	DTCER69
	IRQ6	70	0118h	Edge/ Level	✓	✓	✓	✓	IER8.IEN6	IPR70	DTCER70
	IRQ7	71	011Ch	Edge/ Level	✓	✓	✓	✓	IER8.IEN7	IPR71	DTCER71
	IRQ8	72	0120h	Edge/ Level	✓	✓	✓	✓	IER9.IEN0	IPR72	DTCER72
	IRQ9	73	0124h	Edge/ Level	✓	✓	✓	✓	IER9.IEN1	IPR73	DTCER73
	IRQ10	74	0128h	Edge/ Level	✓	✓	✓	✓	IER9.IEN2	IPR74	DTCER74

Table 15.6 Interrupt Vector Table (3/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
ICU	IRQ11	75	012Ch	Edge/ Level	✓	✓	✓	✓	IER9.IEN3	IPR75	DTCER75
	IRQ12	76	0130h	Edge/ Level	✓	✓	✓	✓	IER9.IEN4	IPR76	DTCER76
	IRQ13	77	0134h	Edge/ Level	✓	✓	✓	✓	IER9.IEN5	IPR77	DTCER77
	IRQ14	78	0138h	Edge/ Level	✓	✓	✓	✓	IER9.IEN6	IPR78	DTCER78
	IRQ15	79	013Ch	Edge/ Level	✓	✓	✓	✓	IER9.IEN7	IPR79	DTCER79
RI3C0	WUI	80	0140h	Level	✓	N/A	N/A	✓	IER10.IEN0	IPR80	—
RIICHS1	WUI	81	0144h	Level	✓	N/A	N/A	✓	IER10.IEN1	IPR81	—
RIICHS2	WUI	82	0148h	Level	✓	N/A	N/A	✓	IER10.IEN2	IPR82	—
VBATT	TAMPDI	83	014Ch	Level	✓	N/A	N/A	✓	IER10.IEN3	IPR83	—
PDMIF	SDET	84	0150h	Level	✓	N/A	N/A	✓	IER10.IEN4	IPR84	—
SOST	SOSTDI*2	85	0154h	Edge	✓	N/A	N/A	✓	IER10.IEN5	IPR85	—
—	Reserved	86	0158h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	87	015Ch	—	N/A	N/A	N/A	N/A	—	—	—
PVD1	PVD1*2	88	0160h	Edge	✓	N/A	N/A	✓	IER11.IEN0	IPR88	—
PVD2	PVD2*2	89	0164h	Edge	✓	N/A	N/A	✓	IER11.IEN1	IPR89	—
USB0	USBR	90	0168h	Level	✓	N/A	N/A	✓	IER11.IEN2	IPR90	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	N/A	—	—	—
RTC	ALM	92	0170h	Edge	✓	N/A	N/A	✓	IER11.IEN4	IPR92	—
	PRD	93	0174h	Edge	✓	N/A	N/A	✓	IER11.IEN5	IPR93	—
USBHS	USBHSR	94	0178h	Level	✓	N/A	N/A	✓	IER11.IEN6	IPR94	—
IWDT	IWUNI*2	95	017Ch	Edge	✓	N/A	N/A	✓	IER11.IEN7	IPR95	—
WDT	WUNI*2	96	0180h	Edge	✓	N/A	N/A	N/A	IER12.IEN0	IPR96	—
—	Reserved	97	0184h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	100	0190h	—	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	101	0194h	—	N/A	N/A	N/A	N/A	—	—	—

Table 15.6 Interrupt Vector Table (4/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
ICU*1	GROUPBL0_S	102	0198h	Level	✓	N/A	N/A	N/A	IER12.IEN6	IPR102	—
	GROUPBL1_S	103	019Ch	Level	✓	N/A	N/A	N/A	IER12.IEN7	IPR103	—
	GROUPAL0_S	104	01A0h	Level	✓	N/A	N/A	N/A	IER13.IEN0	IPR104	—
	GROUPAL1_S	105	01A4h	Level	✓	N/A	N/A	N/A	IER13.IEN1	IPR105	—
	GROUPAL2_S	106	01A8h	Level	✓	N/A	N/A	N/A	IER13.IEN2	IPR106	—
	GROUPAL3_S	107	01ACh	Level	✓	N/A	N/A	N/A	IER13.IEN3	IPR107	—
	GROUPAL4_S	108	01B0h	Level	✓	N/A	N/A	N/A	IER13.IEN4	IPR108	—
	GROUPAL5_S	109	01B4h	Level	✓	N/A	N/A	N/A	IER13.IEN5	IPR109	—
	GROUPAL6_S	110	01B8h	Level	✓	N/A	N/A	N/A	IER13.IEN6	IPR110	—
	GROUPBL0_NS	111	01BCh	Level	✓	N/A	N/A	N/A	IER13.IEN7	IPR111	—
	GROUPBL1_NS	112	01C0h	Level	✓	N/A	N/A	N/A	IER14.IEN0	IPR112	—
	GROUPAL0_NS	113	01C4h	Level	✓	N/A	N/A	N/A	IER14.IEN1	IPR113	—
	GROUPAL1_NS	114	01C8h	Level	✓	N/A	N/A	N/A	IER14.IEN2	IPR114	—
	GROUPAL2_NS	115	01CCh	Level	✓	N/A	N/A	N/A	IER14.IEN3	IPR115	—
	GROUPAL3_NS	116	01D0h	Level	✓	N/A	N/A	N/A	IER14.IEN4	IPR116	—
	GROUPAL4_NS	117	01D4h	Level	✓	N/A	N/A	N/A	IER14.IEN5	IPR117	—
	GROUPAL5_NS	118	01D8h	Level	✓	N/A	N/A	N/A	IER14.IEN6	IPR118	—
	GROUPAL6_NS	119	01DCh	Level	✓	N/A	N/A	N/A	IER14.IEN7	IPR119	—
DMAC	DMAC0I	120	01E0h	Edge	✓	✓	N/A	N/A	IER15.IEN0	IPR120	DTCER120
	DMAC1I	121	01E4h	Edge	✓	✓	N/A	N/A	IER15.IEN1	IPR121	DTCER121
	DMAC2I	122	01E8h	Edge	✓	✓	N/A	N/A	IER15.IEN2	IPR122	DTCER122
	DMAC3I	123	01ECh	Edge	✓	✓	N/A	N/A	IER15.IEN3	IPR123	DTCER123
	DMAC74I	124	01F0h	Level	✓	N/A	N/A	N/A	IER15.IEN4	IPR124	—
OST	OSTDI*2	125	01F4h	Edge	✓	N/A	N/A	N/A	IER15.IEN5	IPR125	—
EXDMAC	EXDMAC0I	126	01F8h	Edge	✓	✓	N/A	N/A	IER15.IEN6	IPR126	DTCER126
	EXDMAC1I	127	01FCh	Edge	✓	✓	N/A	N/A	IER15.IEN7	IPR127	DTCER127
PERIX (software configurable interrupt*3)	SLIX128 (SLI128/ IRQ16)	128	0200h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN0	IPR128	DTCER128
	SLIX129 (SLI129/ IRQ17)	129	0204h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN1	IPR129	DTCER129
	SLIX130 (SLI130/ IRQ18)	130	0208h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN2	IPR130	DTCER130
	SLIX131 (SLI131/ IRQ19)	131	020Ch	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN3	IPR131	DTCER131
	SLIX132 (SLI132/ IRQ20)	132	0210h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN4	IPR132	DTCER132
	SLIX133 (SLI133/ IRQ21)	133	0214h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN5	IPR133	DTCER133
	SLIX134 (SLI134/ IRQ22)	134	0218h	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN6	IPR134	DTCER134
	SLIX135 (SLI135/ IRQ23)	135	021Ch	Edge/ Level	✓	✓	✓	✓*4	IER16.IEN7	IPR135	DTCER135

Table 15.6 Interrupt Vector Table (5/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
PERIX (software configurable interrupt*3)	SLIX136 (SLI136/ IRQ24)	136	0220h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN0	IPR136	DTCER136
	SLIX137 (SLI137/ IRQ25)	137	0224h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN1	IPR137	DTCER137
	SLIX138 (SLI138/ IRQ26)	138	0228h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN2	IPR138	DTCER138
	SLIX139 (SLI139/ IRQ27)	139	022Ch	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN3	IPR139	DTCER139
	SLIX140 (SLI140/ IRQ28)	140	0230h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN4	IPR140	DTCER140
	SLIX141 (SLI141/ IRQ29)	141	0234h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN5	IPR141	DTCER141
	SLIX142 (SLI142/ IRQ30)	142	0238h	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN6	IPR142	DTCER142
	SLIX143 (SLI143/ IRQ31)	143	023Ch	Edge/ Level	✓	✓	✓	✓*4	IER17.IEN7	IPR143	DTCER143
PERI (software configurable interrupt*3)	SLI144	144	0240h	Edge	✓	✓	✓	N/A	IER18.IEN0	IPR144	DTCER144
	SLI145	145	0244h	Edge	✓	✓	✓	N/A	IER18.IEN1	IPR145	DTCER145
	SLI146	146	0248h	Edge	✓	✓	✓	N/A	IER18.IEN2	IPR146	DTCER146
	SLI147	147	024Ch	Edge	✓	✓	✓	N/A	IER18.IEN3	IPR147	DTCER147
	SLI148	148	0250h	Edge	✓	✓	✓	N/A	IER18.IEN4	IPR148	DTCER148
	SLI149	149	0254h	Edge	✓	✓	✓	N/A	IER18.IEN5	IPR149	DTCER149
	SLI150	150	0258h	Edge	✓	✓	✓	N/A	IER18.IEN6	IPR150	DTCER150
	SLI151	151	025Ch	Edge	✓	✓	✓	N/A	IER18.IEN7	IPR151	DTCER151
	SLI152	152	0260h	Edge	✓	✓	✓	N/A	IER19.IEN0	IPR152	DTCER152
	SLI153	153	0264h	Edge	✓	✓	✓	N/A	IER19.IEN1	IPR153	DTCER153
	SLI154	154	0268h	Edge	✓	✓	✓	N/A	IER19.IEN2	IPR154	DTCER154
	SLI155	155	026Ch	Edge	✓	✓	✓	N/A	IER19.IEN3	IPR155	DTCER155
	SLI156	156	0270h	Edge	✓	✓	✓	N/A	IER19.IEN4	IPR156	DTCER156
	SLI157	157	0274h	Edge	✓	✓	✓	N/A	IER19.IEN5	IPR157	DTCER157
	SLI158	158	0278h	Edge	✓	✓	✓	N/A	IER19.IEN6	IPR158	DTCER158
	SLI159	159	027Ch	Edge	✓	✓	✓	N/A	IER19.IEN7	IPR159	DTCER159
	SLI160	160	0280h	Edge	✓	✓	✓	N/A	IER20.IEN0	IPR160	DTCER160
	SLI161	161	0284h	Edge	✓	✓	✓	N/A	IER20.IEN1	IPR161	DTCER161
	SLI162	162	0288h	Edge	✓	✓	✓	N/A	IER20.IEN2	IPR162	DTCER162
	SLI163	163	028Ch	Edge	✓	✓	✓	N/A	IER20.IEN3	IPR163	DTCER163
	SLI164	164	0290h	Edge	✓	✓	✓	N/A	IER20.IEN4	IPR164	DTCER164
	SLI165	165	0294h	Edge	✓	✓	✓	N/A	IER20.IEN5	IPR165	DTCER165
	SLI166	166	0298h	Edge	✓	✓	✓	N/A	IER20.IEN6	IPR166	DTCER166
	SLI167	167	029Ch	Edge	✓	✓	✓	N/A	IER20.IEN7	IPR167	DTCER167
	SLI168	168	02A0h	Edge	✓	✓	✓	N/A	IER21.IEN0	IPR168	DTCER168
	SLI169	169	02A4h	Edge	✓	✓	✓	N/A	IER21.IEN1	IPR169	DTCER169
	SLI170	170	02A8h	Edge	✓	✓	✓	N/A	IER21.IEN2	IPR170	DTCER170
	SLI171	171	02ACh	Edge	✓	✓	✓	N/A	IER21.IEN3	IPR171	DTCER171

Table 15.6 Interrupt Vector Table (6/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
PERI (software configurable interrupt*3)	SLI172	172	02B0h	Edge	✓	✓	✓	N/A	IER21.IEN4	IPR172	DTCER172
	SLI173	173	02B4h	Edge	✓	✓	✓	N/A	IER21.IEN5	IPR173	DTCER173
	SLI174	174	02B8h	Edge	✓	✓	✓	N/A	IER21.IEN6	IPR174	DTCER174
	SLI175	175	02BCh	Edge	✓	✓	✓	N/A	IER21.IEN7	IPR175	DTCER175
	SLI176	176	02C0h	Edge	✓	✓	✓	N/A	IER22.IEN0	IPR176	DTCER176
	SLI177	177	02C4h	Edge	✓	✓	✓	N/A	IER22.IEN1	IPR177	DTCER177
	SLI178	178	02C8h	Edge	✓	✓	✓	N/A	IER22.IEN2	IPR178	DTCER178
	SLI179	179	02CCh	Edge	✓	✓	✓	N/A	IER22.IEN3	IPR179	DTCER179
	SLI180	180	02D0h	Edge	✓	✓	✓	N/A	IER22.IEN4	IPR180	DTCER180
	SLI181	181	02D4h	Edge	✓	✓	✓	N/A	IER22.IEN5	IPR181	DTCER181
	SLI182	182	02D8h	Edge	✓	✓	✓	N/A	IER22.IEN6	IPR182	DTCER182
	SLI183	183	02DCh	Edge	✓	✓	✓	N/A	IER22.IEN7	IPR183	DTCER183
	SLI184	184	02E0h	Edge	✓	✓	✓	N/A	IER23.IEN0	IPR184	DTCER184
	SLI185	185	02E4h	Edge	✓	✓	✓	N/A	IER23.IEN1	IPR185	DTCER185
	SLI186	186	02E8h	Edge	✓	✓	✓	N/A	IER23.IEN2	IPR186	DTCER186
	SLI187	187	02ECh	Edge	✓	✓	✓	N/A	IER23.IEN3	IPR187	DTCER187
	SLI188	188	02F0h	Edge	✓	✓	✓	N/A	IER23.IEN4	IPR188	DTCER188
	SLI189	189	02F4h	Edge	✓	✓	✓	N/A	IER23.IEN5	IPR189	DTCER189
	SLI190	190	02F8h	Edge	✓	✓	✓	N/A	IER23.IEN6	IPR190	DTCER190
	SLI191	191	02FCh	Edge	✓	✓	✓	N/A	IER23.IEN7	IPR191	DTCER191
	SLI192	192	0300h	Edge	✓	✓	✓	N/A	IER24.IEN0	IPR192	DTCER192
	SLI193	193	0304h	Edge	✓	✓	✓	N/A	IER24.IEN1	IPR193	DTCER193
	SLI194	194	0308h	Edge	✓	✓	✓	N/A	IER24.IEN2	IPR194	DTCER194
	SLI195	195	030Ch	Edge	✓	✓	✓	N/A	IER24.IEN3	IPR195	DTCER195
	SLI196	196	0310h	Edge	✓	✓	✓	N/A	IER24.IEN4	IPR196	DTCER196
	SLI197	197	0314h	Edge	✓	✓	✓	N/A	IER24.IEN5	IPR197	DTCER197
	SLI198	198	0318h	Edge	✓	✓	✓	N/A	IER24.IEN6	IPR198	DTCER198
	SLI199	199	031Ch	Edge	✓	✓	✓	N/A	IER24.IEN7	IPR199	DTCER199
	SLI200	200	0320h	Edge	✓	✓	✓	N/A	IER25.IEN0	IPR200	DTCER200
	SLI201	201	0324h	Edge	✓	✓	✓	N/A	IER25.IEN1	IPR201	DTCER201
	SLI202	202	0328h	Edge	✓	✓	✓	N/A	IER25.IEN2	IPR202	DTCER202
	SLI203	203	032Ch	Edge	✓	✓	✓	N/A	IER25.IEN3	IPR203	DTCER203
	SLI204	204	0330h	Edge	✓	✓	✓	N/A	IER25.IEN4	IPR204	DTCER204
	SLI205	205	0334h	Edge	✓	✓	✓	N/A	IER25.IEN5	IPR205	DTCER205
	SLI206	206	0338h	Edge	✓	✓	✓	N/A	IER25.IEN6	IPR206	DTCER206
	SLI207	207	033Ch	Edge	✓	✓	✓	N/A	IER25.IEN7	IPR207	DTCER207
	SLI208	208	0340h	Edge	✓	✓	✓	N/A	IER26.IEN0	IPR208	DTCER208
	SLI209	209	0344h	Edge	✓	✓	✓	N/A	IER26.IEN1	IPR209	DTCER209
	SLI210	210	0348h	Edge	✓	✓	✓	N/A	IER26.IEN2	IPR210	DTCER210
	SLI211	211	034Ch	Edge	✓	✓	✓	N/A	IER26.IEN3	IPR211	DTCER211
	SLI212	212	0350h	Edge	✓	✓	✓	N/A	IER26.IEN4	IPR212	DTCER212
	SLI213	213	0354h	Edge	✓	✓	✓	N/A	IER26.IEN5	IPR213	DTCER213

Table 15.6 Interrupt Vector Table (7/7)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	IER	IPR	DTCER
PERI (software configurable interrupt*3)	SLI214	214	0358h	Edge	✓	✓	✓	N/A	IER26.IEN6	IPR214	DTCER214
	SLI215	215	035Ch	Edge	✓	✓	✓	N/A	IER26.IEN7	IPR215	DTCER215
	SLI216	216	0360h	Edge	✓	✓	✓	N/A	IER27.IEN0	IPR216	DTCER216
	SLI217	217	0364h	Edge	✓	✓	✓	N/A	IER27.IEN1	IPR217	DTCER217
	SLI218	218	0368h	Edge	✓	✓	✓	N/A	IER27.IEN2	IPR218	DTCER218
	SLI219	219	036Ch	Edge	✓	✓	✓	N/A	IER27.IEN3	IPR219	DTCER219
	SLI220	220	0370h	Edge	✓	✓	✓	N/A	IER27.IEN4	IPR220	DTCER220
	SLI221	221	0374h	Edge	✓	✓	✓	N/A	IER27.IEN5	IPR221	DTCER221
	SLI222	222	0378h	Edge	✓	✓	✓	N/A	IER27.IEN6	IPR222	DTCER222
	SLI223	223	037Ch	Edge	✓	✓	✓	N/A	IER27.IEN7	IPR223	DTCER223
	SLI224	224	0380h	Edge	✓	✓	✓	N/A	IER28.IEN0	IPR224	DTCER224
	SLI225	225	0384h	Edge	✓	✓	✓	N/A	IER28.IEN1	IPR225	DTCER225
	SLI226	226	0388h	Edge	✓	✓	✓	N/A	IER28.IEN2	IPR226	DTCER226
	SLI227	227	038Ch	Edge	✓	✓	✓	N/A	IER28.IEN3	IPR227	DTCER227
	SLI228	228	0390h	Edge	✓	✓	✓	N/A	IER28.IEN4	IPR228	DTCER228
	SLI229	229	0394h	Edge	✓	✓	✓	N/A	IER28.IEN5	IPR229	DTCER229
	SLI230	230	0398h	Edge	✓	✓	✓	N/A	IER28.IEN6	IPR230	DTCER230
	SLI231	231	039Ch	Edge	✓	✓	✓	N/A	IER28.IEN7	IPR231	DTCER231
	SLI232	232	03A0h	Edge	✓	✓	✓	N/A	IER29.IEN0	IPR232	DTCER232
	SLI233	233	03A4h	Edge	✓	✓	✓	N/A	IER29.IEN1	IPR233	DTCER233
	SLI234	234	03A8h	Edge	✓	✓	✓	N/A	IER29.IEN2	IPR234	DTCER234
	SLI235	235	03ACh	Edge	✓	✓	✓	N/A	IER29.IEN3	IPR235	DTCER235
	SLI236	236	03B0h	Edge	✓	✓	✓	N/A	IER29.IEN4	IPR236	DTCER236
	SLI237	237	03B4h	Edge	✓	✓	✓	N/A	IER29.IEN5	IPR237	DTCER237
	SLI238	238	03B8h	Edge	✓	✓	✓	N/A	IER29.IEN6	IPR238	DTCER238
	SLI239	239	03BCh	Edge	✓	✓	✓	N/A	IER29.IEN7	IPR239	DTCER239
	SLI240	240	03C0h	Edge	✓	✓	✓	N/A	IER30.IEN0	IPR240	DTCER240
	SLI241	241	03C4h	Edge	✓	✓	✓	N/A	IER30.IEN1	IPR241	DTCER241
	SLI242	242	03C8h	Edge	✓	✓	✓	N/A	IER30.IEN2	IPR242	DTCER242
	SLI243	243	03CCh	Edge	✓	✓	✓	N/A	IER30.IEN3	IPR243	DTCER243
	SLI244	244	03D0h	Edge	✓	✓	✓	N/A	IER30.IEN4	IPR244	DTCER244
	SLI245	245	03D4h	Edge	✓	✓	✓	N/A	IER30.IEN5	IPR245	DTCER245
	SLI246	246	03D8h	Edge	✓	✓	✓	N/A	IER30.IEN6	IPR246	DTCER246
	SLI247	247	03DCh	Edge	✓	✓	✓	N/A	IER30.IEN7	IPR247	DTCER247
	SLI248	248	03E0h	Edge	✓	✓	✓	N/A	IER31.IEN0	IPR248	DTCER248
	SLI249	249	03E4h	Edge	✓	✓	✓	N/A	IER31.IEN1	IPR249	DTCER249
	SLI250	250	03E8h	Edge	✓	✓	✓	N/A	IER31.IEN2	IPR250	DTCER250
	SLI251	251	03ECh	Edge	✓	✓	✓	N/A	IER31.IEN3	IPR251	DTCER251
	SLI252	252	03F0h	Edge	✓	✓	✓	N/A	IER31.IEN4	IPR252	DTCER252
	SLI253	253	03F4h	Edge	✓	✓	✓	N/A	IER31.IEN5	IPR253	DTCER253
	SLI254	254	03F8h	Edge	✓	✓	✓	N/A	IER31.IEN6	IPR254	DTCER254
	SLI255	255	03FCh	Edge	✓	✓	✓	N/A	IER31.IEN7	IPR255	DTCER255

Note: This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, refer to Table 1.2, Comparison of Functions for Different Packages.
Note 1. For the group interrupt sources, refer to Table 15.8, Group Interrupt Requests.
Note 2. This is the case where the corresponding non-maskable interrupt enable bit is set to 0 (disabled).
Note 3. For the software configurable interrupt sources, refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt. Note that some interrupt sources cannot start the DTC, DMAC, or EXDMAC.‘
Note 4. Only external pin interrupts can be used as a source to exit software standby mode.

15.3.2 Fast Interrupt Vector Area

The interrupt set as the fast interrupt uses the FINTV register in the CPU. Set the FINTV register before enabling the fast interrupt.

15.3.3 Non-maskable Interrupt Vector Area

Non-maskable interrupts use the vector area in the exception vector table.
The exception vector table is allocated to the 128-byte area (4 bytes × 32 sources) beginning with the address set in the EXTB register in the CPU. Set the EXTB register before enabling non-maskable interrupts. Set a multiple of 4 in the EXTB register.

15.4 Types of Interrupts

Interrupts are divided into maskable interrupts and non-maskable interrupts. Maskable interrupts can be masked by the PSW.I bit or IPL[4:0] bits of the processor status word in the CPU. Non-maskable interrupts can be accepted by the CPU regardless of those bits. While interrupt sources assigned to vector numbers 0 to 127 are fixed, an interrupt source assigned to each vector number from 128 to 255 (software configurable interrupt) can be selected from multiple sources. Note that maskable interrupts are referred to as interrupts in this chapter.
Figure 15.2 shows types of interrupts.

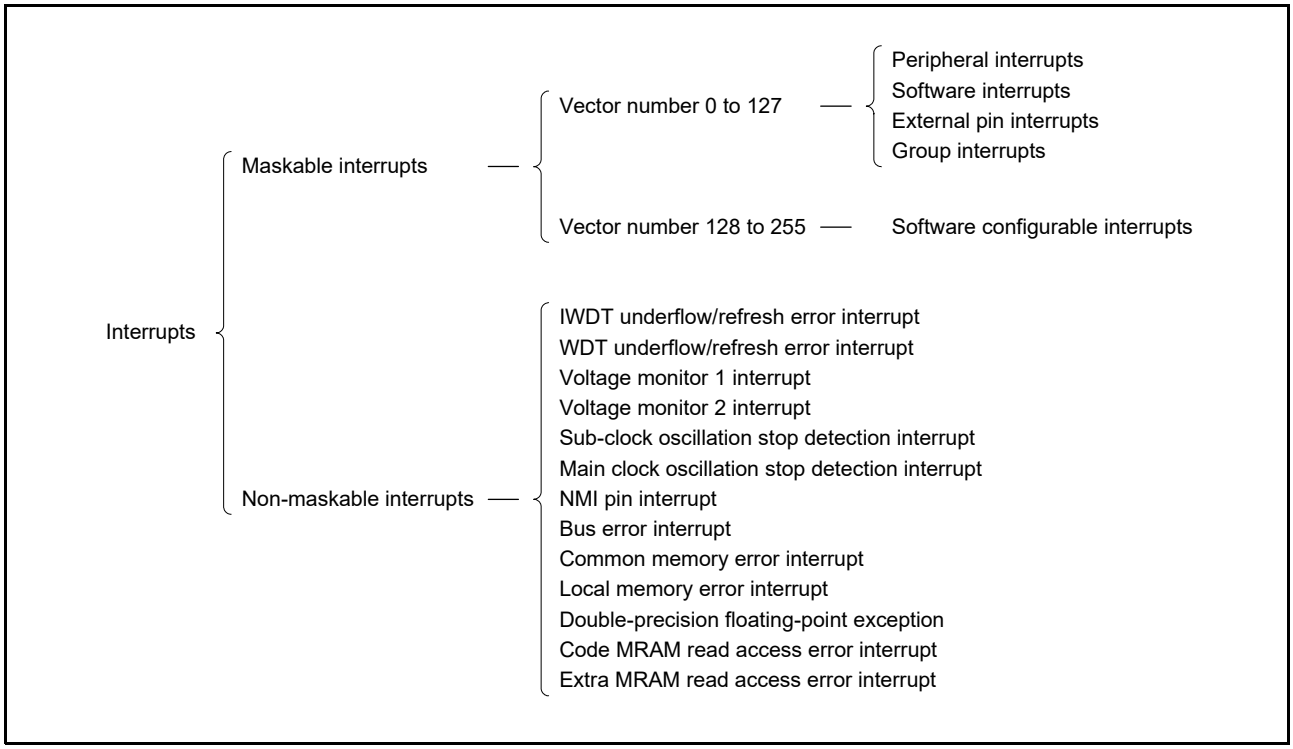


Figure 15.2 Types of Interrupts

15.4.1 Peripheral Interrupts

Peripheral interrupt is generated by peripherals. Peripheral interrupts sources assigned to vector numbers 0 to 127 cannot be assigned to the software configurable interrupts. Refer to section 15.4.5, Software Configurable Interrupts for details on software configurable interrupts.

15.4.2 Software Interrupts

When the SWINTR.SWINT bit and SWINT2R.SWINT2 bit are set to 1, the SWINT interrupt and SWINT2 interrupt occur, respectively.

15.4.3 External Pin Interrupt

An external pin interrupt is generated by signals input to the IRQi pin (i = 0 to 31). Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupt.

15.4.4 Group Interrupts

Multiple peripheral interrupt requests (up to 32 requests) are grouped together as one interrupt request. Interrupts are grouped depending on the peripheral operating clock (ICLK, PCLKB, or PCLKA) and the security attribute.

(1) Types of Group Interrupts

Table 15.7 lists types of group interrupts.

Table 15.7 Types of Group Interrupts

Interrupt Vector Number	Interrupt Name	Group Interrupt Source	
		Peripheral operating clock	Security attribute
17	GROUPIL0_S	ICLK	Secure
18	GROUPIL0_NS		Non-secure
19	GROUPIL1_S		Secure
20	GROUPIL1_NS		Non-secure
102	GROUPBL0_S	PCLKB	Secure
103	GROUPBL1_S		
104	GROUPAL0_S	PCLKA	
105	GROUPAL1_S		
106	GROUPAL2_S		
107	GROUPAL3_S		
108	GROUPAL4_S		
109	GROUPAL5_S		
110	GROUPAL6_S		
111	GROUPBL0_NS	PCLKB	Non-secure
112	GROUPBL1_NS		
113	GROUPAL0_NS	PCLKA	
114	GROUPAL1_NS		
115	GROUPAL2_NS		
116	GROUPAL3_NS		
117	GROUPAL4_NS		
118	GROUPAL5_NS		
119	GROUPAL6_NS		

(2) Configuration of Group Interrupts

When an interrupt request is generated while the corresponding ENj bit in the group interrupt request enable register (GENIL0, GENIL1, GENBL0, GENBL1, and GENAL0 to GENAL6*1) is 1, the ISj flag in the group interrupt request register (GRPIL0, GRPIL1, GRPBL0, GRPBL1, and GRPAL0 to GRPAL6*1) becomes 1 (j = 0 to 31).

Note 1. There is no register in the group to which no interrupt source is assigned.

Figure 15.3 shows the configuration of group interrupts.

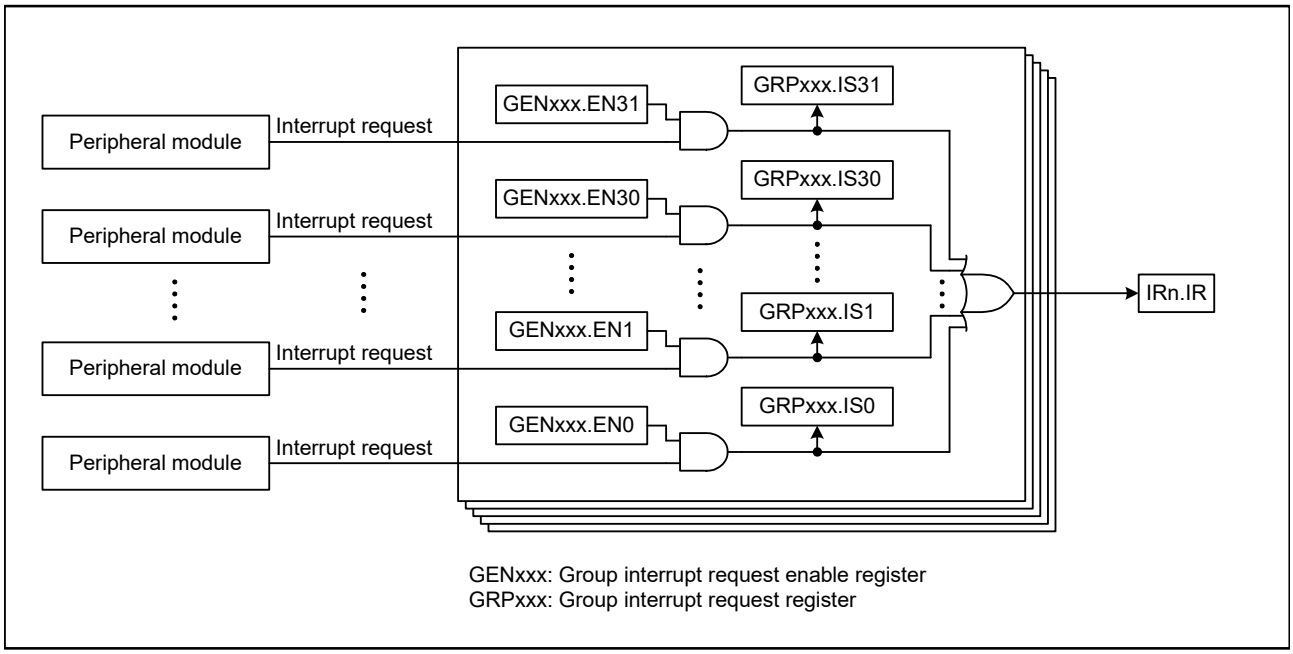


Figure 15.3 Group Interrupt Configuration (n = 17 to 20, 102 to 119)

(3) Group Interrupt Sources

Table 15.8 lists peripheral interrupt sources that are assigned to group interrupts.

Table 15.8 Group Interrupt Requests (1/5)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Vector No. (IRn.IR)
IL0	0	DPFPU	DPFPUEX (double-precision floating-point exception)	GENIL0.EN0	GRPIL0.IS0	17 or 18
	1	LM0	ERR (local memory error)	GENIL0.EN1	GRPIL0.IS1	
	2	CM	ERR (common memory error)	GENIL0.EN2	GRPIL0.IS2	
	3	BUS	HLDTO (external bus hold timeout)	GENIL0.EN3	GRPIL0.IS3	
	4 to 7	Reserved	—	—	—	
	8	MRAM	MRCRDI (code MRAM ECC error)	GENIL0.EN8	GRPIL0.IS8	
	9		MRERDI (extra MRAM ECC error)	GENIL0.EN9	GRPIL0.IS9	
	10		MRCPRI (code MRAM access error)	GENIL0.EN10	GRPIL0.IS10	
	11		MRDPRI (data MRAM access error)	GENIL0.EN11	GRPIL0.IS11	
	12		MREPRI (extra MRAM access error)	GENIL0.EN12	GRPIL0.IS12	
	13 to 31	Reserved	—	—	—	
IL1	0 to 31	Reserved	—	—	—	19 or 20
BL0	0	SDHI	CDETI (card detection interrupt)	GENBL0.EN0	GRPBL0.IS0	102 or 111
	1		CACI (card access interrupt)	GENBL0.EN1	GRPBL0.IS1	
	2		SDACI (SDIO access interrupt)	GENBL0.EN2	GRPBL0.IS2	
	3	XSPI0	CMP (complete interrupt)	GENBL0.EN3	GRPBL0.IS3	
	4		ERR (error interrupt)	GENBL0.EN4	GRPBL0.IS4	
	5	XSPI1	CMP (complete interrupt)	GENBL0.EN5	GRPBL0.IS5	
	6		ERR (error interrupt)	GENBL0.EN6	GRPBL0.IS6	
	7	SSIE0	SSIF0 (status interrupt)	GENBL0.EN7	GRPBL0.IS7	
	8	SSIE1	SSIF1 (status interrupt)	GENBL0.EN8	GRPBL0.IS8	
	9	PDC	PCFEI (frame end interrupt)	GENBL0.EN9	GRPBL0.IS9	
	10		PCERI (error interrupt)	GENBL0.EN10	GRPBL0.IS10	
	11	PDMIF	ERR0 (error interrupt)	GENBL0.EN11	GRPBL0.IS11	
	12 to 31	Reserved	—	—	—	
BL1	0	CAC0	FERRI (frequency error)	GENBL1.EN0	GRPBL1.IS0	103 or 112
	1		MENDI (measurement end)	GENBL1.EN1	GRPBL1.IS1	
	2		OVFI (overflow interrupt)	GENBL1.EN2	GRPBL1.IS2	
	3	CAC1	FERRI (frequency error)	GENBL1.EN3	GRPBL1.IS3	
	4		MENDI (measurement end)	GENBL1.EN4	GRPBL1.IS4	
	5		OVFI (overflow interrupt)	GENBL1.EN5	GRPBL1.IS5	
	6	CAC2	FERRI (frequency error)	GENBL1.EN6	GRPBL1.IS6	
	7		MENDI (measurement end)	GENBL1.EN7	GRPBL1.IS7	
	8		OVFI (overflow interrupt)	GENBL1.EN8	GRPBL1.IS8	
	9 to 31	Reserved	—	—	—	

Table 15.8 Group Interrupt Requests (2/5)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Vector No. (IRn.IR)
AL0	0	RSCI0	TEI (transmission end/condition generated)	GENAL0.EN0	GRPAL0.IS0	104 or 113
	1		ERI (receive error)	GENAL0.EN1	GRPAL0.IS1	
	2		BFD (break field detect)	GENAL0.EN2	GRPAL0.IS2	
	3	RSCI1	TEI (transmission end/condition generated)	GENAL0.EN3	GRPAL0.IS3	
	4		ERI (receive error)	GENAL0.EN4	GRPAL0.IS4	
	5		BFD (break field detect)	GENAL0.EN5	GRPAL0.IS5	
	6	RSCI2	TEI (transmission end/condition generated)	GENAL0.EN6	GRPAL0.IS6	
	7		ERI (receive error)	GENAL0.EN7	GRPAL0.IS7	
	8		BFD (break field detect)	GENAL0.EN8	GRPAL0.IS8	
	9	RSCI3	TEI (transmission end/condition generated)	GENAL0.EN9	GRPAL0.IS9	
	10		ERI (receive error)	GENAL0.EN10	GRPAL0.IS10	
	11		BFD (break field detect)	GENAL0.EN11	GRPAL0.IS11	
	12	RSCI4	TEI (transmission end/condition generated)	GENAL0.EN12	GRPAL0.IS12	
	13		ERI (receive error)	GENAL0.EN13	GRPAL0.IS13	
	14		BFD (break field detect)	GENAL0.EN14	GRPAL0.IS14	
	15	RSCI5	TEI (transmission end/condition generated)	GENAL0.EN15	GRPAL0.IS15	
	16		ERI (receive error)	GENAL0.EN16	GRPAL0.IS16	
	17		BFD (break field detect)	GENAL0.EN17	GRPAL0.IS17	
	18	RSCI6	TEI (transmission end/condition generated)	GENAL0.EN18	GRPAL0.IS18	
	19		ERI (receive error)	GENAL0.EN19	GRPAL0.IS19	
	20		BFD (break field detect)	GENAL0.EN20	GRPAL0.IS20	
	21	RSCI7	TEI (transmission end/condition generated)	GENAL0.EN21	GRPAL0.IS21	
	22		ERI (receive error)	GENAL0.EN22	GRPAL0.IS22	
	23		BFD (break field detect)	GENAL0.EN23	GRPAL0.IS23	
	24	RSCI8	TEI (transmission end/condition generated)	GENAL0.EN24	GRPAL0.IS24	
	25		ERI (receive error)	GENAL0.EN25	GRPAL0.IS25	
	26		BFD (break field detect)	GENAL0.EN26	GRPAL0.IS26	
	27	RSCI9	TEI (transmission end/condition generated)	GENAL0.EN27	GRPAL0.IS27	
	28		ERI (receive error)	GENAL0.EN28	GRPAL0.IS28	
	29		BFD (break field detect)	GENAL0.EN29	GRPAL0.IS29	
	30, 31	Reserved	—	—	—	

Table 15.8 Group Interrupt Requests (3/5)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Vector No. (IRn.IR)
AL1	0	RSCI10	TEI (transmission end/condition generated)	GENAL1.EN0	GRPAL1.IS0	105 or 114
	1		ERI (receive error)	GENAL1.EN1	GRPAL1.IS1	
	2		BFD (break field detect)	GENAL1.EN2	GRPAL1.IS2	
	3	RSCI11	TEI (transmission end/condition generated)	GENAL1.EN3	GRPAL1.IS3	
	4		ERI (receive error)	GENAL1.EN4	GRPAL1.IS4	
	5		BFD (break field detect)	GENAL1.EN5	GRPAL1.IS5	
	6	RSCI12	TEI (transmission end/condition generated)	GENAL1.EN6	GRPAL1.IS6	
	7		ERI (receive error)	GENAL1.EN7	GRPAL1.IS7	
	8		BFD (break field detect)	GENAL1.EN8	GRPAL1.IS8	
	9, 10	Reserved	—	—	—	
	11	RIICHS1	TEI (transmission end)	GENAL1.EN11	GRPAL1.IS11	
	12		EEI (communication error/communication event)	GENAL1.EN12	GRPAL1.IS12	
	13	RIICHS2	TEI (transmission end)	GENAL1.EN13	GRPAL1.IS13	
	14		EEI (communication error/communication event)	GENAL1.EN14	GRPAL1.IS14	
	15	RI3C0	TEI (transmission end)	GENAL1.EN15	GRPAL1.IS15	
	16		EEI (communication error/communication event)	GENAL1.EN16	GRPAL1.IS16	
	17 to 22	Reserved	—	—	—	
	23	RSPIA0	SPII (idle interrupt)	GENAL1.EN23	GRPAL1.IS23	
	24		SPEI (error interrupt)	GENAL1.EN24	GRPAL1.IS24	
	25	RSPIA1	SPII (idle interrupt)	GENAL1.EN25	GRPAL1.IS25	
	26		SPEI (error interrupt)	GENAL1.EN26	GRPAL1.IS26	
	27	RSPIA2	SPII (idle interrupt)	GENAL1.EN27	GRPAL1.IS27	
	28		SPEI (error interrupt)	GENAL1.EN28	GRPAL1.IS28	
	29	Reserved	—	—	—	
	30	DOC	DOPCI (data operation circuit interrupt)	GENAL1.EN30	GRPAL1.IS30	
	31	Reserved	—	—	—	
AL2	0	CANFDG0	RFRI (receive FIFO)	GENAL2.EN0	GRPAL2.IS0	106 or 115
	1		GLEI (global error)	GENAL2.EN1	GRPAL2.IS1	
	2	CANFD0	CHTI (channel transmit)	GENAL2.EN2	GRPAL2.IS2	
	3		CHEI (channel error)	GENAL2.EN3	GRPAL2.IS3	
	4		CHRI (channel receive)	GENAL2.EN4	GRPAL2.IS4	
	5	CANFD1	CHTI (channel transmit)	GENAL2.EN5	GRPAL2.IS5	
	6		CHEI (channel error)	GENAL2.EN6	GRPAL2.IS6	
	7		CHRI (channel receive)	GENAL2.EN7	GRPAL2.IS7	
	8 to 15	Reserved	—	—	—	
	16	CANFD2	CHEI (channel error)	GENAL2.EN16	GRPAL2.IS16	
	17		CFRI (common FIFO receive)	GENAL2.EN17	GRPAL2.IS17	
	18	CANFD	GLEI (global error)	GENAL2.EN18	GRPAL2.IS18	
	19		RFRI (receive FIFO)	GENAL2.EN19	GRPAL2.IS19	
	20	CANFD2	CHTI (channel transmit)	GENAL2.EN20	GRPAL2.IS20	
	21	CANFD	RMRI (receive message buffer)	GENAL2.EN21	GRPAL2.IS21	
	22 to 31	Reserved	—	—	—	

Table 15.8 Group Interrupt Requests (4/5)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Vector No. (IRn.IR)
AL3	0 to 15	Reserved	—	—	—	107 or 116
	16	POEG	POEGGAI (group A interrupt)	GENAL3.EN16	GRPAL3.IS16	
	17		POEGGBI (group B interrupt)	GENAL3.EN17	GRPAL3.IS17	
	18		POEGGCI (group C interrupt)	GENAL3.EN18	GRPAL3.IS18	
	19		POEGGDI (group D interrupt)	GENAL3.EN19	GRPAL3.IS19	
	20 to 31	Reserved	—	—	—	
AL4	0	ESWM	FWEI (MFWD error interrupt)	GENAL4.EN0	GRPAL4.IS0	108 or 117
	1		CAEI (COMA error interrupt)	GENAL4.EN1	GRPAL4.IS1	
	2		GWEI0 (GWCA0 error interrupt)	GENAL4.EN2	GRPAL4.IS2	
	3		EAEI0 (ETHA0 error interrupt)	GENAL4.EN3	GRPAL4.IS3	
	4		Reserved	GENAL4.EN4	GRPAL4.IS4	
	5		PTPSI0 (gPTP timer status interrupt 0)	GENAL4.EN5	GRPAL4.IS5	
	6		PTPSI1 (gPTP timer status interrupt 1)	GENAL4.EN6	GRPAL4.IS6	
	7		FWSI (MFWD status interrupt)	GENAL4.EN7	GRPAL4.IS7	
	8		Reserved	GENAL4.EN8	GRPAL4.IS8	
	9		CASI (COMA status interrupt)	GENAL4.EN9	GRPAL4.IS9	
	10		EASI0 (ETHA0 status interrupt)	GENAL4.EN10	GRPAL4.IS10	
	11		Reserved	GENAL4.EN11	GRPAL4.IS11	
	12		GWDI00 (GWCA0 data interrupt 0)	GENAL4.EN12	GRPAL4.IS12	
	13		GWDI01 (GWCA0 data interrupt 1)	GENAL4.EN13	GRPAL4.IS13	
	14		GWDI02 (GWCA0 data interrupt 2)	GENAL4.EN14	GRPAL4.IS14	
	15		GWDI03 (GWCA0 data interrupt 3)	GENAL4.EN15	GRPAL4.IS15	
	16		GWDI04 (GWCA0 data interrupt 4)	GENAL4.EN16	GRPAL4.IS16	
	17		GWDI05 (GWCA0 data interrupt 5)	GENAL4.EN17	GRPAL4.IS17	
	18		GWDI06 (GWCA0 data interrupt 6)	GENAL4.EN18	GRPAL4.IS18	
	19		GWDI07 (GWCA0 data interrupt 7)	GENAL4.EN19	GRPAL4.IS19	
	20		TSDI00 (GWCA0 timestamp interrupt 0)	GENAL4.EN20	GRPAL4.IS20	
	21		TSDI01 (GWCA0 timestamp interrupt 1)	GENAL4.EN21	GRPAL4.IS21	
	22		MDIO0 (ETHA0 MDIO interrupt)	GENAL4.EN22	GRPAL4.IS22	
	23		Reserved	GENAL4.EN23	GRPAL4.IS23	
	24		RMPI0 (RMAC0 PHY interrupt)	GENAL4.EN24	GRPAL4.IS24	
	25		Reserved	GENAL4.EN25	GRPAL4.IS25	
	26 to 31	Reserved	—	—	—	

Table 15.8 Group Interrupt Requests (5/5)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Vector No. (IRn.IR)
AL5	0	ADC (ADC16H)	LIMCLPI (limiter clip)	GENAL5.EN0	GRPAL5.IS0	109 or 118
	1		FIFOOVF (FIFO overflow)	GENAL5.EN1	GRPAL5.IS1	
	2		CMPI0 (compare match 0)	GENAL5.EN2	GRPAL5.IS2	
	3		CMPI1 (compare match 1)	GENAL5.EN3	GRPAL5.IS3	
	4		ERR0 (ADC0 error)	GENAL5.EN4	GRPAL5.IS4	
	5		RESOVF0 (ADC0 conversion result overflow)	GENAL5.EN5	GRPAL5.IS5	
	6		Reserved	—	—	
	7		CALEND0 (ADC0 self-calibration end)	GENAL5.EN7	GRPAL5.IS7	
	8		CMPI2 (compare match 2)	GENAL5.EN8	GRPAL5.IS8	
	9		CMPI3 (compare match 3)	GENAL5.EN9	GRPAL5.IS9	
	10		ERR1 (ADC1 error)	GENAL5.EN10	GRPAL5.IS10	
	11		RESOVF1 (ADC1 conversion result overflow)	GENAL5.EN11	GRPAL5.IS11	
	12		Reserved	—	—	
	13		CALEND1 (ADC1 self-calibration end)	GENAL5.EN13	GRPAL5.IS13	
	14		CMPI4 (compare match 4)	GENAL5.EN14	GRPAL5.IS14	
	15		CMPI5 (compare match 5)	GENAL5.EN15	GRPAL5.IS15	
	16		ERR2 (ADC2 error)	GENAL5.EN16	GRPAL5.IS16	
	17		RESOVF2 (ADC2 conversion result overflow)	GENAL5.EN17	GRPAL5.IS17	
	18		Reserved	—	—	
	19		CALEND2 (ADC2 self-calibration end)	GENAL5.EN19	GRPAL5.IS19	
	20		CMPI6 (compare match 6)	GENAL5.EN20	GRPAL5.IS20	
	21		CMPI7 (compare match 7)	GENAL5.EN21	GRPAL5.IS21	
	22 to 31	Reserved	—	—	—	
AL6	0 to 31	Reserved	—	—	—	110 or 119

15.4.5 Software Configurable Interrupts

An interrupt source assigned to each interrupt vector number from 128 to 255 can be selected from multiple sources. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt for interrupt sources that can be assigned to software configurable interrupt. Figure 15.4 shows the software configurable interrupt configuration.

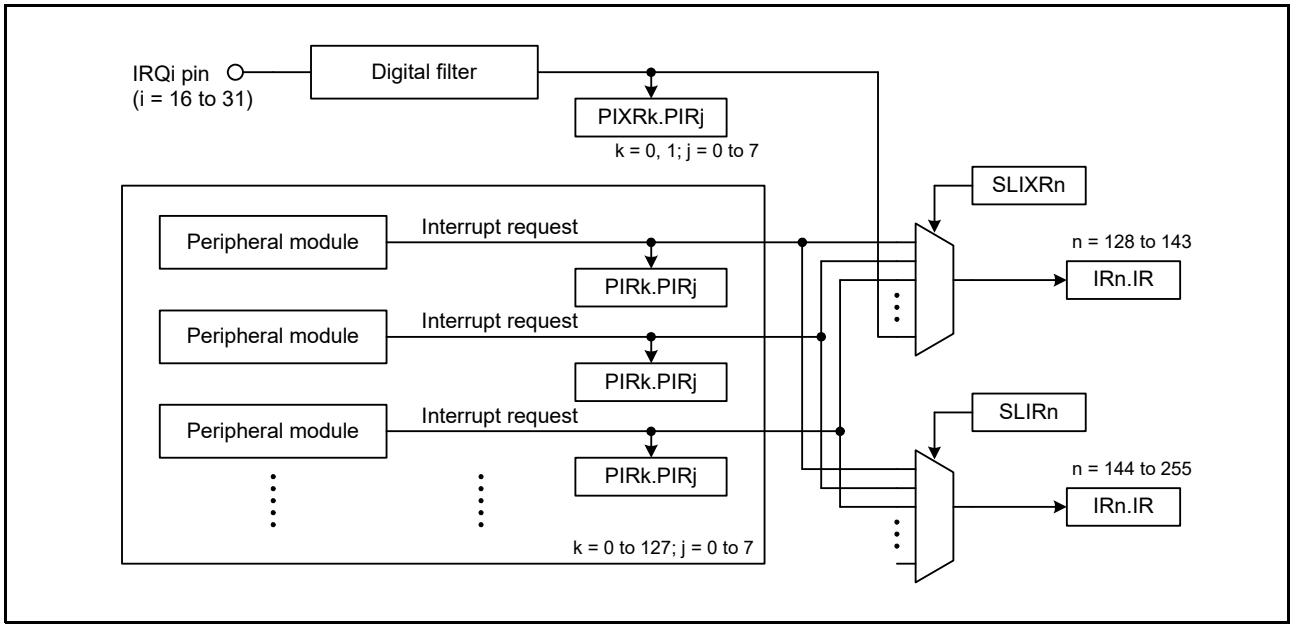


Figure 15.4 Software Configurable Interrupt Configuration

15.4.6 Non-Maskable Interrupts

Non-maskable interrupts include the 13-types of interrupts listed in Table 15.9.

Table 15.9 Types of Non-Maskable Interrupts

Name	Description
IWDT underflow/refresh error interrupt	Underflow or refresh error has occurred in IWDT.
WDT underflow/refresh error interrupt	Underflow or refresh error has occurred in WDT.
Voltage monitoring 1 interrupt	The interrupt generation condition set in PVD1 has been met.
Voltage monitoring 2 interrupt	The interrupt generation condition set in PVD2 has been met.
Sub-clock oscillation stop detection interrupt	The sub-clock oscillator has stopped oscillating.
Main clock oscillation stop detection interrupt	The main clock oscillator has stopped oscillating.
NMI pin interrupt	The specified edge is input from the NMI pin.
Bus error interrupt	Bus error has occurred.
Common memory error interrupt	An ECC error has occurred in SRAM0 or SRAM1.
Local memory error interrupt	An ECC error has occurred in CPU cache or LRAM.
Double-precision floating-point exception	An exception has occurred in the double-precision floating-point coprocessor.
Code MRAM read access error interrupt	An ECC error has occurred in the code area of MRAM.
Extra MRAM read access error interrupt	An ECC error has occurred in the extra area of MRAM.

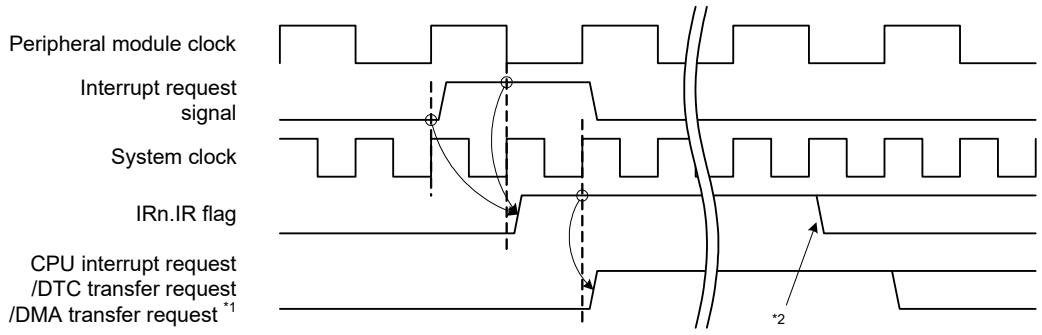
Non-maskable interrupts have the highest priority in all interrupts, including the fast interrupt, so are accepted regardless of the settings of the PSW.I bit (interrupt enable) and IPL[4:0] bits (processor interrupt priority level) in the CPU.
The NMISR register can be used to confirm whether a non-maskable interrupt is generated.
Only the CPU can be selected as the interrupt request destination of non-maskable interrupt. The DTC and DMAC cannot be selected.

15.5 Interrupt Detection

Level detection or edge detection can be used for detecting an interrupt request.
For interrupt requests from the peripherals, edge detection or level detection is fixed for each interrupt source. For interrupt requests of the external pin interrupt, edge detection or level detection can be selected with the IRQCRi.IRQMD[1:0] bits (i = 0 to 31).
Refer to Table 15.6, Interrupt Vector Table for the method to detect each interrupt request.
All interrupt requests by interrupt sources assigned to the group interrupts are level-detected, and the group interrupts themselves (GROUPIL0, GROUPIL1, GROUPBL0, GROUPBL1, and GROUPAL0 to GROUPAL6) are also level-detected.
Refer to section 15.4.4, Group Interrupts for group interrupts. Refer to section 15.5.3, Interrupt Detection for Group Interrupts for interrupt requests of group interrupts.

15.5.1 Edge Detection

Figure 15.5 shows the operation of the IRn.IR flag at edge detection (n = 26 to 255).
The IRn.IR flag becomes 1 when the rising edge of the interrupt request signal is detected. Then, the IRn.IR flag does not become 0 by disabling the interrupt request of the peripheral module. When the CPU accepts the interrupt request or the DTC/DMAC accepts the transfer request, the IRn.IR flag automatically becomes 0. It is not required to set the IRn.IR flag to 0 by software. Refer to Table 15.10, Operations When Starting the DTC/DMAC for details on clearing the IRn.IR flag by DTC/DMAC.
For the external pin interrupts of interrupt vector number 64 to 79, software configurable interrupts of interrupt vector number 128 to 143 with external pin interrupts selected, and interrupt sources of interrupt vector number 80 to 95, the timing when the IRn.IR flag becomes 1 after an interrupt signal occurs is different from the other interrupts. For the external pin interrupts, the timing is delayed for internal delay plus two cycles of PCLKB after a signal is input to the IRQ pin (i = 0 to 31). For interrupts of interrupt vector number 80 to 95, the timing is delayed for two cycles of PCLKB.



Note 1. A CPU interrupt request, DTC transfer request, or DMA transfer request is generated. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure.
Note 2. For a CPU interrupt request, the IRn.IR flag becomes 0 when the CPU accepts an interrupt request. For the timing for a DTC transfer request or DMA transfer request, refer to Table 15.10, Operations When Starting the DTC/DMAC.

Figure 15.5 IRn.IR Flag Operation for Interrupts Detected by Edge Detection

(1) Detecting Consecutive Interrupt Request Signals

When interrupt request signals occur every cycle, the latter interrupt signal cannot be detected. To accept consecutive interrupt request signals, the interval of at least two cycles of the system clock or peripheral module clock, whichever is lower, is required between interrupt signals. Figure 15.6 shows the interval for accepting consecutive interrupt request signals.

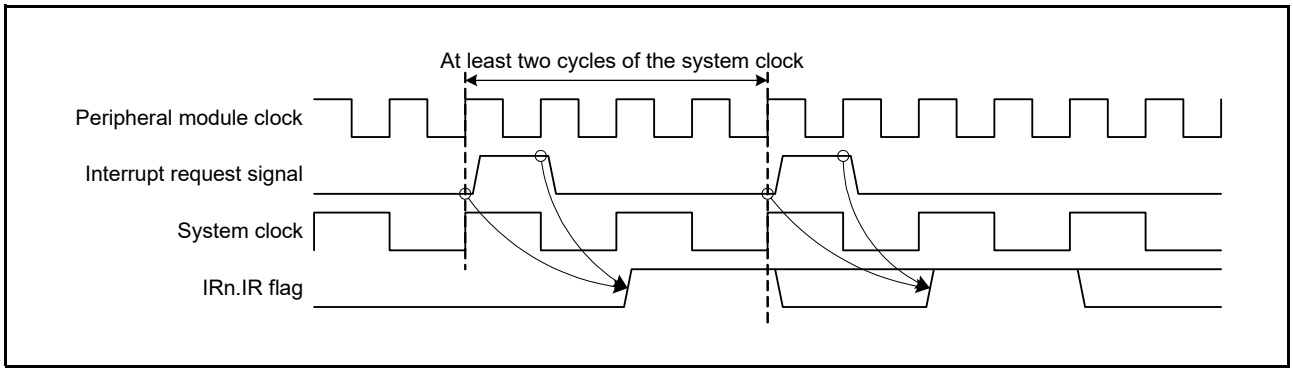


Figure 15.6 Accepting Consecutive Interrupt Signals (when the system clock frequency is lower than the peripheral clock frequency)

When an interrupt request is generated again while the IRn.IR flag is 1, the interrupt request is ignored (n = 26 to 255). However, for transmit interrupt requests, receive interrupt requests, and buffer access interrupt requests of the RSCI, RI3C, RIICHS, RSPIA, SSIE, SDHI, and PDC, when an interrupt request occurs while the IRn.IR flag is 1, the interrupt request is retained in the module. After the IRn.IR flag becomes 0, the IRn.IR flag is set to 1 again by the retained request. Refer to the descriptions for interrupts in each chapter of peripheral modules for details.
Figure 15.7 shows the timing when the IRn.IR flag is set again.

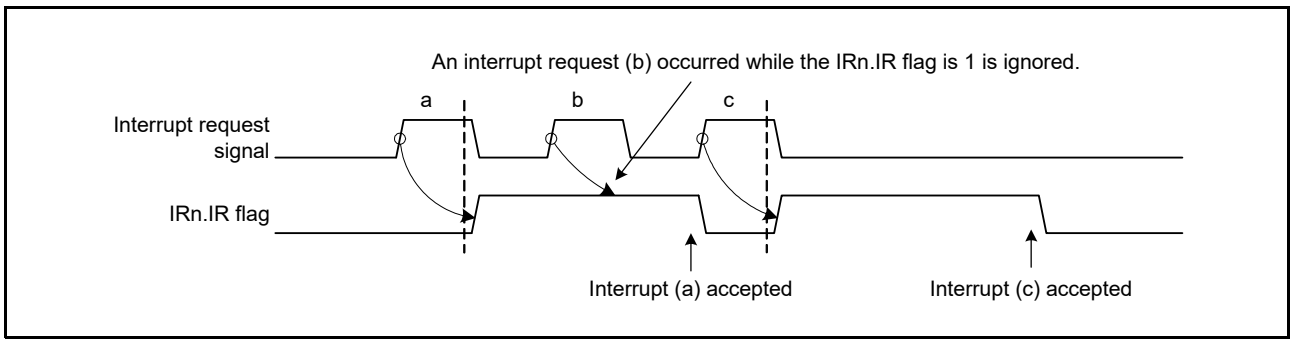


Figure 15.7 Timing to Set the IRn.IR Flag Again

(2) Relation between the IRn.IR flag and Interrupt Request Enable Bits

After the IRn.IR flag becomes 1, the IRn.IR flag does not become 0 even when an interrupt request enable bit in the corresponding peripheral module is set to 0.

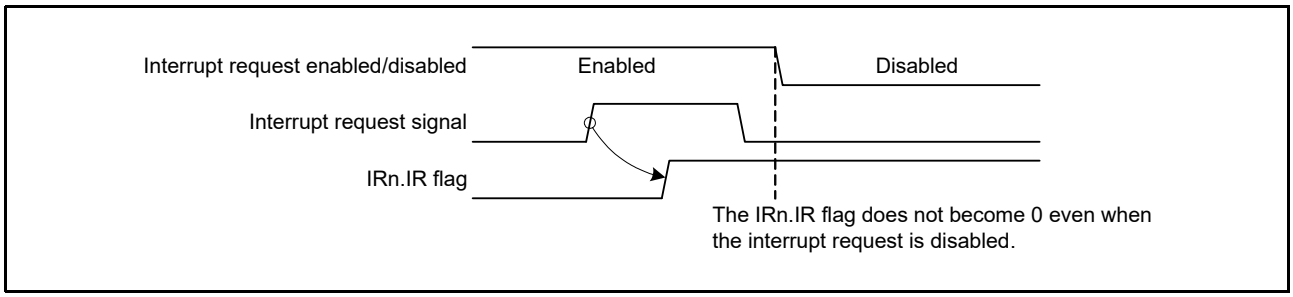


Figure 15.8 Relation Between Disabling the Interrupt Request and the IRn.IR Flag

15.5.2 Level Detection

Figure 15.9 shows operation of the interrupt request signal and the IRn.IR flag for level detection (n = 16 to 124). The IRn.IR flag is 1 while the interrupt request signal is 1. To set the IRn.IR flag to 0, set the corresponding interrupt request signal of the peripheral module to 0. Set the corresponding interrupt status flag of the peripheral module to 0 and wait for the time until the value is reflected in the IRn.IR flag before exiting the interrupt handler. Refer to (2) Notes on Writing to I/O Registers in section 5, I/O Registers for details on waiting for the reflection.

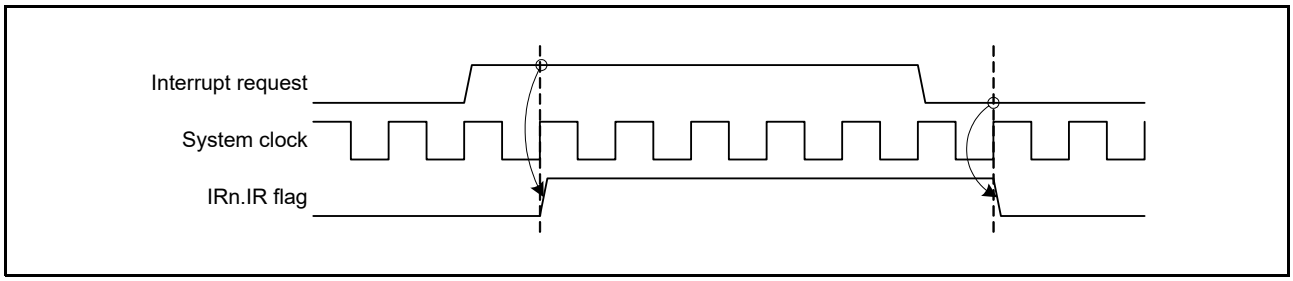


Figure 15.9 IRn.IR Flag Operation for Level Detection

Figure 15.10 shows an example of the procedure to handle interrupts for level detection.

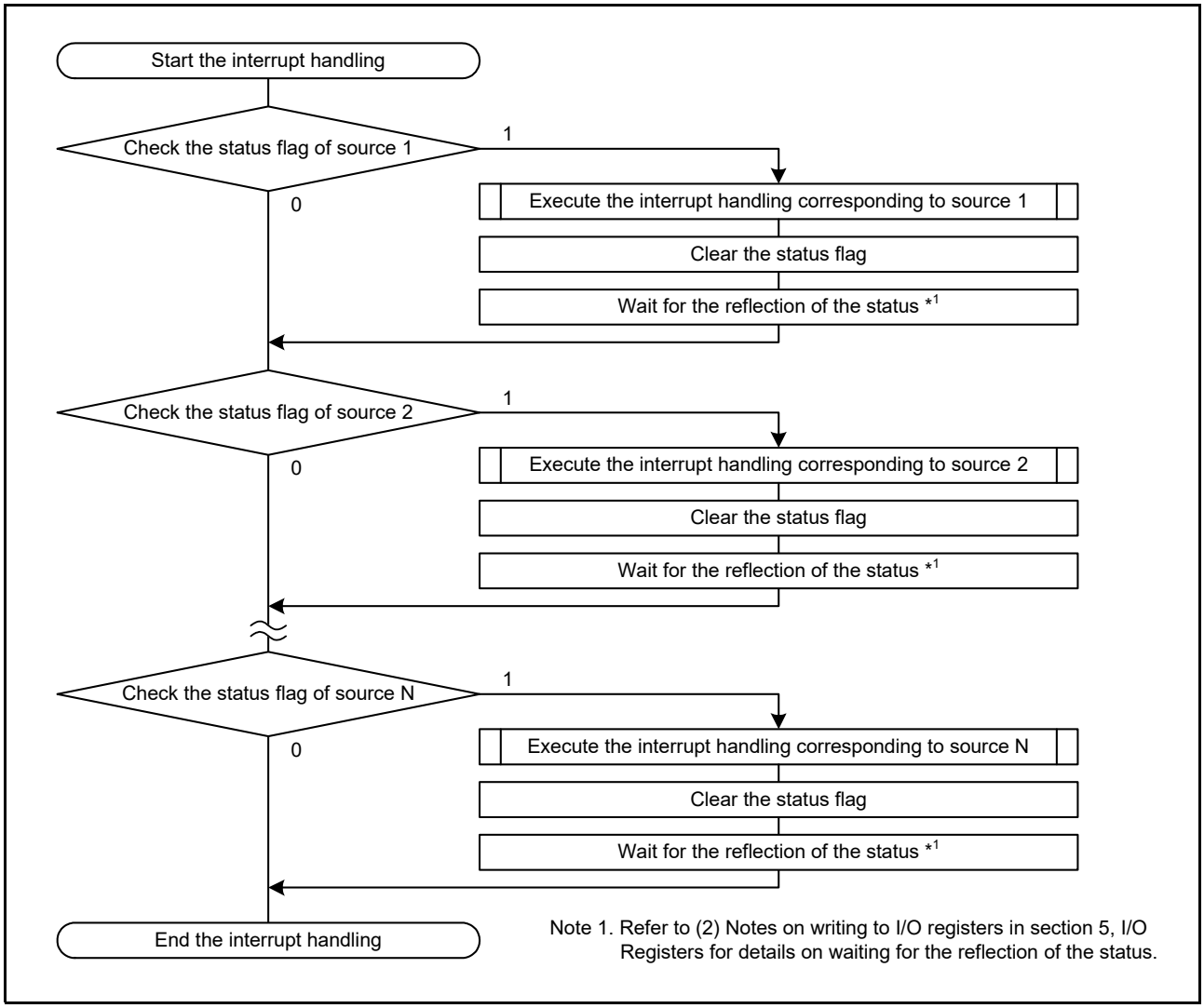


Figure 15.10 Example of Level Detection Interrupt Handling Procedure (N indicates the number of status flags)

15.5.3 Interrupt Detection for Group Interrupts

Groups IL0, IL1, BL0, BL1, and AL0 to AL6 includes interrupt sources that are detected by level detection.
The IR17.IR to IR20.IR, and IR102.IR to IR119.IR flags corresponding to the group interrupts change under the same conditions as level detection.
When an interrupt signal becomes 1 while the corresponding ENj bit in the GENIL0, GENIL1, GENBL0, GENBL1, or GENAL0 to GENAL6 register is 1, the ISj flag in the GRPIL0, GRPIL1, GRPBL0, GRPBL1, or GRPAL0 to GRPAL6 register and the IRn.IR flag become 1 (j = 0 to 31). Then, the ISj flag in the GRPIL0, GRPIL1, GRPBL0, GRPBL1, or GRPAL0 to GRPAL6 register and IRn.IR flag becomes 0 when the corresponding interrupt request signal becomes 0.
Also, when the ENj bit in the GENIL0, GENIL1, GENBL0, GENBL1, or GENAL0 to GENAL6 register is set to 0, the corresponding ISj flag in the GRPIL0, GRPIL1, GRPBL0, GRPBL1, or GRPAL0 to GRPAL6 register and IRn.IR flag become 0.
Figure 15.11 shows an operation example of group interrupts using level detection. Figure 15.12 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

Note: • There is no register in the group to which no interrupt source is assigned.

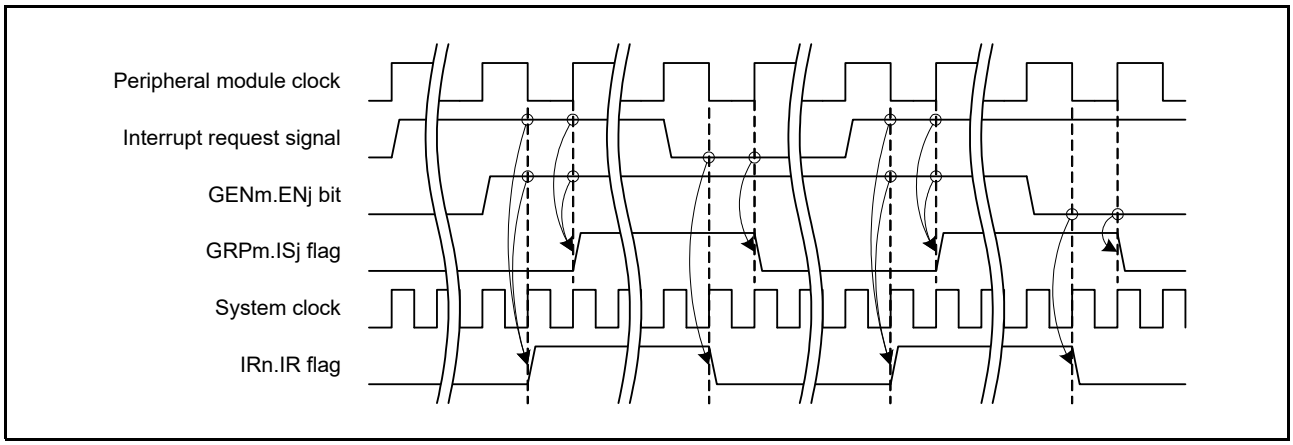


Figure 15.11 Operation Example of Group Interrupt Using Level Detection (m = IL0, IL1, BL0, BL1, and AL0 to AL6)

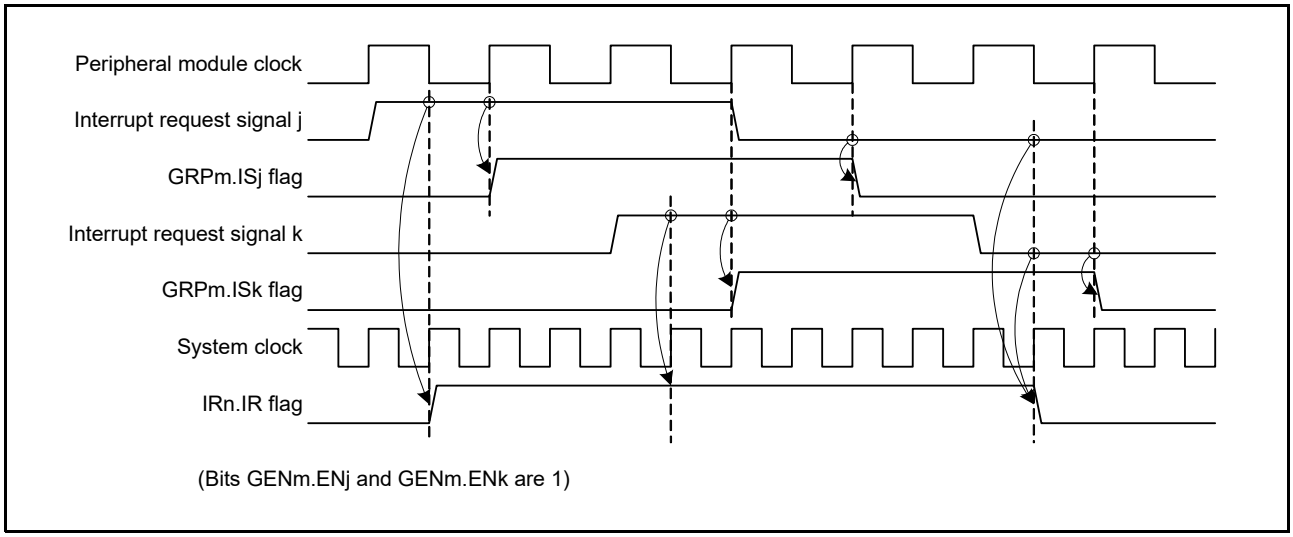


Figure 15.12 Operation Example When Multiple Interrupt Requests are Generated in the Same Group (m = IL0, IL1, BL0, BL1, and AL0 to AL6)

Figure 15.13 shows the procedure to handle group interrupts for level detection.

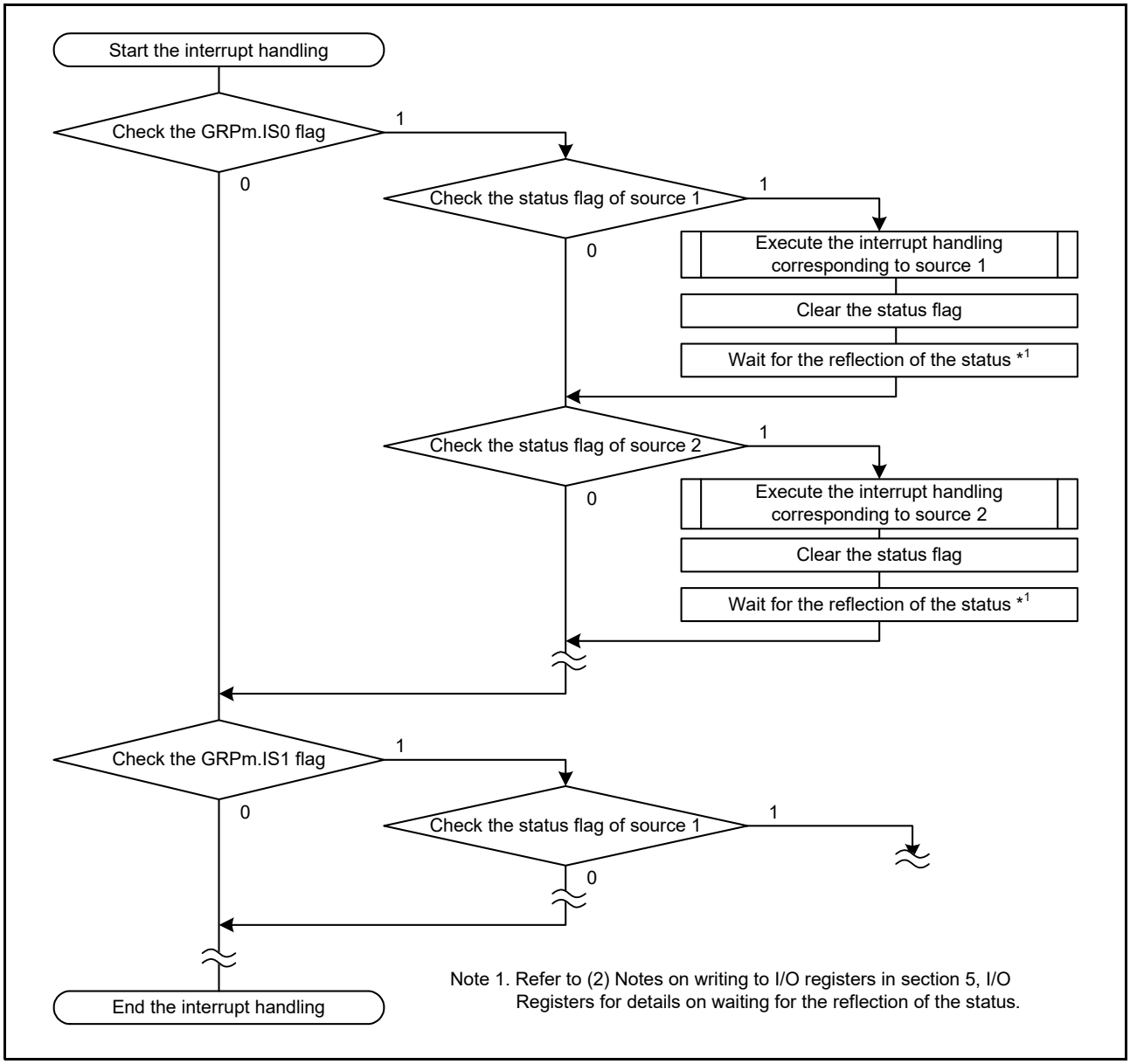


Figure 15.13 Example of Group Interrupt Handling Procedure for Level Detection (m = IL0, IL1, BL0, BL1, and AL0 to AL6)

15.5.4 Interrupt Detection for Software Configurable Interrupts

Interrupt sources and interrupt requests for software configurable interrupts are detected by edge detection.
Figure 15.14 shows an operation example of the interrupt request and interrupt status flag for software configurable interrupts.

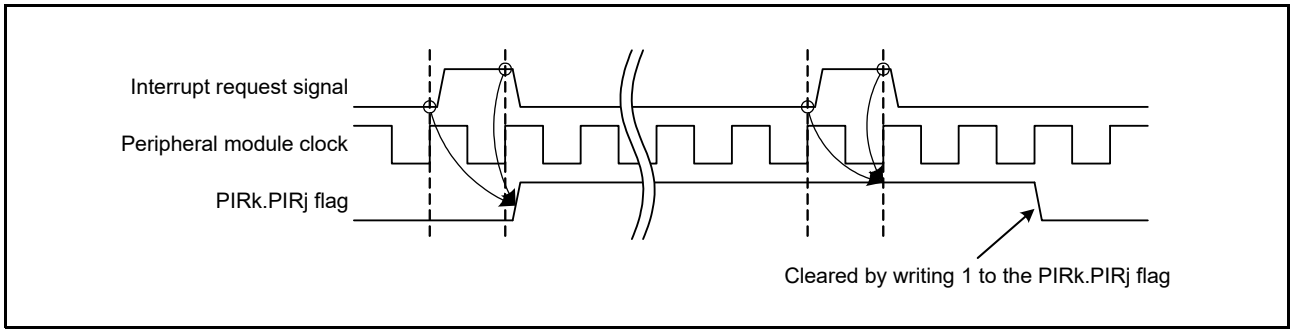


Figure 15.14 Operation Example of the Interrupt Request and Interrupt Status Flag for Software Configurable Interrupts

15.6 Determining Priority of Interrupt Requests

The ICU determines the priority for each interrupt request destination. The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Interrupt Request Destination

A source selected for the fast interrupt has the highest priority. Next to the fast interrupt, the priority is determined by the IPRr.IPR[4:0] bit value, and an interrupt source with a larger value has priority (r = 000 to 255). If multiple sources has the same IPRr.IPR[4:0] bit value, the priority is determined by the interrupt vector number, and the source with the smaller number has priority.

(2) Determining Priority when the DTC is the Interrupt Request Destination

The IPRr.IPR[4:0] bits have no effect (r = 000 to 255). The priority is determined by only the interrupt vector number, and an interrupt source with a smaller number has priority.

(3) Determining Priority when the DMAC is the Interrupt Request Destination

The IPRr.IPR[4:0] bits have no effect. The priority is determined by the DMAC channel number. Refer to section 18, DMA Controller (DMACAa) for details on the DMAC channel priority.

15.7 Interrupt Setting Procedure

15.7.1 Enabling Interrupt Requests

The following describes the procedure to enable interrupt requests.

(1) Set the interrupt request enable bit of the peripheral modules to enable output of the interrupt request.
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 1 to enable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the corresponding IERm.IENj bit to 1 to enable output of the interrupt request to the interrupt request destination (m = 2 to 31; j = 0 to 7).

After the above procedure is completed, when a peripheral interrupt occurs, the IRn.IR flag corresponding to the interrupt source becomes 1 (n = 16 to 255).
For group interrupts, the ISj flag in the group interrupt request register becomes 1, the IRn.IR flag corresponding to the group becomes 1, and an interrupt request is output to the interrupt request destination.
When the IERm.IENj bit is 0, the interrupt request corresponding to the interrupt source is not output to the interrupt request destination.

15.7.2 Disabling Interrupt Requests

The following describes the procedure to disable interrupt requests.

(1) Set the corresponding IERm.IENj bit to 0 (m = 2 to 31; j = 0 to 7).
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 0 to disable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the interrupt request enable bit of the peripheral modules to disable output of the interrupt request. Read the register that has been set to confirm that the value is reflected.
(4) As needed, read the IRn.IR flag or set the IR flag to 0.*1
For group interrupts, confirm that the ISj flag in the group interrupt request register is 0 or set the ISj flag to 0.

Note 1. When disabling the transmit interrupt request, receive interrupt request, or buffer access interrupt requests of the RSCI, RI3C, RIICHS, RSPIA, SSIE, SDHI, or PDC, set the IRn.IR flag to 0 according to the above procedure. Refer to the description of interrupts in the corresponding section of peripheral modules for details.

15.7.3 Selecting Interrupt Request Destination

15.7.3.1 Interrupt Request Destination Setting Procedure

The destination of an interrupt request can be selected from the CPU, DTC, or DMAC for each interrupt source. Destinations that can be selected differ depending on the interrupt source. Refer to Table 15.6, Interrupt Vector Table for details on the destinations. Do not select a destination that is not indicated as "✓" in Table 15.6.
When set the external pin interrupt as the DTC or DMAC trigger, set the IRQCRi.IRQMD[1:0] bits to select edge detection (i = 0 to 31).
The following describes the procedure to select a destination of an interrupt request.

(1) Setting Interrupt Sources as the DMAC trigger

Perform the following settings while the IERm.IENj bit is 0 of the interrupt source that is selected as the DMAC trigger (m = 2 to 31; j = 0 to 7).

(1) Set the interrupt vector number of the interrupt source used as the DMAC trigger in the DMRSRm register corresponding to the DMAC channel (m = DMAC channel number).*1
(2) Set the DMTMD.DCTG[1:0] bits corresponding to the DMAC channel to 01b in order to select the peripheral interrupt or external pin interrupt as the DMAC trigger.
(3) Set the DMCNT.DTE bit corresponding to the DMAC channel to 1.

After the above settings are completed, set the corresponding IERm.IENj bit to 1.
Also, set the DMAST.DMST bit to 1 before or after the above settings.
Refer to section 18.3.9, Activating the DMAC in section 18, DMA Controller (DMACAa) for the procedure to set the DMAC.

(2) Setting Interrupt Sources as the DTC trigger

Perform the following setting while the IERm.IENj bit of the interrupt source that is selected as the DTC trigger is 0.

(1) Set the DTCERn.DTCE bit corresponding to the interrupt vector number n used for the DTC trigger to 1 (n = 26 to 255).*1

After the above setting is completed, set the IERm.IENj bit to 1.
Also set the DTCST.DTCST bit to 1 before or after the above settings.
Refer to section 20.5, DTC Setting Procedure in section 20, Data Transfer Controller (DTCA) for the procedure to set the DTC.

Note 1. Do not set the same interrupt source as DTC and DMAC triggers. Also, do not set the same interrupt source as triggers of multiple DMAC channels.

(3) Setting Interrupt Sources for the CPU

When an interrupt source is not selected as the DTC or DMAC trigger, the interrupt request is output to the CPU.
Set the IERm.IENj bit to 1 while the interrupt source is not selected as a DTC or DMAC trigger.

15.7.3.2 Operations When the DTC/DMAC Selected

Table 15.10 lists operations when the DTC or DMAC is set as an interrupt request destination.

Table 15.10 Operations When Starting the DTC/DMAC

Interrupt Request Destination	DISEL *1	Number of Remaining Transfers	Operation per Request	IR Flag Clear Timing *2	Interrupt Request Destination after Transfer
DTC *3	1	≠ 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DTC
		= 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DTCERn.DTCE bit becomes 0)
	0	≠ 0	DTC transfer	Cleared when the DTC starts data transfer.	DTC
		= 0	DTC transfer → CPU interrupt *4	Cleared when the CPU accepts an interrupt request. *4	CPU (DTCERn.DTCE bit becomes 0)
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DMACm.DMCNT.DTE bit becomes 0)
	0	≠ 0	DMA transfer	Cleared when the DMAC starts data transfer.	DMAC
		= 0	DMA transfer *4	Cleared when the DMAC starts data transfer. *4	CPU (DMACm.DMCNT.DTE bit becomes 0)

Note 1. For the DTC, set the DTC.MRB.DISEL bit; For the DMAC, set the DMACm.DMCSL.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again is ignored.
Note 3. For chain transfer, the DTC transfer continues until the chain transfer ends. After chain transfer ends, whether a CPU interrupt occurs, the IRn.IR flag clear timing, and the interrupt request destination differ depending on the DISEL bit value the number of remaining transfers. For the chain transfer, refer to Table 20.4, Chain Transfer Conditions in section 20, Data Transfer Controller (DTCA).
Note 4. When the DISEL bit is 0 and the number of remaining transfers is 0, operations in the DTC and DMAC are different.

15.7.3.3 Changing the Interrupt Request Destination

Set the IERm.IENj bit to 0 before changing the interrupt request destination (m = 2 to 31; j = 0 to 7).

(1) When the current interrupt request destination is the DMAC

To change interrupt request destinations or change the DMAC trigger to another interrupt source while the DMA transfer is not completed (DMCNT.DTE bit is not cleared) after the procedure described in (1) Setting Interrupt Sources as the DMAC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below.

(1) Set the IERm.IENj bit of both the current and new triggers to 0.
(2) Check the DMAC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

(2) When the current interrupt request destination is the DTC

To change interrupt request destinations or change the DTC transfer information while the DTC transfer is not completed (the DTCERn.DTCE bit is not cleared) after the procedure described in (2) Setting Interrupt Sources as the DTC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below (n = 26 to 255).

(1) Set the IERm.IENj bit of both the current and new triggers to 0 (m = 2 to 31; j = 0 to 7).
(2) Check the DTC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

15.7.4 Setting the External Pin Interrupt

The following describes the procedure to use the external pin interrupt.

(1) Set the IERm.IENj bit corresponding to the IRQi pin to 0 (interrupt request is disabled) (m = 2 to 31; j = 0 to 7; i = 0 to 31).
(2) Set the IRQCRi.FLTEN bit to 0 (digital noise filter is disabled).
(3) Set the IRQCRi.FCLKSEL[2:0] bits to select the sampling clock and FLTMD bit to select the operating mode of the digital noise filter.
(4) Set the I/O port and confirm the setting.
(5) Set the IRQCRi.IRQMD[1:0] bits to select the detection method.
(6) When edge detection is selected, set the corresponding IRn.IR flag to 0 (n = 16 to 255).
(7) Set the IRQCRi.FLTEN bit to 1 (digital noise filter is enabled).
(8) To select the DTC as the interrupt request destination, set the DTCERn.DTCE bit. To select the DMAC as the interrupt request destination, set the DMRSRm register. When neither the DTCERn.DTCE bit nor the DMRSRm register is set, the interrupt request is sent to the CPU (m = DMAC channel number; n = 26 to 255).
(9) Set the corresponding IERm.IENj bit to 1 (interrupt request is enabled).

15.7.5 Setting Non-Maskable Interrupts

After reset, non-maskable interrupts are disabled. To use non-maskable interrupts, follow the procedure below.

(1) Set the stack pointer (SP).
(2) When using the NMI pin, set the NMICR.NFLTEN bit to 0 (digital noise filter is disabled).
(3) When using the NMI pin, set the NMICR.NFCLKSEL[2:0] bits to select the sampling clock and NFLTMD bit to select the operating mode of the digital noise filter.
(4) When using the NMI pin, set the NMICR.NMIMD bit to select the edge for detection.
(5) When using the NMI pin, write 1 to the NMICLR.NMICLR bit to set the NMISR.NMIST flag to 0.
(6) When using the NMI pin, set the NMICR.NFLTEN bit to 1 (digital noise filter is enabled).
(7) To enable generation of the non-maskable interrupt, set the bit in the NMIER register corresponding to the used interrupt source to 1.

Once a bit in the NMIER register is set to 1 (enabled), the bit cannot be rewritten so it cannot be set to 0 (disabled). To disable a non-maskable interrupt that has been enabled, reset the MCU.
Refer to section 14, Exception Handling for details on the flow of non-maskable interrupt handling.
The NMIST, OSTST, SOSTST, PVD2ST, PVD1ST, WDTST, and IWDTST flags in the NMISR register become 0 by writing 1 to the corresponding bit in the NMICLR register.
Confirm that all flags in the NMISR register are 0 before exiting the interrupt handler of non-maskable interrupts.
Non-maskable interrupts, excluding the NMI pin interrupt, can be used as a maskable interrupt. When using as a maskable interrupt, do not change the NMIER register value from the value after reset. In addition, set the PVD1CR1.IRQSEL bit and PVD2CR1.IRQSEL bit to 1 when using voltage monitoring 1 interrupt and voltage monitoring 2 interrupt as a maskable interrupt.

15.7.6 Digital Noise Filter

Noise included in signals input to pins IRQi and NMI can be reduced by enabling the digital noise filter (i = 0 to 31). The digital noise filter samples signals input to pins using the sampling clock (PCLKB, PCLKB/4, PCLKB/8, PCLKB/16, PCLKB/32, PCLKB/64, or PCLKB/128) for the digital noise filter, and passes the input signal only when three or four consecutive sampled signals are the same level.

When using the digital noise filter for the IRQi pin, refer to section 15.7.4, Setting the External Pin Interrupt to set the associated registers. When using the digital noise filter for the NMI pin, refer to section 15.7.5, Setting Non-Maskable Interrupts to set the associated registers.

When the external pin interrupt or the NMI pin interrupt is used as a source to exit software standby mode, the digital noise filter cannot be used. Set the IRQCRi.FLTEN bit or NMICR.NFLTEN bit to 0 before entering software standby mode. To enable the digital noise filter again, set the IRQCRi.FLTEN bit or NMICR.NFLTEN bit to 1.

When changing the digital noise filter settings, also set the IRQCRi.FLTEN bit or NMICR.NFLTEN bit to 0.

Figure 15.15 shows an example of digital noise filter operation.

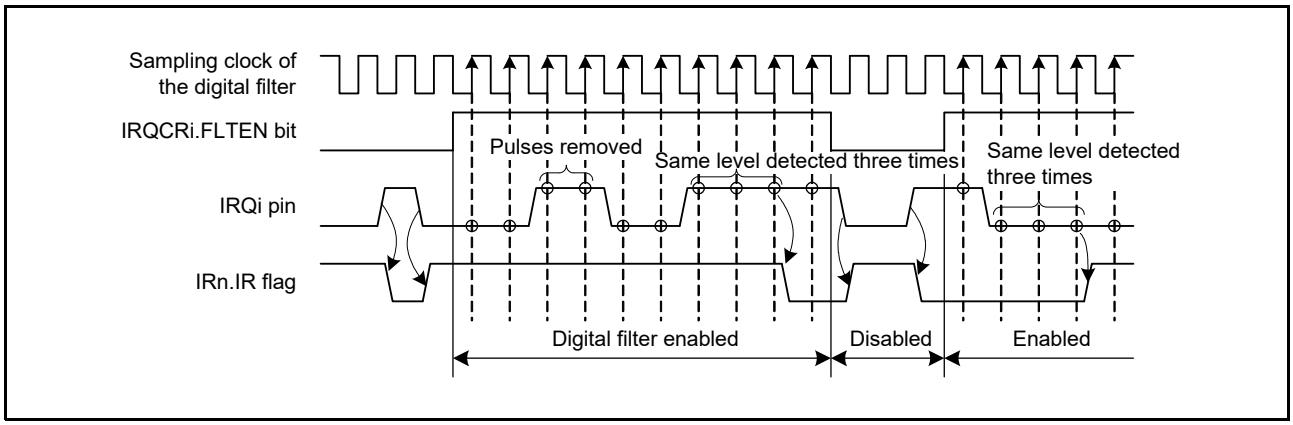


Figure 15.15 Digital Noise Filter Operation Example (when the IRQCRi.IRQMD[1:0] bits are 11b (low level))

15.7.7 Setting Software Configurable Interrupts

The following describes the procedure to assign interrupt sources to software configurable interrupts.

(1) Set the corresponding IERm.IENj bit to 0 (m = 2 to 31; j = 0 to 7). This setting is not required when the value does not change from the value after reset.
(2) Set the interrupt source number in the SLIXRn (n = 128 to 143) or SLIRn (n = 144 to 255) register. The source number for the external pin interrupt is 255. For the source numbers for other interrupts, refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt.
(3) Set the SLIPRCR.WPRC or SLIPRCR_NS.WPRC bit to 1.*1
(4) Confirm that the WPRC bit set in (3) is 1.*1
(5) Select the interrupt request destination from the CPU, DTC, or DMAC. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for details on the setting procedure.
(6) Write 0 to the IRn.IR flag (n = 128 to 255).
(7) Set the IERm.IENj bit to 1.

Note 1. When not using write protection, this setting is not required.

15.7.7.1 Polling for Software Configurable Interrupts

When polling an interrupt request by reading the PIXRk.PIRj flag (k = 0, 1; j = 0 to 7) or the PIRk.PIRj flag (k = 0 to 127; j = 0 to 7), follow the procedure below.

(1) Set the external pin or peripheral interrupt used.
(2) Clear the PIXRk.PIRj*1 or PIRk.PIRj flag for polling by writing 1 to the flag.*2
(3) Enable output of the peripheral interrupt request.
(4) As needed, read the PIXRk.PIRj or PIRk.PIRj flag to check the value.
(5) When clearing the PIXRk.PIRj*1 or PIRk.PIRj flag, write 1 to the targeted flag.*2
(6) As needed, repeat step (4) and (5).

Note 1. Only when any of the edge detection is selected by IRQCRi.IRQMD[1:0] bits
Note 2. Do not use bit manipulation instructions. Multiple status flags may be cleared if a bit manipulation instruction is used. To clear a flag, write the PIRk register in 8-bit units as follows: set the flag that is to be cleared to 1 and set the other flags to 0.

15.8 Multiple Interrupt

To enable another interrupt while processing an interrupt (multiple interrupt), set the PSW.I bit to 1 (interrupt enabled) in the interrupt handler of an accepted interrupt.
The PSW.IPL[4:0] bits in the interrupt handler are the same value as the priority level of the accepted interrupt request. In this case, when an interrupt request with the higher priority level than the PSW.IPL[4:0] bit value is generated, the interrupt request is accepted.
The PSW.I bit can be rewritten only in supervisor mode. Since the PSW.PM bit becomes 0 (supervisor mode is selected) when an interrupt is accepted, the PSW.I bit can be rewritten in the interrupt handler.

15.9 Fast interrupt

The fast interrupt is an interrupt that the CPU can respond to fast. Only one interrupt source can be assigned to the fast interrupt. The fast interrupt can be used when the PRTZCR.PRTZC bit is 0.
The priority level of the fast interrupt is 31 (highest) regardless of the IPRr.IPR[4:0] bit setting (r = 000 to 255). Also, the fast interrupt has higher priority than the other interrupt sources of which the priority level is 31. Note that the fast interrupt cannot be accepted when the PSW.IPL[4:0] bits are 11111b (priority level 31).
To assign an interrupt source to the fast interrupt, set the FIR.FVCT[7:0] bits to select the vector number of the interrupt source, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
The fast interrupt is enabled only when the CPU is selected as the interrupt request destination. When the DTC or DMAC is selected as the destination, the fast interrupt is disabled.
Refer to section 2, CPU and section 14, Exception Handling for details on the fast interrupt.

15.10 Exiting Low Power Consumption State

Interrupts can be used for exiting sleep mode, deep sleep mode, and software standby mode.
Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for exiting each low power consumption mode.
Refer to section 11.6.3, Deep Software Standby Mode for details on exiting deep software standby mode.

15.10.1 Exiting Sleep Mode or Deep Sleep Mode

Non-maskable interrupts and all interrupt sources can be used for exiting sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 2 to 31; j = 0 to 7).
- The priority level is higher than the PSW.IPL[4:0] bit value in the CPU.
- For group interrupts, the interrupt request is enabled by the corresponding ENj bit in the group interrupt request enable register (GENIL0, GENIL1, GENBL0, GENBL1, and GENAL0 to GENAL6) (j = 0 to 31).

15.10.2 Exiting Software Standby Mode

Non-maskable interrupts and interrupt sources that have a "✓" in the Exit from SSBY column in Table 15.6, Interrupt Vector Table can be used for exiting software standby mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.
- When using the NMI pin interrupt, the digital noise filter is disabled.

(2) Interrupts

- The interrupt source can be used for exiting software standby mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 2 to 31; j = 0 to 7).
- The priority level is higher than the PSW.IPL[4:0] bit value in the CPU.
  When using the fast interrupt, set not only the FIR register but also the corresponding IPRr.IPR[4:0] bits (r = 000 to 255). Set the IPRr.IPR[4:0] bits to a higher level than the PSW.IPL[4:0] bit value in the CPU.
- When using the external pin interrupt, the digital noise filter of the IRQi pin used is disabled.

Refer to section 15.7.6, Digital Noise Filter for details on the procedure to set the digital noise filter.

15.11 Usage Notes

15.11.1 Notes on the WAIT instruction When Using the Non-Maskable Interrupt

Confirm that all status flags in the NMISR register are 0 before executing the WAIT instruction.

15.11.2 Interrupt Requests in Software Standby Mode

When an interrupt request occurs in software standby mode but the interrupt source is not set as a source for exiting software standby mode, the request is held in the ICU. The request is handled after exiting by another interrupt source. Note that the interrupt request for the external pin interrupt is not held.

15.11.3 Notes on Bus Error during DTC/DMA Transfer

If a bus error occurs during DTC/DMA transfer, clear the corresponding IRn.IR flag by software. Also, set the corresponding DTCERn or DMRSRm register to 00h to prevent the same DTC/DMA transfer from being initiated again.

16. Buses

16.1 Overview

In this section, the EDMAC indicates the Ethernet CPU agent (GWCA) of the Ethernet module (ESWM), and SPI0M and SPI1M indicate XSPI0 and XSPI1 serving as target modules, respectively. Table 16.1 lists the bus specifications, Figure 16.1 shows the system bus configuration, and Table 16.3 lists the addresses assigned for each bus.

Table 16.1 Bus Specifications

Bus Master/Bus Slave	Sync clock	Description
Classification: Bus masters		
CPU0IFBI	CPUCLK0	Connected to the CPU0IF AHB Interface
CPU0OABI	CPUCLK0	Connected to the CPU0OA AHB Interface
DMAC/DTC0BI	ICLK	Connected to the DMAC/DTC0 Interface
EXDMACBI	ICLK	Connected to the EXDMAC Interface
EDMACBI	ICLK	Connected to EDMAC Interface
SPI0MBI	ICLK	Connected to the SPI0 Master (SPI0M) AXI Interface
SPI1MBI	ICLK	Connected to the SPI1 Master (SPI1M) AXI Interface
Classification: Bus slaves		
MRC0BI	MRICLK	Connected to the code MRAM bank 0 (MRC0)
MRC1BI	MRICLK	Connected to the code MRAM bank 1 (MRC1)
MRE0BI	MRPCLK	Connected to the extra MRAM
LRAM0BI	CPUCLK0	Connected to the LRAM0 AHB Interface
LRAM1BI	CPUCLK0	Connected to the LRAM1 AHB Interface
S0BI	ICLK	Connected to the SRAM0
S1BI	ICLK	Connected to the SRAM1
ECBI	BCLK	Connected to the external devices (External Memory Interface)
XSPI0BI	ICLK	Connected to the XSPI0 controller (External Memory Interface)
XSPI1BI	ICLK	Connected to the XSPI1 controller (External Memory Interface)
PBBI	PCLKB	Connected to peripheral modules synchronizes with PCLKB
PABI	PCLKA	Connected to peripheral modules synchronizes with PCLKA
PIBI	ICLK	Connected to peripheral modules synchronizes with ICLK
PS0BI	ICLK	Connected to ICU0 controller Connected to DMAC/DTC0 controller
PC0BI	CPUCLK0	Connected to peripheral system modules. (TFU, MPU, CPU controller, and CPU0-cache controller)
PSBI	ICLK	Connected to peripheral system modules. (CSC/SDRAM, SRAM, Bus controller, EXDMAC, and Security attribution controller)
	MRPCLK	Connected to peripheral system module. (MRAM controller)
	PCLKB	Connected to peripheral system module. (System controller)

Note: CPU0IFBI: CPU0 IF AHB interface
CPU0OABI: CPU0 OA AHB interface
DMAC/DTC0BI: DMAC/DTC0 bus interface
EXDMACBI: EXDMAC bus interface
EDMACBI: EDMAC bus interface
SPI0MBI: SPI0 master bus interface
SPI1MBI: SPI1 master bus interface
MRC0BI: MRC0 bus interface
MRC1BI: MRC1 bus interface
MRE0BI: MRE0 bus interface
LRAM0BI: LRAM0 AHB interface
LRAM1BI: LRAM1 AHB interface

S0BI: SRAM0 bus interface
S1BI: SRAM1 bus interface
ECBI: External memory CSC and SDRAM bus interface
XSPI0BI: External memory XSPI0 controller bus interface
XSPI1BI: External memory XSPI1 controller bus interface
PBBI: Peripheral bus interface synchronized with PCLKB
PABI: Peripheral bus interface synchronized with PCLKA
PIBI: Peripheral bus interface synchronized with ICLK
PS0BI: ICU0, DMAC/DTC0 controller bus interface
PC0BI: CPU peripheral system bus interface
PSBI: Peripheral system bus interface

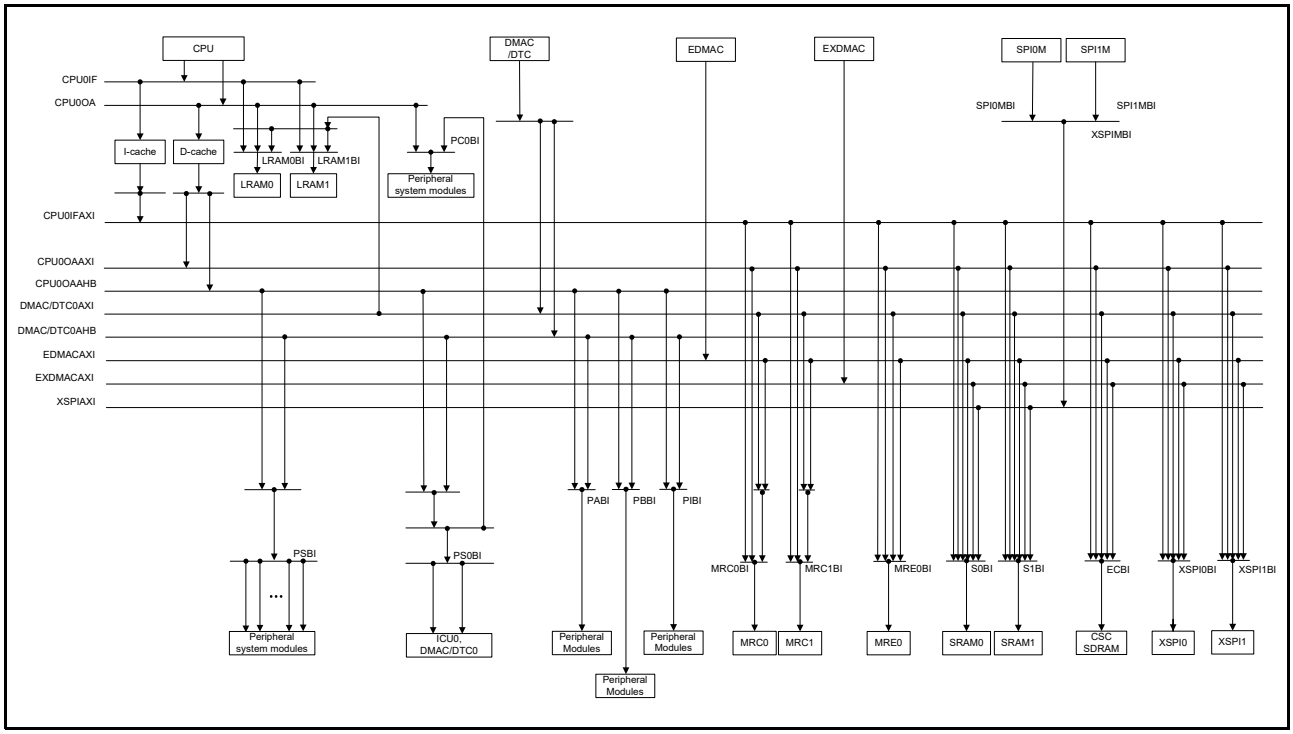


Figure 16.1 System Bus Connection

Table 16.2 System Bus Access Path

		Master					
	Name	**CPU0IFBI*1**	**CPU0OABI**	**DMAC/DTC0BI**	**EXDMACBI**	**EDMACBI**	**SPI0MBI/SPI1MBI**
Slave	LRAM0BI/LRAM1BI	T	T	T	F	F	F
	PC0BI	F	T	T	F	F	F
	PS0BI	F	T	T	F	F	F
	MRC0BI/MRC1BI	T	T	T	F	T*2	F
	MRE0BI	T	T	T	F	T*2	F
	PBBI	F	T	T	F	F	F
	PABI	F	T	T	F	F	F
	PSBI	F	T	T	F	F	F
	PIBI	F	T	T	F	F	F
	S0BI/S1BI	T	T	T	T	T	T
	ECBI	T	T	T	T	T	F
	XSPI0BI/XSPI1BI	T	T	T	T	T	F

T : The Master can access slaves
F : The master cannot access slaves
Note 1. CPU0IFBI is read-only master
Note 2. Read-only

Table 16.3 Address Assigned to Each Slave

Address	Bus Interface	Area
0000 0000h to 0001 FFFFh (Non-secure) 8000 0000h to 8001 FFFFh (Secure)	LRAM0BI	LRAM0 (Area when accessed by the CPU)
0002 0000h to 0003 FFFFh (Non-secure) 8002 0000h to 8003 FFFFh (Secure)	LRAM1BI	LRAM1 (Area when accessed by the CPU)
0100 0000h to 010F FFFFh (Non-secure) 8100 0000h to 810F FFFFh (Secure)	PC0BI	TFU, CPU0-cache, LRAM, MPU controller
0200 0000h to 020F FFFFh (Non-secure) 8200 0000h to 820F FFFFh (Secure)	SRAM0BI	SRAM0
0210 0000h to 0217 FFFFh (Non-secure) 8210 0000h to 8217 FFFFh (Secure)	SRAM1BI	SRAM1
0280 0000h to 0281 FFFFh (Non-secure) 8280 0000h to 8281 FFFFh (Secure)	LRAM0BI	LRAM0 (Area when accessed by a bus master other than the CPU)
0282 0000h to 0283 FFFFh (Non-secure) 8282 0000h to 8283 FFFFh (Secure)	LRAM1BI	LRAM1 (Area when accessed by a bus master other than the CPU)
0600 0000h to 06FF FFFFh (Non-secure) 8600 0000h to 86FF FFFFh (Secure)	MRE0BI	MRE0
0700 0000h to 0700 FFFFh (Non-secure) 8700 0000h to 8700 FFFFh (Secure)	PS0BI	ICU0, DMAC0, DTC0
0701 0000h to 070F FFFFh (Non-secure) 8701 0000h to 870F FFFFh (Secure)	PSBI	System controller, MRAM, CPSCU, BUS, EXBUS, EXDMAC, SRAM controller
0740 0000h to 077F FFFFh (Non-secure) 8740 0000h to 877F FFFFh (Secure)	PABI	Peripheral modules synchronized to PCLKA
0780 0000h to 07BF FFFFh (Non-secure) 8780 0000h to 87BF FFFFh (Secure)	PBBI	Peripheral modules synchronized to PCLKB
07C0 0000h to 07FF FFFFh (Non-secure) 87C0 0000h to 87FF FFFFh (Secure)	PIBI	Peripheral modules synchronized to ICLK
2000 0000h to 3FFF FFFFh (Non-secure)	XSPI0BI	XSPI0 controller
4000 0000h to 5FFF FFFFh (Non-secure)	XSPI1BI	XSPI1 controller
7000 0000h to 77FF FFFFh (Non-secure)	ECBI	SDRAM controller
7800 0000h to 7EFF FFFFh (Non-secure)	ECBI	CS1 to CS7 controller
7F00 0000h to 7FFF FFFFh (Non-secure)*1 FF00 0000h to FFFF FFFFh (Secure)	MRC0BI/MRC1BI	MRC0/MRC1
7F00 0000h to 7FFF FFFFh (Non-secure)*2	ECBI	CS0 controller

Note 1. MRAM (MRC0/MRC1) is not allocated when the on-chip code MRAM is disabled.
Note 2. CS0 is allocated when the on-chip code MRAM is disabled.

16.2 Description of Buses

16.2.1 Arbitration

Except for PC0BI, LRAM0BI, LRAM1BI, and XSPIMBI, for arbitration between masters, fixed-priority and round-robin methods can be selected. For details, refer to section 16.3.27, Bus Slave Arbitration Control Register 0 (BUSSABT0MRE0BI, BUSSABT0S0BI, BUSSABT0S1BI, BUSSABT0ECBI, BUSSABT0XSPI0BI, BUSSABT0XSPI1BI, BUSSABT0PBBI, BUSSABT0PABI, BUSSABT0PIBI, BUSSABT0PSBI, BUSSABT0PS0BI), and section 16.3.28, Bus Slave Arbitration Control Register 1 (BUSSABT1MRC0BI, BUSSABT1MRC1BI).
Table 16.4 lists the priority of PC0BI, LRAM0BI, LRAM1BI, and XSPIMBI.

Table 16.4 Priority of PC0BI, LRAM0BI, LRAM1BI, and XSPIMBI

Bus Interface	Priority*1
PC0BI	CPU0OA <-> (CPU0OAAHB or DMAC0)
LRAM0BI	CPU0OA > CPU0IF > DMAC0
LRAM1BI	CPU0OA > CPU0IF > DMAC0
XSPIMBI	SPI0M > SPI1M

Note 1. > : Indicates that the left side has higher priority.
<-> : Indicates round-robin.

16.2.2 External Bus

The external bus controller arbitrates bus access requests in the external address space from CPU0IFBI, CPU0OABI, DMAC/DTC0BI, EXDMACBI, SPI0MBI, SPI1MBI, and EDMACBI. For details about the priority and arbitration method for each master, refer to section 16.2.1, Arbitration.
The bus system provides an external space for XSPI (XSPI0BI/XSPI1BI). For details, refer to section 48, Expanded Serial Peripheral Interface (XSPI).
Table 16.5 lists the external bus specifications and Table 16.6 lists the input/output pins.
The external bus controller transfers data between the AXI master and the external bus via a FIFO.
For external address space accesses, the FIFO captures the bus access request and releases the internal bus before the external bus access is completed, allowing the internal bus to begin the next bus access. However, if the FIFO is full and the next access is an access to external address space, it will wait until the previous external bus operation is completed. Figure 16.2 shows an example of operation when the FIFO function is used. When this function is used, if the next operation after access to an external device is to access a peripheral module, the peripheral module access is executed in parallel without waiting for the completion of the external device access.

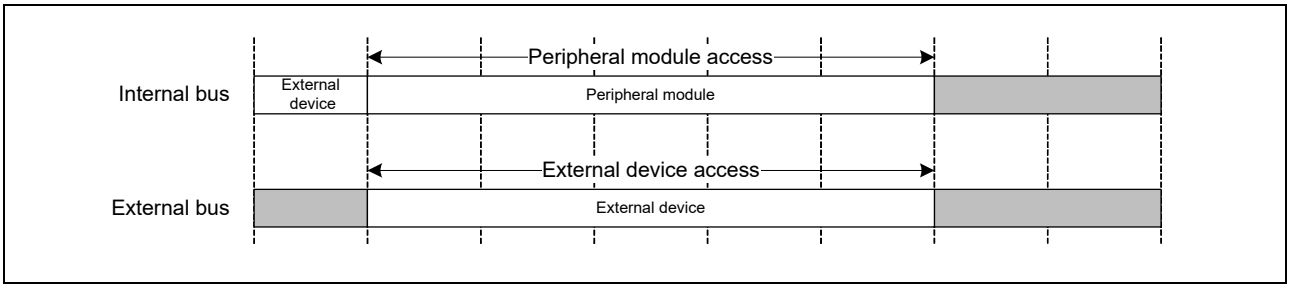


Figure 16.2 Example operation when using the FIFO function

Table 16.5 Specifications of the External Bus

Parameter	Description
External address space	• An external address space is divided into eight CS areas (CS0 to CS7) and the SDRAM area (SDCS) for management. • Chip select signals can be output for each area. • Bus width can be set for each area. Separate bus: An 8, 16, or 32-bit bus space is selectable. Address/data multiplexed bus: An 8 or 16-bit bus space is selectable. • Endian mode can be specified for each area. • Big endian mode has the following restrictions. See section 16.2.5.1, Restriction on Endian.
CS area controller	• Recovery cycles can be inserted. Read recovery: Up to 15 cycles Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles) • Wait control can be used to set up the following. Timing of assertion and negation for chip-select signals (CS0# to CS7#) The timing of assertion of the read signal (RD#) and write signals (WR0#/WR# to WR3#) The timing with which data output starts and ends • Write access mode: Single write strobe mode/byte strobe mode Separate bus or address/data multiplexed bus can be set for each area.
SDRAM area controller	• Multiplexing output of row address/column address (8, 9, 10, or 11 bits) • Self-refresh and auto-refresh selectable • CAS latency can be specified from one to three cycles
FIFO	FIFO stages : 16 for read data / write data
Frequency	• The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK)*1 • The frequency of the EBCLK pin output is the same as BCLK by default. Half of the BCLK cycles can be supplied by setting the EBCLK Pin Output Select bit, BCKCR.BCLKDIV, in the External Bus Clock Control Register. For more information, see section 9, Clock Generation Circuit. • The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).*1

Note 1. The EBCLK and the SDCLK should be operated with the same frequency when the SDRAM is in use.

Table 16.6 Pin Configuration of the External Bus (1/2)

Pin Name	I/O	Related Functions	Description
EBCLK, SDCLK*1	Output	CSC, SDRAMC	Clock output pins
A23 to A0*2	Output	CSC, SDRAMC	Address output pins
D31 to D0	I/O	CSC, SDRAMC	Data input/output pins • D31 to D0 pins are enabled when the 32-bit bus space is specified. • D15 to D0 pins are enabled when the 16-bit bus space is specified. • D7 to D0 pins are enabled when the 8-bit bus space is specified.
BC0#	Output	CSC	• A strobe signal; (the BC0# signal being at the low level) during access to an external address space in single write strobe mode indicates that D7 to D0 are valid, active-low. • When an 8-bit bus space is specified, this output pin is always held low regardless of write access mode.
BC1#	Output	CSC	• A strobe signal; (the BC1# signal being at the low level) during access to an external address space in single write strobe mode indicates that D15 to D8 are valid, active-low. • This pin is not used when the 8-bit bus space is specified.
BC2#	Output	CSC	• A strobe signal; (the BC2# signal being at the low level) during access to an external address space in single write strobe mode indicates that D23 to D16 are valid, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
BC3#	Output	CSC	• A strobe signal; (the BC3# signal being at the low level) during access to an external address space in single write strobe mode indicates that D31 to D24 are valid, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
CS0#*3	Output	CSC	A chip select signal for area 0 (CS0), active-low
CS1#*3	Output	CSC	A chip select signal for area 1 (CS1), active-low
CS2#*3	Output	CSC	A chip select signal for area 2 (CS2), active-low
CS3#*3	Output	CSC	A chip select signal for area 3 (CS3), active-low
CS4#	Output	CSC	A chip select signal for area 4 (CS4), active-low
CS5#	Output	CSC	A chip select signal for area 5 (CS5), active-low
CS6#	Output	CSC	A chip select signal for area 6 (CS6), active-low
CS7#	Output	CSC	A chip select signal for area 7 (CS7), active-low
RD#	Output	CSC	A strobe signal indicating that reading from an external address space (CS0 to CS7) is in progress, active-low.
WR0#/WR#*4	Output	CSC	• WR0# signal is a strobe signal indicates that (the WR0# signal being at the low level) writing to an external address space is in progress in byte strobe mode, and D7 to D0 are valid, active-low. • WR# signal is a strobe signal that indicates writing to an external address space is in progress in single write strobe mode, active-low. • When an 8-bit bus space is specified, this output pin is held low during a write access regardless of write access mode.
WR1#	Output	CSC	• A strobe signal; (the WR1# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D15 to D8 are valid, active-low. • This signal is invalid in single write strobe mode. • This pin is not used when the 8-bit bus space is specified.
WR2#	Output	CSC	• A strobe signal; (the WR2# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D23 to D16 are valid, active-low. • This signal is invalid in single write strobe mode. • This pin is not used when the 8- or 16-bit bus space is specified.

Table 16.6 Pin Configuration of the External Bus (2/2)

Pin Name	I/O	Related Functions	Description
WR3#	Output	CSC	• A strobe signal; (the WR3# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D31 to D24 are valid, active-low. • This signal is invalid in single write strobe mode. • This pin is not used when the 8- or 16-bit bus space is specified.
ALE	Output	CSC	Address latch signal when address/data multiplexed bus is selected.
WAIT#	Input	CSC	A wait request signal when accessing the external address space (CS0 to CS7), active-low.
BREQ#	Input	CSC/SDRAM	External bus hold request, active-low
BACK#	Output	CSC/SDRAM	External bus hold accepted, active-low
CKE	Output	SDRAMC	SDRAM clock enable signal
SDCS#	Output	SDRAMC	SDRAM chip select signal, active-low.
RAS#	Output	SDRAMC	SDRAM low address strobe signal, active-low.
CAS#	Output	SDRAMC	SDRAM column address strobe signal, active-low.
WE#	Output	SDRAMC	SDRAM write enable signal, active-low.
DQM0	Output	SDRAMC	SDRAM I/O data mask enable signal for D7 to D0
DQM1	Output	SDRAMC	SDRAM I/O data mask enable signal for D15 to D8
DQM2	Output	SDRAMC	SDRAM I/O data mask enable signal for D23 to D16
DQM3	Output	SDRAMC	SDRAM I/O data mask enable signal for D31 to D24

Note 1. The EBCLK and SDCLK pin functions are shared by the CS region controller (CSC) and the SDRAM region controller (SDRAMC).
When using the CSC and the SDRAMC simultaneously, the SDCLK pin function is valid.

Note 2. The A23 to A0 pin functions are shared by the CSC and the SDRAMC.
When using the CSC only:
The A0 and BC0# pin functions share the same pin, and either becomes valid according to the area, with the function being A0 in byte strobe mode and BC0# in single-write strobe mode.
Setting the 8-bit external bus width is prohibited in single-write strobe mode.
When using the SDRAMC only:
The A15 to A0 pin functions are valid.
The A0 and DQM1 pin functions share the same pin, and either pin function becomes valid according to the external bus width. When selecting 8-bit bus width, the pin function is A0. When selecting 16-bit bus width, the pin function is DQM1.
When using the CSC and the SDRAMC simultaneously:
The A23 to A16 pin functions are valid for CSC. The A15 to A0 pin functions are shared by the CSC and the SDRAMC. In the SDRAMC functions, the A0 and the DQM1 pin functions work as described.
In the CSC functions, the A0 and the BC0# pin function works as described.

Note 3. The CS0# to CS3# (CSC) and SDRAMC pin functions share the same pin. When using the CSC and the SDRAMC simultaneously, the CS0# to CS3# pin functions are invalid.

Note 4. The WR0# signal and WR# signal are identical. The WR0# signal is referred to as WR# in single-write strobe mode.

16.2.3 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from code MRAM, the DMAC is able to handle transfer between a peripheral module and the external memory interface at the same time.
Figure 16.3 shows an example of parallel operations. In this example, the CPU uses CPUMAXIBI to access MRC0BI. Furthermore, the DMAC/DTC simultaneously accesses the peripheral bus or the external bus during access to MRC0BI by the CPU.

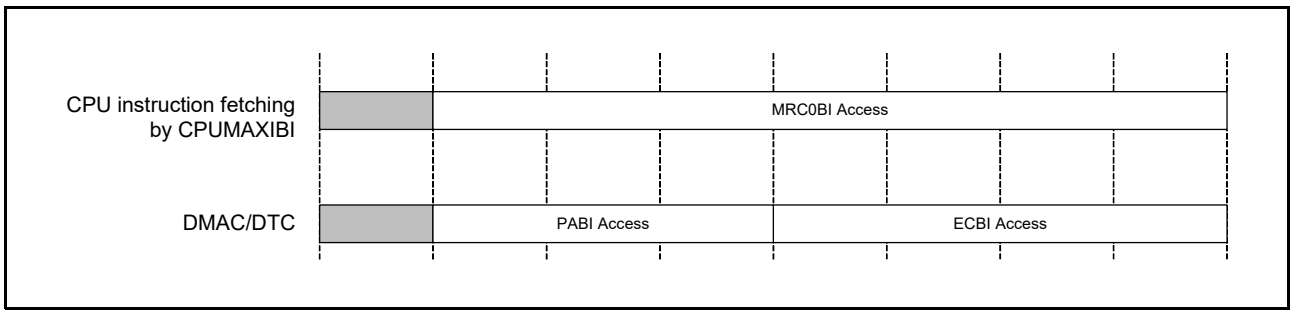


Figure 16.3 Example of Parallel Operations

16.2.4 Bus Settings

Set up the external bus with the following registers.

- Mode settings :
  CSn Mode Register (CSnMOD)
  CSn Wait Control Register 1 (CSnWCR1)
  CSn Wait Control Register 2 (CSnWCR2)
  CSn Control Register (CSnCR)
  CSn Recovery Cycle Setting Register (CSnREC)
  CS Recovery Cycle Insertion Enable Register (CSRECEN)
- I/O port assignments :
  PmnPFS.PMR = 1 and PmnPFS.PSEL[6:0] = 3Dh or 3Eh
- Frequency of the external bus clock (BCLK) and SDRAM clock (SDCLK) :
  SCKDIVCR register.

For details on PmnPFS, refer to section 22, I/O Ports. For details on SCKDIVCR, refer to section 9, Clock Generation Circuit.

16.2.5 Restrictions

16.2.5.1 Restriction on Endian

The following restrictions apply to areas where the endian setting for each area is different from that for the chip:

- No instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.
- When allocating data to be handled by the RMPA instruction or a string-manipulation instruction to an external space, be sure to allocate the data to an area with the same endian as that set for the chip. Also, do not allocate data to be handled by the RMPA instruction or a string-manipulation instruction to an I/O register.
- Set the endian for the area not to be cached. If data in an area that differs from the endian setting of the chip is registered in the cache, it will not operate properly. For information on caching settings, see section 17, Memory-Protection Unit (MPU).
- 64-bit access is not possible. When performing 64-bit access from a master other than CPU0IF, access an area whose endian setting is the same as that for the chip.
- Unaligned access is not possible. When performing unaligned access, access an area whose endian setting is the same as that for the chip.
- Do not access from EDMAC.

In addition, an XSPI is fixed to little endian. Accordingly, if the chip is set as big endian, data cannot be written correctly in unaligned access.

16.2.5.2 Bufferable Write

In bufferable writes, as shown below, the write response does not return from the endpoint of the write access, but from an intermediate point on the bus.

1. CPU0OA accesses any area.
2. DMAC0 accesses PABI, PBBI, PIBI, PSBI, PS0BI, PC0BI, LRAM0BI, and LRAM1IB.

If an access violation occurs due to a bufferable write, no error response is returned to the bus master, but an interrupt or reset request is issued depending on the setting of the BUSOAD register. For details, refer to 16.8.2 Operations When a Bus Error Occurs.

16.2.5.3 Notes on the Timing of Writing Completion

In writing to the MRC0BI or MRC1BI, regardless of whether the write operation is or is not bufferable, the write response is returned from the write buffer for use with the MRAM, not from the MRAM itself.
In writing to the ECBI, regardless of whether the write operation is or is not bufferable, the write response is returned from the CS area controller (CSC) or the SDRAM area controller (SDRAMC) when writing to a CS area or SDRAM starts.

16.3 Register Descriptions

16.3.1 Bus Security Attribution Register A (BUSSARA)

Address(es): CPSCU.BUSSARA 8701 8100h,
CPSCU_NS.BUSSARA 0701 8100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BUSSA0	Bus Security Attribution A0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

BUSSA0 Bit (Bus Security Attribution A0)

This bit specifies the security attributes for arbitration and throughput control register. The target registers are as follows:

- BUSSABT0MRE0BI, BUSSABT0S0BI, BUSSABT0S1BI, BUSSABT0ECBI, BUSSABT0XSPI0BI, BUSSABT0XSPI1BI, BUSSABT0PBBI, BUSSABT0PABI, BUSSABT0PIBI, BUSSABT0PSBI, BUSSABT0PS0BI, BUSSABT1MRC0BI, BUSSABT1MRC1BI

16.3.2 Bus Security Attribution Register B (BUSSARB)

Address(es): CPSCU.BUSSARB 8701 8104h,
CPSCU_NS.BUSSARB 0701 8104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BUSSB0	Bus Security Attribution B0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

BUSSB0 Bit (Bus Security Attribution B0)

This bit specifies the security attributes of registers related to bus errors. The target registers are as follows:

- BUSERRCLRx, MBWERRCLR, BUSOAD, BUSOADPT, BUSINTENx
  x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

16.3.3 Bus Security Attribution Register C (BUSSARC)

Address(es): CPSCU.BUSSARC 8701 8110h,
CPSCU_NS.BUSSARC 0701 8110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BUSSC0	Bus Security Attribution C0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

BUSSC0 Bit (Bus Security Attribution C0)

This bit specifies the security attribute of the bus (SDRAMC/CSC) control register. The target registers are as follows:

- CSnCR (n = 0 to 7), CSnREC (n = 0 to 7), CSRECEN, CSnMOD (n = 0 to 7), CSnWCR1 (n = 0 to 7), CSnWCR2 (n = 0 to 7), SDCCR, SDCMOD, SDAMOD, SDSELF, SDRFCR, SDRFEN, SDICR, SDIR, SDADR, SDTR, SDMOD, SDSR, EXBERR, EXBEREN, EXBECLR, EXBAD, EXHLDEN

16.3.4 CSn Control Register (CSnCR) (n = 0 to 7)

Address(es): BUS.CS0CR 8701 D802h,
BUS_NS.CS0CR 0701 D802h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Address(es): BUS.CS1CR 8701 D812h, BUS.CS2CR 8701 D822h, BUS.CS3CR 8701 D832h, BUS.CS4CR 8701 D842h, BUS.CS5CR 8701 D852h, BUS.CS6CR 8701 D862h, BUS.CS7CR 8701 D872h,
BUS_NS.CS1CR 0701 D812h, BUS_NS.CS2CR 0701 D822h, BUS_NS.CS3CR 0701 D832h, BUS_NS.CS4CR 0701 D842h, BUS_NS.CS5CR 0701 D852h, BUS_NS.CS6CR 0701 D862h, BUS_NS.CS7CR 0701 D872h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: A 32-bit bus space is selected 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n is the same as the endian of operating mode. 1: Endian of area n is not the endian of operating mode.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the CSnCR register while access to the CSn area is in progress.

EXENB Bit (Operation Enable)

This bit enables operation of the associated CS area. On MCU reset, operation is enabled (EXENB = 1) only for area 0. Operation in other areas is disabled (EXENB = 0). Attempts to access disabled areas have no effect.
When the CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] Bits (External Bus Width Select)

These bits specify the data bus width of each area.
The data bus width of area 0 (CS0) after a reset depends on the setting of the bus width in operating mode.
When the address/data multiplexed I/O interface is selected with the MPXEN bit, setting other than 8- and 16-bit bus spaces is prohibited. If set, the operation cannot be guaranteed.

EMODE Bit (Endian Mode)

This bit specifies the endian of each area. For details, refer to section 16.2.5.1, Restriction on Endian.

MPXEN Bit (Address/Data Multiplexed I/O Interface Select)

This bit specifies separate bus interface or address/data multiplexed I/O interface of each area.

16.3.5 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7)

Address(es): BUS.CS0REC 8701 D80Ah, BUS.CS1REC 8701 D81Ah, BUS.CS2REC 8701 D82Ah, BUS.CS3REC 8701 D83Ah, BUS.CS4REC 8701 D84Ah, BUS.CS5REC 8701 D85Ah, BUS.CS6REC 8701 D86Ah, BUS.CS7REC 8701 D87Ah, BUS_NS.CS0REC 0701 D80Ah, BUS_NS.CS1REC 0701 D81Ah, BUS_NS.CS2REC 0701 D82Ah, BUS_NS.CS3REC 0701 D83Ah, BUS_NS.CS4REC 0701 D84Ah, BUS_NS.CS5REC 0701 D85Ah, BUS_NS.CS6REC 0701 D86Ah, BUS_NS.CS7REC 0701 D87Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the CSnREC register while access to the CSn area is in progress.

When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENi (i = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj (j = 0 to 7)) in CSRECEN. For details, refer to section 16.5.4, Insertion of Recovery Cycles.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus for CSn (n = 0 to 7). When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, 1 to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus for CSn (n = 0 to 7). When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, 1 to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

16.3.6 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): BUS.CSRECEN 8701 D880h,
BUS_NS.CSRECEN 0701 D880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the CSRECEN register while access to the CSn area is in progress. For more information on insertion recovery cycles, refer to section 16.6.3, Insertion of Recovery Cycles.

RCVENi Bit (Separate Bus Recovery Cycle Insertion Enable i) (i = 0 to 7)

This bit enables or disables the insertion of read or write recovery cycles when, after a read or write access to the external bus, a read or write access is made to the external bus in the same or different area.

RCVENMj Bit (Multiplexed Bus Recovery Cycle Insertion Enable j) (j = 0 to 7)

This bit enables or disables the insertion of read or write recovery cycles when, after a read or write access to the external bus, a read or write access is made to the external bus in the same or different area.

Table 16.7 Access type associations with the RCVENn bits

Access Type	External Address Space	Insertion of Recovery Cycles	Corresponding Bits (Separate/Multiplexed)
Read access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted for the priority access area.	RCVEN0/RCVENM0
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted for the priority access area.	RCVEN1/RCVENM1
Write access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted for the priority access area.	RCVEN2/RCVENM2
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted for the priority access area.	RCVEN3/RCVENM3
Read access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted for the priority access area.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted for the priority access area.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted for the priority access area.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted for the priority access area.	RCVEN7/RCVENM7

16.3.7 CSn Mode Register (CSnMOD) (n = 0 to 7)

Address(es): BUS.CS0MOD 8701 D002h, BUS.CS1MOD 8701 D012h, BUS.CS2MOD 8701 D022h, BUS.CS3MOD 8701 D032h, BUS.CS4MOD 8701 D042h, BUS.CS5MOD 8701 D052h, BUS.CS6MOD 8701 D062h, BUS.CS7MOD 8701 D072h, BUS_NS.CS0MOD 0701 D002h, BUS_NS.CS1MOD 0701 D012h, BUS_NS.CS2MOD 0701 D022h, BUS_NS.CS3MOD 0701 D032h, BUS_NS.CS4MOD 0701 D042h, BUS_NS.CS5MOD 0701 D052h, BUS_NS.CS6MOD 0701 D062h, BUS_NS.CS7MOD 0701 D072h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.
Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0 to 3) signal corresponding to the respective byte positions.
Writing 1 to this bit selects single write strobe mode where data write operation is controlled by the BCn# (n = 0 to 3) signal and the WR# signal corresponding to respective byte positions.

Note: In single write strobe mode, the external bus width must not be set to 8 bits.

Table 16.8 Control Signals for Write Access Mode

Mode	Pin Name							
Write Access Mode	WR3#	WR2#	WR1#	WR0#/ WR#	BC3#	BC2#	BC1#	BC0#
Byte strobe mode	✓	✓	✓	✓ (WR0#)	—	—	—	—
Single write strobe mode	—	—	—	✓ (WR#)	✓	✓	✓	✓

✓: Enabled, —: Disabled

EWENB Bit (External Wait Enable)

This bit enables or disables external wait.
Writing 0 to this bit disables the WAIT# signal.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

PRENB Bit (Page Read Access Enable)

This bit enables or disables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB Bit (Page Write Access Enable)

This bit enables or disables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, this bit should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD Bit (Page Read Access Mode Select)

This bit selects a page read access operating mode.
Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.
Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

16.3.8 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Address(es): BUS.CS0WCR1 8701 D004h, BUS.CS1WCR1 8701 D014h, BUS.CS2WCR1 8701 D024h, BUS.CS3WCR1 8701 D034h, BUS.CS4WCR1 8701 D044h, BUS.CS5WCR1 8701 D054h, BUS.CS6WCR1 8701 D064h, BUS.CS7WCR1 8701 D074h, BUS_NS.CS0WCR1 0701 D004h, BUS_NS.CS1WCR1 0701 D014h, BUS_NS.CS2WCR1 0701 D024h, BUS_NS.CS3WCR1 0701 D034h, BUS_NS.CS4WCR1 0701 D044h, BUS_NS.CS5WCR1 0701 D054h, BUS_NS.CS6WCR1 0701 D064h, BUS_NS.CS7WCR1 0701 D074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. : : 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. : : 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The CSPWWAIT[2:0] value is valid only when the CSnMOD.PWENB bit is set to 1.

Note 2. The CSPRWAIT[2:0] value is valid only when the CSnMOD.PRENB bit is set to 1.

Do not write to the CSnWCR1 register while access to the CSn area is in progress.
Set each of these bits within a range of the restrictions described in section 16.5.6 (1) Limitations on Using Separate Bus Interface or section 16.5.6 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

CSPWWAIT[2:0] Bits (Page Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle. This setting is enabled when the CSnMOD.PWENB bit is set to 1.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle. This setting is enabled when the CSnMOD.PRENB bit is set to 1.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] Bits (Normal Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

16.3.9 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Address(es): BUS.CS0WCR2 8701 D008h, BUS.CS1WCR2 8701 D018h, BUS.CS2WCR2 8701 D028h, BUS.CS3WCR2 8701 D038h, BUS.CS4WCR2 8701 D048h, BUS.CS5WCR2 8701 D058h, BUS.CS6WCR2 8701 D068h, BUS.CS7WCR2 8701 D078h, BUS_NS.CS0WCR2 0701 D008h, BUS_NS.CS1WCR2 0701 D018h, BUS_NS.CS2WCR2 0701 D028h, BUS_NS.CS3WCR2 0701 D038h, BUS_NS.CS4WCR2 0701 D048h, BUS_NS.CS5WCR2 0701 D058h, BUS_NS.CS6WCR2 0701 D068h, BUS_NS.CS7WCR2 0701 D078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles are inserted. 1 1: Wait with a length of 3 clock cycles are inserted.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the CSnWCR2 register while access to the CSn area is in progress.
Set each of these bits within a range of the restrictions described in section 16.5.6 (1) Limitations on Using Separate Bus Interface or section 16.5.6 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

CSROFF[2:0] Bits (Read-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (RD# signal negated) until the CSn# signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] Bits (Write-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the CSn# signal (n = 0 to 7) is negated in write access mode.

Note: Be sure to satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] Bits (Write Data Output Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the write data output is completed in write access mode.

Note: Be sure to satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] Bits (Address Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value
For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] Bits (RD Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] Bits (WR Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the WRn# signal (n = 0 to 3)) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] Bits (Write Data Output Wait Select)

These bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] Bits (CS Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

16.3.10 SDC Control Register (SDCCR)

Address(es): BUS.SDCCR 8701 DC00h,
BUS_NS.SDCCR 0701 DC00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	SDRAM Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: A 32-bit bus space is selected 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the SDCCR register while an SDRAM area is being accessed.

EXENB Bit (Operation Enable)

This bit enables or disables the operation of the SDRAM address space. After reset, this bit is set to 0 (operation disabled). Attempts to access disabled areas have no effect.
When the CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] Bits (SDRAM Bus Width Select)

These bits specify the data bus width for the corresponding area.

16.3.11 SDC Mode Register (SDCMOD)

Address(es): BUS.SDCMOD 8701 DC01h,
BUS_NS.SDCMOD 0701 DC01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EMODE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EMODE	Endian Mode	0: Endian of SDRAM address space is the same as the endian of operating mode. 1: Endian of SDRAM address space is not the endian of operating mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Writing to this register is possible only once after reset is released when SDCCR.EXENB = 0. Write access more than once is prohibited. If write access is performed more than once, the SDRAM access operation is not guaranteed.

EMODE Bit (Endian Mode)

This bit specifies the endian of the SDRAM address space. For details, re1fer to section 16.2.5.1, Restriction on Endian

16.3.12 SDRAM Access Mode Register (SDAMOD)

Address(es): BUS.SDAMOD 8701 DC02h,
BUS_NS.SDAMOD 0701 DC02h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BE	Continuous Access Enable	0: Continuous access is disabled 1: Continuous access is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set SDAMOD while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

BE Bit (Continuous Access Enable)

This bit enables or disables continuous access to the SDRAM access space.

16.3.13 SDRAM Self-Refresh Control Register (SDSELF)

Address(es): BUS.SDSELF 8701 DC10h,
BUS_NS.SDSELF 0701 DC10h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SFEN	SDRAM Self-Refresh Enable	0: Self-refresh is disabled 1: Self-refresh is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set SDSELF while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

SFEN Bit (SDRAM Self-Refresh Enable)

This bit controls self-refresh operation. Setting this bit to 1 performs auto-refresh cycle operation, after which self-refresh operation begins. Setting this bit to 0 ends self-refresh operation, and auto-refresh operation resumes afterwards. If this bit is set to 1, the value written to this bit is reflected when self-refresh operation starts. If this bit is set to 0, the value written to this bit is reflected after the end of self-refresh operation.

16.3.14 SDRAM Refresh Control Register (SDRFCR)

Address(es): BUS.SDRFCR 8701 DC14h,
BUS_NS.SDRFCR 0701 DC14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFW[3:0]				RFC[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFC[11:0]	Auto-Refresh Request Interval Setting	b11 b0 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles : 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
b15 to b12	REFW[3:0]	Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting	b15 b12 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

RFC[11:0] Bits (Auto-Refresh Request Interval Setting)

These bits specify the auto-refresh request interval. These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN. If auto-refresh is enabled, the value written to these bits is reflected after the end of auto-refresh cycles. The refresh counter operates in SDCLK.

REFW[3:0] Bits (Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting)

These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN. If an auto-refresh cycle is in progress, the value written to these bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes, so the auto-refresh interval may become enlarged in some cases. Set the RFC[11:0] bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. Note that the auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation; in this case, perform self-refresh operation and set the auto-refresh interval appropriate for the frequency again.

16.3.14.1 Auto-Refresh Request Interval and RFC Set Value

SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval.

RFC = (Auto-refresh request interval / SDCLK cycle) – 1

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. If two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

16.3.15 SDRAM Auto-Refresh Control Register (SDRFEN)

Address(es): BUS.SDRFEN 8701 DC16h,
BUS_NS.SDRFEN 0701 DC16h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFEN	Auto-Refresh Operation Enable	0: Auto-refresh operation is disabled 1: Auto-refresh operation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

RFEN Bit (Auto-Refresh Operation Enable)

The RFEN bit enables auto-refresh operation. If auto-refresh is required, set the RFEN bit to 1 before accessing the SDRAM.

When the RFEN bit is set to 0 when auto-refresh is enabled, the RFEN bit becomes 0 after the auto-refresh cycle ends, and auto-refresh operation stops. The interval at which refresh requests are generated is determined by the value of the auto-refresh request interval setting bits (RFC[11:0]) in the SDRAM Refresh Control Register (SDRFCR).

16.3.16 SDRAM Initialization Sequence Control Register (SDICR)

Address(es): BUS.SDICR 8701 DC20h,
BUS_NS.SDICR 0701 DC20h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	INIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INIRQ	Initialization Sequence Start	0: Invalid 1: Initialization sequence starts	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

INIRQ Bit (Initialization Sequence Start)

Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (INIST) in the SDRAM status register (SDSR) to 1. The INIST bit is cleared automatically after the initialization sequence ends.
The value written to the INIRQ bit is not retained.
Please initialize the SDRAM according to the procedure in section 16.6.11.1, SDRAMC Access Procedure.

Note: Set the INIRQ bit to start the SDRAM initialization sequence while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

16.3.17 SDRAM Initialization Register (SDIR)

Address(es): BUS.SDIR 8701 DC24h,
BUS_NS.SDIR 0701 DC24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PRC[2:0]			ARFC[3:0]				ARFI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ARFI[3:0]	Initialization Auto-Refresh Interval	b3 b0 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
b7 to b4	ARFC[3:0]	Initialization Auto-Refresh Count	b7 b4 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W
b10 to b8	PRC[2:0]	Initialization Precharge Cycle Count	b10 b8 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

ARFI[3:0] Bits (Initialization Auto-Refresh Interval)

These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] Bits (Initialization Auto-Refresh Count)

These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

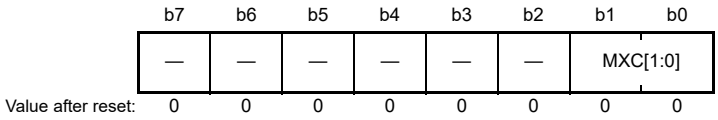
PRC[2:0] Bits (Initialization Precharge Cycle Count)

These bits specify the number of precharge cycles in the SDRAM initialization sequence.

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

16.3.18 SDRAM Address Register (SDADR)

Address(es): BUS.SDADR 8701 DC40h,
BUS_NS.SDADR 0701 DC40h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	MXC[1:0]	Address Multiplex Select	b1 b0 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set SDADR while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

MXC[1:0] Bits (Address Multiplex Select)

These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing.
For details, refer to Table 16.25, Address Multiplexing.

16.3.19 SDRAM Timing Register (SDTR)

Address(es): BUS.SDTR 8701 DC44h,
BUS_NS.SDTR 0701 DC44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RAI[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RCD[1:0]		RP[2:0]			WR	—	—	—	—	—	CL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CL[2:0]	SDRAMC Column Latency	b2 b0 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WR	Write Recovery Interval	0: 1 cycle 1: 2 cycles	R/W
b11 to b9	RP[2:0]	Row Precharge Interval	b11 b9 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
b13, b12	RCD[1:0]	Row Column Latency	b13b12 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RAI[2:0]	Row Active Interval	b18 b16 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

When the SDCCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

The SDTR register specifies the timing for read and write accesses to SDRAM. For details, refer to section 16.6.11.3, Timing Register Settings and Access Timing.

Set the SDTR register while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

CL[2:0] Bits (SDRAMC Column Latency)

These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD).

WR Bit (Write Recovery Interval)

This bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] Bits (Row Precharge Interval)

These bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RCD[1:0] Bits (Row Column Latency)

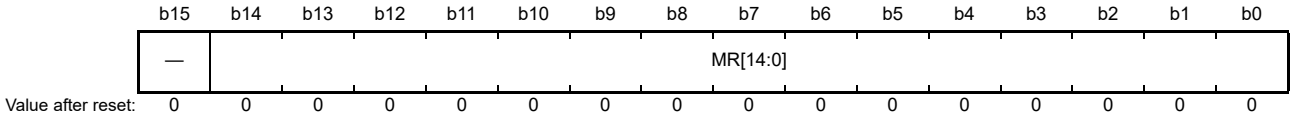
These bits specify the row-column latency of the SDRAM.

RAI[2:0] Bits (Row Active Interval)

These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified by these bits should be less than or equal to the sum of the row-column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

16.3.20 SDRAM Mode Register (SDMOD)

Address(es): BUS.SDMOD 8701 DC48h,
BUS_NS.SDMOD 0701 DC48h



Bit	Symbol	Bit Name	Description	R/W
b14 to b0	MR[14:0]	Mode Register Setting	Writing to these bits: Mode register set command is issued	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The SDMOD register specifies the value to be written to the SDRAM mode register. Writing to this register causes a mode register set command to be issued automatically to SDRAM. Set the SDMOD register while the conditions listed in Table 16.18, Limitations at the Time of Normal Access, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.
Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

MR[14:0] Bits (Mode Register Setting)

Writing to these bits causes a mode register set command to be issued to SDRAM. The setting of the MR[14:0] bits is output to the lower bits of the address. For details, refer to section 16.6.9, Setting Mode Register.
Please write the mode register according to the procedure in section 16.6.11.1, SDRAMC Access Procedure.

Note: The following points should be kept in mind regarding SDMOD settings.
- Make sure to set a burst length of 1 for SDRAM. Operation cannot be guaranteed with settings other than burst length 1.
- The SDRAM column latency must match the setting of the SDRAMC column latency setting bits (CL[2:0]) in the SDRAM timing register (SDTR). Operation cannot be guaranteed if the latency settings do not agree.
- Whenever writing is to proceed, confirm that all of the status bits (SDSR.SRFST, INIST, and MRSST) in the SDRAM status register are 0 beforehand.

16.3.21 SDRAM Status Register (SDSR)

Address(es): BUS.SDSR 8701 DC50h,
BUS_NS.SDSR 0701 DC50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SRFST	INIST	—	—	MRSST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRSST	Mode Register Setting Status	0: Mode register setting not in progress 1: Mode register setting in progress	R
b2, b1	—	Reserved	These bits are read as 0.	R
b3	INIST	Initialization Status	0: Initialization sequence not in progress 1: Initialization sequence in progress	R
b4	SRFST	Self-Refresh Transition/Recovery Status	0: Transition/recovery not in progress 1: Transition/recovery in progress	R
b7 to b5	—	Reserved	These bits are read as 0.	R

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

MRSST Bit (Mode Register Setting Status)

When set to 1, this bit indicates that SDRAM mode register setting is in progress.

INIST Bit (Initialization Status)

When set to 1, this bit indicates that the SDRAM initialization sequence is in progress.

SRFST Bit (Self-Refresh Transition/Recovery Status)

When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for SDRAM. “Transition to or recovery from self-refresh operation in progress” refers to the interval from the point at which the bits listed in Table 16.9 are written until the corresponding commands are issued.

Note: Execution of a self-refresh, an initialization sequence, or mode register setting may only be performed when all the status bits are 0. Do not rewrite the registers (bits) listed in Table 16.9 when any of the status bits (SRFST, INIST, MRSST) is set to 1.

Table 16.9 List of Registers and Bits Requiring Checking Status Bits

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

16.3.22 EXBUS Error Status Register (EXBERR)

Address(es): BUS.EXBERR 8701 D8A0h,
BUS_NS.EXBERR 0701 D8A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BCAN	HLDTO	TO
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TO	External Wait Timeout Error Status Flag	0: No external wait timeout has occurred 1: An external wait timeout has occurred	R
b1	HLDTO	External Bus Hold Timeout Error Status Flag	0: No external bus hold timeout error has occurred 1: An external bus hold timeout error has occurred	R
b2	BCAN	Bus Access Cancel Error Status Flag	0: No bus access cancel has occurred 1: Bus access cancel has occurred	R
b7 to b3	—	Reserved	These bits are read as 0.	R

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

In this register, if there are no errors, the error flag for the first error detected is set to 1. However, if a BCAN error is detected when the HLDTO flag is 1, the BCAN flag will be 1. To enable error detection, clear all error flags and confirm that all flags are 0.

TO Flag (External Wait Timeout Error Status Flag)

The TO flag indicates that a timeout error due to the WAIT# signal has occurred.
If a bus access to the CS area does not finish within 768 external bus clock (BCLK) cycles after the bus access starts, the TO flag becomes 1. Writing 1 to the EXBECLR.TOCLR bit sets the EXBERR.TO flag to 0.
For more information about timeout errors, refer to section 16.8.4, Timeout.

HLDTO Flag (External Bus Hold Timeout Error Status Flag)

The HLDTO flag indicates that an external bus hold timeout error has occurred.
The HLDTO flag is set to 1 when the LOW period of both the BREQ# and BACK# signals is equal to or greater than 768 external bus clock cycles. Writing 1 to the EXBECLR.HLDTOCLR bit sets the EXBERR.HLDTO flag to 0.
For more information about external bus hold timeouts, refer to section 16.8.4, Timeout.

BCAN Flag (Bus Access Cancel Error Status Flag)

When any of the following events occur, the BCAN flag becomes 1, an error response is returned to the bus master, and a slave bus error occurs:

- The CSnCR.EXENB bit is 0 and the CSn area is accessed (n = 0 to 7)
- The SDCCR.EXENB bit is 0 and the SDRAM area is accessed
- The HLDTO flag is 1 and the CSn area or SDRAM area is accessed (n = 0 to 7)

Writing 1 to the EXBECLR.BCANCLR bit sets the EXBERR.BCAN flag to 0.

16.3.23 EXBUS Error Enable Register (EXBEREN)

Address(es): BUS.EXBEREN 8701 D8A4h,
BUS_NS.EXBEREN 0701 D8A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TOEN	External Bus Timeout Detection Enable*1, *2	0: External bus timeout detection is disabled. 1: External bus timeout detection is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. When detection is disabled (the TOEN bit is set to 0), bus access can cause the bus to freeze.

Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

For timeout errors, refer to section 16.8.4, Timeout.

16.3.24 EXBUS Error Status Clear Register (EXBECLR)

Address(es): BUS.EXBECLR 8701 D8A8h,
BUS_NS.EXBECLR 0701 D8A8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BCANCLR	HLDTOCLR	TOCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TOCLR	External Bus Wait Timeout Error Status Clear	0: No effect 1: Clears the EXBERR.TO flag	R/W
b1	HLDTOCLR	External Bus Hold Timeout Error Status Clear	0: No effect 1: Clears the EXBERR.HLDTO flag	R/W
b2	BCANCLR	Bus Access Cancel Error Status Clear	0: No effect 1: Clears the EXBERR.BCAN flag	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

TOCLR Bit (External Bus Wait Timeout Error Status Clear)

Writing 1 to this bit clears the EXBERR.TO flag. Before writing 1 to this bit, release the WAIT# signal assertion.
Writing 0 to this bit has no effect. This bit returns 0.
For information on timeout errors, refer to 16.8.4 Timeout.

HLDTOCLR Bit (External Bus Hold Timeout Error Status Clear)

Writing 1 to this bit clears the EXBERR.HLDTO flag. Before writing 1 to this bit, release the BREQ# signal assertion.
Writing 0 to this bit has no effect. This bit returns 0.
For information on timeout errors, refer to 16.8.4 Timeout.

BCANCLR Bit (Bus Access Cancel Error Status Clear)

Writing 1 to this bit clears the EXBERR.BCAN flag. Before writing 1 to this bit, release the BREQ# signal assertion.
Writing 0 to this bit has no effect. This bit returns 0.

16.3.25 EXBUS Error Address Register (EXBEAD)

Address(es): BUS.EXBEAD 8701 D8ACh,
BUS_NS.EXBEAD 0701 D8ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b18 to b0	—	Reserved	These bits are read as 0.	R
b31 to b19	EADDR[12:0]	EXBUS Error Address	These bits indicate the 13 higher-order bits (bits 31 to 19) of the address corresponding to the bus-access error.	R

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register indicates the address of the bus access that corresponds to an error flag that is set to 1 in the EXBERR register.

If a new error occurs when either the TO or BCAN flag in the EXBERR register is 1, the address is not retained. Writing to this register has no effect. When all flags in the EXBERR register are 0, the EXBEAD register is read as 0.

For more information about timeout errors, refer to section 16.8.4, Timeout.

16.3.26 EXBUS Hold Enable Register (EXHLDEN)

Address(es): BUS.EXHLDEN 8701 D8B4h,
BUS_NS.EXHLDEN 0701 D8B4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HLDEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HLDEN	External bus hold enable	0: External bus hold function is disabled. 1: External bus hold function is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

HLDEN Bit (External bus hold enable)

Setting this bit to 1 enables the external bus hold function.

Do not set this bit to 0 while the external bus is held.

For details, refer to section 16.7, External Bus Hold.

16.3.27 Bus Slave Arbitration Control Register 0 (BUSSABT0MRE0BI, BUSSABT0S0BI, BUSSABT0S1BI, BUSSABT0ECBI, BUSSABT0XSPI0BI, BUSSABT0XSPI1BI, BUSSABT0PBBI, BUSSABT0PABI, BUSSABT0PIBI, BUSSABT0PSBI, BUSSABT0PS0BI)

Address(es): BUS.BUSSABT0MRE0BI 8701 E210h, BUS.BUSSABT0S0BI 8701 E220h, BUS.BUSSABT0S1BI 8701 E228h, BUS.BUSSABT0ECBI 8701 E250h, BUS.BUSSABT0XSPI0BI 8701 E258h, BUS.BUSSABT0XSPI1BI 8701 E260h, BUS.BUSSABT0PBBI 8701 E278h, BUS.BUSSABT0PABI 8701 E280h, BUS.BUSSABT0PIBI 8701 E288h, BUS.BUSSABT0PSBI 8701 E290h, BUS.BUSSABT0PS0BI 8701 E298h, BUS_NS.BUSSABT0MRE0BI 8701 E210h, BUS_NS.BUSSABT0S0BI 8701 E220h, BUS_NS.BUSSABT0S1BI 8701 E228h, BUS_NS.BUSSABT0ECBI 8701 E250h, BUS_NS.BUSSABT0XSPI0BI 8701 E258h, BUS_NS.BUSSABT0XSPI1BI 8701 E260h, BUS_NS.BUSSABT0PBBI 8701 E278h, BUS_NS.BUSSABT0PABI 8701 E280h, BUS_NS.BUSSABT0PIBI 8701 E288h, BUS_NS.BUSSABT0PSBI 8701 E290h, BUS_NS.BUSSABT0PS0BI 8701 E298h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ARBS	Arbitration Select for Each Slave Bus Interface	0: Fixed priority 1: Round-robin	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

ARBS Bit (Arbitration Select for Each Slave Bus Interface)

This bit specifies the priority order among the bus masters connected to each bus interface. For fixed priority, refer to Table 16.10. For round-robin priority, refer to Table 16.11.

Table 16.10 Bus Priorities with Fixed-Priority Arbitration (ARBS = 0)

Bus Slave Arbitration Control Register	Bus Interface	Priority*1
BUSSABT0MRE0BI	MRE0BI	EDMAC > DMAC0 > CPU0OA > CPU0IF
BUSSABT0SnBI (n = 0, 1)	S0BI, S1BI	EDMAC > EXDMAC > DMAC0 > CPU0OA > CPU0IF > XSPIM
BUSSABT0ECBI BUSSABT0XSPI0BI BUSSABT0XSPI1BI	ECBI XSPI0BI XSPI1BI	EDMAC > EXDMAC > DMAC0 > CPU0OA > CPU0IF
BUSSABT0PBBI BUSSABT0PABI BUSSABT0PIBI BUSSABT0PSBI BUSSABT0PS0BI	PBBI PABI PIBI PSBI PS0BI	DMAC0 (W > R) > CPU0OA

Note 1. > : Indicates that the left side has higher priority.

Table 16.11 Bus Priorities with Round-Robin Arbitration (ARBS = 1)

Bus Slave Arbitration Control Register	Bus Interface	Priority*1
BUSSABT0MRE0BI	MRE0BI	EDMAC <-> DMAC0 <-> CPU0OA <-> CPU0IF
BUSSABT0SnBI (n = 0, 1)	S0BI, S1BI	EDMAC <-> EXDMAC <-> DMAC0 <-> CPU0OA <-> CPU0IF <-> XSPIM
BUSSABT0ECBI BUSSABT0XSPI0BI BUSSABT0XSPI1BI	ECBI XSPI0BI XSPI1BI	EDMAC <-> EXDMAC <-> DMAC0 <-> CPU0OA <-> CPU0IF
BUSSABT0PBBI BUSSABT0PABI BUSSABT0PIBI BUSSABT0PSBI BUSSABT0PS0BI	PBBI PABI PIBI PSBI PS0BI	DMAC0 <-> CPU0OA

Note 1. <-> : Indicates round-robin.

16.3.28 Bus Slave Arbitration Control Register 1 (BUSSABT1MRC0BI, BUSSABT1MRC1BI)

Address(es): BUS.BUSSABT1MRC0BI 8701 E200h, BUS.BUSSABT1MRC1BI 8701 E208h, BUS_NS.BUSSABT1MRC0BI 0701 E200h, BUS_NS.BUSSABT1MRC1BI 0701 E208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ARBS[1:0]	Arbitration Select for Each Slave Bus Interface	b1 b0 0 0: Fixed priority 0 1: Setting prohibited 1 0: Combination of round-robin and fixed priority 1 1: Round-robin	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

ARBS[1:0] Bits (Arbitration Select for Each Slave Bus Interface)

These bits specify the priority order among the bus masters connected to each bus interface. For the priority of each setting, see Table 16.12, Table 16.13, and Table 16.14.

Table 16.12 Bus Priorities with Fixed-Priority Arbitration (ARBS[1:0] = 00b)

Bus Slave Arbitration Control Register	Slave Bus Interface	Priority*1
BUSSABT1MRC0BI BUSSABT1MRC1BI	MRC0BI MRC1BI	EDMAC > DMAC0 > CPU0OA > CPU0IF

Note 1. > : Indicates that the left side has higher priority.

Table 16.13 Bus Priorities with Combination of Round-Robin and Fixed Priority Arbitration (ARBS[1:0] = 10b)

Bus Slave Arbitration Control Register	Slave Bus Interface	Priority*1
BUSSABT1MRC0BI BUSSABT1MRC1BI	MRC0BI MRC1BI	(EDMAC <-> DMAC0) > CPU0OA > CPU0IF

Note 1. <-> : Indicates round-robin.
> : Indicates that the left side has higher priority.

Table 16.14 Bus Priorities with Round-Robin Arbitration (ARBS[1:0] = 11b)

Bus Slave Arbitration Control Register	Slave Bus Interface	Priority*1
BUSSABT1MRC0BI BUSSABT1MRC1BI	MRC0BI MRC1BI	(EDMAC <-> DMAC0) <-> CPU0OA <-> CPU0IF

Note 1. <-> : Indicates round-robin.

16.3.29 Bus Error Status Register (BUSERRSTATCPU0IF, BUSERRSTATCPU0OA, BUSERRSTATDMAC0, BUSERRSTATEXDMAC, BUSERRSTATEDMAC, BUSERRSTATXSPI0, BUSERRSTATXSPI1)

Address(es): BUS.BUSERRSTATCPU0IF 8701 E800h, BUS.BUSERRSTATCPU0OA 8701 E810h, BUS.BUSERRSTATDMAC0 8701 E880h, BUS.BUSERRSTATEXDMAC 8701 E890h, BUS.BUSERRSTATEDMAC 8701 E8A0h, BUS.BUSERRSTATXSPI0 8701 E8B0h, BUS.BUSERRSTATXSPI1 8701 E8C0h,
BUS_NS.BUSERRSTATCPU0IF 0701 E800h, BUS_NS.BUSERRSTATCPU0OA 0701 E810h, BUS_NS.BUSERRSTATDMAC0 0701 E880h, BUS_NS.BUSERRSTATEXDMAC 0701 E890h, BUS_NS.BUSERRSTATEDMAC 0701 E8A0h, BUS_NS.BUSERRSTATXSPI0 0701 E8B0h, BUS_NS.BUSERRSTATXSPI1 0701 E8C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MSERRSTAT	ILERRSTAT	MMERRSTAT	—	—	SLERRSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SLERRSTAT	Slave Bus Error Status	0: No error occurred. 1: Error occurred.	R
b2, b1	—	Reserved	These bits are read as 0.	R
b3	MMERRSTAT	Bus Master MPU Error Status	0: No error occurred. 1: Error occurred.	R
b4	ILERRSTAT	Illegal Address Access Error Status	0: No error occurred. 1: Error occurred.	R
b5	MSERRSTAT	Master Security Attribution Unit Error Status	0: No error occurred. 1: Error occurred.	R
b7, b6	—	Reserved	These bits are read as 0.	R

This register is cleared by any reset other than a bus error reset, common memory error reset, or local memory error reset.
If multiple bus errors occur simultaneously, the STAT bit of the highest priority error will be 1.
The priority order is as follows (> : indicates that the left side has higher priority):

Master security attribute unit error > Slave bus error, Protect Zone access error > Bus master MPU error > Invalid address access error

MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT flags are not updated if any of them are set to 1.

SLERRSTAT Flag (Slave Bus Error Status)

When a slave error or Protect Zone access error occurs on the bus, the BUSERRSTATx.SLERRSTAT flag becomes 1.
[Clearing condition]

- Set the BUSERRCLRx.SLERRSTAT bit to 1.

A slave error is an error that occurs in the slave, such as a timeout.
For violations when using a debugger, the SLERRSTAT bit is not set.
For more information about slave errors that occur on the bus, refer to section 16.8, Bus Error Monitoring Section.
x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

MMERRSTAT Flag (Bus Master MPU Error Status)

When a bus master MPU error occurs on the bus, the BUSERRSTATx.MMERRSTAT flag becomes 1.
[Clearing condition]

- Set the BUSERRCLRx.MMERRCLR bit to 1.

For more information about master MPU errors that occur on the bus, refer to section 17, Memory-Protection Unit

(MPU).

ILERRSTAT Flag (Illegal Address Access Error Status)

When an illegal address access error occurs on the bus or a write access occurs to MRE0BI, the BUSERRSTATx.ILERRSTAT flag becomes 1.
[Clearing condition]

- Set the BUSERRCLRx.ILERRCLR bit to 1.

The ILERRSTAT flag is not set if a violation occurs while using a debugger.
For details about illegal address access errors that occur on the bus, refer to 16.8 Bus Error Monitoring Section.

MSERRSTAT Flag (Master Security Attribution Unit Error Status)

When a master security attribute error occurs on the bus, the BUSERRSTATx.MSERRSTAT flag becomes 1.
[Clearing condition]

- Set the BUSERRCLRx.MSERRCLR bit to 1.

For details on master security attribute errors that occur on the bus, refer to 16.8 Bus Error Monitoring Section.

Table 16.15 Associations between Bus Types and Registers

n	Bus Master	Bus Error Status Register	Bus Error Clear Register	Bus Error Interrupt Enable Register
0	CPU0IF	BUSERRSTATCPU0IF	BUSERRCLRCPU0IF	BUSINTENCPU0IF
1	CPU0OA	BUSERRSTATCPU0OA	BUSERRCLRCPU0OA	BUSINTENCPU0OA
8	DMAC0	BUSERRSTATDMAC0	BUSERRCLRDMAC0	BUSINTENDMAC0
9	EXDMAC	BUSERRSTATEXDMAC	BUSERRCLREXDMAC	BUSINTENEXDMAC
10	EDMAC	BUSERRSTATEDMAC	BUSERRCLREDMAC	BUSINTENEDMAC
11	XSPI0	BUSERRSTATXSPI0	BUSERRCLRXSPI0	BUSINTENXSPI0
12	XSPI1	BUSERRSTATXSPI1	BUSERRCLRXSPI1	BUSINTENXSPI1

16.3.30 Bus Error Clear Register (BUSERRCLRCPU0IF, BUSERRCLRCPU0OA, BUSERRCLRDMAC0, BUSERRCLREXDMAC, BUSERRCLREDMAC, BUSERRCLRXSPI0, BUSERRCLRXSPI1)

Address(es): BUS.BUSERRCLRCPU0IF 8701 E804h, BUS.BUSERRCLRCPU0OA 8701 E814h, BUS.BUSERRCLRDMAC0 8701 E884h, BUS.BUSERRCLREXDMAC 8701 E894h, BUS.BUSERRCLREDMAC 8701 E8A4h, BUS.BUSERRCLRXSPI0 8701 E8B4h, BUS.BUSERRCLRXSPI1 8701 E8C4h,
BUS_NS.BUSERRCLRCPU0IF 0701 E804h, BUS_NS.BUSERRCLRCPU0OA 0701 E814h, BUS_NS.BUSERRCLRDMAC0 0701 E884h, BUS_NS.BUSERRCLREXDMAC 0701 E894h, BUS_NS.BUSERRCLREDMAC 0701 E8A4h, BUS_NS.BUSERRCLRXSPI0 0701 E8B4h, BUS_NS.BUSERRCLRXSPI1 0701 E8C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MSERRCLR	ILERRCLR	MMERRCLR	—	—	SLERRCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SLERRCLR	Slave Bus Error Clear	Writing 1 to this bit clears the BUSERRSTATx.SLERRSTAT flag.	R/(W) *1
b2 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	MMERRCLR	Bus Master MPU Error Clear	Writing 1 to this bit clears the BUSERRSTATx.MMERRSTAT flag.	R/(W) *1
b4	ILERRCLR	Illegal Address Access Error Clear	Writing 1 to this bit clears the BUSERRSTATx.ILERRSTAT flag.	R/(W) *1
b5	MSERRCLR	Master Security Attribution Unit Error Clear	Writing 1 to this bit clears the BUSERRSTATx.MSERRSTAT flag.	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 has no effect.

When writing 1 to the error clear bit in the Bus Error Clear Register for each bus, stop the bus access that causes an error in the target bus master.

SLERRCLR Bit (Slave Bus Error Clear)

Writing 1 to this bit clears the BUSERRSTATx.SLERRSTAT flag. This bit is read as 0.

MMERRCLR Bit (Bus Master MPU Error Clear)

Writing 1 to this bit clears the BUSERRSTATx.MMERRSTAT flag. This bit is read as 0.

ILERRCLR Bit (Illegal Address Access Error Clear)

Writing 1 to this bit clears the BUSERRSTATx.ILERRSTAT flag. This bit is read as 0.

MSERRCLR Bit (Master Security Attribution Unit Error Clear)

Writing 1 to this bit clears the BUSERRSTATx.MSERRSTAT flag. This bit is read as 0.

16.3.31 Bus Operation after Detection Register (BUSOAD)

Address(es): BUS.BUSOAD 8701 E000h,
BUS_NS.BUSOAD 0701 E000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BWERROAD	SLERROAD	ILERROAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ILERROAD	Illegal Address Access Error Operation After Detection	0: Interrupt 1: Reset	R/W
b1	SLERROAD	Slave Bus Error Operation After Detection	0: Interrupt 1: Reset	R/W
b2	BWERROAD	Bufferable Write Error Operation After Detection	0: Interrupt 1: Reset	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: When an error occurs during an unexpected access from the CPU while the operation bit after each error detection or the BUSINTENx.EN bit is set to 1, a system reset request may be generated.
(x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1)

ILERROAD Bit (Illegal Address Access Error Operation After Detection)

This bit specifies the operation to be taken when the illegal address access error is detected.
If this bit is 0 and BUSINTENx.EN = 1, an error response is returned and interrupt request is generated. If this bit is 1 and BUSINTENx.EN = 1, error response is returned and reset request is generated.
x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

SLERROAD Bit (Slave Bus Error Operation After Detection)

This bit specifies the operation to be taken when the slave bus error is detected.
If this bit is 0 and BUSINTENx.EN = 1, an error response is returned and interrupt request is generated. If this bit is 1 and BUSINTENx.EN = 1, error response is returned and reset request is generated.

BWERROAD Bit (Bufferable Write Error Operation After Detection)

This bit specifies the operation to be taken when the bufferable write error is detected.
If this bit is 0 and BUSINTENx.EN = 1, an error response is returned and interrupt request is generated. If this bit is 1 and BUSINTENx.EN = 1, error response is returned and reset request is generated.
In addition, interrupts due to bus errors (including bufferable errors) can also be used as non-maskable interrupts. For details, refer to the NMIER register specifications in section 15, Interrupt Controller (ICUM)

16.3.32 Bus Operation after Detection Protect Register (BUSOADPT)

Address(es): BUS.BUSOADPT 8701 E004h,
BUS_NS.BUSOADPT 0701 E004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PROTECT	Protection of Register	0: Writing to the BUSOAD register is enabled. 1: Writing to the BUSOAD register is disabled. Reading is possible.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Enable or disable writing to the PROTECT bit	W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

PROTECT Bit (Protection of Register)

This bit enables or disables writing to the associated registers to be protected.

This bit controls the following registers:

- BUSOAD

When the PROTECT bit is set simultaneously, write A5h to the KEY[7:0] bits by accessing them in 16-bit units.

KEY[7:0] Bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit simultaneously, write A5h to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

16.3.33 Master Bufferable Write Error Status Register (MBWERRSTAT)

Address(es): BUS.MBWERRSTAT 8701 EB00h,
BUS_NS.MBWERRSTAT 0701 EB00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	MBWERR19	MBWERR18	MBWERR17	MBWERR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MBWERR8	—	—	—	—	—	—	—	MBWERR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MBWERR0	Master Bufferable Write Error 0 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	MBWERR8	Master Bufferable Write Error 8 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b15 to b9	—	Reserved	These bits are read as 0.	R
b16	MBWERR16	Master Bufferable Write Error 16 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b17	MBWERR17	Master Bufferable Write Error 17 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b18	MBWERR18	Master Bufferable Write Error 18 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b19	MBWERR19	Master Bufferable Write Error 19 Flag	0: No bufferable write error occurred 1: Bufferable write error occurred	R
b31 to b20	—	Reserved	These bits are read as 0.	R

Note: This register is cleared by reset other than bus error reset, common memory error reset, and local memory error reset.

MBWERRn Flag (Master Bufferable Write Error n Flag) (n = 0, 8, 16 to 19)

When bufferable write access error has occurred, the MBWERRSTAT.MBWERRn flag is set to 1.

[Clearing condition]

- Set the MBWERRCLR.MBWECLRn bit to 1.

Table 16.16 lists the registers associated with each bus master type.

Table 16.16 Associations between Bus Master Types and Bits

Bus Master type	Master Bufferable Write Error Status Flag	Master Bufferable Write Error Clear Bit
CPU0OA	MBWERR0	MBWECLR0
DMAC/DTC0	MBWERR8	MBWECLR8
EXDMAC	MBWERR16	MBWECLR16
EDMAC	MBWERR17	MBWECLR17
XSPI0	MBWERR18	MBWECLR18
XSPI1	MBWERR19	MBWECLR19

16.3.34 Master Bufferable Write Error Clear Register (MBWERRCLR)

Address(es): BUS.MBWERRCLR 8701 EB08h,
BUS_NS.MBWERRCLR 0701 EB08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	MBWECLR19	MBWECLR18	MBWECLR17	MBWECLR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MBWECLR8	—	—	—	—	—	—	—	MBWECLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MBWECLR0	Master Bufferable Write Error 0 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR0 flag.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MBWECLR8	Master Bufferable Write Error 8 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR8 flag.	R/(W)*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	MBWECLR16	Master Bufferable Write Error 16 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR16 flag.	R/(W)*1
b17	MBWECLR17	Master Bufferable Write Error 17 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR17 flag.	R/(W)*1
b18	MBWECLR18	Master Bufferable Write Error 18 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR18 flag.	R/(W)*1
b19	MBWECLR19	Master Bufferable Write Error 19 Flag Clear	Writing 1 to this bit clears the MBWERSTAT.MBWERR19 flag.	R/(W)*1
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 has no effect.

MBWECLRn Bit (Master Bufferable Write Error n Flag Clear) (n = 0, 8, 16 to 19)

Writing 1 to this bit clears the MBWERRSTAT.MBWERRn flag.

This bit is read as 0.

16.3.35 Bus Error Interrupt Enable Register (BUSINTENCPU0IF, BUSINTENCPU0OA, BUSINTENDMAC0, BUSINTENEXDMAC, BUSINTENEDMAC, BUSINTENXSPI0, BUSINTENXSPI1)

Address(es): BUS.BUSINTENCPU0IF 8701 E808h, BUS.BUSINTENCPU0OA 8701 E818h, BUS.BUSINTENDMAC0 8701 E888h, BUS.BUSINTENEXDMAC 8701 E898h, BUS.BUSINTENEDMAC 8701 E8A8h, BUS.BUSINTENXSPI0 8701 E8B8h, BUS.BUSINTENXSPI1 8701 E8C8h,
BUS_NS.BUSINTENCPU0IF 0701 E808h, BUS_NS.BUSINTENCPU0OA 0701 E818h, BUS_NS.BUSINTENDMAC0 0701 E888h, BUS_NS.BUSINTENEXDMAC 0701 E898h, BUS_NS.BUSINTENEDMAC 0701 E8A8h, BUS_NS.BUSINTENXSPI0 0701 E8B8h, BUS_NS.BUSINTENXSPI1 0701 E8C8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN	Bus Error Interrupt Request Enable	0: Disabled 1: Enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

When writing 1 to the BUSINTENx.EN bit, stop the bus access that causes an error in the target bus master.
x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

EN Bit (Bus Error Interrupt Request Enable)

To output a bus error interrupt request to the ICU due to an error except for a master bufferable write error, select the interrupt in the BUSOAD setting corresponding to each error, and enable the interrupt request with this bit. The setting of this bit does not depend on the issuance of a bus error interrupt request due to a master bufferable write error.

16.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), data size, and endian format when accessing the external address space (the CS and SDRAM areas).

16.4.1 Data Alignment Control for CS Area

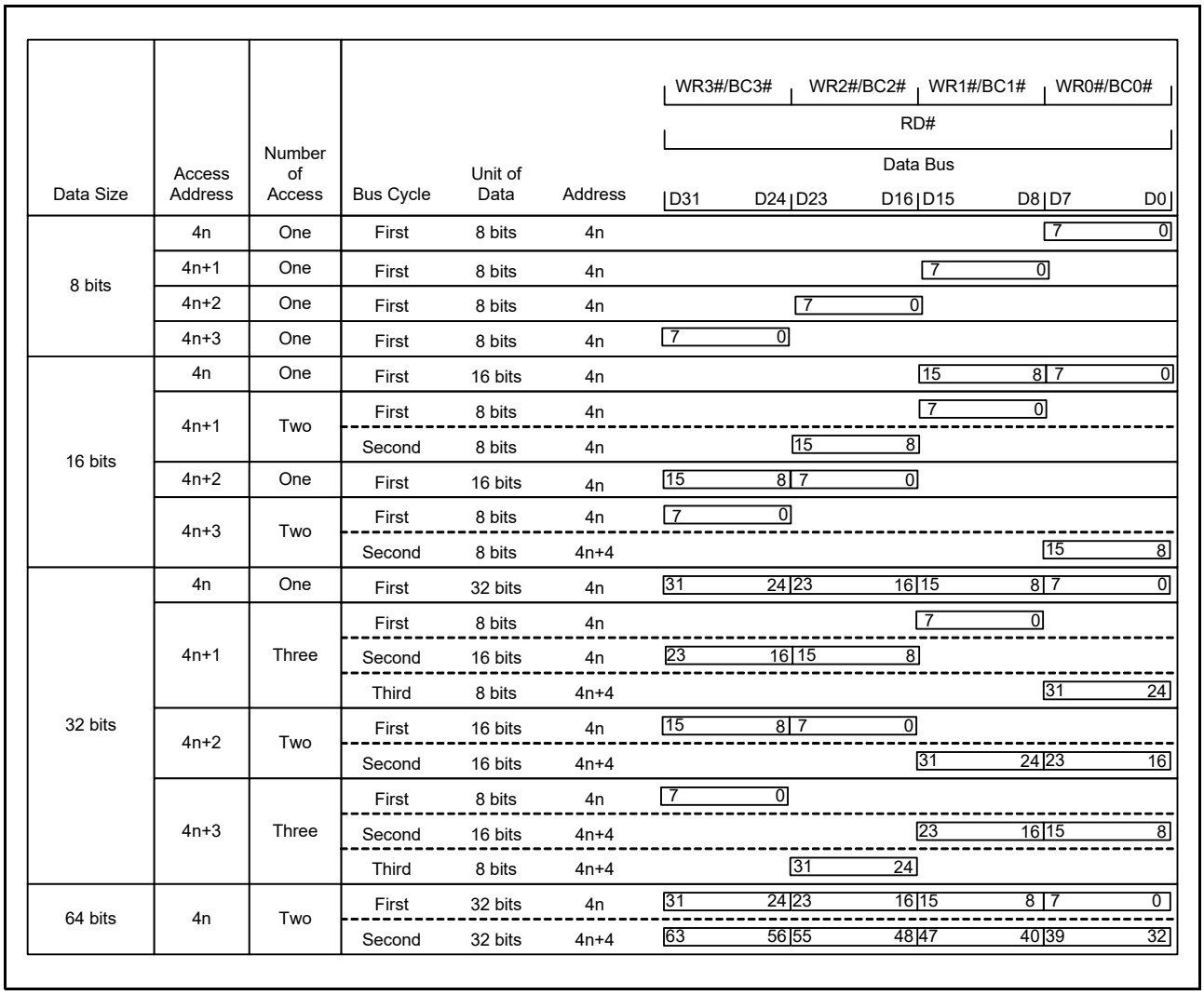
(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output the low level).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# to WR3# pins are enabled. The BC0# to BC3# pins are not used.

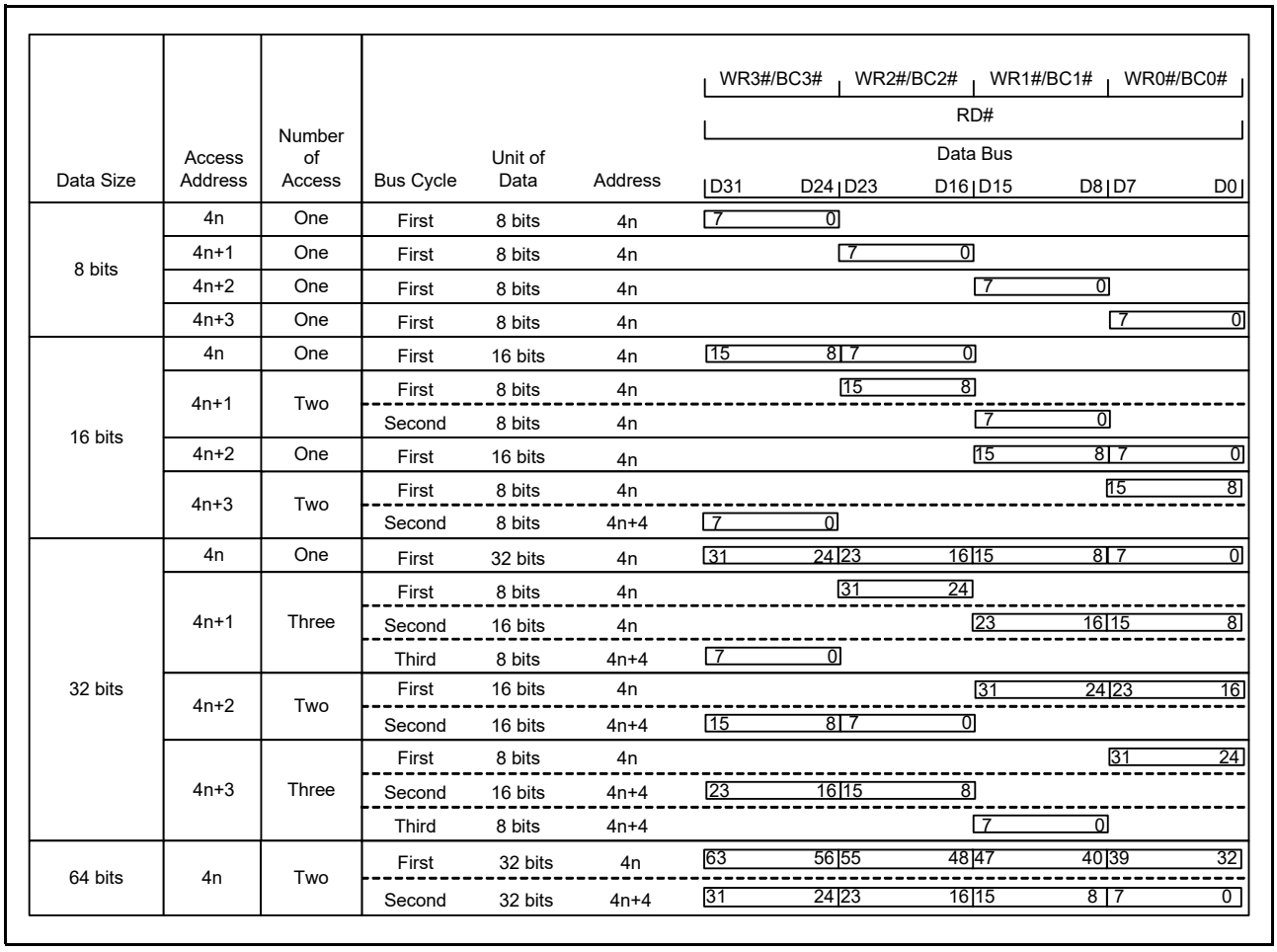
When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# to BC3# pins.

In 32-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address
8 bits	4n	One	First	8 bits	4n
	4n+1	One	First	8 bits	4n
	4n+2	One	First	8 bits	4n
	4n+3	One	First	8 bits	4n
16 bits	4n	One	First	16 bits	4n
	4n+1	Two	First	8 bits	4n
			Second	8 bits	4n
	4n+2	One	First	16 bits	4n
	4n+3	Two	First	8 bits	4n
			Second	8 bits	4n+4
32 bits	4n	One	First	32 bits	4n
	4n+1	Three	First	8 bits	4n
			Second	16 bits	4n
			Third	8 bits	4n+4
	4n+2	Two	First	16 bits	4n
			Second	16 bits	4n+4
	4n+3	Three	First	8 bits	4n
			Second	16 bits	4n+4
			Third	8 bits	4n+4
64 bits	4n	Two	First	32 bits	4n
			Second	32 bits	4n+4

Figure 16.4 Data Alignment (Little Endian) in 32-Bit Bus Space



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address
8 bits	4n	One	First	8 bits	4n
	4n+1	One	First	8 bits	4n
	4n+2	One	First	8 bits	4n
	4n+3	One	First	8 bits	4n
16 bits	4n	One	First	16 bits	4n
	4n+1	Two	First	8 bits	4n
			Second	8 bits	4n
	4n+2	One	First	16 bits	4n
	4n+3	Two	First	8 bits	4n
			Second	8 bits	4n+4
32 bits	4n	One	First	32 bits	4n
	4n+1	Three	First	8 bits	4n
			Second	16 bits	4n
			Third	8 bits	4n+4
	4n+2	Two	First	16 bits	4n
			Second	16 bits	4n+4
	4n+3	Three	First	8 bits	4n
			Second	16 bits	4n+4
			Third	8 bits	4n+4
64 bits	4n	Two	First	32 bits	4n
			Second	32 bits	4n+4

Figure 16.5 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address bus A0 is disabled (always output the low level).
When byte strobe mode is selected (CSnMOD.WRMOD = 0), the WR0# and WR1# pins are enabled, and the WR2# and WR3# pins are disabled (fixed high). The BC0# to BC3# pins are not used.
When single write strobe mode is selected (CSnMOD.WRMOD = 1), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# and BC1# pins.
In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little. See Figure 16.6 and Figure 16.7.
In 16-bit bus space, page access can occur in access to data in 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary and causes no change in BC0# and BC1# signals. The situations in which page access occurs are indicated by the letter (p) in Figure 16.6 and Figure 16.7.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# / Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n		15 8 7 0
	4n+1	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+2		15 8
	4n+2	One	First	16 bits	4n+2		15 8 7 0
	4n+3	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Two	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2	(p)	31 24 23 16
	4n+1	Three	First	8 bits	4n		7 0
			Second	16 bits	4n+2		23 16 15 8
			Third	8 bits	4n+4		31 24
	4n+2	Two	First	16 bits	4n+2		15 8 7 0
			Second	16 bits	4n+4		31 24 23 16
	4n+3	Three	First	8 bits	4n+2		7 0
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+6		31 24
64 bits	4n	Four	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2	(p)	31 24 23 16
			Third	16 bits	4n+4		47 40 39 32
			Fourth	16 bits	4n+6	(p)	63 56 55 48

(p): Page access (only when page access is enabled with the CSnMOD.PRENB and PWENB bits)

Figure 16.6 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# Data Bus D15 D8	WR0#/BC0# RD# Data Bus D7 D0
8 bits	4n	One	First	8 bits	4n		7 0	
	4n+1	One	First	8 bits	4n			7 0
	4n+2	One	First	8 bits	4n+2		7 0	
	4n+3	One	First	8 bits	4n+2			7 0
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+1	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+2		7 0	
	4n+2	One	First	16 bits	4n+2		15 8	7 0
	4n+3	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+4		7 0	
32 bits	4n	Two	First	16 bits	4n		31 24	23 16
			Second	16 bits	4n+2	(p)	15 8	7 0
	4n+1	Three	First	8 bits	4n			31 24
			Second	16 bits	4n+2		23 16	15 8
			Third	8 bits	4n+4		7 0	
	4n+2	Two	First	16 bits	4n+2		31 24	23 16
			Second	16 bits	4n+4		15 8	7 0
	4n+3	Three	First	8 bits	4n+2			31 24
			Second	16 bits	4n+4		23 16	15 8
			Third	8 bits	4n+6		7 0	
64 bits	4n	Four	First	16 bits	4n		63 56	55 48
			Second	16 bits	4n+2	(p)	47 40	39 32
			Third	16 bits	4n+4		31 24	23 16
			Fourth	16 bits	4n+6	(p)	15 8	7 0

(p): Page access (only when page access is enabled with the CSnMOD.PRENB and PWENB bits)

Figure 16.7 Data Alignment (Big Endian) in 16-Bit Bus Space

(3) 8-Bit Bus Space

When an 8-bit bus space is selected by the CSnCR.BSIZE[1:0] bits, the address buses A23 to A0 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and always outputs the low level during write access. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.

In 8-bit bus space, page access can occur in access to data in 16-, 32- or 64-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 16.8 and Figure 16.9.

In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# WR0#/BC0# RD# Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n+1		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+3		7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
	4n+1	Two	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
	4n+3	Two	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
			Third	8 bits	4n+2	(p)	23 16
			Fourth	8 bits	4n+3	(p)	31 24
	4n+1	Four	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2	(p)	15 8
			Third	8 bits	4n+3	(p)	23 16
			Fourth	8 bits	4n+4		31 24
	4n+2	Four	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3	(p)	15 8
			Third	8 bits	4n+4		23 16
			Fourth	8 bits	4n+5	(p)	31 24
	4n+3	Four	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
			Third	8 bits	4n+5	(p)	23 16
			Fourth	8 bits	4n+6	(p)	31 24
64 bits	4n	Eight	First	8 bits	4n		7 0
			Second	8 bits	4n+1	(p)	15 8
			Third	8 bits	4n+2	(p)	23 16
			Fourth	8 bits	4n+3	(p)	31 24
			Fifth	8 bits	4n+4		39 32
			Sixth	8 bits	4n+5	(p)	47 40
			Seventh	8 bits	4n+6	(p)	55 48
			Eighth	8 bits	4n+7	(p)	63 56

(p): Page access (only when page access is enabled with the CSnMOD.PRENB and PWENB bits)

Figure 16.8 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# Data Bus D15 D8	WR0#/BC0# Data Bus D7 D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+1	Two	First	8 bits	4n+1			15 8
			Second	8 bits	4n+2	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
	4n+3	Two	First	8 bits	4n+3			15 8
			Second	8 bits	4n+4			7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0
	4n+1	Four	First	8 bits	4n+1			31 24
			Second	8 bits	4n+2	(p)		23 16
			Third	8 bits	4n+3	(p)		15 8
			Fourth	8 bits	4n+4			7 0
	4n+2	Four	First	8 bits	4n+2			31 24
			Second	8 bits	4n+3	(p)		23 16
			Third	8 bits	4n+4			15 8
			Fourth	8 bits	4n+5	(p)		7 0
	4n+3	Four	First	8 bits	4n+3			31 24
			Second	8 bits	4n+4			23 16
			Third	8 bits	4n+5	(p)		15 8
			Fourth	8 bits	4n+6	(p)		7 0
64 bits	4n	Eight	First	8 bits	4n			63 56
			Second	8 bits	4n+1	(p)		55 48
			Third	8 bits	4n+2	(p)		47 40
			Fourth	8 bits	4n+3	(p)		39 32
			Fifth	8 bits	4n+4			31 24
			Sixth	8 bits	4n+5	(p)		23 16
			Seventh	8 bits	4n+6	(p)		15 8
			Eighth	8 bits	4n+7	(p)		7 0

(p): Page access (only when page access is enabled with the CSnMOD.PRENB and PWENB bits)

Figure 16.9 Data Alignment (Big Endian) in 8-Bit Bus Space

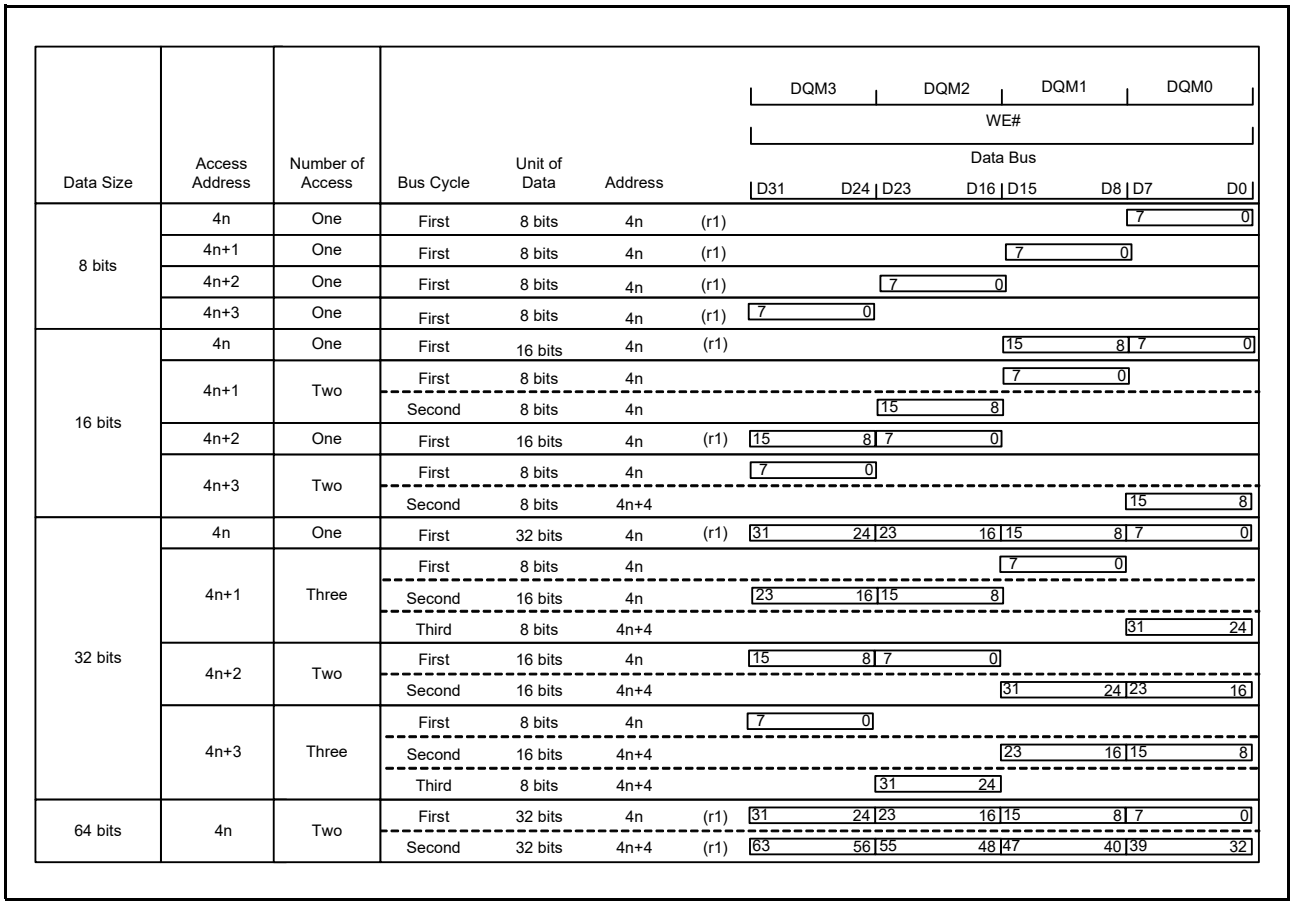
16.4.2 Data Alignment Control for SDRAM Area

(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the SDCCR.BSIZE[1:0] bits, the address buses A26 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output the low level). The valid byte position is indicated by DQM0 to DQM3 signals. The external data is accessed using the D31 to D24, D23 to D16, D15 to D8, and D7 to D0 pins. Data widths that can be handled in a single cycle of access are 8, 16 and 32 bits.

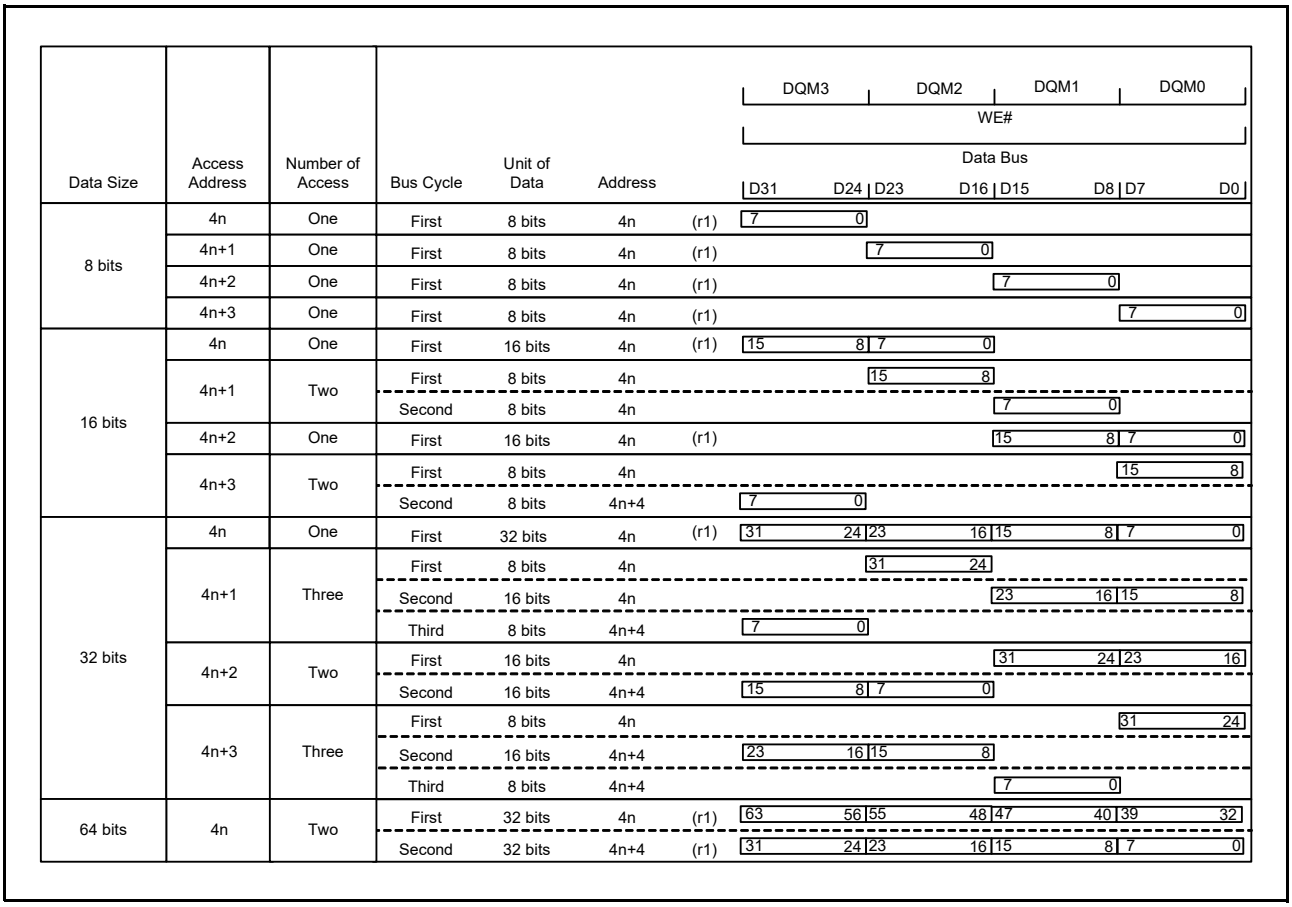
In 32-bit bus space, the valid positions of data external to the chip and of SDRAM control signals (DQM0 to DQM3) differ according to whether the endian is big or little. Figure 16.10 and Figure 16.11 show data alignment control when the endian is little and big, respectively.

When the conditions indicated by (r1) in Figure 16.10 and Figure 16.11 are satisfied while the SDAMOD.BE bit is set to 1, the SDRAM area controller continues the access without reasserting the same row address, enabling consecutive rounds of memory accesses to the SDRAM.



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	
8 bits	4n	One	First	8 bits	4n	(r1)
	4n+1	One	First	8 bits	4n	(r1)
	4n+2	One	First	8 bits	4n	(r1)
	4n+3	One	First	8 bits	4n	(r1)
16 bits	4n	One	First	16 bits	4n	(r1)
	4n+1	Two	First	8 bits	4n	
			Second	8 bits	4n	
	4n+2	One	First	16 bits	4n	(r1)
	4n+3	Two	First	8 bits	4n	
			Second	8 bits	4n+4	
32 bits	4n	One	First	32 bits	4n	(r1)
	4n+1	Three	First	8 bits	4n	
			Second	16 bits	4n	
			Third	8 bits	4n+4	
	4n+2	Two	First	16 bits	4n	
			Second	16 bits	4n+4	
	4n+3	Three	First	8 bits	4n	
			Second	16 bits	4n+4	
			Third	8 bits	4n+4	
64 bits	4n	Two	First	32 bits	4n	(r1)
			Second	32 bits	4n+4	(r1)

Figure 16.10 Data Alignment (Little Endian) in 32-Bit Bus Space of SDRAM Area



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	
8 bits	4n	One	First	8 bits	4n	(r1)
	4n+1	One	First	8 bits	4n	(r1)
	4n+2	One	First	8 bits	4n	(r1)
	4n+3	One	First	8 bits	4n	(r1)
16 bits	4n	One	First	16 bits	4n	(r1)
	4n+1	Two	First	8 bits	4n	
			Second	8 bits	4n	
	4n+2	One	First	16 bits	4n	(r1)
	4n+3	Two	First	8 bits	4n	
			Second	8 bits	4n+4	
32 bits	4n	One	First	32 bits	4n	(r1)
	4n+1	Three	First	8 bits	4n	
			Second	16 bits	4n	
			Third	8 bits	4n+4	
	4n+2	Two	First	16 bits	4n	
			Second	16 bits	4n+4	
	4n+3	Three	First	8 bits	4n	
			Second	16 bits	4n+4	
			Third	8 bits	4n+4	
64 bits	4n	Two	First	32 bits	4n	(r1)
			Second	32 bits	4n+4	(r1)

Figure 16.11 Data Alignment (Big Endian) in 32-Bit Bus Space of SDRAM Area

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the SDCCR.BSIZE[1:0] bits, the address buses A26 to A1 are enabled to output address signals in units of 16 bits, and the address buses A0 is disabled (always output the low level). The valid byte position is indicated by DQM0 and DQM1 signals.

In 16-bit bus space, the external data is accessed using the D15 to D8 and D7 to D0 pins and DQM0 and DQM1 control signals. Data widths that can be handled in a single cycle of access are 8 and 16 bits.

When the conditions indicated by (r1) in Figure 16.12 and Figure 16.13 are satisfied while the SDAMOD.BE bit is set to 1, the SDRAM area controller continues the access without reasserting the same row address, enabling consecutive rounds of memory accesses to the SDRAM.

The valid positions of control signals and data outside the MCU depend on the endian order. For details, see Figure 16.12 and Figure 16.13.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus (D15 D8 D7 D0)
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+2	(r1)	7 0
16 bit	4n	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+1	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+2		15 8
	4n+2	One	First	16 bits	4n+2	(r1)	15 8 7 0
	4n+3	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Two	First	16 bits	4n	(r1)	15 8 7 0
			Second	16 bits	4n+2	(r1)	31 24 23 16
	4n+1	Three	First	8 bits	4n		7 0
			Second	16 bits	4n+2		23 16 15 8
			Third	8 bits	4n+4		31 24
	4n+2	Two	First	16 bits	4n+2		15 8 7 0
			Second	16 bits	4n+4		31 24 23 16
	4n+3	Three	First	8 bits	4n+2		7 0
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+6		31 24
64 bits	4n	Four	First	16 bits	4n	(r1)	15 8 7 0
			Second	16 bits	4n+2	(r1)	31 24 23 16
			Third	16 bits	4n+4	(r1)	47 40 39 32
			Fourth	16 bits	4n+6	(r1)	63 56 55 48

Figure 16.12 Data Alignment (Little Endian) in 16-Bit Bus Space of SDRAM Area

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus (D15 D8 | D7 D0)
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+2	(r1)	7 0
16 bits	4n	One	First	16 bits	4n	(r1)	15 8 | 7 0
	4n+1	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+2		7 0
	4n+2	One	First	16 bits	4n+2	(r1)	15 8 | 7 0
	4n+3	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+4		7 0
32 bits	4n	Two	First	16 bits	4n	(r1)	31 24 | 23 16
			Second	16 bits	4n+2	(r1)	15 8 | 7 0
	4n+1	Three	First	8 bits	4n		31 24
			Second	16 bits	4n+2		23 16 | 15 8
			Third	8 bits	4n+4		7 0
	4n+2	Two	First	16 bits	4n+2		31 24 | 23 16
			Second	16 bits	4n+4		15 8 | 7 0
	4n+3	Three	First	8 bits	4n+2		31 24
			Second	16 bits	4n+4		23 16 | 15 8
			Third	8 bits	4n+6		7 0
64 bits	4n	Four	First	16 bits	4n	(r1)	63 56 | 55 48
			Second	16 bits	4n+2	(r1)	47 40 | 39 32
			Third	16 bits	4n+4	(r1)	31 24 | 23 16
			Fourth	16 bits	4n+6	(r1)	15 8 | 7 0

Figure 16.13 Data Alignment (Big Endian) in 16-Bit Bus Space of SDRAM Area

(3) 8-Bit Bus Space

When an 8-bit width is selected for a bus space by the SDCCR.BSIZE[1:0] bits, the address buses A26 to A0 are enabled to output address signals in units of 8 bits.
In 8-bit bus space, the external data is accessed using the D7 to D0 pins and DQM0 control signal. Eight-bit data can be accessed at a time; 16-bit data is accessed with two 8-bit accesses, and 32-bit data is accessed with four 8-bit accesses.
When the conditions indicated by (r1) in Figure 16.14 and Figure 16.15 are satisfied while the SDAMOD.BE bit is set to 1, the SDRAM area controller continues the access without reasserting the same row address, enabling consecutive rounds of memory accesses to the SDRAM.
The valid positions of control signals and data outside the MCU depend on the endian order. For details, see Figure 16.14 and Figure 16.15.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n	(r1)	7 0
			Second	8 bits	4n+1	(r1)	15 8
	4n+1	Two	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2		15 8
	4n+2	Two	First	8 bits	4n+2	(r1)	7 0
			Second	8 bits	4n+3	(r1)	15 8
	4n+3	Two	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Four	First	8 bits	4n	(r1)	7 0
			Second	8 bits	4n+1	(r1)	15 8
			Third	8 bits	4n+2	(r1)	23 16
			Fourth	8 bits	4n+3	(r1)	31 24
	4n+1	Four	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2		15 8
			Third	8 bits	4n+3		23 16
			Fourth	8 bits	4n+4		31 24
	4n+2	Four	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3		15 8
			Third	8 bits	4n+4		23 16
			Fourth	8 bits	4n+5		31 24
	4n+3	Four	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
			Third	8 bits	4n+5		23 16
			Fourth	8 bits	4n+6		31 24
64 bits	4n	Eight	First	8 bits	4n	(r1)	7 0
			Second	8 bits	4n+1	(r1)	15 8
			Third	8 bits	4n+2	(r1)	23 16
			Fourth	8 bits	4n+3	(r1)	31 24
			Fifth	8 bits	4n+4	(r1)	39 32
			Sixth	8 bits	4n+5	(r1)	47 40
			Seventh	8 bits	4n+6	(r1)	55 48
			Eighth	8 bits	4n+7	(r1)	63 56

Figure 16.14 Data Alignment (Little Endian) in 8-Bit Bus Space of SDRAM Area

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n	(r1)	15 8
			Second	8 bits	4n+1	(r1)	7 0
	4n+1	Two	First	8 bits	4n+1		15 8
			Second	8 bits	4n+2		7 0
	4n+2	Two	First	8 bits	4n+2	(r1)	15 8
			Second	8 bits	4n+3	(r1)	7 0
	4n+3	Two	First	8 bits	4n+3		15 8
			Second	8 bits	4n+4		7 0
32 bits	4n	Four	First	8 bits	4n	(r1)	31 24
			Second	8 bits	4n+1	(r1)	23 16
			Third	8 bits	4n+2	(r1)	15 8
			Fourth	8 bits	4n+3	(r1)	7 0
	4n+1	Four	First	8 bits	4n+1		31 24
			Second	8 bits	4n+2		23 16
			Third	8 bits	4n+3		15 8
			Fourth	8 bits	4n+4		7 0
	4n+2	Four	First	8 bits	4n+2		31 24
			Second	8 bits	4n+3		23 16
			Third	8 bits	4n+4		15 8
			Fourth	8 bits	4n+5		7 0
	4n+3	Four	First	8 bits	4n+3		31 24
			Second	8 bits	4n+4		23 16
			Third	8 bits	4n+5		15 8
			Fourth	8 bits	4n+6		7 0
64 bits	4n	Eight	First	8 bits	4n	(r1)	63 56
			Second	8 bits	4n+1	(r1)	55 48
			Third	8 bits	4n+2	(r1)	47 40
			Fourth	8 bits	4n+3	(r1)	39 32
			Fifth	8 bits	4n+4	(r1)	31 24
			Sixth	8 bits	4n+5	(r1)	23 16
			Seventh	8 bits	4n+6	(r1)	15 8
			Eighth	8 bits	4n+7	(r1)	7 0

Figure 16.15 Data Alignment (Big Endian) in 8-Bit Bus Space of SDRAM Area

16.5 Operation of CS Area Controller

16.5.1 Separate Bus

The various periods in the timing charts are described below.
The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and EBCLK pin output are the same, unless otherwise noted.
Access via the external bus starts at the same point as the output of a rising edge on the EBCLK pin. However, if the external bus clock (BCLK) and the output on the EBCLK pin are at different frequencies so that a single request from a bus master for transfer leads to two or more rounds of access via the external bus, the wait settings may cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin (refer to Figure 16.18 to Figure 16.27). If recovery cycles are inserted for bus access, the setting for the number of recovery cycles may also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin (refer to Figure 16.45).

(a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles are selectable within the range from zero to 31. The respective wait settings determine the timing of the assertion of the CSn#, RD#, and WRn# signals during this period. Specifically, the periods of waiting are controlled by the following bits in CSn wait control register 2 (CSnWCR2).

- the CS assert wait select bits (CSON[2:0])
- the RD assert wait select bits (RDON[2:0])
- the WR assert wait select bits (WRON[2:0])
- the write data output wait select bits (WDON[2:0])

The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

(b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.
If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.
After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (see (e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (see (d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF[2:0]) and the write-access CS extension cycle select bits (CSWOFF[2:0]) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles are counted from the cycle following the cycle where the strobe signal is valid.
In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.
For write access, setting the write data output extension cycle select bits (WDOFF[2:0]) controls extension of the period where the address and output data are valid.

(d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).
In the case of normal access, this is inserted within the period of clock cycles of CS extension (see (c) Tn1 to Tnm (Clock Cycles of CS Extension)).
In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (see (c) Tn1 to Tnm (Clock Cycles of CS Extension)). Valid address and data output are extended over this period, and the WRn# signal is negated.

(e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.
CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, refer to section 16.5.4, Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in the CSnMOD register are set to 0 to disable page-read and page-write access, respectively, all bus accesses will take the form of normal access operations.
Even when the PRENB and PWENB bits are set to 1 to enable page-read and page-write access, respectively, bus access other than page access will take the form of normal access operations.
Figure 16.16 to Figure 16.18 show the normal access operations.

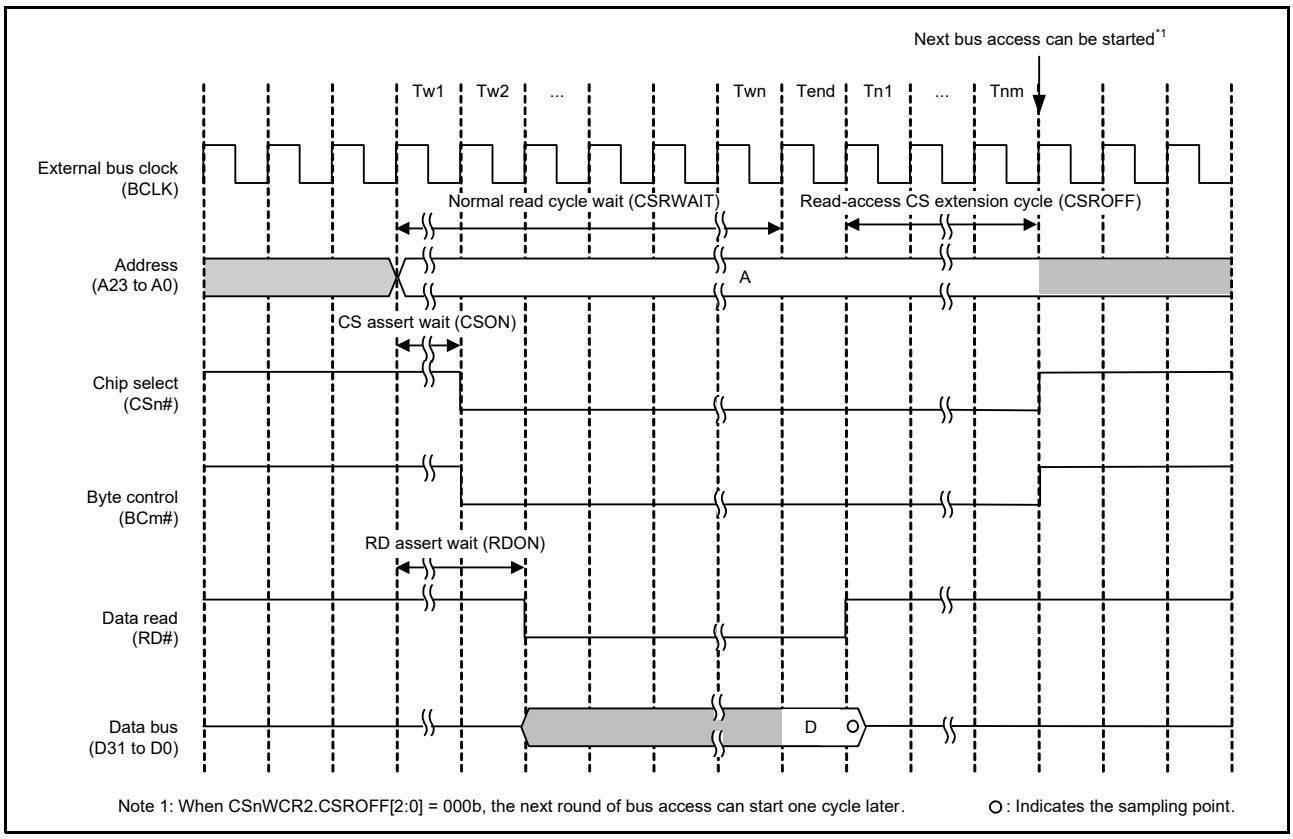


Figure 16.16 Bus Timing (Normal-Read Operation) (n = 0 to 7, m = 0 to 3)

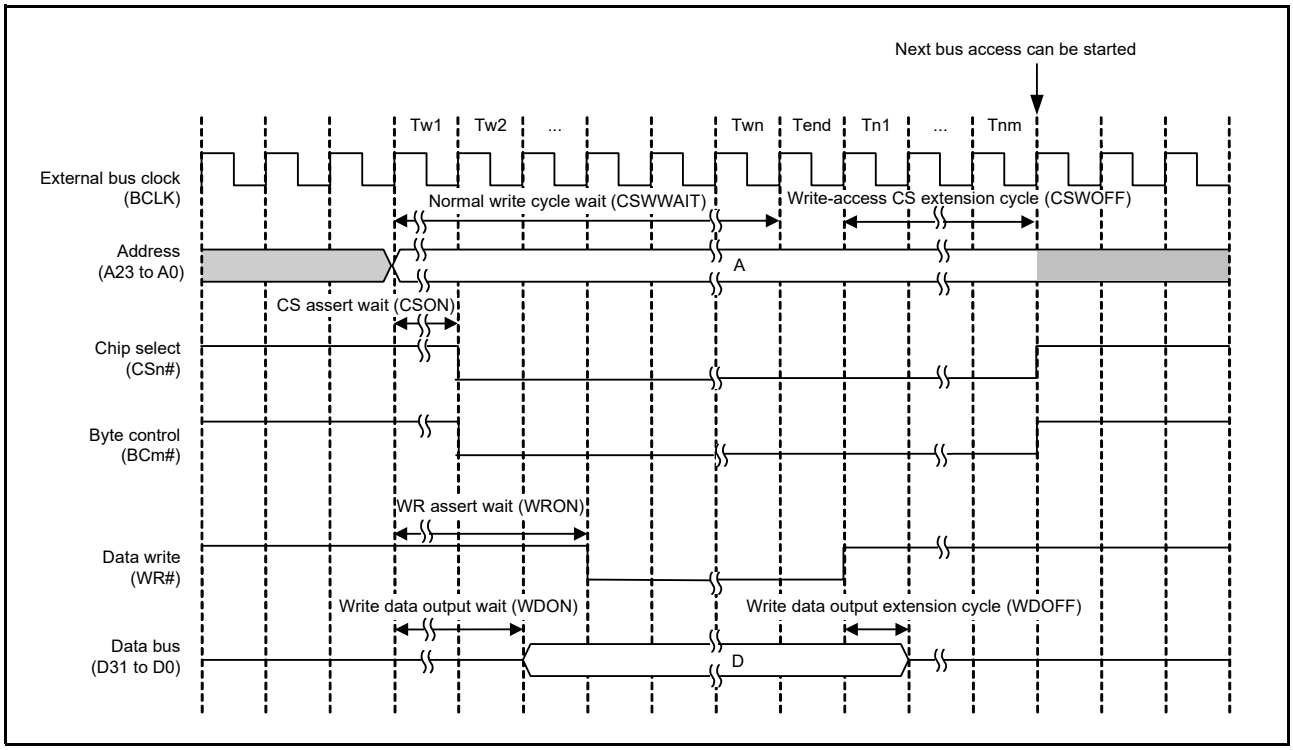


Figure 16.17 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

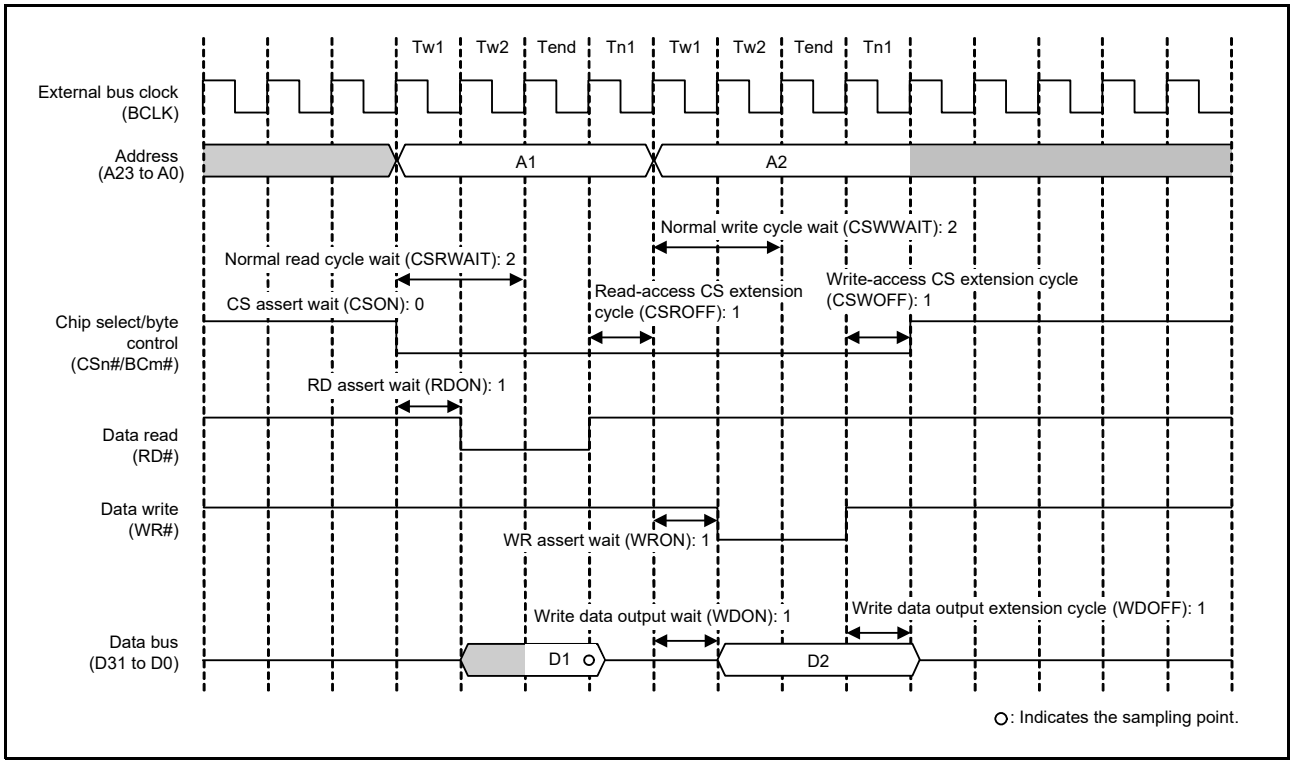


Figure 16.18 Example of Normal Access Operation (Read/Write) (n = 0 to 7, m = 0 to 3)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations (steps (a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle) to (d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)) are repeated. Figure 16.19 and Figure 16.20 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. If the recovery cycle insertion condition is satisfied, recovery cycles (step (f) Tr1 to Trn (Recovery Cycles)) are also inserted in the second and subsequent external bus accesses (refer to Figure 16.43). The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

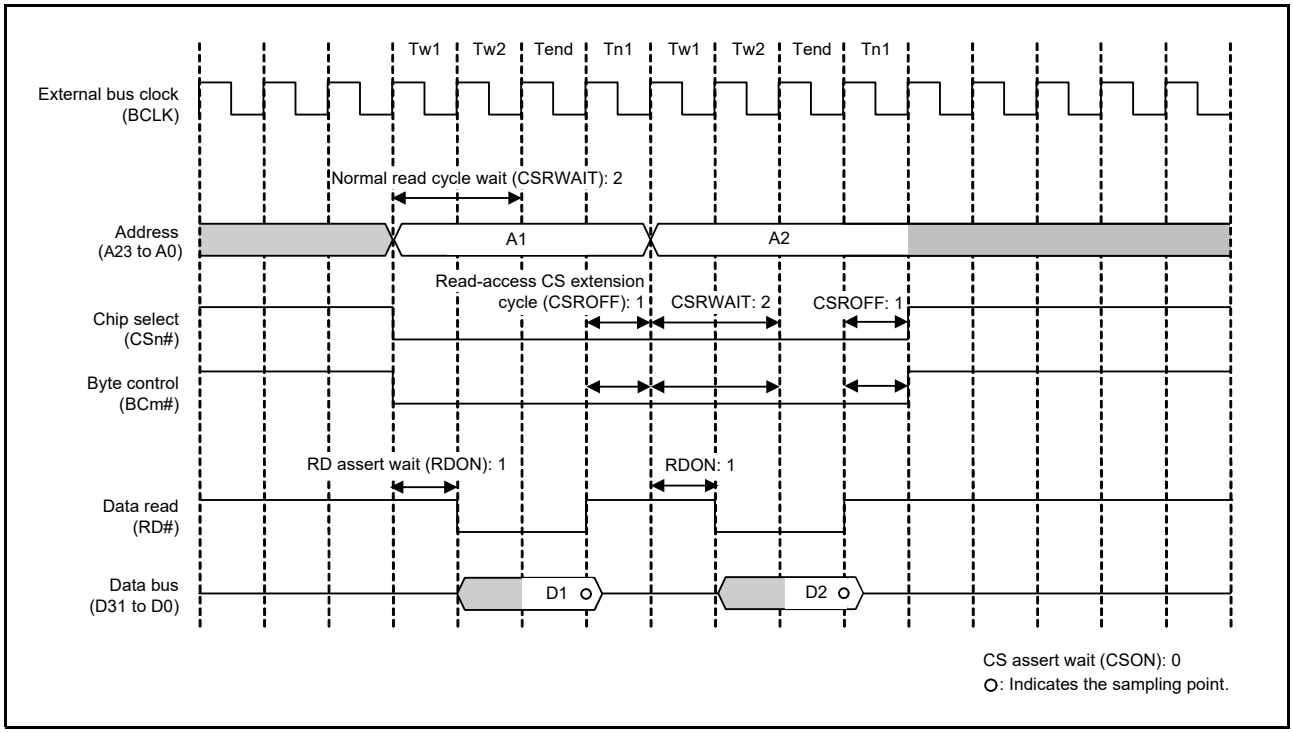


Figure 16.19 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

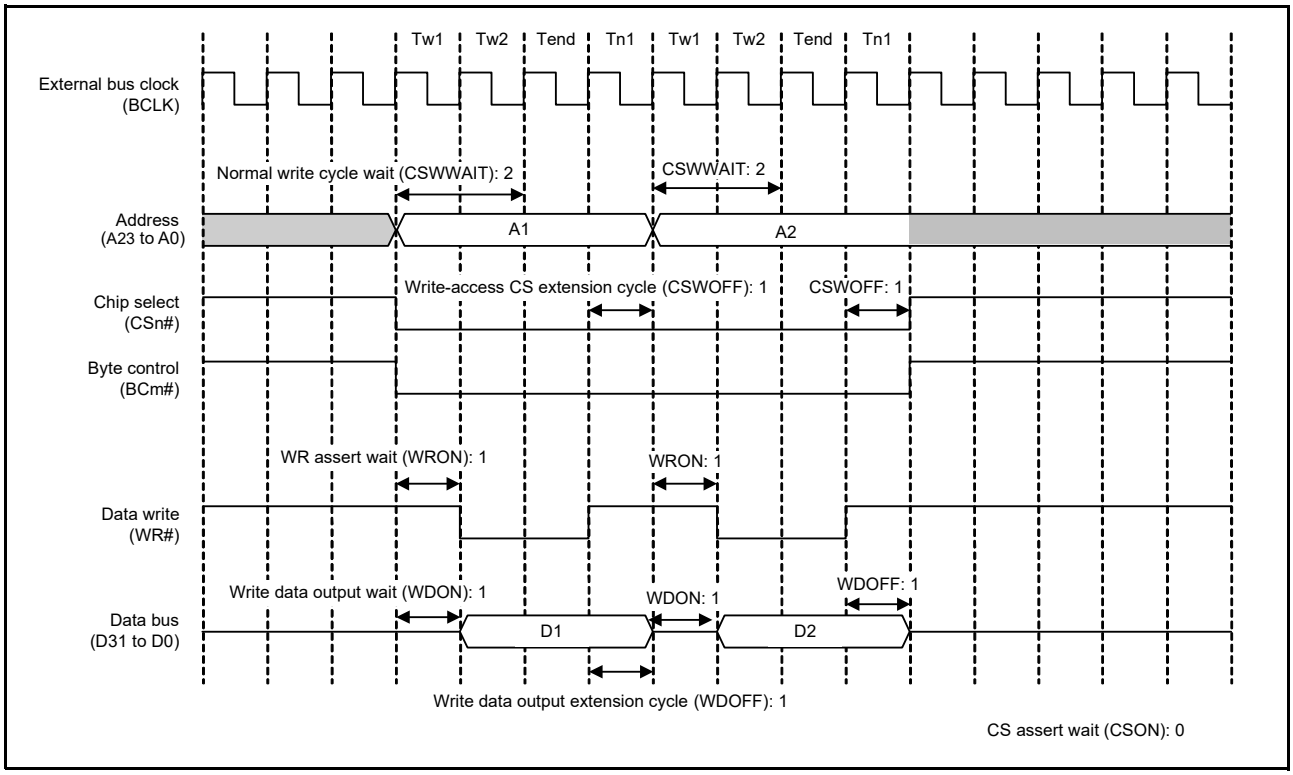


Figure 16.20 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

Figure 16.21 and Figure 16.22 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

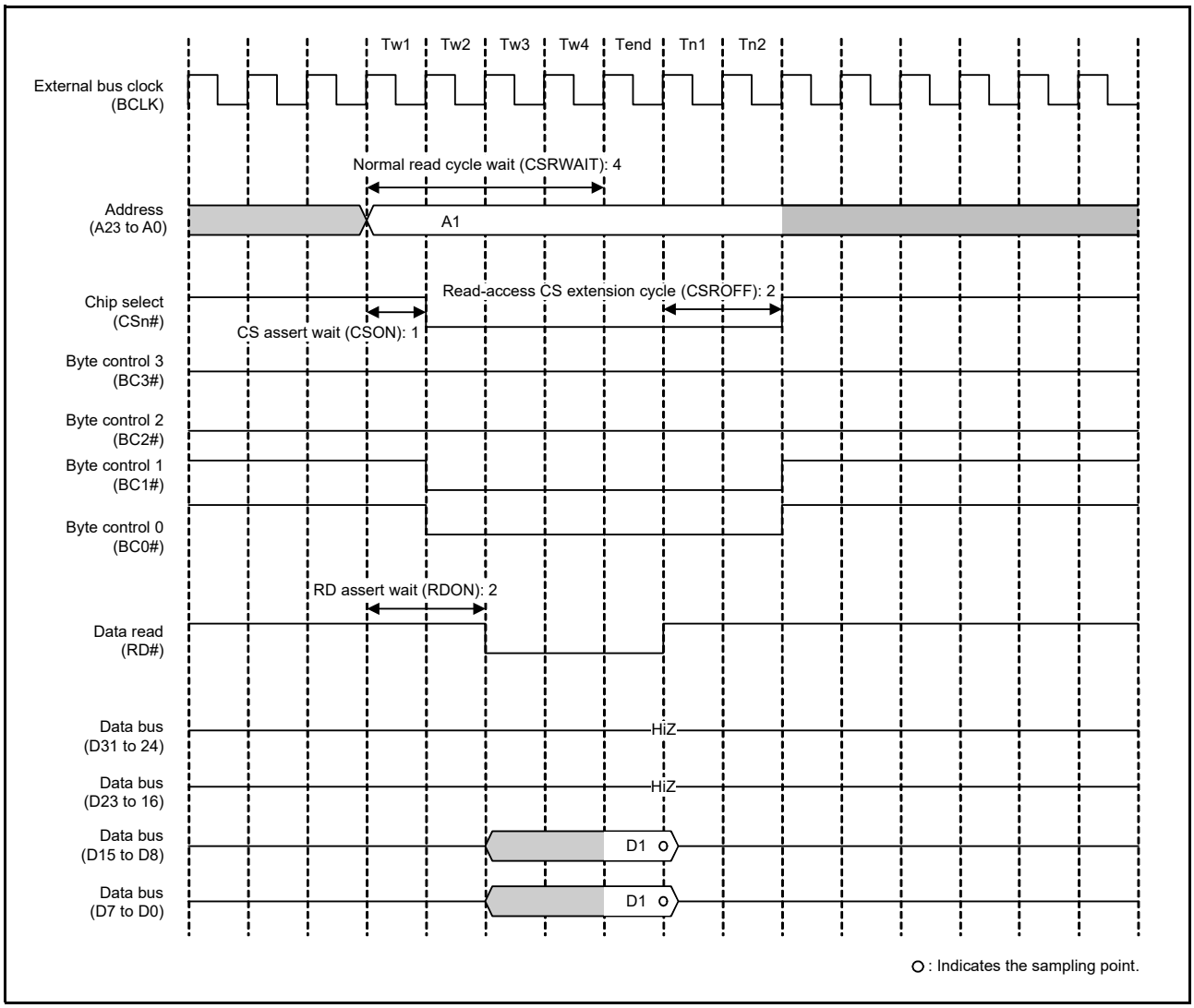


Figure 16.21 Example of Normal-Read Operation (when 32-Bit Bus Space is Accessed in 16 Bits) (n = 0 to 7)

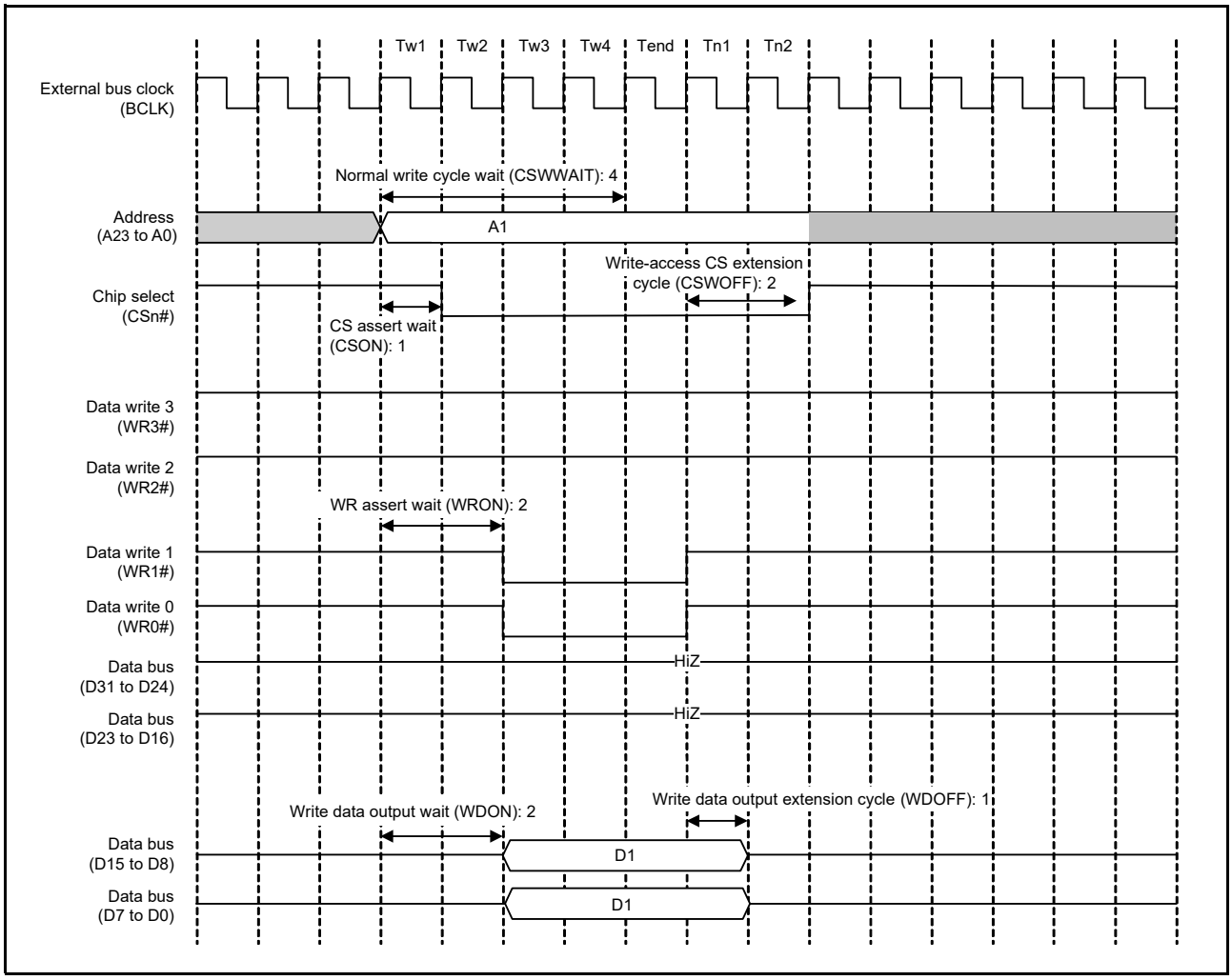


Figure 16.22 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode) (n = 0 to 7)

Figure 16.23 to Figure 16.27 show examples of normal accesses made with the 1/2 BCLK selected with the EBCLK pin output select bit.

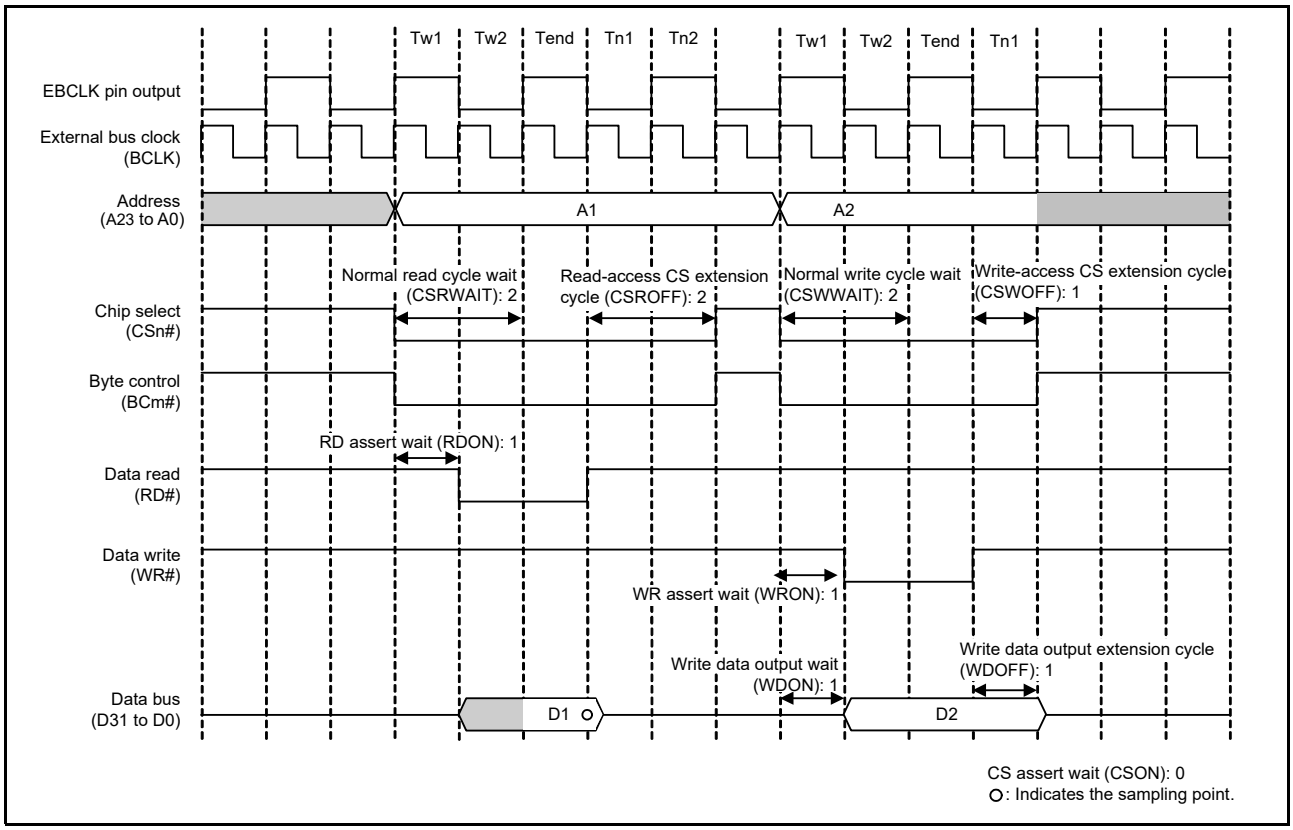


Figure 16.23 Example of Normal Access (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

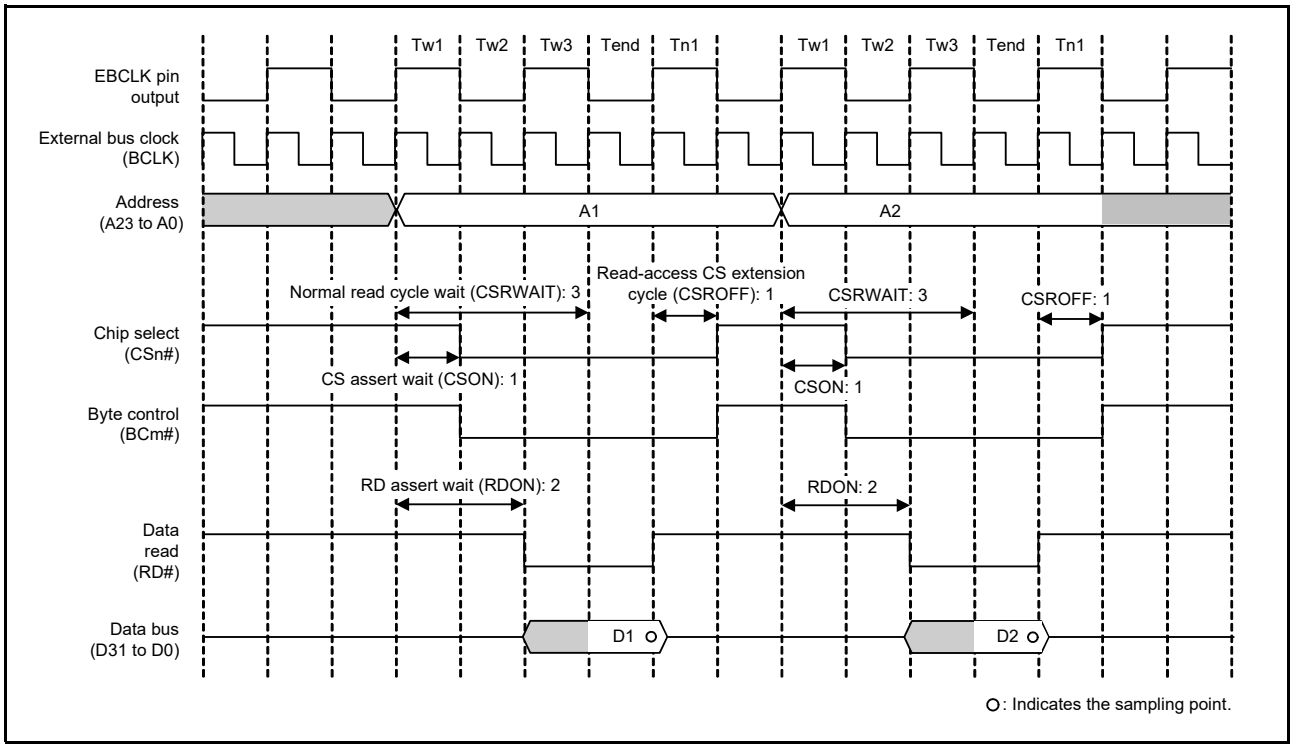


Figure 16.24 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

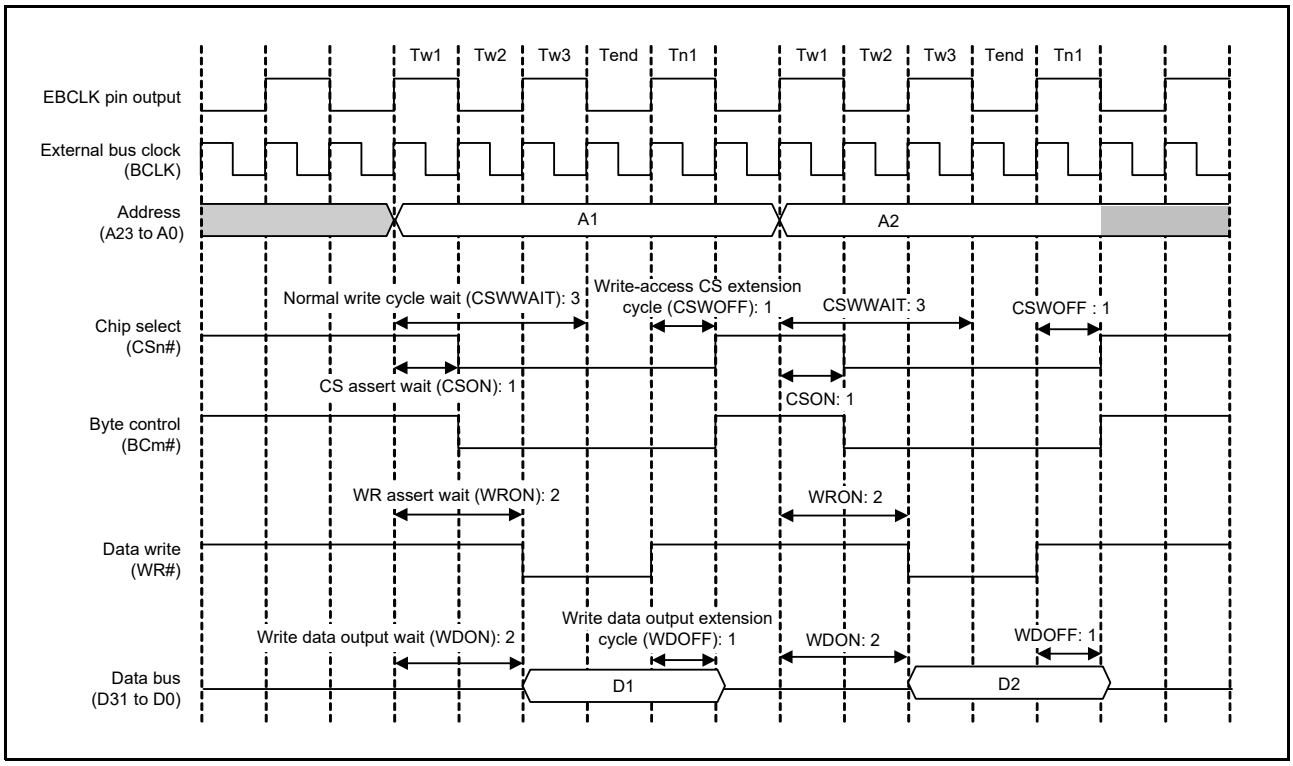


Figure 16.25 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

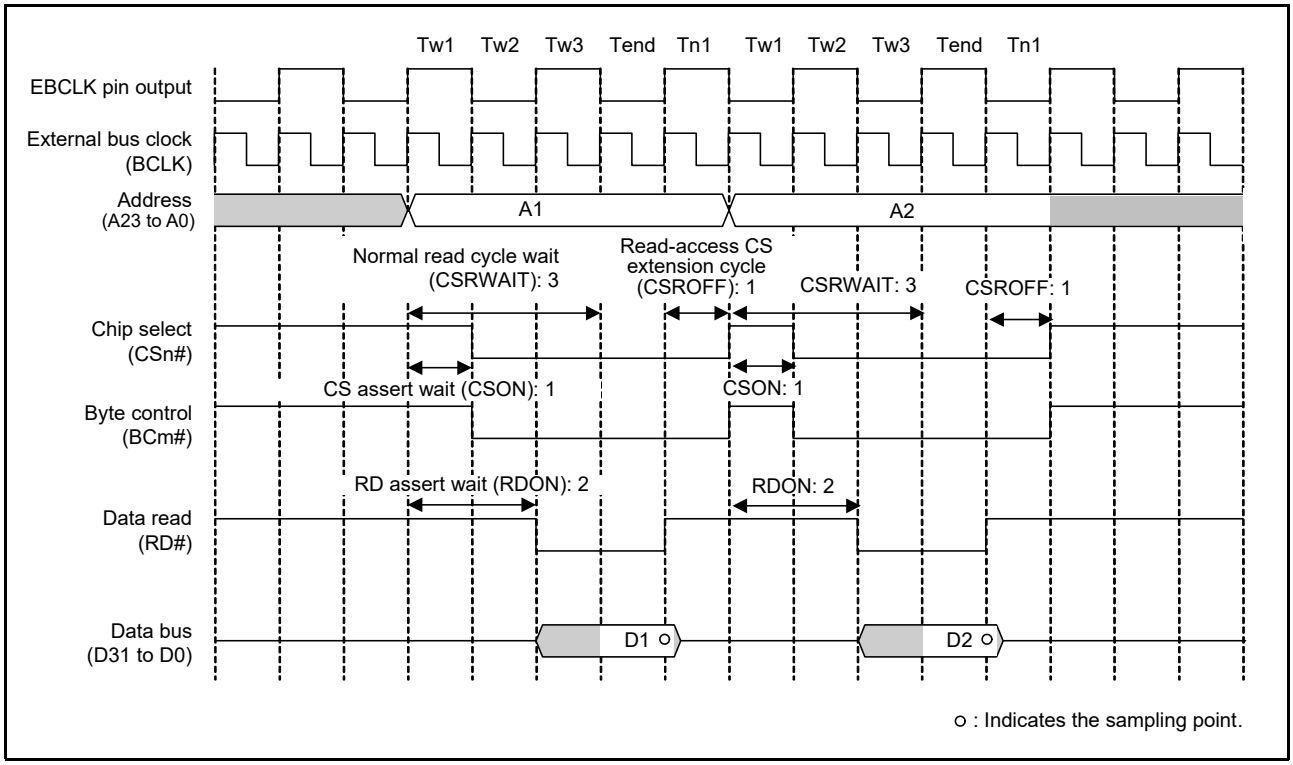


Figure 16.26 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

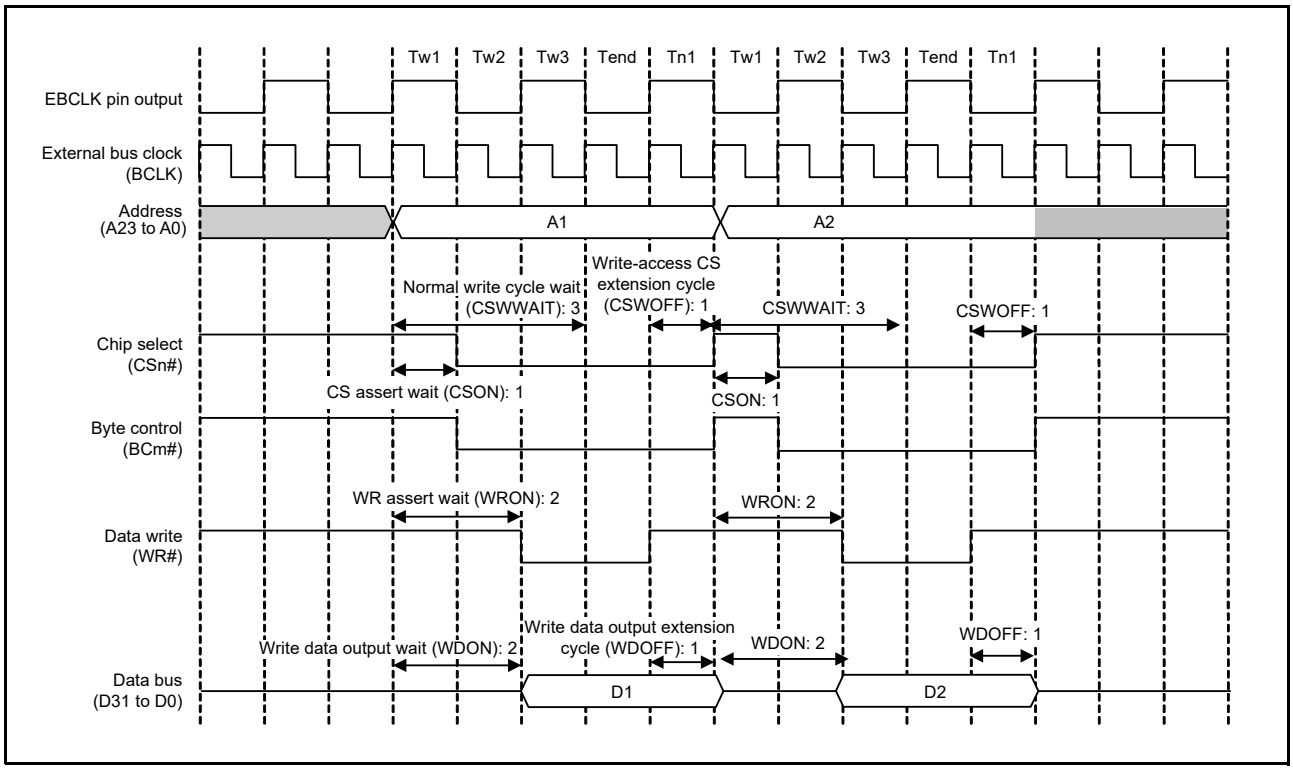


Figure 16.27 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, the bus access for page access operations becomes page reading and writing. Specifically, page access can occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when the split accesses are not aligned or the access spreads over the 32-bit boundary. Refer to Figure 16.6 to Figure 16.9 for the conditions under which page access occurs.
Figure 16.28 and Figure 16.29 show examples of page access operations.

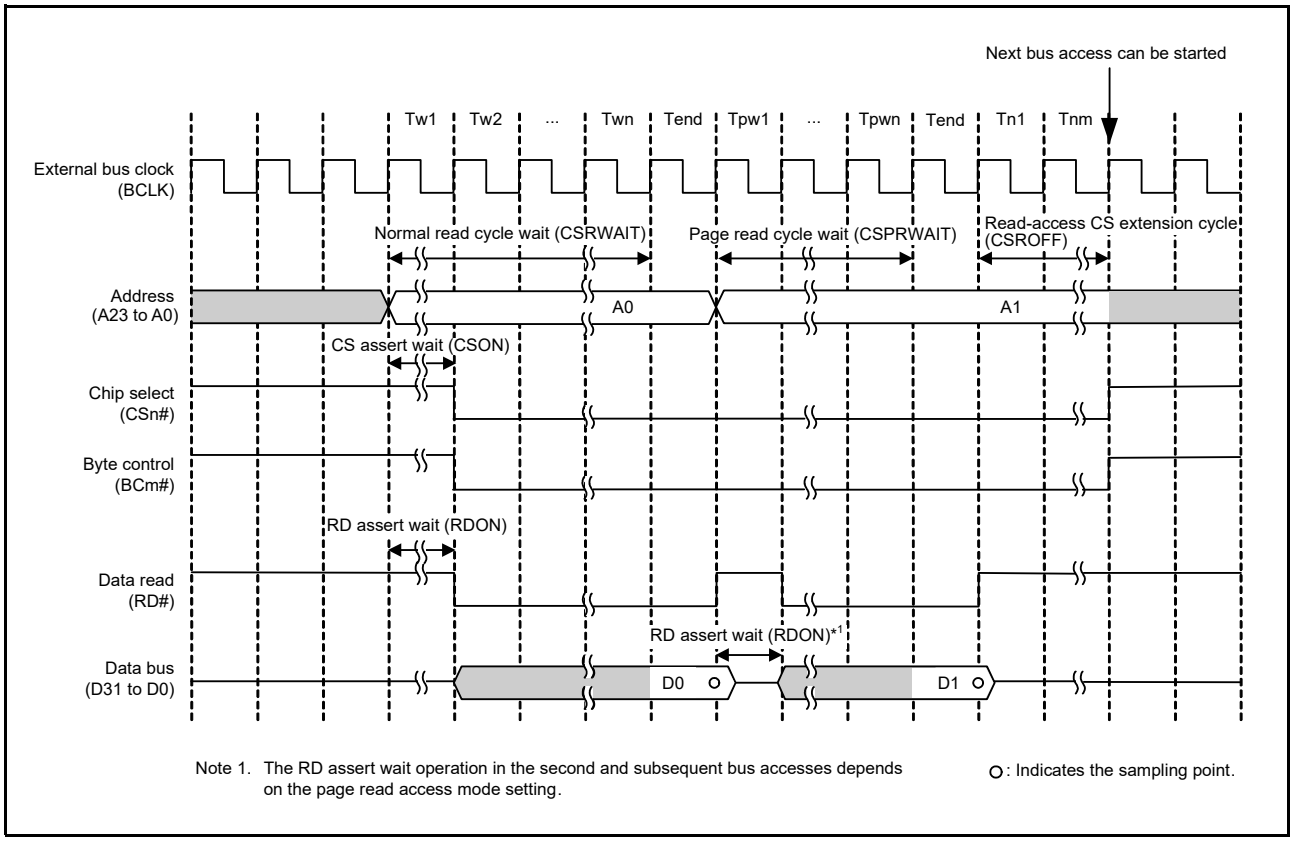


Figure 16.28 Page-Read Access Timing (n = 0 to 7, m = 0 to 3)

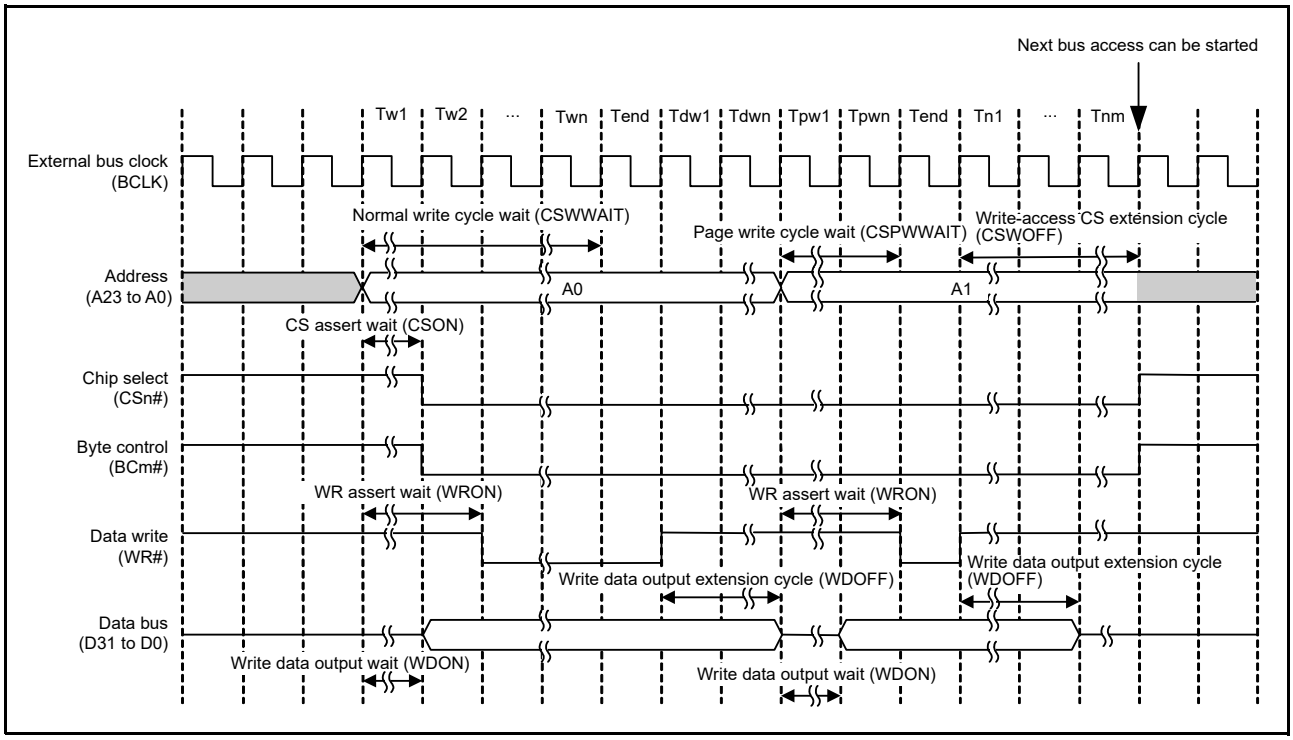


Figure 16.29 Page-Write Access Timing (n = 0 to 7, m = 0 to 3)

Figure 16.30 and Figure 16.31 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers are example settings. In practice, the register settings will correspond to the specifications of connected devices.

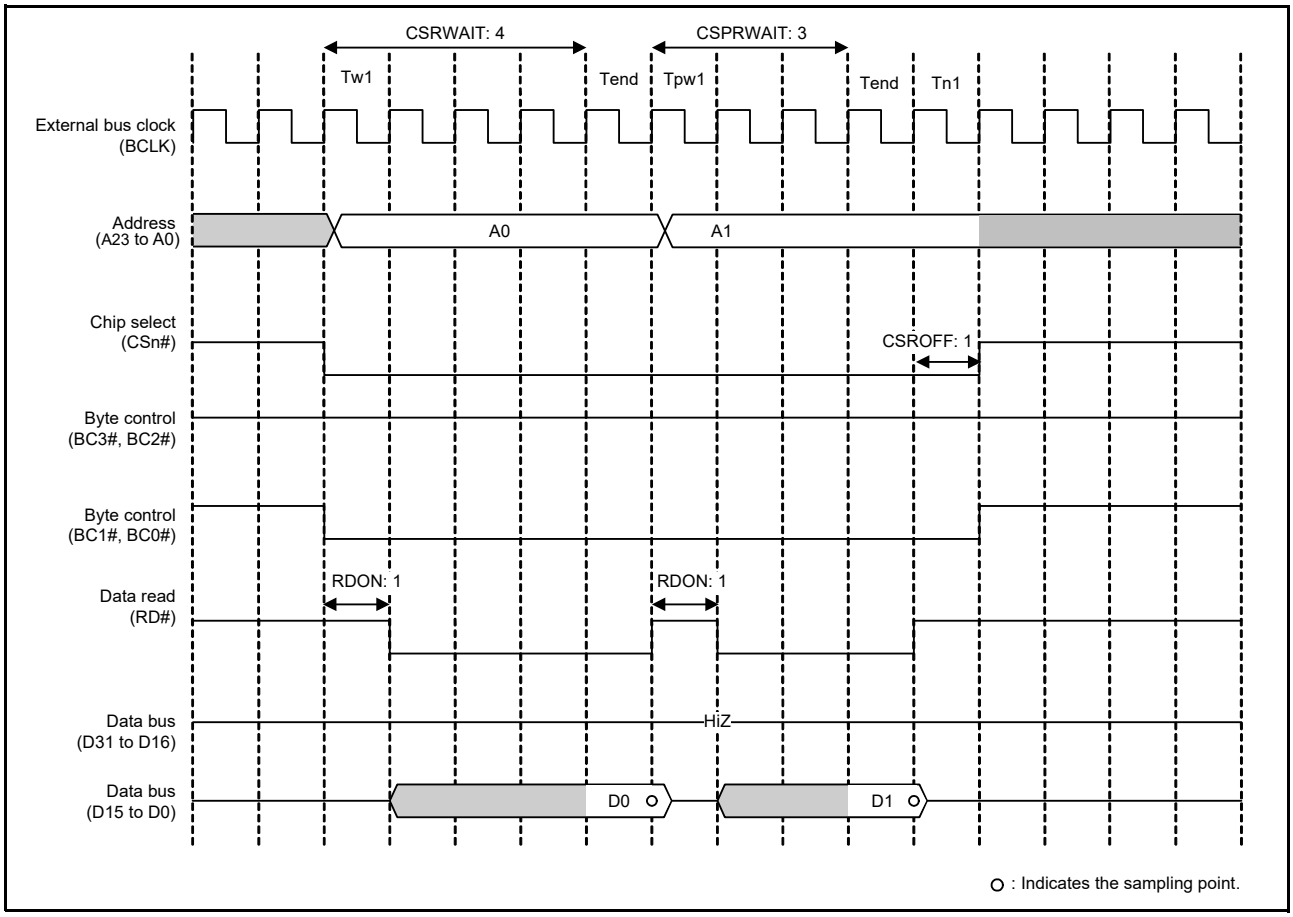


Figure 16.30 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 7)

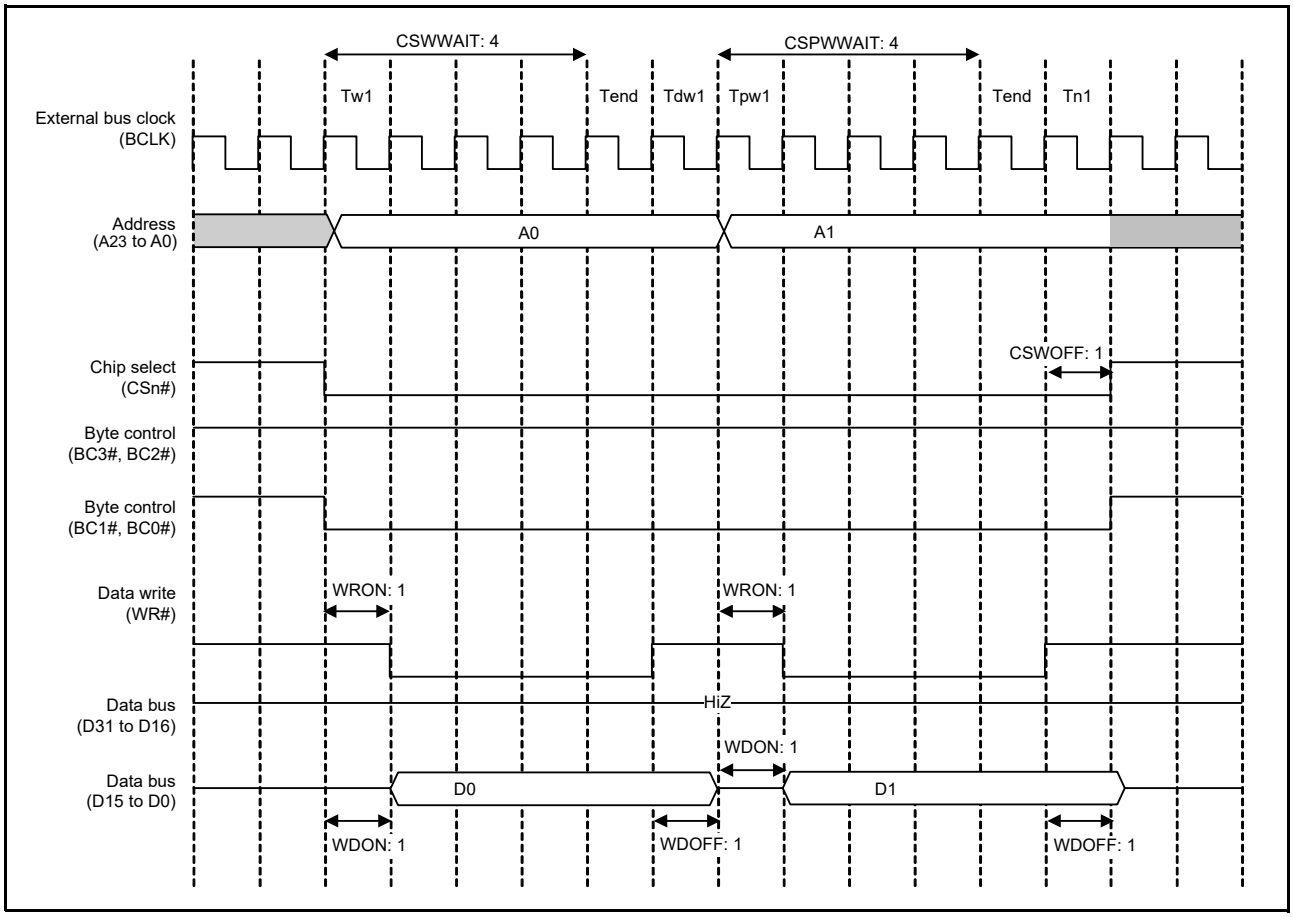


Figure 16.31 Example of Page-Write Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)

Figure 16.32 and Figure 16.33 show examples of page access operations performed with the 1/2 BCLK selected with the EBCLK pin output select bit.

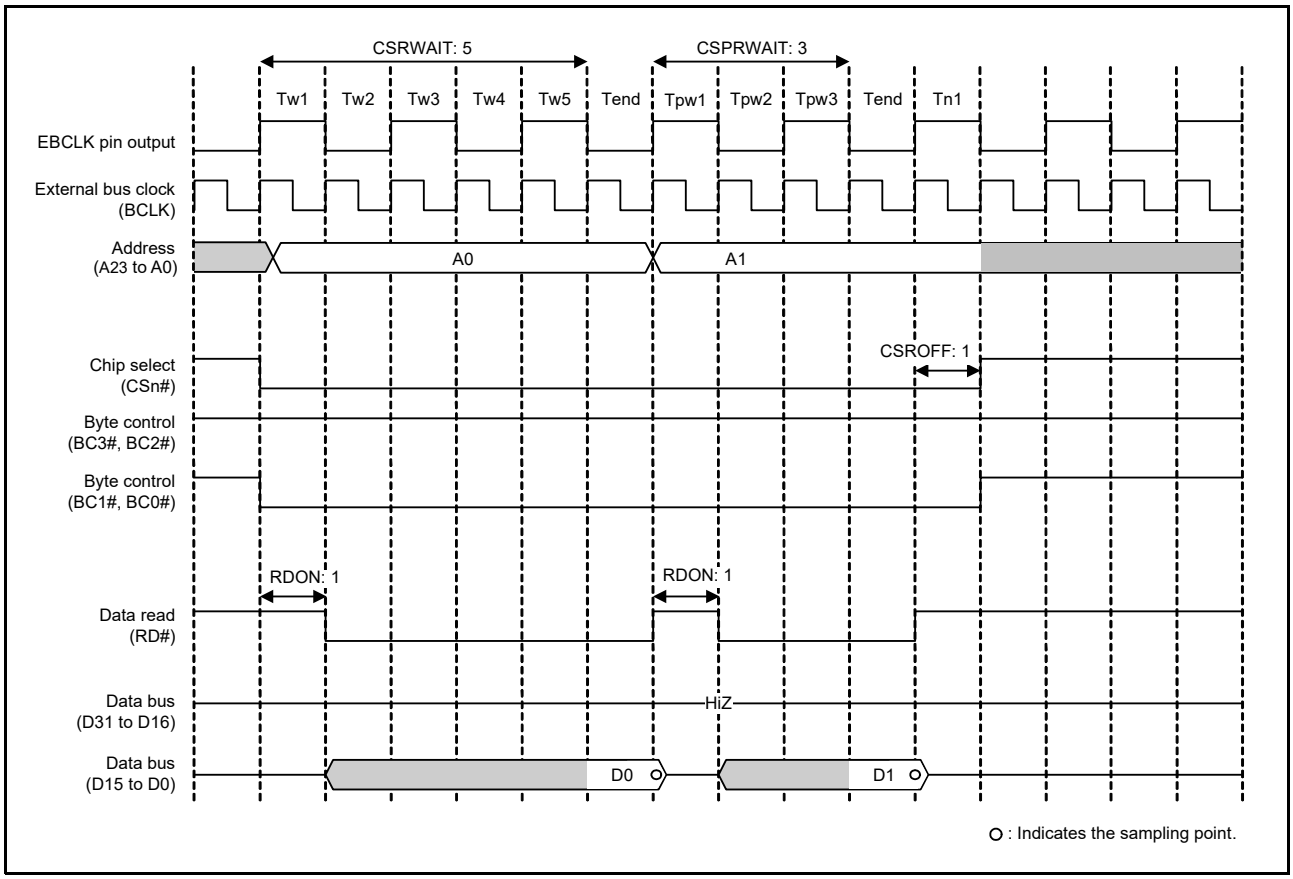


Figure 16.32 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7)

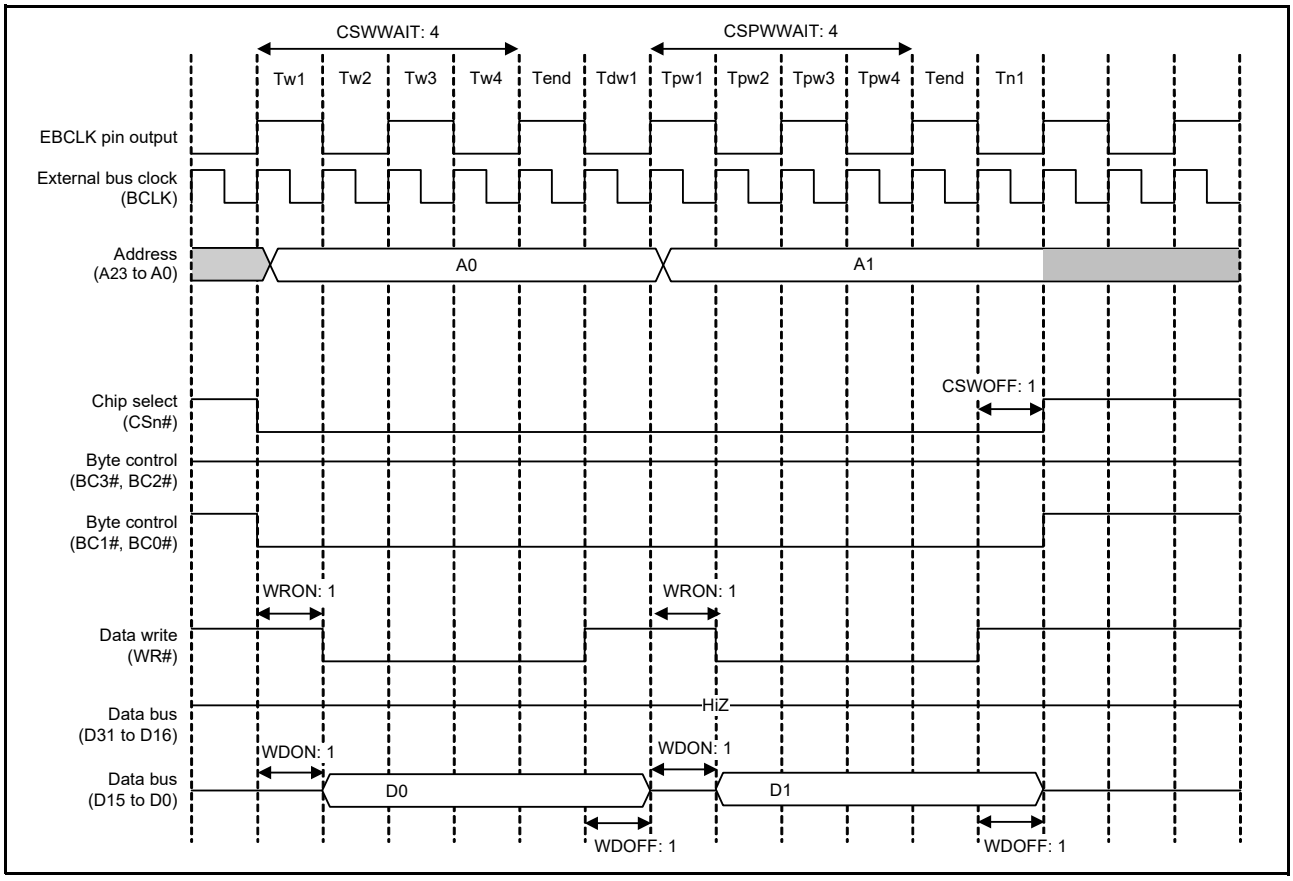


Figure 16.33 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7)

16.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D0 pins in the corresponding area. Using this function enables direct connection of this MCU to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D0 are multiplexed with A7 to A0. When 16-bit width is selected, D15 to D0 are multiplexed with A15 to A0. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD#, WRn#, and BCn# signals.
Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn# signals within the address cycle, the byte-control signal is output for the data being read or written.
During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

- Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and one cycle before the address latch (ALE) signal is negated. The number of cycles is selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS# assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.
Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 16.34 to Figure 16.36 show examples of operations with the address/data multiplexed I/O interface.

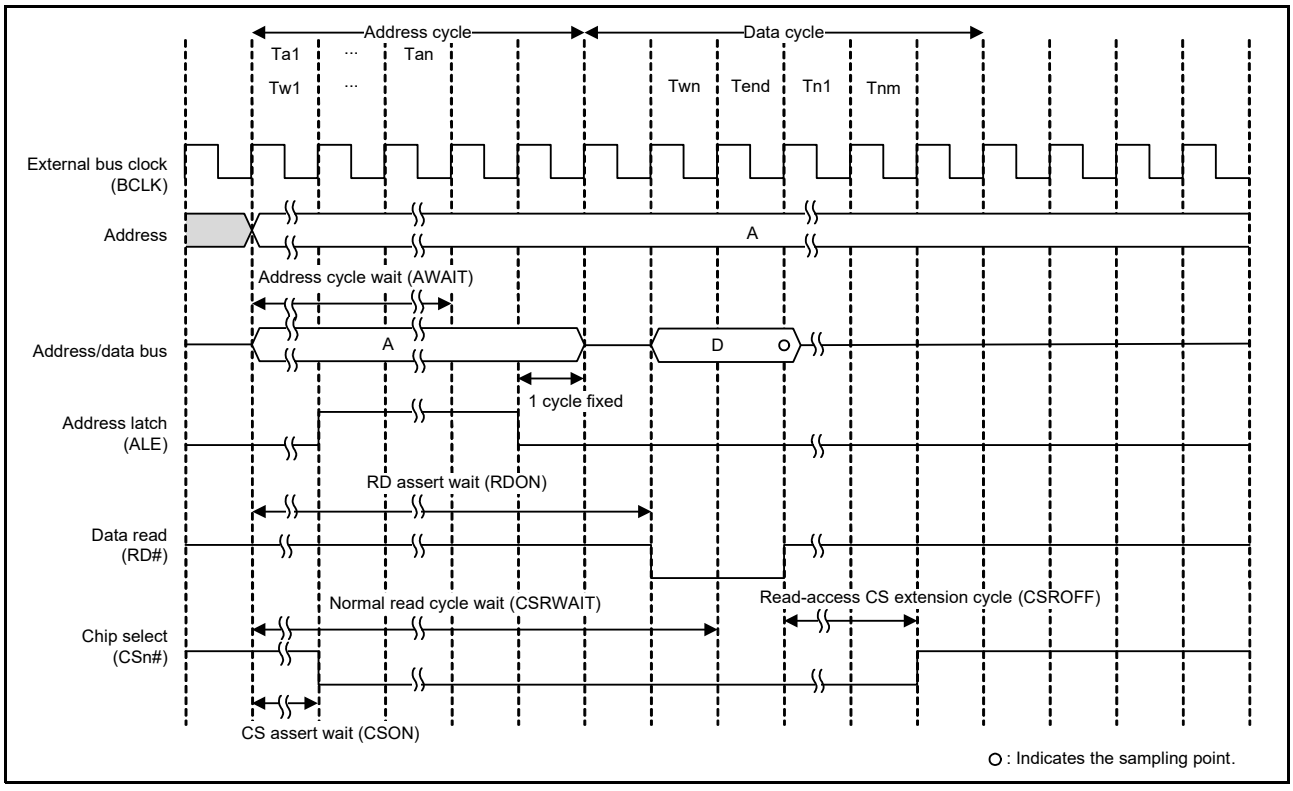


Figure 16.34 Example of Read Access Operation with Address/Data Multiplexed I/O Interface (n = 0 to 7)

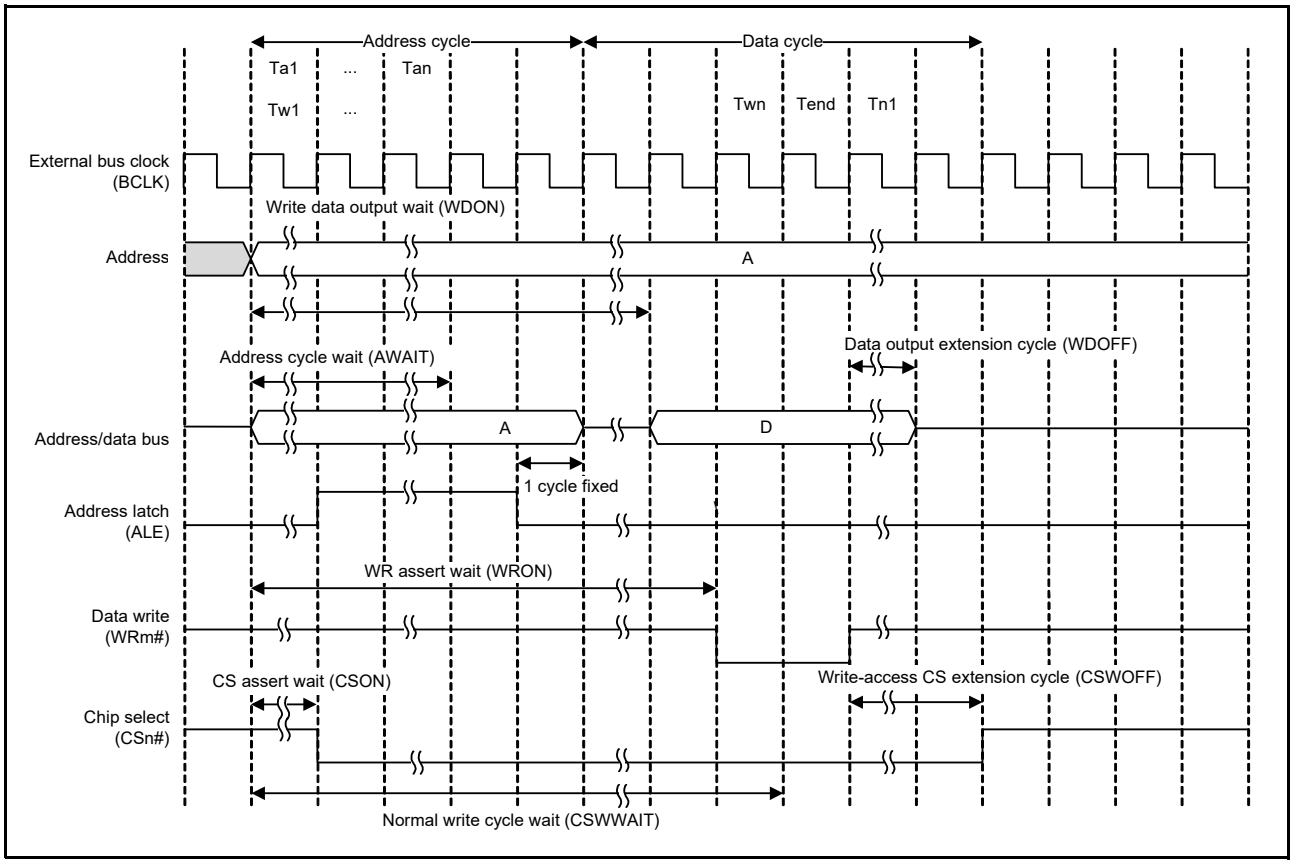


Figure 16.35 Example of Write Access Operation with Address/Data Multiplexed I/O Interface (n = 0 to 7; m = 0, 1)

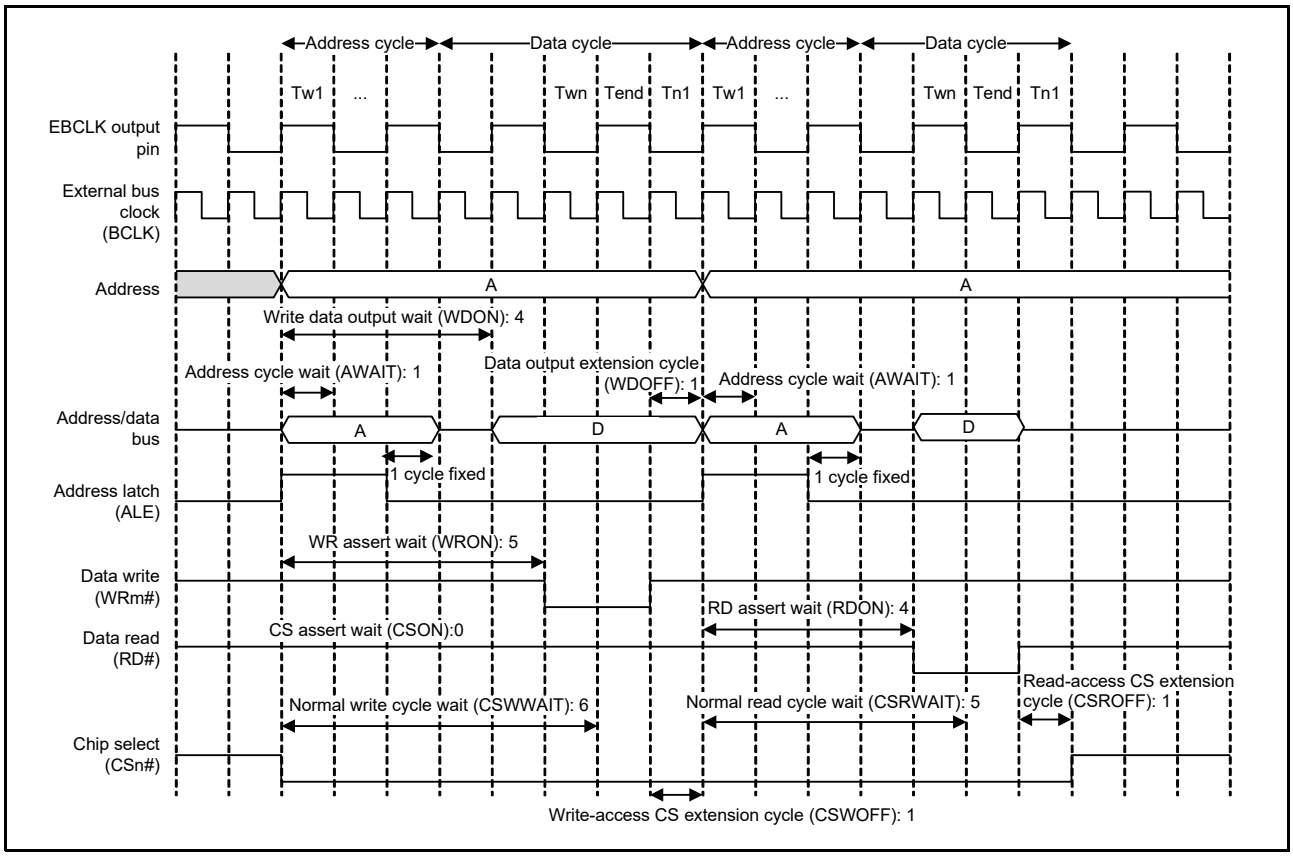


Figure 16.36 Example of Bus Timing with Address/Data Multiplexed I/O Interface (n = 0 to 7; m = 0, 1)

16.5.3 External Wait Function

Wait cycles can be extended by the WAIT# signal over the length of normal access cycle wait (specified by the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1) and page access cycle wait (specified by the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1).
When external wait is enabled (the EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT# signal is held low. When external wait is disabled (the EWENB bit = 0 in CSnMOD), the WAIT# signal has no effect.
All wait cycles specified in CSnWCR1 are inserted independently of the WAIT# signal.

(1) Normal Access

Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.
With respect to the second and subsequent accesses, sampling of the WAIT# signal begins upon completion of the wait cycle of the page access (Tend). The wait cycle of the page access is extended while the WAIT# signal is held low, and ends (Tend) at the next cycle after the WAIT# signal becomes high.
Figure 16.37 to Figure 16.40 show examples of external wait insertion timing with the separate bus interface.

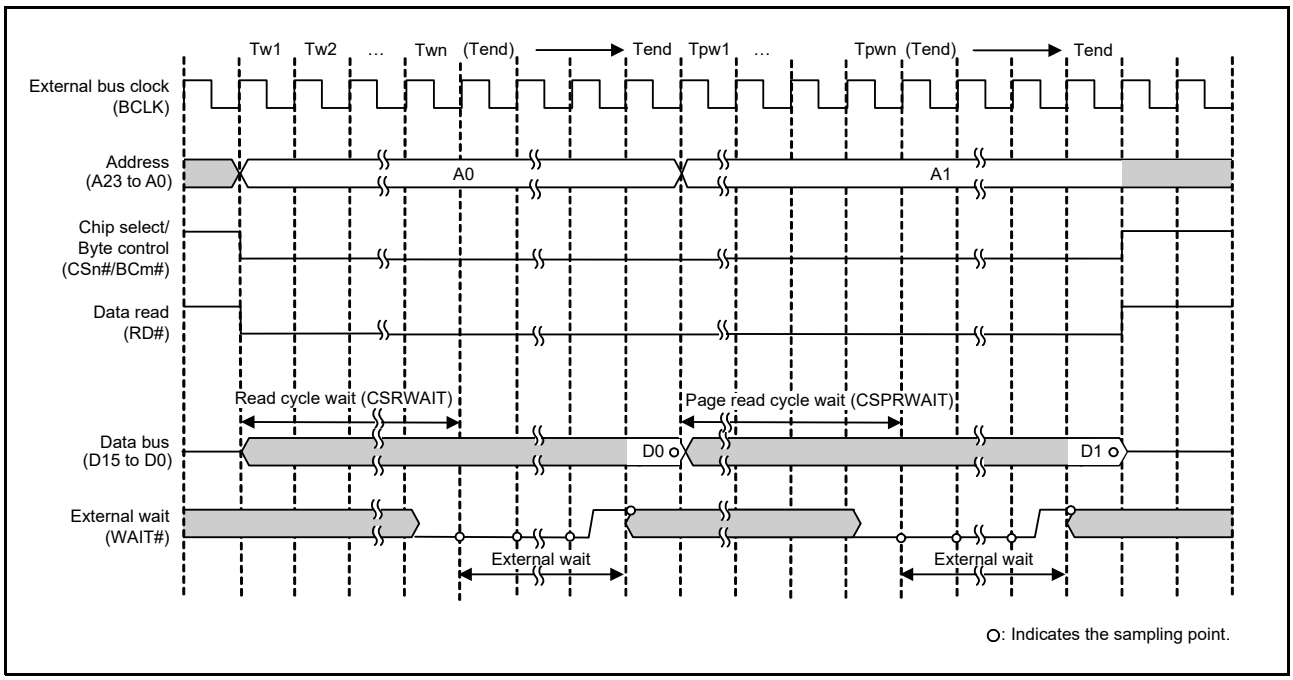


Figure 16.37 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space, when 1/1 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7; m = 0, 1)

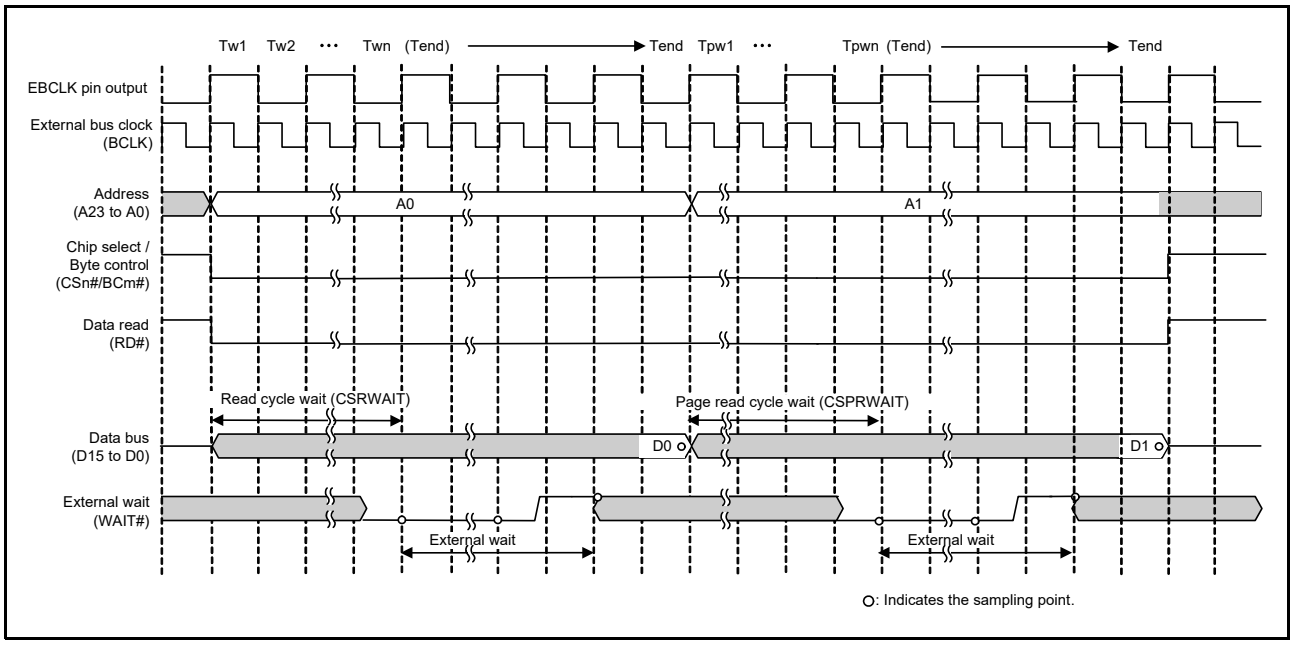


Figure 16.38 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space, when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7; m = 0, 1)

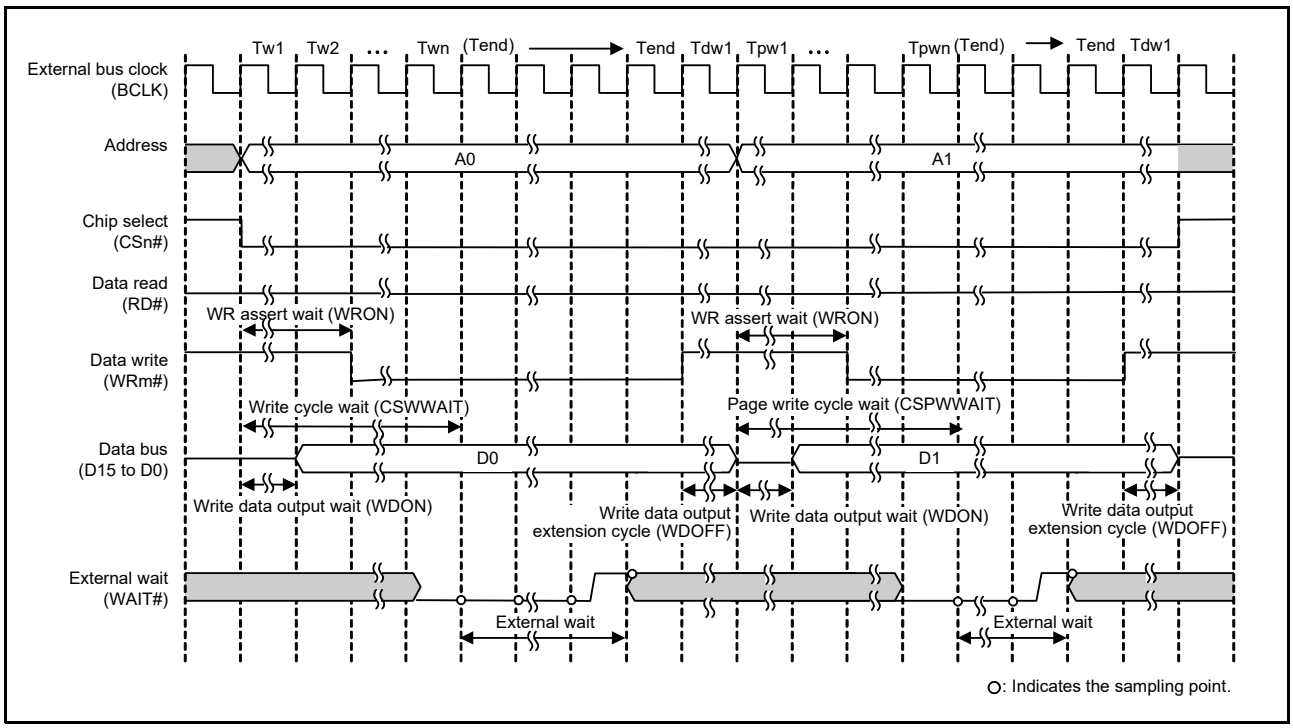


Figure 16.39 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space in Byte Strobe Mode, when 1/1 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7; m = 0, 1)

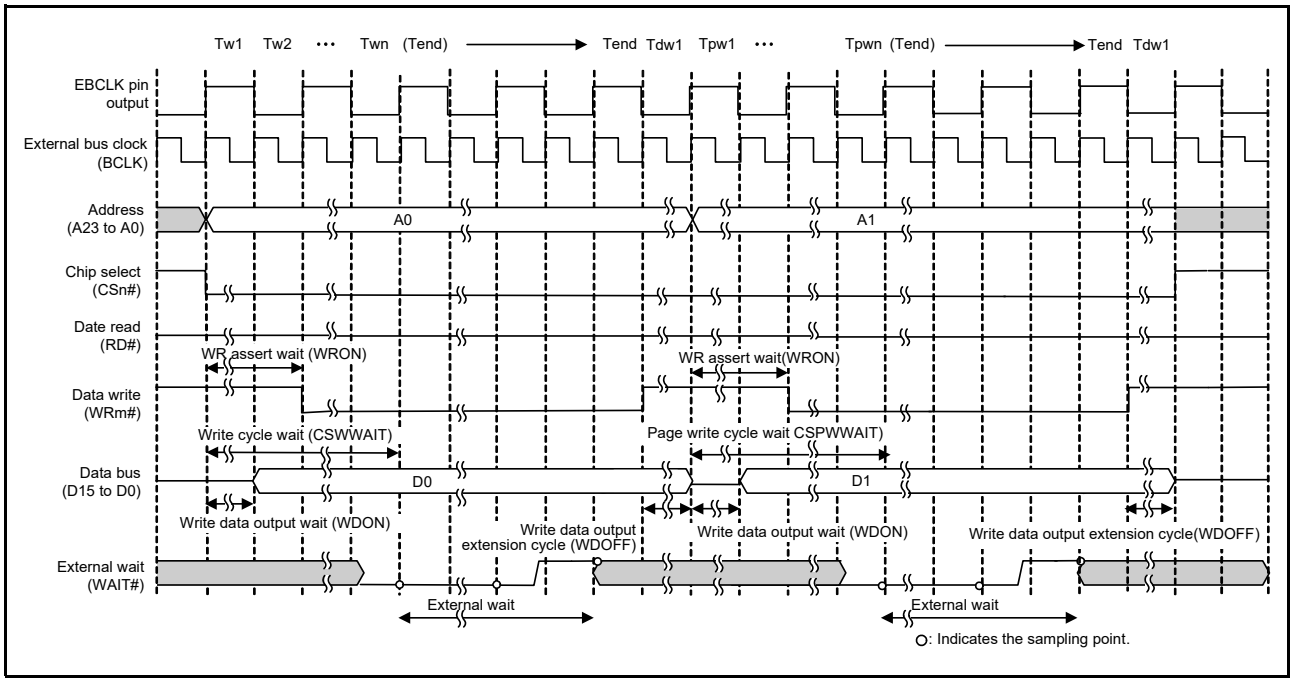


Figure 16.40 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space in Byte Strobe Mode, when 1/2 BCLK is Selected with the EBCLK Pin Output Select Bit) (n = 0 to 7; m = 0, 1)

(3) Address/Data Multiplexed I/O Interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT# pin can be inserted in the same way as that with the separate bus interface.
Address cycles are not affected by the wait control settings. Figure 16.41 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

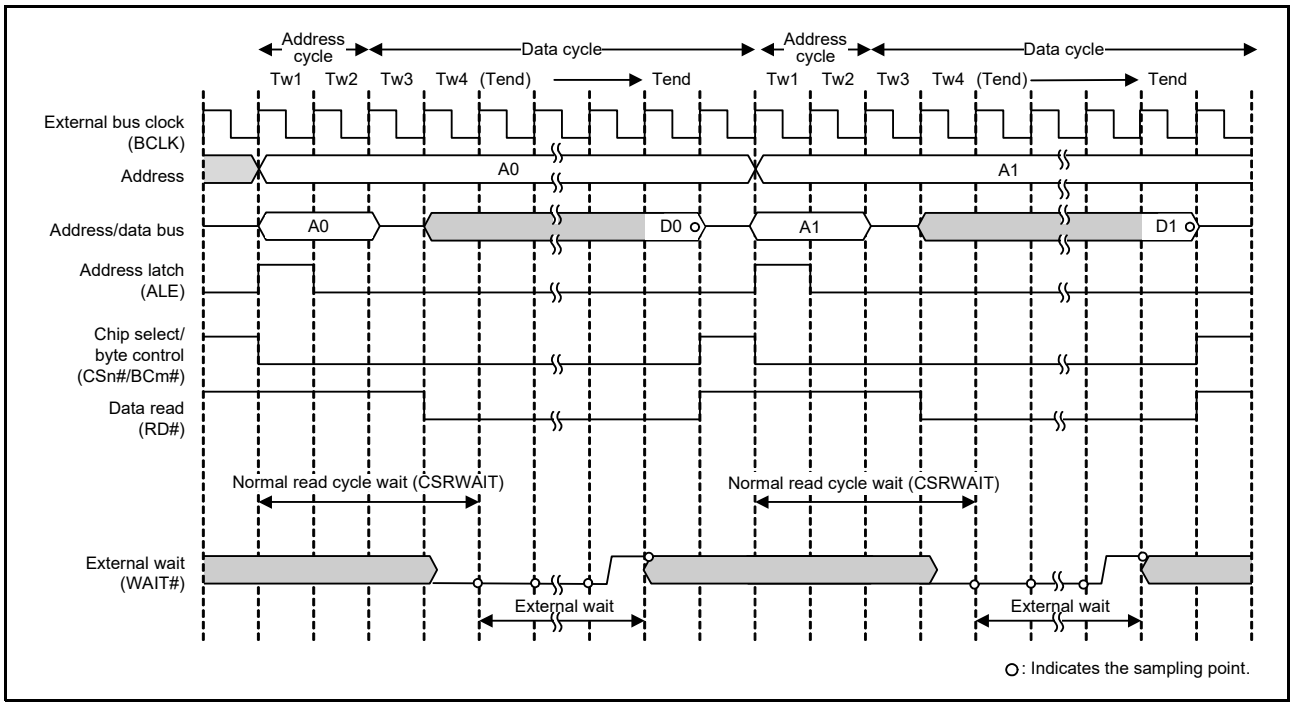


Figure 16.41 Example of External Wait Insertion Timing with Address/Data Multiplexed I/O Interface (n = 0 to 7; m = 0, 1)

16.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the recovery cycle insertion enable bit in CSRECEN to 1.
The number of recovery cycles to be inserted after read cycles and write cycles can be separately set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles should be set with the WRCV[3:0] bits for the area. When the preceding bus cycle is a read access, the number of read recovery cycles should be set with the RRCV[3:0] bits for the area. For example, when CS1 read access occurs after CS0 read access, the number of recovery cycles to be inserted between them is set by the RRCV[3:0] bits in CS0REC.
Recovery cycles can be inserted on any of the following eight conditions. The recovery cycle insertion can be enabled or disabled with the RCVENi (i = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, i.e. when the CSn# signal (n = 0 to 7) is negated. A high-level period of the CSn# signal is inserted for the specified recovery cycle period starting from this point.
The CSn# signal for the next round of bus access is asserted immediately after the end of recovery cycles in the fastest case. Even if the next request for access to an external address space is generated during the recovery period, the next round of access over the external bus will start immediately after the end of recovery cycles.
When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer even if the recovery cycle insertion condition is satisfied (see Figure 16.44).
Similarly, during normal accesses with page access enabled, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.
With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 16.42 to Figure 16.44 show examples of recovery cycle insertion with the separate bus interface.

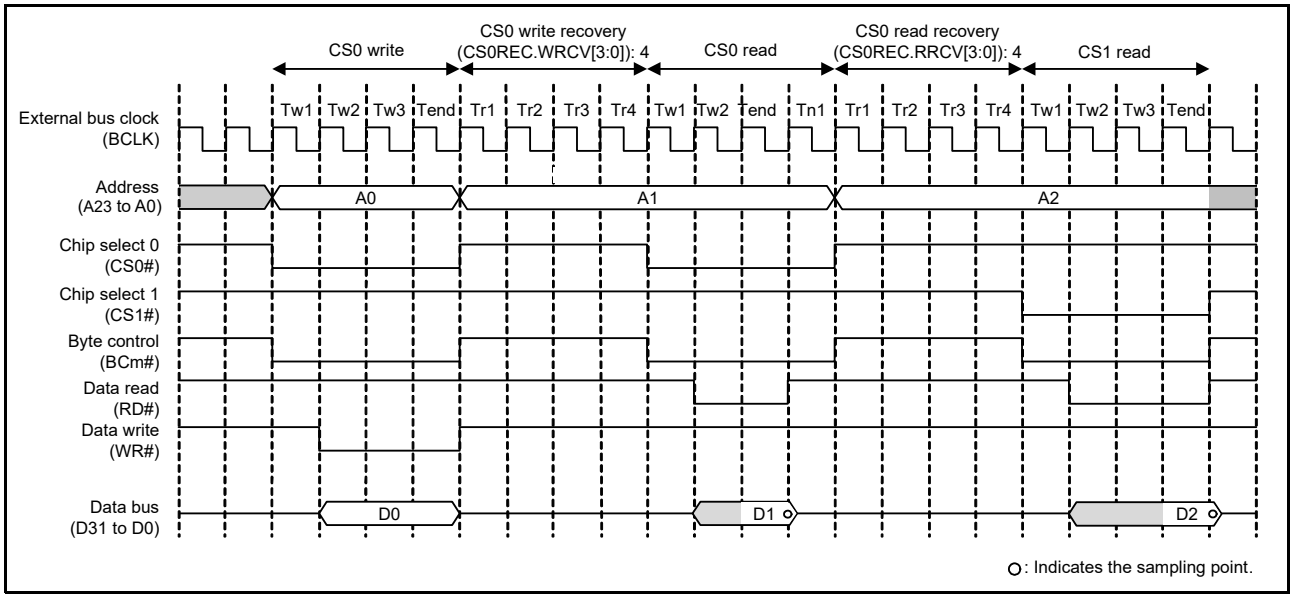


Figure 16.42 Example of Recovery Cycle Insertion with Separate Bus Interface (m = 0 to 3)

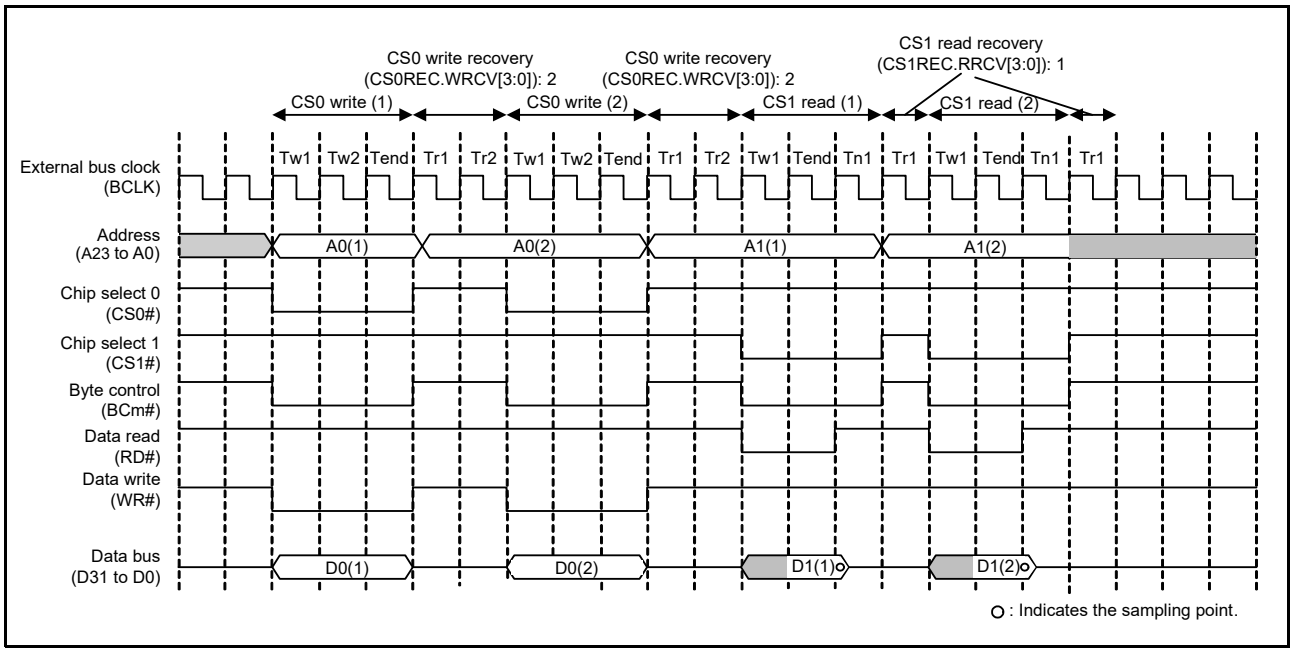


Figure 16.43 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0 to 3)

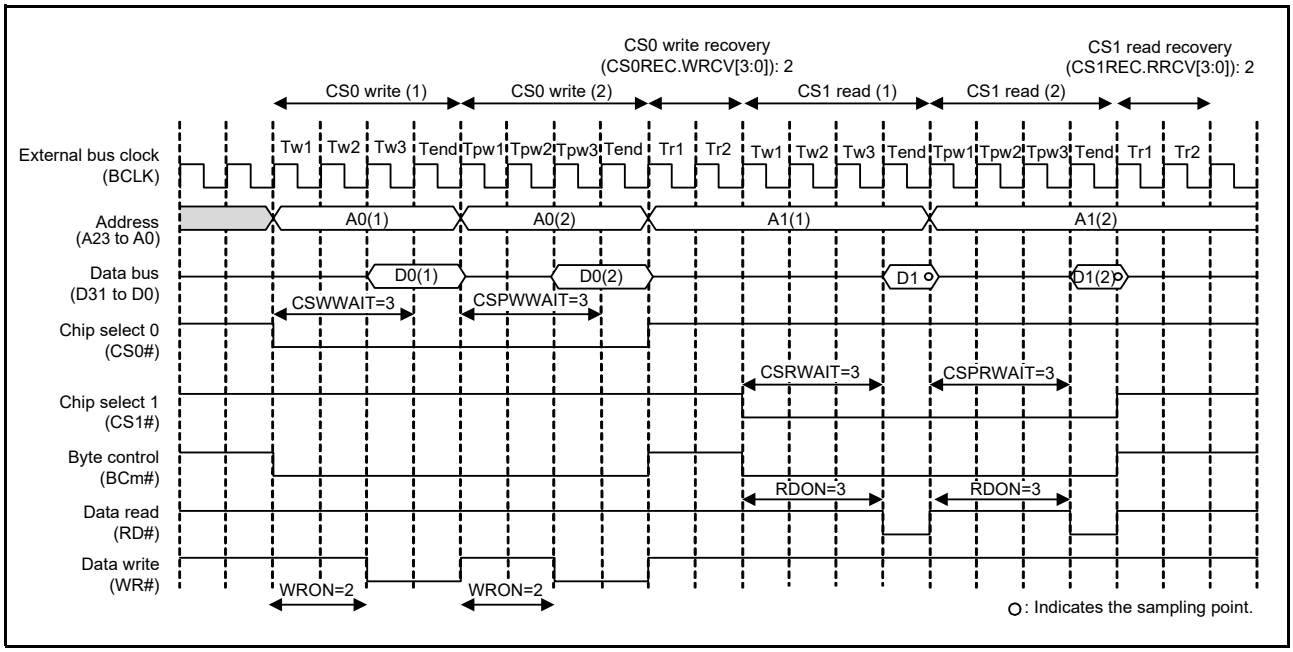


Figure 16.44 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0 to 3)

Figure 16.45 shows examples of operations when the EBCLK pin output selection bits are set for frequency-division of BCLK by 2.

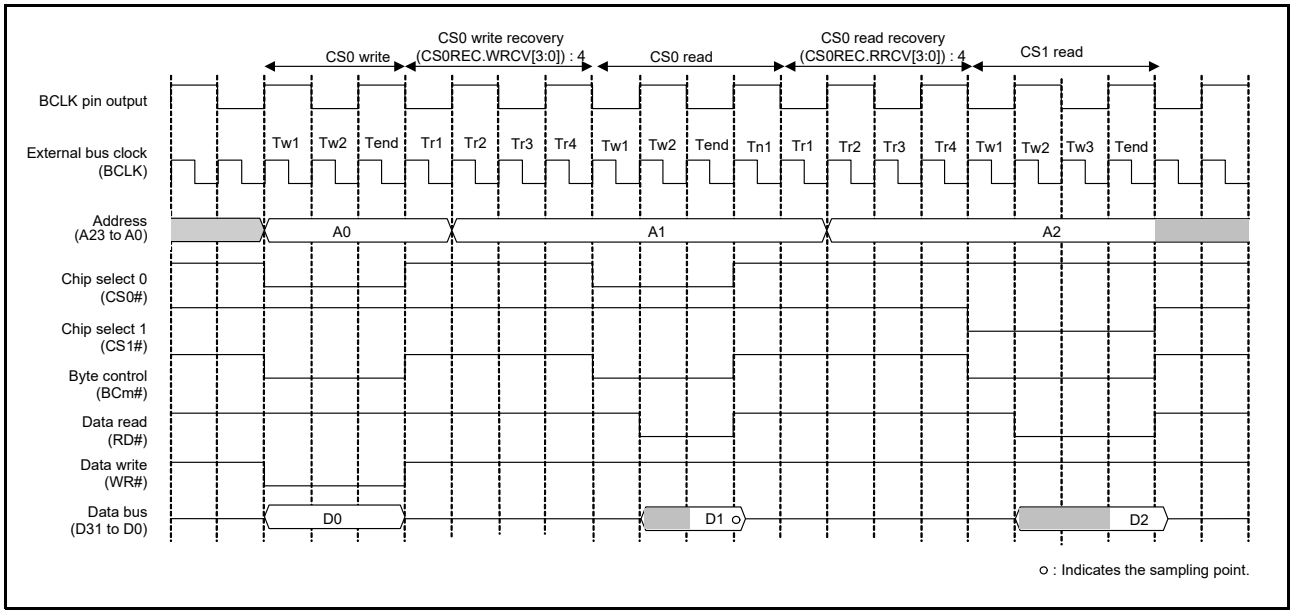


Figure 16.45 Example of Operation for Recovery Cycles when the EBCLK Pin Output Selection Bits are Set for Frequency-Division of BCLK by 2 (for the Case of Normal Access through a Separate Bus Interface) (m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 16.46 and Figure 16.47 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

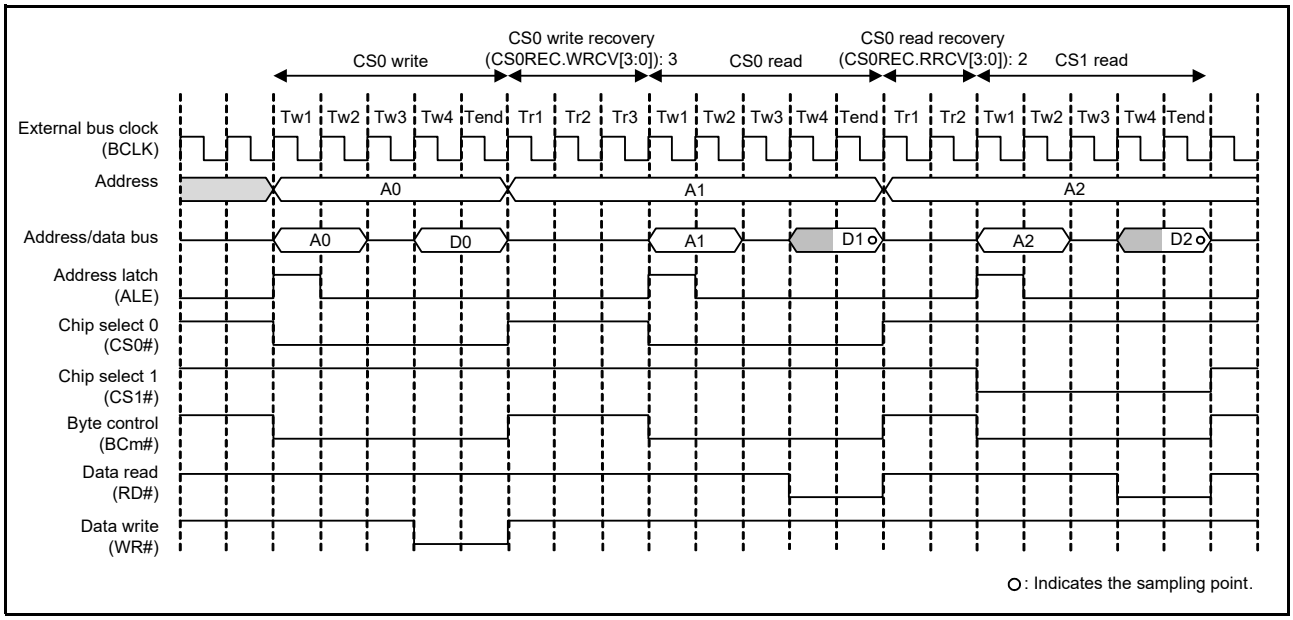


Figure 16.46 Example of Recovery Cycle Insertion with Address/Data Multiplexed I/O Interface (m = 0, 1)

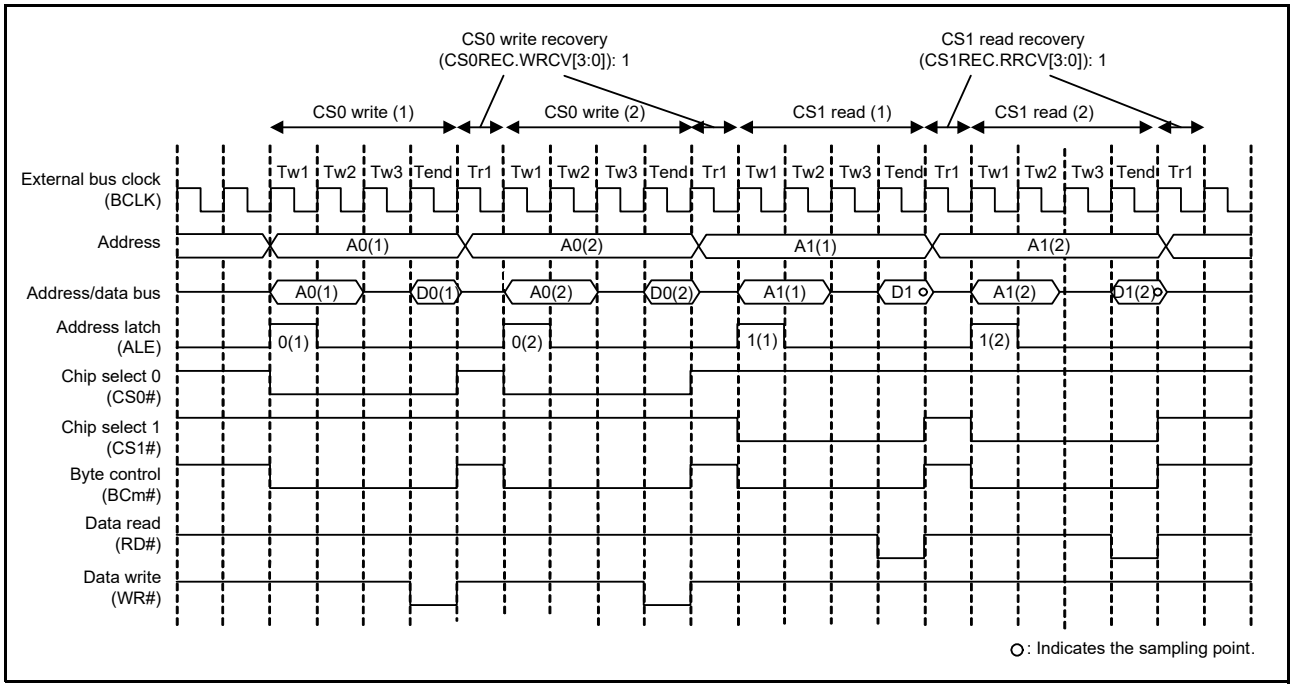


Figure 16.47 Example of Recovery Cycle Insertion When a Bus Access is Split with Address/Data Multiplexed I/O Interface (m = 0, 1)

16.5.5 No Access State

When no external address space is accessed, CSn#, BCn#, WRn#, and RD# signals are high, ALE signal is low, and D31 to D0 are in the high-impedance state.

16.5.6 Limitations

(1) Limitations on Using Separate Bus Interface

Limitations that apply to various bits of CSn wait control register 1 (CSnWCR1) and CSn wait control register 2 (CSnWCR2) at the times of normal and page accesses are listed in Table 16.17.
Even if the setting of the page-read access enable bit in the CSn mode register or the page-write access enable bit in the CSn mode register selects permission (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first round of access for page access and the access that does not fall within the scope of page access are normal access operation, and thus limitations on normal access must be satisfied.

Table 16.17 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

Note: When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.

(2) Limitations on Using Address/Data Multiplexed Bus Interface

In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed.
When the address/data multiplexed I/O interface is set, the BSIZE[1:0] bits in CSnCR should not be set to other than 8- and 16-bit bus spaces. If set, the operation cannot be guaranteed.

Table 16.18 Limitations at the Time of Normal Access

Limitations at the Time of Normal Access	
Reading	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON AWAIT[1:0] + 2 ≤ RDON CSON[2:0] ≤ AWAIT	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON AWAIT[1:0] + 2 ≤ WRON AWAIT[1:0] + 2 ≤ WDON CSON[2:0] ≤ AWAIT

(3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, setting the single write strobe mode is prohibited in the 8-bit bus space; otherwise the operation is not guaranteed.

(4) Limitations when 1/2 BCLK is Selected with EBCLK Pin Output Select Bit

When 1/2 BCLK is selected through the EBCLK pin output select bit, the external bus access cycle starts at the rising edge of the EBCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle may start at the falling edge of the EBCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

(5) Restriction on Instruction Code

The instruction code should be fixed to little endian. For details, refer to section 16.2.5.1, Restriction on Endian.

16.6 SDRAM Area Controller Operation

The following sections describe how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, which is followed by the description of SDRAMC operations including read, write, auto-refresh, self-refresh, initialization sequence, and mode register setting.

16.6.1 Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width

SDRAM access can be enabled or disabled using the SDC control register (SDCCR). The SDRAM bus width can also be set using SDCCR. Even when the operation of the SDRAM address space is disabled, refresh operation is available as long as self-refresh or auto-refresh operation is enabled.

16.6.2 No Access State

When no external address space is accessed, SDCS#, WE#, RAS#, and CAS# signals are high.

16.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn# signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued two cycles after the specified recovery cycle period elapsed after negation of CSn# signal at the earliest. Since no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to zero cycle).

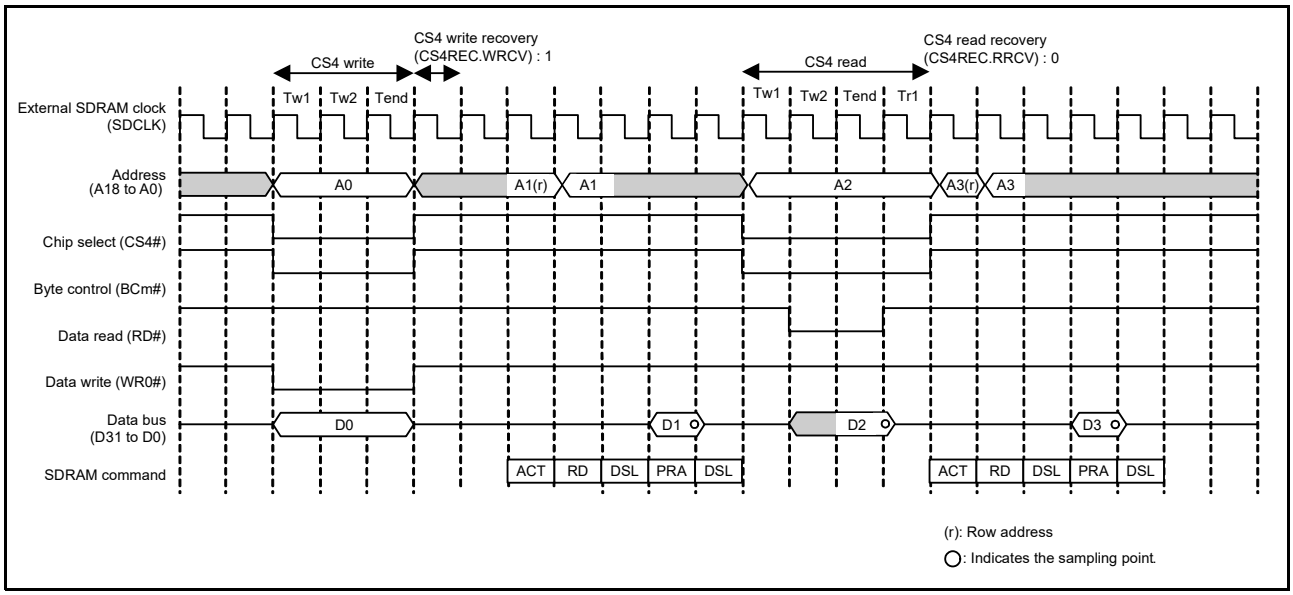


Figure 16.48 Example of Recovery Timing (for SDRAM Access)

16.6.4 SDRAM Commands

The SDRAMC issues a command for each bus cycle to control SDRAM. Commands are defined by combination of SDCS#, RAS#, CAS#, WE#, CKE, and other signals. Table 16.19 lists the commands issued by the SDRAMC.

Table 16.19 List of SDRAMC Commands

Name	Abbreviation	Command	SDCS#	RAS#	CAS#	WE#	CKE	
							n-1	n
DESL	DSL	Device deselect	H	x	x	x	H	x
ACTV	ACT	Bank active	L	L	H	H	H	x
READ	RD	Read	L	H	L	H	H	x
WRIT	WRI	Write	L	H	L	L	H	x
PALL	PRA	All bank precharge	L	L	H	L	H	x
REF	RFA	Auto-refresh	L	L	L	H	H	x
MRS	MRS	Mode register set	L	L	L	L	H	x
SELF	RFS	Self-refresh entry	L	L	L	H	H	L
SELFX	RFX	Self-refresh end	H	x	x	x	L	H

Note: H: High level, L: Low level, x: Don't care. (High level or low level)
n: Command issue cycle, n-1: One cycle before the command is issued.

16.6.5 Conditions for Setting SDRAMC Registers

SDRAMC registers should be modified only when all the corresponding conditions are satisfied as listed in Table 16.20.

Table 16.20 Conditions for Register Modification

Function/Operation	Registers	Conditions
Self-refresh	SDSELF*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled. (SDRFEN.RFEN = 1)
Auto-refresh	SDRFCR	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
	SDRFEN	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Initialization sequence	SDIR*1	The SDICR has not been set yet, and the same conditions as SDICR modification should be satisfied.
	SDICR*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Address register	SDADR	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Timing register	SDTR	• During self-refresh operation (SDSELF.SFEN = 1) or • SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Mode register	SDMOD*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
SDRAM Access Mode Register	SDAMOD	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)

Note 1. Before modification, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that the EXENB bit is set to 0.

16.6.6 Self-Refresh

Transition to or recovery from self-refresh mode can be controlled using the SDRAM self-refresh control register (SDSELF). Immediately before transition to self-refresh mode, auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle is started.
Figure 16.49 and Figure 16.50 show timing examples of transition to self-refresh mode and recovery from self-refresh mode, respectively.

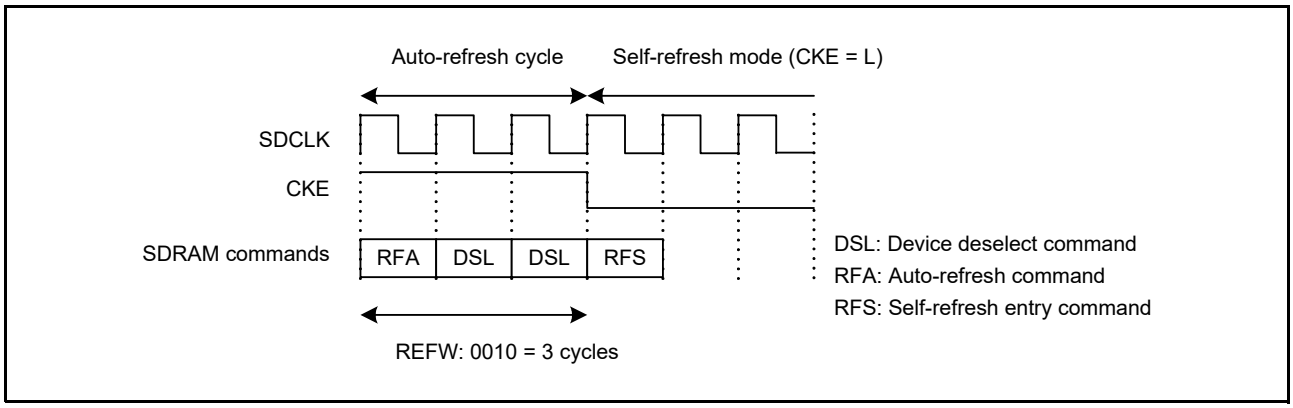


Figure 16.49 Timing Example of Transition to Self-Refresh Mode (when SDRFCR.REFW[3:0] = 0010b: 3 Cycles)

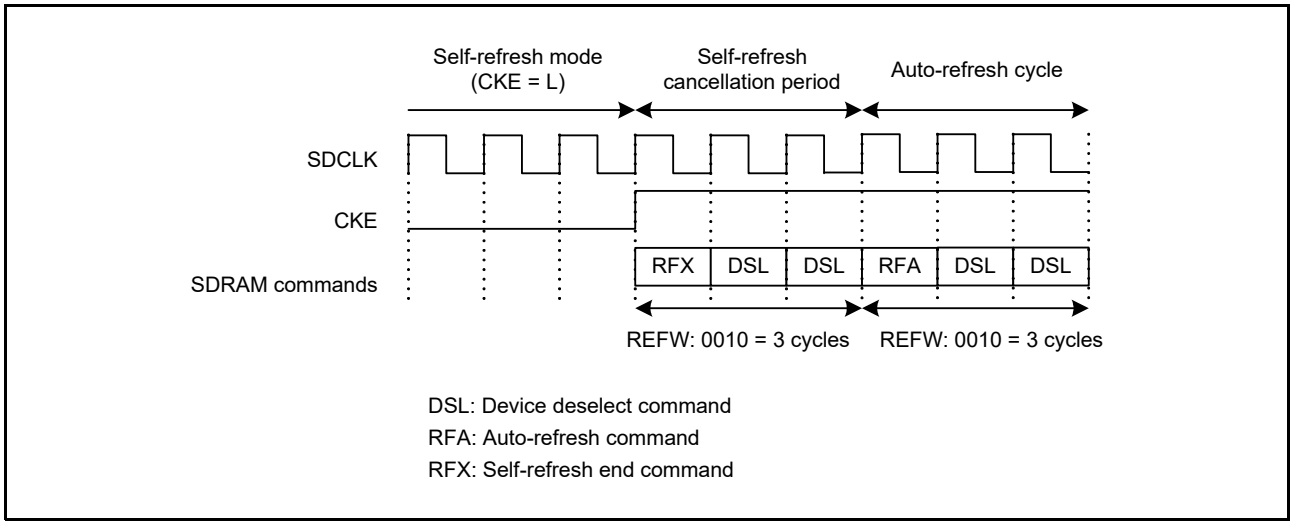


Figure 16.50 Timing Example of Recovery from Self-Refresh Mode

(1) Self-Refresh in Software Standby Mode

When causing transition to self-refresh mode in software standby mode, first cause transition to self-refresh mode according to the procedure shown in section 16.6.11.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to software standby mode. In software standby mode, set the output port enable bit (OPE) in the standby control register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.
After canceling software standby mode, follow the procedure shown in section 16.6.11.2, Procedure for Transition to and Recovery from Self-Refresh Mode. For details of transition and cancellation of software standby mode, refer to section 11, Low Power Consumption.

(2) Self-Refresh in Deep Software Standby Mode

Transition to deep software standby mode is performed via software standby mode. On transition to deep software standby mode from software standby mode, the state of pins remains unchanged. Therefore, transition to self-refresh mode in deep software standby mode can be made according to the same procedure as that in software standby mode. In deep software standby mode, however, additional setting is necessary to cause transition to self-refresh mode; it is necessary to set the I/O port keep bit (IOKEEP) in the deep software standby control register (DPSBYCR) to 1.
Since the SDRAMC is internally reset by an internal reset signal when deep software standby mode is canceled, the SDRAM control registers need to be set again. After canceling deep software standby mode, follow the procedure shown below to cancel self-refresh mode. Figure 16.51 shows self-refresh timing in deep software standby mode.
For details of transition and cancellation of deep software standby mode, refer to section 11, Low Power Consumption.

1. Set DPSBYCR.IOKEEP to 1 to keep the CKE signal output low in deep software standby mode.
2. Start clock supply to SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDADR, and SDTR) again, which have been initialized by an internal reset upon transition to deep software standby mode.
4. Enable auto-refresh operation (RFEN bit in SDRFEN = 1).
5. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 1 to set self-refresh mode again.
6. Modify port settings for the SDRAM interface.
7. Set the SDCKOEN bit in SDCKOCR to 1 to start clock supply to the SDRAM via the SDCLK pin.
8. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 0 to cancel self-refresh mode.

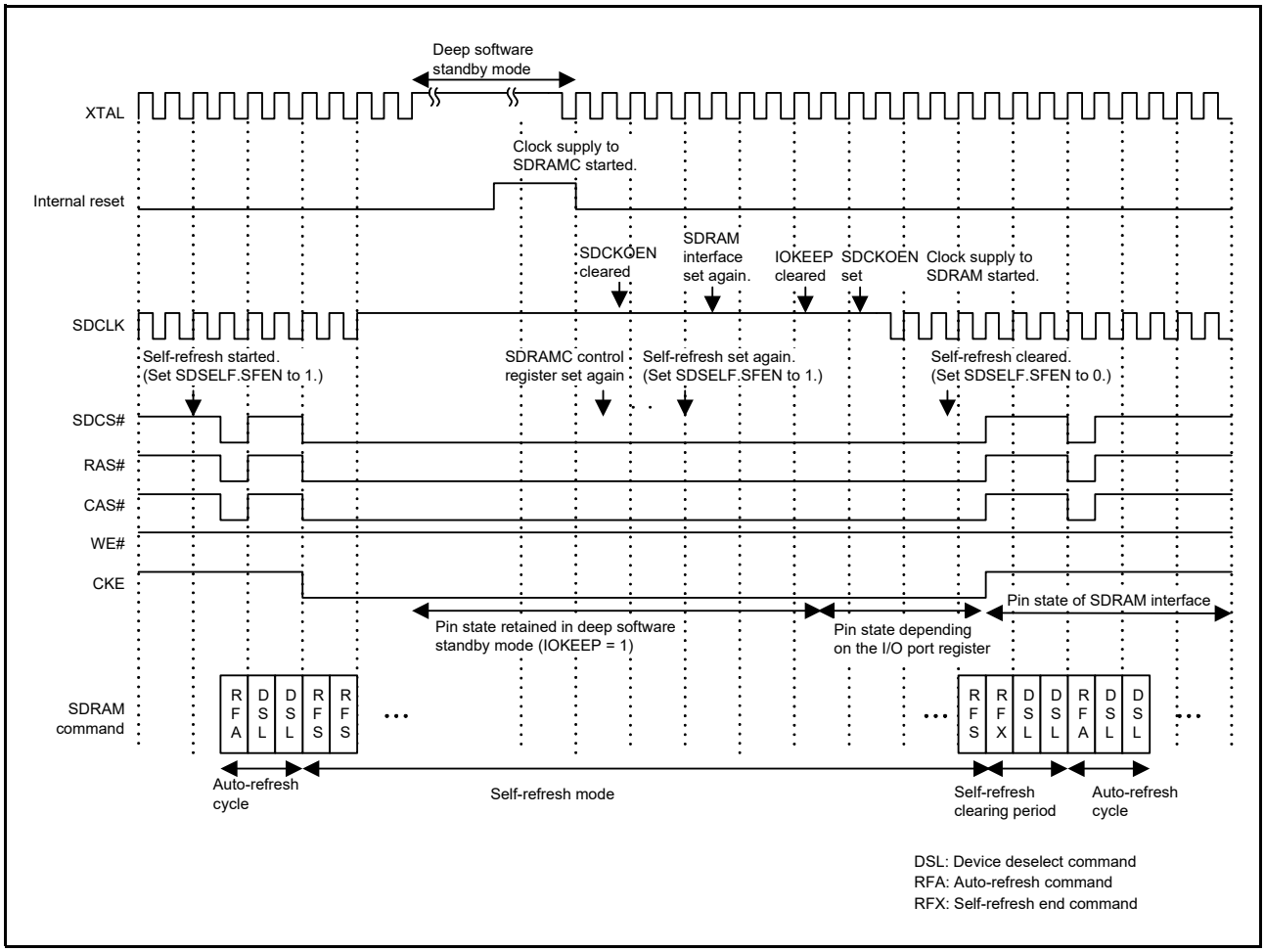


Figure 16.51 Timing Example of Self-Refresh Cycle (in Deep Software Standby Mode)

16.6.7 Auto-Refresh

The auto-refresh cycle can be started by setting the auto-refresh operation enable bit (RFEN) in the SDRAM auto-refresh control register (SDRFEN) to 1. Once the cycle is started, refresh requests are generated at fixed intervals determined by the refresh counter to start the auto-refresh cycle. However, since refresh requests are not accepted during read/write access, the auto-refresh cycle may be suspended.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS#, RAS#, CAS#, WE#, and CKE signals, which are necessary for issuing the refresh command, are exclusively provided for SDRAM access.

The refresh counter is halted during self-refresh operation. After recovery from self-refresh mode, the auto-refresh cycle is started and the counter value is reset thus resuming the counter operation.

Figure 16.52 shows an example of the timing of an auto-refresh cycle.

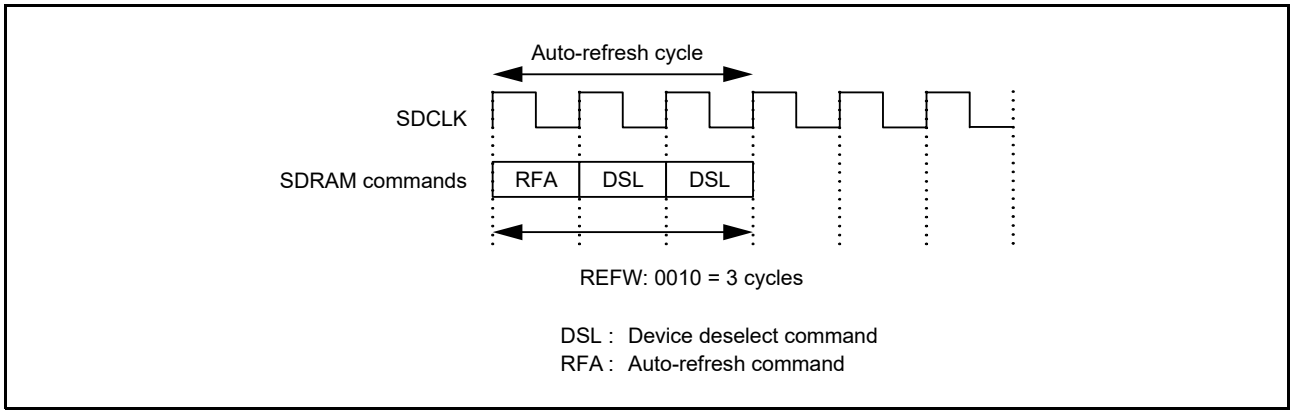


Figure 16.52 Timing Example of Auto-Refresh Cycle (1)

Figure 16.53 and Figure 16.54 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

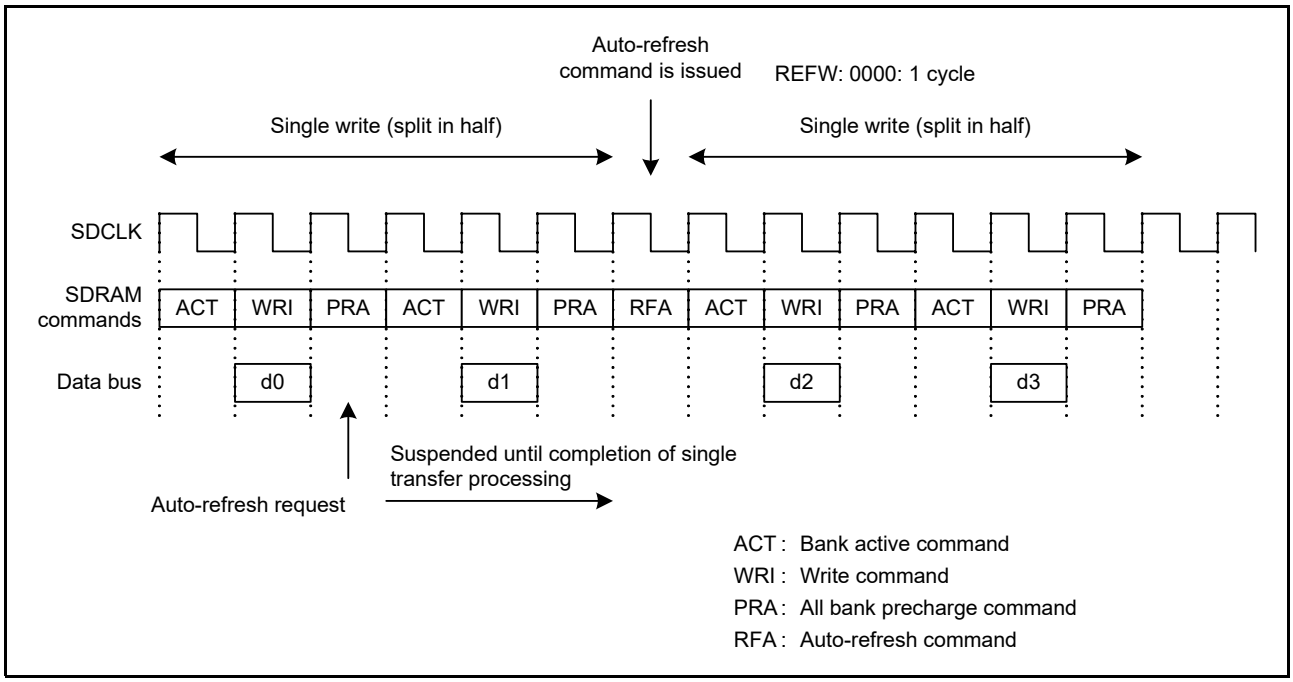


Figure 16.53 Timing Example of Auto-Refresh Cycle (2) (Auto-Refresh Request is Made during Single Access)

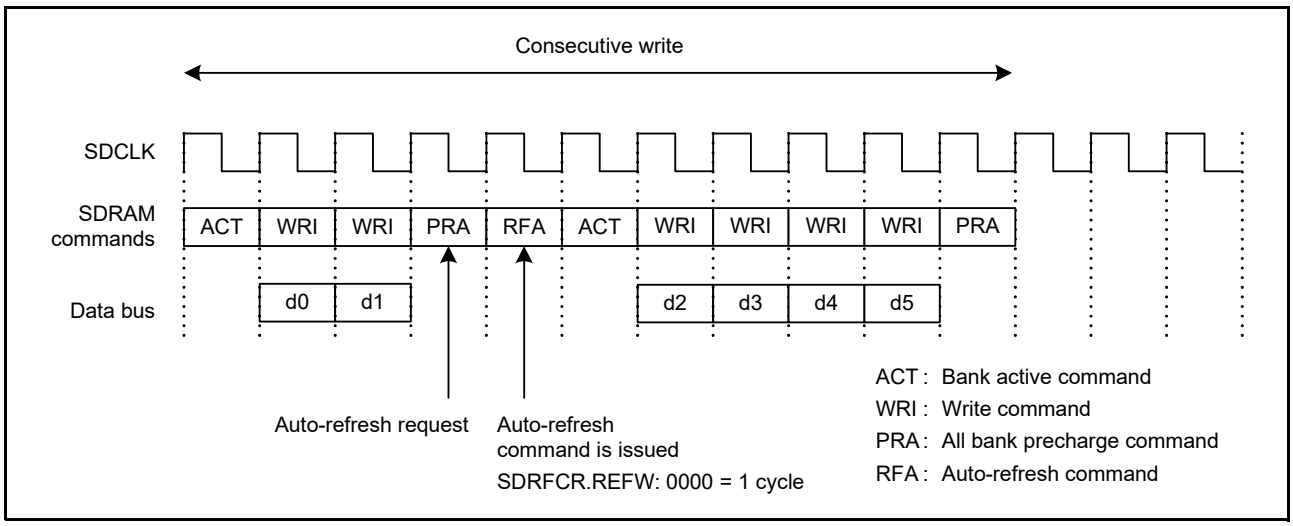


Figure 16.54 Timing Example of Auto-Refresh Cycle (3) (Auto-Refresh Request is Made during Continuous Access)

16.6.8 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequence must be activated; the operation is not guaranteed if the SDRAM is not initialized.

The SDRAM initialization sequencer issues an all-bank-precharge command followed by auto-refresh commands n times (n = 1 to 15). The SDRAM initialization sequence timing can be set using the SDRAM initialization register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM initialization sequence control register (SDICR). These registers should be set only when the conditions listed in Table 16.20, Conditions for Register Modification.

Figure 16.55 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

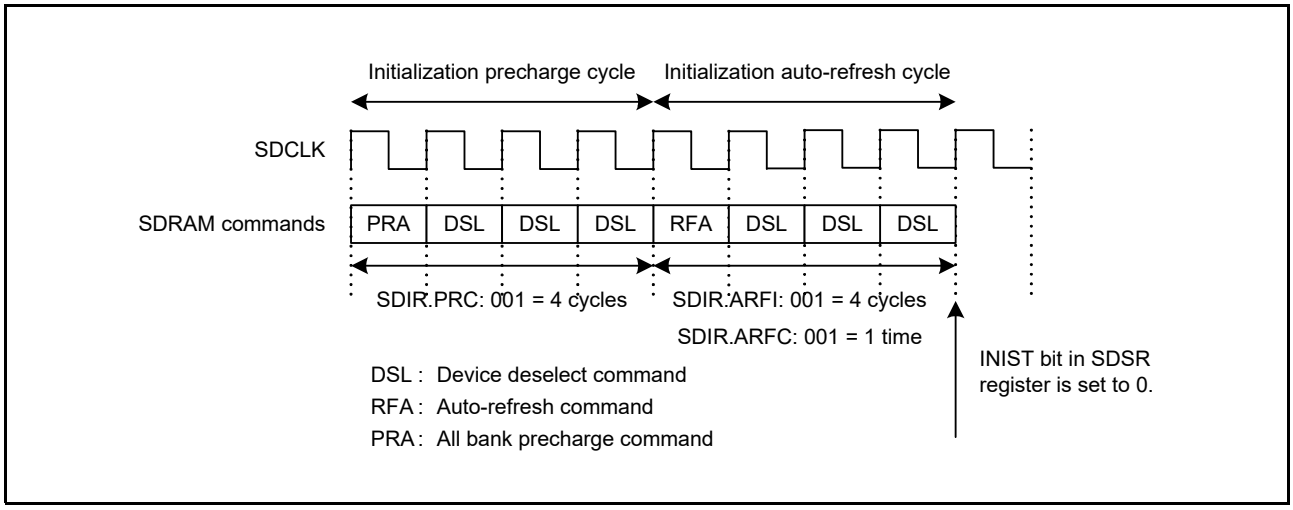


Figure 16.55 Timing Example of SDRAM Initialization Sequence

16.6.9 Setting Mode Register

Setting the SDRAM mode register (SDMOD) allows the mode register set command to be issued to SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address; specifically, to the A14 to A0 for 8-bit bus width, A15 to A1 for 16-bit bus width, and A16 to A2 for 32-bit bus width. Therefore, set the SDCCR.BSIZE[1:0] bits before setting the mode register, to determine the data bus width of the SDRAM.
Figure 16.56 shows the mode register setting timing.

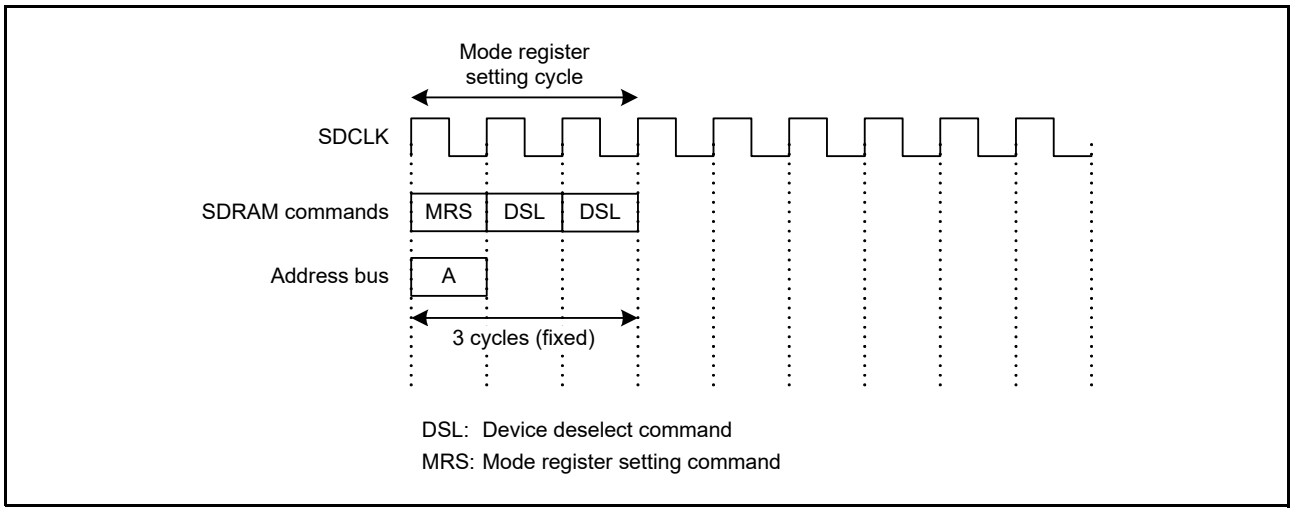


Figure 16.56 Mode Register Setting Timing

16.6.10 Read/Write Access

The SDRAMC controls read/write access in the following two modes.

- Single access mode: the row address is output each time data is accessed
- Consecutive access mode: when the same row address is accessed consecutively, only the column address is changed after the row address is output, enabling quick data access.

Consecutive SDRAM access is enabled by setting the continuous access enable bit (BE) in SDRAM access mode register (SDAMOD) to 1.

If the transfer from the bus master has a burst length of 2 or more and the transfer data size is less than or equal to the external bus width, consecutive access will occur.

(1) Single Access

Figure 16.57 shows timing examples for single read and single write. The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 16.6.11.3, Timing Register Settings and Access Timing.

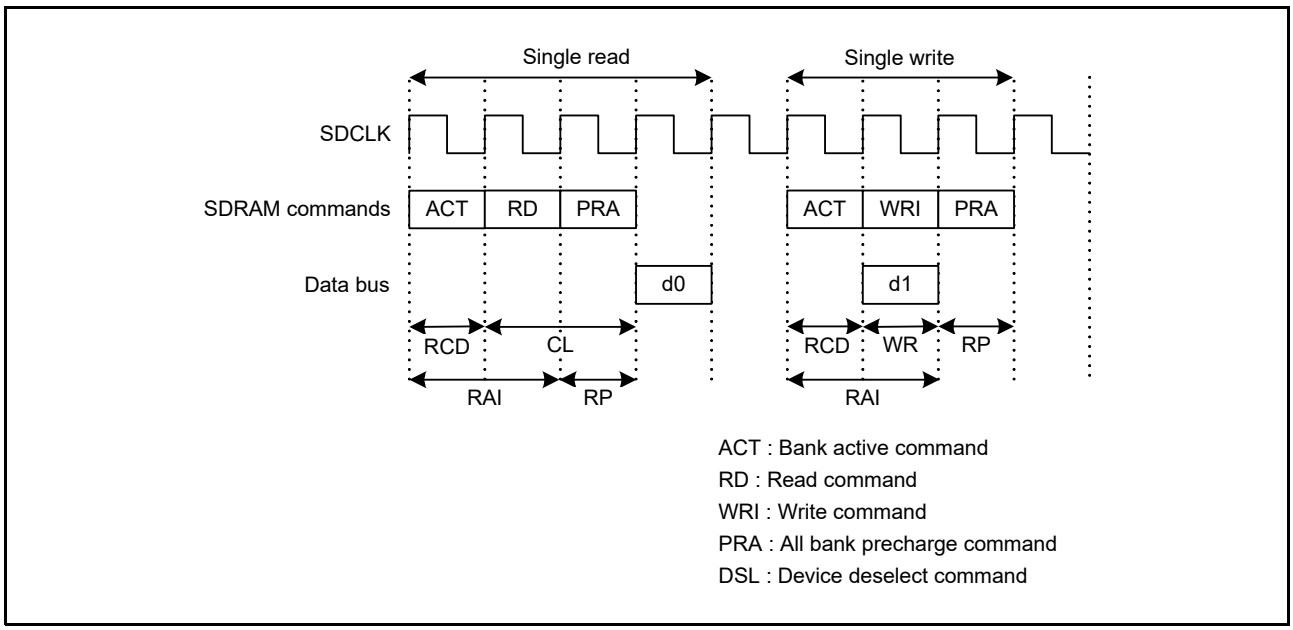


Figure 16.57 Timing Example of Single Read / Write

(2) Consecutive Access

Figure 16.58 show timing examples of consecutive read and consecutive write for four data, respectively.
When the SDRAM row address changes during transfer, the pertinent row is automatically deactivated, and then activated.
The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 16.6.11.3, Timing Register Settings and Access Timing.

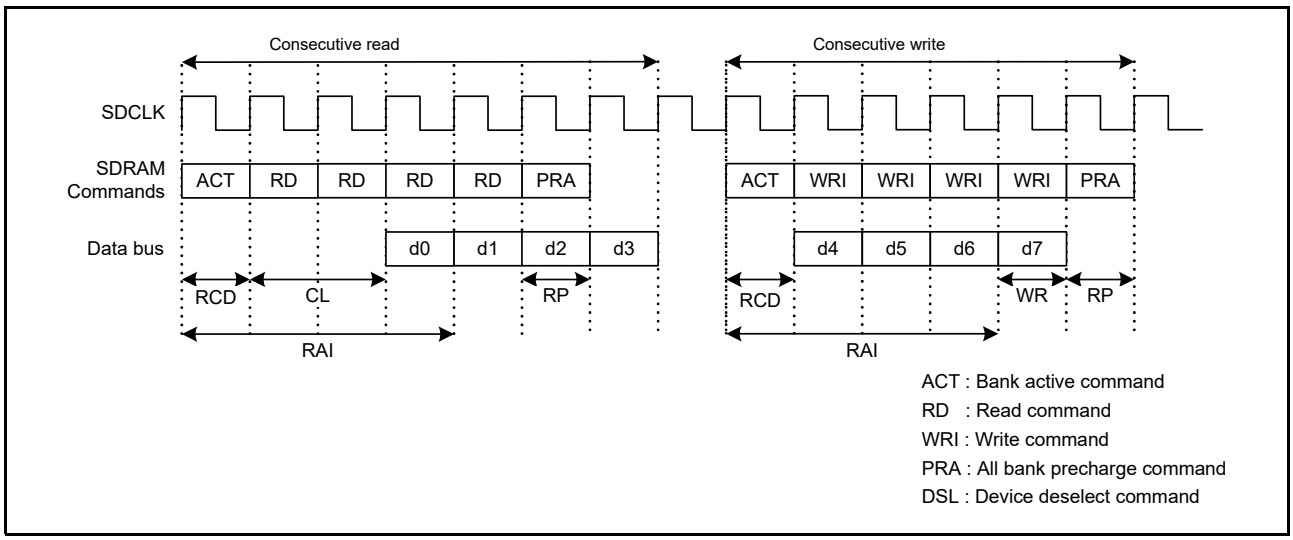


Figure 16.58 Timing Example of Consecutive Read / Write

Figure 16.59 shows a timing example of consecutive write in which the row address changes.

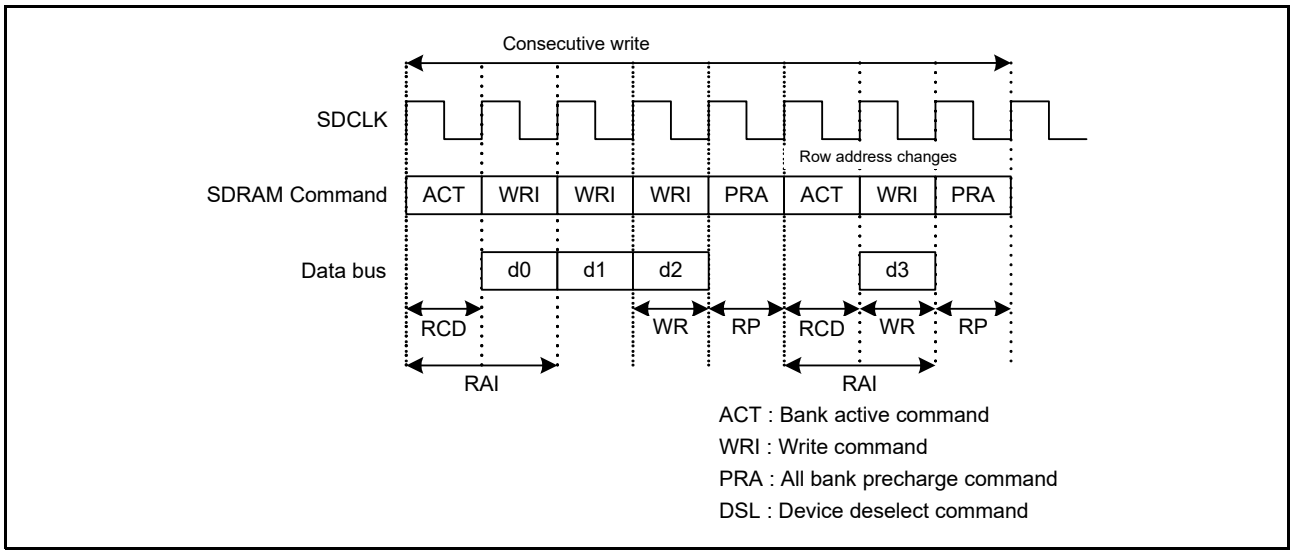


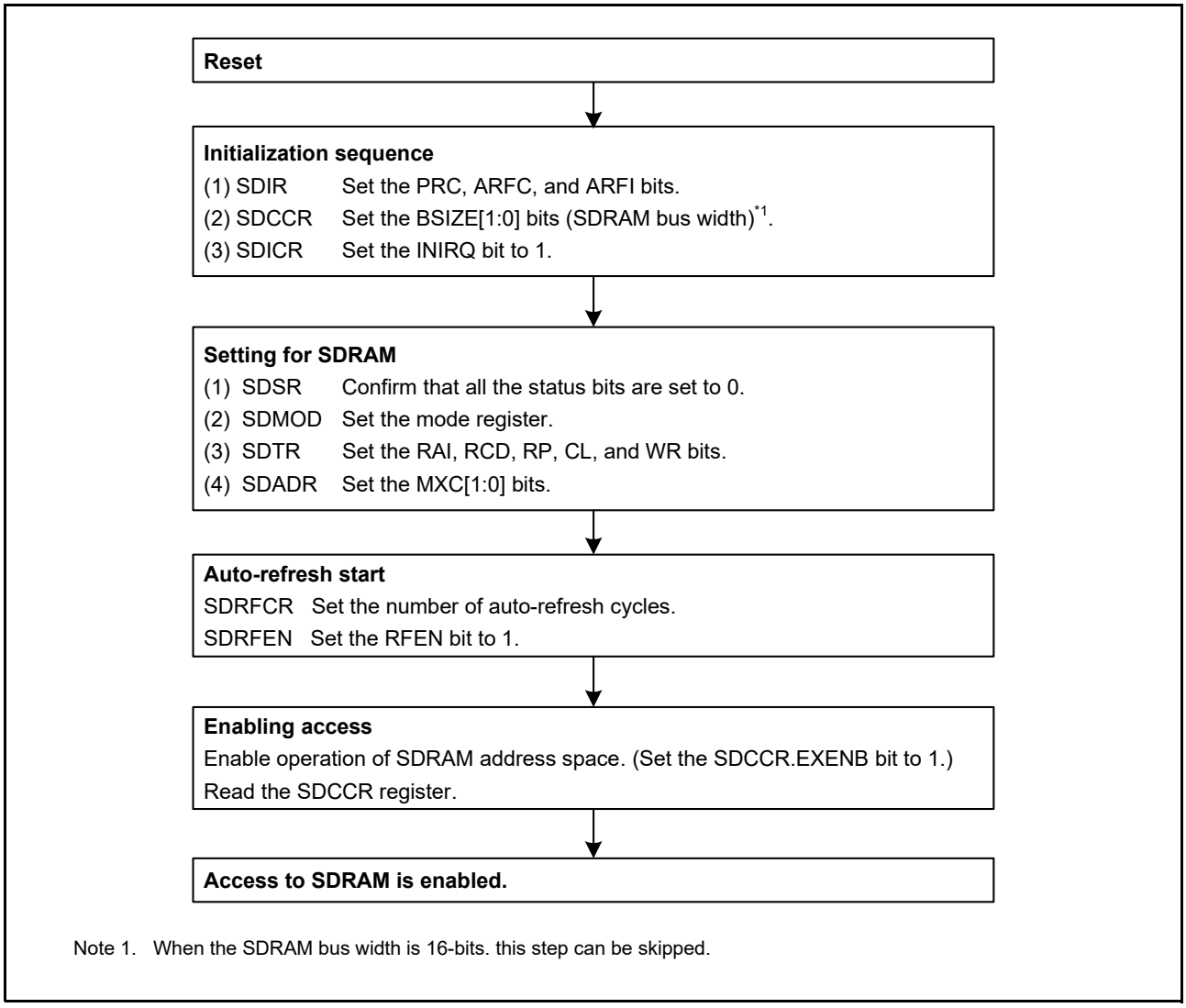
Figure 16.59 Timing Example of Consecutive Write in which the Row Address Changes

16.6.11 SDRAMC Setting Examples

This section describes the SDRAMC setting procedure, timing register setting examples, and procedure for transition to and recovery from self-refresh mode.

16.6.11.1 SDRAMC Access Procedure

Figure 16.60 shows the SDRAMC setting procedure.



Note 1. When the SDRAM bus width is 16-bits. this step can be skipped.

Figure 16.60 SDRAMC Setting Procedure

16.6.11.2 Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 16.61 shows the procedure for transition to and recovery from self-refresh mode.

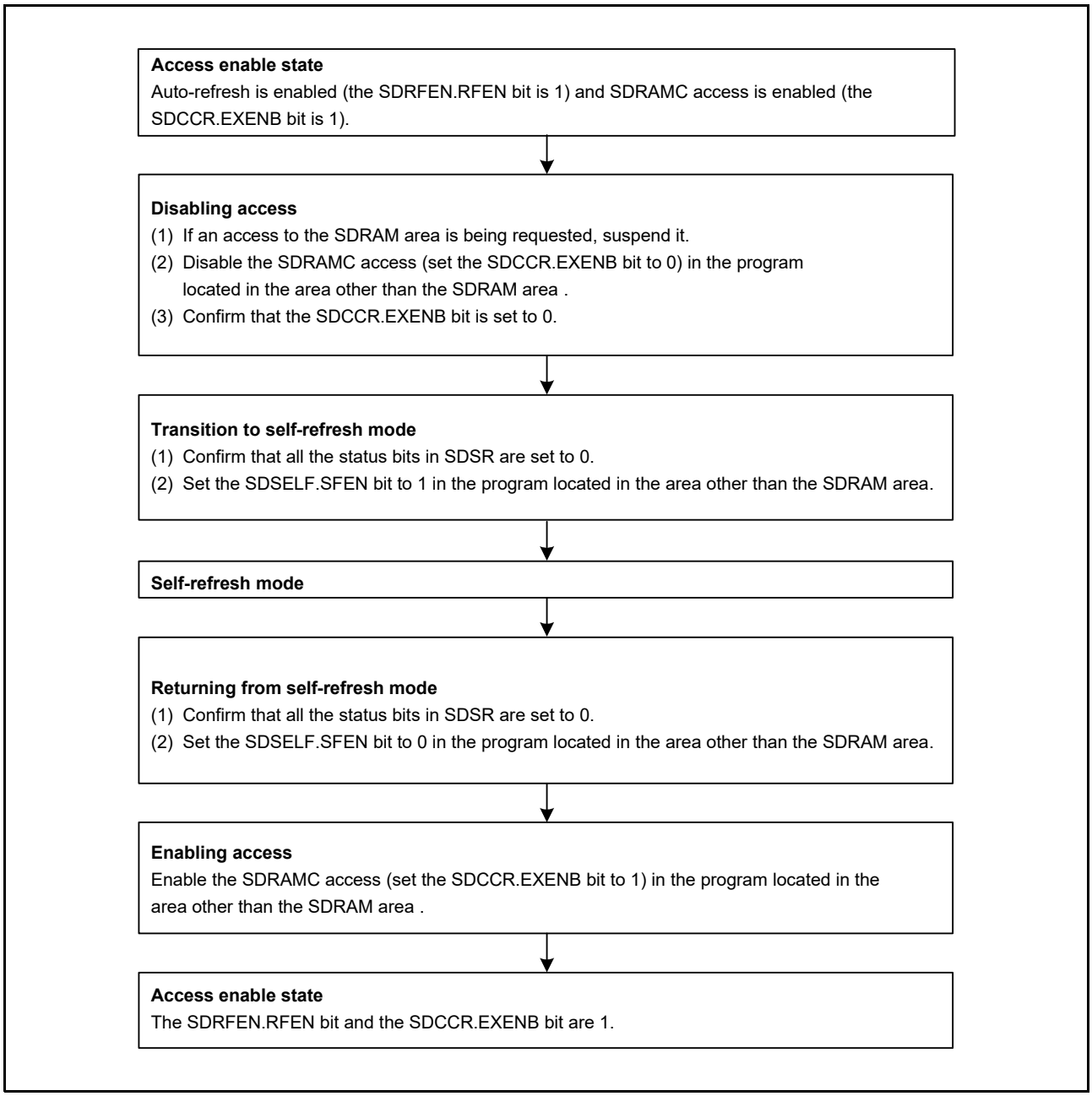


Figure 16.61 Procedure for Transition to and Recovery from Self-Refresh Mode

Note: Transition to and recovery from self-refresh mode requires SDRAM access to be disabled. Accordingly, transition to and recovery from self-refresh mode cannot be made during SDRAM access. The instructions in Figure 16.62 should be followed in programming.
- Before making transition to self-refresh mode, disable the access to the SDRAM area.
- During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch (including prefetch) to the SDRAM area to be generated.

Figure 16.62 shows the procedure for transition to and recovery from self-refresh mode in deep software standby mode.

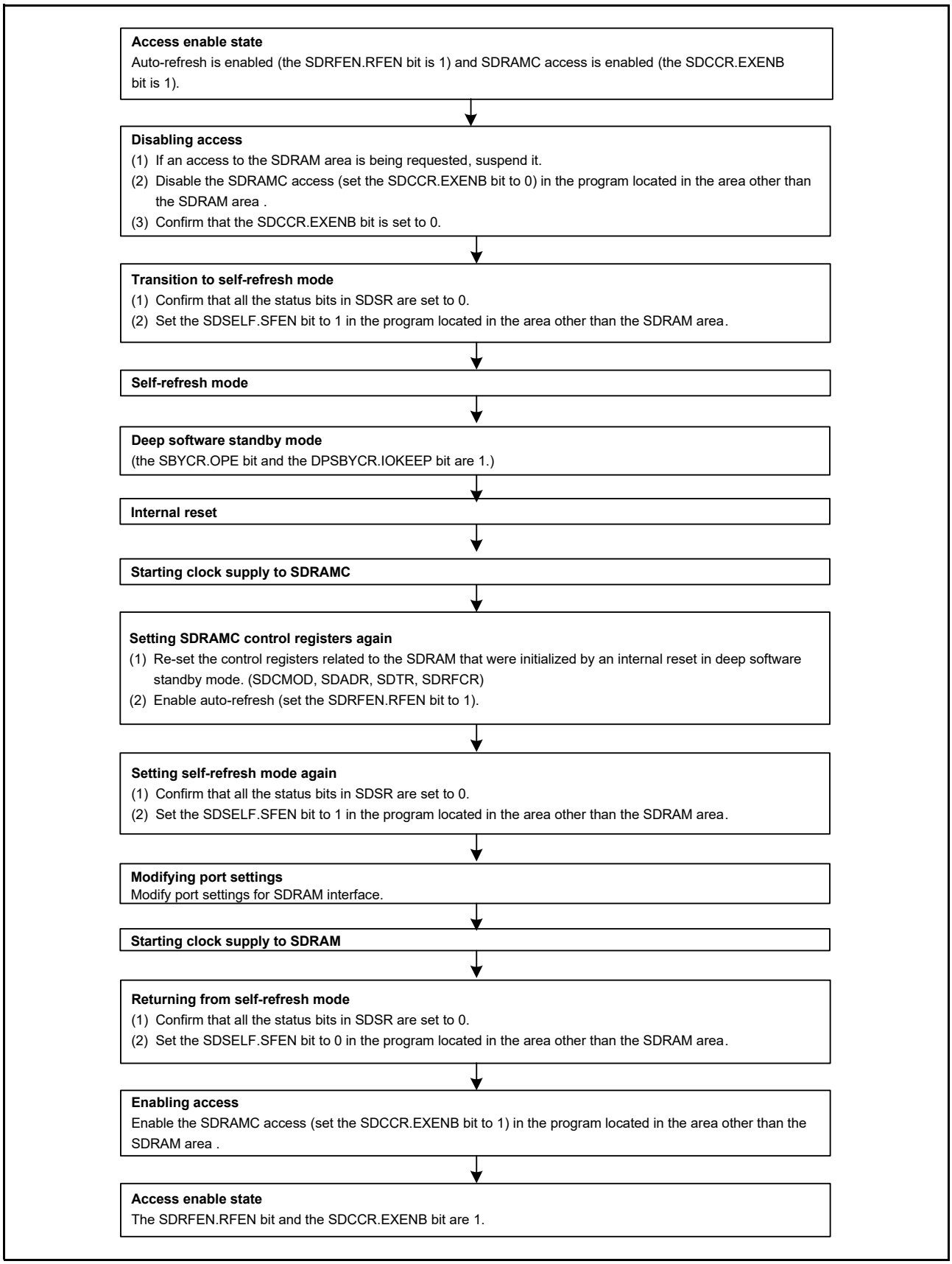


Figure 16.62 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode

16.6.11.3 Timing Register Settings and Access Timing

This section describes the relationship between the read/write timing and the settings of the SDRAM timing register (SDTR).

(1) Single Read Timing Examples

Figure 16.63 to Figure 16.67 show the relationship between the single read timing and the SDTR register settings. Table 16.21 lists the correspondence between the figures and the SDTR register settings.
During read access, the next bus access is enabled two cycles after the read data becomes valid at the earliest. However, if two or more accesses occur for one transfer request, the next bus access is enabled one cycle after the read data becomes valid at the earliest, as shown in Figure 16.67.

Table 16.21 Correspondence between Target Figures and SDTR Register Settings (Single Read Timing)

Figure No.	RAI[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	CL[2:0] Settings	Number of Cycles
Figure 16.63	010	3	00	1	001	2	010	2
Figure 16.64	000	1	01	2	001	2	010	2
Figure 16.65	000	1	01	2	001	2	011	3
Figure 16.66, Figure 16.67	010	3	00	1	000	1	010	2

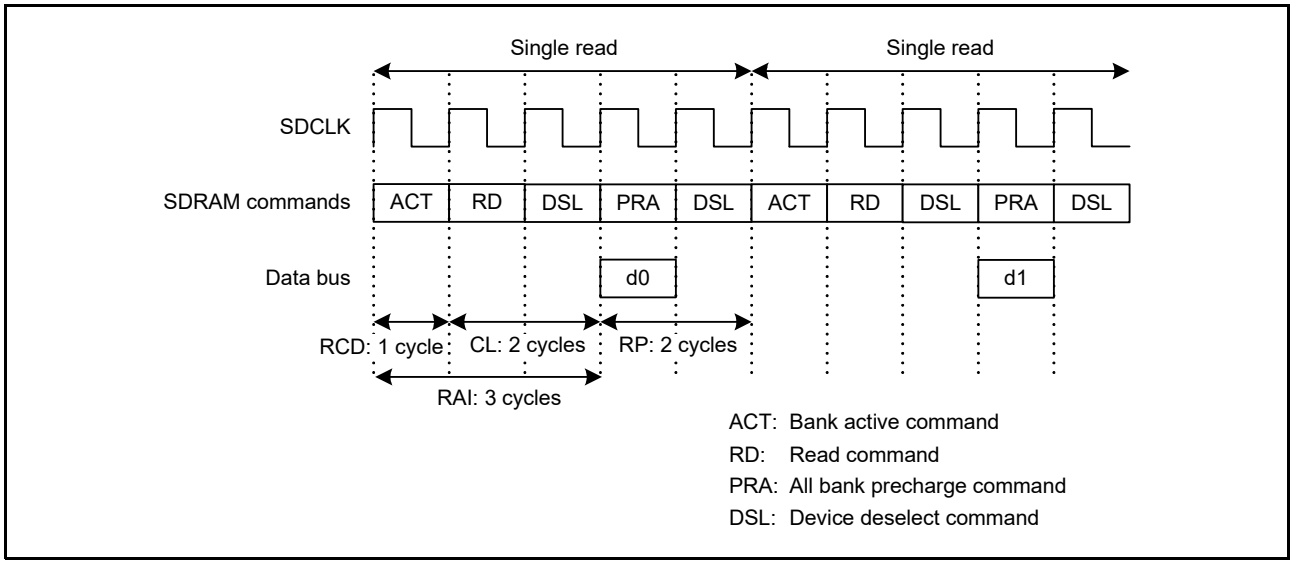


Figure 16.63 Timing Example of Single Read (1)

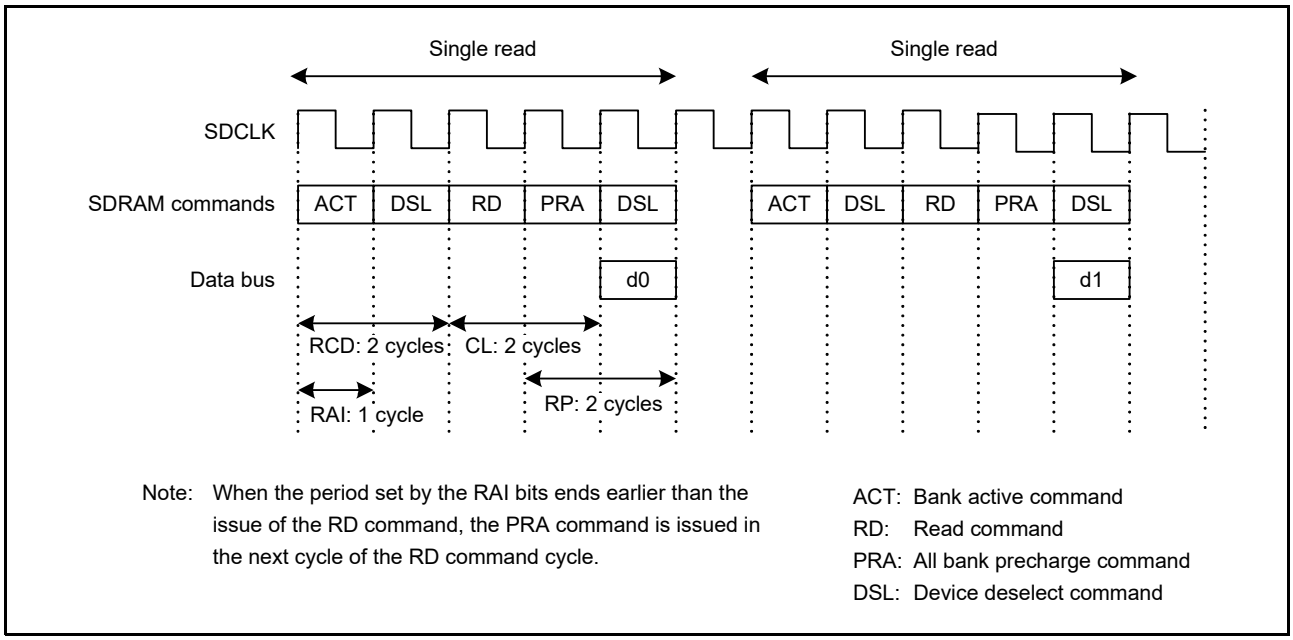


Figure 16.64 Timing Example of Single Read (2)

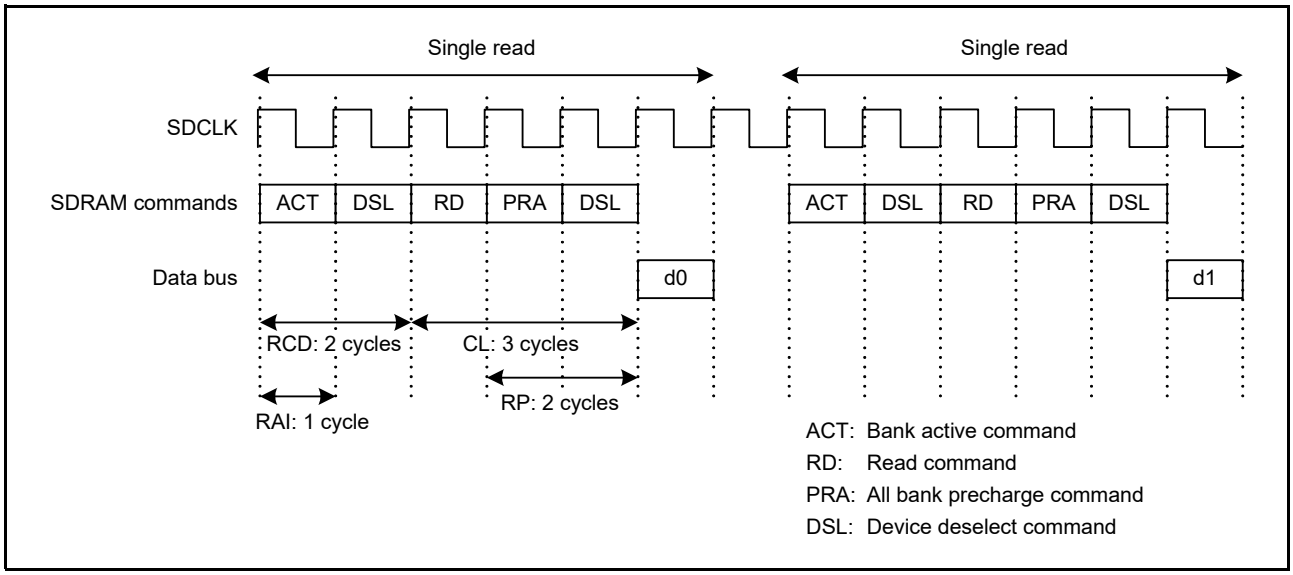


Figure 16.65 Timing Example of Single Read (3)

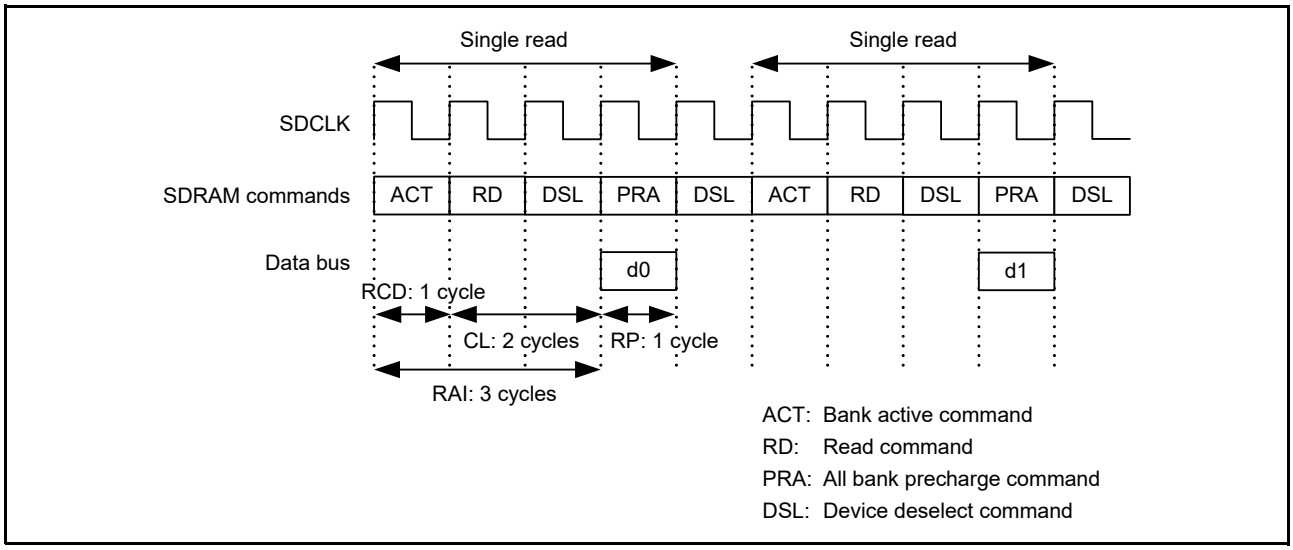


Figure 16.66 Timing Example of Single Read (4)

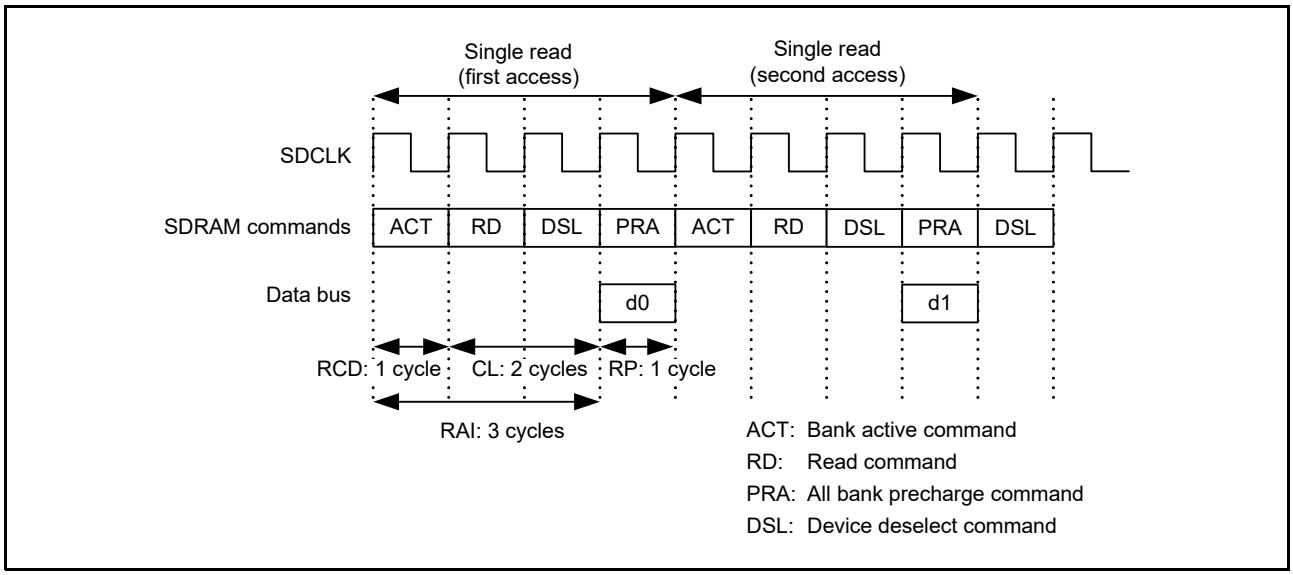


Figure 16.67 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request)

(2) Single Write Timing Examples

Figure 16.68 to Figure 16.72 show the relationship between the single write timing and the SDTR register settings. Table 16.22 lists the correspondence between the figures and the SDTR register settings.
During write access, the next bus access is enabled two cycles after an all-bank-precharge command (PRA) is issued at the earliest. However, if two or more accesses occur for one transfer request, the next bus access is enabled one cycle after the PRA is issued at the earliest, as shown in Figure 16.72.

Table 16.22 Correspondence between Target Figures and SDTR Register Settings (Single Write Timing)

Figure No.	RAI[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.68	010	3	00	1	001	2	0	1
Figure 16.69	000	1	01	2	001	2	0	1
Figure 16.70	000	1	01	2	001	2	1	2
Figure 16.71, Figure 16.72	010	3	00	1	000	1	0	1

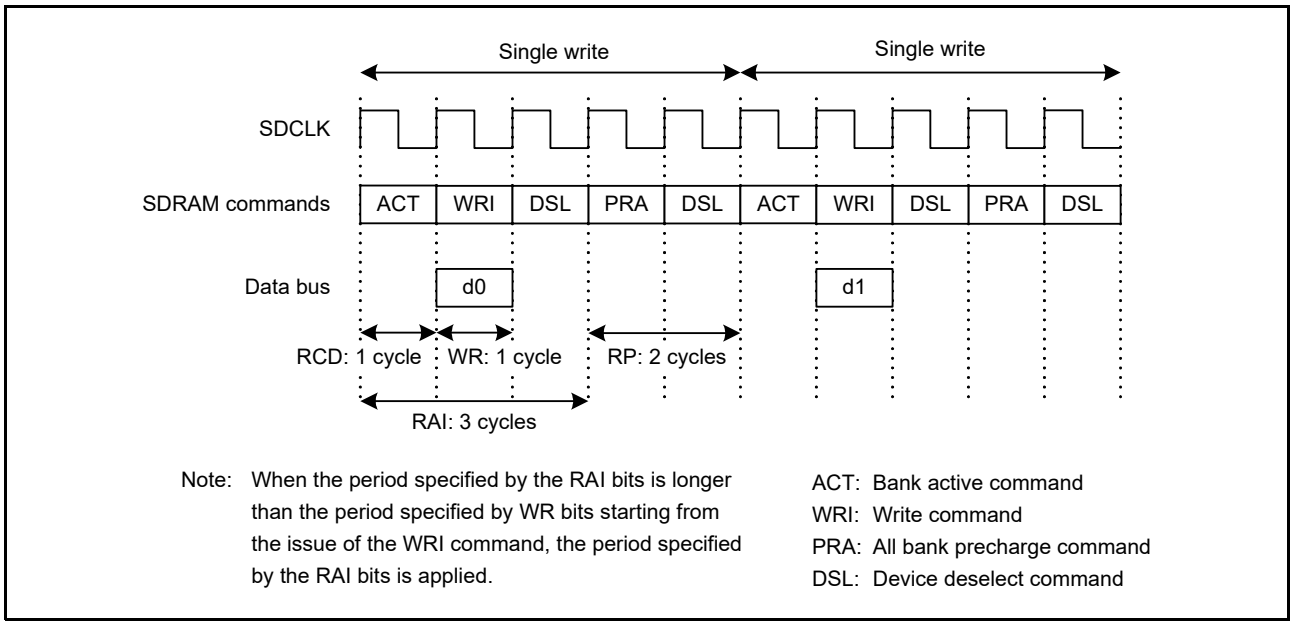


Figure 16.68 Timing Example of Single Write (1)

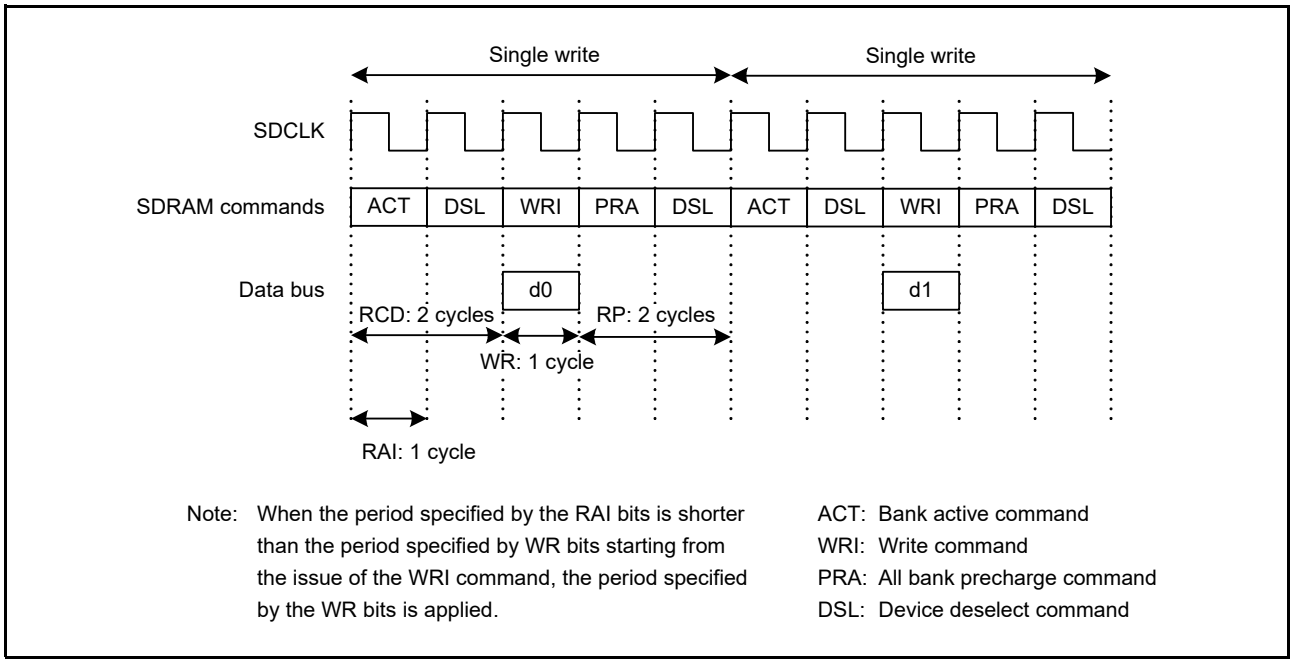


Figure 16.69 Timing Example of Single Write (2)

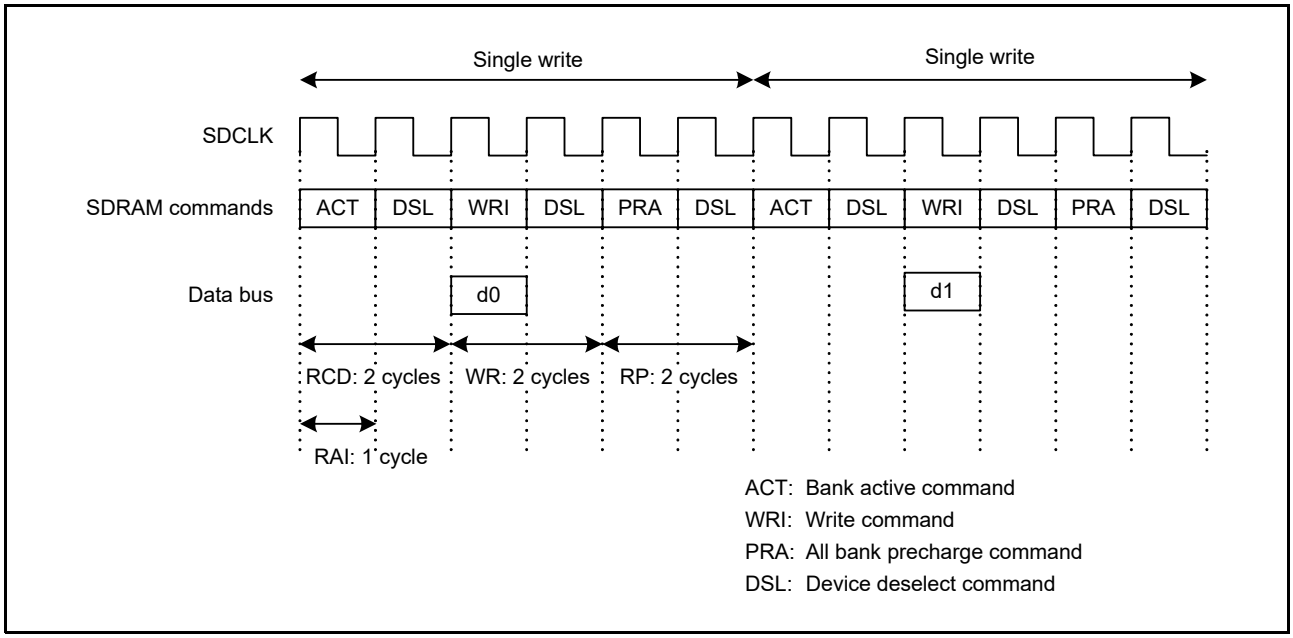
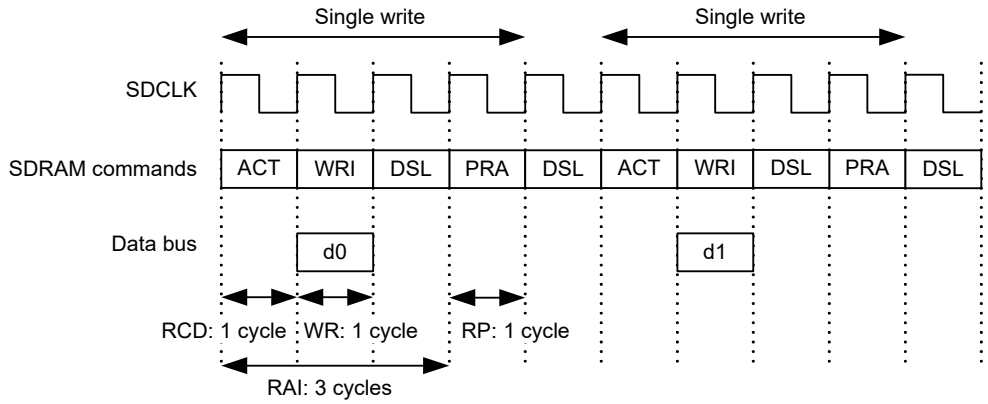


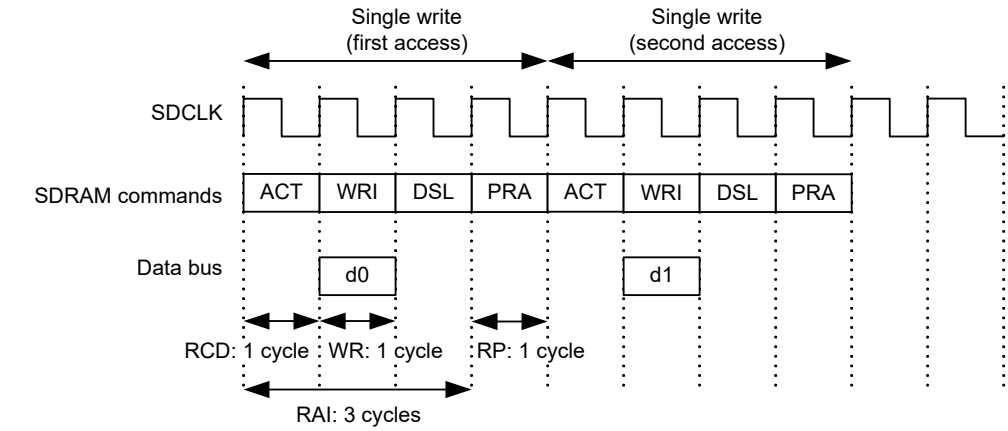
Figure 16.70 Timing Example of Single Write (3)



Note: When the period set by the RAI bits outlasts the WR period, which begins at issuance of the WRI command, the RAI setting is used.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 16.71 Timing Example of Single Write (4)



Note: When the period set by the RAI bits is outlasts the WR period, which begins at issuance of the WRI command, the RAI setting is used.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 16.72 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request)

(3) Consecutive Read Timing Examples

Figure 16.73 to Figure 16.75 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 16.23 lists the correspondence between the figures and the SDTR register settings.

Table 16.23 Correspondence between Target Figures and SDTR Register Settings (Consecutive Read Timing)

Figure No.	RAI[2:0] Settings	Number of Cycle	RCD[1:0] Settings	Number of Cycle	RP[2:0] Settings	Number of Cycle	CL[2:0] Settings	Number of Cycle
Figure 16.73	010	3	00	1	001	2	010	2
Figure 16.74	000	1	01	2	001	2	010	2
Figure 16.75	000	1	01	2	001	2	011	3

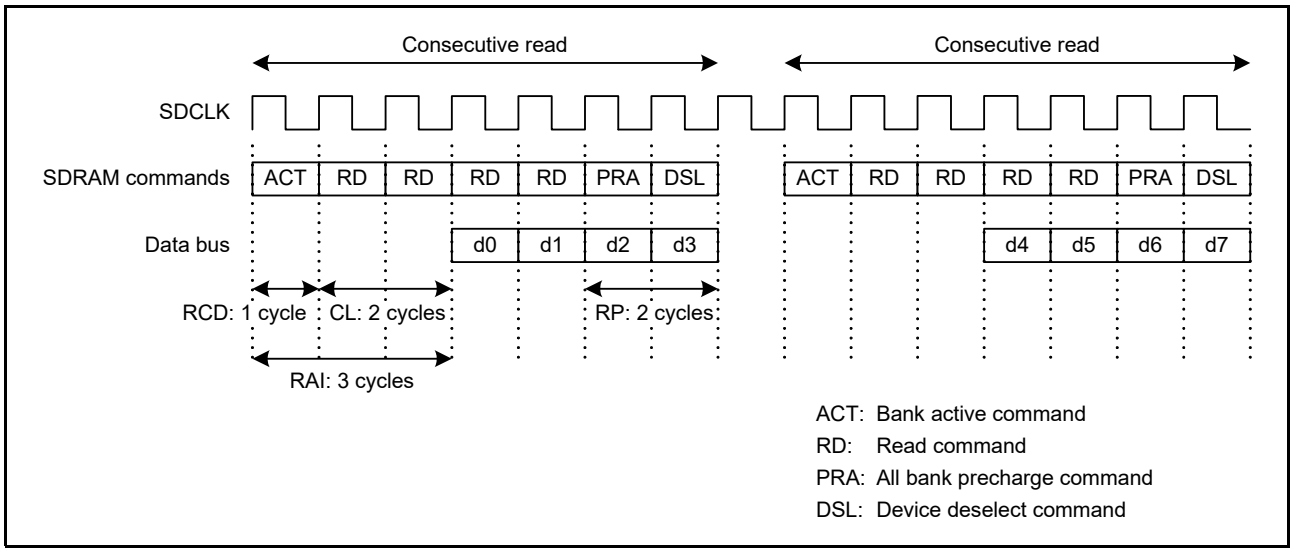


Figure 16.73 Timing Example of Consecutive Read (1)

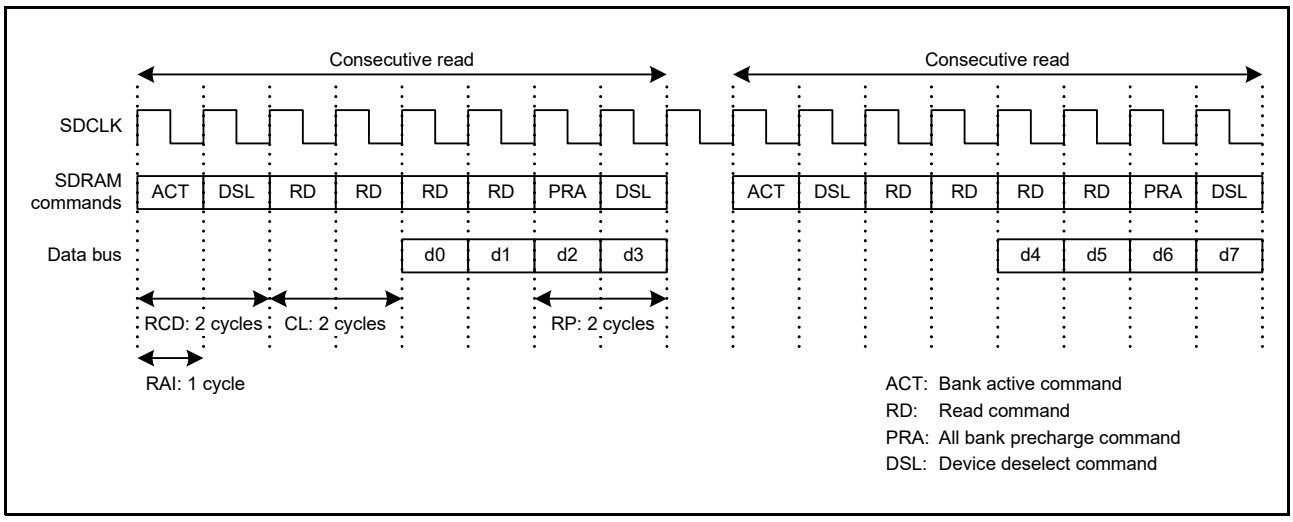


Figure 16.74 Timing Example of Consecutive Read (2)

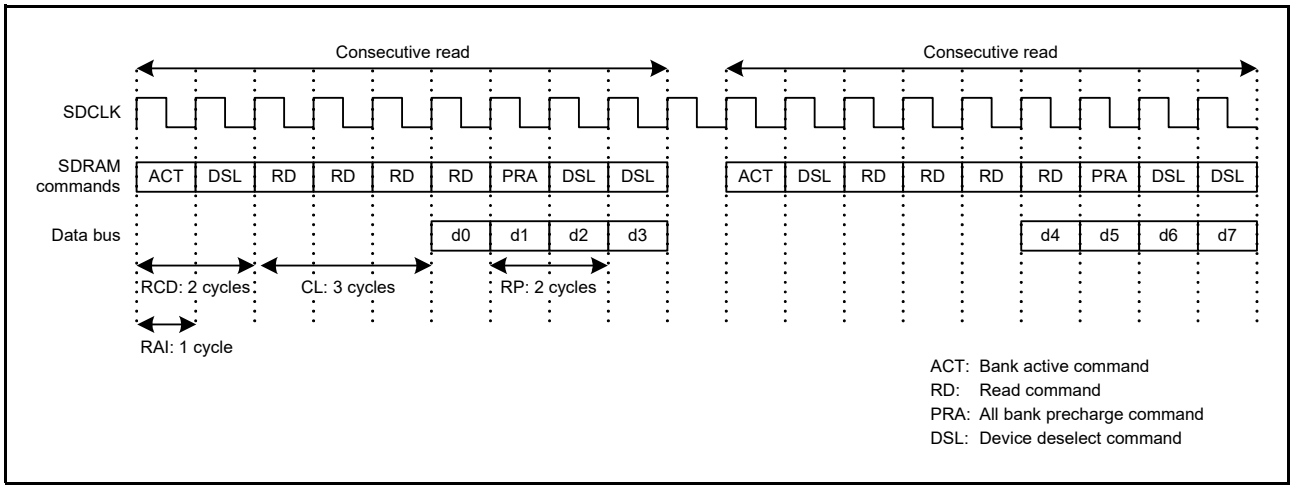


Figure 16.75 Timing Example of Consecutive Read (3)

(4) Consecutive Write Timing Examples

Figure 16.76 to Figure 16.78 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 16.24 lists the correspondence between the figures and the SDTR register settings.

Table 16.24 Correspondence between Target Figures and SDTR Register Settings (Consecutive Write Timing)

Figure No.	RAI[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.76	010	3	00	1	001	2	0	1
Figure 16.77	000	1	01	2	001	2	0	1
Figure 16.78	000	1	01	2	001	2	1	2

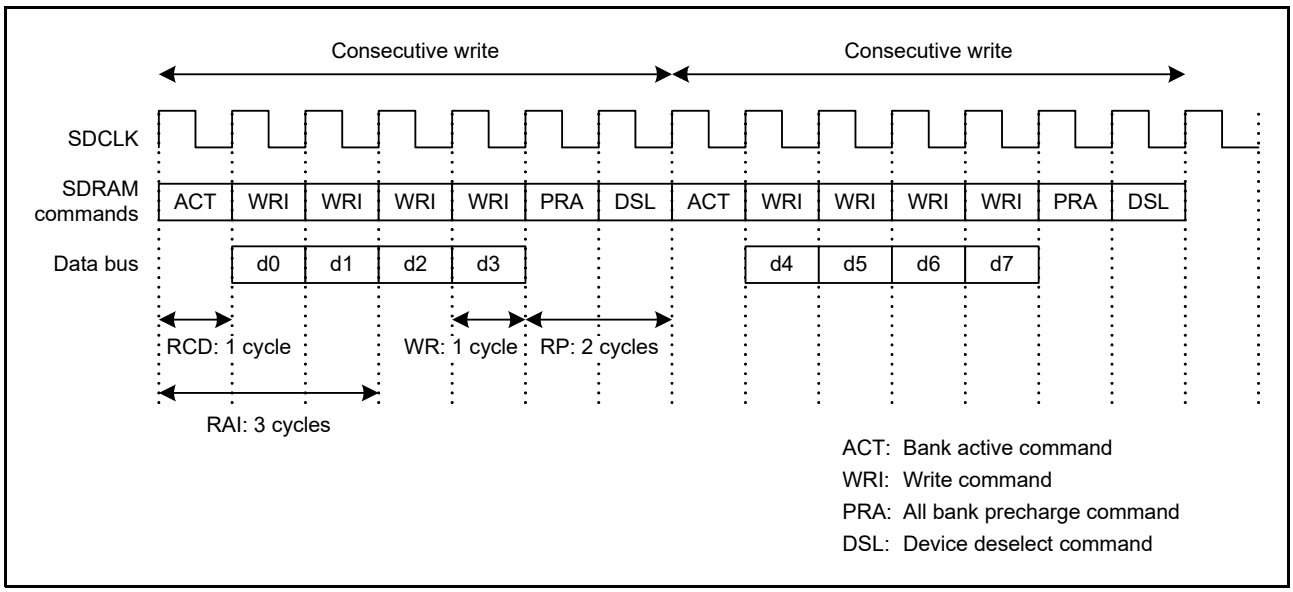


Figure 16.76 Timing Example of Consecutive Write (1)

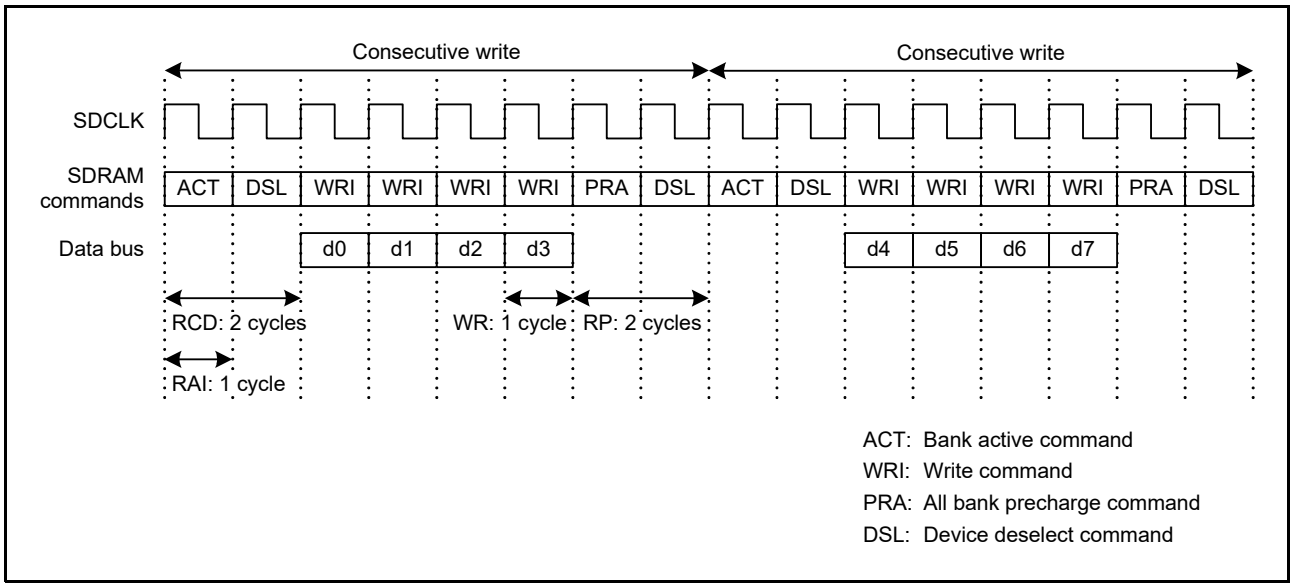


Figure 16.77 Timing Example of Consecutive Write (2)

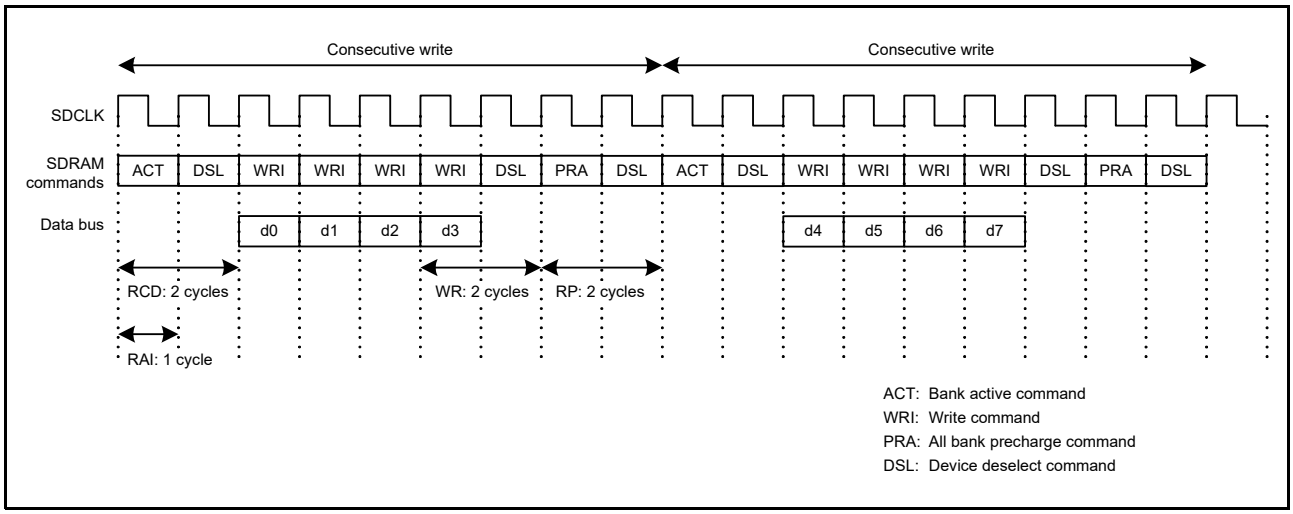


Figure 16.78 Timing Example of Consecutive Write (3)

16.6.12 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address should be specified for address multiplexing by the address multiplex select bits (SDADR.MXC[1:0]) in the SDRAM address register (SDADR). Moreover, in the SDRAM space, the address precharge select command (precharge-sel) is output to the upper bits of column addresses. Table 16.25 lists the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 16.25 Address Multiplexing

MXC [1:0]	Shift Amount	Data Bus Width	Address	Address Pins External to the Microcomputer																		
				A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	8 bits	8 bits	Row	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	A[19]	A[18]*	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]
			Column	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	A[19]	P	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		16 bits	Row	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	A[19]*	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]
			Column	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	P	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		32 bits	Row	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]*	P19	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]
			Column	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	P	P11	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
01	9 bits	8 bits	Row	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	A[19]*	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]
			Column	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	P	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		16 bits	Row	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]*	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]
			Column	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	P	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		32 bits	Row	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]*	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]
			Column	—	A[26]	A[25]	A[24]	A[23]	A[22]	P	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
10	10 bits	8 bits	Row	—	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]*	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]
			Column	—	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	P	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		16 bits	Row	—	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]*	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]
			Column	—	—	A[26]	A[25]	A[24]	A[23]	A[22]	P	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		32 bits	Row	—	—	A[26]	A[25]	A[24]	A[23]	A[22]*	A[21]	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]	A[10]
			Column	—	—	A[26]	A[25]	A[24]	A[23]	P	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
11	11 bits	8 bits	Row	—	—	—	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]*	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]
			Column	—	—	—	A[26]	A[25]	A[24]	A[23]	A[10]	P	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		16 bits	Row	—	—	—	A[26]	A[25]	A[24]	A[23]	A[22]*	A[21]	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]
			Column	—	—	—	A[26]	A[25]	A[24]	A[11]	P	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
		32 bits	Row	—	—	—	A[26]	A[25]	A[24]	A[23]*	A[22]	A[21]	A[20]	A[19]	A[18]	A[17]	A[16]	A[15]	A[14]	A[13]	A[12]	A[11]
			Column	—	—	—	A[26]	A[25]	A[12]	P	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]

P: Precharge setting command (Precharge-sel) is output.
*: When the PALL command is issued, Precharge-sel = 1 (High) is output. When the Active command is issued, the corresponding address is output.
—: Don't care

16.6.13 Examples for Connecting with SDRAMs

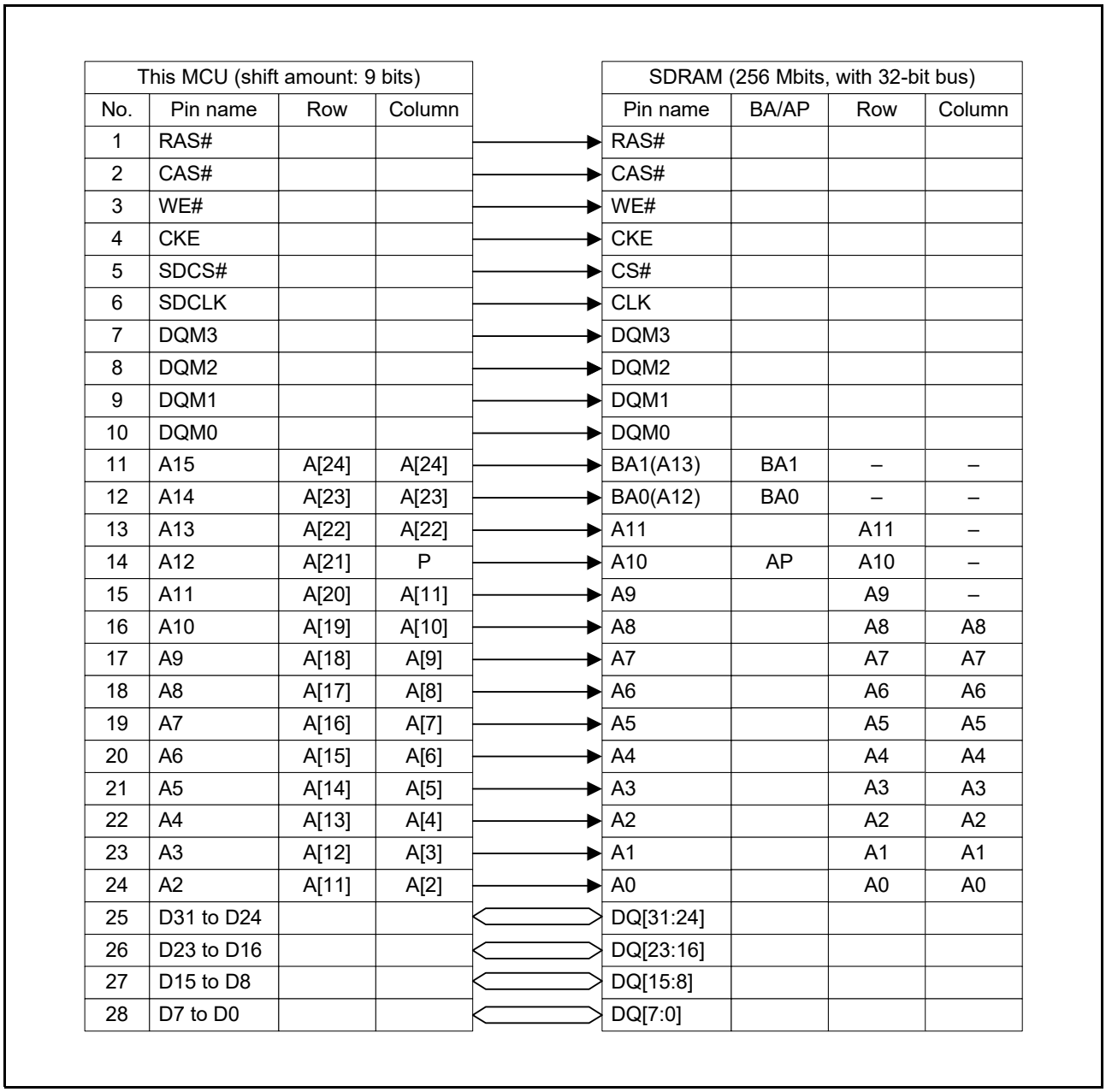
16.6.13.1 32-Bit Bus Space

Figure 16.79 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 10-bit column address and 16-bit bus.

This MCU (shift amount: 10 bits)					SDRAM (512 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS#			→	RAS#			
2	CAS#			→	CAS#			
3	WE#			→	WE#			
4	CKE			→	CKE			
5	SDCS#			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A16	A[26]	A[26]	→	BA1(A14)	BA1	-	-
10	A15	A[25]	A[25]	→	BA0(A13)	BA0	-	-
11	A14	A[24]	A[24]	→	A12		A12	-
12	A13	A[23]	A[23]	→	A11		A11	-
13	A12	A[22]	P	→	A10	AP	A10	-
14	A11	A[21]	A[11]	→	A9		A9	A9
15	A10	A[20]	A[10]	→	A8		A8	A8
16	A9	A[19]	A[9]	→	A7		A7	A7
17	A8	A[18]	A[8]	→	A6		A6	A6
18	A7	A[17]	A[7]	→	A5		A5	A5
19	A6	A[16]	A[6]	→	A4		A4	A4
20	A5	A[15]	A[5]	→	A3		A3	A3
21	A4	A[14]	A[4]	→	A2		A2	A2
22	A3	A[13]	A[3]	→	A1		A1	A1
23	A2	A[12]	A[2]	→	A0		A0	A0
24	D15 to D8			⇔	DQ[15:8]			
25	D7 to D0			⇔	DQ[7:0]			
No.					SDRAM (512 Mbits, with 16-bit bus)			
(1)	RAS#			→	RAS#			
(2)	CAS#			→	CAS#			
(3)	WE#			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS#			→	CS#			
(6)	SDCLK			→	CLK			
26	DQM3			→	UDQM			
27	DQM2			→	LDQM			
(9)	A16	A[26]	A[26]	→	BA1(A14)	BA1	-	-
(10)	A15	A[25]	A[25]	→	BA0(A13)	BA0	-	-
(11)	A14	A[24]	A[24]	→	A12		A12	-
(12)	A13	A[23]	A[23]	→	A11		A11	-
(13)	A12	A[22]	P	→	A10	AP	A10	-
(14)	A11	A[21]	A[11]	→	A9		A9	A9
(15)	A10	A[20]	A[10]	→	A8		A8	A8
(16)	A9	A[19]	A[9]	→	A7		A7	A7
(17)	A8	A[18]	A[8]	→	A6		A6	A6
(18)	A7	A[17]	A[7]	→	A5		A5	A5
(19)	A6	A[16]	A[6]	→	A4		A4	A4
(20)	A5	A[15]	A[5]	→	A3		A3	A3
(21)	A4	A[14]	A[4]	→	A2		A2	A2
(22)	A3	A[13]	A[3]	→	A1		A1	A1
(23)	A2	A[12]	A[2]	→	A0		A0	A0
28	D31 to D24			⇔	DQ[15:8]			
29	D23 to D16			⇔	DQ[7:0]			

Figure 16.79 SDRAM Connection Example (512-Mbit × 2, with 16-Bit Bus)

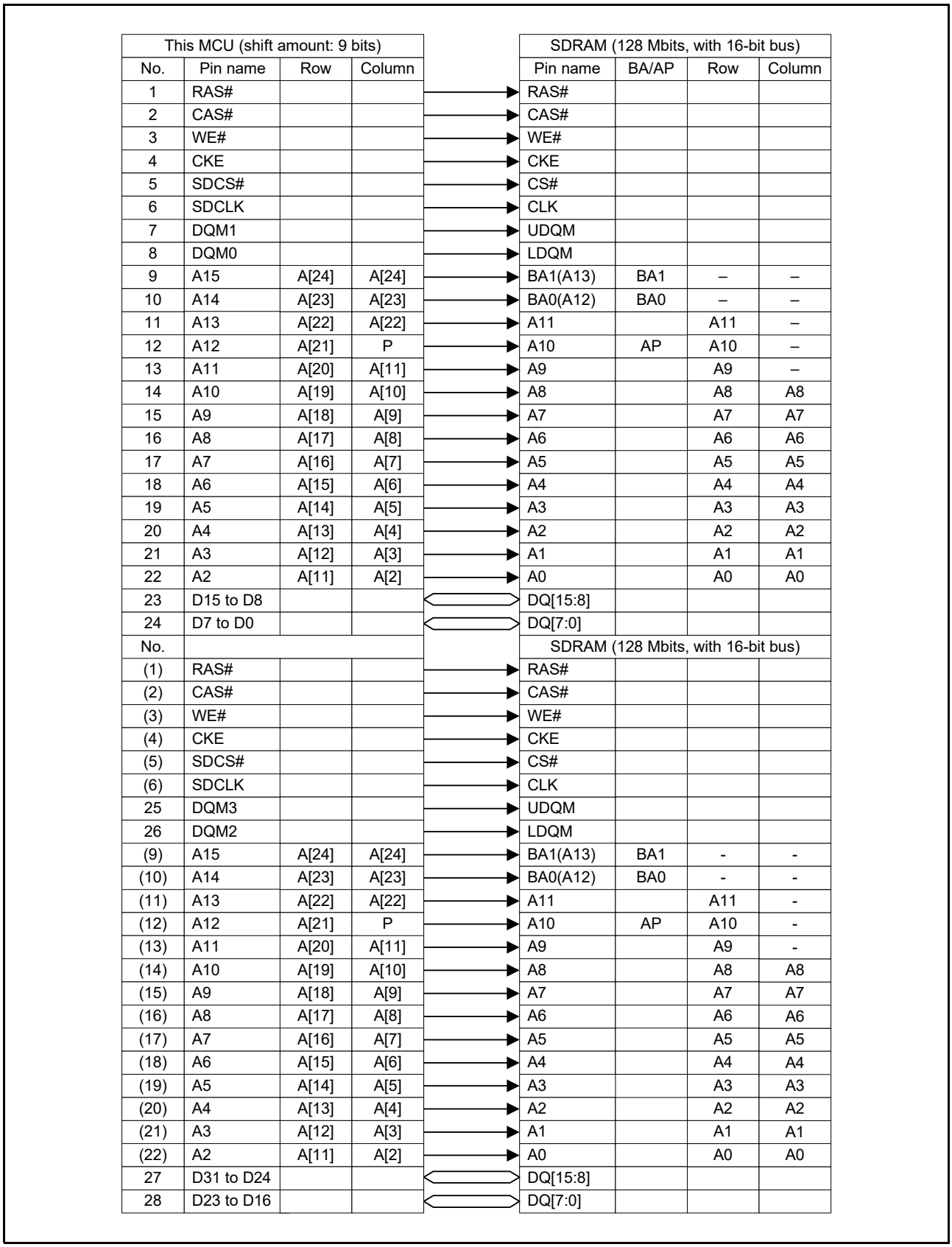
Figure 16.80 shows an example for connecting to a 256-Mbit SDRAM with 12-bit row address, 9-bit column address and 32-bit bus.



This MCU (shift amount: 9 bits)				SDRAM (256 Mbits, with 32-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM3			DQM3			
8	DQM2			DQM2			
9	DQM1			DQM1			
10	DQM0			DQM0			
11	A15	A[24]	A[24]	BA1(A13)	BA1	–	–
12	A14	A[23]	A[23]	BA0(A12)	BA0	–	–
13	A13	A[22]	A[22]	A11		A11	–
14	A12	A[21]	P	A10	AP	A10	–
15	A11	A[20]	A[11]	A9		A9	–
16	A10	A[19]	A[10]	A8		A8	A8
17	A9	A[18]	A[9]	A7		A7	A7
18	A8	A[17]	A[8]	A6		A6	A6
19	A7	A[16]	A[7]	A5		A5	A5
20	A6	A[15]	A[6]	A4		A4	A4
21	A5	A[14]	A[5]	A3		A3	A3
22	A4	A[13]	A[4]	A2		A2	A2
23	A3	A[12]	A[3]	A1		A1	A1
24	A2	A[11]	A[2]	A0		A0	A0
25	D31 to D24			DQ[31:24]			
26	D23 to D16			DQ[23:16]			
27	D15 to D8			DQ[15:8]			
28	D7 to D0			DQ[7:0]			

Figure 16.80 SDRAM Connection Example (256-Mbit × 1, with 32-Bit Bus)

Figure 16.81 shows an example for connecting to two 128-Mbit SDRAMs with 12-bit row address, 9-bit column address and 16-bit bus.

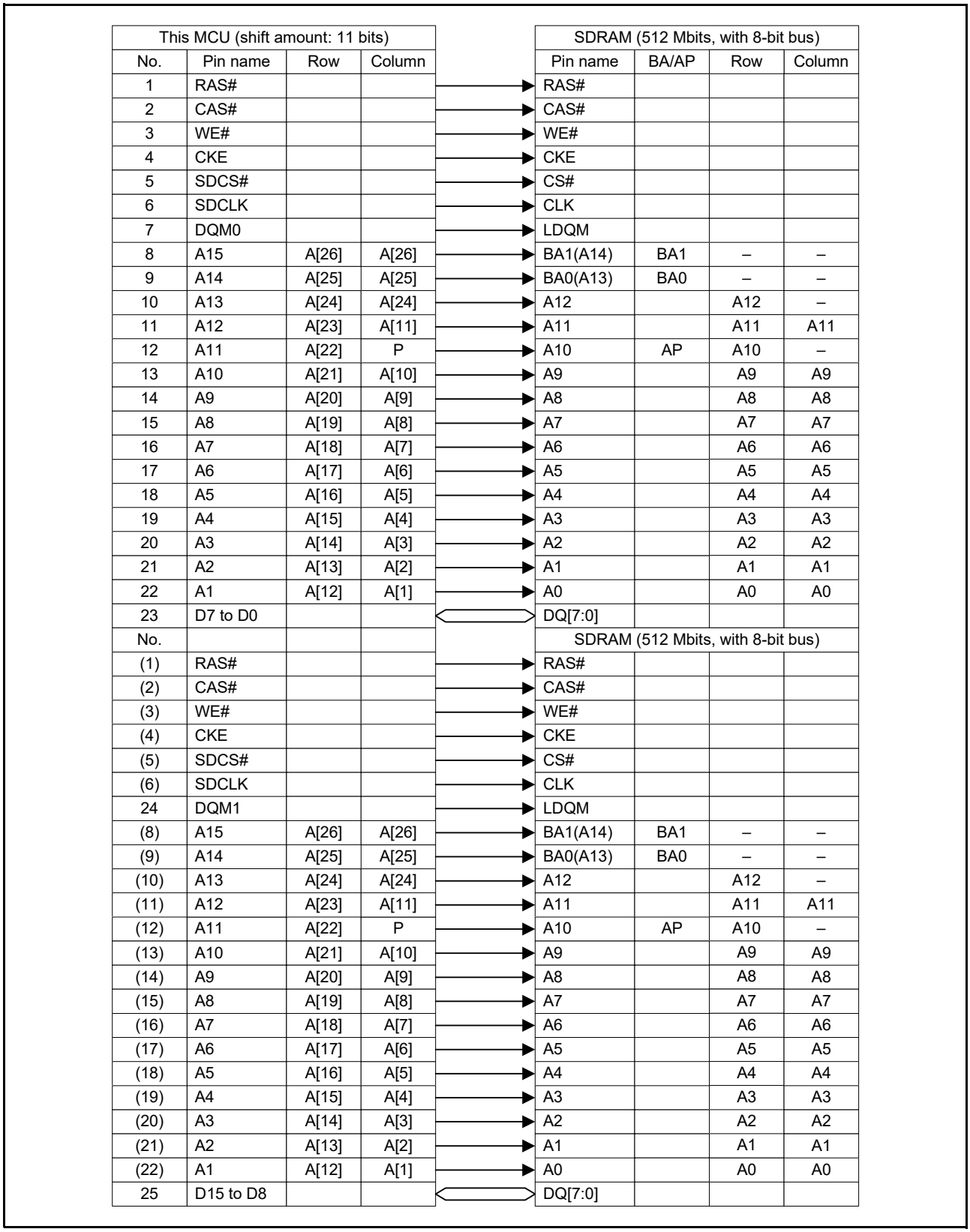


This MCU (shift amount: 9 bits)				SDRAM (128 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A[24]	A[24]	BA1(A13)	BA1	–	–
10	A14	A[23]	A[23]	BA0(A12)	BA0	–	–
11	A13	A[22]	A[22]	A11		A11	–
12	A12	A[21]	P	A10	AP	A10	–
13	A11	A[20]	A[11]	A9		A9	–
14	A10	A[19]	A[10]	A8		A8	A8
15	A9	A[18]	A[9]	A7		A7	A7
16	A8	A[17]	A[8]	A6		A6	A6
17	A7	A[16]	A[7]	A5		A5	A5
18	A6	A[15]	A[6]	A4		A4	A4
19	A5	A[14]	A[5]	A3		A3	A3
20	A4	A[13]	A[4]	A2		A2	A2
21	A3	A[12]	A[3]	A1		A1	A1
22	A2	A[11]	A[2]	A0		A0	A0
23	D15 to D8			DQ[15:8]			
24	D7 to D0			DQ[7:0]			
No.				SDRAM (128 Mbits, with 16-bit bus)			
(1)	RAS#			RAS#			
(2)	CAS#			CAS#			
(3)	WE#			WE#			
(4)	CKE			CKE			
(5)	SDCS#			CS#			
(6)	SDCLK			CLK			
25	DQM3			UDQM			
26	DQM2			LDQM			
(9)	A15	A[24]	A[24]	BA1(A13)	BA1	-	-
(10)	A14	A[23]	A[23]	BA0(A12)	BA0	-	-
(11)	A13	A[22]	A[22]	A11		A11	-
(12)	A12	A[21]	P	A10	AP	A10	-
(13)	A11	A[20]	A[11]	A9		A9	-
(14)	A10	A[19]	A[10]	A8		A8	A8
(15)	A9	A[18]	A[9]	A7		A7	A7
(16)	A8	A[17]	A[8]	A6		A6	A6
(17)	A7	A[16]	A[7]	A5		A5	A5
(18)	A6	A[15]	A[6]	A4		A4	A4
(19)	A5	A[14]	A[5]	A3		A3	A3
(20)	A4	A[13]	A[4]	A2		A2	A2
(21)	A3	A[12]	A[3]	A1		A1	A1
(22)	A2	A[11]	A[2]	A0		A0	A0
27	D31 to D24			DQ[15:8]			
28	D23 to D16			DQ[7:0]			

Figure 16.81 SDRAM Connection Example (128-Mbit × 2, with 16-Bit Bus)

16.6.13.2 16-Bit Bus Space

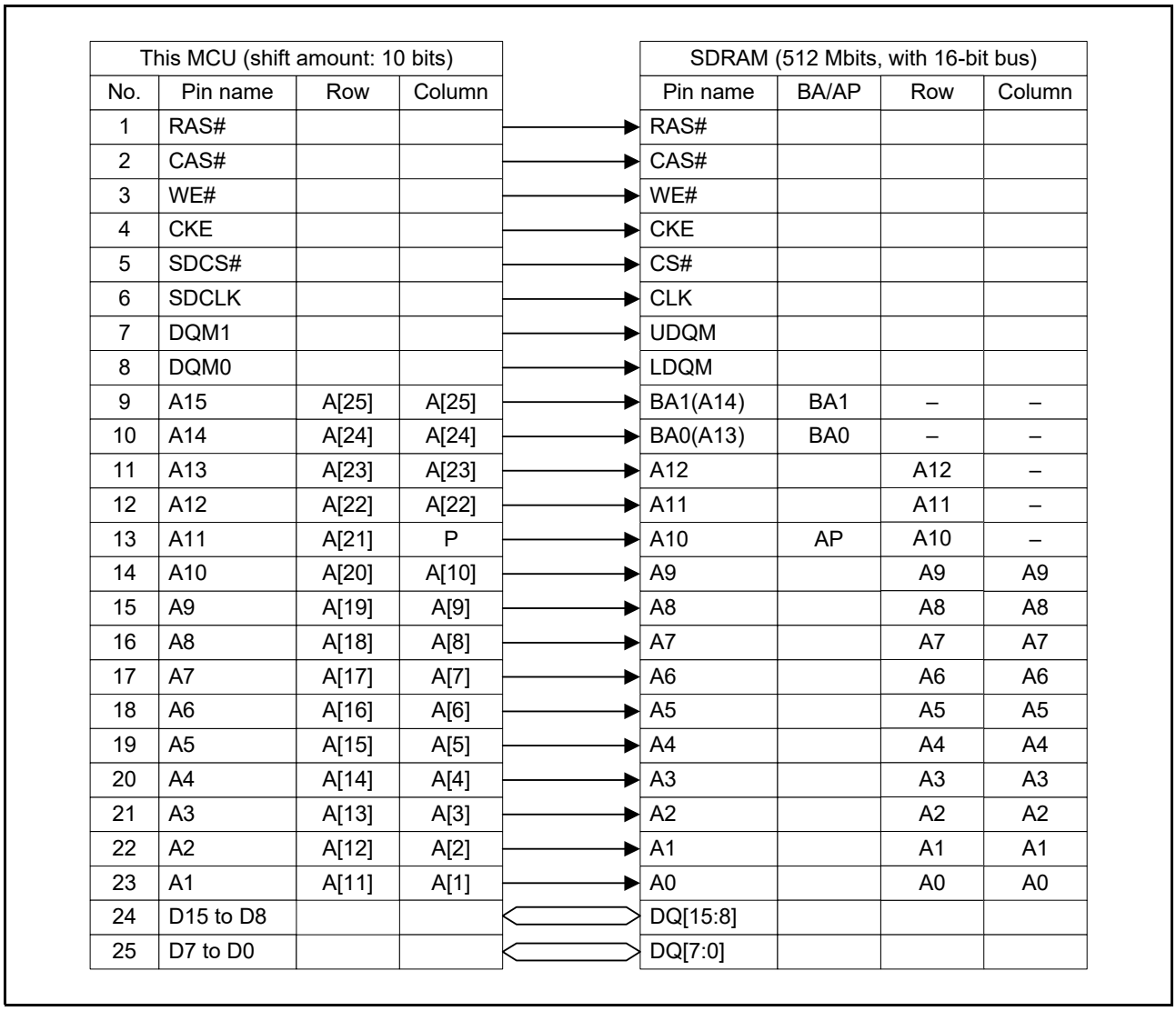
Figure 16.82 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 11-bit column address and 8-bit bus.



This MCU (shift amount: 11 bits)					SDRAM (512 Mbits, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS#			→	RAS#			
2	CAS#			→	CAS#			
3	WE#			→	WE#			
4	CKE			→	CKE			
5	SDCS#			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A[26]	A[26]	→	BA1(A14)	BA1	–	–
9	A14	A[25]	A[25]	→	BA0(A13)	BA0	–	–
10	A13	A[24]	A[24]	→	A12		A12	–
11	A12	A[23]	A[11]	→	A11		A11	A11
12	A11	A[22]	P	→	A10	AP	A10	–
13	A10	A[21]	A[10]	→	A9		A9	A9
14	A9	A[20]	A[9]	→	A8		A8	A8
15	A8	A[19]	A[8]	→	A7		A7	A7
16	A7	A[18]	A[7]	→	A6		A6	A6
17	A6	A[17]	A[6]	→	A5		A5	A5
18	A5	A[16]	A[5]	→	A4		A4	A4
19	A4	A[15]	A[4]	→	A3		A3	A3
20	A3	A[14]	A[3]	→	A2		A2	A2
21	A2	A[13]	A[2]	→	A1		A1	A1
22	A1	A[12]	A[1]	→	A0		A0	A0
23	D7 to D0			⟷	DQ[7:0]			
No.					SDRAM (512 Mbits, with 8-bit bus)			
(1)	RAS#			→	RAS#			
(2)	CAS#			→	CAS#			
(3)	WE#			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS#			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A[26]	A[26]	→	BA1(A14)	BA1	–	–
(9)	A14	A[25]	A[25]	→	BA0(A13)	BA0	–	–
(10)	A13	A[24]	A[24]	→	A12		A12	–
(11)	A12	A[23]	A[11]	→	A11		A11	A11
(12)	A11	A[22]	P	→	A10	AP	A10	–
(13)	A10	A[21]	A[10]	→	A9		A9	A9
(14)	A9	A[20]	A[9]	→	A8		A8	A8
(15)	A8	A[19]	A[8]	→	A7		A7	A7
(16)	A7	A[18]	A[7]	→	A6		A6	A6
(17)	A6	A[17]	A[6]	→	A5		A5	A5
(18)	A5	A[16]	A[5]	→	A4		A4	A4
(19)	A4	A[15]	A[4]	→	A3		A3	A3
(20)	A3	A[14]	A[3]	→	A2		A2	A2
(21)	A2	A[13]	A[2]	→	A1		A1	A1
(22)	A1	A[12]	A[1]	→	A0		A0	A0
25	D15 to D8			⟷	DQ[7:0]			

Figure 16.82 SDRAM Connection Example (512-Mbit × 2, with 8-Bit Bus)

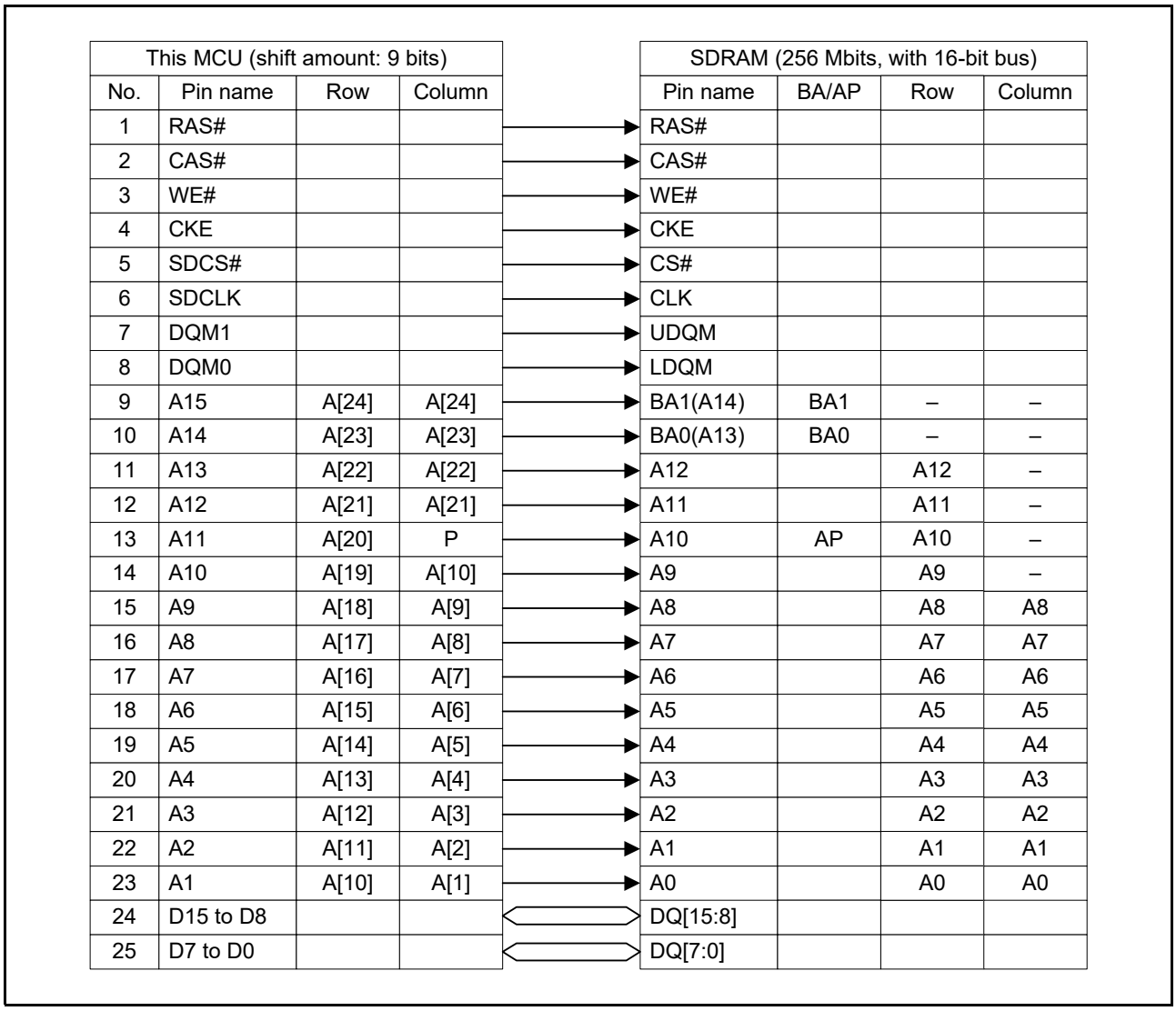
Figure 16.83 shows an example for connecting to a 512-Mbit SDRAM with 13-bit row address, 10-bit column address and 16-bit bus.



This MCU (shift amount: 10 bits)				SDRAM (512 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A[25]	A[25]	BA1(A14)	BA1	–	–
10	A14	A[24]	A[24]	BA0(A13)	BA0	–	–
11	A13	A[23]	A[23]	A12		A12	–
12	A12	A[22]	A[22]	A11		A11	–
13	A11	A[21]	P	A10	AP	A10	–
14	A10	A[20]	A[10]	A9		A9	A9
15	A9	A[19]	A[9]	A8		A8	A8
16	A8	A[18]	A[8]	A7		A7	A7
17	A7	A[17]	A[7]	A6		A6	A6
18	A6	A[16]	A[6]	A5		A5	A5
19	A5	A[15]	A[5]	A4		A4	A4
20	A4	A[14]	A[4]	A3		A3	A3
21	A3	A[13]	A[3]	A2		A2	A2
22	A2	A[12]	A[2]	A1		A1	A1
23	A1	A[11]	A[1]	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.83 SDRAM Connection Example (512-Mbit × 1, with 16-Bit Bus)

Figure 16.84 shows an example for connecting to a 256-Mbit SDRAM with 13-bit row address, 9-bit column address and 16-bit bus.



This MCU (shift amount: 9 bits)				SDRAM (256 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A[24]	A[24]	BA1(A14)	BA1	–	–
10	A14	A[23]	A[23]	BA0(A13)	BA0	–	–
11	A13	A[22]	A[22]	A12		A12	–
12	A12	A[21]	A[21]	A11		A11	–
13	A11	A[20]	P	A10	AP	A10	–
14	A10	A[19]	A[10]	A9		A9	–
15	A9	A[18]	A[9]	A8		A8	A8
16	A8	A[17]	A[8]	A7		A7	A7
17	A7	A[16]	A[7]	A6		A6	A6
18	A6	A[15]	A[6]	A5		A5	A5
19	A5	A[14]	A[5]	A4		A4	A4
20	A4	A[13]	A[4]	A3		A3	A3
21	A3	A[12]	A[3]	A2		A2	A2
22	A2	A[11]	A[2]	A1		A1	A1
23	A1	A[10]	A[1]	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.84 SDRAM Connection Example (256-Mbit × 1, with 16-Bit Bus)

16.6.14 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans multiple areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single 16-bit or 32-bit access.

(2) Low Power Consumption State

In software standby mode and deep software standby mode, auto-refresh operation is not available since the clock supply to SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transition to and recovery from self-refresh mode, refer to section 16.6.6, Self-Refresh.

(3) Setting the SDRAM Timing Register

Set the RAI[2:0] bits in the SDRAM timing register (SDTR) to a value less than or equal to the sum of the row column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(4) Restriction on the Placement of Instruction Codes

Do not allocate an instruction code to an area with an endian different from that set for the chip. When allocating an instruction code to an external space, be sure to allocate it to an area with the same endian as that set for the chip.
For details, refer to section 16.2.5.1, Restriction on Endian.

(5) Consecutive-Access Mode

When the SDAMOD.BE bit is set to 1 (continuous access mode), setting the SDTR.CL[2:0] bits to 1 is prohibited. If set, operation is not guaranteed.

16.7 External Bus Hold

When the EXHLDEN.HLDEN bit is 1, the external bus can be held by the BREQ# pin.
When the EXHLDEN.HLDEN bit is 1 and the BREQ# pin goes low, the external bus controller puts the external bus into a hold state. When the transition to the external bus hold state is complete, the BACK# pin goes low to notify that the external bus has been released.
When the BREQ# pin goes high, the external bus controller puts the BACK# pin high to release the bus hold state.

Note: The SDRAM cannot be refreshed during an external bus hold state. When using an external bus hold, take into consideration the SDRAM refresh interval.
Note: After the BREQ# pin goes low, do not set the BREQ# pin to high before the BACK# pin goes low.
Note: After the BREQ# pin goes high, do not set the BREQ# pin to low before the BACK# pin goes high.
Note: Set the EXHLDEN.HLDEN bit to 0 before entering the software standby mode or deep software standby mode.

16.8 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

16.8.1 Types of Bus Error

The following types of errors can occur on each bus:

- Master security attribution unit error
- Bus master MPU error
- Illegal address access
- Protect Zone access error
- Bus error transmitted from each slave module

Table 16.29 shows the address ranges that cause an illegal address access error when accessed. The reserved areas in the slave do not cause an illegal address access error. For information about the bus master MPU, refer to section 17, Memory-Protection Unit (MPU).
If an instruction fetch causes an illegal address access error, a Protect Zone access error, or a bus error sent from each slave IP, and the instruction code at the time the error occurred is executed by the CPU, an instruction secure exception occurs in the CPU.

16.8.2 Operations When a Bus Error Occurs

If an access violation occurs due to these bufferable writes, an error response may not be returned to the bus master. If a bufferable write error that does not return an error response is detected, the access is blocked and the flag corresponding to the location where the error occurred in the MBWERRSTAT register is set to 1. In addition, an interrupt request or reset request is generated according to the setting of the BUSOAD.BWERROAD bit. The MBWERRSTAT register is retained until it is cleared by a reset other than a bus error reset or memory error reset or by the MBWERRCLR register, so it can be checked in the interrupt handler or after a reset.
Bufferable write errors may occur in the following cases:

1. CPU0OA accesses an illegal address area.
2. CPU0OA accesses an accessible area and a slave bus error or Protect Zone access error occurs.
3. DMAC0 accesses PABI, PBBI, PIBI, PSBI, PS0BI, PC0BI, LRAM0BI, or LRAM1BI and a slave error or Protect Zone access error occurs.

Table 16.26 Bus Error Notice Method for Bufferable Write Access

Operation Settings after Error Detection	All Masters
BUSOAD.BWERROAD = 0 (Interrupt)	BUSERR.IRQ
BUSOAD.BWERROAD = 1 (Reset)	Reset

If a bus error occurs due to an access other than a bufferable write, the error access is blocked and the error operation determined by the bus master and the operation setting after error detection for that error is performed. Table 16.27 shows the operation setting after error detection corresponding to each error. Table 16.28 shows the notification method when a bus error occurs during non-bufferable write.

- When the operation setting for error detection corresponding to each error is interrupt
  An error response is returned to the bus master, and if BUSINTENx.EN = 1, an interrupt request is generated. To generate an interrupt request, set BUSINTENx.EN to 1. Interrupts due to bus errors (including bufferable errors) can be used as non-maskable interrupts (NMI). For details, refer to the ICU.NMIER register specifications.

- When the operation setting for error detection corresponding to each error is reset
  An error response is returned to the bus master and a reset request is generated.
  x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

Table 16.27 OAD Bit (Operation after Detection Bit) for Each Error

Error Type	Operation after Detection Bit
MSAU error	MSAOAD.OAD
MPU error	MMPUOAD.OAD
Illegal access error	BUSOAD.ILERROAD
Slave bus error	BUSOAD.SLERROAD
Protect Zone access error	BUSOAD.SLERROAD

Table 16.28 Bus Error Notice Method for Non-Bufferable Write Access

Operation Settings after Error Detection	All Masters
BUSOAD.ILERROAD = 0 (Interrupt)	BUSERR.IRQ
BUSOAD.SLERROAD = 0 (Interrupt)	
BUSOAD.ILERROAD = 1 (Reset)	Reset
BUSOAD.SLERROAD = 1 (Reset)	

Note: DMAC/DTC0, EDMAC, EXDMAC, and XSPI0/XSPI1 can issue an interrupt when they receive an error response. This non-maskable interrupt is output to the ICU and sent from the ICU to the CPU.

Figure 16.85 shows the operation setting flow when an error is detected after a reset. Bus masters must be stopped while setting this register.

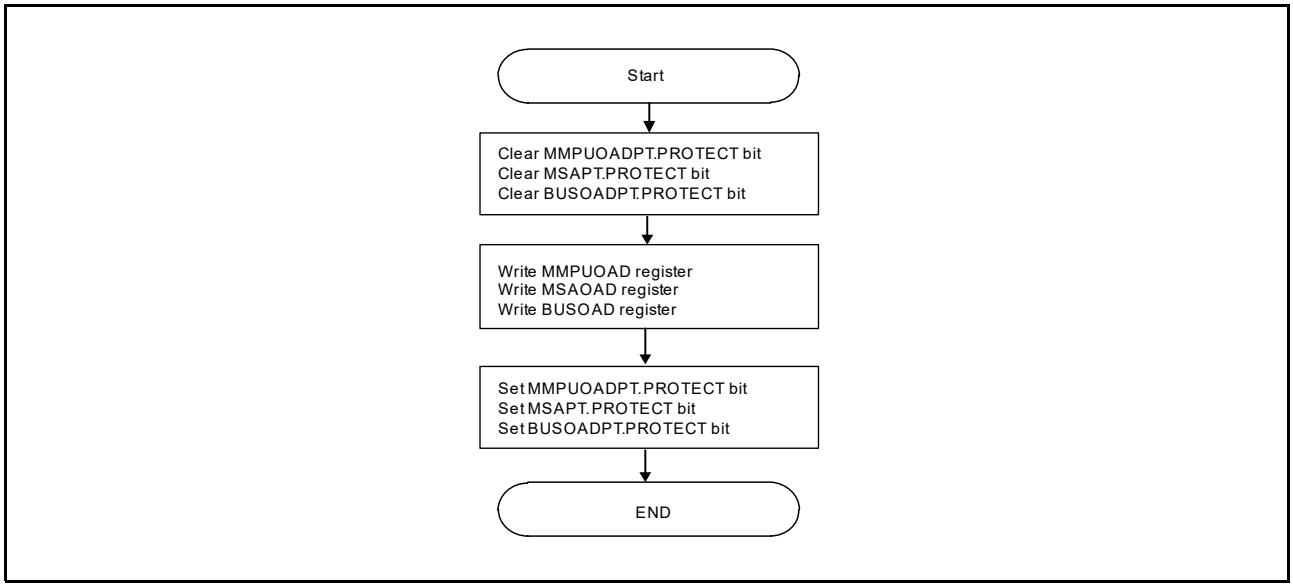


Figure 16.85 Operation Setting Flow when an Error is Detected after a Reset

For details on the MMPUOADPT and MMPUOAD registers, refer to section 17, Memory-Protection Unit (MPU). For details on the MSAPT and MSAOAD registers, refer to section 56, Security Features.

(1) Master Security Attribution Unit Error

When the master security attribute unit (MSAU) detects a master security attribution unit error, it returns an error response to the master that caused the error and the value of the BUSERRSTATx.MSERRSTAT flag corresponding to that master becomes 1. If the value of the BUSINTENx.EN bit is 1, either a bus error interrupt request or reset request is also generated according to the setting of the MSAOAD.OAD bit. The values of the flags in the BUSERRSTATx registers are retained until they are cleared by a reset other than a bus error reset request or common memory error reset request, or by writing to the corresponding bits of the BUSERRCLRx registers. Accordingly, the values of the flags can be checked within the interrupt handler or after a reset. An interrupt request is only generated in response to the first master security attribution unit error to be generated following a reset or clearing of the BUSERRSTATx.MSERRSTAT flag by the corresponding BUSERRCLRx.MSERRCLR bit.
x = CPU0IF, CPU0OA, DMAC0, EXDMAC, EDMAC, XSPI0, XSPI1

(2) Bus Master MPU Error

When the bus master MPU detects a bus master MPU error, it returns an error response to the master that caused the error and the value of the BUSERRSTATx.MMERRSTAT flag corresponding to that master becomes 1. If the value of the BUSINTENx.EN bit is 1, either a bus error interrupt request or reset request is also generated according to the setting of the MMPUOAD.OAD bit. The values of the flags in the BUSERRSTATx registers are retained until they are cleared by a reset other than a bus error reset request or common memory error reset request, or by writing to the corresponding bits of the BUSERRCLRx registers. Accordingly, the values of the flags can be checked within the interrupt handler or after a reset. An interrupt request is only generated in response to the first bus master MPU error to be generated following a reset or clearing of the BUSERRSTATx.MMERRSTAT flag by the corresponding BUSERRCLRx.MMERRCLR bit.

(3) Illegal Address Access Error

Meeting any of the conditions listed in section 16.8.3, Conditions Leading to Illegal Address Access Errors, leads to an illegal address access error, for which an error response is returned to the master that caused the error, and the value of the BUSERRSTATx.ILERRSTAT flag corresponding to that master becomes 1. If the value of the BUSINTENx.EN bit is 1, either a bus error interrupt request or reset request is also generated according to the setting of the BUSOAD.ILERROAD bit. The values of the flags in the BUSERRSTATx registers are retained until they are cleared by a reset other than a bus error reset request or common memory error reset request, or by writing to the corresponding bits of the BUSERRCLRx registers. Accordingly, the values of the flags can be checked within the interrupt handler or after a reset. An interrupt request is only generated in response to the first illegal address access error to be generated following a reset or clearing of the BUSERRSTATx.ILERRSTAT flag by the corresponding BUSERRCLRx.ILERRCLR bit.

(4) Protect Zone Access Error

When a Protect Zone access error is detected, an error response is returned to the master that caused the error and the value of the BUSERRSTATx.SLERRSTAT flag corresponding to that master becomes 1. If the value of the BUSINTENx.EN bit is 1, either a bus error interrupt request or reset request is also generated according to the setting of the BUSOAD.SLERROAD bit. The values of the flags in the BUSERRSTATx registers are retained until they are cleared by a reset other than a bus error reset request or common memory error reset request, or by writing to the corresponding bits of the BUSERRCLRx registers. Accordingly, the values of the flags can be checked within the interrupt handler or after a reset. An interrupt request is only generated in response to the first slave error or Protect Zone access error to be generated following a reset or clearing of the BUSERRSTATx.SLERRSTAT flag by the corresponding BUSERRCLRx.SLERRCLR bit.

(5) Slave Bus Error

When a bus error is detected in the destination module for bus access, an error response is returned to the master that caused the error and the value of the BUSERRSTATx.SLERRSTAT flag corresponding to that master becomes 1. If the value of the BUSINTENx.EN bit is 1, either a bus error interrupt request or reset request is also generated according to the setting of the BUSOAD.SLERROAD bit. The values of the flags in the BUSERRSTATx registers are retained until they are cleared by a reset other than a bus error reset request or common memory error reset request, or by writing to the corresponding bits of the BUSERRCLRx registers. Accordingly, the values of the flags can be checked within the interrupt handler or after a reset. An interrupt request is only generated in response to the first slave error or Protect Zone access error to be generated following a reset or clearing of the BUSERRSTATx.SLERRSTAT flag by the corresponding BUSERRCLRx.SLERRCLR bit.

16.8.3 Conditions Leading to Illegal Address Access Errors

Table 16.29 lists illegal address access error by bus access occurs in the address space.

Table 16.29 Conditions Leading to Illegal Address Access Errors (1/3)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *6	CPU0OA *6	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
0000 0000h	0001 FFFFh	LRAM0 (for CPU access only)	—	—	✓	✓	✓	✓
0002 0000h	0003 FFFFh	LRAM1 (for CPU access only)	—	—	✓	✓	✓	✓
0004 0000h	000F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0010 0000h	00FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0100 0000h	0101 0FFFh	PC0BI	—	—	—	✓	✓	✓
0101 1000h	01FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0200 0000h	020F FFFFh	SRAM0	—	—	—	—	—	—
0210 0000h	0217 FFFFh	SRAM1	—	—	—	—	—	—
0218 0000h	021F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0220 0000h	027F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0280 0000h	0281 FFFFh	LRAM0 (for bus master access other than CPU)	✓	✓	—	✓	✓	✓
0282 0000h	0283 FFFFh	LRAM1 (for bus master access other than CPU)	✓	✓	—	✓	✓	✓
0284 0000h	028F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0290 0000h	02EF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
02F0 0000h	02F0 1FFFh	Reserved	✓	✓	✓	✓	✓	✓
02F0 2000h	02F0 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
02F1 0000h	02FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0300 0000h	05FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0600 0000h	068F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0690 0000h	0690 D7FFh	MRAM (MRE)	—	—	—	✓	—	✓
0690 D800h	069F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
06A0 0000h	06AF FFFFh	Reserved	—	—	—	✓	—	✓
06B0 0000h	06CF FFFFh	MRAM (MRE)*1	—	—	—	✓	—	✓
06D0 0000h	06DF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
06E0 0000h	06FF FFFFh	MRAM (MRE)*1	—	—	—	✓	—	✓
0700 0000h	0700 3FFFh	Reserved	✓	✓	✓	✓	✓	✓
0700 4000h	0700 5FFFh	Reserved	✓	✓	✓	✓	✓	✓
0700 6000h	0700 6FFFh	PS0BI	✓	—	—	✓	✓	✓
0700 7000h	0700 7FFFh	PS0BI	✓	—	—	✓	✓	✓
0700 8000h	0700 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0701 0000h	0701 1FFFh	PSBI	✓	—	—	✓	✓	✓
0701 2000h	0701 7FFFh	Reserved	✓	✓	✓	✓	✓	✓
0701 8000h	0701 8FFFh	PSBI	✓	—	—	✓	✓	✓
0701 9000h	0701 9FFFh	PSBI	✓	—	—	✓	✓	✓
0701 A000h	0701 AFFFh	PSBI	✓	—	—	✓	✓	✓
0701 B000h	0701 BFFFh	Reserved	✓	✓	✓	✓	✓	✓
0701 C000h	0701 CFFFh	PSBI	✓	—	—	✓	✓	✓
0701 D000h	0701 DFFFh	PSBI	✓	—	—	✓	✓	✓
0701 E000h	0701 EFFFh	PSBI	✓	—	—	✓	✓	✓

Table 16.29 Conditions Leading to Illegal Address Access Errors (2/3)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *6	CPU0OA *6	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
0701 F000h	0701 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0702 0000h	0703 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0704 0000h	0707 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0708 0000h	0709 FFFFh	PSBI	✓	—	—	✓	✓	✓
070A 0000h	070F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0710 0000h	073F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
0740 0000h	077F FFFFh	Peripheral I/O Registers (PABI)	✓	—	—	✓	✓	✓
0780 0000h	07BF FFFFh	Peripheral I/O Registers (PBBI)	✓	—	—	✓	✓	✓
07C0 0000h	07FF FFFFh	Peripheral I/O Registers (PIBI)	✓	—	—	✓	✓	✓
0800 0000h	1FFF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
2000 0000h	3FFF FFFFh	External address area (XSPI0)	—	—	—	—	—	✓
4000 0000h	5FFF FFFFh	External address area (XSPI1)	—	—	—	—	—	✓
6000 0000h	6FFF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
7000 0000h	77FF FFFFh	External address area (SDRAM)	—	—	—	—	—	✓
7800 0000h	78FF FFFFh	External address area (CS7)	—	—	—	—	—	✓
7900 0000h	79FF FFFFh	External address area (CS6)	—	—	—	—	—	✓
7A00 0000h	7AFF FFFFh	External address area (CS5)	—	—	—	—	—	✓
7B00 0000h	7BFF FFFFh	External address area (CS4)	—	—	—	—	—	✓
7C00 0000h	7CFF FFFFh	External address area (CS3)	—	—	—	—	—	✓
7D00 0000h	7DFF FFFFh	External address area (CS2)	—	—	—	—	—	✓
7E00 0000h	7EFF FFFFh	External address area (CS1)	—	—	—	—	—	✓
7F00 0000h	7FFF FFFFh	MRAM (MRC)*3, *5	—	—	—	✓	—	✓
8000 0000h	8001 FFFFh	LRAM0 (for CPU access only)	—	—	✓	✓	✓	✓
8002 0000h	8003 FFFFh	LRAM1 (for CPU access only)	—	—	✓	✓	✓	✓
8004 0000h	800F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8010 0000h	80FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8100 0000h	8101 0FFFh	PC0BI	—	—	—	✓	✓	✓
8101 0000h	81FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8200 0000h	820F FFFFh	SRAM0	—	—	—	—	—	—
8210 0000h	8217 FFFFh	SRAM1	—	—	—	—	—	—
8218 0000h	821F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8220 0000h	827F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8280 0000h	8281 FFFFh	LRAM0 (for bus master access other than CPU)	✓	✓	—	✓	✓	✓
8282 0000h	8283 FFFFh	LRAM1 (for bus master access other than CPU)	✓	✓	—	✓	✓	✓
8284 0000h	828F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8290 0000h	82EF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
82F0 0000h	82F0 1FFFh	Reserved	✓	✓	✓	✓	✓	✓
82F0 2000h	82F0 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
82F1 0000h	82FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8300 0000h	85FF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8600 0000h	868F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8690 0000h	8690 D7FFh	MRAM (MRE)	—	—	—	✓	—	✓

Table 16.29 Conditions Leading to Illegal Address Access Errors (3/3)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *6	CPU0OA *6	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
8690 D800h	869F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
86A0 0000h	86AF FFFFh	Reserved	—	—	—	✓	—	✓
86B0 0000h	86CF FFFFh	MRAM (MRE)*1	—	—	—	✓	—	✓
86D0 0000h	86D7 FFFFh	MRAM (MRE)*1, *2	—	—	—	✓	—	✓
86D8 0000h	86FF FFFFh	MRAM (MRE)*1	—	—	—	✓	—	✓
8700 0000h	8700 3FFFh	Reserved	✓	✓	✓	✓	✓	✓
8700 4000h	8700 5FFFh	Reserved	✓	✓	✓	✓	✓	✓
8700 6000h	8700 6FFFh	PS0BI	✓	—	—	✓	✓	✓
8700 7000h	8700 7FFFh	PS0BI	✓	—	—	✓	✓	✓
8700 8000h	8700 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8701 0000h	8701 1FFFh	PSBI	✓	—	—	✓	✓	✓
8701 2000h	8701 7FFFh	Reserved	✓	✓	✓	✓	✓	✓
8701 8000h	8701 8FFFh	PSBI	✓	—	—	✓	✓	✓
8701 9000h	8701 9FFFh	PSBI	✓	—	—	✓	✓	✓
8701 A000h	8701 AFFFh	PSBI	✓	—	—	✓	✓	✓
8701 B000h	8701 BFFFh	Reserved	✓	✓	✓	✓	✓	✓
8701 C000h	8701 CFFFh	PSBI	✓	—	—	✓	✓	✓
8701 D000h	8701 DFFFh	PSBI	✓	—	—	✓	✓	✓
8701 E000h	8701 EFFFh	PSBI	✓	—	—	✓	✓	✓
8701 F000h	8701 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8702 0000h	8703 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8704 0000h	8707 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8708 0000h	8709 FFFFh	PSBI	✓	—	—	✓	✓	✓
870A 0000h	870F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8710 0000h	873F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
8740 0000h	877F FFFFh	Peripheral I/O Registers (PABI)	✓	—	—	✓	✓	✓
8780 0000h	87BF FFFFh	Peripheral I/O Registers (PBBI)	✓	—	—	✓	✓	✓
87C0 0000h	87FF FFFFh	Peripheral I/O Registers (PIBI)	✓	—	—	✓	✓	✓
8800 0000h	8FFF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
9000 0000h	EFFF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
F000 0000h	FEFF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF00 0000h	FFFF FFFFh	MRAM (MRC)*4, *5	—	—	—	✓	—	✓

✓: A bus error has occurred.

—: A bus error has not occurred. Even if there is a reserved area, a bus error has not occurred.

Note 1. Only read access is possible.

Note 2. Access is possible only while the first stage boot loader is running.

Note 3. When the built-in code MRAM is disabled, the CS0 area is allocated. The access conditions are the same as for the CS1 area.

Note 4. When the built-in code MRAM is disabled, the MRAM (MRC0/MRC1) area is not allocated. It becomes a reserved area.

Note 5. For the address map of MRAM (MRC), see Table 16.30 to Table 16.33.

Note 6. When CPU0IF and CPU0OA cause an illegal address access error, it may be set as a slave bus error in the bus error status register.

Table 16.30 Conditions Leading to Illegal Address Access Errors (Linear Mode, 1 MByte)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *1	CPU0OA *1	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
FF00 0000h	FF6F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF70 0000h	FF70 1FFFh	Reserved	—	—	—	✓	—	✓
FF70 2000h	FF77 FFFFh	Reserved	—	—	—	✓	—	✓
FF78 0000h	FF78 1FFFh	Reserved	—	—	—	✓	—	✓
FF78 2000h	FF7F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF80 0000h	FFEF FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFF0 0000h	FFF7 FFFFh	MRAM (MRC)	—	—	—	✓	—	✓
FFF8 0000h	FFFF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓

✓: A bus error has occurred.

—: A bus error has not occurred. Even if there is a reserved area, a bus error has not occurred.

Note 1. When CPU0IF and CPU0OA cause an illegal address access error, it may be set as a slave bus error in the bus error status register.

Table 16.31 Conditions Leading to Illegal Address Access Errors (Linear Mode, 512 KByte)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *1	CPU0OA *1	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
FF00 0000h	FF6F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF70 0000h	FF70 1FFFh	Reserved	—	—	—	✓	—	✓
FF70 2000h	FF77 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF78 0000h	FF78 1FFFh	Reserved	—	—	—	✓	—	✓
FF78 2000h	FF7F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF80 0000h	FFF7 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFF8 0000h	FFFB FFFFh	MRAM (MRC)	—	—	—	✓	—	✓
FFFC 0000h	FFFF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓

✓: A bus error has occurred.

—: A bus error has not occurred. Even if there is a reserved area, a bus error has not occurred.

Note 1. When CPU0IF and CPU0OA cause an illegal address access error, it may be set as a slave bus error in the bus error status register.

Table 16.32 Conditions Leading to Illegal Address Access Errors (Dual Mode, 1 MByte)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *1	CPU0OA *1	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
FF00 0000h	FF6F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF70 0000h	FF70 1FFFh	Reserved	—	—	—	✓	—	✓
FF70 2000h	FF77 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF78 0000h	FF78 1FFFh	Reserved	—	—	—	✓	—	✓
FF78 2000h	FF7F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF80 0000h	FFB7 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFB8 0000h	FFBF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓
FFC0 0000h	FFF7 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFF8 0000h	FFFF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓

✓: A bus error has occurred.

—: A bus error has not occurred. Even if there is a reserved area, a bus error has not occurred.

Note 1. When CPU0IF and CPU0OA cause an illegal address access error, it may be set as a slave bus error in the bus error status register.

Table 16.33 Conditions Leading to Illegal Address Access Errors (Dual Mode, 512 KByte)

Address			Master Bus					
Start	End	Slave Bus	CPU0IF *1	CPU0OA *1	DMAC/ DTC0	EXDMAC	EDMAC	XSPI0/ XSPI1
FF00 0000h	FF6F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF70 0000h	FF70 1FFFh	Reserved	—	—	—	✓	—	✓
FF70 2000h	FF77 FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF78 0000h	FF78 1FFFh	Reserved	—	—	—	✓	—	✓
FF78 2000h	FF7F FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FF80 0000h	FFBB FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFBC 0000h	FFBF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓
FFC0 0000h	FFFB FFFFh	Reserved	✓	✓	✓	✓	✓	✓
FFFC 0000h	FFFF FFFFh	MRAM (MRC)	—	—	—	✓	—	✓

✓: A bus error has occurred.

—: A bus error has not occurred. Even if there is a reserved area, a bus error has not occurred.

Note 1. When CPU0IF and CPU0OA cause an illegal address access error, it may be set as a slave bus error in the bus error status register.

16.8.4 Timeout

16.8.4.1 Peripheral Bus

As for some peripheral modules, a timeout error can occur in the module-stop function. When there is no response from the slave for a certain period of time, a timeout error is detected.

16.8.4.2 External Bus Access Wait with WAIT# Signal

When the external bus timeout detection enable bit (EXBEREN.TOEN) is set to 1, it is possible to detect an external bus access wait timeout error caused by the WAIT# signal.
When the EXBEREN.TOEN bit is 1 and the wait time for external bus access caused by the WAIT# signal exceeds 768 BCLK cycles, a timeout error occurs and the EXBERR.TO flag is set to 1.
Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.
When a timeout error occurs, do not write to the CSnCR.EXENB bit or the SDCCR.EXENB bit.

16.8.4.3 External Bus Hold with BREQ# and BACK# Signals

When the external bus timeout detection enable bit (EXBEREN.TOEN) is set to 1, it is possible to detect an external bus hold timeout error caused by the BREQ# and BACK# signals.
When the EXBEREN.TOEN bit is 1 and the external bus hold time caused by the BREQ# and BACK# signals exceeds 768 BCLK cycles, a timeout error occurs and the EXBERR.HLDTO flag is set to 1.
Any access to the external bus from 256 BCLK cycles after an external bus hold timeout error has been detected until the BREQ# signal is negated is treated as a slave error.
When a timeout error occurs, do not write to the CSnCR.EXENB bit or the SDCCR.EXENB bit.

17. Memory-Protection Unit (MPU)

17.1 Overview

This MCU has three memory-protection units: two for the CPU (CPU MPUs) and one for the xSPI bus masters (bus master MPU).
Table 17.1 lists the specifications of the memory-protection units, Table 17.2 lists the memory-protection error detection operations.

Table 17.1 Specifications of Memory Protection (1 / 2)

Specification		Description
CPU MPU	Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection applies in supervisor mode and only the cache attribute settings are effective.
	Number of regions	8
	Page size (smallest unit of protection)	32 bytes
	Specifying addresses of individual regions	Setting the page numbers where regions start and end
	Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
	Access control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
	Cache attribute information settings for individual regions	Set cache attributes for individual regions. Instruction access: Not cached Instruction cached Data access: Not cached Cached (write-through) Cached (write-back)
	Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
	Memory-protection error processing	Generation of access exceptions
	Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
	Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
	Background region access control settings	Access control information can be set for the background region (the whole address space).
	Background region cache settings	Caches can be set for the background region (the whole address space).
	Processing where regions overlap	If region settings overlap for an address and the access control information for these regions is different, the information is handled as the logical OR of each bit setting. As a result, permission is given priority in access control. The information for the cache attribute is also handled in the same way, as the logical OR of the settings.

Table 17.1 Specifications of Memory Protection (2 / 2)

Specification		Description
Bus master MPU	Memory protection (except for the CPU)	Memory protection function for the bus master ・ xSPI0: 4 regions ・ xSPI1: 4 regions
	Protected region	0000 0000h to FFFF FFFFh
	Number of regions	xSPI0: 4 regions; xSPI1: 4 regions
	Specifying addresses of individual regions	The start/end addresses of individual regions can be set.
	Memory protection permission/prohibition setting for individual regions	Setting for permission or prohibition for the relevant region
	Access control settings for individual regions	Permitting writing or reading
	Error detection operation	Reset or error response
	Register protection	Protection against illegal writing

Table 17.2 Memory-Protection Error Detection Operations

Type	Access Permission Setting	Minimum Unit for Boundary Address Setting	Error Response for Error Notification	Bus Access at Error Detection	Whether Error Access Information is Held
CPU MPU	Read access Write access Execution	32 bytes	Supported	Illegal write access Illegal read access	Held
Bus master MPU	Read access Write access	xSPI0/1: 4 Kbytes	Supported	Write access is ignored For read access, the value read is 0	Held

17.2 CPU MPU

17.2.1 Overview of CPU MPU

The RXv3e CPU incorporates two memory-protection units (MPUs), one for secure access and the other for non-secure access, which check the addresses accessed by the CPU over the whole address space (0000 0000h to FFFF FFFFh). The secure MPU checks the addresses for secure access, and the non-secure MPU checks the addresses for non-secure access. For secure access and non-secure access, see section 56, Security Features.

Up to eight regions can be set for each of the secure and non-secure MPUs, and access is permitted or prohibited in accord with the access control information for each region. If an access outside the set regions is detected, a memory-protection error occurs by default.

The access control information for each region corresponds to permitting reading, permitting writing, and permitting execution. This access control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Cache attributes for instruction access and data access are set for each region. The cache attribute settings are effective in supervisor mode as well as in user mode.

The registers of the memory-protection units are only accessible from the CPU. Access to the MPU for the secure world is only possible through the CPU when it is in the secure state. Access to the MPU in the non-secure world is possible through the CPU whether it is in either the non-secure or secure state.

Figure 17.1 is a block diagram of a CPU MPU.

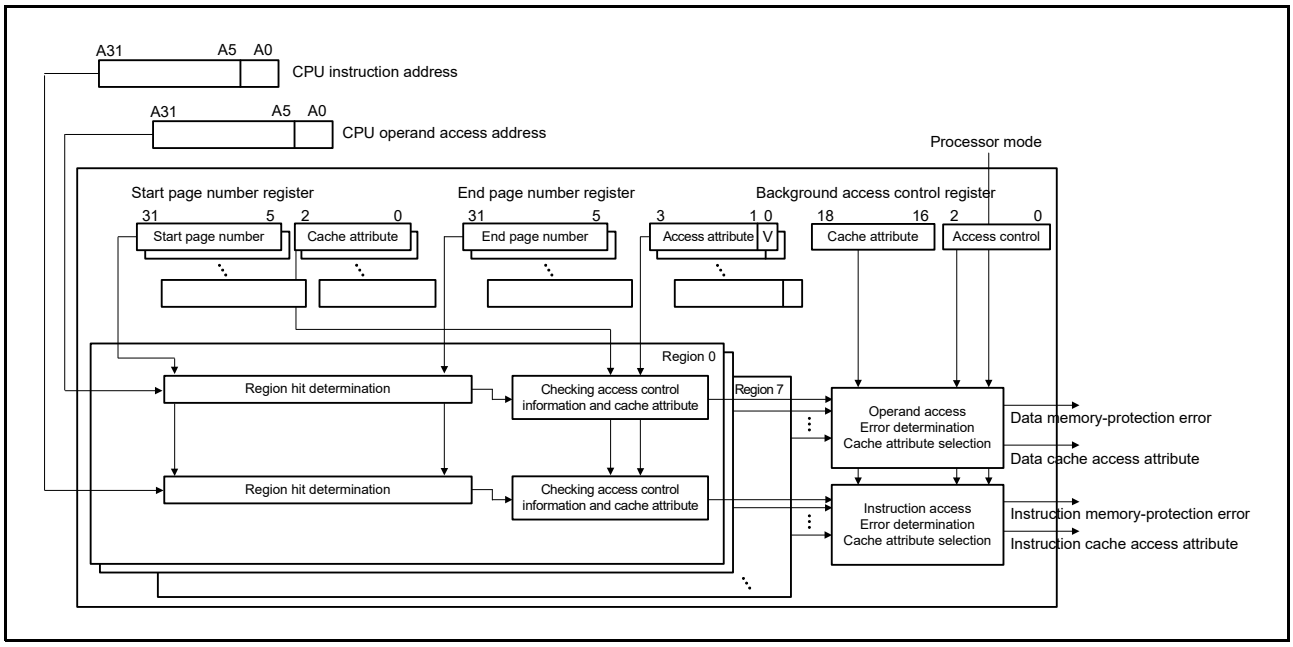


Figure 17.1 Block Diagram of CPU MPU

17.2.1.1 Types of Access Control

There are three types of access control: controlling permission for instruction execution, permission for the reading of operands, and permission for the writing of operands. Detection of a violation of any of these forms of access control only proceeds for a program in user mode. A violation is not detected for a program in supervisor mode.

17.2.1.2 Access Control Regions

Up to eight regions for access control can be defined. The range of each access control region is set by the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn) (n = 0 to 7).
The minimum unit for access control is the “page”, by which the address space is divided into 32-byte units. The 27 higher-order bits ([31:5]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access control information for each region and whether the settings for the region are validated or not.

17.2.1.3 Cache Attribute

For each access control region, the cache attribute can be set as “instruction not cached” or “instruction cached” for instruction access, and as “data not cached”, “data cached (write-through)”, or “data cached (write-back)” for data access.
However, for the IO registers region, “instruction not cached” and “data not cached” are selected regardless of the setting. As for the allocation of caches, read allocation proceeds in the case of “data cached (write-through)”, and write allocation and read allocation proceed in the case of “data cached (write-back)”.

17.2.1.4 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). The access control information for the background region is set by the MPBAC.UBAC bits. Unlike the eight access control regions, the access control information for the background region is effective as long as memory protection is enabled (the MPEN.MPEN bit is 1). The cache settings for the background region are set by using the MPBAC.DCA and MPBAC.ICA bits. These settings are effective even while memory protection is disabled (the MPEN.MPEN bit is 0).

17.2.1.5 Overlap between Regions

In cases of overlap between multiple regions, the access control information becomes the logical OR of the access control bits for the overlapping regions (including the background region), with permission given priority. In the same way as for the access control information, overlapping settings for the cache attribute are also handled as the logical OR of the settings. Accordingly, the order of priority is “cache enabled” > “cache disabled” for instruction access, and “cache enabled (write-back) > “cache enabled (write-through)” > “cache disabled” for data access. In the case of the cache attribute, the background region setting is not included in the logical OR to obtain the applicable setting.

17.2.1.6 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access control settings and cache settings have been made are undefined. Ensure that instructions and data do not span regions for which different access control settings and cache settings have been made.

17.2.2 Register Descriptions of CPU MPU

17.2.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): MPU.RSPAGE0 8100 0800h, MPU.RSPAGE1 8100 0808h, MPU.RSPAGE2 8100 0810h, MPU.RSPAGE3 8100 0818h, MPU.RSPAGE4 8100 0820h, MPU.RSPAGE5 8100 0828h, MPU.RSPAGE6 8100 0830h, MPU.RSPAGE7 8100 0838h, MPU_NS.RSPAGE0 0100 0800h, MPU_NS.RSPAGE1 0100 0808h, MPU_NS.RSPAGE2 0100 0810h, MPU_NS.RSPAGE3 0100 0818h, MPU_NS.RSPAGE4 0100 0820h, MPU_NS.RSPAGE5 0100 0828h, MPU_NS.RSPAGE6 0100 0830h, MPU_NS.RSPAGE7 0100 0838h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[26:11]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[10:0]											—	—	ICA	DCA[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCA[1:0]	Data Cache Attribute Setting	b1 b0 0 0: Data not cached 0 1: Data cached (write-through) 1 0: Setting prohibited 1 1: Data cached (write-back)	R/W
b2	ICA	Instruction Cache Attribute Setting	0: Instruction not cached 1: Instruction cached	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b5	RSPN[26:0]	Region Start Page Number	Region-start page number information for use in region determination	R/W

DCA[1:0] Bits (Data Cache Attribute Setting)

These bits set the data cache attribute. These bits are valid in both user mode and supervisor mode.

ICA Bit (Instruction Cache Attribute Setting)

This bit sets the instruction cache attribute. This bit is valid in both user mode and supervisor mode.

RSPN[26:0] Bits (Region Start Page Number)

These bits set the region-start page number.

17.2.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): MPU.REPAGE0 8100 0804h, MPU.REPAGE1 8100 080Ch, MPU.REPAGE2 8100 0814h, MPU.REPAGE3 8100 081Ch, MPU.REPAGE4 8100 0824h, MPU.REPAGE5 8100 082Ch, MPU.REPAGE6 8100 0834h, MPU.REPAGE7 8100 083Ch, MPU_NS.REPAGE0 0100 0804h, MPU_NS.REPAGE1 0100 080Ch, MPU_NS.REPAGE2 0100 0814h, MPU_NS.REPAGE3 0100 081Ch, MPU_NS.REPAGE4 0100 0824h, MPU_NS.REPAGE5 0100 082Ch, MPU_NS.REPAGE6 0100 0834h, MPU_NS.REPAGE7 0100 083Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[26:11]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[10:0]											—	UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	V	Valid	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b5	REPN[26:0]	Region-End Page Number	Region-end page number information for use in region determination	R/W

V Bit (Valid)

This bit selects whether the corresponding region settings are to be valid or invalid. The setting of this bit determines validation or invalidation of both the memory protection and cache attribute setting functions.
This bit is set to 0 when the region invalidation operation register (MPOPI) invalidates all access control regions.

UAC[2:0] Bits (Access Control in User Mode)

These bits set access control in user mode.

REPN[26:0] Bits (Region-End Page Number)

These bits set the region-end page number. Set a value greater than or equal to the start page number of the corresponding region. The page specified by the region-end page number is also part of the target region for memory protection.

17.2.2.3 Memory-Protection Enable Register (MPEN)

Address(es): MPU.MPEN 8100 0900h,
MPU_NS.MPEN 0100 0900h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPEN	Memory-Protection Enable	0: Memory protection disabled 1: Memory protection enabled	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables memory protection.

After 1 is written to this bit, address checking for memory protection by the CPU starts upon execution of a branch instruction (RTE or RTFI) that shifts operation to user mode.

17.2.2.4 Background Access Control Register (MPBAC)

Address(es): MPU.MPBAC 8100 0904h,
MPU_NS.MPBAC 0100 0904h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	ICA	DCA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UBAC[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	DCA[1:0]	Data Cache Attribute Setting	b17 b16 0 0: Data not cached 0 1: Data cached (write-through) 1 0: Setting prohibited 1 1: Data cached (write-back)	R/W
b18	ICA	Instruction Cache Attribute Setting	0: Instruction not cached 1: Instruction cached	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UBAC[2:0] Bits (Background Access Control in User Mode)

These bits set background access control in user mode.

DCA[1:0] Bits (Data Cache Attribute Setting)

These bits set the background data cache attribute.

The setting of these bits is effective even while memory protection is disabled (MPEN = 0).

ICA Bit (Instruction Cache Attribute Setting)

These bits set the background instruction cache attribute. The setting of these bits is effective even while memory protection is disabled (MPEN = 0).

17.2.2.5 Memory-Protection Error Status Clear Register (MPECLR)

Address(es): MPU.MPECLR 8100 0908h,
MPU_NS.MPECLR 0100 0908h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR	Error Status Clear	[Reading] 0: Fixed for reading [Writing] 0: Nothing is done. 1: The MPESTS.DRW, DMPER, and IMPER bits are set to 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CLR Bit (Error Status Clear)

This bit clears the data read/write bit (DRW), data memory-protection error generation bit (DMPER), and instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

17.2.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): MPU.MPESTS 8100 090Ch,
MPU_NS.MPESTS 0100 090Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DMPER	IMPER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IMPER	Instruction Memory-Protection Error Generation Flag	0: No instruction memory-protection error occurred 1: Instruction memory-protection error occurred	R
b1	DMPER	Data Memory-Protection Error Generation Flag	0: No data memory-protection error occurred 1: Data memory-protection error occurred	R
b2	DRW	Data Read/Write Flag	0: Data read 1: Data write	R
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IMPER Flag (Instruction Memory-Protection Error Generation Flag)

This flag indicates the state of memory-protection error generation by instruction execution.
It is only cleared to 0 by setting the error status clear bit (CLR) of the memory-protection error status clear register (MPECLR) to 1.

DMPER Flag (Data Memory-Protection Error Generation Flag)

This flag indicates the state of memory-protection error generation by operand access.
It is only cleared to 0 by setting the error status clear bit (CLR) of the memory-protection error status clear register (MPECLR) to 1.

DRW Flag (Data Read/Write Flag)

This flag indicates the read/write attribute for operand access that led to a memory-protection error. This flag is only valid while the DMPER bit is 1.
It is only cleared to 0 by setting the error status clear bit (CLR) of the memory-protection error status clear register (MPECLR) to 1.

17.2.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): MPU.MPDEA 8100 0914h,
MPU_NS.MPDEA 0100 0914h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits hold the address where a memory-protection error occurred due to operand access.

17.2.2.8 Region Search Address Register (MPSA)

Address(es): MPU.MPSA 8100 0920h,
MPU_NS.MPSA 0100 0920h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region search	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn) in a region search operation.

17.2.2.9 Region Search Operation Register (MPOPS)

Address(es): MPU.MPOPS 8100 0924h,
MPU_NS.MPOPS 0100 0924h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	S	Regions Search Operation	[Reading] 0: Fixed for reading [Writing] 0: Nothing is done. 1: A region search operation proceeds.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S Bit (Regions Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits in user mode (UHACD[2:0]).

17.2.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): MPU.MPOPI 8100 0926h,
MPU_NS.MPOPI 0100 0926h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INV	Region Invalidation Start	[Reading] 0: Fixed for reading [Writing] 0: Nothing is done. 1: All access control regions are invalidated.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

INV Bit (Region Invalidation Start)

Setting this bit to 1 clears the valid bits (V) in all of the region-n end page number registers (REPAGEn) to 0. After a REPAGEn.V bit is set to 0, all settings other than the access control settings for the background region are invalidated.

17.2.2.11 Instruction-Hit Region Register (MHITI)

Address(es): MPU.MHITI 8100 0928h,
MPU_NS.MHITI 0100 0928h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACI[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than the above b23 0: No instruction memory-protection error in region 7 1: Instruction memory-protection error in region 7 b22 0: No instruction memory-protection error in region 6 1: Instruction memory-protection error in region 6 b21 0: No instruction memory-protection error in region 5 1: Instruction memory-protection error in region 5 b20 0: No instruction memory-protection error in region 4 1: Instruction memory-protection error in region 4 b19 0: No instruction memory-protection error in region 3 1: Instruction memory-protection error in region 3 b18 0: No instruction memory-protection error in region 2 1: Instruction memory-protection error in region 2 b17 0: No instruction memory-protection error in region 1 1: Instruction memory-protection error in region 1 b16 0: No instruction memory-protection error in region 0 1: Instruction memory-protection error in region 0	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control in User Mode)

These bits hold the access control bits in user mode (REPAGEn.UAC[2:0]) for the region where an instruction memory-protection error occurred.

When an error occurred in overlapping regions, the value held here is the logical OR of the access control bits in user mode for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error occurred. They are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

17.2.2.12 Data-Hit Region Register (MHITD)

Address(es): MPU.MHITD 8100 092Ch,
MPU_NS.MHITD 0100 092Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACD[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than the above b23 0: No data memory-protection error or no search hit in region 7 1: A data memory-protection error or a search hit in region 7 b22 0: No data memory-protection error or no search hit in region 6 1: A data memory-protection error or a search hit in region 6 b21 0: No data memory-protection error or no search hit in region 5 1: A data memory-protection error or a search hit in region 5 b20 0: No data memory-protection error or no search hit in region 4 1: A data memory-protection error or a search hit in region 4 b19 0: No data memory-protection error or no search hit in region 3 1: A data memory-protection error or a search hit in region 3 b18 0: No data memory-protection error or no search hit in region 2 1: A data memory-protection error or a search hit in region 2 b17 0: No data memory-protection error or no search hit in region 1 1: A data memory-protection error or a search hit in region 1 b16 0: No data memory-protection error or no search hit in region 0 1: A data memory-protection error or a search hit in region 0	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control in User Mode)

These bits hold the access control bits in user mode (REPAGEn.UAC[2:0]) for the region where a data memory-protection error occurred or the region that was hit in a region search.
When an error occurred in overlapping regions or a region was hit in a region search, the value held here is the logical OR of the access control bits in user mode for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error occurred or the region that was hit in a region search. These bits are set to 0000 0000b when a data memory-protection error occurred in the background region.

Note: If access to a register of the memory-protection unit in user mode causes a data memory-protection error, the value of this register is set to 0000 0000h.

17.2.3 Functions of CPU MPU

17.2.3.1 Memory Protection Function

Memory protection monitors whether or not access by programs running in user mode violates the access control settings, in accord with the access control information set for the individual access control regions and the background region. When the memory-protection unit detects an access control violation (memory-protection error), it notifies the CPU of this information, and the CPU starts access-exception processing.
Memory protection is enabled by setting the memory-protection enable bit (MPEN) of the memory-protection enable register (MPEN) to 1.
An instruction memory-protection error occurs on detection of a violation in terms of executing an instruction, and a data memory-protection error occurs on detection of a violation in terms of reading or writing for operand access. When a data memory-protection error occurs, the operand access that caused the access control violation does not proceed.

17.2.3.2 Cache Attribute Setting Function

The cache attribute setting function is for setting the cache attribute for each region.
The cache attribute setting function is enabled by setting the memory-protection enable bit (MPEN) of the memory-protection enable register (MPEN) to 1.
When this bit is 0 or the corresponding region does not exist, the cache attribute set in the background access control register (MPBAC.DCA/ICA) is used.

17.2.3.3 Region Search

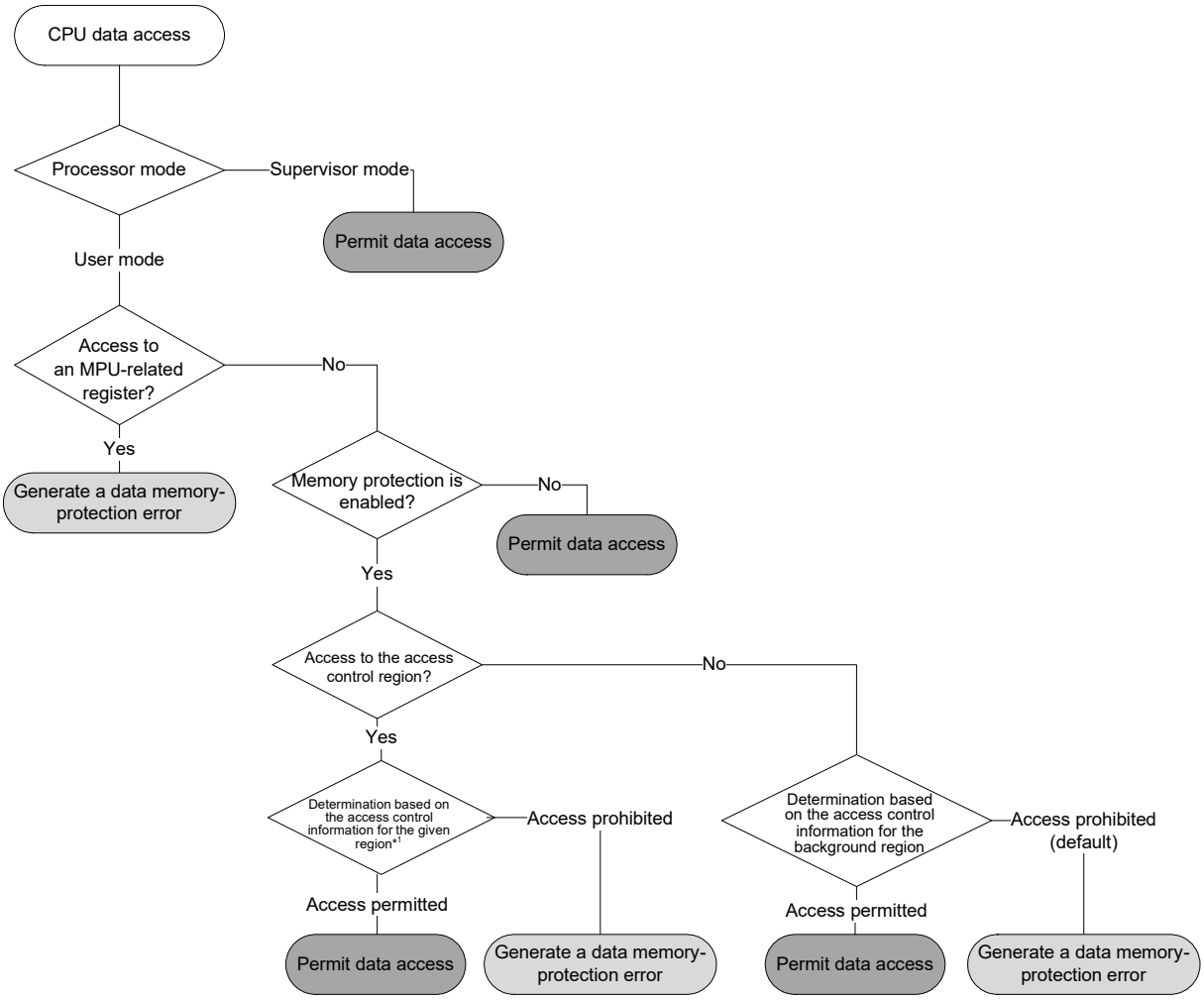
Region search is a function to search for the region among the eight access control regions in which a certain address was hit and the access control information (execution permitted, reading permitted, and writing permitted) settings for that address.
When the region search operation bit (S) of the region search operation register (MPOPS) is set to 1, the address specified in the region search address register (MPSA) is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access control information for the region which was “hit” and for the other regions.
Region search cannot be used to check on the cache attribute.

17.2.3.4 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). These registers are only accessible in supervisor mode. Attempted access to them in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

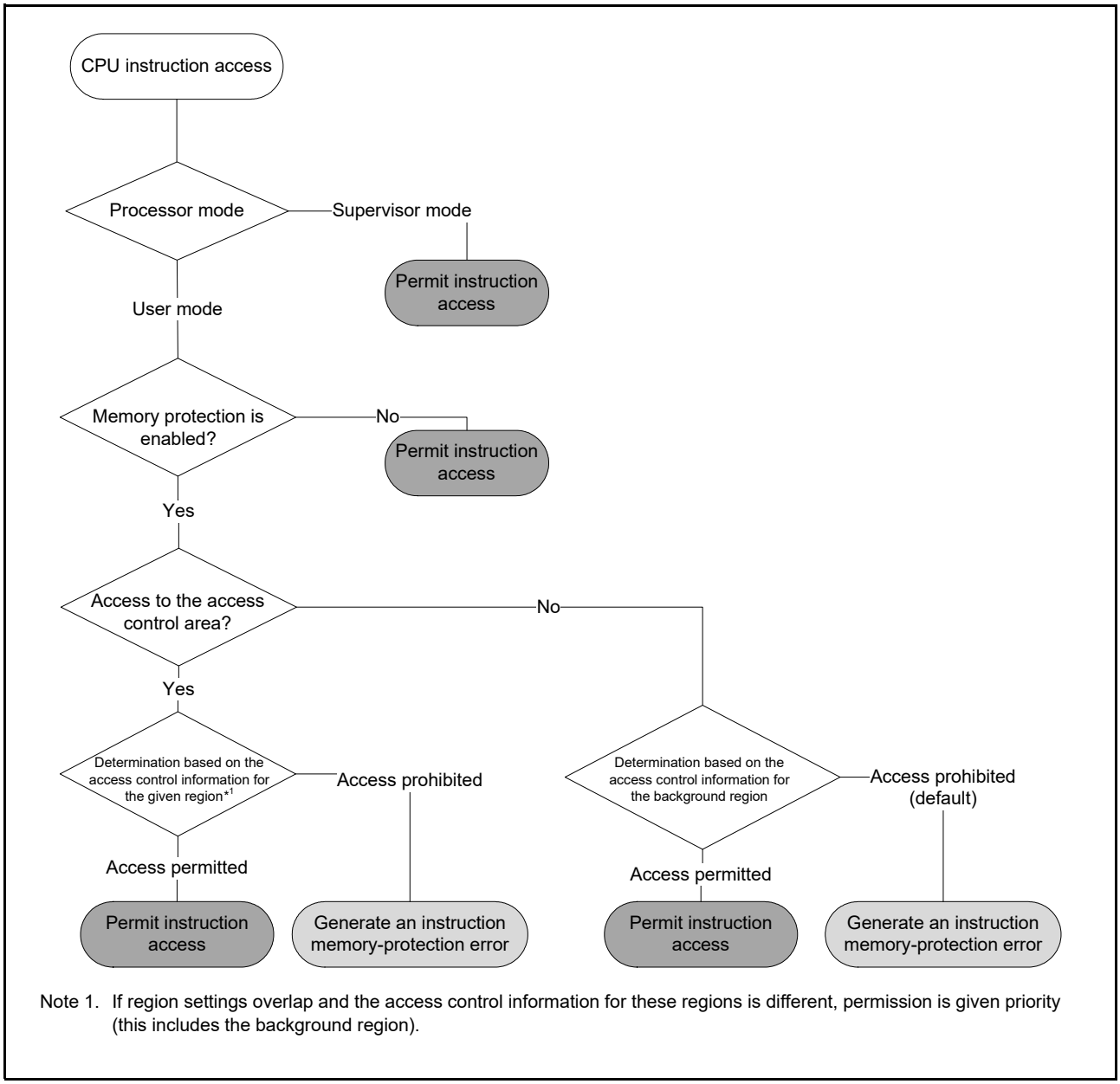
17.2.3.5 Flow for Determining Access by the Memory Protection Function

Figure 17.2 shows the flow for determining data access and Figure 17.3 shows the flow for determining instruction access.



Note 1. If region settings overlap and the access control information for these regions is different, permission is given priority (this includes the background region).

Figure 17.2 Flow for Determining Data Access



Note 1. If region settings overlap and the access control information for these regions is different, permission is given priority (this includes the background region).

Figure 17.3 Flow for Determining Instruction Access

17.2.3.6 Flow for Determining the Cache Attributes

Regardless of the type of MPU, it determines the cache attributes through the flow below, as well as handling the access determination.

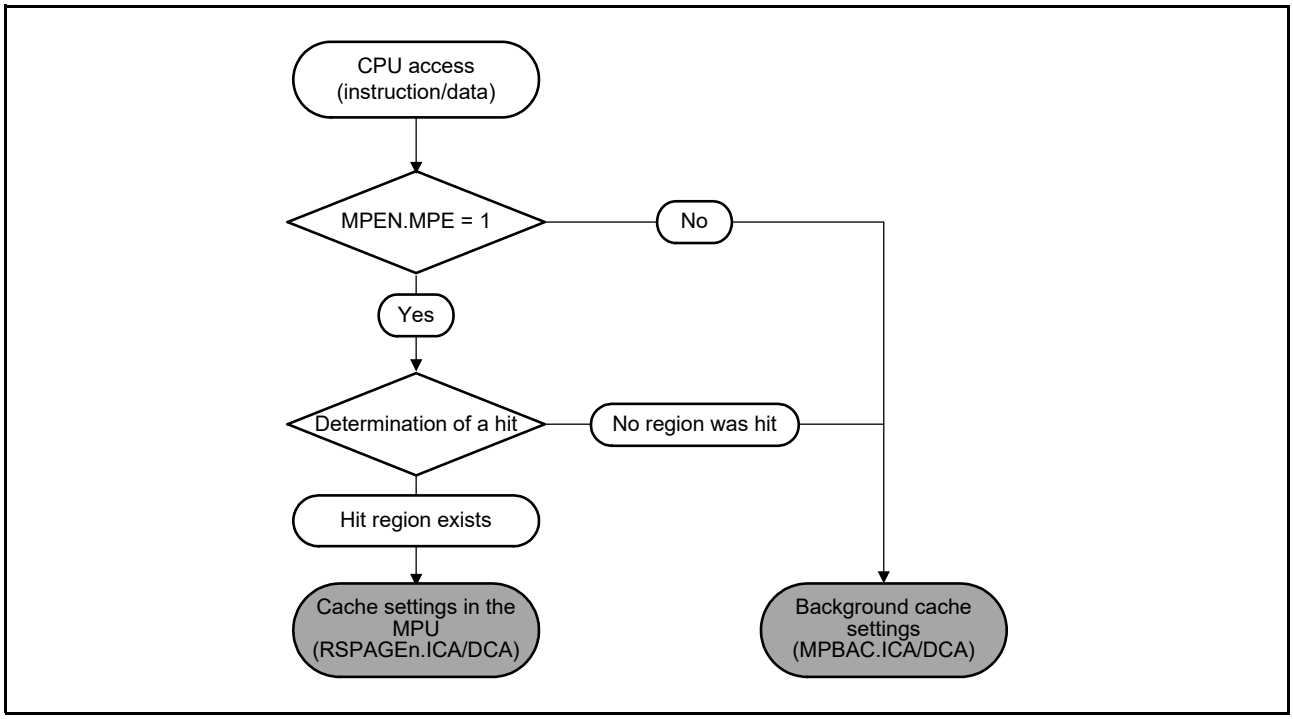


Figure 17.4 Flow for Determining the Cache Attributes

17.2.4 Procedures for Using the Memory Protection Function

17.2.4.1 Setting the Access Control Information and Cache Attributes

The access control information for the individual regions is set in supervisor mode.
Up to eight access control regions are set by using the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn) (n = 0 to 7).
The background access control region is set by using the background access control register (MPBAC).

17.2.4.2 Enabling the Memory Protection and Cache Attribute Setting Functions

Set the memory-protection enable bit (MPEN) of the memory-protection enable (MPEN) register to 1 in supervisor mode. However, the background cache settings are effective even when MPEN = 0

17.2.4.3 Transition to User Mode

After modifying the settings of the registers related to the memory-protection unit, read any of these registers and check that the settings have been made before the transition to user mode. Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting bit (PM) of the processor status word (PSW) saved on the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit of the backup PSW (BPSW) to 1 and then execute an RTFI instruction.

Note: Writing to the PSW.PM bit by an MVTC or POPC instruction has no effect. Use an RTE or RTFI instruction to change the value of the PSW.PM bit.

The memory-protection unit starts checking the addresses for the CPU's instruction execution and operand access following a transition to user mode.

17.2.4.4 Modifying the Cache Attributes

The cache attributes are effective even in supervisor mode. Therefore, when modifying the cache attributes, that is; when modifying the attributes for a given region, enabling or disabling the MPU and the attributes for a given region, or modifying a background attribute; although the modified settings will in general be reflected from the next access for both data access and instruction access, this also depends on the state of the pipeline, so reflection of the modified settings from execution of the next instruction is not guaranteed.
By executing a branch instruction, reflection of the settings after the modification of cache attributes is guaranteed from the destination of the branch instruction. In addition, proceed with a write-back from the cache and cache invalidation processing as required.

Modify cache attributes either while the cache is disabled or by following the procedure below.

Procedure for modifying cache attributes while the cache is enabled:

In this procedure, we define region X as the area for which the cache attributes change before and after modifying the attributes, and region Y as the area for which they do not change.

1. Place the code for modifying cache attributes in region Y in advance. At this time, make sure that not only the instruction that modifies the cache attributes but also the 32 bytes that follow the instruction are placed within region Y, in consideration of instruction prefetching.
2. Execute the code placed in region Y to modify the cache attributes of region X. Proceed with a write-back from the cache and cache invalidation processing as required.

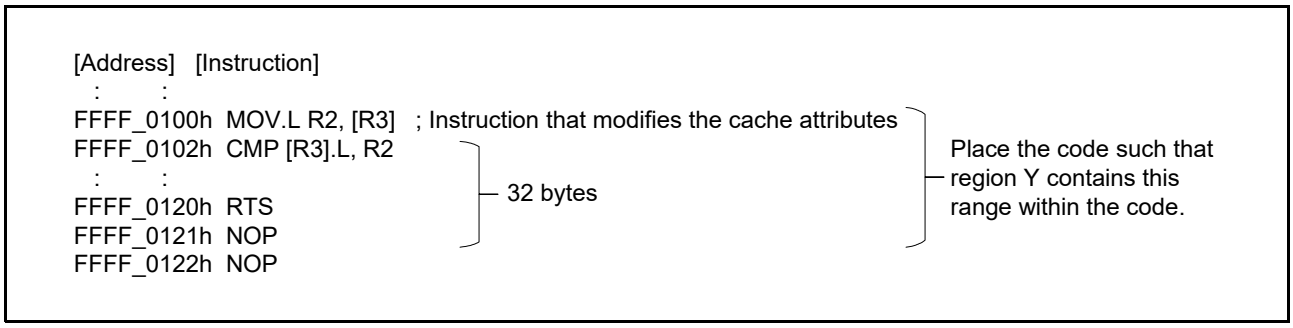


Figure 17.5 Depiction of Placing the Code for Modifying the Cache Attributes in Region Y

17.2.4.5 Processing in Response to a Memory-Protection Error

The CPU starts access exception processing on detection of an access control violation (memory-protection error). For details of the CPU operation for access exception processing, see section 14, Exception Handling.
Check the instruction memory-protection error generation bit (IMPER) and data memory-protection error generation bit (DMPER) in the memory-protection error status register (MPESTS) in the exception processing routine to determine which type of error occurred, instruction memory-protection error or data memory-protection error.
After confirming the type of error, clear the MPESTS register by writing 1 to the error status clear bit (CLR) in the memory-protection error status clear register (MPECLR).

(1) When a data memory-protection error occurs

The address of the instruction that caused a memory-protection error is saved on the stack by the access exception processing by the CPU. Furthermore, the operand access address where a memory-protection error occurred is stored in the data memory-protection error address register (MPDEA), and the region information that caused a memory-protection error is stored in the data-hit region register (MHITD).

- When access to valid regions 0 to 7 violates access control

The data-hit region bits (MHITD.HITD[7:0]) corresponding to the number of the region where the error occurred is set to 1. The logical OR of the access control information for the region where the error occurred is set in the data-hit region access control bits in user mode (MHITD.UHACD[2:0]).

- When access to valid regions 0 to 7 is outside the regions and access control for the background region is violated

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. The access control information for the background region is set in the data-hit region access control bits in user mode (MHITD.UHACD[2:0]).

By referring to this information, processing such as identifying the sources of errors can proceed.

(2) When an instruction memory-protection error occurs

The address of the instruction that caused a memory-protection error is saved on the stack by the access exception processing by the CPU. Furthermore, the information for the region where the error occurred is stored in the instruction-hit region register (MHITI).

- When access to valid regions 0 to 7 violates access control

The instruction-hit region bits (MHITI.HITI[7:0]) corresponding to the number of the region where the error occurred is set to 1. The logical OR of the access control information for the region where the error occurred is set in the instruction-hit region access control bits in user mode (MHITI.UHACI[2:0]).

- When access to valid regions 0 to 7 is outside the regions and access control for the background region is violated

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. The access control information for the background region is set in the instruction-hit region access control bits in user mode (MHITI.UHACI[2:0]).

(3) When an CPU access error occurs

The address of the instruction executed by the CPU that caused an access exception is saved on the stack by the access exception processing by the CPU. The registers of the memory protection function are not changed.

17.3 Bus Master MPU

17.3.1 Overview of Bus Master MPU

The bus master MPU monitors addresses accessed by the bus master over the whole address space (0000 0000h to FFFF FFFFh). The access monitoring information can be set as follows:

- xSPI0: 4 regions
- xSPI1: 4 regions

The bus master MPU monitors access to individual regions based on these settings.
In the case of access to a region protected by the bus master MPU or access that crosses the boundary of a permitted MPU region in a burst access, the bus master MPU returns an error response.
Regarding error access, see section 17.3.2.1, Master Memory-Protection Unit Security Attribute Register B (MMPUSARB) and section 16.8.2, Operations When a Bus Error Occurs in section 16, Buses.

17.3.2 Register Descriptions of Bus Master MPU

17.3.2.1 Master Memory-Protection Unit Security Attribute Register B (MMPUSARB)

Address(es): CPSCU.MMPUSARB 8701 8134h,
CPSCU_NS.MMPUSARB 0701 8134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MMPUBSA8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MMPUBSA8	MMPUB8 Security Select	0: Secure 1: Non-secure	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

MMPUBSA8 Bit (MMPUB8 Security Select)

This bit selects the security attribute of the following registers.

- MMPUOAD
- MMPUOADPT

17.3.2.2 MMPU Start Address Register for xSPI (MMPUSXSPImn) (m = 0, 1; n = 0 to 3)

Address(es): BUSMPU.MMPUSXSPI00 8701 1404h, BUSMPU.MMPUSXSPI01 8701 1414h, BUSMPU.MMPUSXSPI02 8701 1424h, BUSMPU.MMPUSXSPI03 8701 1434h, BUSMPU.MMPUSXSPI10 8701 1604h, BUSMPU.MMPUSXSPI11 8701 1614h, BUSMPU.MMPUSXSPI12 8701 1624h, BUSMPU.MMPUSXSPI13 8701 1634h,
BUSMPU_NS.MMPUSXSPI00 0701 1404h, BUSMPU_NS.MMPUSXSPI01 0701 1414h, BUSMPU_NS.MMPUSXSPI02 0701 1424h, BUSMPU_NS.MMPUSXSPI03 0701 1434h, BUSMPU_NS.MMPUSXSPI10 0701 1604h, BUSMPU_NS.MMPUSXSPI11 0701 1614h, BUSMPU_NS.MMPUSXSPI12 0701 1624h, BUSMPU_NS.MMPUSXSPI13 0701 1634h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
b31 to b0	n/a	Region start address register for xSPI Region start address (for use in region determination) The MPU region start address must be set in the range from 0000 0000h to FFFF F000h. Writing to bits 0 to 11 will be ignored. These bits are read as 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MMPUSXSPImn (m = 0, 1; n = 0 to 3) register specifies the start address where the region starts.
This register must be read or written in 32-bit units. It is neither readable nor writable in 8-bit or 16-bit units. If it is read or written in 8-bit or 16-bit units, the operation is not guaranteed.

17.3.2.3 MMPU End Address Register for xSPI (MMPUEXSPImn) (m = 0, 1; n = 0 to 3)

Address(es): BUSMPU.MMPUEXSPI00 8701 1408h, BUSMPU.MMPUEXSPI01 8701 1418h, BUSMPU.MMPUEXSPI02 8701 1428h, BUSMPU.MMPUEXSPI03 8701 1438h, BUSMPU.MMPUEXSPI10 8701 1608h, BUSMPU.MMPUEXSPI11 8701 1618h, BUSMPU.MMPUEXSPI12 8701 1628h, BUSMPU.MMPUEXSPI13 8701 1638h, BUSMPU_NS.MMPUEXSPI00 0701 1408h, BUSMPU_NS.MMPUEXSPI01 0701 1418h, BUSMPU_NS.MMPUEXSPI02 0701 1428h, BUSMPU_NS.MMPUEXSPI03 0701 1438h, BUSMPU_NS.MMPUEXSPI10 0701 1608h, BUSMPU_NS.MMPUEXSPI11 0701 1618h, BUSMPU_NS.MMPUEXSPI12 0701 1628h, BUSMPU_NS.MMPUEXSPI13 0701 1638h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Description	R/W
b31 to b0	n/a	Region end address register for xSPI Region end address (for use in region determination) The MPU region end address must be set in the range from 0000 0FFFh to FFFF FFFF FFFFh. Writing to bits 0 to 11 will be ignored. These bits are read as 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MMPUSXSPImn (m = 0, 1; n = 0 to 3) register specifies the end address where the region ends.
This register must be read or written in 32-bit units. It is neither readable nor writable in 8-bit or 16-bit units. If it is read or written in 8-bit or 16-bit units, the operation is not guaranteed.

17.3.2.4 MMPU Access Control Register for xSPI (MMPUACXSPImn) (m = 0, 1, n = 0 to 3)

Address(es): BUSMPU.MMPUACXSPI00 8701 1400h, BUSMPU.MMPUACXSPI01 8701 1410h, BUSMPU.MMPUACXSPI02 8701 1420h, BUSMPU.MMPUACXSPI03 8701 1430h, BUSMPU.MMPUACXSPI10 8701 1600h, BUSMPU.MMPUACXSPI11 8701 1610h, BUSMPU.MMPUACXSPI12 8701 1620h, BUSMPU.MMPUACXSPI13 8701 1630h, BUSMPU_NS.MMPUACXSPI00 0701 1400h, BUSMPU_NS.MMPUACXSPI01 0701 1410h, BUSMPU_NS.MMPUACXSPI02 0701 1420h, BUSMPU_NS.MMPUACXSPI03 0701 1430h, BUSMPU_NS.MMPUACXSPI10 0701 1600h, BUSMPU_NS.MMPUACXSPI11 0701 1610h, BUSMPU_NS.MMPUACXSPI12 0701 1620h, BUSMPU_NS.MMPUACXSPI13 0701 1630h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENABLE	Region Enable	0: XSPI region mn unit disabled 1: XSPI region mn unit enabled	R/W
b1	RP	Read Protect	0: Reading permitted 1: Protected against reading	R/W
b2	WP	Write Protect	0: Writing permitted 1: Protected against writing	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

ENABLE Bit (Region Enable)

This bit enables or disables the xSPImn (m = 0, 1; n = 0 to 3) region.
When the ENABLE bit is set to 1, the RP and WP bits are enabled for permission or prohibition of access to the region set by the MMPUSXSPImn and MMPUEXSPImn registers.
When the ENABLE bit is set to 0, access to the xSPI0 region is handled as being outside the region.

RP Bit (Read Protect)

This bit enables or disables read protection for the xSPImn (m = 0, 1; n = 0 to 3) region.
This bit can be used when the ENABLE bit is set to 1.

WP Bit (Write Protect)

This bit enables or disables write protection for the xSPImn (m = 0, 1; n = 0 to 3) region.
This bit can be used when the ENABLE bit is set to 1.

17.3.2.5 MMPU Enable Register for xSPI (MMPUENXSPIm) (m = 0, 1)

Address(es): BUSMPU.MMPUENXSPI0 8701 1300h, BUSMPU.MMPUENXSPI1 8701 1500h,
BUSMPU_NS.MMPUENXSPI0 0701 1300h, BUSMPU_NS.MMPUENXSPI1 0701 1500h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENABLE	xSPI Bus Master MPU Enable	0: xSPI bus master MPU disabled 1: xSPI bus master MPU enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits permit or prohibit writing to the ENABLE bit.	W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register must be written in 16-bit units. Reading or writing to the register in 8-bit units is prohibited. If it is read or written in 8-bit units, the operation is not guaranteed.

ENABLE Bit (xSPI Bus Master MPU Enable)

This bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, the MMPUACXSPImn (m = 0, 1; n = 0 to 3) register can be used.

When the ENABLE bit is set to 0, the MMPUACXSPImn (m = 0, 1; n = 0 to 3) register cannot be used for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When setting the ENABLE bit, write A5h to the KEY[7:0] bits at the same time.

KEY[7:0] Bits (Key Code)

These bits permit or prohibit writing to the ENABLE bit.

When writing to the ENABLE bit, write A5h to the KEY[7:0] bits at the same time.

If a value other than A5h is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 00h.

17.3.2.6 MMPU Enable Protect Register for xSPI (MMPUENPTXSPIm) (m = 0, 1)

Address(es): BUSMPU.MMPUENPTXSPI0 8701 1304h, BUSMPU.MMPUENPTXSPI1 8701 1504h, BUSMPU_NS.MMPUENPTXSPI0 0701 1304h, BUSMPU_NS.MMPUENPTXSPI1 0701 1504h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PROTECT	Write Protect	0: The MMPUENXSPI register is writable. 1: The MMPUENXSPI register is write-protected. The register is readable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits permit or prohibit writing to the PROTECT bit.	W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register must be written in 16-bit units. Reading or writing to the register in 8-bit units is prohibited. If it is read or written in 8-bit units, the operation is not guaranteed.

PROTECT Bit (Write Protect)

This bit controls protection of the MMPUENXSPIm (m = 0, 1) register.

When setting the PROTECT bit, write A5h to the KEY[7:0] bits at the same time.

KEY[7:0] Bits (Key Code)

These bits permit or prohibit writing to the PROTECT bit.

When writing to the PROTECT bit, write A5h to the KEY[7:0] bits at the same time.

If a value other than A5h is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

17.3.2.7 MMPU Region Protect Register for xSPI (MMPURPTXSPIm) (m = 0, 1)

Address(es): BUSMPU.MMPURPTXSPI0 8701 1308h, BUSMPU.MMPURPTXSPI1 8701 1508h, BUSMPU_NS.MMPURPTXSPI0 0701 1308h, BUSMPU_NS.MMPURPTXSPI1 0701 1508h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PROTECT	Write Protect	0: The bus master MPU register for xSPI is writable. 1: The bus master MPU register for xSPI is write-protected. The register is readable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits permit or prohibit writing to the PROTECT bit.	W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register must be written in 16-bit units. Reading or writing to the register in 8-bit units is prohibited. If it is read or written in 8-bit units, the operation is not guaranteed.

PROTECT Bit (Write Protect)

This bit permits or prohibits writing to the registers associated with protection.
The MMPURPTXSPIm.PROTECT (m = 0, 1) bit controls the following registers.

- MMPUSXSPImn (m = 0, 1; n = 0 to 3)
- MMPUEXSPImn (m = 0, 1; n = 0 to 3)
- MMPUACXSPImn (m = 0, 1; n = 0 to 3)

When writing to the PROTECT bit, write A5h to the KEY[7:0] bits at the same time in an overall 16-bit unit.

KEY[7:0] Bits (Key Code)

These bits permit or prohibit writing to the PROTECT bit.
When writing to the PROTECT bit, write A5h to the KEY[7:0] bits at the same time.
If a value other than A5h is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

17.3.2.8 MMPU Post-Detection Operation Register (MMPUOAD)

Address(es): BUSMPU.MMPUOAD 8701 0000h,
BUSMPU_NS.MMPUOAD 0701 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OAD	Post-Detection Operation Select	0: Interrupt request 1: Reset request	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits permit or prohibit writing to the OAD bit.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register must be written in 16-bit units. Reading or writing to the register in 8-bit units is prohibited. If it is read or written in 8-bit units, the operation is not guaranteed.

OAD Bit (Post-Detection Operation Select)

This bit specifies operation when an access violation is detected.

When the OAD bit is 0, an error response is returned, and an interrupt request is generated when BUSINTEN<Master Name>.EN = 1. For details, see section 16.3.35, Bus Error Interrupt Enable Register (BUSINTENCPU0IF, BUSINTENCPU0OA, BUSINTENDMAC0, BUSINTENEXDMAC, BUSINTENEDMAC, BUSINTENXSPI0, BUSINTENXSPI1).

An interrupt caused by a bus error can be used as an NMI. For details, refer to the specification of the ICU.NMIER register.

When the OAD bit is 1, a reset request is generated.

When writing to the OAD bit, write A5h to the KEY[7:0] bits at the same time in an overall 16-bit unit.

KEY[7:0] Bit (Key Code)

These bits permit or prohibit writing to the OAD bit. When writing to the OAD bit, write A5h to the KEY[7:0] bits at the same time. If any other value is written, the OAD bit is not updated.

The KEY[7:0] bits are always read as 00h.

17.3.2.9 MMPU Post-Detection Operation Protect Register (MMPUOADPT)

Address(es): BUSMPU.MMPUOADPT 8701 0004h,
BUSMPU_NS.MMPUOADPT 0701 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PROTECT	Write Protect	0: The MMPUOAD register is writable. 1: The MMPUOAD register is write-protected. The register is readable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits permit or prohibit writing to the PROTECT bit.	W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register must be written in 16-bit units. Reading or writing to the register in 8-bit units is prohibited. If it is read or written in 8-bit units, the operation is not guaranteed.

PROTECT Bit (Write Protect)

This bit permits or prohibits writing to the registers associated with protection.
The MMPUOADPT.PROTECT bit controls the following register.

- MMPUOAD

When writing to the PROTECT bit, write A5h to the KEY[7:0] bits at the same time in an overall 16-bit unit.

KEY[7:0] Bits (Key Code)

These bits permit or prohibit writing to the PROTECT bit. When writing to the PROTECT bit, write A5h to the KEY[7:0] bits at the same time. If any other value is written, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

17.3.3 Functions of Bus Master MPU

17.3.3.1 Bus Master Memory Protection

The bus master MPU monitors memory access by using the control setting made individually to the access control region. Access to a region protected by the bus master MPU causes the MPU to generate a memory-protection error. “Bus master MPU” refers to either MPU in the master group of xSPI0 and xSPI1. The memory protection function checks the bus addresses of the bus master group and blocks illegal access through the bus master group to a region protected by a bus master MPU. The region setting registers for the xSPI bus master MPU can be configured for secure masters and non-secure masters by using the PSCU.PSARB register. The MPU region settings while the setting of the PSCU.PSARB.PSARBn bit (n = 16, 17) is 0 only apply to access by secure masters. The MPU region settings while the setting of the given bit is 1 only apply to access by non-secure masters.

The bus master MPU permits access and execution for all regions after a reset. If MMPUENXXXX.ENABLE is 1 and there is no MPU region setting for the corresponding bus master, all regions are protected. (XXXX = Master Group name)

Figure 17.6 shows access permission or protection in the case of overlapping bus master MPU regions. Access control for the overlapping regions is as follows.

- The region is handled as a protected region when output of one or more region units is a protected region.
- The region is handled as a protected region when output of all region units is outside the regions.
- Other cases are handled as permitted regions.

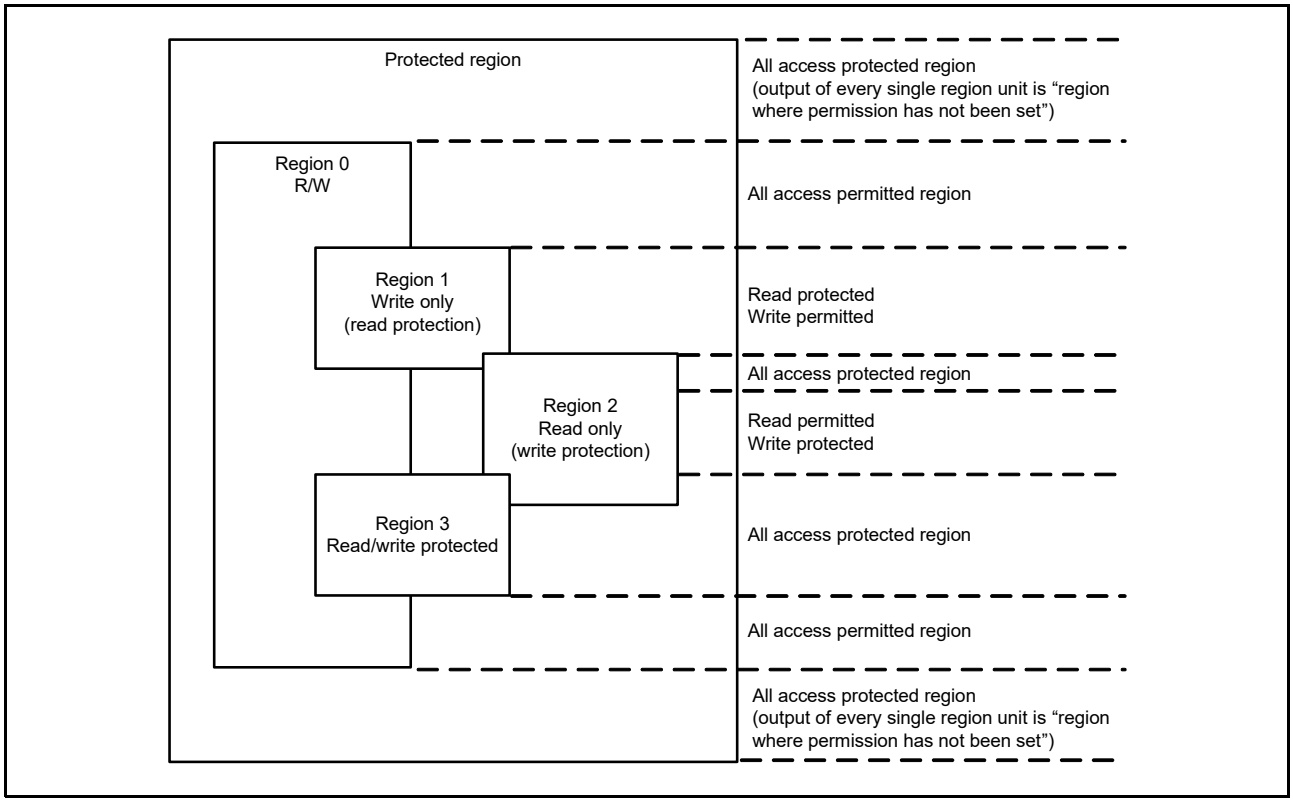


Figure 17.6 Access Permission or Protection in the Case of Overlapping Bus Master MPU Regions

Figure 17.7 shows the flow for setting the registers after a reset. While making these register settings, stop each bus master except for the CPU.

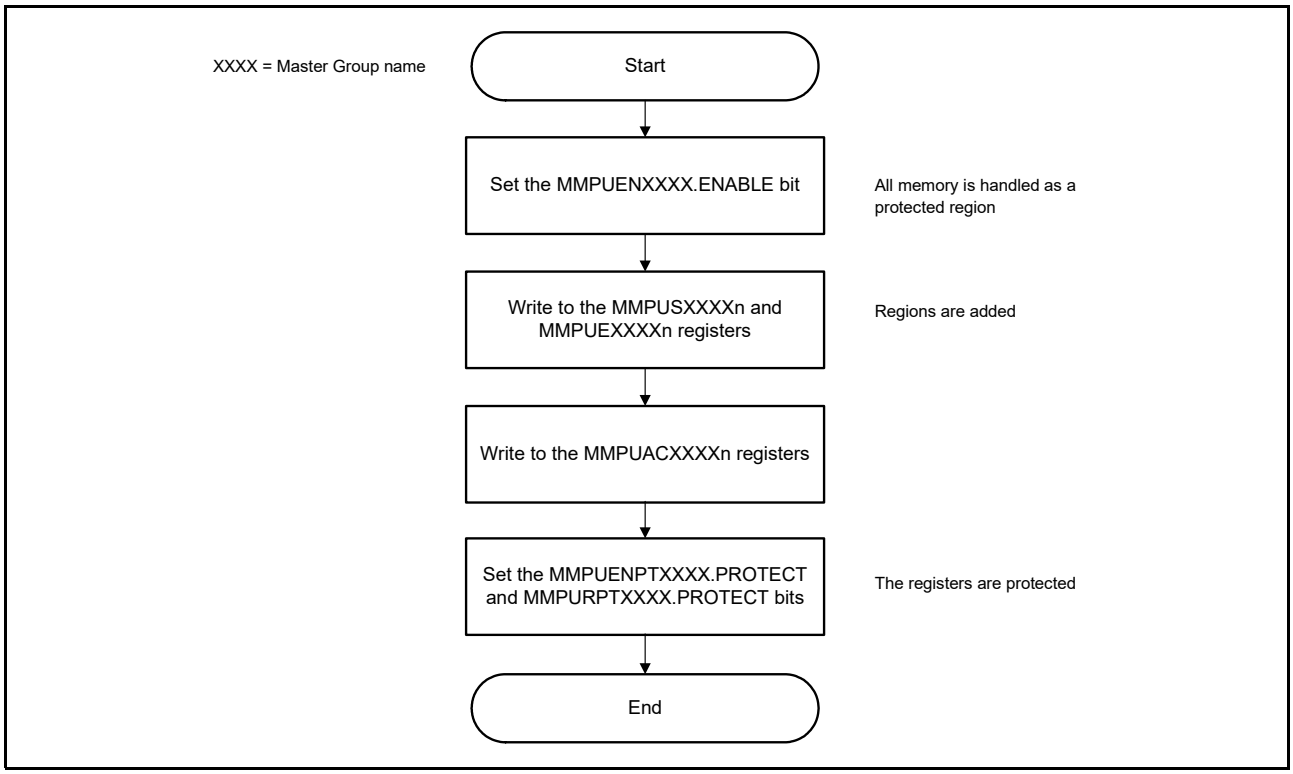


Figure 17.7 Flow for Setting Registers after Reset

Figure 17.8 shows the flow for setting the registers to add regions. While making these register settings, stop each bus master except for the CPU.

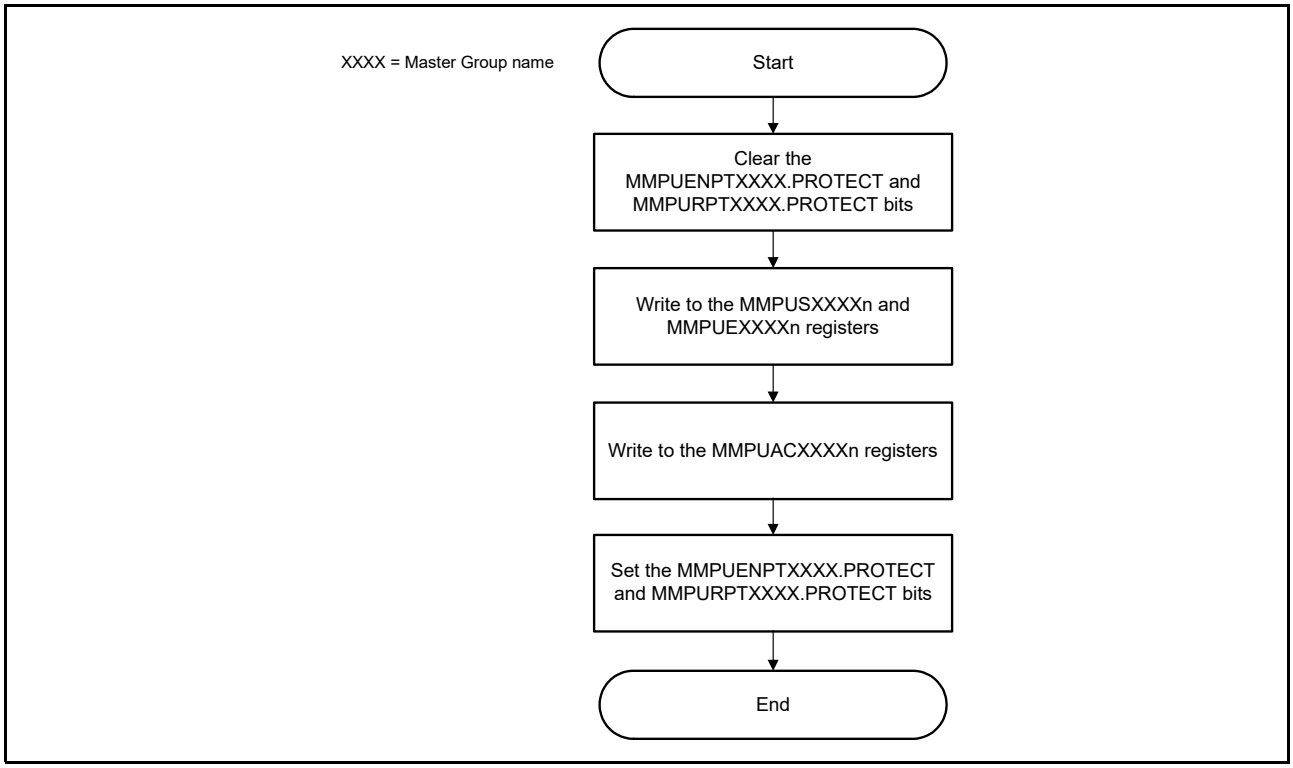


Figure 17.8 Flow for Setting Registers for Region Addition

17.3.3.2 Register Protection

The registers related to the bus master MPU can be protected by using the PROTECT bits as listed in Table 17.3.

Table 17.3 PROTECT Bits and Target Registers for Protection

PROTECT Bit	Target Registers for Protection
MMPUENPTXSPIm.PROTECT	MMPUENXSPI m (m = 0, 1)
MMPURPTXSPIm.PROTECT	MMPUSXSPImn (m = 0, 1; n = 0 to 3) MMPUEXSPImn (m = 0, 1; n = 0 to 3) MMPUACXSPImn (m = 0, 1; n = 0 to 3)
MMPUOADPT.PROTECT	MMPUOAD

17.3.3.3 Memory-Protection Error

The bus master MPU generates an error on detection of access to the protected region.
Set the MMPUOAD.OAD bit to select whether the error is reported as an interrupt or reset.
For details of the bus master MPU error, see section 16.8.2, Operations When a Bus Error Occurs.
The state of the interrupt is indicated in ICU.IR16.IR. For details, see section 15, Interrupt Controller (ICUM).
The state of the reset is indicated in SYSC.RSTSR1.BUSRF. For details, see section 6, Resets.
An interrupt caused by a bus error can be used as an NMI. For details, refer to the specification of the ICU.NMIER register.

18. DMA Controller (DMACAa)

This MCU incorporates an 8-channel direct memory access controller (DMAC). The DMAC is a module to transfer data without the CPU. When a transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

18.1 Overview

Table 18.1 lists the specifications of the DMAC, and Figure 18.1 shows a block diagram of the DMAC.

Table 18.1 Specifications of DMAC (1/2)

Item		Description DMAC
Number of channels		8 (DMACm (m = 0 to 7))
Transfer space		4 Gbytes (0000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		64M data (Maximum number of transfers in block transfer mode: 1024 data × 65536 blocks)
DMA trigger source		• Trigger source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Selectable from fixed priority or round-robin • Fixed priority: Channel 0 > channel 1 > channel 2 > channel 3... > channel 7 > DTC (channel 0: Highest) • Round-robin
Transfer data	Single data	Bit length: 8, 16, 32, 64 bits
	Block size	Number of data: 1 to 1024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running function (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1024 • Free running function settable
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 data • Free running function settable
	Repeat-block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 data • Block transfer can be repeated • Maximum settable repeat size: 64K • Free running function settable
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Processing on DMA transfer error		• When DMA transfer error occurs, the transfer that caused the error is stopped • For clearing requests to the registers for trigger source in the ICU, clearing targets are selectable for each channel or all channels.
CPU interrupt source (DMACnI (n = 0 to 3, 74))	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	• Generated when the repeat size of data transfer is completed. • Generated when the source address extended repeat area overflows. • Generated when the destination address extended repeat area overflows.
Event link function		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Power consumption reduction function		Module-stop state can be set.

Table 18.1 Specifications of DMAC (2/2)

Item	Description
	DMAC
Protect Zone Filter	Security can be set for each channel.

Note 1. For details on DMAC trigger sources, see Table 15.6, Interrupt Vector Table, Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

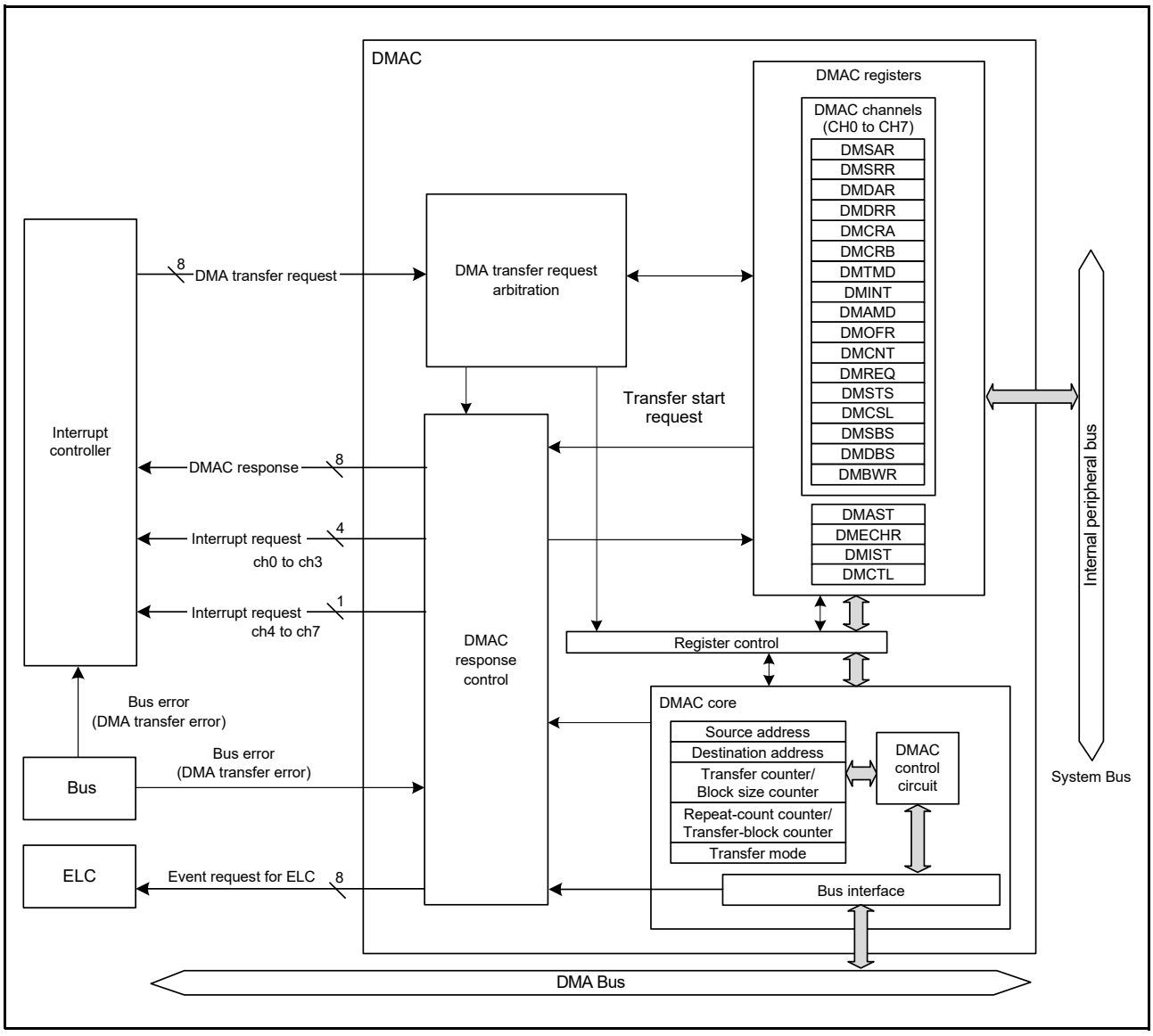


Figure 18.1 Block Diagram of DMAC

18.2 Register Descriptions

Security attribution can be set for each register and is used to determine whether the target register is a secure register or a non-secure register. The attribution can be set by the DMACCHSAR register.

18.2.1 DMAC Channel Security Attribution Register (DMACCHSAR)

Address(es): CPSCU.DMACCHSAR 8701 81A0h, CPSCU_NS.DMACCHSAR 0701 81A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SADMAC7	SADMAC6	SADMAC5	SADMAC4	SADMAC3	SADMAC2	SADMAC1	SADMAC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SADMAC0	Security Attributes of Output and Registers for DMAC Channel 0 Select	0: Secure 1: Non-secure	R/W
b1	SADMAC1	Security Attributes of Output and Registers for DMAC Channel 1 Select	0: Secure 1: Non-secure	R/W
b2	SADMAC2	Security Attributes of Output and Registers for DMAC Channel 2 Select	0: Secure 1: Non-secure	R/W
b3	SADMAC3	Security Attributes of Output and Registers for DMAC Channel 3 Select	0: Secure 1: Non-secure	R/W
b4	SADMAC4	Security Attributes of Output and Registers for DMAC Channel 4 Select	0: Secure 1: Non-secure	R/W
b5	SADMAC5	Security Attributes of Output and Registers for DMAC Channel 5 Select	0: Secure 1: Non-secure	R/W
b6	SADMAC6	Security Attributes of Output and Registers for DMAC Channel 6 Select	0: Secure 1: Non-secure	R/W
b7	SADMAC7	Security Attributes of Output and Registers for DMAC Channel 7 Select	0: Secure 1: Non-secure	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

SADMACn Bits (Security Attributes of Output and Registers for DMAC Channel n Select) (n = 0 to 7)

Each bit specifies the security attribute of the register for a DMAC Channel n. For details, see section 18.3.13, Channel Security. Set the same security attribute to the four bits, SADMAC4 to SADMAC7. The target registers are ICU.DMRSRm, DMIST, DMSAR, DMSRR, DMDAR, DMDRR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, DMCNT, DMREQ, DMSTS, DMCSL, DMSBS, DMDBS, DMBWR, and DMIST.

18.2.2 DMA Controller Security Attribution Register (DMACSAR)

Address(es): CPSCU.DMACSAR 8701 8034h, CPSCU_NS.DMACSAR 0701 8034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMASTSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMASTSA0	DMAST Security Attribution Setting 0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

For DMAC, security attribution is set for each channel. However, security attribution of the DMAST and DMCTL registers is set by this register.

DMASTSA0 Bit (DMAST Security Attribution Setting 0)

The DMASTSA0 bit specifies security attribution of the DMAST and DMCTL registers.

18.2.3 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 8700 7000h, DMAC1.DMSAR 8700 7040h, DMAC2.DMSAR 8700 7080h, DMAC3.DMSAR 8700 70C0h, DMAC4.DMSAR 8700 7100h, DMAC5.DMSAR 8700 7140h, DMAC6.DMSAR 8700 7180h, DMAC7.DMSAR 8700 71C0h, DMAC0_NS.DMSAR 0700 7000h, DMAC1_NS.DMSAR 0700 7040h, DMAC2_NS.DMSAR 0700 7080h, DMAC3_NS.DMSAR 0700 70C0h, DMAC4_NS.DMSAR 0700 7100h, DMAC5_NS.DMSAR 0700 7140h, DMAC6_NS.DMSAR 0700 7180h, DMAC7_NS.DMSAR 0700 71C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

18.2.4 DMA Source Reload Address Register (DMSRR)

Address(es): DMAC0.DMSRR 8700 7020h, DMAC1.DMSRR 8700 7060h, DMAC2.DMSRR 8700 70A0h, DMAC3.DMSRR 8700 70E0h, DMAC4.DMSRR 8700 7120h, DMAC5.DMSRR 8700 7160h, DMAC6.DMSRR 8700 71A0h, DMAC7.DMSRR 8700 71E0h, DMAC0_NS.DMSRR 0700 7020h, DMAC1_NS.DMSRR 0700 7060h, DMAC2_NS.DMSRR 0700 70A0h, DMAC3_NS.DMSRR 0700 70E0h, DMAC4_NS.DMSRR 0700 7120h, DMAC5_NS.DMSRR 0700 7160h, DMAC6_NS.DMSRR 0700 71A0h, DMAC7_NS.DMSRR 0700 71E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source reload address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DMSRR register is used to specify the DMA transfer source start and reload addresses in repeat-block transfer mode. Set the same value to DMSRR register as that of DMSAR register. For details on the repeat-block transfer mode operation, see section (4), Repeat-Block Transfer Mode. In normal transfer mode, repeat transfer mode and block transfer mode, DMSRR register is not used. The setting is invalid.
Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

18.2.5 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 8700 7004h, DMAC1.DMDAR 8700 7044h, DMAC2.DMDAR 8700 7084h, DMAC3.DMDAR 8700 70C4h, DMAC4.DMDAR 8700 7104h, DMAC5.DMDAR 8700 7144h, DMAC6.DMDAR 8700 7184h, DMAC7.DMDAR 8700 71C4h, DMAC0_NS.DMDAR 0700 7004h, DMAC1_NS.DMDAR 0700 7044h, DMAC2_NS.DMDAR 0700 7084h, DMAC3_NS.DMDAR 0700 70C4h, DMAC4_NS.DMDAR 0700 7104h, DMAC5_NS.DMDAR 0700 7144h, DMAC6_NS.DMDAR 0700 7184h, DMAC7_NS.DMDAR 0700 71C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

18.2.6 DMA Destination Reload Address Register (DMDRR)

Address(es): DMAC0.DMDRR 8700 7024h, DMAC1.DMDRR 8700 7064h, DMAC2.DMDRR 8700 70A4h, DMAC3.DMDRR 8700 70E4h, DMAC4.DMDRR 8700 7124h, DMAC5.DMDRR 8700 7164h, DMAC6.DMDRR 8700 71A4h, DMAC7.DMDRR 8700 71E4h, DMAC0_NS.DMDRR 0700 7024h, DMAC1_NS.DMDRR 0700 7064h, DMAC2_NS.DMDRR 0700 70A4h, DMAC3_NS.DMDRR 0700 70E4h, DMAC4_NS.DMDRR 0700 7124h, DMAC5_NS.DMDRR 0700 7164h, DMAC6_NS.DMDRR 0700 71A4h, DMAC7_NS.DMDRR 0700 71E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination reload address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DMDRR register is used to specify the DMA transfer destination start and reload addresses in repeat-block transfer mode. Set the same value to DMDRR register as that of DMDAR register. For details on the repeat-block transfer mode operation, see section (4), Repeat-Block Transfer Mode. In normal transfer mode, repeat transfer mode and block transfer mode, DMDRR register is not used. The setting is invalid.
Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

18.2.7 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 8700 7008h, DMAC1.DMCRA 8700 7048h, DMAC2.DMCRA 8700 7088h, DMAC3.DMCRA 8700 70C8h, DMAC4.DMCRA 8700 7108h, DMAC5.DMCRA 8700 7148h, DMAC6.DMCRA 8700 7188h, DMAC7.DMCRA 8700 71C8h, DMAC0_NS.DMCRA 0700 7008h, DMAC1_NS.DMCRA 0700 7048h, DMAC2_NS.DMCRA 0700 7088h, DMAC3_NS.DMCRA 0700 70C8h, DMAC4_NS.DMCRA 0700 7108h, DMAC5_NS.DMCRA 0700 7148h, DMAC6_NS.DMCRA 0700 7188h, DMAC7_NS.DMCRA 0700 71C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	DMCRAH[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMCRAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DMCRAL[15:0]	Lower Bits of Transfer Count	Specifies the number of transfer operations	R/W
b25 to b16	DMCRAH[9:0]	Upper Bits of Transfer Count		R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the same value for DMCRAH[9:0] and DMCRAL[15:0] bits in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode and block transfer mode.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

(1) Normal Transfer Mode (DMTMD.MD[1:0] Bits = 00b)

The DMCRAL[15:0] bits function as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred. When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Free running function is not selected by DMTMD.TKP bit in normal transfer mode. The DMCRAH[9:0] bits are not used in normal transfer mode. Write 000h to the DMCRAH[9:0] bits.

(2) Repeat Transfer Mode (DMTMD.MD[1:0] Bits = 01b)

The DMCRAH[9:0] bits specify the repeat size and the DMCRAL[15:0] bits function as a 10-bit transfer counter. The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh can be set for the DMCRAH[9:0] and DMCRAL[15:0] bits. Setting bits 15 to 10 in the DMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the DMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the DMCRAH[9:0] bits is loaded into the DMCRAL[15:0] bits.

(3) Block Transfer Mode (DMTMD.MD[1:0] Bits = 10b)

The DMCRAH[9:0] bits specify the block size and the DMCRAL[15:0] bits function as a 10-bit block size counter. The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for the DMCRAH[9:0] and DMCRAL[15:0] bits. Setting bits 15 to 10 in the DMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the DMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the DMCRAH[9:0] bits is loaded into the

DMCRAL[15:0] bits.

(4) Repeat-Block Transfer Mode (DMTMD.MD[1:0] Bits = 11b)

The DMCRAH[9:0] bits specify the block size and the DMCRAL[15:0] bits function as a 10-bit block size counter. The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat-block transfer mode, a value in the range of 000h to 3FFh can be set for the DMCRAH[9:0] and DMCRAL[15:0] bits. Setting bits 15 to 10 in the DMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the DMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the DMCRAH[9:0] bits is loaded into the DMCRAL[15:0] bits.

18.2.8 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 8700 700Ch, DMAC1.DMCRB 8700 704Ch, DMAC2.DMCRB 8700 708Ch, DMAC3.DMCRB 8700 70CCh, DMAC4.DMCRB 8700 710Ch, DMAC5.DMCRB 8700 714Ch, DMAC6.DMCRB 8700 718Ch, DMAC7.DMCRB 8700 71CCh, DMAC0_NS.DMCRB 0700 700Ch, DMAC1_NS.DMCRB 0700 704Ch, DMAC2_NS.DMCRB 0700 708Ch, DMAC3_NS.DMCRB 0700 70CCh, DMAC4_NS.DMCRB 0700 710Ch, DMAC5_NS.DMCRB 0700 714Ch, DMAC6_NS.DMCRB 0700 718Ch, DMAC7_NS.DMCRB 0700 71CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DMCRBH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMCRBL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DMCRBL[15:0]	Lower Bits of Block Transfer Count	Counts the number of block, repeat or repeat-block transfer operations	R/W
b31 to b16	DMCRBH[15:0]	Upper Bits of Block Transfer Count		R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the same value for DMCRBH[15:0] and DMCRBL[15:0] bits in repeat transfer mode, block transfer mode, and repeat-block transfer mode. DMCRBH[15:0] and DMCRBL[15:0] bits specifies and counts, respectively, the number of transfer operations in block, repeat, and repeat-block transfer mode. The number of transfer operations is one when the setting is 0001h, 65535 when it is FFFFh, and 65536 when it is 0000h. In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred. In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final block data is transferred. In normal transfer mode, do not use the DMCRB register because the setting is invalid. When DMTMD.TKP bit is 1 and the final data of one repeat size or one block size is transferred, DMCRBL[15:0] bits reloads the value of DMCRBH[15:0] bits automatically.
Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

18.2.9 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 8700 7010h, DMAC1.DMTMD 8700 7050h, DMAC2.DMTMD 8700 7090h, DMAC3.DMTMD 8700 70D0h, DMAC4.DMTMD 8700 7110h, DMAC5.DMTMD 8700 7150h, DMAC6.DMTMD 8700 7190h, DMAC7.DMTMD 8700 71D0h, DMAC0_NS.DMTMD 0700 7010h, DMAC1_NS.DMTMD 0700 7050h, DMAC2_NS.DMTMD 0700 7090h, DMAC3_NS.DMTMD 0700 70D0h, DMAC4_NS.DMTMD 0700 7110h, DMAC5_NS.DMTMD 0700 7150h, DMAC6_NS.DMTMD 0700 7190h, DMAC7_NS.DMTMD 0700 71D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	TKP	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Data Transfer Size	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: 64 bits	R/W
b10	TKP	Transfer Keeping	0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. DMAC trigger source is selected using the ICU.DMRSRm registers. For details on DMAC trigger sources, see Table 15.6, Interrupt Vector Table, Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] bits select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

TKP Bit (Transfer Keeping)

TKP bit selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

18.2.10 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 8700 7013h, DMAC1.DMINT 8700 7053h, DMAC2.DMINT 8700 7093h, DMAC3.DMINT 8700 70D3h, DMAC4.DMINT 8700 7113h, DMAC5.DMINT 8700 7153h, DMAC6.DMINT 8700 7193h, DMAC7.DMINT 8700 71D3h, DMAC0_NS.DMINT 0700 7013h, DMAC1_NS.DMINT 0700 7053h, DMAC2_NS.DMINT 0700 7093h, DMAC3_NS.DMINT 0700 70D3h, DMAC4_NS.DMINT 0700 7113h, DMAC5_NS.DMINT 0700 7153h, DMAC6_NS.DMINT 0700 7193h, DMAC7_NS.DMINT 0700 71D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DMCNT.DTE bit is set to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt due to an extended repeat area overflow on the destination address is requested. When block transfer mode is used with the extended repeat area function, an interrupt is requested when transfer of each single-block size of data is completed. When you set 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped. When the extended repeat area is not specified for the destination address, this bit is ignored. When you use repeat-block transfer mode, do not set this bit to 1.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DMCNT.DTE bit is set to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt due to an extended repeat area overflow on the source address is requested. When block transfer mode is used with the extended repeat area function, an interrupt is requested when transfer of each single-block size of data is completed. When you set 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped. When the extended repeat area is not specified for the source address, this bit is ignored. When you use repeat-block transfer mode, do not set this bit to 1.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is set to 0 when transfer of each single-repeat size of data is completed. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified). When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is set to 0 when transfer of each single-block size of data is completed in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified). When you use repeat-block transfer mode, do not set this bit to 1.

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and interrupt requests for extended repeat area overflows on the source and destination addresses) that are generated during DMA transfer. The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by setting this bit or the DMSTS.ESIF flag to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers. The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the DMSTS.DTIF flag to 0.

18.2.11 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 8700 7014h, DMAC1.DMAMD 8700 7054h, DMAC2.DMAMD 8700 7094h, DMAC3.DMAMD 8700 70D4h, DMAC4.DMAMD 8700 7114h, DMAC5.DMAMD 8700 7154h, DMAC6.DMAMD 8700 7194h, DMAC7.DMAMD 8700 71D4h, DMAC0_NS.DMAMD 0700 7014h, DMAC1_NS.DMAMD 0700 7054h, DMAC2_NS.DMAMD 0700 7094h, DMAC3_NS.DMAMD 0700 70D4h, DMAC4_NS.DMAMD 0700 7114h, DMAC5_NS.DMAMD 0700 7154h, DMAC6_NS.DMAMD 0700 7194h, DMAC7_NS.DMAMD 0700 71D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		SADR	SARA[4:0]					DM[1:0]		DADR	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area Setting	Specifies the extended repeat area on the destination address. For details on the settings, see Table 18.2.	R/W
b5	DADR	Destination Address Update Setting after Reload	0: Only reloading. 1: Add index after reloading.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode Select	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area Setting	Specifies the extended repeat area on the source address. For details on the settings, see Table 18.2.	R/W
b13	SADR	Source Address Update Setting after Reload	0: Only reloading. 1: Add index after reloading.	R/W
b15, b14	SM[1:0]	Source Address Update Mode Select	b15 b14 0 0: Source address is fixed. 0 1: Offset addition 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

DARA[4:0] Bits (Destination Address Extended Repeat Area Setting)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes. That is, the interval between settings is any power of two. When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set. When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits. In repeat-block transfer mode, write 00000b in the DARA[4:0] bits. An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

Note: When setting the extended repeat area, set the lowest bit or lower bits of DMDAR as follows according to the setting of DMTMD.SZ[1:0].
When DMTMD.SZ[1:0] = 01b (16 bits), set DMDAR[0] to 0b.

When DMTMD.SZ[1:0] = 10b (32 bits), set DMDAR[1:0] to 00b.
When DMTMD.SZ[1:0] = 11b (64 bits), set DMDAR[2:0] to 000b.

DADR Bit (Destination Address Update Setting after Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMDAR register after reloading the value of DMDRR register. When this bit is set to 1, an index value ((DMDBSH – DMDBSL) × transfer data size) is added to DMDAR register after reloading the value of DMDRR register. When this bit is set to 0, DMDAR register only reloads DMDRR register. This behavior is described in Table 18.13. In normal, repeat or block transfer mode, this bit is ignored.

DM[1:0] Bits (Destination Address Update Mode Select)

These bits select the mode of updating the destination address. When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the destination address is incremented by 1, 2, 4, and 8, respectively. When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the destination address is decremented by 1, 2, 4, and 8, respectively. When offset addition is selected, the offset specified by the DMOFR register is added to the address.

SARA[4:0] Bits (Source Address Extended Repeat Area Setting)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes. When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set. When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits. In repeat-block transfer mode, write 00000b in the SARA[4:0] bits. An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

Note: When setting the extended repeat area, set the lowest bit or lower bits of DMSAR as follows according to the setting of DMTMD.SZ[1:0].
When DMTMD.SZ[1:0] = 01b (16 bits), set DMSAR[0] to 0b.
When DMTMD.SZ[1:0] = 10b (32 bits), set DMSAR[1:0] to 00b.
When DMTMD.SZ[1:0] = 11b (64 bits), set DMSAR[2:0] to 000b.

SADR Bit (Source Address Update Setting after Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMSAR register after reloading the value of DMSRR register. When this bit is set to 1, an index value ((DMSBSH – DMSBSL) × transfer data size) is added to DMSAR register after reloading the value of DMSRR register. When this bit is set to 0, DMSAR register only reloads DMSRR register. This behavior is described in Table 18.12. In normal, repeat or block transfer mode, this bit is ignored.

SM[1:0] Bits (Source Address Update Mode Select)

These bits select the mode of updating the source address. When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the source address is incremented by 1, 2, 4, and 8, respectively. When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the source address is decremented by 1, 2, 4, and 8, respectively. When offset addition is selected, the offset specified by the DMOFR register is added to the address.

Table 18.2 Settings and Corresponding Extended Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

18.2.12 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 8700 7018h, DMAC1.DMOFR 8700 7058h, DMAC2.DMOFR 8700 7098h, DMAC3.DMOFR 8700 70D8h, DMAC4.DMOFR 8700 7118h, DMAC5.DMOFR 8700 7158h, DMAC6.DMOFR 8700 7198h, DMAC7.DMOFR 8700 71D8h, DMAC0_NS.DMOFR 0700 7018h, DMAC1_NS.DMOFR 0700 7058h, DMAC2_NS.DMOFR 0700 7098h, DMAC3_NS.DMOFR 0700 70D8h, DMAC4_NS.DMOFR 0700 7118h, DMAC5_NS.DMOFR 0700 7158h, DMAC6_NS.DMOFR 0700 7198h, DMAC7_NS.DMOFR 0700 71D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR register returns the extended value. In repeat-block transfer mode, the offset cannot be specified by DMOFR register when offset addition is selected, write 0000 0000h to DMOFR register.

18.2.13 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 8700 701Ch, DMAC1.DMCNT 8700 705Ch, DMAC2.DMCNT 8700 709Ch, DMAC3.DMCNT 8700 70DCh, DMAC4.DMCNT 8700 711Ch, DMAC5.DMCNT 8700 715Ch, DMAC6.DMCNT 8700 719Ch, DMAC7.DMCNT 8700 71DCh, DMAC0_NS.DMCNT 0700 701Ch, DMAC1_NS.DMCNT 0700 705Ch, DMAC2_NS.DMCNT 0700 709Ch, DMAC3_NS.DMCNT 0700 70DCh, DMAC4_NS.DMCNT 0700 711Ch, DMAC5_NS.DMCNT 0700 715Ch, DMAC6_NS.DMCNT 0700 719Ch, DMAC7_NS.DMCNT 0700 71DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DTE Bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel. If the DTE bit is cleared during DMA transfer, no transfer requests for the corresponding channel will be accepted after the transfer ends.

[Setting condition]

- 1 is written to this bit.

[Clearing conditions]

- 0 is written to this bit.
- The specified total volume of data transfer is completed.
- The DMA transfer is stopped by the repeat size end interrupt.
- The DMA transfer is stopped by the extended repeat area overflow interrupt.
- The DMA transfer is stopped due to the occurrence of an access error. For details, see section 18.5, Processing on DMA Transfer Error.

18.2.14 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 8700 701Dh, DMAC1.DMREQ 8700 705Dh, DMAC2.DMREQ 8700 709Dh, DMAC3.DMREQ 8700 70DDh, DMAC4.DMREQ 8700 711Dh, DMAC5.DMREQ 8700 715Dh, DMAC6.DMREQ 8700 719Dh, DMAC7.DMREQ 8700 71DDh, DMAC0_NS.DMREQ 0700 701Dh, DMAC1_NS.DMREQ 0700 705Dh, DMAC2_NS.DMREQ 0700 709Dh, DMAC3_NS.DMREQ 0700 70DDh, DMAC4_NS.DMREQ 0700 711Dh, DMAC5_NS.DMREQ 0700 715Dh, DMAC6_NS.DMREQ 0700 719Dh, DMAC7_NS.DMREQ 0700 71DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	Writing 1 to this bit triggers the generation of a DMA transfer request, and DMA transfer is started in response to the request	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Setting	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer. Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC trigger source is software). Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b. To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- 1 is written to this bit.

[Clearing conditions]

- The DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- The DMA transfer is stopped due to the occurrence of an access error. See section 18.5, Processing on DMA Transfer Error.
- 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Setting)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

18.2.15 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 8700 701Eh, DMAC1.DMSTS 8700 705Eh, DMAC2.DMSTS 8700 709Eh, DMAC3.DMSTS 8700 70DEh, DMAC4.DMSTS 8700 711Eh, DMAC5.DMSTS 8700 715Eh, DMAC6.DMSTS 8700 719Eh, DMAC7.DMSTS 8700 71DEh, DMAC0_NS.DMSTS 0700 701Eh, DMAC1_NS.DMSTS 0700 705Eh, DMAC2_NS.DMSTS 0700 709Eh, DMAC3_NS.DMSTS 0700 70DEh, DMAC4_NS.DMSTS 0700 711Eh, DMAC5_NS.DMSTS 0700 715Eh, DMAC6_NS.DMSTS 0700 719Eh, DMAC7_NS.DMSTS 0700 71DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only 0 can be written to clear the flag.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- Transfer of each single-repeat size of data is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- Transfer of each single-block size of data is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- An extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address).
- An extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- 0 is written to this bit.
- 1 is written to the DMCNT.DTE bit.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- The specified number of unit-transfers is completed in normal transfer mode (the value of the DMCRAL[15:0] bits has become 0000h on completion of transfer).
- The specified number of repeat transfer operations is completed in repeat transfer mode (the value of the DMCRBL[15:0] bits has become 0000h on completion of transfer (DMTMD.TKP = 0), or the DMCRBL[15:0] bits reload the value of the DMCRBH[15:0] bits (DMTMD.TKP = 1)).

- The specified number of blocks has been transferred in block transfer mode (the value of the DMCRBL[15:0] bits has become 0000h on completion of transfer (DMTMD.TKP = 0), or the DMCRBL[15:0] bits have been reloaded with the value of the DMCRBH[15:0] bits (DMTMD.TKP = 1)).
- The specified number of blocks has been transferred in repeat-block transfer mode (the value of the DMCRBL[15:0] bits has become 0000h on completion of transfer (DMTMD.TKP = 0), or the DMCRBL[15:0] bits have been reloaded with the value of the DMCRBH[15:0] bits (DMTMD.TKP = 1)).

[Clearing conditions]

- 0 is written to this bit.
- 1 is written to the DMCNT.DTE bit.

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- The DMAC starts data transfer.

[Clearing condition]

- The data transfer in response to one transfer request is completed.

18.2.16 DMAC Trigger Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 8700 701Fh, DMAC1.DMCSL 8700 705Fh, DMAC2.DMCSL 8700 709Fh, DMAC3.DMCSL 8700 70DFh, DMAC4.DMCSL 8700 711Fh, DMAC5.DMCSL 8700 715Fh, DMAC6.DMCSL 8700 719Fh, DMAC7.DMCSL 8700 71DFh, DMAC0_NS.DMCSL 0700 701Fh, DMAC1_NS.DMCSL 0700 705Fh, DMAC2_NS.DMCSL 0700 709Fh, DMAC3_NS.DMCSL 0700 70DFh, DMAC4_NS.DMCSL 0700 711Fh, DMAC5_NS.DMCSL 0700 715Fh, DMAC6_NS.DMCSL 0700 719Fh, DMAC7_NS.DMCSL 0700 71DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt status flag of the trigger source to 0. 1: At the end of transfer, the interrupt status flag of the trigger source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt status flag of the trigger source is set to 0 or issues an interrupt request to the CPU, at the beginning of DMA transfer. When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.

18.2.17 DMA Source Buffer Size Register (DMSBS)

Address(es): DMAC0.DMSBS 8700 7028h, DMAC1.DMSBS 8700 7068h, DMAC2.DMSBS 8700 70A8h, DMAC3.DMSBS 8700 70E8h, DMAC4.DMSBS 8700 7128h, DMAC5.DMSBS 8700 7168h, DMAC6.DMSBS 8700 71A8h, DMAC7.DMSBS 8700 71E8h, DMAC0_NS.DMSBS 0700 7028h, DMAC1_NS.DMSBS 0700 7068h, DMAC2_NS.DMSBS 0700 70A8h, DMAC3_NS.DMSBS 0700 70E8h, DMAC4_NS.DMSBS 0700 7128h, DMAC5_NS.DMSBS 0700 7168h, DMAC6_NS.DMSBS 0700 71A8h, DMAC7_NS.DMSBS 0700 71E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DMSBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMSBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DMSBSL[15:0]	Lower Bits of Source Buffer Size	Functions as data transfer counter in repeat-block transfer mode	R/W
b31 to b16	DMSBSH[15:0]	Upper Bits of Source Buffer Size	Specifies the repeat-area size in repeat-block transfer mode	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.
Set the same value for the DMSBSH[15:0] and DMSBSL[15:0] bits in repeat-block transfer mode. Write 00000000h to the DMSBS register in normal, repeat and block transfer mode. The DMSBSH[15:0] bits specify buffer size and the DMSBSL[15:0] bits function as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by the DMSBSH[15:0] bits. When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting the DMSBSH[15:0] and DMSBSL[15:0] bits to 0000h is prohibited. When final data of one buffer size is transferred, the DMSBSL[15:0] bits reload value of the DMSBSH[15:0] bits. When address update mode is fixed address, this register is ignored. Table 18.3 lists the setting values of Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

Table 18.3 Available Setting for DMSBS Register in Repeat-Block Transfer Mode

Source Address Update Mode (DMAMD.SM)	Transfer Data Size (DMTMD.SZ)	Available Setting for DMSBSH[15:0] and DMSBSL[15:0] Bits
Source address is fixed. (SM = 00b)	Don't care	0000h (DMSBS is not used)
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0001h to FFFFh (1 to 65535)
	16 bits (SZ = 01b)	0001h to 7FFFh (1 to 32767)
	32 bits (SZ = 10b)	0001h to 3FFFh (1 to 16383)
	64 bits (SZ = 11b)	0001h to 1FFFh (1 to 8191)
Source address is incremented or decremented. (SM = 1xb)	Don't care	0000h (infinite) 0001h to FFFFh (1 to 65535)

18.2.18 DMA Destination Buffer Size Register (DMDBS)

Address(es): DMAC0.DMDBS 8700 702Ch, DMAC1.DMDBS 8700 706Ch, DMAC2.DMDBS 8700 70ACh, DMAC3.DMDBS 8700 70ECh, DMAC4.DMDBS 8700 712Ch, DMAC5.DMDBS 8700 716Ch, DMAC6.DMDBS 8700 71ACh, DMAC7.DMDBS 8700 71ECh, DMAC0_NS.DMDBS 0700 702Ch, DMAC1_NS.DMDBS 0700 706Ch, DMAC2_NS.DMDBS 0700 70ACh, DMAC3_NS.DMDBS 0700 70ECh, DMAC4_NS.DMDBS 0700 712Ch, DMAC5_NS.DMDBS 0700 716Ch, DMAC6_NS.DMDBS 0700 71ACh, DMAC7_NS.DMDBS 0700 71ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMDBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DMDBSL[15:0]	Lower Bits of Destination Buffer Size	Functions as data transfer counter in repeat-block transfer mode	R/W
b31 to b16	DMDBSH[15:0]	Upper Bits of Destination Buffer Size	Specifies the repeat-area size in repeat-block transfer mode	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.
Set the same value for the DMDBSH[15:0] and DMDBSL[15:0] bits in repeat-block transfer mode. Write 00000000h to the DMDBS register in normal, repeat and block transfer mode. The DMDBSH[15:0] bits specify buffer size and the DMDBSL[15:0] bits function as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by the DMDBSH[15:0] bits. When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting the DMDBSH[15:0] and DMDBSL[15:0] bits to 0000h is prohibited. When final data of one buffer size is transferred, the DMDBSL[15:0] bits reload value of the DMDBSH[15:0] bits. When address update mode is fixed address, this register is ignored. Table 18.4 lists the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

Table 18.4 Available Setting for DMDBS Register in Repeat-Block Transfer Mode

Destination Address Update Mode (DMAMD.DM)	Transfer Data Size (DMTMD.SZ)	Available Setting for DMDBSH[15:0] and DMDBSL[15:0] Bits
Destination address is fixed. (DM = 00b)	Don't care	0000h (DMDBS is not used)
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0001h to FFFFh (1 to 65535)
	16 bits (SZ = 01b)	0001h to 7FFFh (1 to 32767)
	32 bits (SZ = 10b)	0001h to 3FFFh (1 to 16383)
	64 bits (SZ = 11b)	0001h to 1FFFh (1 to 8191)
Destination address is incremented or decremented. (DM = 1xb)	Don't care	0000h (infinite) 0001h to FFFFh (1 to 65535)

18.2.19 DMA Bufferable Write Enable Register (DMBWR)

Address(es): DMAC0.DMBWR 8700 7030h, DMAC1.DMBWR 8700 7070h, DMAC2.DMBWR 8700 70B0h, DMAC3.DMBWR 8700 70F0h, DMAC4.DMBWR 8700 7130h, DMAC5.DMBWR 8700 7170h, DMAC6.DMBWR 8700 71B0h, DMAC7.DMBWR 8700 71F0h, DMAC0_NS.DMBWR 0700 7030h, DMAC1_NS.DMBWR 0700 7070h, DMAC2_NS.DMBWR 0700 70B0h, DMAC3_NS.DMBWR 0700 70F0h, DMAC4_NS.DMBWR 0700 7130h, DMAC5_NS.DMBWR 0700 7170h, DMAC6_NS.DMBWR 0700 71B0h, DMAC7_NS.DMBWR 0700 71F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BWE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BWE	Bufferable Write Enable	0: Writing to bufferable areas bypasses the write buffers. 1: Writing to bufferable areas proceeds through the write buffers.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

BWE Bit (Bufferable Write Enable)

This bit enables or disables buffered writing to bufferable areas. When the setting of this bit is 1 (enabling use of the buffers), even if the write access by the DMAC is completed, the actual write operation might not have been finished. The DMAC treats writing as complete once it has written to a write buffer, even though writing to the destination may not yet be finished. For details, see section 16.2.5.2, Bufferable Write and section 16.2.5.3, Notes on the Timing of Writing Completion. If an error occurs during writing to a slave that is capable of supporting buffered write operations, this is detected as a buffered write error, triggering an interrupt or reset. However, the DMA transfer does not stop at such times.

18.2.20 DMAC Module Start Register (DMAST)

Address(es): DMAC.DMAST 8700 7800h,
DMAC_NS.DMAST 0700 7800h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Module Start	0: DMAC module stop 1: DMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DMST Bit (DMAC Module Start)

When this bit is set to 1, DMAC is ready to accept transfer requests for all channels. When 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC module start), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time. When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer can be resumed by setting the DMST bit to 1 again.

[Setting condition]

- 1 is written to this bit.

[Clearing condition]

- 0 is written to this bit.

18.2.21 DMAC Error Channel Register (DMECHR)

Address(es): DMAC.DMECHR 8700 7840h, DMAC_NS.DMECHR 0700 7840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMESTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DMERRW	—	—	—	DMECHSAM	—	—	—	—	DMECH[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DMECH[3:0]	DMAC Error Channel Flag	Indicates the channel number causing the error 0000: Error occurred on channel 0 0001: Error occurred on channel 1 0010: Error occurred on channel 2 : : 0111: Error occurred on channel 7	R
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	DMECHSAM	DMAC Error Channel Security Attribution Monitor Flag	Indicates the security attribution of a channel causing the error 0: Secure channel 1: Non-secure channel	R
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	DMERRW	DMAC Error Access Read/Write Attribute Flag	Indicates the read/write attribute of the error access 0: Read access 1: Write access	R
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DMESTA	DMAC Error Status Flag	0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W*1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing to the DMESTA flag depends on the value of the DMECHSAM flag.

When reading this register, it can be accessed from both secure and non-secure access. When writing this register, it depends on the value of the DMECHR.DMECHSAM flag.

- When the value of the DMECHR.DMECHSAM flag is 1, it can be accessed from secure and non-secure access.
- When the value of the DMECHR.DMECHSAM flag is 0, it can be accessed from secure. An error is returned when write access is performed in a non-secure access.

This register is cleared by a reset. On debugging a program, select an interrupt instead of a reset in the corresponding bits of the registers listed in the interrupt/reset selection column of Table 18.20 in section 18.6.2, Transfer Error Interrupt.

DMECH[3:0] Flag (DMAC Error Channel Flag)

When a transfer error due to DMA transfer occurs, this flag stores the channel number of the DMAC that was violated. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- A DMA transfer error occurs while the DMESTA flag is 0.

[Clearing condition]

- 1 is written to the DMESTA flag.

DMECHSAM Flag (DMAC Error Channel Security Attribution Monitor Flag)

When a transfer error due to DMA transfer occurs, this flag indicates the security attribution of the violating DMAC channel. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- A DMA transfer error occurs while the DMESTA flag is 0.

[Clearing condition]

- 1 is written to DMESTA flag.

DMERRW Flag (DMAC Error Access Read/Write Attribute Flag)

Indicates the read/write attribute of the error access.

[Setting condition]

- A DMA transfer error occurs during write access to the on-chip memory or registers of other modules.

[Clearing condition]

- 1 is written to the DMESTA flag.

DMESTA Flag (DMAC Error Status Flag)

Indicates whether or not a DMA transfer error occurred. DMECH[3:0] bits, DMECHSAM bit, DMESTA flag are cleared by writing 1 to DMESTA flag. Writing 0 to DMESTA flag is ignored. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- A DMA transfer error occurs.

[Clearing condition]

- 1 is written to the DMESTA flag.

Note: When DMECHSAM = 1, it can be cleared in the secure state and non-secure state. When DMECHSAM = 0, it cannot be cleared in the non-secure state.

18.2.22 DMAC74 Interrupt Status Monitor Register (DMIST)

Address(es): DMAC.DMIST 8700 7804h,
DMAC_NS.DMIST 0700 7804h

	b7	b6	b5	b4	b3	b2	b1	b0
	DMIS7	DMIS6	DMIS5	DMIS4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DMIS4	DMAC4 Interrupt Status Flag	0: DMAC4 interrupt is not requested. 1: DMAC4 interrupt is requested.	R
b5	DMIS5	DMAC5 Interrupt Status Flag	0: DMAC5 interrupt is not requested. 1: DMAC5 interrupt is requested.	R
b6	DMIS6	DMAC6 Interrupt Status Flag	0: DMAC6 interrupt is not requested. 1: DMAC6 interrupt is requested.	R
b7	DMIS7	DMAC7 Interrupt Status Flag	0: DMAC7 interrupt is not requested. 1: DMAC7 interrupt is requested.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: The security attribute depends on the setting of the DMACCHSAR.SADMAC4 bit.

DMISm Flag (DMACm Interrupt Status Flag) (m = 4 to 7)

This flag monitors the DMACm interrupt request. Writing to this flag will be ignored. While the DMACm.DMINT.DTIE bit is 1 and the DMACm.DMSTS.DTIF flag is 1, or the DMACm.DMINT.ESIE bit is 1 and the DMACm.DMSTS.ESIF flag is 1, the DMIST.DMISm flag is set to 1.

18.2.23 DMAC Control Register (DMCTL)

Address(es): DMAC.DMCTL 8700 7810h,
DMAC_NS.DMCTL 0700 7810h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ERCH	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PR	Priority Control Select	0: Fixed priority 1: Round-robin	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	ERCH	Clear Channel Select	0: Target channel only 1: All channels	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Do not write to the register for a given channel while either the DMSTS.ACT flag is 1 (DMA transfer in progress) or the DMCNT.DTE bit is 1 (enabling DMA transfer) for the same channel.

PR Bit (Priority Control Select)

When the DMCTL.PR bit is set to 0, arbitration is performed in the fixed priority mode of channel 0 > channel 1 > channel 2 > channel 3 …> channel 7 (channel 0: highest priority) > DTC and DMA transfer is started. When the DMCTL.PR bit is set to 1, arbitration is performed in the round-robin mode and DMA transfer is started.

ERCH Bit (Clear Channel Select)

When the DMCTL.ERCH bit is set to 0, the ICU clears all bits of the target channel of DMRSRm register (m = 0 to 7) by the error response during transfer. DMRSRm register (m = 0 to 7) other than the target channel is not cleared. When the DMCTL.ERCH bit is set to 1, the ICU clears all bits of all channels of DMRSRm register (m = 0 to 7) by the error response during transfer.

18.3 Operation

18.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL[15:0] bits. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Setting the DMCRB register is invalid in normal transfer mode. Except in free-running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations. Table 18.5 lists the register update operation in normal transfer mode, and Figure 18.2 shows an operation example in normal transfer mode.

Table 18.5 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
DMCRA.DMCRAL[15:0]	Transfer counter	Decremented by one/not updated (in free-running function)
DMCRA.DMCRAH[9:0]	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)

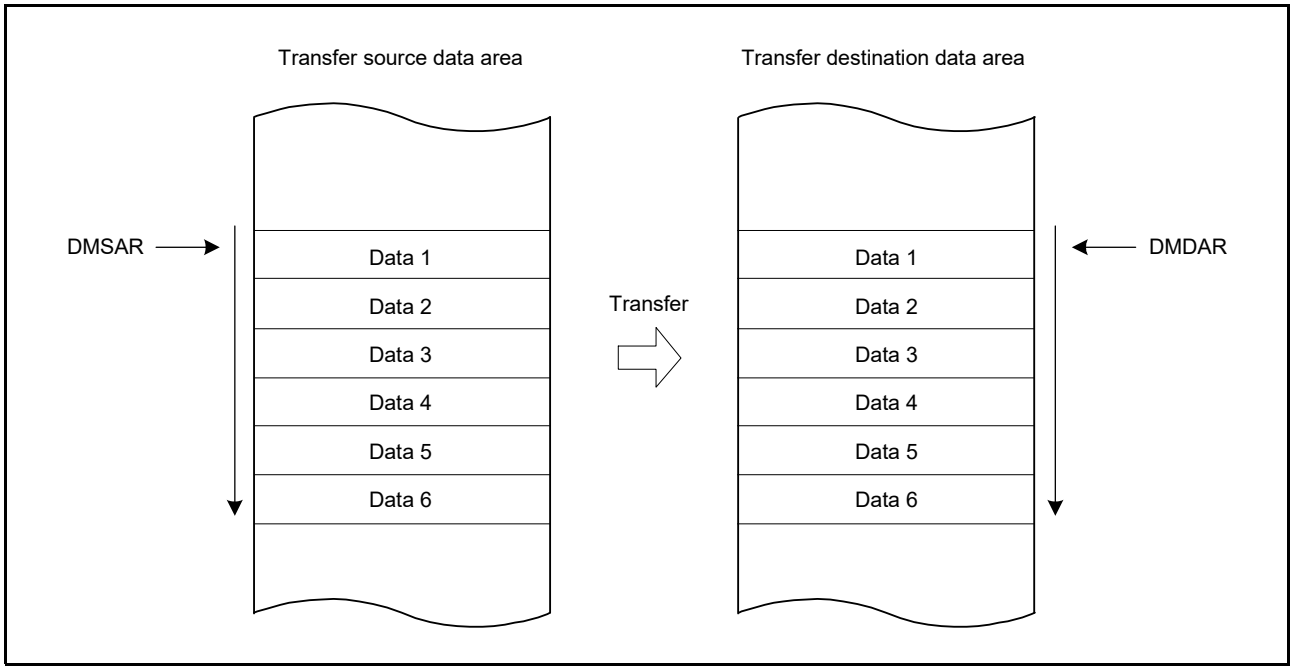


Figure 18.2 Operation Example in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request. A maximum of 1K data can be set as a total repeat transfer size using the DMCRA register. A maximum of 64K can be set as the number of repeat transfer operations using the DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size. Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling. A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations. Table 18.6 lists the register update operation in repeat transfer mode, and Figure 18.3 shows an operation example in repeat transfer mode.

Table 18.6 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When the DMCRA.DMCRAL[15:0] bits are not 1	When the DMCRA.DMCRAL[15:0] bits are 1 (Transfer of the Last Data in Repeat Size)
DMSAR	Transfer source address	Increment/decrement/fixed/ offset addition	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/ offset addition	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRA.DMCRAH [9:0]	Repeat size	Not updated	Not updated
DMCRA.DMCRAL [15:0]	Transfer counter	Decremented by one	DMCRA.DMCRAH[9:0]
DMCRB.DMCRBH [15:0]	Number of repeat transfer operations	Not updated	Not updated
DMCRB.DMCRBL [15:0]	Count of repeat transfer operations	Not updated	Decremented by one

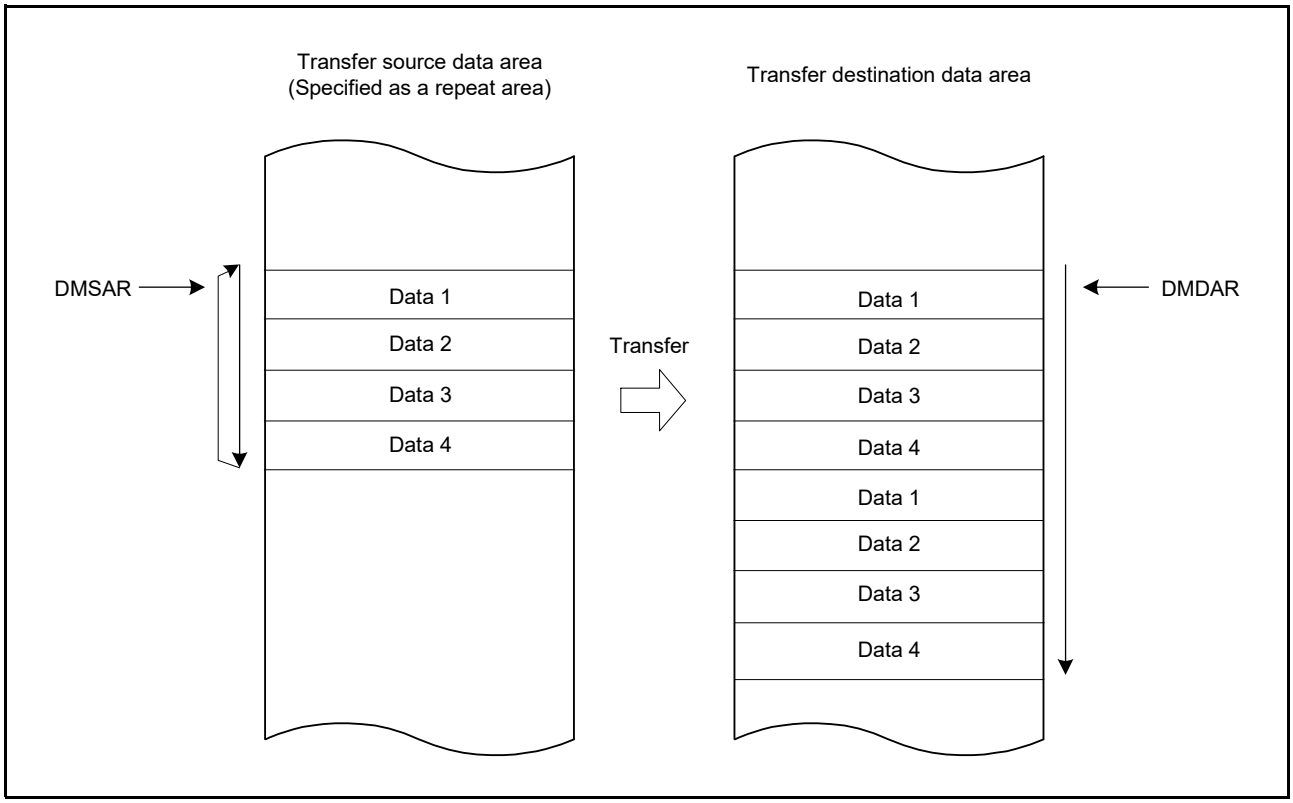


Figure 18.3 Operation Example in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request. A maximum of 1K data can be set as a total block transfer size using the DMCRA register. A maximum of 64K can be set as the number of block transfer operations using the DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size. Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling. Transfer end interrupt request can be generated after completion of the specified number of block transfer operations. Table 18.7 lists the register update operation in block transfer mode, and Figure 18.4 shows an operation example in block transfer mode.

Table 18.7 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMSAR	Transfer source address	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRA.DMCRAH[9:0]	Block size	Not updated
DMCRA.DMCRAL[15:0]	Transfer count	DMCRA.DMCRAH[9:0]
DMCRB.DMCRBH[15:0]	Number of block transfer operations	Not updated
DMCRB.DMCRBL[15:0]	Count of block transfer operations	Decremented by one

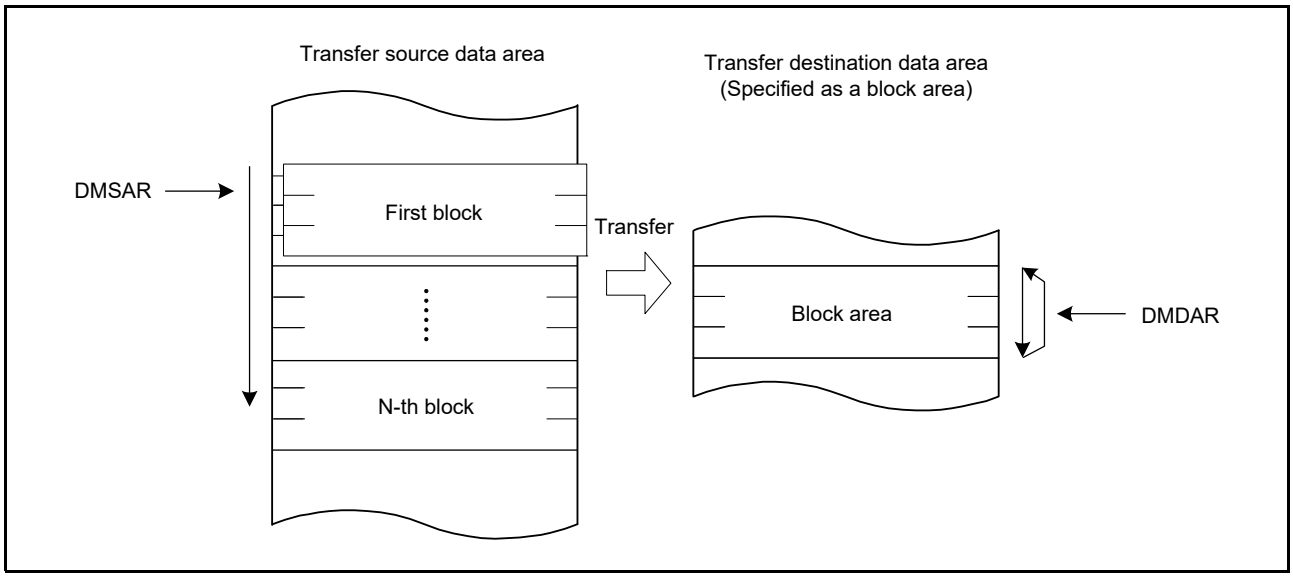


Figure 18.4 Operation Example in Block Transfer Mode

(4) Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.
Repeat function: Added function (ring buffer) to repeat specified address area.
Offset function: Multiple areas with offset can be specified within one block transfer.
The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer. Figure 18.5 shows an example of adding a repeat function to the transfer destination. Figure 18.6 shows repeat-block transfer with an offset to the transfer destination. In repeat-block transfer mode, a single block data is transferred by one transfer request. A maximum of 1K data can be set as a total block transfer size using DMCRA register. A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size. Generation of a transfer end interrupt request after completion of the specified number of block-transfer operations is selectable.

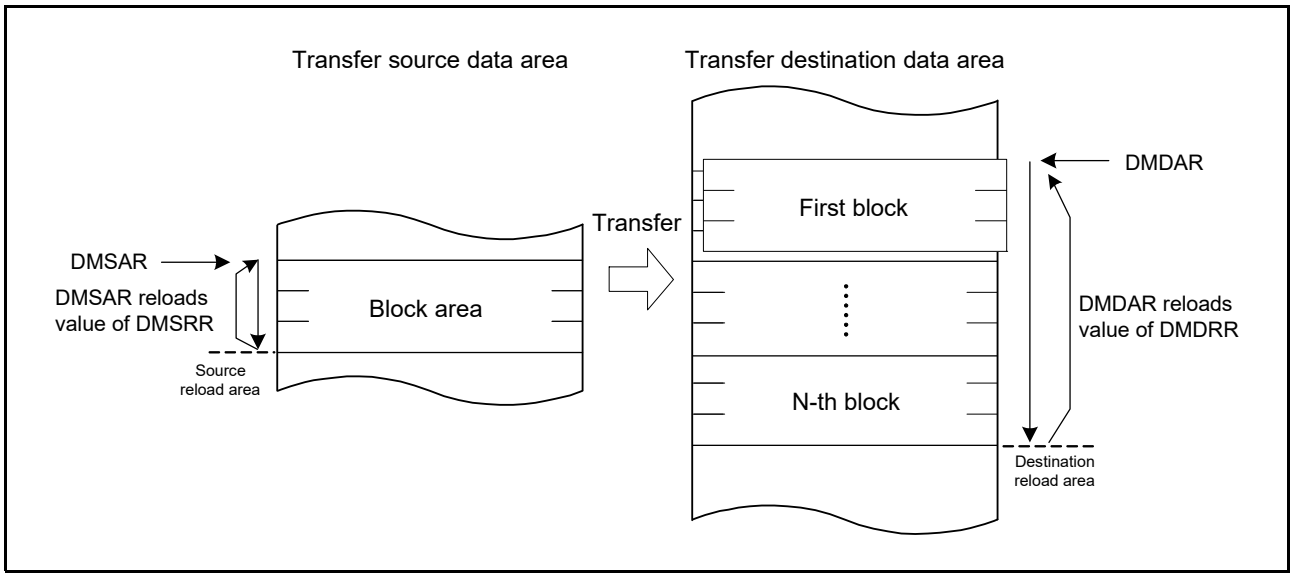


Figure 18.5 Operation Example in Repeat-Block Transfer Mode

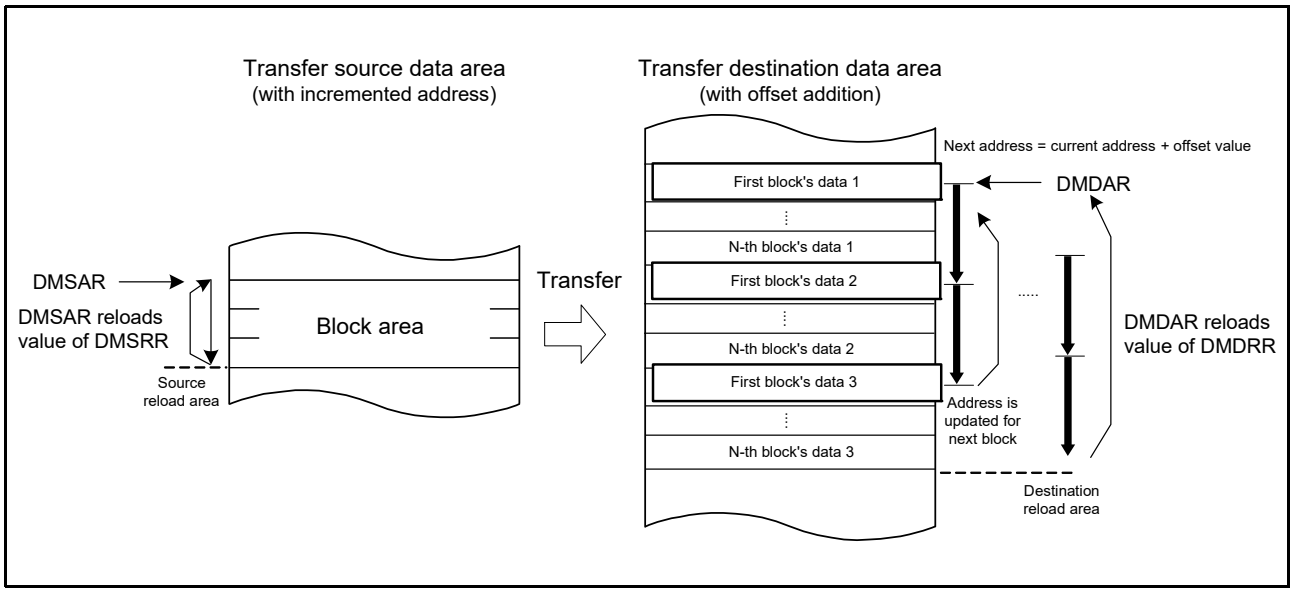


Figure 18.6 Operation Example in Repeat-Block Transfer Mode with Offset Addition

Table 18.8 to Table 18.13 list the register update operations in repeat-block transfer mode.

Table 18.8 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Fixed Address DMAMD.SM[1:0] = 00b)

		Update operation after single data is transferred		
			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)	
Register	Function	DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated
DMSAR	Transfer source address	Not updated	Not updated	Not updated
DMCRA.DMCRAH [9:0]	Block size	Not updated	Not updated	Not updated
DMCRA.DMCRAL [15:0]	Block size count	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMCRB.DMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRB.DMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

Table 18.9 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Fixed Address DMAMD.DM[1:0] = 00b)

		Update operation after single data is transferred		
			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)	
Register	Function	DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Not updated	Not updated	Not updated
DMCRA.DMCRAH [9:0]	Block size	Not updated	Not updated	Not updated
DMCRA.DMCRAL [15:0]	Block size count	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMCRB.DMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRB.DMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

Table 18.10 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Incremented or Decremented Address DMAMD.SM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		DMSBS.DMSBSL[15:0] is not 1			DMSBS.DMSBSL[15:0] is 1		
			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)	
		DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1	DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSAR	Transfer source address	DMTMD.SM[1:0] = 10b Incremented by Data Size DMTMD.SM[1:0] = 11b Decremented by Data Size	DMTMD.SM[1:0] = 10b Incremented by Data Size DMTMD.SM[1:0] = 11b Decremented by Data Size	DMTMD.SM[1:0] = 10b Incremented by Data Size DMTMD.SM[1:0] = 11b Decremented by Data Size	DMSRR	DMSRR	DMSRR
DMCRA.DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRA.DMCRAL[15:0]	Block size count	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMSBS.DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBS.DMSBSL[15:0]	Count of transfer data in source buffer	Decremented by one	Decremented by one	Decremented by one	DMSBS.DMSBSH[15:0]	DMSBS.DMSBSH[15:0]	DMSBS.DMSBSH[15:0]
DMCRB.DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRB.DMCRBL[15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

Table 18.11 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Incremented or Decremented Address DMAMD.DM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		DMSBS.DMSBSL[15:0] is not 1			DMSBS.DMSBSL[15:0] is 1		
			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)			DMCRA.DMCRAL[15:0] is 1 (single block is transferred)	
		DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1	DMCRA.DMCRAL[15:0] is not 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	DMTMD.DM[1:0] = 10b Incremented by Data Size DMTMD.DM[1:0] = 11b Decremented by Data Size	DMTMD.DM[1:0] = 10b Incremented by Data Size DMTMD.DM[1:0] = 11b Decremented by Data Size	DMTMD.DM[1:0] = 10b Incremented by Data Size DMTMD.DM[1:0] = 11b Decremented by Data Size	DMDRR	DMDRR	DMDRR
DMCRA.DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRA.DMCRAL[15:0]	Block size count	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]	Decremented by one	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMDBS.DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBS.DMDBSL[15:0]	Count of transfer data in destination buffer	Decremented by one	Decremented by one	Decremented by one	DMDBS.DMDBSH[15:0]	DMDBS.DMDBSH[15:0]	DMDBS.DMDBSH[15:0]
DMCRB.DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRB.DMCRBL[15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

Table 18.12 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Offset Addition DMAMD.SM[1:0] = 01b)

Register	Function	DMCRA.DMCRAL [15:0] is not 1	DMCRA.DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBS.DMSBSL[15:0] is not 1		DMSBS.DMSBSL[15:0] is 1	
			DMCRB.DMCRBL [15:0] is not 1	DMCRB.DMCRBL [15:0] is 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMSAR	Transfer source address	Offset addition by DMSBS.DMSBSH [15:0]	DMAMD.SADR = 0 DMSRR DMAMD.SADR = 1 DMSRR + (DMSBSH – DMSBSL) × transfer data size	DMAMD.SADR = 0 DMSRR DMAMD.SADR = 1 DMSRR + (DMSBSH – DMSBSL) × transfer data size	DMSRR	DMSRR
DMCRA. DMCRAH [9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRA. DMCRAL [15:0]	Block size count	Decremented by one	DMCRA.DMCRAH [9:0]	DMCRA.DMCRAH [9:0]	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMSBS. DMSBSH [15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBS. DMSBSL [15:0]	Count of transfer data in source buffer	Not updated	Decremented by one	Decremented by one	DMSBS.DMSBSH[15:0]	DMSBS.DMSBSH[15:0]
DMCRB. DMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRB. DMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

Table 18.13 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Offset Addition DMAMD.DM[1:0] = 01b)

Register	Function	DMCRA.DMCRAL [15:0] is not 1	DMCRA.DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBS.DMSBSL[15:0] is not 1		DMSBS.DMSBSL[15:0] is 1	
			DMCRB.DMCRBL [15:0] is not 1	DMCRB.DMCRBL [15:0] is 1	DMCRB.DMCRBL[15:0] is not 1	DMCRB.DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Offset addition by DMDBS.DMDBSH [15:0]	DMAMD.DADR = 0 DMDRR DMAMD.DADR = 1 DMDRR + (DMDBSH – DMDBSL) × transfer data size	DMAMD.DADR = 0 DMDRR DMAMD.DADR = 1 DMDRR + (DMDBSH – DMDBSL) × transfer data size	DMDRR	DMDRR
DMCRA. DMCRAH [9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRA. DMCRAL [15:0]	Block size count	Decremented by one	DMCRA.DMCRAH [9:0]	DMCRA.DMCRAH [9:0]	DMCRA.DMCRAH[9:0]	DMCRA.DMCRAH[9:0]
DMDBS. DMDBSH [15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBS. DMDBSL [15:0]	Count of transfer data in destination buffer	Not updated	Decremented by one	Decremented by one	DMDBS.DMDBSH[15:0]	DMDBS.DMDBSH[15:0]
DMCRB. DMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRB. DMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]	Decremented by one	DMTMD.TKP = 0 0 DMTMD.TKP = 1 DMCRB.DMCRBH[15:0]

For more information about address update operation in repeat-block transfer mode, see section 18.3.5, Address-Update Function in Repeat-Block Transfer Mode.

18.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas. The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR). The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides. However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area. When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt due to an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is set to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt due to an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling. Figure 18.7 shows an example of the extended repeat area operation.

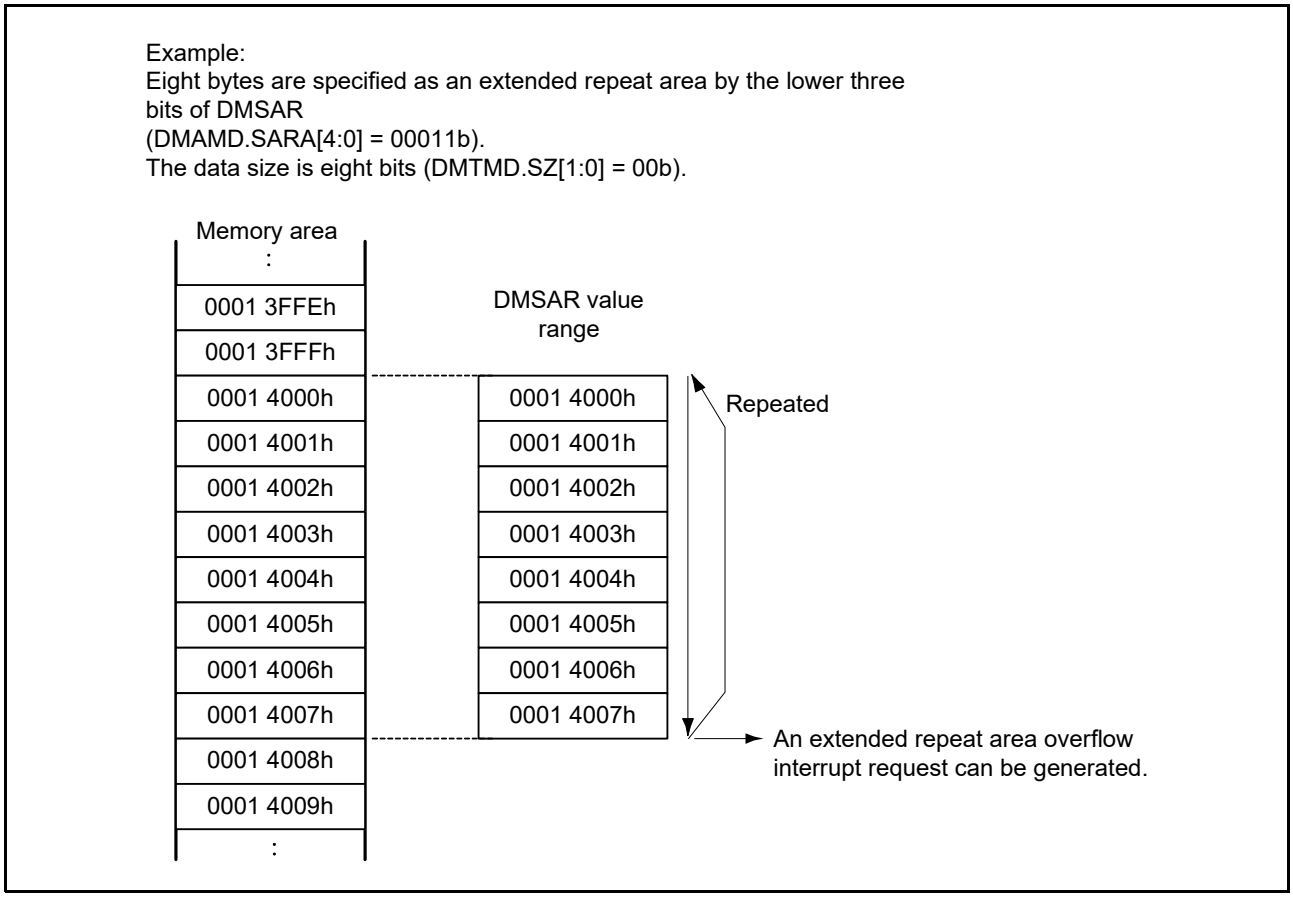


Figure 18.7 Example of Extended Repeat Area Operation

When an interrupt due to an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration. When a transfer is stopped by an interrupt due to an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt due to the overflow is suspended until transfer of the block is completed, and the transfer overruns. Figure 18.8 shows an example when the extended repeat area function is used in block transfer mode.

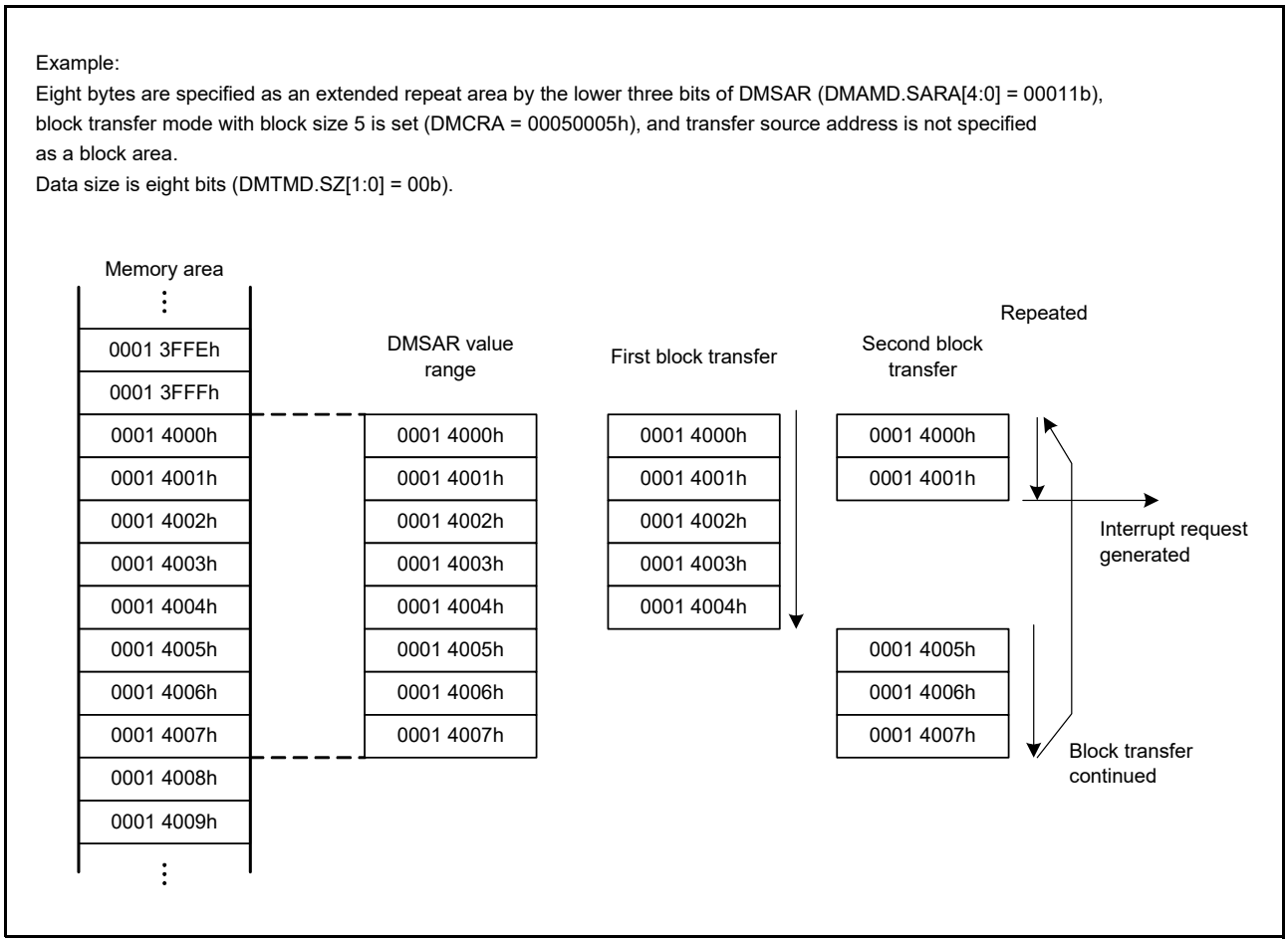


Figure 18.8 Example of Extended Repeat Area Function in Block Transfer Mode

18.3.3 Free-Running Function

The DMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

(1) In Normal Transfer Mode

In normal transfer mode, when DMCRA.DMCRAL[15:0] bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped. For more information, see section 18.3.1, Transfer Mode.

(2) In Other Transfer Modes

In repeat, block and repeat-block transfer mode, the DMAC supports free-running function using the DMTMD.TKP bit. If the DMTMD.TKP bit is set to 1, the transfer is not stopped by completion of specified total number of transfer operations and the value of the DMCRBH[9:0] bits is reloaded repeatedly. Figure 18.9 shows an example of block transfer operation without free-running function.

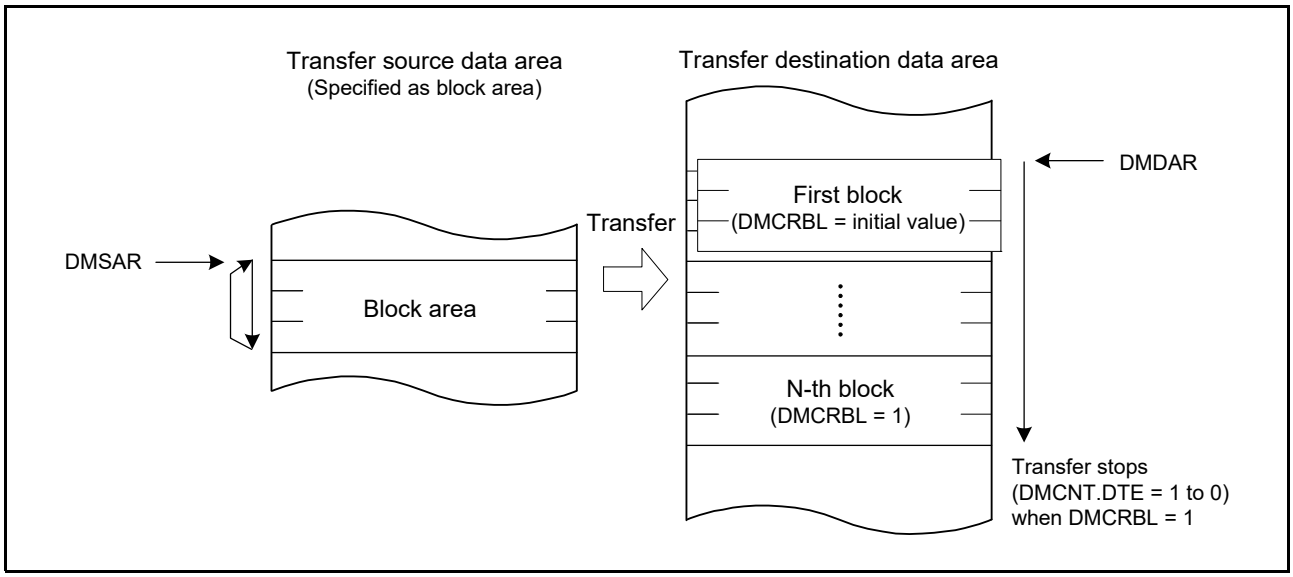


Figure 18.9 Operation in Block Transfer Mode with DMTMD.TKP = 0

Figure 18.10 shows an example of block transfer operation with free-running function.

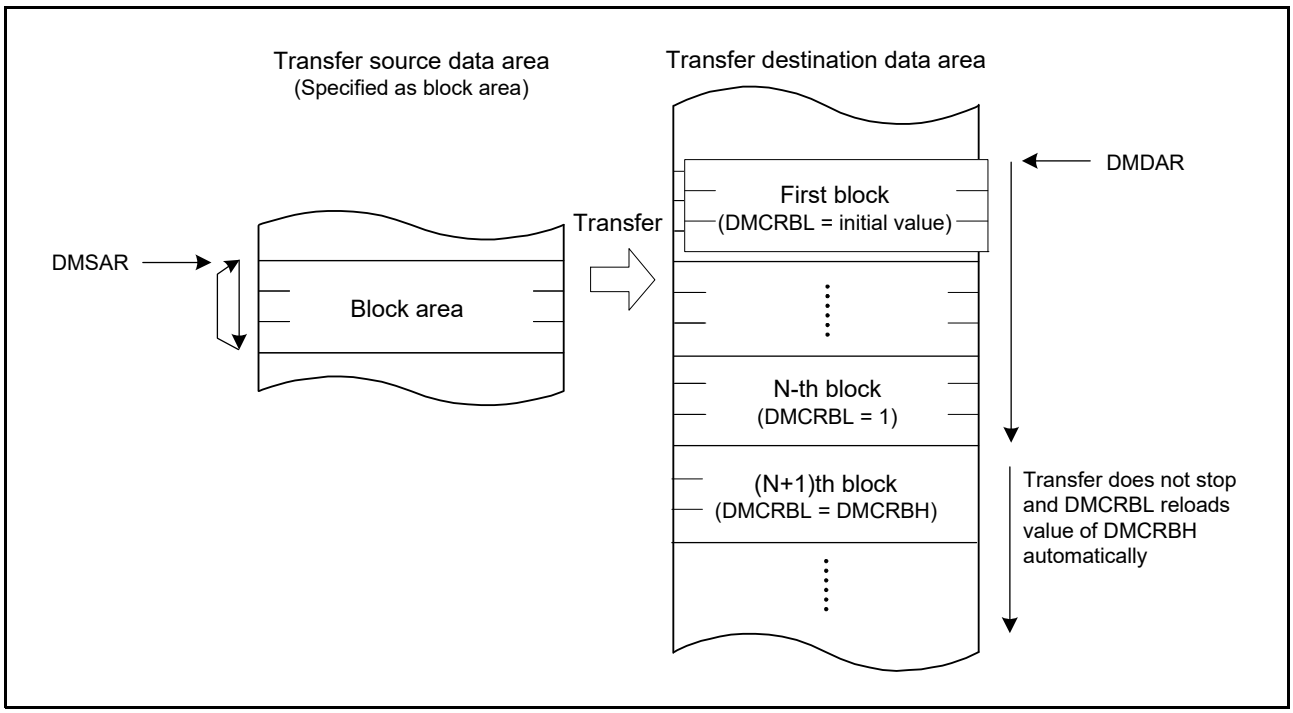


Figure 18.10 Operation in Block Transfer Mode with DMTMD.TKP = 1

18.3.4 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas. Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2's complement. DMSBS or DMDBS register are used instead of DMOFR register in repeat-block transfer mode. For more information, see section (4), Repeat-Block Transfer Mode and section 18.3.5, Address-Update Function in Repeat-Block Transfer Mode. Table 18.14 lists the address update method in each address update mode.

Table 18.14 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different DMTMD.SZ[1:0] Settings)			
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b	SZ[1:0] = 11b
Address fixed	00b	Fixed			
Offset addition	01b	+DMOFR*1			
Increment	10b	+1	+2	+4	+8
Decrement	11b	–1	–2	–4	–8

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: Bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 18.11 shows an example of address updating using offset addition.

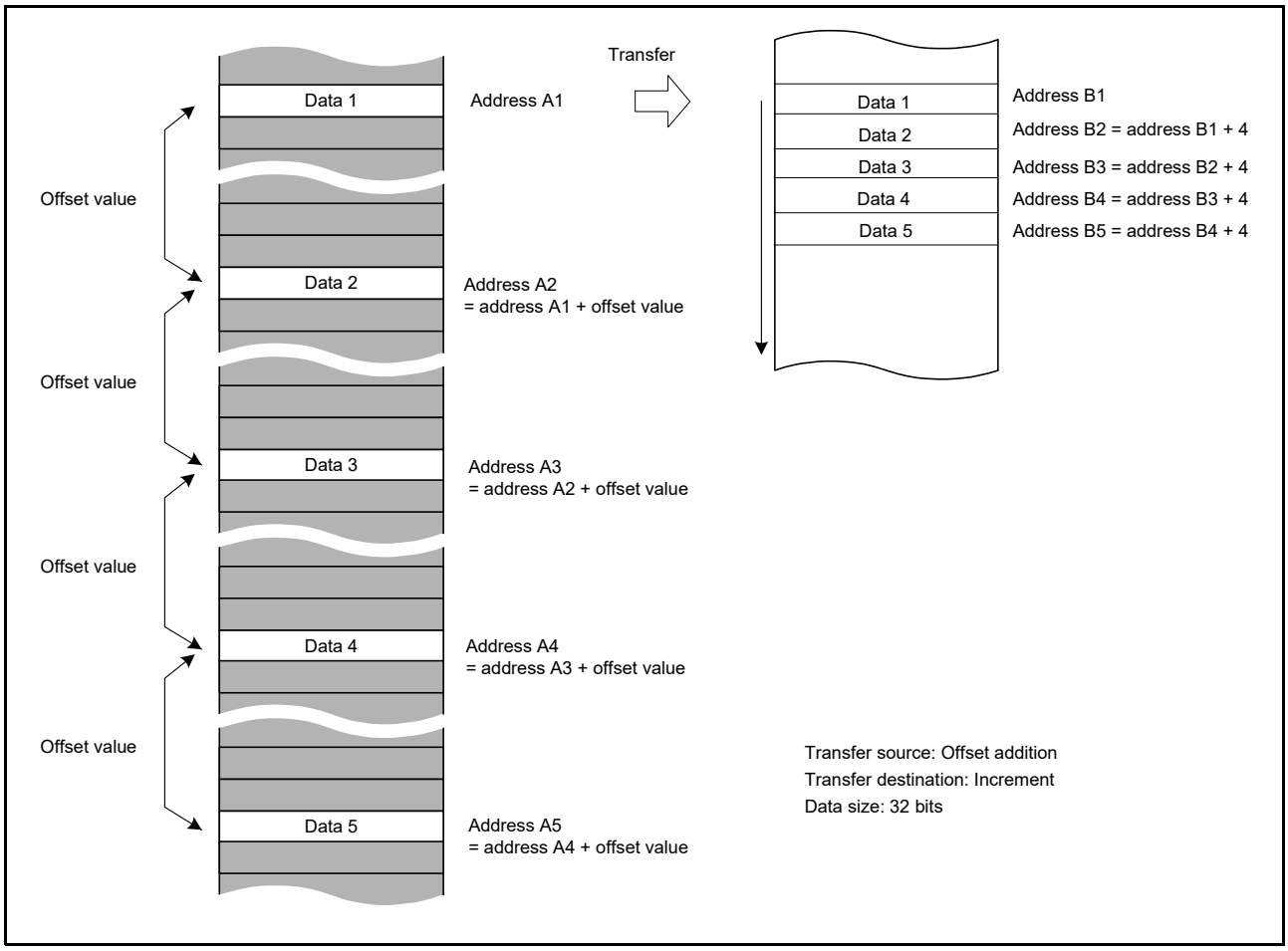


Figure 18.11 Example of Address Updating by Offset Addition

In Figure 18.11, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 18.12 shows the XY conversion using offset addition in repeat transfer mode. Settings are as follows:

- DMAMD.SM[1:0]: Source address update mode select: Offset addition
- DMAMD.DM[1:0]: Destination address update mode select: Destination address is incremented.
- DMTMD.SZ[1:0]: Data transfer size: 32 bits
- DMTMD.MD[1:0]: Transfer mode select: Repeat transfer
- DMTMD.DTS[1:0]: Repeat area select: The source is specified as the repeat area.
- DMOFR: Offset address: 10h
- DMCRA.DMCRAH[9:0]: Repeat size: 4h
- DMINT.RPTIE: The repeat size end interrupt is enabled.

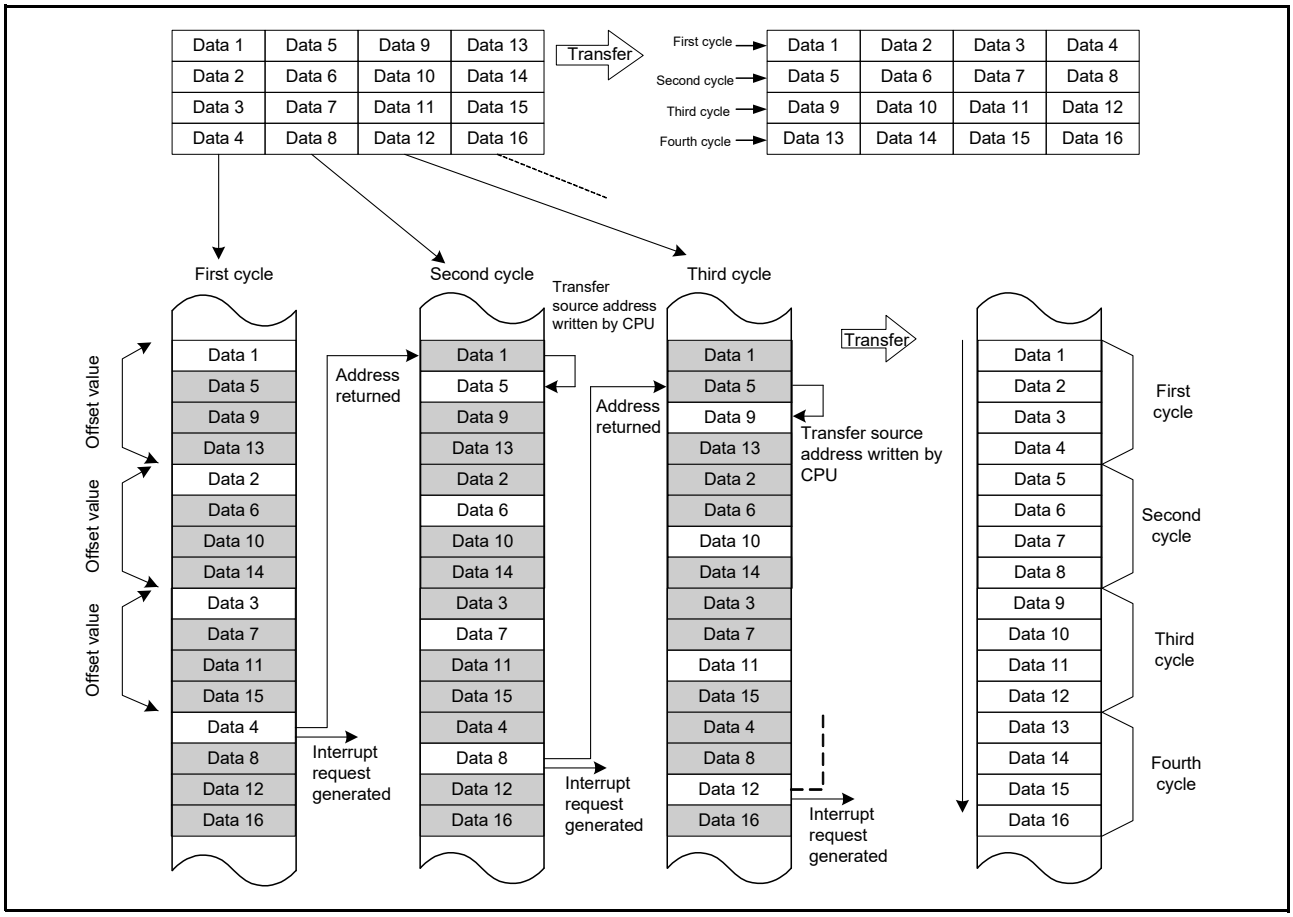


Figure 18.12 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMSAR register: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMCNT register: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 18.13 shows a flowchart of the XY conversion.

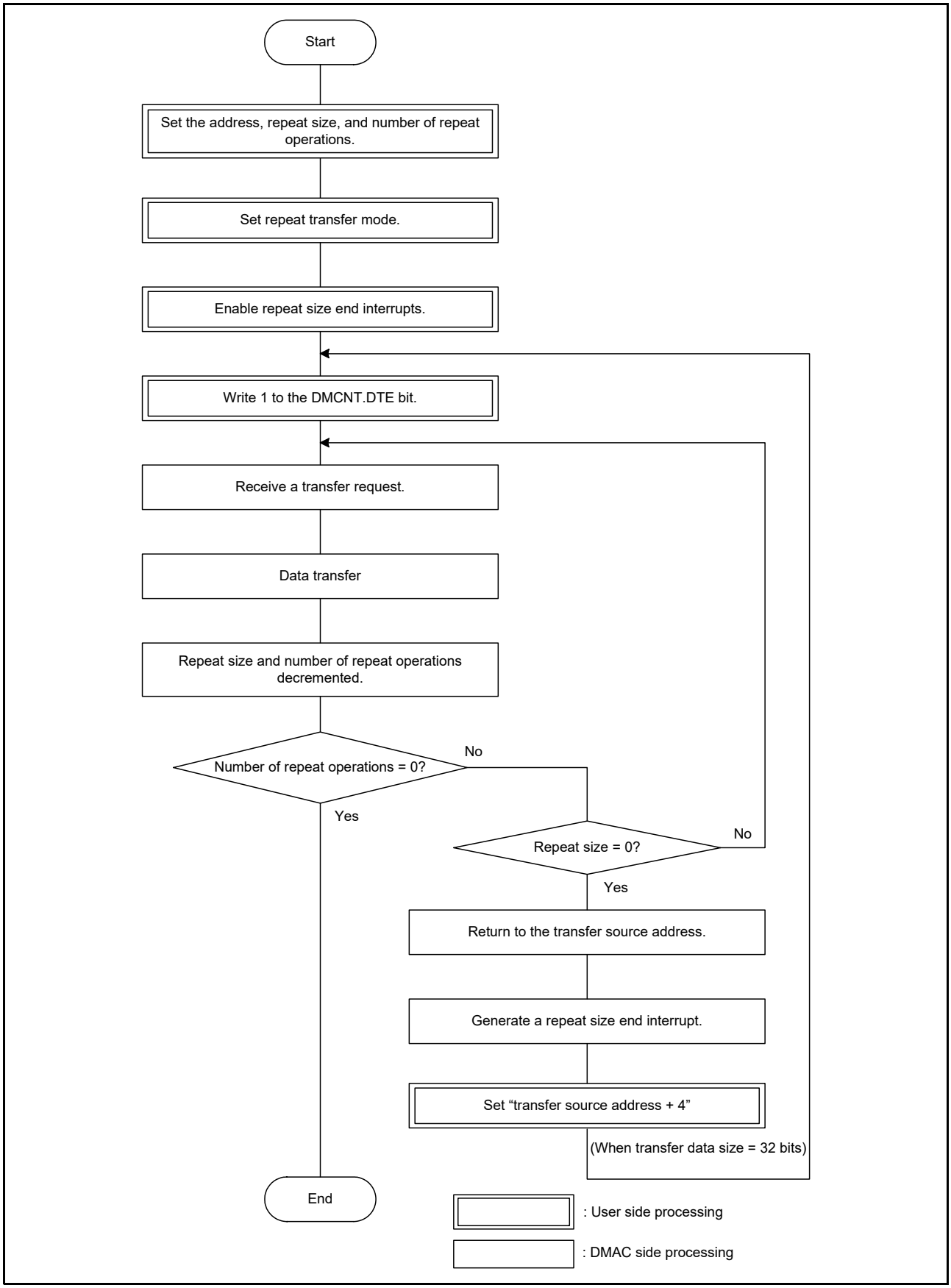


Figure 18.13 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

18.3.5 Address-Update Function in Repeat-Block Transfer Mode

The repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior in address updating differs from the behavior in the two other modes. This section describes the address-update function in the repeat-block transfer mode.

(1) Fixed Address Mode

When DMAMD.SM[1:0] bits is set to 00b, the address update mode of the source is fixed address. And when DMAMD.DM[1:0] bits is set to 00b, the address update mode of the destination is fixed address. In fixed address, the address is not updated from the initial value of DMSAR and DMDAR registers. If the block size (DMCRA) is larger than 1, the same data will be transferred multiple times for one request. Figure 18.14 shows address update in fixed address.

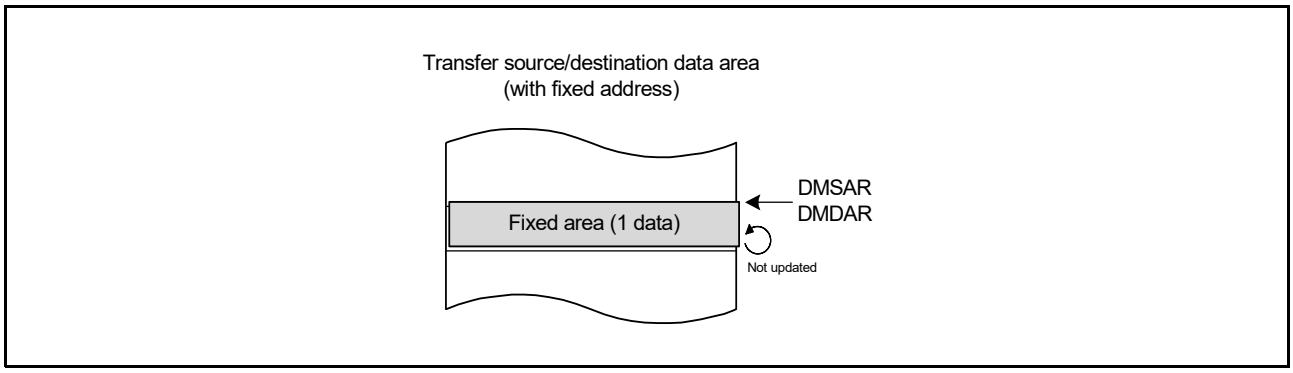


Figure 18.14 Address Update in Fixed Address

(2) Incremental and Decremental Address Mode

When DMAMD.SM[1:0] bits is set to 10b, the address update mode of the source is incremental address. And when DMAMD.DM[1:0] bits is set to 10b, the address update mode of the destination is incremental address. When DMAMD.SM[1:0] bits is set to 11b, the address update mode of the source is decremental address. And when DMAMD.DM[1:0] bits is set to 11b, the address update mode of the destination is decremental address. In these update modes, the address is incremented or decremented according to the setting of DMTMD.SZ[1:0] bits. In these update modes DMSBS and DMDBS registers indicates a reload area. The unit of DMSBS and DMDBS registers is “number of data”. At the start of transfer, DMSBSL and DMDBSL registers, which is the lower 16 bits of DMSBS and DMDBS registers, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, DMSAR and DMDAR registers reloads the value of DMSRR and DMDRR registers. Figure 18.15 shows address update in incremental address.

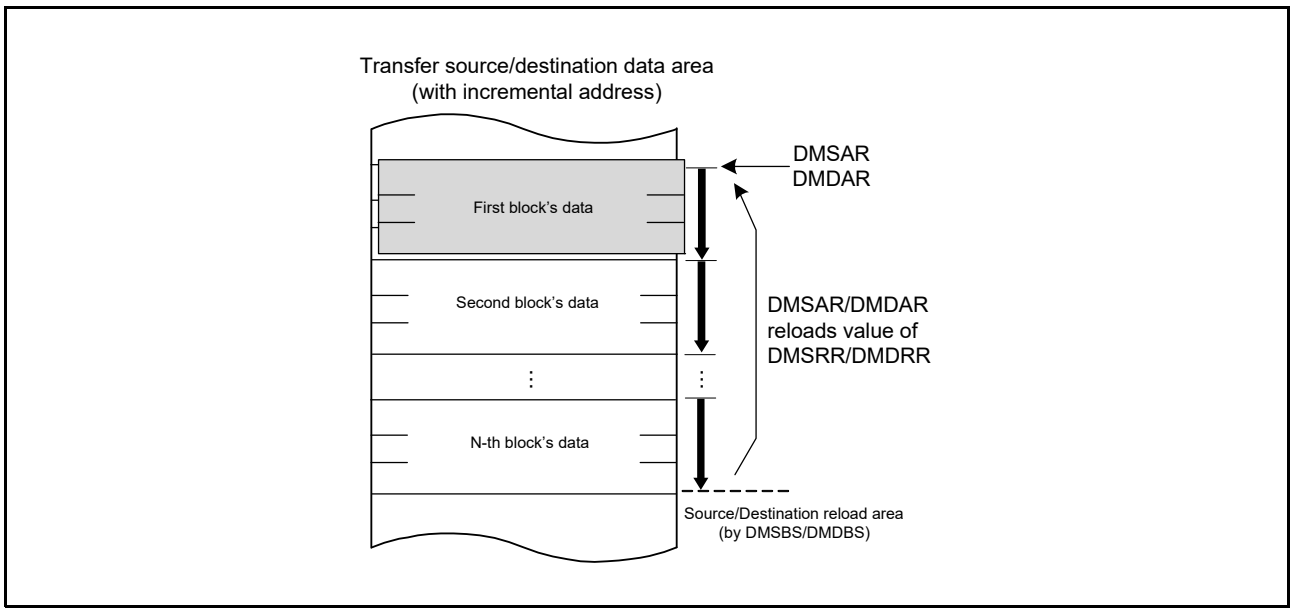


Figure 18.15 Address Update in Incremental Address

(3) Offset Addition Mode

When DMAMD.SM[1:0] bits is set to 01b, the address update mode of the source is offset addition. And when DMAMD.DM[1:0] bits is set to 01b, the address update mode of the destination is offset addition. In offset addition, DMSBS and DMDBS registers indicates reload area and also works as an access offset value. Unlike other transfer modes, DMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of DMSBS and DMDBS registers is the number of blocks. When the transfer starts, DMCRAL[15:0] bits operates as a down counter, DMSAR and DMDAR registers reloads the value of DMSRR and DMDRR registers every time one block is transferred. In addition, DMSBSL[15:0] and DMDBSL[15:0] bits, which is the lower 16 bits of DMSBS and DMDBS registers, also operates as a down counter and decrements every time one block is transferred. When the DMSBS and DMDBS registers value becomes 1, DMSAR and DMDAR registers reloads the value of DMSRR and DMDRR registers. When DMAMD.SADR and DMAMD.DADR registers is set to 0, offset addition operation of the same area is repeated. DMDAR register only reloads DMDRR register. Figure 18.16 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR = 0.

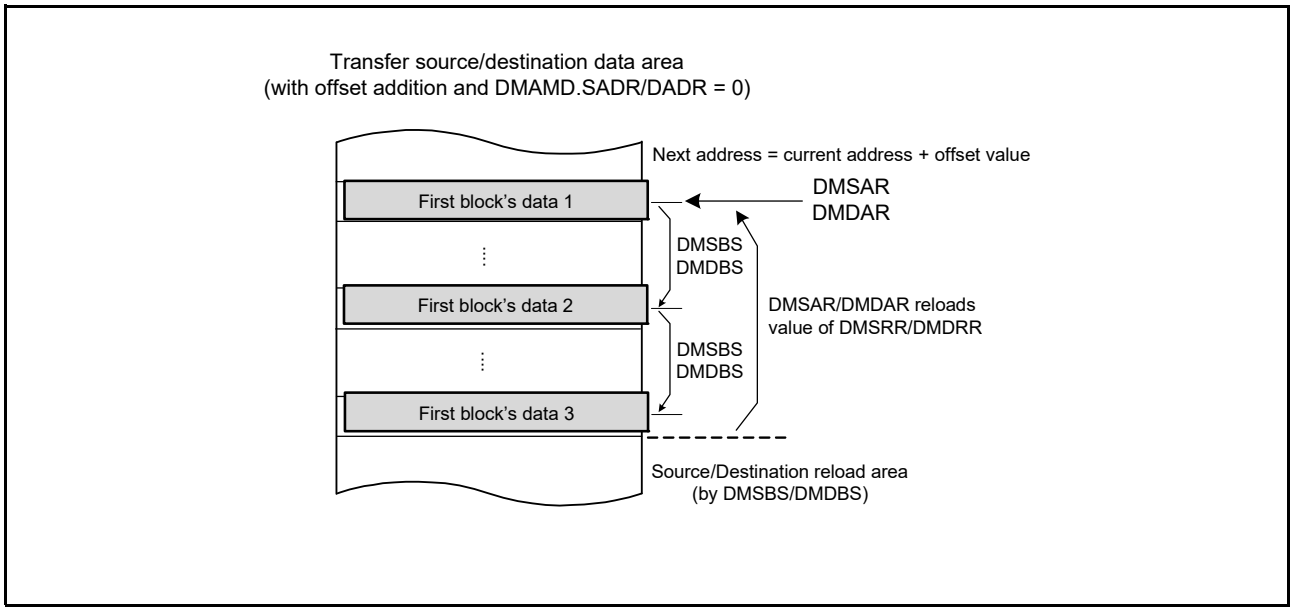


Figure 18.16 Address Update in Offset Addition with DMAMD.SADR and DMAMD.DADR = 0

When DMAMD.SADR and DMAMD.DADR bits is set to 1, the address is incremented by one data unit after DMSRR and DMDRR registers is reloaded by DMCRAL[15:0] = 1. In other words, an index value ((DMDBSH – DMDBSL) × transfer data size) is added to DMDAR after DMDRR is reloaded. This behavior is used to implement multiple ring buffers. Figure 18.17 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR = 1.

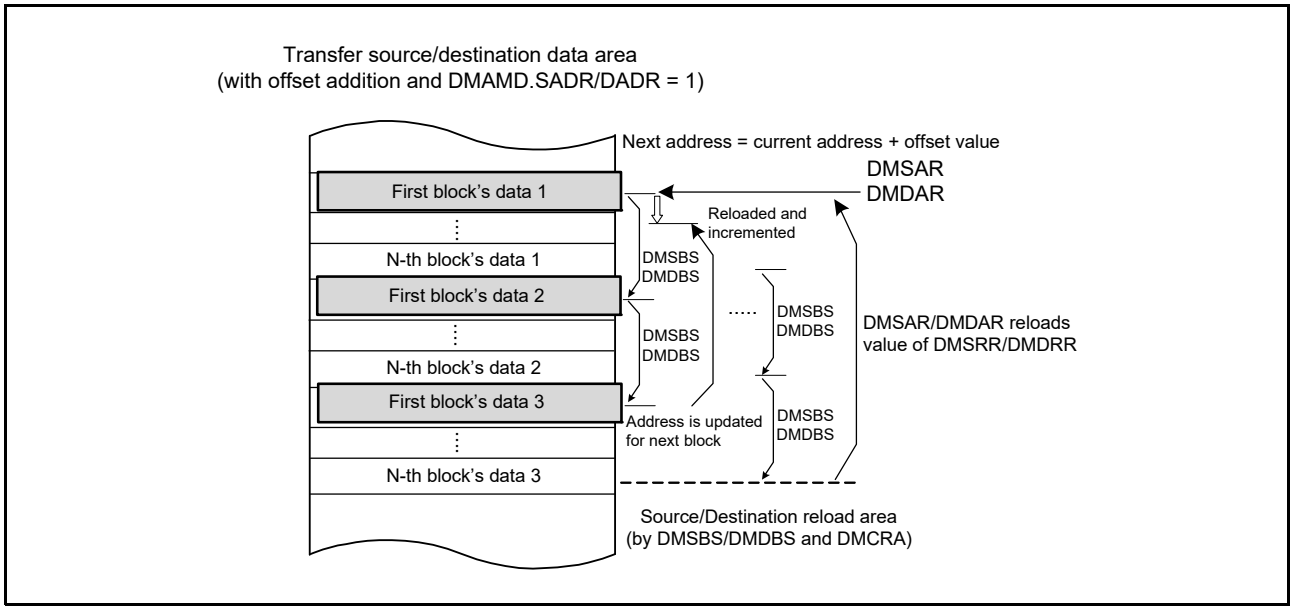


Figure 18.17 Address Update in Offset Addition with DMAMD.SADR and DMAMD.DADR = 1

18.3.6 Example of Using Repeat-Block Transfer Mode

In repeat-block transfer mode, it is possible to realize repeated access to interval data and single or multiple ring buffers by combining the above address update modes. Following sections shows some usage examples.

(1) Use Case Example: Interval Address to Single Ring Buffer

Figure 18.18 shows an example of reading interval ADDRn registers (data register) of ADC module and storing it in single ring buffer. It transfers 2 data every 4 words per 1 request. The value of DMSAR register is incremented by 1 data per 1 request. The following settings enable this: The transfer source is offset addition, DMAMD.SADR = 1, the block size (DMCRA) is 2, and the transfer source offset (DMSBS) is 4. Table 18.15 lists a setting example.

Table 18.15 Setting Example: Interval Address to Single Ring Buffer

Register	Value	Description
DMSAR, DMSRR	8743 A000h	Initial source address
DMDAR, DMDRR	8200 0000h	Initial destination address
DMTMD.SZ[1:0]	10b	Data size is word
DMAMD.SADR	1	Source address update setting after reload: Add index after reloading
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	10b	Destination update mode is increment
DMCRAH[9:0], DMCRAL[15:0]	2	Transfer block size
DMSBSH[15:0], DMSBSL[15:0]	4	Source whole buffer size (unit is ‘blocks’) and Source access offset (unit is ‘data’)
DMDBSH[15:0], DMDBSL[15:0]	N × 2 (DMCRA)	Destination buffer size (unit is ‘data’)

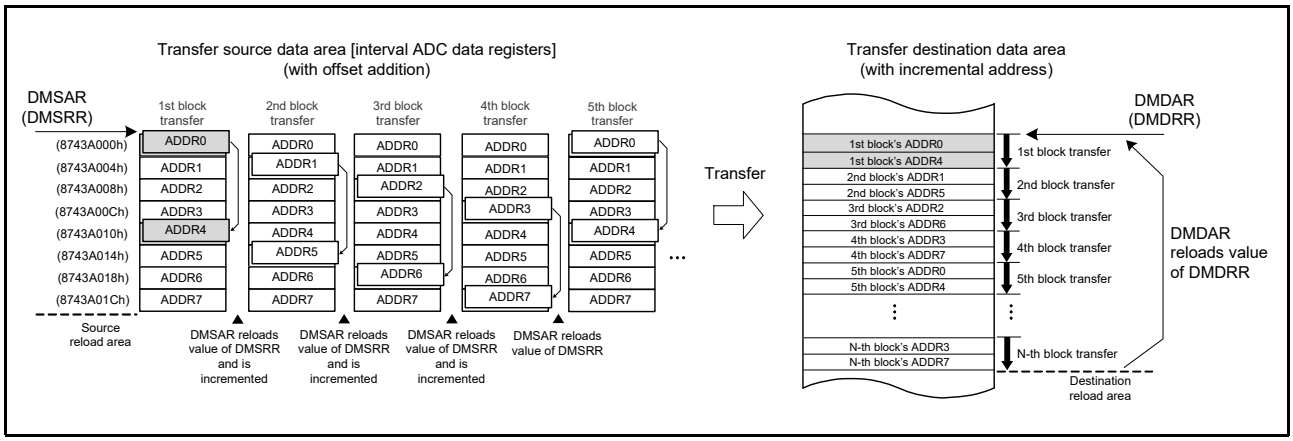


Figure 18.18 Use Case Example: Interval Address to Single Ring Buffer

(2) Use Case Example: Single Block to Multiple Ring Buffer

Figure 18.19 shows an example of storing the continuous ADDRn registers (data register) of ADC module individually in multiple ring buffers. In this example, a ring buffer in which only the first element (ADDR0) in a single block is arranged in transfer order is created at the destination. Also, in the next area, a ring buffer in which only the second element (ADDR1) is arranged in transfer order is created at the destination. In the following case, a ring buffer of length N defined by DMDBS register is created. The number of data elements in the block is 3, which is defined by DMCRA register. Table 18.16 lists a setting example.

Table 18.16 Setting Example: Single Block to Multiple Ring Buffer

Register	Value	Description
DMSAR, DMSRR	8743 A000h	Initial source address
DMDAR, DMDRR	8200 0000h	Initial destination address
DMTMD.SZ[1:0]	10b	Data size is word
DMAMD.SADR	0	Source address update setting after reload: Only reloading
DMAMD.DADR	1	Destination address update setting after reload: Add index after reloading
DMAMD.SM[1:0]	10b	Source update mode is increment
DMAMD.DM[1:0]	10b	Destination update mode is increment
DMCRAH[9:0], DMCRAL[15:0]	3	Transfer block size
DMSBSH[15:0], DMSBSL[15:0]	3	Source buffer size (unit is ‘data’)
DMDBSH[15:0], DMDBSL[15:0]	N	Destination whole buffer size (unit is ‘blocks’) and Destination access offset (unit is ‘data’)

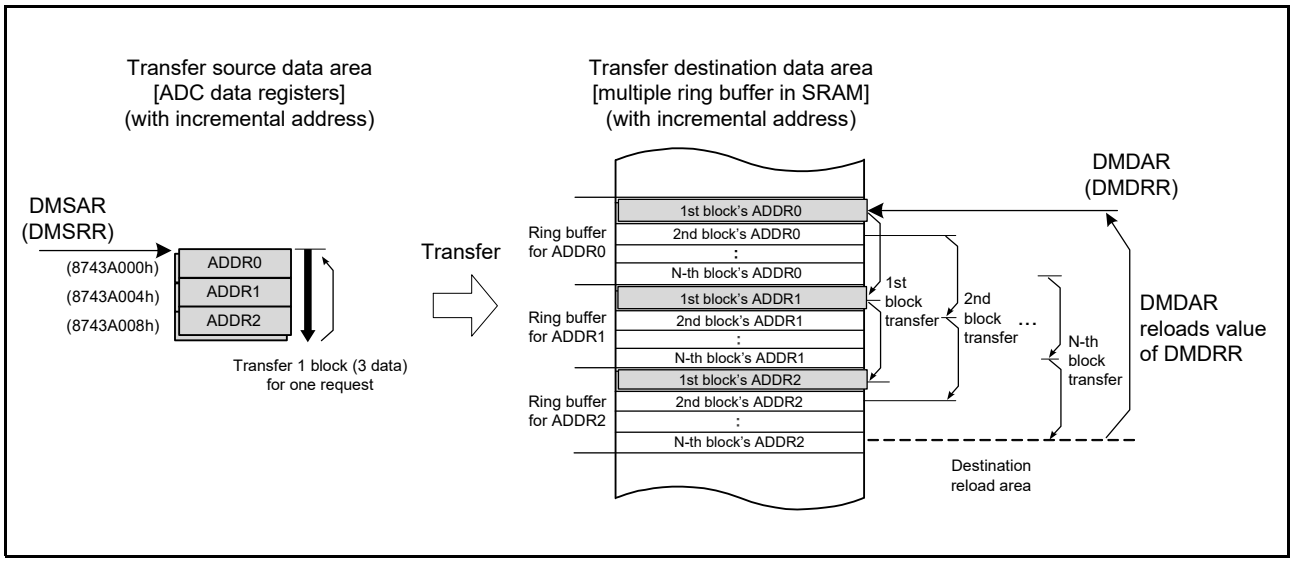


Figure 18.19 Use Case Example: Single Block to Multiple Ring Buffer

18.3.7 Trigger Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC trigger sources. Setting the DMTMD.DCTG[1:0] bits selects the trigger source.

(1) Trigger by Software

When DMA transfer is started by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b.
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).
4. Set the DMREQ.SWREQ bit to 1 (DMA requested).

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request. When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by Interrupt Request

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC trigger sources. The trigger sources can be selected individually for each channel in the ICU.DMRSRm register (m = 0 to 7). To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set the ICU.DMRSRm register (m = 0 to 7) bits to the event number, and then clear the corresponding IR flag.
2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit to 1 (DMAC activation enabled).
5. Set the IERi.IENj bit to 0 so that the corresponding interrupt is enabled.

For interrupt requests specified as DMAC trigger sources, see Table 15.6, Interrupt Vector Table, Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

18.3.8 Operation Timing

Figure 18.20 and Figure 18.21 show DMAC operation timing examples.

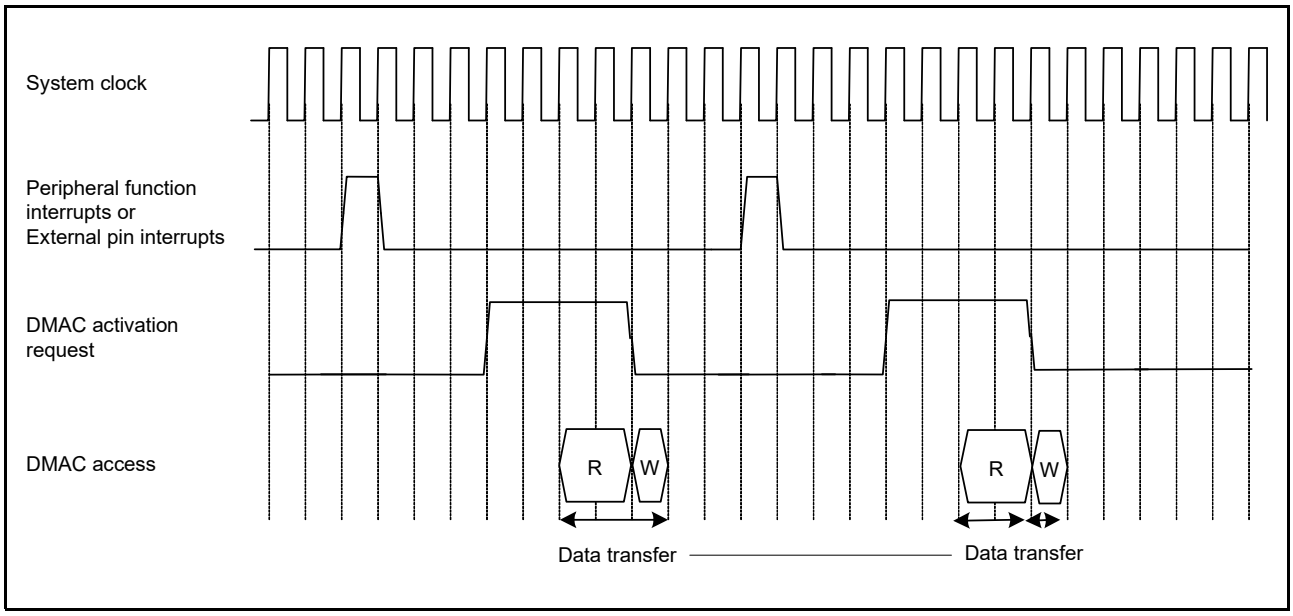


Figure 18.20 DMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

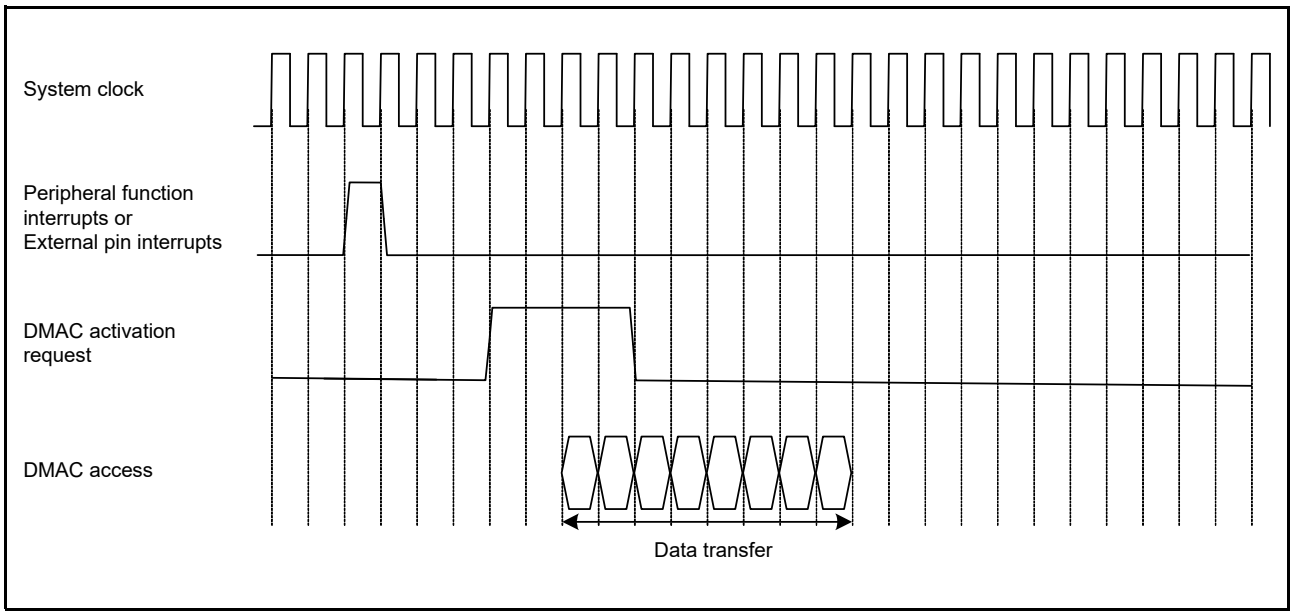


Figure 18.21 DMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

18.3.9 Activating the DMAC

Table 18.17 lists the register setting procedure of normal, repeat and block transfer mode and Table 18.18 lists register setting procedure of repeat-block transfer mode.

Table 18.17 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the DMAC trigger source.	To use peripheral function interrupts as DMAC trigger sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMAC trigger source.	To use external pin interrupts as DMAC trigger sources.
3	Clear the ICU.IERi.IENj bit corresponding to the interrupt vector number for the DMAC trigger	Disable the interrupt trigger for the DMAC
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt source as a DMAC trigger source to the DMRSRm register by using the Interrupt Controller Unit	To use internal peripheral interrupts or external pin interrupts as DMAC trigger sources. Enable the interrupt bit for the trigger source. Set the DMAC trigger source.
6	Set the peripheral module as a DMAC trigger source	To use peripheral function interrupt as a DMAC trigger source. Set the control register for the peripheral function.
7	Set the IRQn pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMAC trigger source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Destination address update mode select bits Set the Source address update mode select bits Set the Destination address extended repeat area setting bits Set the Source address extended repeat area setting bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request source select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits Set the Transfer keeping bit
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the transfer end interrupts. Enable the transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.
16	Set the DMAST.DMST bit to 1	Enable DMAC operation.*1 Common settings for DMAC
17	Set the ICU.IERi.IENj bit corresponding to the interrupt vector number for the DMAC trigger	Enable the interrupt trigger for the DMAC
18	Start the peripheral function as a DMAC trigger source	To use peripheral function interrupt as a DMAC trigger source
19	Enable the IRQn pin as a DMAC trigger source	To use external pin interrupt as a DMAC trigger source
20	End of initial settings	For trigger by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

m: DMAC channel (m = 0 to 7)

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual trigger sources.

Table 18.18 Register Setting Procedure of Repeat-Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the DMAC trigger source.	To use peripheral function interrupts as DMAC trigger sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMAC trigger source.	To use external pin interrupts as DMAC trigger sources.
3	Clear the ICU.IERi.IENj bit corresponding to the interrupt vector number for the DMAC trigger	Disable the interrupt trigger for the DMAC
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt source as a DMAC trigger source to the DMRSRm register by using the Interrupt Controller Unit	To use internal peripheral interrupts or external pin interrupts as DMA trigger sources. Enable the interrupt bit for the trigger source. Set the DMAC trigger source.
6	Set the peripheral module as a DMAC trigger source	To use peripheral function interrupt as a DMAC trigger source. Set the control register for the peripheral function.
7	Set the IRQn pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMAC trigger source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DADR bit Set the DMAMD.SADR bit	Set the Destination address update mode select bits Set the Source address update mode select bits Set the Destination address update setting after reload bit Set the Source address update setting after reload bit
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request source select bits Set the Data transfer size bits Set the transfer mode to repeat-block transfer mode Set the Transfer keeping bit
10	Set the DMSAR register Set the DMDAR register Set the DMSRR register Set the DMDRR register Set the DMCRA register Set the DMCRB register	Set the transfer source start address Set the transfer destination start address Set the initial value of source start address Set the initial value of destination start address Set the number of transfer operations Set the number of block transfer operations
11	Set the DMSBS register Set the DMDBS register	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset Set the destination buffer size and access offset
12	Set the DMINT.DTIE bit to 1	To use the transfer end interrupts. Enable the transfer end interrupts.
13	Set the DMCNT.DTE bit to 1	Enable DMA transfer
14	Set the DMAST.DMST bit to 1	Enable DMAC operation.*1
15	Set the ICU.IERi.IENj bit corresponding to the interrupt vector number for the DMAC trigger	Enable the interrupt trigger for the DMAC
16	Start the peripheral function as a DMAC trigger source	To use peripheral function interrupt as a DMAC trigger source
17	Enable the IRQn pin as a DMAC trigger source	To use external pin interrupt as a DMAC trigger source
18	End of initial settings	For trigger by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

m: DMAC channel (m = 0 to 7)

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual trigger sources.

18.3.10 Starting DMA Transfer

To enable the DMA transfer of DMACm (m = 0 to 7), set the DMCNT.DTE bit to 1 (DMA transfer enabled), and then set the DMAST.DMST bit to 1 (DMAC module start). Another transfer request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

18.3.11 Registers during DMA Transfer

The DMACm registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMSBS, DMCNT, and DMSTS.

(1) DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(2) DMA Destination Address Register (DMDAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(3) DMA Transfer Count Register (DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(4) DMA Block Transfer Count Register (DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(5) DMA Source Buffer Size Register (DMSBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(6) DMA Destination Buffer Size Register (DMDBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 18.5 to Table 18.7.

(7) DMA Transfer Enable Bit (DMCNT.DTE)

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically set to 0 by the DMAC according to the DMA transfer state. The conditions for clearing this bit by the DMAC are as follows:

- The specified total volume of data transfer is completed.
- The DMA transfer is stopped by the repeat size end interrupt.
- The DMA transfer is stopped by the extended repeat area overflow interrupt.

- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for the DMCNT register). In this case, writing must be performed after the bit is set to 0.

(8) DMA Active Flag (DMSTS.ACT)

The DMSTS.ACT flag indicates whether the DMACm is in the idle or active state. This flag is set to 1 when the DMACm starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed. Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

(9) Transfer End Interrupt Flag (DMSTS.DTIF)

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed. When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested. This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is set to 0 indicating the DMA transfer end. This flag is automatically set to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

(10) Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When both this flag and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested. This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is set to 0 indicating the DMA transfer end. This flag is automatically set to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling. Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).

18.3.12 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests. When DMCTL.PR = 0, the channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7 (channel 0: Highest) > DTC. When DMCTL.PR = 1, the DMAC determines the priority of channels at round-robin. When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

18.3.13 Channel Security

The security attribute can be set with CPSCU.DMACCHSAR.SADMACn bit for each DMAC channel.
When the CPSCU.DMACCHSAR.SADMACn bit is 0.

- When the corresponding channel transfers, it behaves as a secure master.
- The register of corresponding channel has a secure attribute.

When the CPSCU.DMACCHSAR.SADMACn bit is 1.

- When the corresponding channel transfers, it behaves as a non-secure master.
- The register of corresponding channel has a non-secure attribute.

Do not change the CPSCU.DMACCHSAR register while DMA transfer.

18.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0.

18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

When the value of the DMCRAL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is set to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of the DMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is set to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.
When the DMTMD.TKP bit is set to 1 (free-running function), the DMSTS.DTIF flag is set to 1, but the DMCNT.DTE bit does not become 0.

(3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of the DMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is set to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.
When the DMTMD.TKP bit is set to 1 (free-running function), the DMSTS.DTIF flag is set to 1, but the DMCNT.DTE bit does not become 0.

(4) In Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

When the value of the DMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is set to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.
When the DMTMD.TKP bit is set to 1 (free-running function), the DMSTS.DTIF flag is set to 1, but the DMCNT.DTE bit does not become 0.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).

18.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of each single-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested, DMA transfer is terminated regardless of the value of the DMTMD.TKP bit, the DMCNT.DTE bit is set to 0, and the DMSTS.ESIF flag is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit. A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of each single-block size of data is completed. Repeat size end interrupt cannot be requested in repeat-block transfer mode. Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).
For the interrupt outputs, refer to Figure 18.22 and Figure 18.23.

18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area address occurs while the source address extended repeat area is specified with the DMINT.SARIE bit set to 1 or the destination address extended repeat area is specified with the DMINT.DARIE bit set to 1, an interrupt due to an extended repeat area overflow is requested. When the interrupt is requested, DMA transfer is terminated regardless of the value of the DMTMD.TKP bit, the DMCNT.DTE bit is set to 0, and the DMSTS.ESIF flag is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Even if an interrupt due to an extended repeat area overflow is requested during a read cycle, the subsequent write cycle proceeds. In block transfer mode, even if an interrupt due to an extended repeat area overflow is requested during a single-block transfer, the remaining data in the block are transferred; transfer is terminated after a block transfer. An interrupt due to an extended repeat area overflow is not possible in repeat-block transfer mode. Before interrupt requests can be sent from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM). For the interrupt outputs, see Figure 18.22 and Figure 18.23.

18.5 Processing on DMA Transfer Error

If the DMBWR.BWE bit is 0 and an access error occurs during a DMA transfer, the DMAC will abort the transfer. The IR status flag for the vector number that caused the error and the DMCNT.DTE bit for the channel on which the error occurred are cleared. The DMA transfer setting registers are not written back. Information about the channel that caused the error is set in the DMECHR register. If there is then a transfer request from a source other than the one that started the DMA transfer that caused the error, arbitration will be re-established. For details on bus errors, see section 16.8, Bus Error Monitoring Section.

18.6 Interrupts

18.6.1 Transfer End Interrupt

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. The interrupt request is generated after the bus response to the final write of the DMA transfer. The timing of the bus response to a DMA write access depends on the setting of the DMBWR.BWE bit and transfer destination. For details, see section 18.2.19, DMA Bufferable Write Enable Register (DMBWR). In repeat-block transfer mode, do not enable escape transfer end interrupt. Table 18.19 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 18.22 shows the schematic logic diagram of interrupt outputs (DMAC0 to DMAC3). Figure 18.23 shows the schematic logic diagram of interrupt outputs (DMAC4 to DMAC7). Figure 18.24 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 18.19 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMSTS.DTIF	DMINT.DTIE
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		

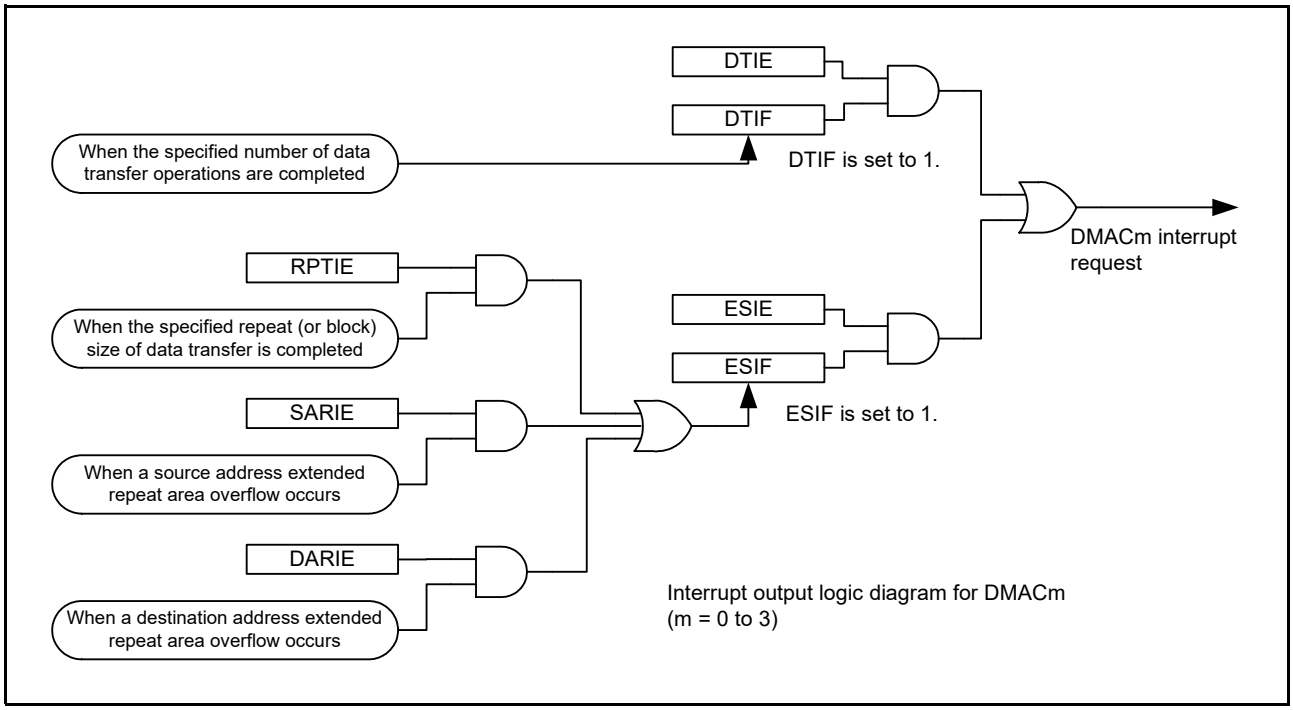


Figure 18.22 Schematic Logic Diagram of Interrupt Outputs (DMAC0 to DMAC3)

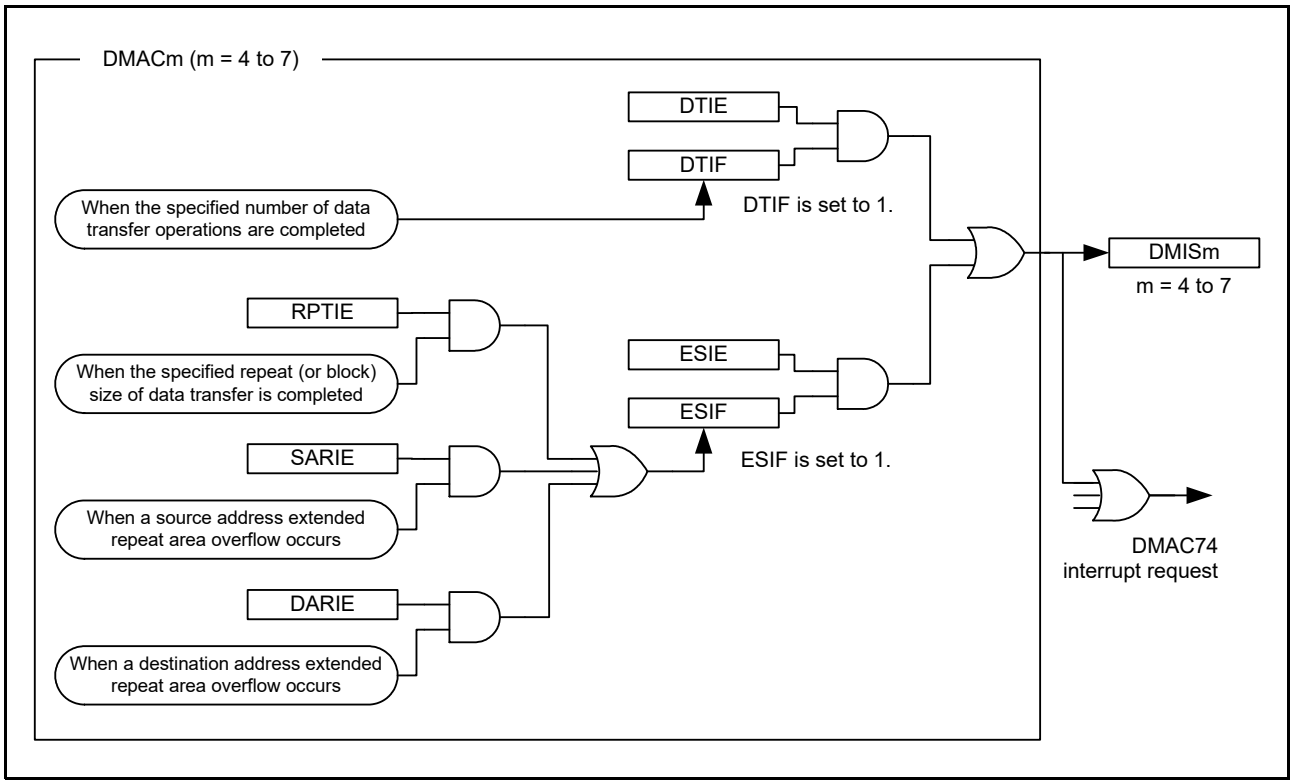


Figure 18.23 Schematic Logic Diagram of Interrupt Outputs (DMAC4 to DMAC7)

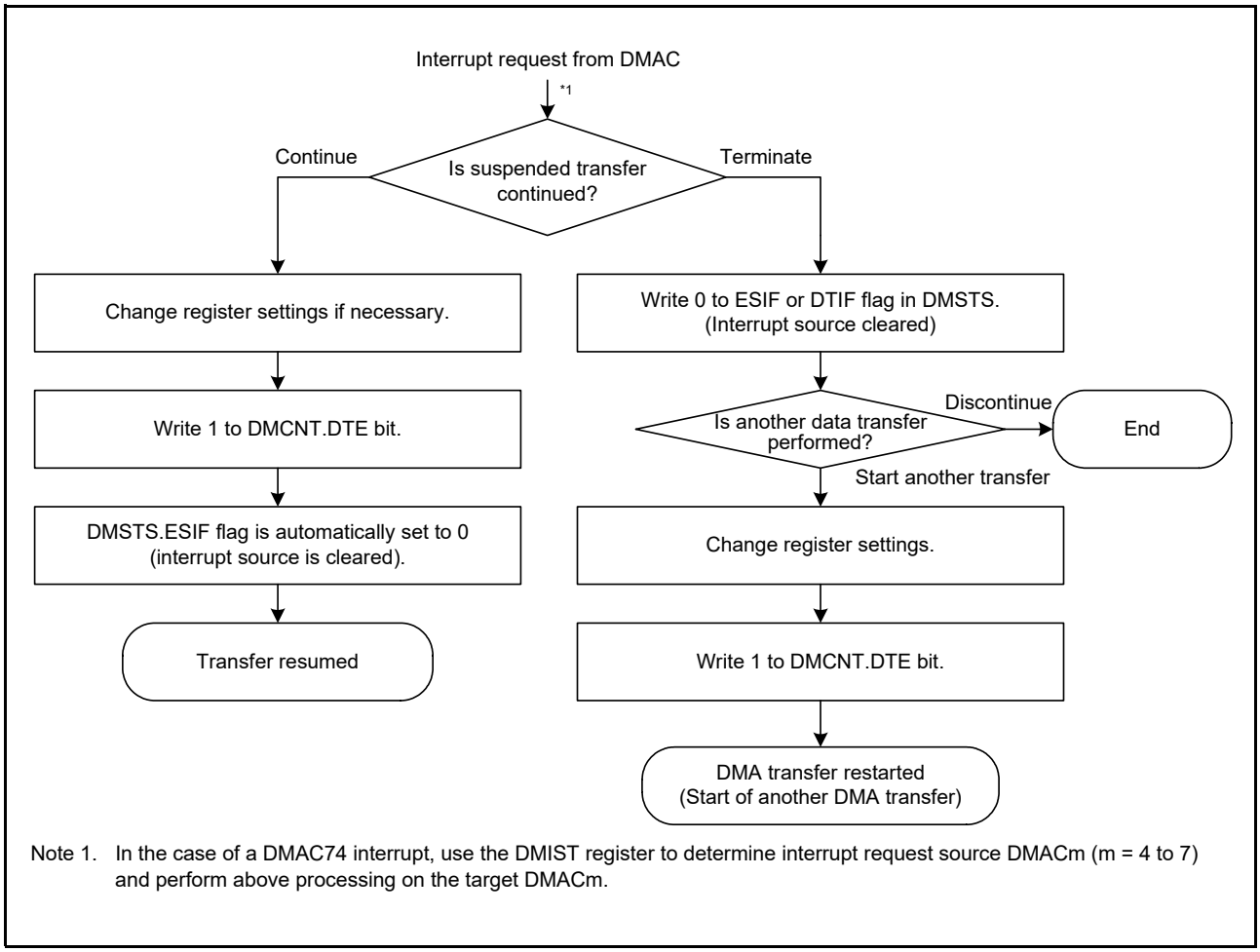
Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.



Note 1. In the case of a DMAC74 interrupt, use the DMIST register to determine interrupt request source DMACm (m = 4 to 7) and perform above processing on the target DMACm.

Figure 18.24 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

18.6.2 Transfer Error Interrupt

A transfer error occurring during DMA transfer is detected as a bus error. For details, see section 16.8, Bus Error Monitoring Section.

Table 18.20 Interrupt and Error Information Due to DMA Transfer Error Cause

Transfer Error Factor	Interrupt/Reset Selection	NMI	Interrupt	Bus Error Status	Error Channel Information
MSAU Error	BUS.MSAOAD.OAD	ICU.NMISR.BUSST	ICU.IR16.IR	BUS.BUSERRSTATDMAC0.MSERRSTAT	DMECHR
Illegal Access Error	BUS.BUSOAD.ILERROAD			BUS.BUSERRSTATDMAC0.ILERRSTAT	
Slave Bus Error	BUS.BUSOAD.SLERROAD			BUS.BUSERRSTATDMAC0.SLERRSTAT	
Protect Zone Access Error	BUS.BUSOAD.SLERROAD			BUS.BUSERRSTATDMAC0.SLERRSTAT	

18.7 Event Link

Each DMAC channel outputs an event link request each time one data transfer is completed or, in block and repeat-block transfer mode, each time one block transfer is completed. The timing of the bus response from the DMA transfer destination depends on the transfer destination and bus configuration. For details, see section 16.2.5, Restrictions in section 16, Buses. Enabling or disabling buffered writing to bufferable areas in DMA transfers can be specified by the DMBWR register. For details, see section 18.2.19, DMA Bufferable Write Enable Register (DMBWR).

For details on the event link function, see section 21, Event Link Controller (ELC).

18.8 Low-Power Consumption Function

Before transition to the module-stop state, software standby mode, or deep software standby mode, set the DMAST.DMST bit to 0 (the DMAC suspended), and then perform the following.

(1) Module-Stop Function

Writing 1 to the module-stop setting bit (transition to the module-stop state) enables the module-stop function of the DMAC. If DMA transfer is in progress at the time a 1 is written to the module-stop setting bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the module-stop setting bit is 1, accessing the DMAC registers are prohibited. Writing 0 to the module-stop setting bit releases the DMAC from the module-stop state.

(2) Software Standby and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.2.1, Transitioning to Software Standby Mode or section 11.6.3.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption. If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby or deep software standby follows the completion of DMA transfer.

(3) Note on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.8.6, Timing of the WAIT Instruction in section 11, Low Power Consumption. To perform DMA transfer after returning from low-power consumption mode, set the DMAST.DMST bit to 1 again. To use a request that is generated in software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the settings described in section 15.7.3, Selecting Interrupt Request Destination in section 15, Interrupt Controller (ICUM), and then execute the WAIT instruction.

18.9 Usage Notes

18.9.1 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

18.9.2 Setting of DMAC Trigger Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC trigger select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMCNT.DTE) is 0 (DMA transfer is disabled). Moreover, the DTC transfer request enable register (ICU.DTCERn) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm registers, see section 15, Interrupt Controller (ICUM).

18.9.3 Suspending or Restarting DMA Transfer

To suspend a DMA transfer request, write 0 to the interrupt enable bit for the trigger source (ICU.IERi.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERi.IENj bit with the setting shown in section 18.3.9, Activating the DMAC.

19. EXDMA Controller (EXDMACAa)

This MCU incorporates a 2-channel direct memory access controller (EXDMAC) designed exclusively for external bus transfer. The EXDMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the EXDMAC transfers data stored at the transfer source address to the transfer destination address.

19.1 Overview

Table 19.1 lists the specifications of the EXDMAC, and Figure 19.1 shows a block diagram of the EXDMAC.

Table 19.1 Specifications of EXDMAC

Item		Description EXDMAC
Number of channels		2 (EXDMACn (n = 0, 1))
Transfer space		SRAM0/SRAM1 (Secure: 8200 0000h to 8217 FFFFh, Non-secure: 0200 0000h to 0217 FFFFh) CS0 to CS7/SDRAM (7000 0000h to 7FFF FFFFh) xSPI0/xSPI1 (2000 0000h to 5FFF FFFFh)
Maximum transfer volume		64M data (Maximum number of transfers in block transfer mode: 1024 data × 65536 blocks)
EXDMAC trigger source		• Trigger source selectable for each channel Software trigger Interrupt requests from peripheral modules*1 or trigger input to external interrupt input pins
Channel priority		Selectable from fixed priority or round-robin • Fixed priority: Channel 0 > channel 1 (channel 0: Highest) • Round-robin
Transfer data	Single data	Bit length: 8, 16, 32, 64 bits
	Block size	Number of data: 1 to 1024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1024 • Selectable free running function
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 data • Selectable free running function
	Repeat-block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 data • Block transfer can be repeated • Maximum settable repeat size: 64K • Selectable free running function
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Processing on DMA transfer error		• When DMA transfer error occurs, the transfer that caused the error is stopped • For clearing requests to the registers for trigger source in the ICU, clearing targets are selectable for each channel or all channels.
CPU interrupt source (EXDMACnI)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	• Generated when the repeat size of data transfer is completed. • Generated when the source address extended repeat area overflows. • Generated when the destination address extended repeat area overflows.
Power consumption reduction function		Module-stop state can be set.
Protect Zone Filter		Security can be set for each channel.

Note 1. For details on EXDMAC trigger sources, see Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

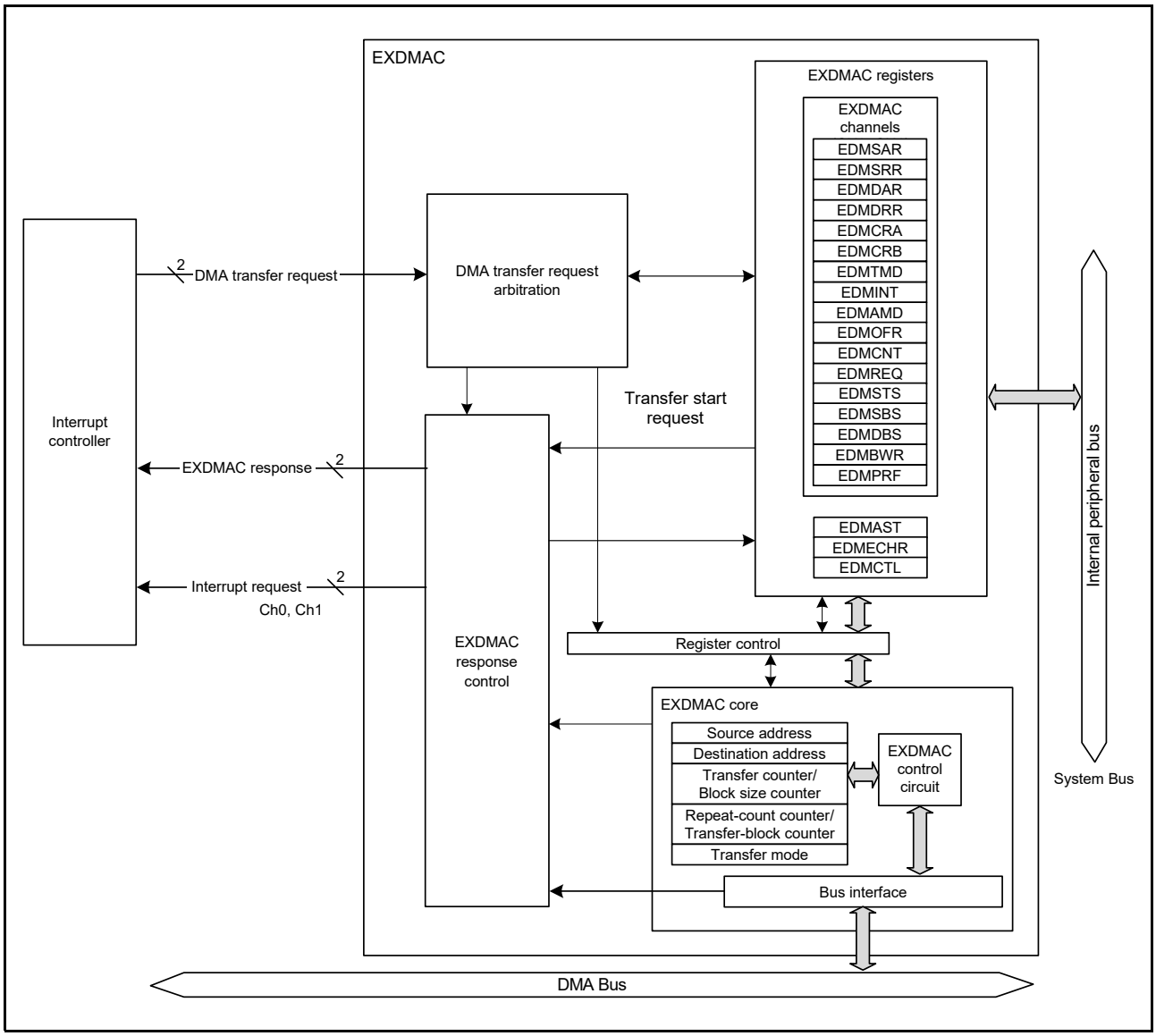


Figure 19.1 Block Diagram of EXDMAC

19.2 Register Descriptions

Security attribution can be set for each register and is used to determine whether the target register is a secure register or a non-secure register. The attribution can be set by the EXDMACCHSAR, EXDMACSAR registers.

19.2.1 EXDMAC Channel Security Attribution Register (EXDMACCHSAR)

Address(es): CPSCU.EXDMACCHSAR 8701 819Ch, CPSCU_NS.EXDMACCHSAR 0701 819Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SADMACX1	SADMACX0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SADMACX0	Security Attributes of Output and Registers for EXDMAC Channel 0 Select	0: Secure 1: Non-secure	R/W
b1	SADMACX1	Security Attributes of Output and Registers for EXDMAC Channel 1 Select	0: Secure 1: Non-secure	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

SADMACXn Bits (Security Attributes of Output and Registers for EXDMAC Channel n Select) (n = 0, 1)

Each bit specifies the security attribute of the register for a EXDMAC channel n. For details, see section 19.3.12, Channel Security. The target registers are ICU.SELEXDRn (n = 0, 1), EDMSAR, EDMSRR, EDMDAR, EDMDRR, EDMCRA, EDMCRB, EDMTMD, EDMINT, EDMAMD, EDMOFR, EDMCNT, EDMREQ, EDMSTS, EDMSBS, EDMDBS, EDMBWR, and EDMPRF.

19.2.2 EXDMAC Controller Security Attribution Register (EXDMACSAR)

Address(es): CPSCU.EXDMACSAR 8701 8038h, CPSCU_NS.EXDMACSAR 0701 8038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAXSTSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMAXSTSA0	EXDMA EDMAST Security Attribution Setting 0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

For EXDMAC, security attribution is set for each channel. However, security attribution of the EDMAST and EDMCTL registers is set by this register.

DMAXSTSA0 Bit (EXDMA EDMAST Security Attribution Setting 0)

The DMAXSTSA0 bit specifies security attribution of the EDMAST and EDMCTL registers.

19.2.3 EXDMA Source Address Register (EDMSAR)

Address(es): EXDMAC0.EDMSAR 8701 A000h, EXDMAC1.EDMSAR 8701 A040h,
EXDMAC0_NS.EDMSAR 0701 A000h, EXDMAC1_NS.EDMSAR 0701 A040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the EDMSAR register while EXDMAC activation is disabled (EDMAST.DMST = 0) or DMA transfer is disabled (EDMCNT.DTE = 0).

19.2.4 EXDMA Source Reload Address Register (EDMSRR)

Address(es): EXDMAC0.EDMSRR 8701 A020h, EXDMAC1.EDMSRR 8701 A060h,
EXDMAC0_NS.EDMSRR 0701 A020h, EXDMAC1_NS.EDMSRR 0701 A060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source reload address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

EDMSRR register is used to specify the DMA transfer source start and reload addresses in repeat-block transfer mode. Set EDMSRR register while EXDMAC activation is disabled (EDMAST.DMST = 0) or DMA transfer is disabled (EDMCNT.DTE = 0). Set the same value to EDMSRR register as that of EDMSAR register. For details on the repeat-block transfer mode operation, see section (4), Repeat-Block Transfer Mode. In normal transfer mode, repeat transfer mode and block transfer mode, EDMSRR register is not used. The setting is invalid.

19.2.5 EXDMA Destination Address Register (EDMDAR)

Address(es): EXDMAC0.EDMDAR 8701 A004h, EXDMAC1.EDMDAR 8701 A044h, EXDMAC0_NS.EDMDAR 0701 A004h, EXDMAC1_NS.EDMDAR 0701 A044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the EDMDAR register while EXDMAC activation is disabled (EDMAST.DMST = 0) or DMA transfer is disabled (EDMCNT.DTE = 0).

19.2.6 EXDMA Destination Reload Address Register (EDMDRR)

Address(es): EXDMAC0.EDMDRR 8701 A024h, EXDMAC1.EDMDRR 8701 A064h, EXDMAC0_NS.EDMDRR 0701 A024h, EXDMAC1_NS.EDMDRR 0701 A064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination reload address	0000 0000h to FFFF FFFFh (4 Gbytes)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

EDMDRR register is used to specify the DMA transfer destination start and reload addresses in repeat-block transfer mode. Set EDMDRR register while EXDMAC activation is disabled (EDMAST.DMST = 0) or DMA transfer is disabled (EDMCNT.DTE = 0). Set the same value to EDMDRR register as that of EDMDAR register. For details on the repeat-block transfer mode operation, see section (4), Repeat-Block Transfer Mode. In normal transfer mode, repeat transfer mode and block transfer mode, EDMDRR register is not used. The setting is invalid.

19.2.7 EXDMA Transfer Count Register (EDMCRA)

Address(es): EXDMAC0.EDMCRA 8701 A008h, EXDMAC1.EDMCRA 8701 A048h,
EXDMAC0_NS.EDMCRA 0701 A008h, EXDMAC1_NS.EDMCRA 0701 A048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	EDMCRAH[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDMCRAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EDMCRAL[15:0]	Lower Bits of Transfer Count	Specifies the number of transfer operations	R/W
b25 to b16	EDMCRAH[9:0]	Upper Bits of Transfer Count		R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Set the same value for EDMCRAH[9:0] and EDMCRAL[15:0] bits in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (EDMTMD.MD[1:0] Bits = 00b)

The EDMCRAL[15:0] bits function as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred. When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function). Free running function is not selected by EDMTMD.TKP bit in normal transfer mode. The EDMCRAH[9:0] bits are not used in normal transfer mode. Write 000h to the EDMCRAH[9:0] bits.

(2) Repeat Transfer Mode (EDMTMD.MD[1:0] Bits = 01b)

The EDMCRAH[9:0] bits hold the repeat size and the EDMCRAL[15:0] bits function as a 10-bit transfer counter. The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh can be set for the EDMCRAH[9:0] and EDMCRAL[15:0] bits. Setting bits 15 to 10 in the EDMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the EDMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the EDMCRAH[9:0] bits is loaded into the EDMCRAL[15:0] bits.

(3) Block Transfer Mode (EDMTMD.MD[1:0] Bits = 10b)

The EDMCRAH[9:0] bits hold the block size and the EDMCRAL[15:0] bits function as a 10-bit block size counter. The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for the EDMCRAH[9:0] and EDMCRAL[15:0] bits. Setting bits 15 to 10 in the EDMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the EDMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the EDMCRAH[9:0] bits is loaded into the EDMCRAL[15:0] bits.

(4) Repeat-Block Transfer Mode (EDMTMD.MD[1:0] Bits = 11b)

The EDMCRAH[9:0] bits hold the block size and the EDMCRAL[15:0] bits function as a 10-bit block size counter. The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat-block transfer

mode, a value in the range of 000h to 3FFh can be set for the EDMCRAH[9:0] and EDMCRAL[15:0] bits. Setting bits 15 to 10 in the EDMCRAL[15:0] bits is invalid. Write 0 to these bits. The value in the EDMCRAL[15:0] bits is decremented by one each time data is transferred until it reaches 000h, at which the value in the EDMCRAH[9:0] bits is loaded into the EDMCRAL[15:0] bits.

19.2.8 EXDMA Block Transfer Count Register (EDMCRB)

Address(es): EXDMAC0.EDMCRB 8701 A00Ch, EXDMAC1.EDMCRB 8701 A04Ch, EXDMAC0_NS.EDMCRB 0701 A00Ch, EXDMAC1_NS.EDMCRB 0701 A04Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EDMCRBH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDMCRBL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EDMCRBL[15:0]	Lower Bits of Block Transfer Count	Counts the number of block, repeat or repeat-block transfer operations	R/W
b31 to b16	EDMCRBH[15:0]	Upper Bits of Block Transfer Count		R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the same value for EDMCRBH[15:0] and EDMCRBL[15:0] bits in repeat transfer mode, block transfer mode, and repeat-block transfer mode. EDMCRBH[15:0] and EDMCRBL[15:0] bits specifies and counts, respectively, the number of transfer operations in block, repeat, and repeat-block transfer mode. The number of transfer operations is one when the setting is 0001h, 65535 when it is FFFFh, and 65536 when it is 0000h. In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred. In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final block data is transferred. In normal transfer mode, do not use the EDMCRB register because the setting is invalid. When EDMTMD.TKP bit is 1 and the final data of one repeat size or one block size is transferred, EDMCRBL[15:0] bits are reloaded with the value of EDMCRBH[15:0] bits automatically.

19.2.9 EXDMA Transfer Mode Register (EDMTMD)

Address(es): EXDMAC0.EDMTMD 8701 A010h, EXDMAC1.EDMTMD 8701 A050h, EXDMAC0_NS.EDMTMD 0701 A010h, EXDMAC1_NS.EDMTMD 0701 A050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	TKP	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Data Transfer Size	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: 64 bits	R/W
b10	TKP	Transfer Keeping	0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. EXDMAC trigger source is selected using the ICU.SELEXDRn registers. For details on EXDMAC trigger sources, see Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] bits select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

TKP Bit (Transfer Keeping)

TKP bit selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

19.2.10 EXDMA Interrupt Setting Register (EDMINT)

Address(es): EXDMAC0.EDMINT 8701 A013h, EXDMAC1.EDMINT 8701 A053h, EXDMAC0_NS.EDMINT 0701 A013h, EXDMAC1_NS.EDMINT 0701 A053h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the EDMCNT.DTE bit is set to 0. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt due to an extended repeat area overflow on the destination address is requested. When block transfer mode is used with the extended repeat area function, an interrupt is requested when transfer of each single-block size of data is completed. When you set 1 in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped. When the extended repeat area is not specified for the destination address, this bit is ignored. When you use repeat-block transfer mode, do not set this bit to 1.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the EDMCNT.DTE bit is set to 0. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt due to an extended repeat area overflow on the source address is requested. When block transfer mode is used with the extended repeat area function, an interrupt is requested when transfer of each single-block size of data is completed. When you set 1 in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped. When the extended repeat area is not specified for the source address, this bit is ignored. When you use repeat-block transfer mode, do not set this bit to 1.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the EDMCNT.DTE bit is set to 0 when transfer of each single-repeat size of data is completed. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified). When this bit is set to 1 in block transfer mode, the EDMCNT.DTE bit is set to 0 when transfer of each single-block size of data is completed in the same way as repeat transfer mode. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified). When you use repeat-block transfer mode, do not set this bit to 1.

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and interrupt requests for extended repeat area overflows on the source and destination addresses) that are generated during DMA transfer. The transfer escape end interrupt is generated when the EDMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by setting this bit or the EDMSTS.ESIF flag to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers. The transfer end interrupt is generated when the EDMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the EDMSTS.DTIF flag to 0.

19.2.11 EXDMA Address Mode Register (EDMAMD)

Address(es): EXDMAC0.EDMAMD 8701 A014h, EXDMAC1.EDMAMD 8701 A054h,
EXDMAC0_NS.EDMAMD 0701 A014h, EXDMAC1_NS.EDMAMD 0701 A054h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		SADR	SARA[4:0]					DM[1:0]		DADR	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area Setting	Specifies the extended repeat area on the destination address. For details on the settings, see Table 19.2.	R/W
b5	DADR	Destination Address Update Setting after Reload	0: Only reloading. 1: Add index after reloading.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode Select	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area Setting	Specifies the extended repeat area on the source address. For details on the settings, see Table 19.2.	R/W
b13	SADR	Source Address Update Setting after Reload	0: Only reloading. 1: Add index after reloading.	R/W
b15, b14	SM[1:0]	Source Address Update Mode Select	b15 b14 0 0: Source address is fixed. 0 1: Offset addition 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DARA[4:0] Bits (Destination Address Extended Repeat Area Setting)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes. That is, the interval between settings is any power of two. When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set. When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when EDMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits. In repeat-block transfer mode, write 00000b in the DARA[4:0] bits. An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the EDMINT.DARIE bit set to 1. Table 19.2 lists the settings and the corresponding extended repeat areas.

Note: When setting the extended repeat area, set the lowest bit or lower bits of EDMDAR as follows according to the setting of EDMTMD.SZ[1:0].
When EDMTMD.SZ[1:0] = 01b (16 bits), set EDMDAR[0] to 0b.
When EDMTMD.SZ[1:0] = 10b (32 bits), set EDMDAR[1:0] to 00b.
When EDMTMD.SZ[1:0] = 11b (64 bits), set EDMDAR[2:0] to 000b.

DADR Bit (Destination Address Update Setting after Reload)

In repeat-block transfer mode, this bit specifies the behavior of EDMDAR register after reloading the value of EDMDRR

register. When this bit is set to 1, an index value ((EDMDBSH – EDMDBSL) × transfer data size) is added to EDMDAR register after reloading the value of EDMDRR register. When this bit is set to 0, EDMDAR register only reloads EDMDRR register. This behavior is described in Table 19.13. In normal, repeat or block transfer mode, this bit is ignored.

DM[1:0] Bits (Destination Address Update Mode Select)

These bits select the mode of updating the destination address. When increment is selected and the EDMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the destination address is incremented by 1, 2, 4, and 8, respectively. When decrement is selected and the EDMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the destination address is decremented by 1, 2, 4, and 8, respectively. When offset addition is selected, the offset specified by the EDMOFR register is added to the address.

SARA[4:0] Bits (Source Address Extended Repeat Area Setting)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes. When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set. When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when EDMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits. In repeat-block transfer mode, write 00000b in the SARA[4:0] bits. An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the EDMINT.SARIE bit set to 1. Table 19.2 lists the settings and the corresponding extended repeat areas.

Note: When setting the extended repeat area, set the lowest bit or lower bits of EDMSAR as follows according to the setting of EDMTMD.SZ[1:0].
When EDMTMD.SZ[1:0] = 01b (16 bits), set EDMSAR[0] to 0b.
When EDMTMD.SZ[1:0] = 10b (32 bits), set EDMSAR[1:0] to 00b.
When EDMTMD.SZ[1:0] = 11b (64 bits), set EDMSAR[2:0] to 000b.

SADR Bit (Source Address Update Setting after Reload)

In repeat-block transfer mode, this bit specifies the behavior of EDMSAR after reloading the value of EDMSRR. When this bit is set to 1, an index value ((EDMSBSH – EDMSBSL) × transfer data size) is added to EDMSAR after reloading the value of EDMSRR. When this bit is set to 0, EDMSAR only reloads EDMSRR. This behavior is described in Table 19.12. In normal, repeat or block transfer mode, this bit is ignored.

SM[1:0] Bits (Source Address Update Mode Select)

These bits select the mode of updating the source address. When increment is selected and the EDMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the source address is incremented by 1, 2, 4, and 8, respectively. When decrement is selected and the EDMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, and 11b, the source address is decremented by 1, 2, 4, and 8, respectively. When offset addition is selected, the offset specified by the EDMOFR register is added to the address.

Table 19.2 Settings and Corresponding Extended Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

19.2.12 EXDMA Offset Register (EDMOFR)

Address(es): EXDMAC0.EDMOFR 8701 A018h, EXDMAC1.EDMOFR 8701 A058h,
EXDMAC0_NS.EDMOFR 0701 A018h, EXDMAC1_NS.EDMOFR 0701 A058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Write to this register while the EXDMAC operation is stopped (EDMAST.DMST = 0) or DMA transfer is disabled (EDMCNT.DTE = 0) (not during data transfer). Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading EDMOFR register returns the extended value. In repeat-block transfer mode, the offset cannot be specified by EDMOFR when offset addition is selected, write 0000 0000h to EDMOFR register.

19.2.13 EXDMA Transfer Enable Register (EDMCNT)

Address(es): EXDMAC0.EDMCNT 8701 A01Ch, EXDMAC1.EDMCNT 8701 A05Ch, EXDMAC0_NS.EDMCNT 0701 A01Ch, EXDMAC1_NS.EDMCNT 0701 A05Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DTE Bit (DMA Transfer Enable)

When the EDMAST.DMST bit is set to 1 (EXDMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel. If the DTE bit is cleared during DMA transfer, no transfer requests for the corresponding channel will be accepted after the transfer ends.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.
- When DMA transfer is stopped due to the occurrence of an access error. See section 19.5, Processing on DMA Transfer Error.

19.2.14 EXDMA Software Start Register (EDMREQ)

Address(es): EXDMAC0.EDMREQ 8701 A01Dh, EXDMAC1.EDMREQ 8701 A05Dh, EXDMAC0_NS.EDMREQ 0701 A01Dh, EXDMAC1_NS.EDMREQ 0701 A05Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	Writing 1 to this bit triggers the generation of a DMA transfer request, and DMA transfer is started in response to the request	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Setting	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer. Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the EDMTMD.DCTG[1:0] bits are set to 00b (EXDMAC trigger source is software). Setting this bit is invalid when the EDMTMD.DCTG[1:0] bits are set to a value other than 00b. To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When DMA transfer is stopped due to the occurrence of an access error. See section 19.5, Processing on DMA Transfer Error.
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Setting)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

19.2.15 EXDMA Status Register (EDMSTS)

Address(es): EXDMAC0.EDMSTS 8701 A01Eh, EXDMAC1.EDMSTS 8701 A05Eh, EXDMAC0_NS.EDMSTS 0701 A01Eh, EXDMAC1_NS.EDMSTS 0701 A05Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	EXDMA Active Flag	0: EXDMAC operation is suspended. 1: EXDMAC is operating.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Only 0 can be written to clear the flag.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- Transfer of each single-repeat size of data is completed in repeat transfer mode with the EDMINT.RPTIE bit set to 1.
- Transfer of each single-block size of data is completed in block transfer mode with the EDMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the EDMINT.SARIE bit is set to 1 and the EDMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the EDMINT.DARIE bit is set to 1 and the EDMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the EDMCNT.DTE bit.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of the EDMCRAL[15:0] bits has become 0000h on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of the EDMCRBL[15:0] bits has become 0000h on completion of transfer (EDMTMD.TKP = 0), or the EDMCRBL[15:0] bits have been reloaded with the value of the EDMCRBH[15:0] bits (EDMTMD.TKP = 1))
- When the specified number of blocks has been transferred in block transfer mode (the value of the

EDMCRBL[15:0] bits has become 0000h on completion of transfer (EDMTMD.TKP = 0), or the EDMCRBL[15:0] bits have been reloaded with the value of the EDMCRBH[15:0] bits (EDMTMD.TKP = 1))

- When the specified number of blocks has been transferred in repeat-block transfer mode (the value of the EDMCRBL[15:0] bits has become 0000h on completion of transfer (EDMTMD.TKP = 0), or the EDMCRBL[15:0] bits have been reloaded with the value of the EDMCRBH[15:0] bits (EDMTMD.TKP = 1))

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the EDMCNT.DTE bit

ACT Flag (EXDMA Active Flag)

This flag indicates whether the EXDMAC is in the idle or active state.

[Setting condition]

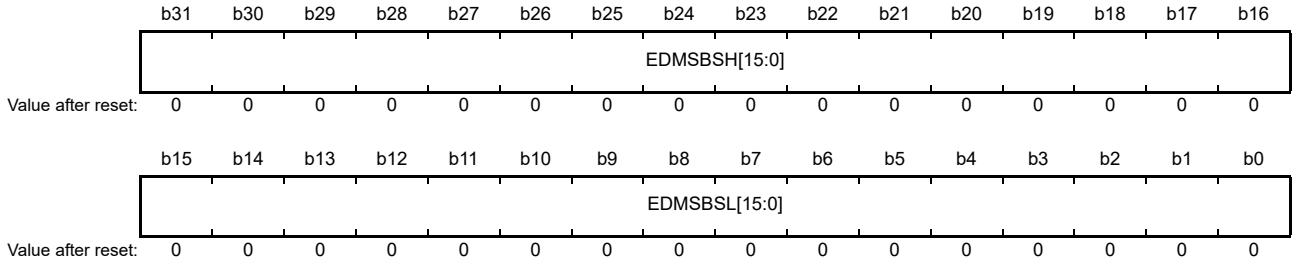
- When the EXDMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

19.2.16 EXDMA Source Buffer Size Register (EDMSBS)

Address(es): EXDMAC0.EDMSBS 8701 A028h, EXDMAC1.EDMSBS 8701 A068h,
EXDMAC0_NS.EDMSBS 0701 A028h, EXDMAC1_NS.EDMSBS 0701 A068h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EDMSBSL[15:0]	Lower Bits of Source Buffer Size	Functions as data transfer counter in repeat-block transfer mode	R/W
b31 to b16	EDMSBSH[15:0]	Upper Bits of Source Buffer Size	Specifies the repeat-area size in repeat-block transfer mode	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the same value for the EDMSBSH[15:0] and EDMSBSL[15:0] bits in repeat-block transfer mode. Write 00000000h to the EDMSBS register in normal, repeat and block transfer mode. The EDMSBSH[15:0] bits specify buffer size and the EDMSBSL[15:0] bits function as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by the EDMSBSH[15:0] bits. When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting the EDMSBSH[15:0] and EDMSBSL[15:0] bits to 0000h is prohibited. When final data of one buffer size is transferred, the EDMSBSL[15:0] bits are reloaded with the value of the EDMSBSH[15:0] bits. When address update mode is fixed address, this register is ignored. Table 19.3 lists the setting values of Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

Table 19.3 Available Setting for EDMSBS Register in Repeat-Block Transfer Mode

Source Address Update Mode (EDMAMD.SM)	Transfer Data Size (EDMTMD.SZ)	Available Setting for EDMSBSH[15:0] and EDMSBSL[15:0] Bits
Source address is fixed. (SM = 00b)	Don't care	0000h (EDMSBS is not used)
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0001h to FFFFh (1 to 65535)
	16 bits (SZ = 01b)	0001h to 7FFFh (1 to 32767)
	32 bits (SZ = 10b)	0001h to 3FFFh (1 to 16383)
Source address is incremented or decremented. (SM = 1xb)	Don't care	0000h (infinite) 0001h to FFFFh (1 to 65535)

19.2.17 EXDMA Destination Buffer Size Register (EDMDBS)

Address(es): EXDMAC0.EDMDBS 8701 A02Ch, EXDMAC1.EDMDBS 8701 A06Ch,
EXDMAC0_NS.EDMDBS 0701 A02Ch, EXDMAC1_NS.EDMDBS 0701 A06Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EDMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDMDBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EDMDBSL[15:0]	Lower Bits of Destination Buffer Size	Functions as data transfer counter in repeat-block transfer mode	R/W
b31 to b16	EDMDBSH[15:0]	Upper Bits of Destination Buffer Size	Specifies the repeat-area size in repeat-block transfer mode	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Set the same value for the EDMDBSH[15:0] and EDMDBSL[15:0] bits in repeat-block transfer mode. Write 00000000h to the EDMDBS register in normal, repeat and block transfer mode. The EDMDBSH[15:0] bits specify buffer size and the EDMDBSL[15:0] bits function as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by the EDMDBSH[15:0] bits. When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting the EDMDBSH[15:0] and EDMDBSL[15:0] bits to 0000h is prohibited. When final data of one buffer size is transferred, the EDMDBSL[15:0] bits are reloaded with the value of the EDMDBSH[15:0] bits. When address update mode is fixed address, this register is ignored. Table 19.4 lists the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

Table 19.4 Available Setting for EDMDBS Register in Repeat-Block Transfer Mode

Destination Address Update Mode (EDMAMD.DM)	Transfer Data Size (EDMTMD.SZ)	Available Setting for EDMDBSH[15:0] and EDMDBSL[15:0] Bits
Destination address is fixed. (DM = 00b)	Don't care	0000h (EDMDBS is not used)
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0001h to FFFFh (1 to 65535)
	16 bits (SZ = 01b)	0001h to 7FFFh (1 to 32767)
	32 bits (SZ = 10b)	0001h to 3FFFh (1 to 16383)
Destination address is incremented or decremented. (DM = 1xb)	Don't care	0000h (infinite) 0001h to FFFFh (1 to 65535)

19.2.18 EXDMA Bufferable Write Enable Register (EDMBWR)

Address(es): EXDMAC0.EDMBWR 8701 A030h, EXDMAC1.EDMBWR 8701 A070h,
EXDMAC0_NS.EDMBWR 0701 A030h, EXDMAC1_NS.EDMBWR 0701 A070h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BWE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BWE	Bufferable Write Enable	0: Disables bufferable write 1: Enables bufferable write	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

BWE Bit (Bufferable Write Enable)

This bit specifies enabling or disabling of bufferable writing. If this bit is 1, even if the write access as the EXDMAC is completed, the actual slave writing may not have ended. When an error occurs during writing to a bufferable area (area in which a buffer is interposed during writing) of a slave device, the detected error is referred to as a “bufferable write error”. DMA transfer is not terminated at this time. For details, see section 19.8.1, Bus Response to DMA Write Access.

19.2.19 EXDMAC Module Start Register (EDMAST)

Address(es): EXDMAC.EDMAST 8701 A800h,
EXDMAC_NS.EDMAST 0701 A800h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	EXDMAC Module Start	0: EXDMAC module stop 1: EXDMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DMST Bit (EXDMAC Module Start)

When this bit is set to 1, EXDMAC is ready to accept transfer requests for all channels. When 1 is written to the EDMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (EXDMAC module start), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time. When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer can be resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

19.2.20 EXDMAC Error Channel Register (EDMECHR)

Address(es): EXDMAC.EDMECHR 8701 A840h, EXDMAC_NS.EDMECHR 0701 A840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMESTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DMERRW	—	—	—	DMECHSAM	—	—	—	—	DMECH[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DMECH[3:0]	EXDMAC Error Channel Flag	Indicates the channel number causing the error 0000: Error occurred on channel 0 0001: Error occurred on channel 1	R
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	DMECHSAM	EXDMAC Error Channel Security Attribution Monitor Flag	Indicates the security attribution of a channel causing the error 0: Secure channel 1: Non-secure channel	R
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	DMERRW	EXDMAC Error Access Read/ Write Attribute Flag	Indicates the read/write attribute of the error access 0: Read access 1: Write access	R
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DMESTA	EXDMAC Error Status Flag	0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

When reading this register, it can be accessed from both secure and non-secure access. When writing this register, it depends on the value of the EDMECHR.DMECHSAM flag.

- When the value of the EDMECHR.DMECHSAM flag is 1, it can be accessed from secure and non-secure access.
- When the value of the EDMECHR.DMECHSAM flag is 0, it can be accessed from secure. An error is returned when write access is performed in a non-secure access.

This register is cleared by a reset. On debugging a program, select an interrupt instead of a reset in the corresponding bits of the registers listed in the interrupt/reset selection column of Table 19.18 in section 19.6.2, Transfer Error Interrupt.

DMECH[3:0] Flag (EXDMAC Error Channel Flag)

When a transfer error due to DMA transfer occurs, this flag stores the channel number of the EXDMAC that was violated. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- When the DMA transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA flag.

DMECHSAM Flag (EXDMAC Error Channel Security Attribution Monitor Flag)

When a transfer error due to DMA transfer occurs, this flag indicates the security attribution of the violating EXDMAC channel. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- When the DMA transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA flag.

DMERRW Flag (EXDMAC Error Access Read/Write Attribute Flag)

Indicates the read/write attribute of the error access.

[Setting condition]

- A DMA transfer error occurs when writing data.

[Clearing condition]

- When 1 is written to DMESTA flag.

DMESTA Flag (EXDMAC Error Status Flag)

Indicates whether or not a DMA transfer error occurred. DMECH[3:0] bits, DMECHSAM flag, DMESTA flag are cleared by writing 1 to DMESTA flag. Writing 0 to DMESTA flag is ignored. When reset was selected in BUSMPU.MMPUOAD.OAD and BUS.MSAOAD.OAD bits, this register is also reset. Select NMI when you want to debug the program.

[Setting condition]

- When the DMA transfer error occurs.

[Clearing condition]

- When 1 is written to DMESTA flag.

Note: When DMECHSAM = 1, it can be cleared in the secure state and non-secure state. When DMECHSAM = 0, it cannot be cleared in the non-secure state.

19.2.21 EXDMAC Control Register (EDMCTL)

Address(es): EXDMAC.EDMCTL 8701 A810h,
EXDMAC_NS.EDMCTL 0701 A810h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PR	Priority Control Select	0: Fixed priority 1: Round-robin	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Write to this register while the EXDMAC operation is stopped or DMA transfer is disabled. Do not write to it during data transfer.

PR Bit (Priority Control Select)

When the EDMCTL.PR bit is set to 0, arbitration is performed in the fixed priority mode of channel 0 > channel 1 (channel 0: highest priority) and DMA transfer is started. When the EDMCTL.PR bit is set to 1, arbitration is performed in the round-robin mode and DMA transfer is started.

19.2.22 EXDMAC Peripheral Module Request Flag Register (EDMPRF)

Address(es): EXDMAC0.EDMPRF 8701 A03Eh, EXDMAC1.EDMPRF 8701 A07Eh, EXDMAC0_NS.EDMPRF 0701 A03Eh, EXDMAC1_NS.EDMPRF 0701 A07Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PREQ	Peripheral Module Request Flag	0: No request 1: Requested	R/W*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. writing 0 has no effect.

The EDMPRF register detects a DMA transfer request generated by peripheral module.

PREQ Flag (Peripheral Module Request Flag)

The PREQ flag indicates that a DMA transfer request has been detected from a peripheral module.

[Setting condition]

- When a DMA transfer request occurs form a peripheral module.

[Clearing condition]

- When a DMA transfer request is generated from a peripheral module and the DMA transfer starts.
- When 1 is written to the PREQ flag.

19.3 Operation

19.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the EDMCRAL[15:0] bits. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting the EDMCRB register is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations. Table 19.5 lists the register update operation in normal transfer mode, and Figure 19.2 shows an operation example in normal transfer mode.

Table 19.5 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
EDMSAR	Transfer source address	Increment/decrement/fixed/offset addition
EDMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
EDMCRA.EDMCRAL[15:0]	Transfer counter	Decremented by one/not updated (in free running mode)
EDMCRA.EDMCRAH[9:0]	—	Not updated (Not used in normal transfer mode)
EDMCRB	—	Not updated (Not used in normal transfer mode)

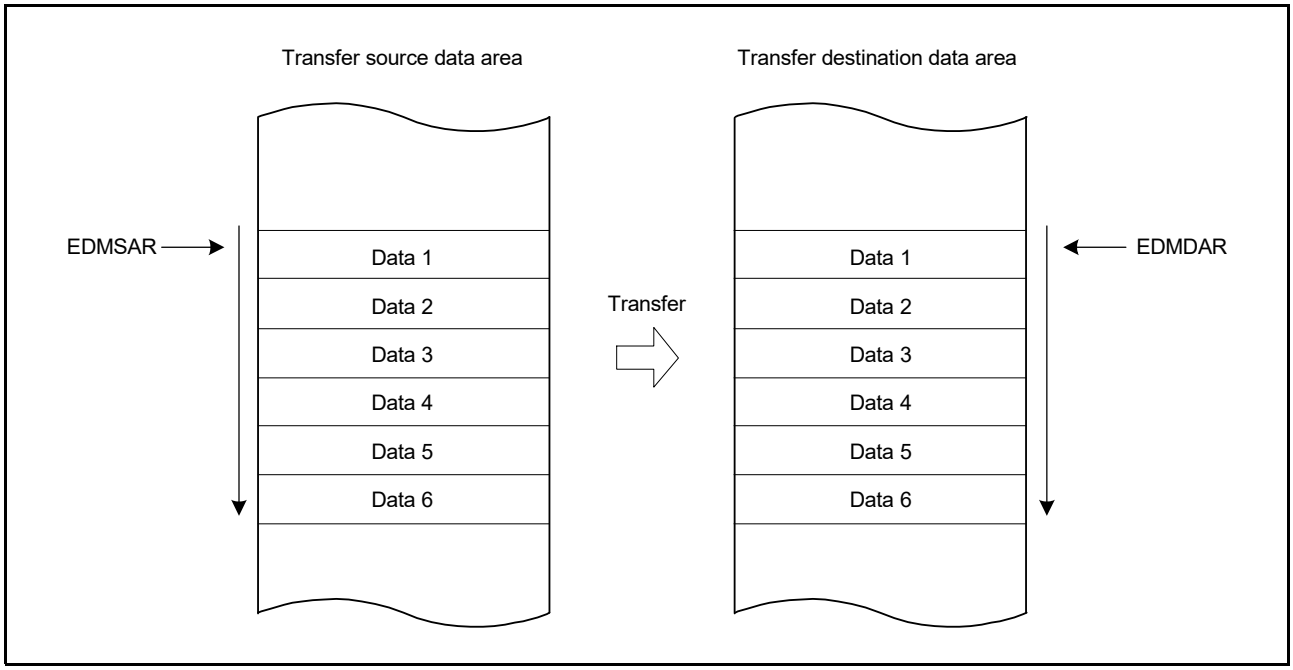


Figure 19.2 Operation Example in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request. A maximum of 1K data can be set as a total repeat transfer size using the EDMCRA register. A maximum of 64K can be set as the number of repeat transfer operations using the EDMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size. Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (EDMSAR or EDMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the EDMCNT.DTE bit in the repeat size end interrupt handling. A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations. Table 19.6 lists the register update operation in repeat transfer mode, and Figure 19.3 shows an operation example in repeat transfer mode.

Table 19.6 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When the EDMCRA.EDMCRAL [15:0] bits are not 1	When the EDMCRA.EDMCRAL[15:0] bits are 1. (Transfer of the Last Data in Repeat Size)
EDMSAR	Transfer source address	Increment/decrement/fixed/ offset addition	• EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • EDMTMD.DTS[1:0] = 01b Initial value of EDMSAR • EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
EDMDAR	Transfer destination address	Increment/decrement/fixed/ offset addition	• EDMTMD.DTS[1:0] = 00b Initial value of DMDAR • EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
EDMCRA.EDMCRAH [9:0]	Repeat size	Not updated	Not updated
EDMCRA.EDMCRAL [15:0]	Transfer counter	Decremented by one	EDMCRA.EDMCRAH[9:0]
EDMCRB.EDMCRBH [15:0]	Number of repeat transfer operations	Not updated	Not updated
EDMCRB.EDMCRBL [15:0]	Count of repeat transfer operations	Not updated	Decremented by one

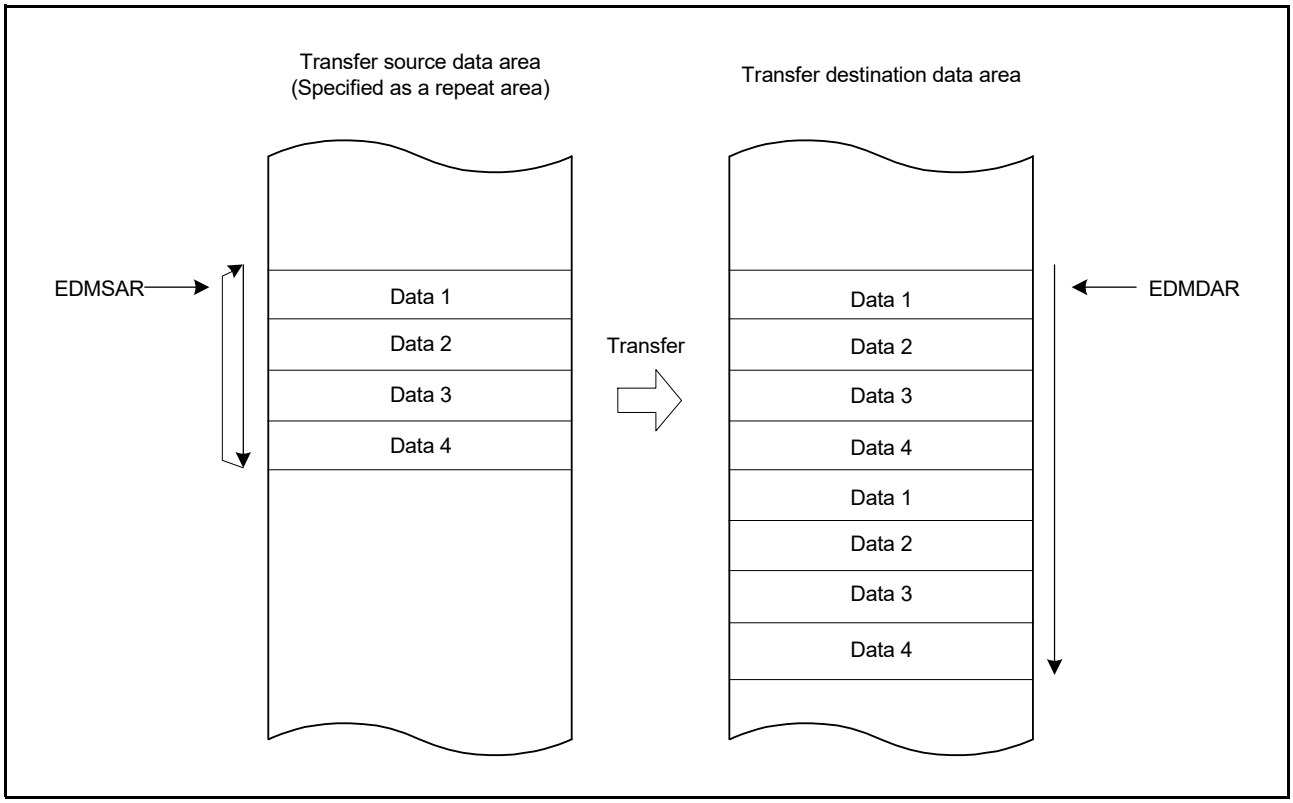


Figure 19.3 Operation Example in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request. A maximum of 1K data can be set as a total block transfer size using the EDMCRA register. A maximum of 64K can be set as the number of block transfer operations using the EDMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size. Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (EDMSAR or EDMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the EDMCNT.DTE bit in the repeat size end interrupt handling. Transfer end interrupt request can be generated after completion of the specified number of block transfer operations. Table 19.7 lists the register update operation in block transfer mode, and Figure 19.4 shows an operation example in block transfer mode.

Table 19.7 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
EDMSAR	Transfer source address	• EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • EDMTMD.DTS[1:0] = 01b Initial value of EDMSAR • EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
EDMDAR	Transfer destination address	• EDMTMD.DTS[1:0] = 00b Initial value of EDMDAR • EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
EDMCRA.EDMCRAH[9:0]	Block size	Not updated
EDMCRA.EDMCRAL[15:0]	Transfer count	EDMCRA.EDMCRAH[9:0]
EDMCRB.EDMCRBH[15:0]	Number of block transfer operations	Not updated
EDMCRB.EDMCRBL[15:0]	Count of block transfer operations	Decremented by one

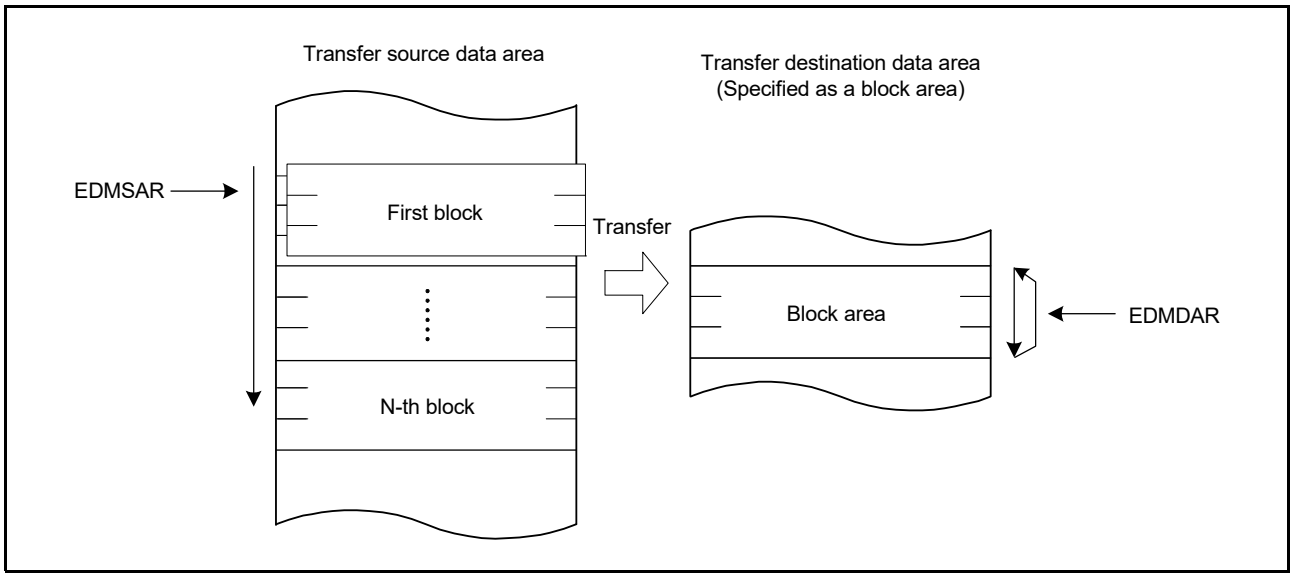


Figure 19.4 Operation Example in Block Transfer Mode

(4) Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.
Repeat function: Added function (ring buffer) to repeat specified address area.
Offset function: Multiple areas with offset can be specified within one block transfer.
The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer. Figure 19.5 shows an example of adding a repeat function to the transfer destination. Figure 19.6 shows repeat-block transfer with an offset to the transfer destination. In repeat-block transfer mode, a single block data is transferred by one transfer request. A maximum of 1K data can be set as a total block transfer size using EDMCRA. A maximum of 64K can be set as the number of block transfer operations using EDMCRB; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.
Generation of a transfer end interrupt request on completion of a specified number of transfer operations is selectable.

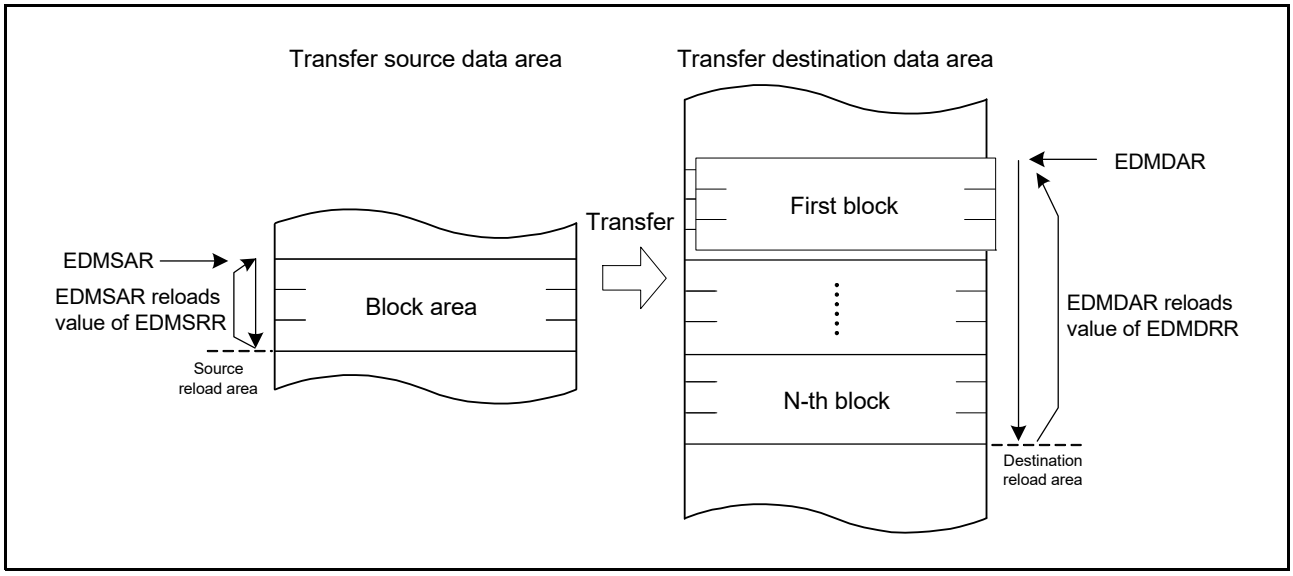


Figure 19.5 Operation Example in Repeat-Block Transfer Mode

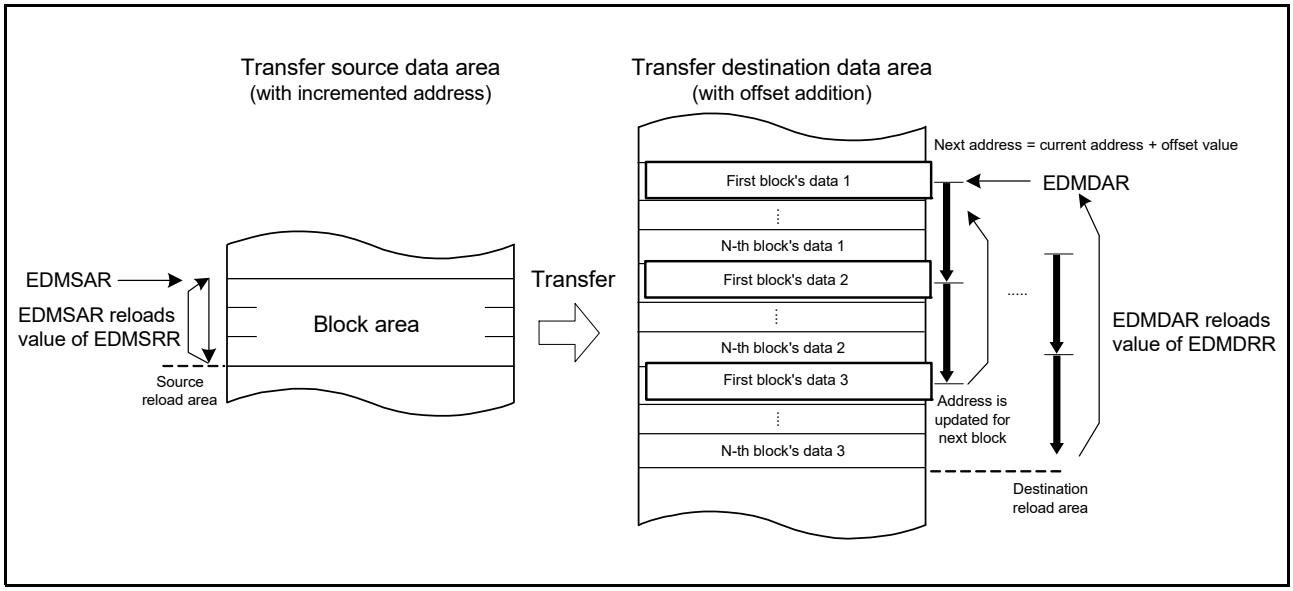


Figure 19.6 Operation Example in Repeat-Block Transfer Mode with Offset Addition

Table 19.8 to Table 19.13 list the register update operations in repeat-block transfer mode.

Table 19.8 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Fixed Address EDMAMD.SM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)	
		EDMCRA.EDMCRAL [15:0] is not 1	EDMCRB.EDMCRBL [15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1
EDMSRR	Transfer source reload address	Not updated	Not updated	Not updated
EDMSAR	Transfer source address	Not updated	Not updated	Not updated
EDMCRA.EDMCRAH [9:0]	Block size	Not updated	Not updated	Not updated
EDMCRA.EDMCRAL [15:0]	Block size count	Decremented by one	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]
EDMCRB.EDMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
EDMCRB.EDMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]

Table 19.9 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Fixed Address EDMAMD.DM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)	
		EDMCRA.EDMCRAL [15:0] is not 1	EDMCRB.EDMCRBL [15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1
EDMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
EDMDAR	Transfer destination address	Not updated	Not updated	Not updated
EDMCRA.EDMCRAH [9:0]	Block size	Not updated	Not updated	Not updated
EDMCRA.EDMCRAL [15:0]	Block size count	Decremented by one	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]
EDMCRB.EDMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
EDMCRB.EDMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]

Table 19.10 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Incremented or Decremented Address EDMAMD.SM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		EDMSBS.EDMSBSL[15:0] is not 1			EDMSBS.EDMSBSL[15:0] is 1		
			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)	
		EDMCRA. EDMCRAL [15:0] is not 1	EDMCRB. EDMCRBL [15:0] is not 1	EDMCRB. EDMCRBL[15:0] is 1	EDMCRA. EDMCRAL [15:0] is not 1	EDMCRB. EDMCRBL [15:0] is not 1	EDMCRB. EDMCRBL[15:0] is 1
EDMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMSAR	Transfer source address	EDMTMD.SM [1:0] = 10b Incremented by Data Size EDMTMD.SM [1:0] = 11b Decremented by Data Size	EDMTMD.SM [1:0] = 10b Incremented by Data Size EDMTMD.SM [1:0] = 11b Decremented by Data Size	EDMTMD.SM [1:0] = 10b Incremented by Data Size EDMTMD.SM [1:0] = 11b Decremented by Data Size	EDMSRR	EDMSRR	EDMSRR
EDMCRA. EDMCRAH [9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRA. EDMCRAL [15:0]	Block size count	Decremented by one	EDMCRA. EDMCRAH[9:0]	EDMCRA. EDMCRAH[9:0]	Decremented by one	EDMCRA. EDMCRAH[9:0]	EDMCRA. EDMCRAH[9:0]
EDMSBS. EDMSBSH [15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMSBS. EDMSBSL [15:0]	Count of transfer data in source buffer	Decremented by one	Decremented by one	Decremented by one	EDMSBS. EDMSBSH [15:0]	EDMSBS. EDMSBSH [15:0]	EDMSBS. EDMSBSH[15:0]
EDMCRB. EDMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRB. EDMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB. EDMCRBH[15:0]	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB. EDMCRBH[15:0]

Table 19.11 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Incremented or Decremented Address EDMAMD.DM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		EDMSBS.EDMSBSL[15:0] is not 1			EDMSBS.EDMSBSL[15:0] is 1		
			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)			EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)	
		EDMCRA. EDMCRAL [15:0] is not 1	EDMCRB. EDMCRBL [15:0] is not 1	EDMCRB. EDMCRBL[15:0] is 1	EDMCRA. EDMCRAL [15:0] is not 1	EDMCRB. EDMCRBL [15:0] is not 1	EDMCRB. EDMCRBL[15:0] is 1
EDMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMDAR	Transfer destination address	EDMTMD.DM [1:0] = 10b Incremented by Data Size EDMTMD.DM [1:0] = 11b Decremented by Data Size	EDMTMD.DM [1:0] = 10b Incremented by Data Size EDMTMD.DM [1:0] = 11b Decremented by Data Size	EDMTMD.DM [1:0] = 10b Incremented by Data Size EDMTMD.DM [1:0] = 11b Decremented by Data Size	EDMDRR	EDMDRR	EDMDRR
EDMCRA. EDMCRAH [9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRA. EDMCRAL [15:0]	Block size count	Decremented by one	EDMCRA. EDMCRAH [9:0]	EDMCRA. EDMCRAH[9:0]	Decremented by one	EDMCRA. EDMCRAH[9:0]	EDMCRA. EDMCRAH[9:0]
EDMDBS. EDMDBSH [15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMDBS. EDMDBSL [15:0]	Count of transfer data in destination buffer	Decremented by one	Decremented by one	Decremented by one	EDMDBS. EDMDBSH [15:0]	EDMDBS. EDMDBSH [15:0]	EDMDBS. EDMDBSH[15:0]
EDMCRB. EDMCRBH [15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRB. EDMCRBL [15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB. EDMCRBH[15:0]	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB. EDMCRBH[15:0]

Table 19.12 Register Update Operation Associated with Source Area in Repeat-Block Transfer Mode (Offset Addition EDMAMD.SM[1:0] = 01b)

Register	Function	EDMCRA.EDMCRAL[15:0] is not 1	EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)			
			EDMSBS.EDMSBSL[15:0] is not 1		EDMSBS.EDMSBSL[15:0] is 1	
			EDMCRB.EDMCRBL[15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1	EDMCRB.EDMCRBL[15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1
EDMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated
EDMSAR	Transfer source address	Offset addition by EDMSBS.EDMSBSH[15:0]	EDMAMD.SADR = 0 EDMSRR EDMAMD.SADR = 1 EDMSRR + (EDMSBSH – EDMSBSL) × transfer data size	EDMAMD.SADR = 0 EDMSRR EDMAMD.SADR = 1 EDMSRR + (EDMSBSH – EDMSBSL) × transfer data size	EDMSRR	EDMSRR
EDMCRA.EDMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRA.EDMCRAL[15:0]	Block size count	Decremented by one	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]
EDMSBS.EDMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
EDMSBS.EDMSBSL[15:0]	Count of transfer data in source buffer	Not updated	Decremented by one	Decremented by one	EDMSBS.EDMSBSH[15:0]	EDMSBS.EDMSBSH[15:0]
EDMCRB.EDMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRB.EDMCRBL[15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]

Table 19.13 Register Update Operation Associated with Destination Area in Repeat-Block Transfer Mode (Offset Addition EDMAMD.DM[1:0] = 01b)

Register	Function	EDMCRA.EDMCRAL[15:0] is not 1	EDMCRA.EDMCRAL[15:0] is 1 (single block is transferred)			
			EDMSBS.EDMSBSL[15:0] is not 1		EDMSBS.EDMSBSL[15:0] is 1	
			EDMCRB.EDMCRBL[15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1	EDMCRB.EDMCRBL[15:0] is not 1	EDMCRB.EDMCRBL[15:0] is 1
EDMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
EDMDAR	Transfer destination address	Offset addition by EDMDBS.EDMDBSH[15:0]	EDMAMD.DADR = 0 EDMDRR EDMAMD.DADR = 1 EDMDRR + (EDMDBSH – EDMDBSL) × transfer data size	EDMAMD.DADR = 0 EDMDRR EDMAMD.DADR = 1 EDMDRR + (EDMDBSH – EDMDBSL) × transfer data size	EDMDRR	EDMDRR
EDMCRA.EDMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRA.EDMCRAL[15:0]	Block size count	Decremented by one	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]	EDMCRA.EDMCRAH[9:0]
EDMDBS.EDMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
EDMDBS.EDMDBSL[15:0]	Count of transfer data in destination buffer	Not updated	Decremented by one	Decremented by one	EDMDBS.EDMDBSH[15:0]	EDMDBS.EDMDBSH[15:0]
EDMCRB.EDMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
EDMCRB.EDMCRBL[15:0]	Count of block transfer operations	Not updated	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]	Decremented by one	EDMTMD.TKP = 0 0 EDMTMD.TKP = 1 EDMCRB.EDMCRBH[15:0]

For more information about address update operation in repeat-block transfer mode, see section 19.3.5, Address Update Function in Repeat-Block Transfer Mode.

19.3.2 Extended Repeat Area Function

The EXDMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas. The extended repeat areas can be specified separately to the transfer source address register (EDMSAR) and transfer destination address register (EDMDAR). The extended repeat area on the source address is specified by the EDMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the EDMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides. However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area. When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt due to an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the EDMINT.SARIE bit is set to 1, the EDMSTS.ESIF flag is set to 1 and the EDMCNT.DTE bit is set to 0 to stop DMA transfer. At this time, if the EDMINT.ESIE bit is set to 1, an interrupt due to an extended repeat area overflow is requested. When the EDMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the EDMCNT.DTE bit in the interrupt handling. Figure 19.7 shows an example of the extended repeat area operation.

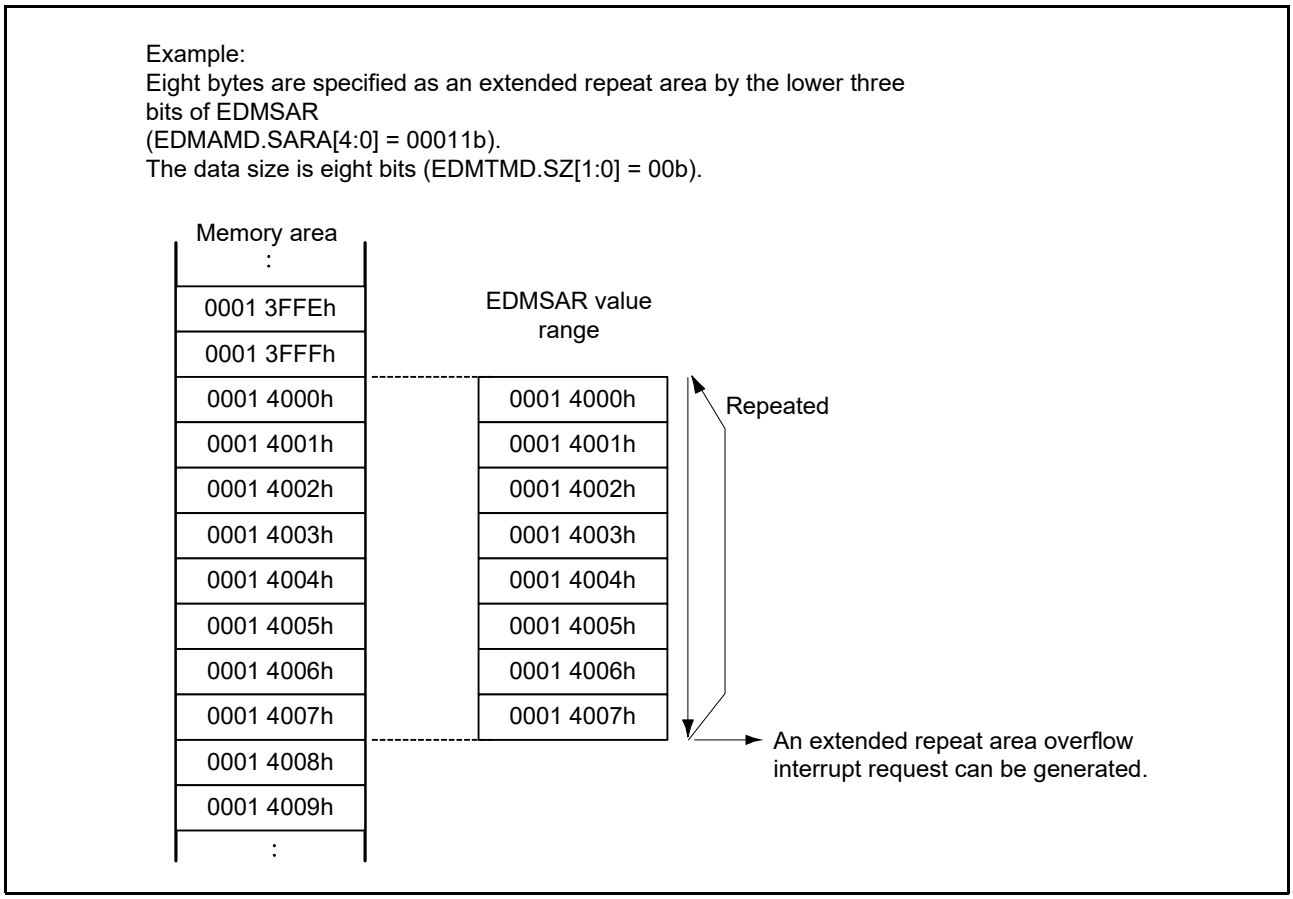


Figure 19.7 Example of Extended Repeat Area Operation

When an interrupt due to an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration. When a transfer is stopped by an interrupt due to an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt due to the overflow is suspended until transfer of the block is completed, and the transfer overruns. Figure 19.8 shows an example when the extended repeat area function is used in block transfer mode.

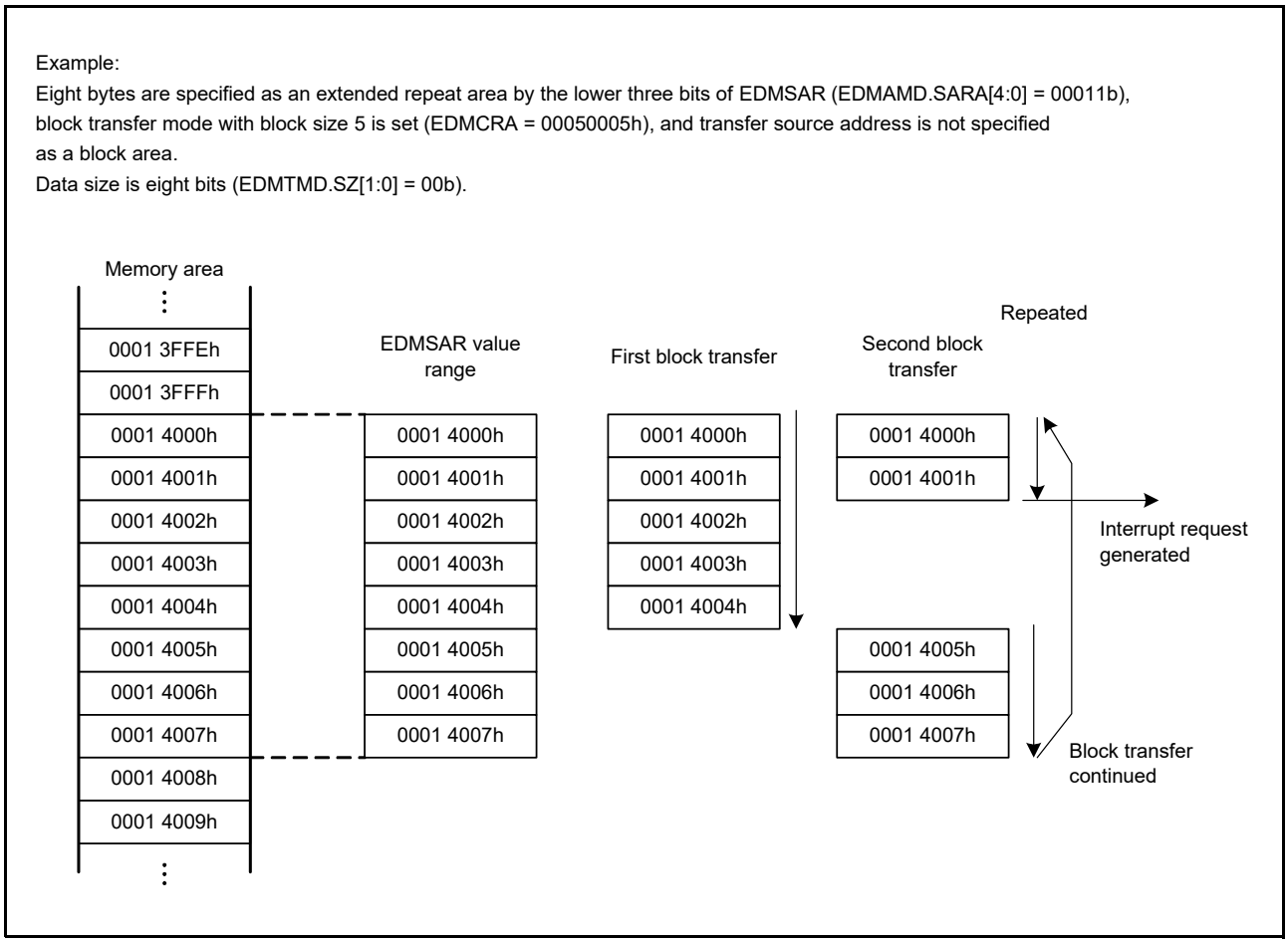


Figure 19.8 Example of Extended Repeat Area Function in Block Transfer Mode

19.3.3 Free-Running Function

The EXDMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

(1) In Normal Transfer Mode

In normal transfer mode, when EDMCRA.EDMCRAL[15:0] bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped. For more information, see section (1), Normal Transfer Mode.

(2) In Other Transfer Modes

In repeat, block and repeat-block transfer mode, the EXDMAC supports free-running function using the EDMTMD.TKP bit. If the EDMTMD.TKP bit is set to 1, the transfer is not stopped by completion of specified total number of transfer operations and the value of the EDMCRBH[9:0] bits is reloaded repeatedly. Figure 19.9 shows an example of block transfer operation without free-running function.

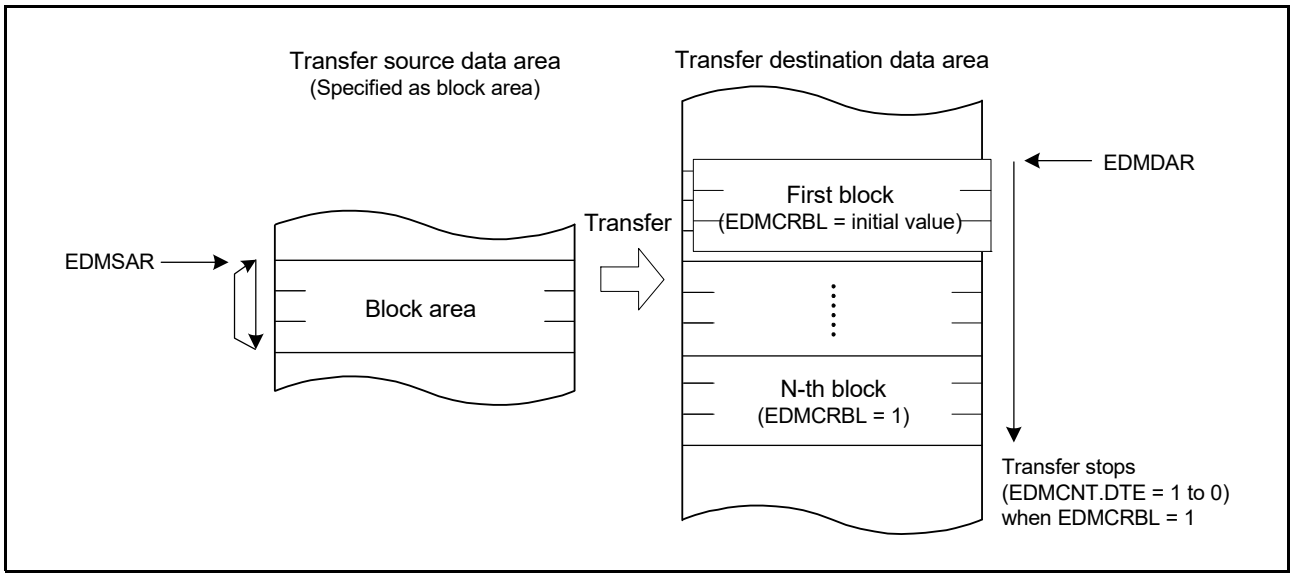


Figure 19.9 Operation in Block Transfer Mode with EDMTMD.TKP = 0

Figure 19.10 shows an example of block transfer operation with free-running function.

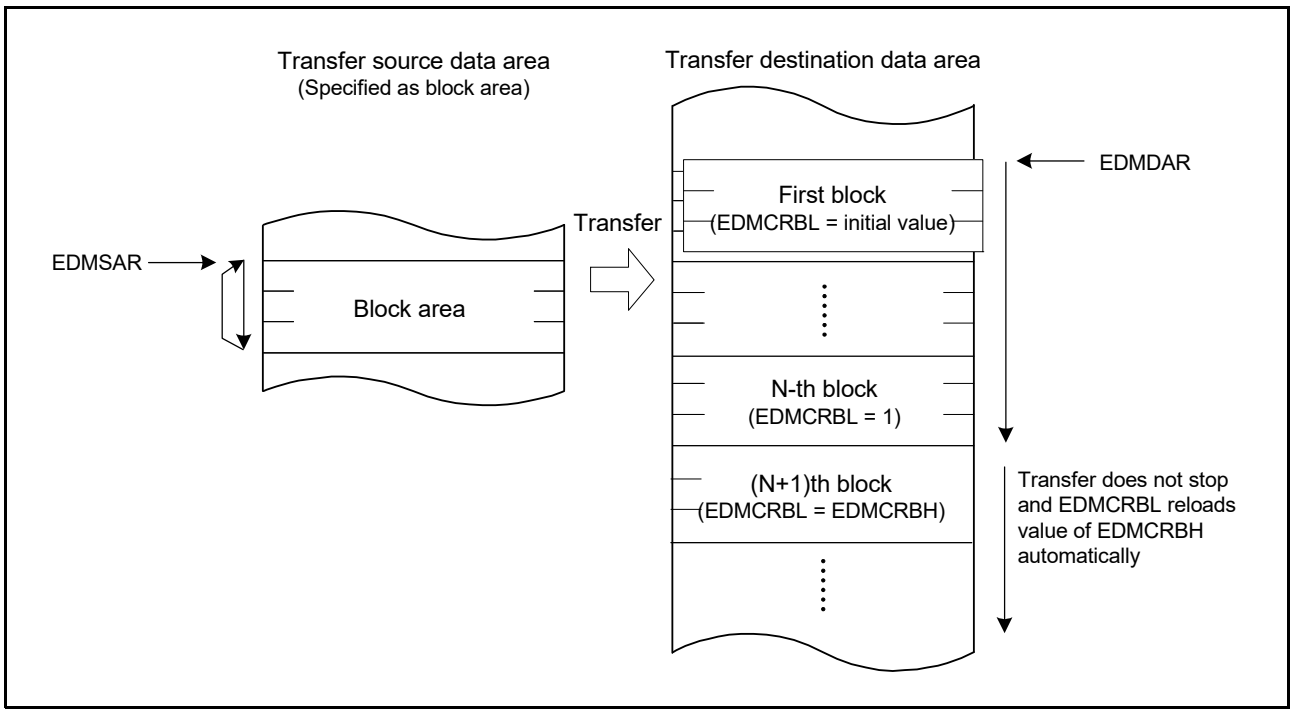


Figure 19.10 Operation in Block Transfer Mode with EDMTMD.TKP = 1

19.3.4 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (EDMOFR) is added to the address every time the EXDMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas. Offset subtraction can also be realized by setting a negative value in EDMOFR register. In this case, the negative value must be 2's complement. EDMSBS or EDMDBS registers are used instead of EDMOFR register in repeat-block transfer mode. For more information, see section (4), Repeat-Block Transfer Mode and section 19.3.5, Address Update Function in Repeat-Block Transfer Mode. Table 19.14 lists the address update method in each address update mode.

Table 19.14 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of EDMAMD.SM[1:0] and EDMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different EDMTMD.SZ[1:0] Settings)			
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b	SZ[1:0] = 11b
Address fixed	00b	Fixed			
Offset addition	01b	+EDMOFR*1			
Increment	10b	+1	+2	+4	+8
Decrement	11b	–1	–2	–4	–8

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: Bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 19.11 shows an example of address updating using offset addition.

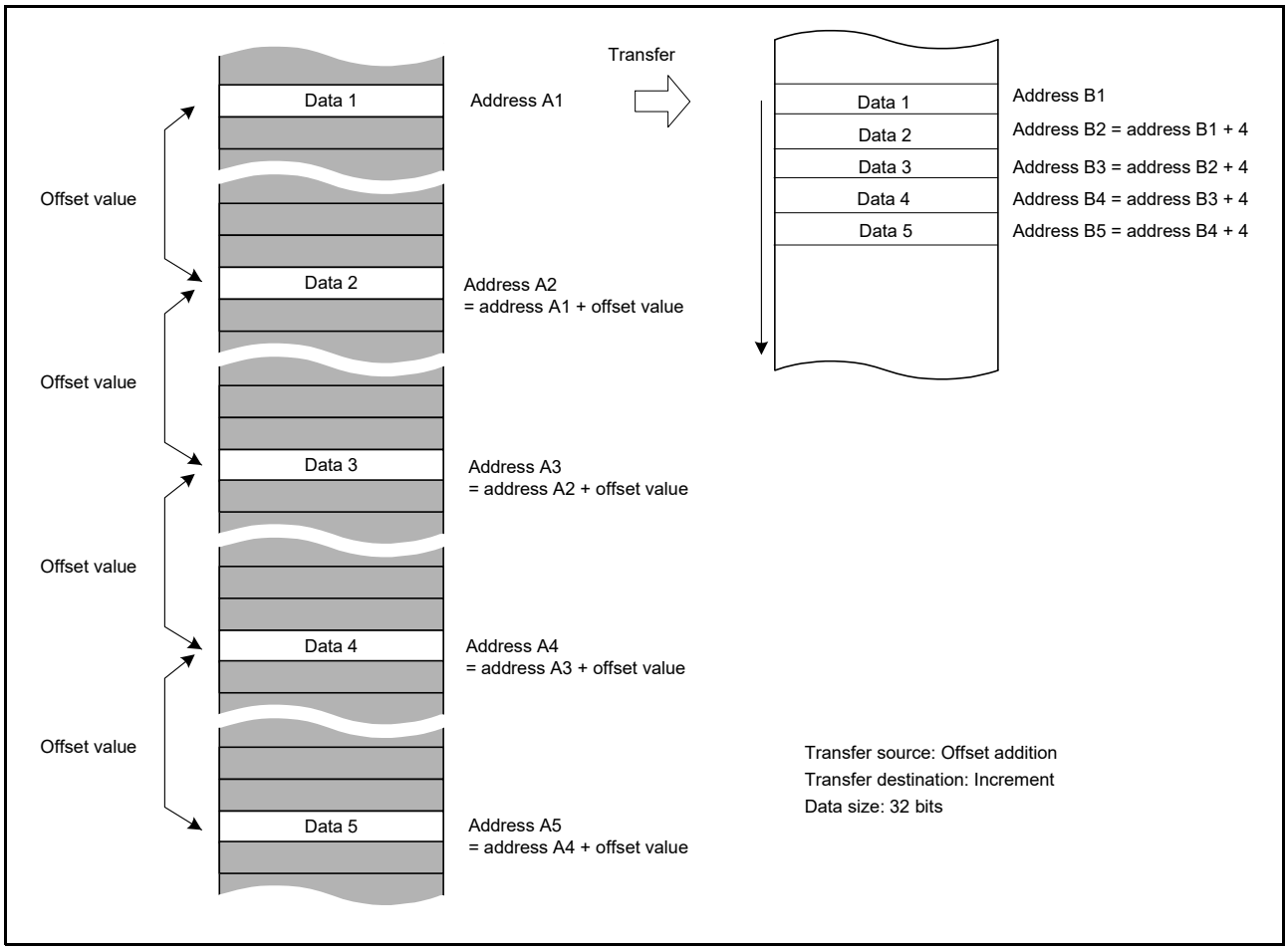


Figure 19.11 Example of Address Updating by Offset Addition

In Figure 19.11, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 19.12 shows the XY conversion using offset addition in repeat transfer mode. Settings are as follows:

- EDMAMD.SM[1:0]: Source address update mode select: Offset addition
- EDMAMD.DM[1:0]: Destination address update mode select: Destination address is incremented.
- EDMTMD.SZ[1:0]: Data transfer size: 32 bits
- EDMTMD.MD[1:0]: Transfer mode select: Repeat transfer
- EDMTMD.DTS[1:0]: Repeat area select: The source is specified as the repeat area.
- EDMOFR: Offset address: 10h
- EDMCRA.EDMCRAH[9:0]: Repeat size: 4h
- EDMINT.RPTIE: The repeat size end interrupt is enabled.

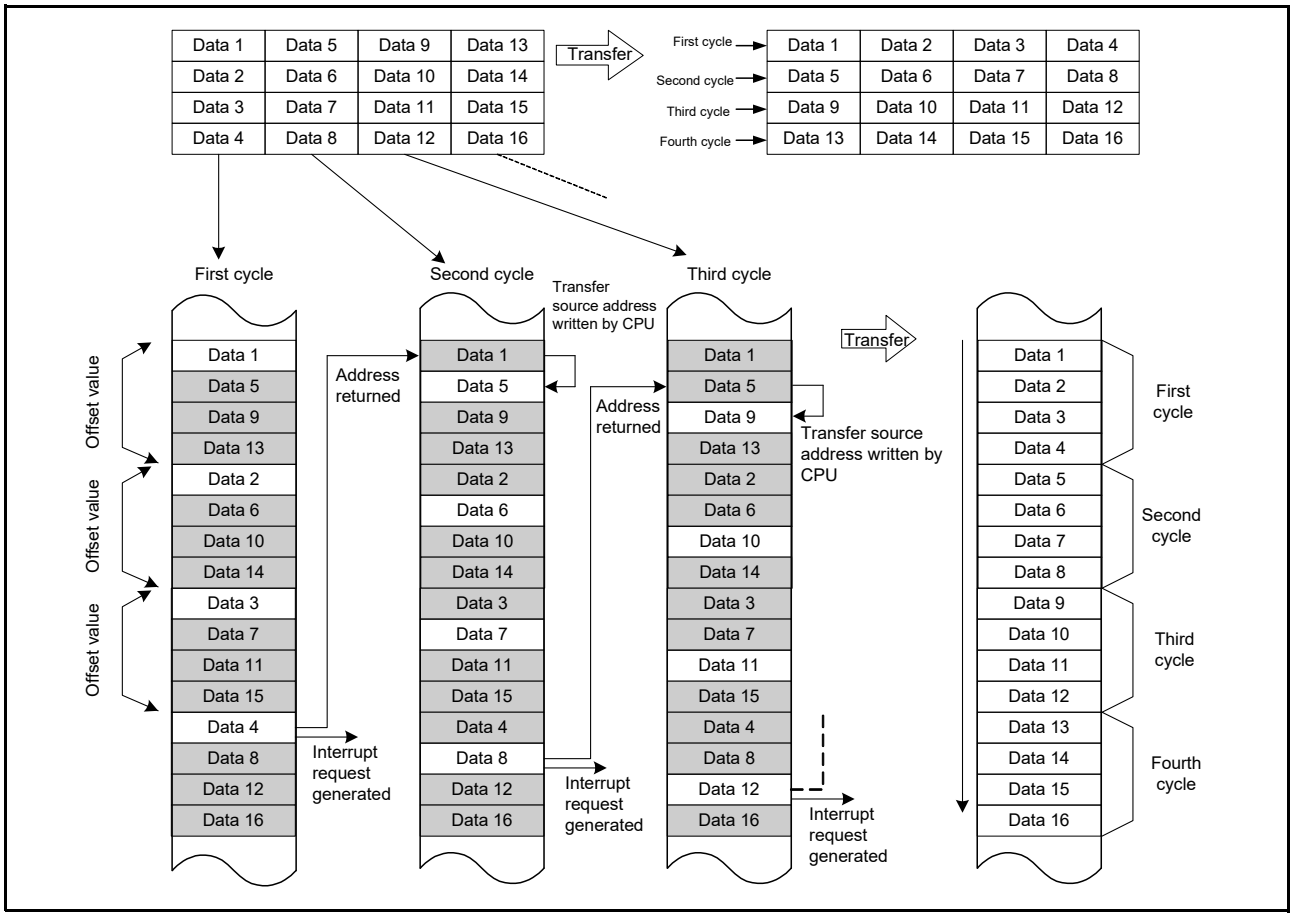


Figure 19.12 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- EDMSAR register: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- EDMCNT register: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 19.13 shows a flowchart of the XY conversion.

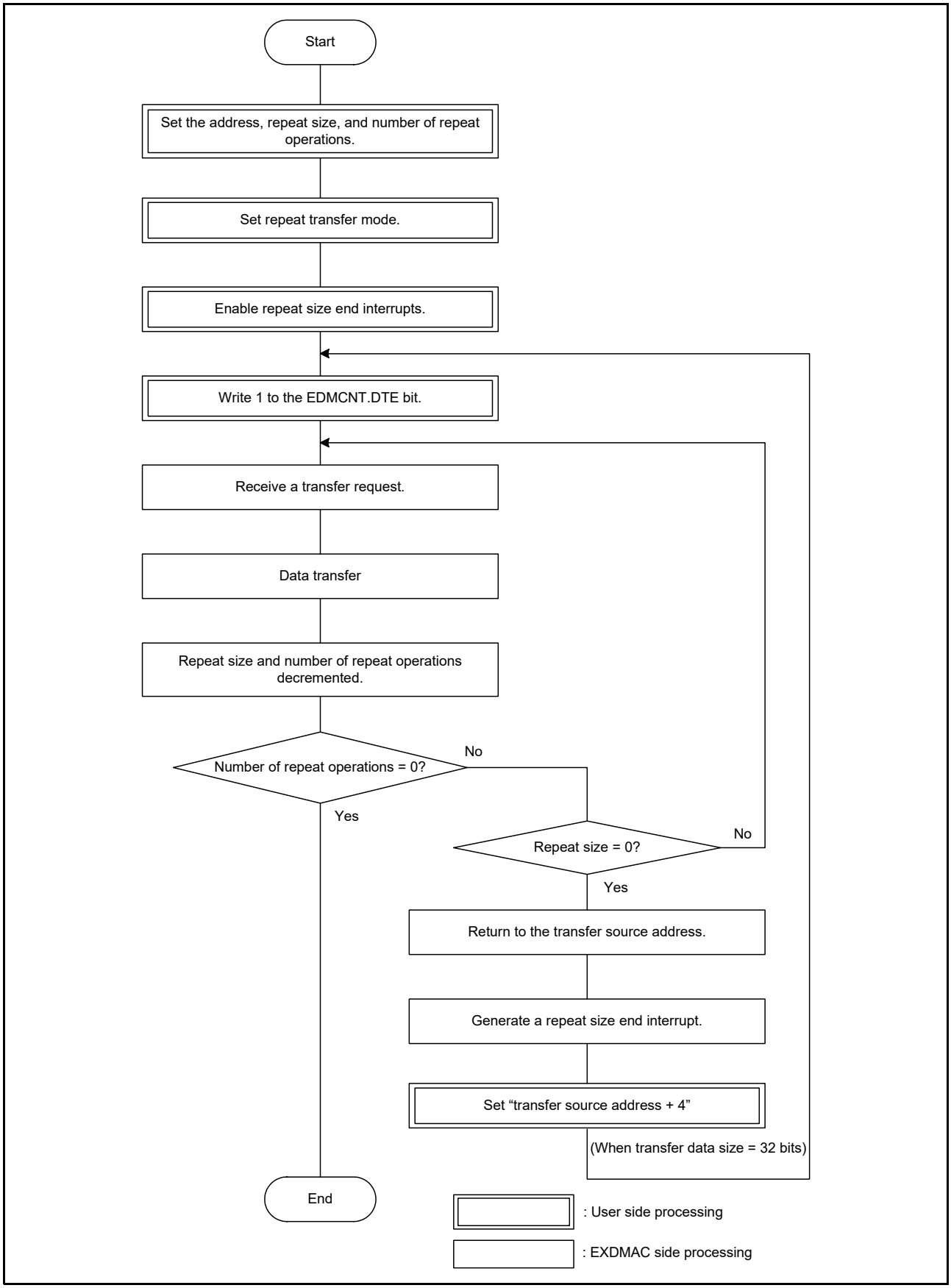


Figure 19.13 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

19.3.5 Address Update Function in Repeat-Block Transfer Mode

Repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior of the address update is different from these two modes. Here are the details of the address update function in repeat-block transfer mode.

(1) Fixed Address Mode

When EDMAMD.SM[1:0] bits is set to 00b, the address update mode of the source is fixed address. And when EDMAMD.DM[1:0] bits is set to 00b, the address update mode of the destination is fixed address. In fixed address, the address is not updated from the initial value of EDMSAR and EDMDAR registers. If the block size (EDMCRA) is larger than 1, the same data will be transferred multiple times for one request. Figure 19.14 shows address update in fixed address.

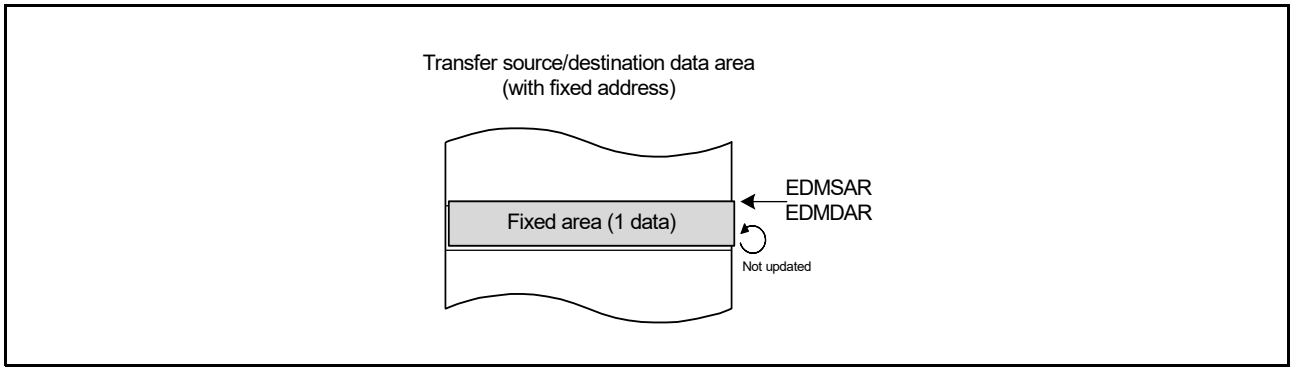


Figure 19.14 Address Update in Fixed Address

(2) Incremental and Decremental Address Mode

When EDMAMD.SM[1:0] bits is set to 10b, the address update mode of the source is incremental address. And when EDMAMD.DM[1:0] bits is set to 10b, the address update mode of the destination is incremental address. When EDMAMD.SM[1:0] bits is set to 11b, the address update mode of the source is decremental address. And when EDMAMD.DM[1:0] bits is set to 11b, the address update mode of the destination is decremental address. In these update modes, the address is incremented or decremented according to the setting of EDMTMD.SZ[1:0] bits. In these update modes EDMSBS and EDMDBS registers indicates a reload area. The unit of EDMSBS and EDMDBS registers is "number of data". At the start of transfer, EDMSBSL and EDMDBSL registers, which is the lower 16 bits of EDMSBS and EDMDBS registers, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, EDMSAR and EDMDAR registers reloads the value of EDMSRR and EDMDRR registers. Figure 19.15 shows address update in incremental address.

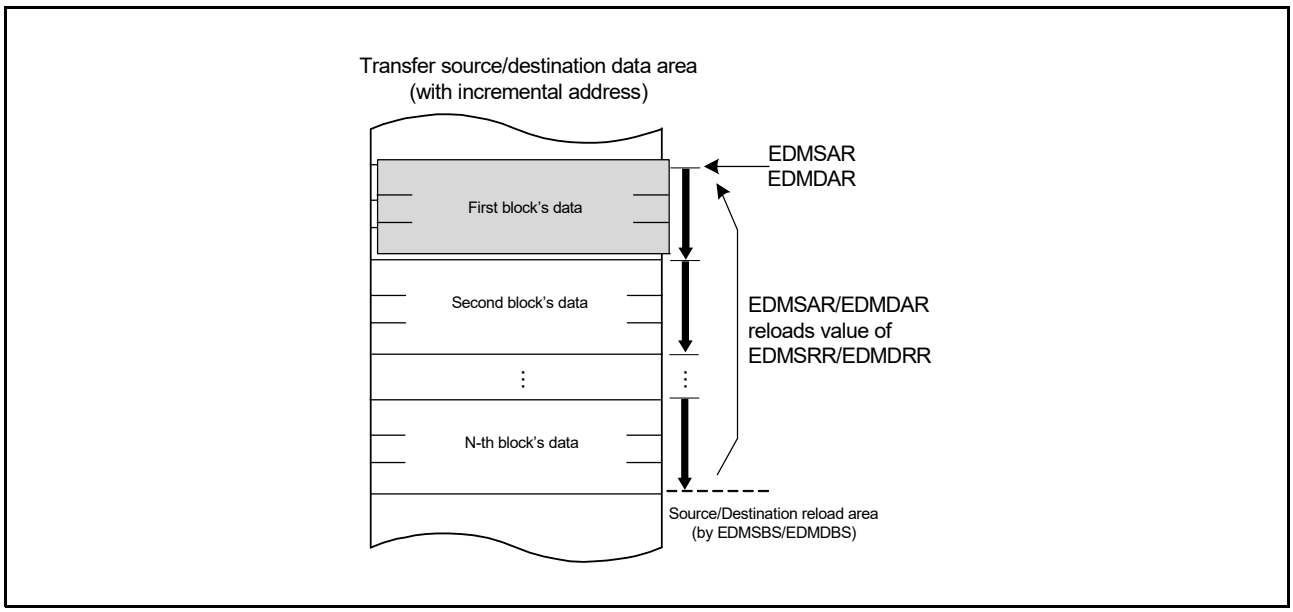


Figure 19.15 Address Update in Incremental Address

(3) Offset Addition Mode

When EDMAMD.SM[1:0] bits is set to 01b, the address update mode of the source is offset addition. And when EDMAMD.DM[1:0] bits is set to 01b, the address update mode of the destination is offset addition. In offset addition, EDMSBS and EDMDBS registers indicates reload area and also works as an access offset value. Unlike other transfer modes, EDMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of EDMSBS and EDMDBS registers is the number of blocks. When the transfer starts, EDMCRAL[15:0] bits operates as a down counter, EDMSAR and EDMDAR registers reloads the value of EDMSRR and EDMDRR registers every time one block is transferred. In addition, EDMSBSL[15:0] and EDMDBSL[15:0] bits, which is the lower 16 bits of EDMSBS and EDMDBS registers, also operates as a down counter and decrements every time one block is transferred. When the EDMSBS and EDMDBS registers value becomes 1, EDMSAR and EDMDAR registers reloads the value of EDMSRR and EDMDRR registers. When EDMAMD.SADR and EDMAMD.DADR bits is set to 0, offset addition operation of the same area is repeated. EDMDAR register only reloads EDMDRR register. Figure 19.16 shows address update in offset addition with EDMAMD.SADR and EDMAMD.DADR = 0.

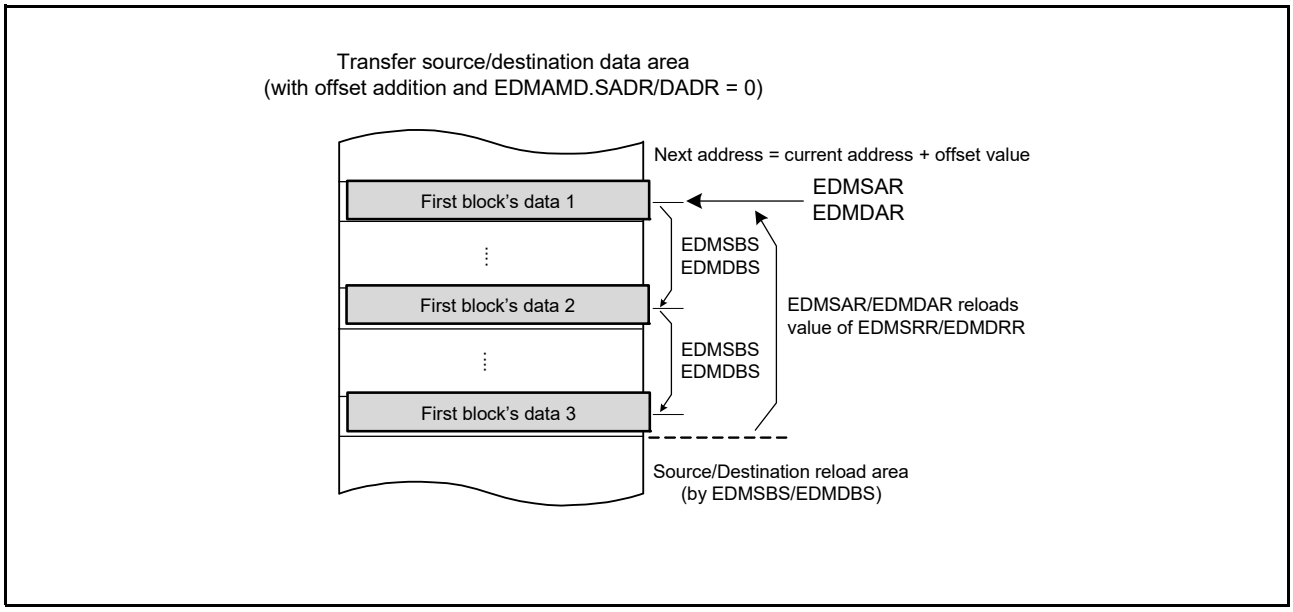


Figure 19.16 Address Update in Offset Addition with EDMAMD.SADR and EDMAMD.DADR = 0

When EDMAMD.SADR and EDMAMD.DADR bits is set to 1, the address is incremented by one data unit after EDMSRR and EDMDRR registers is reloaded by EDMCRAL[15:0] = 1. In other words, an index value ((EDMDBSH – EDMDBSL) × transfer data size) is added to EDMDAR register after EDMDRR register is reloaded. This behavior is used to implement multiple ring buffers. Figure 19.17 shows address update in offset addition with EDMAMD.SADR and EDMAMD.DADR = 1.

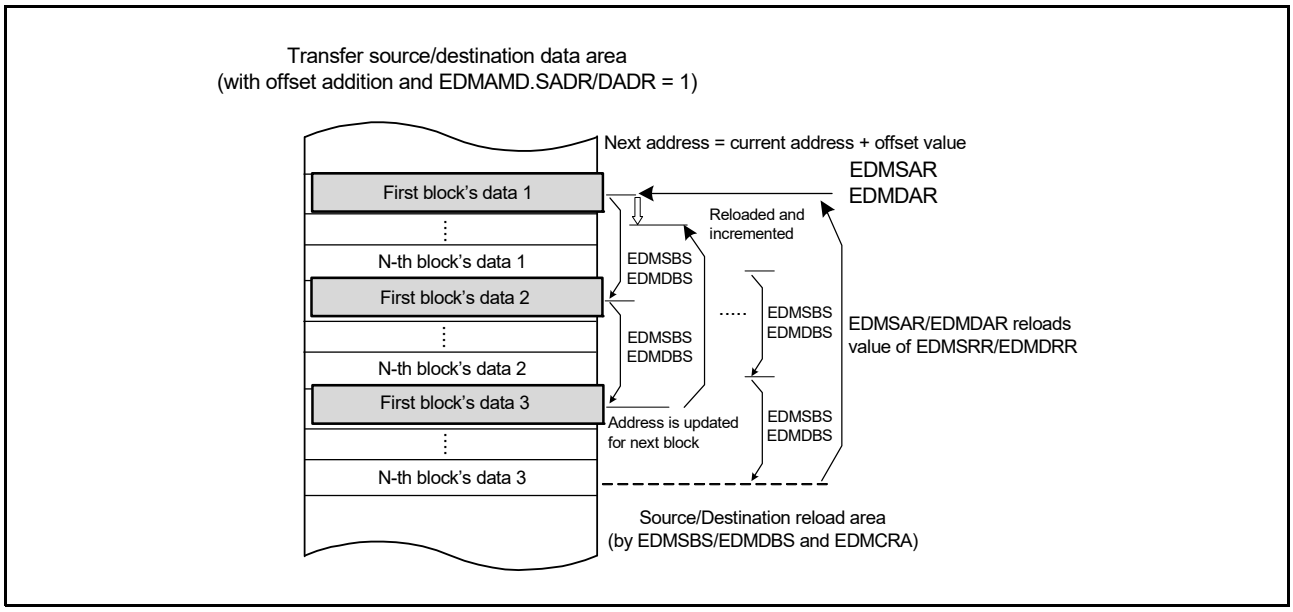


Figure 19.17 Address Update in Offset Addition with EDMAMD.SADR and EDMAMD.DADR = 1

19.3.6 Trigger Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the EXDMAC trigger sources. Setting the EDMTMD.DCTG[1:0] bits selects the trigger source.

(1) Trigger by Software

When DMA transfer is started by software, follow below procedure.

1. Set the EDMTMD.DCTG[1:0] bits to 00b.
2. Set the EDMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the EDMAST.DMST bit set to 1 (EXDMAC activation enabled).
4. Set the EDMREQ.SWREQ bit to 1 (DMA requested).

When the EXDMAC is activated by software while the EDMREQ.CLRS bit is 0, the EDMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request. When the EXDMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by Interrupt Request

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as EXDMAC trigger sources. The trigger sources can be selected individually for each channel in the ICU.SELEXDRn register (n = 0, 1). To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set the EDMCNT.DTE bit to 0 (disable DMA transfer).
2. Set the ICU.SELEXDRn register (n = 0, 1) to the event number, and then clear the corresponding IR flag.
3. Set the EDMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
4. Set EDMPRF.PREQ bit to 1 to clear the PREQ flag to 0.
5. Set the EDMCNT.DTE bit to 1 (enable DMA transfer).
6. Set the EDMAST.DMST bit to 1 (EXDMAC activation enabled).

For interrupt requests specified as EXDMAC trigger sources, see Table 15.3, Interrupt Sources for Software Configurable Interrupt in section 15, Interrupt Controller (ICUM).

19.3.7 Operation Timing

Figure 19.18 and Figure 19.19 show EXDMAC operation timing examples.

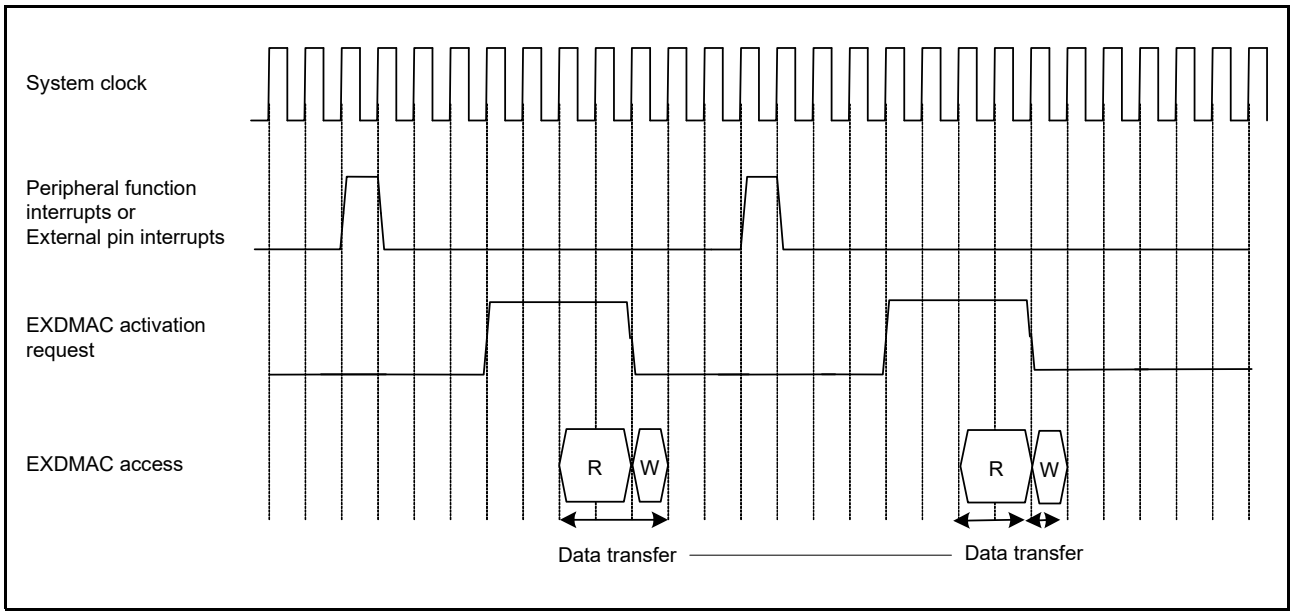


Figure 19.18 EXDMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

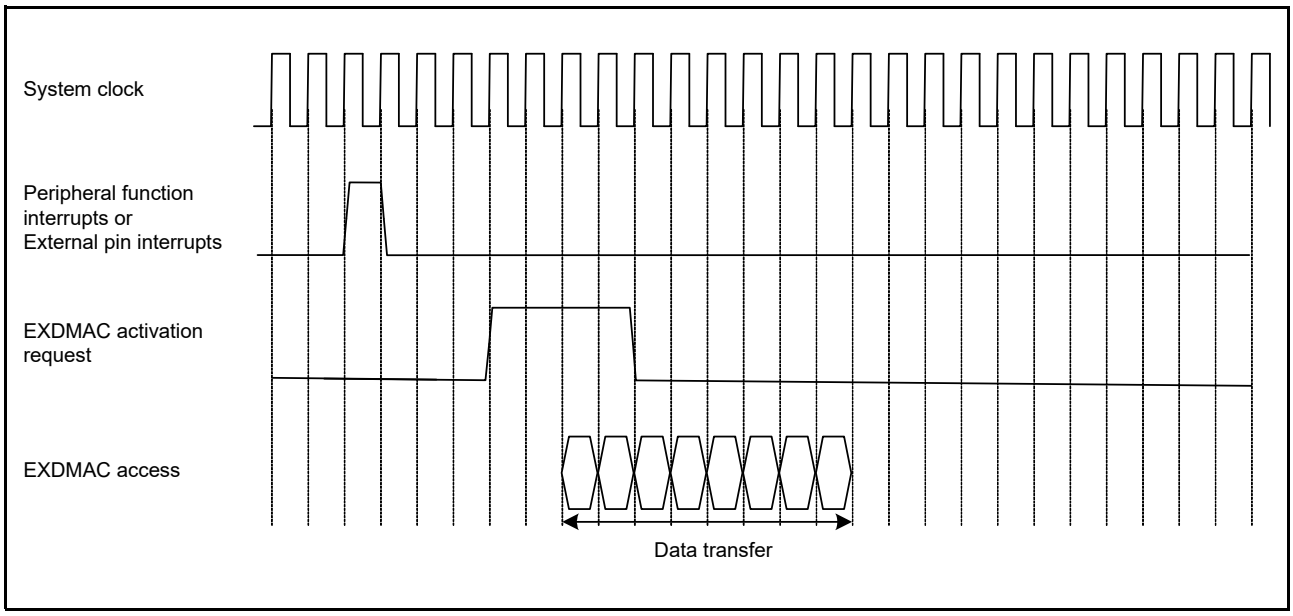


Figure 19.19 EXDMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

19.3.8 Activating the EXDMAC

Table 19.15 lists the register setting procedure of normal, repeat and block transfer mode and Table 19.16 lists register setting procedure of repeat-block transfer mode.

Table 19.15 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the EXDMAC trigger source.	To use peripheral function interrupts as EXDMAC trigger sources. Disable the control register for the peripheral function.
2	Disable the IRQi pin as the EXDMAC trigger source.	To use external pin interrupts as EXDMAC trigger sources.
3	Clear the EDMCNT.DTE bit to 0	Disable DMA transfer.
4	Set the SELEXDRn register of the ICU to specify an interrupt source to trigger operation of the EXDMAC.	Select the use of a peripheral function interrupt or external pin interrupt to trigger operation of the EXDMAC. Enable the interrupt bit for the trigger source. Set the EXDMAC trigger source.
5	Set the peripheral module as a EXDMAC trigger source	To use peripheral function interrupt as a EXDMAC trigger source. Set the control register for the peripheral function.
6	Set the IRQi pin function by using the ICU.	To use external pin interrupt as a EXDMAC trigger source. Set the IRQi pin function by using the ICU.
7	Set the EDMAMD.DM[1:0] bits. Set the EDMAMD.SM[1:0] bits. Set the EDMAMD.DARA[4:0] bits. Set the EDMAMD.SARA[4:0] bits.	Set the destination address update mode select bits. Set the source address update mode select bits. Set the destination address extended repeat area setting bits. Set the source address extended repeat area setting bits.
8	Set the EDMTMD.DCTG[1:0] bits. Set the EDMTMD.SZ[1:0] bits. Set the EDMTMD.DTS[1:0] bits. Set the EDMTMD.MD[1:0] bits. Set the EDMTMD.TKP bit.	Set the transfer request source select bits. Set the data transfer size bits. Set the repeat area select bits. Set the transfer mode select bits. Set the transfer keeping bit.
9	Set the EDMSAR register. Set the EDMDAR register. Set the EDMCRA register.	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
10	Set the EDMCRB register.	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
11	Set the EDMOFR register.	To use the address update function with offset. Set the offset value.
12	Set the EDMINT.DTIE bit to 1.	To use the transfer end interrupts. Enable transfer end interrupts.
13	Set the EDMINT.RPTIE bit. Set the EDMINT.SARIE bit. Set the EDMINT.DARIE bit. Set the EDMINT.ESIE bit to 1.	To use the transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the transfer escape end interrupt.
14	Write 1 to the EDMPRF.PREQ flag.	Clear the RPEQ flag.
15	Set the EDMCNT.DTE bit to 1.	Enable DMA transfer.
16	Set the EDMAST.DMST bit to 1.	Enable EXDMAC operation.*1 Common settings for EXDMAC
17	Start the peripheral function as a EXDMAC trigger source.	To use peripheral function interrupt as a EXDMAC trigger source
18	Enable the IRQi pin as a EXDMAC trigger source.	To use external pin interrupt as a EXDMAC trigger source
19	End of initial settings	For trigger by software On completion of the initial settings, writing 1 to the DMA software start bit (EDMREQ.SWREQ) starts DMA transfer.

n: EXDMAC channel (n = 0, 1)

Note 1. The EDMAST.DMST bit setting does not necessarily have to follow the settings for the individual trigger sources.

Table 19.16 Register Setting Procedure of Repeat-Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the EXDMAC trigger source.	To use peripheral function interrupts as EXDMAC trigger sources. Disable the control register for the peripheral function.
2	Disable the IRQi pin as the EXDMAC trigger source.	To use external pin interrupts as EXDMAC trigger sources.
3	Clear the EDMCNT.DTE bit to 0.	Disable DMA transfer.
4	Set the interrupt source as a EXDMAC trigger source to the SELEXDRn register by using the ICU.	To use internal peripheral interrupts or external pin interrupts as EXDMAC trigger sources. Enable the interrupt bit for the trigger source. Set the EXDMAC trigger source.
5	Set the peripheral module as a EXDMAC request source.	To use peripheral function interrupt as a EXDMAC trigger source. Set the control register for the peripheral function.
6	Set the IRQi pin function by using the ICU.	To use external pin interrupt as a EXDMAC trigger source. Set the IRQi pin function by using the ICU.
7	Set the EDMAMD.DM[1:0] bits. Set the EDMAMD.SM[1:0] bits. Set the EDMAMD.DADR bit. Set the EDMAMD.SADR bit.	Set the destination address update mode select bits. Set the source address update mode select bits. Set the destination address update setting after reload bit. Set the source address update setting after reload bit.
8	Set the EDMTMD.DCTG[1:0] bits. Set the EDMTMD.SZ[1:0] bits. Set the EDMTMD.MD[1:0] bits. Set the EDMTMD.TKP bit.	Set the transfer request source select bits. Set the data transfer size bits. Set the transfer mode to repeat-block transfer mode. Set the transfer keeping bit.
9	Set the EDMSAR register. Set the EDMDAR register. Set the EDMSRR register. Set the EDMDRR register. Set the EDMCRA register. Set the EDMCRB register.	Set the transfer source start address. Set the transfer destination start address. Set the initial value of source start address. Set the initial value of destination start address. Set the number of transfer operations. Set the number of block transfer operations.
10	Set the EDMSBS register. Set the EDMDBS register.	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset. Set the destination buffer size and access offset.
11	Set the EDMINT.DTIE bit to 1.	To use transfer end interrupts. Enable transfer end interrupts.
12	Write 1 to the EDMPRF.PREQ flag.	Clear the PREQ flag.
13	Set the EDMCNT.DTE bit to 1.	Enable DMA transfer.
14	Set the EDMAST.DMST bit to 1.	Enable EXDMAC operation.*1
15	Start the peripheral function as a EXDMAC trigger source.	To use peripheral function interrupt as a DMA trigger source
16	Enable the IRQi pin as a EXDMAC trigger source.	To use external pin interrupt as a DMA trigger source
17	End of initial settings	For trigger by software On completion of the initial settings, writing 1 to the DMA software start bit (EDMREQ.SWREQ) starts DMA transfer.

n: EXDMAC channel (n = 0, 1)

Note 1. The EDMAST.DMST bit setting does not necessarily have to follow the settings for the individual trigger sources.

19.3.9 Starting DMA Transfer

To enable the DMA transfer of EXDMACn (n = 0, 1), set the EDMCNT.DTE bit to 1 (DMA transfer enabled), and then set the EDMAST.DMST bit to 1 (EXDMAC module start). Another transfer request cannot be accepted during the transfer of other EXDMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the EDMSTS.ACT flag is set to 1 (the EXDMAC is in the active state).

19.3.10 Registers during DMA Transfer

The EXDMACn registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMSBS, EDMCNT, and EDMSTS.

(1) EXDMA Source Address Register (EDMSAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(2) EXDMA Destination Address Register (EDMDAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(3) EXDMA Transfer Count Register (EDMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(4) EXDMA Block Transfer Count Register (EDMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(5) EXDMA Source Buffer Size Register (EDMSBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(6) EXDMA Destination Buffer Size Register (EDMDBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected. For details on register update operation in each transfer mode, see Table 19.5 to Table 19.7.

(7) EXDMA Transfer Enable Bit (EDMCNT.DTE)

Although the EDMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically set to 0 by the EXDMAC according to the DMA transfer state. The conditions for clearing this bit by the EXDMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding EDMCNT.DTE bit is set to 1 is prohibited (except for the EDMCNT register). In this case, writing must be performed after the bit is set to 0.

(8) EXDMA Active Flag (EDMSTS.ACT)

The EDMSTS.ACT flag indicates whether the EXDMACn is in the idle or active state. This flag is set to 1 when the EXDMACn starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed. Even when DMA transfer is stopped by writing 0 to the EDMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

(9) Transfer End Interrupt Flag (EDMSTS.DTIF)

The EDMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed. When both this flag and the EDMINT.DTIE bit are set to 1, a transfer end interrupt is requested. This flag is set to 1 when the DMA transfer bus cycle is completed and the EDMSTS.ACT flag is set to 0 indicating the DMA transfer end. This flag is automatically set to 0 when the EDMCNT.DTE bit is set to 1 during the interrupt handling.

(10) Transfer Escape End Interrupt Flag (EDMSTS.ESIF)

The EDMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When both this flag and the EDMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested. This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the EDMSTS.ACT flag is set to 0 indicating the DMA transfer end. This flag is automatically set to 0 when the EDMCNT.DTE bit is set to 1 during an interrupt handling. Before sending an interrupt request from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).

19.3.11 Channel Priority

When multiple DMA transfer requests are present, the EXDMAC determines the priority of channels that have DMA transfer requests. When EDMCTL.PR = 0, the channel priority is fixed as channel 0 > channel 1 (channel 0: Highest). When EDMCTL.PR = 1, the EXDMAC determines the priority of channels at round-robin. When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

19.3.12 Channel Security

The security attribute can be set with CPSCU.EDMACCHSAR.SADMACXn bit for each EXDMAC channel.
When the CPSCU.EDMACCHSAR.SADMACXn bit is 0.

- When the corresponding channel transfers, it behaves as a secure master.
- The register of corresponding channel has a secure attribute.

When the CPSCU.EDMACCHSAR.SADMACXn bit is 1.

- When the corresponding channel transfers, it behaves as a non-secure master.
- The register of corresponding channel has a non-secure attribute.

Do not change the CPSCU.EDMACCHSAR bit while DMA transfer.

19.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the EDMCNT.DTE bit and the EDMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

19.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (EDMTMD.MD[1:0] = 00b)

When the value of the EDMCRAL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the EDMCNT.DTE bit is set to 0 and the EDMSTS.DTIF flag is set to 1 at the same time. If the EDMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (EDMTMD.MD[1:0] = 01b)

When the value of the EDMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the EDMCNT.DTE bit is set to 0 and the EDMSTS.DTIF flag is set to 1 at the same time. If the EDMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.
When the EDMTMD.TKP bit is set to 1 (free-running function), the EDMSTS.DTIF flag is set to 1, but the EDMCNT.DTE bit does not become 0.

(3) In Block Transfer Mode (EDMTMD.MD[1:0] = 10b)

When the value of the EDMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the EDMCNT.DTE bit is set to 0 and the EDMSTS.DTIF flag is set to 1 at the same time. If the EDMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.
When the EDMTMD.TKP bit is set to 1 (free-running function), the EDMSTS.DTIF flag is set to 1, but the EDMCNT.DTE bit does not become 0.

(4) In Repeat-Block Transfer Mode (EDMTMD.MD[1:0] = 11b)

When the value of the EDMCRBL[15:0] bits changes from 1 to 0, DMA transfer ends on the corresponding channel, and the EDMCNT.DTE bit is set to 0 and the EDMSTS.DTIF flag is set to 1 at the same time. If the EDMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Before sending an interrupt request from the
When the EDMTMD.TKP bit is set to 1 (free-running function), the EDMSTS.DTIF flag is set to 1, but the EDMCNT.DTE bit does not become 0.
EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).

19.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of each single-repeat size of data is completed while the EDMINT.RPTIE bit is set to 1. When the interrupt is requested, DMA transfer is terminated regardless of the value of the EDMTMD.TKP bit, the EDMCNT.DTE bit is set to 0, and the EDMSTS.ESIF flag is set to 1. If the EDMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the EDMCNT.DTE bit. A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of each single-block size of data is completed. Repeat size end interrupt cannot be requested in repeat-block transfer mode.
Before sending an interrupt request from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM).
For the interrupt outputs, refer to Figure 19.20, Schematic Logic Diagram of Interrupt Outputs.

19.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area address occurs while the source address extended repeat area is specified with the EDMINT.SARIE bit set to 1 or the destination address extended repeat area is specified with the EDMINT.DARIE bit set to 1, an interrupt due to an extended repeat area overflow is requested. When the interrupt is requested, DMA transfer is terminated regardless of the value of the EDMTMD.TKP bit, the EDMCNT.DTE bit is set to 0, and the EDMSTS.ESIF flag is set to 1. If the EDMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Even if an interrupt due to an extended repeat area overflow is requested during a read cycle, the subsequent write cycle proceeds. In block transfer mode, even if an interrupt due to an extended repeat area overflow is requested during a single-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer. An interrupt due to an extended repeat area overflow is not possible in repeat-block transfer mode. Before interrupt requests can be sent from the EXDMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUM). For the interrupt outputs, see Figure 19.20, Schematic Logic Diagram of Interrupt Outputs.

19.5 Processing on DMA Transfer Error

If an access error occurs during DMA transfer, the EXDMAC stops the transfer.*1 The EXDMAC also clears the EDMCNT.DTE bit of the error causing channel. Write back of the DMA transfer setting to each register is not performed. The information of the channel that caused the error is set in EDMECHR register. For details on bus errors, see section 16.8, Bus Error Monitoring Section.

Note 1. If a bufferable write error occurs, however, the EXDMAC does not stop the transfer.

19.6 Interrupts

19.6.1 Transfer End Interrupt

Each EXDMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. An interrupt request occurs after the bus response to the last write in a DMA transfer. The timing of the bus response to a DMA write access depends on the EDMBWR.BWE bit and the transfer destination. For details, refer to section 19.8.2, Access to the Registers during DMA Transfer. In repeat-block transfer mode, do not enable escape transfer end interrupt. Table 19.17 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 19.20 shows the schematic logic diagram of interrupt outputs . Figure 19.21 shows the EXDMAC interrupt handling routine to resume/terminate DMA transfer.

Table 19.17 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	EDMSTS.DTIF	EDMINT.DTIE
Escape transfer end	Repeat size end	EDMINT.RPTIE	EDMSTS.ESIF	EDMINT.ESIE
	Source address extended repeat area overflow	EDMINT.SARIE		
	Destination address extended repeat area overflow	EDMINT.DARIE		

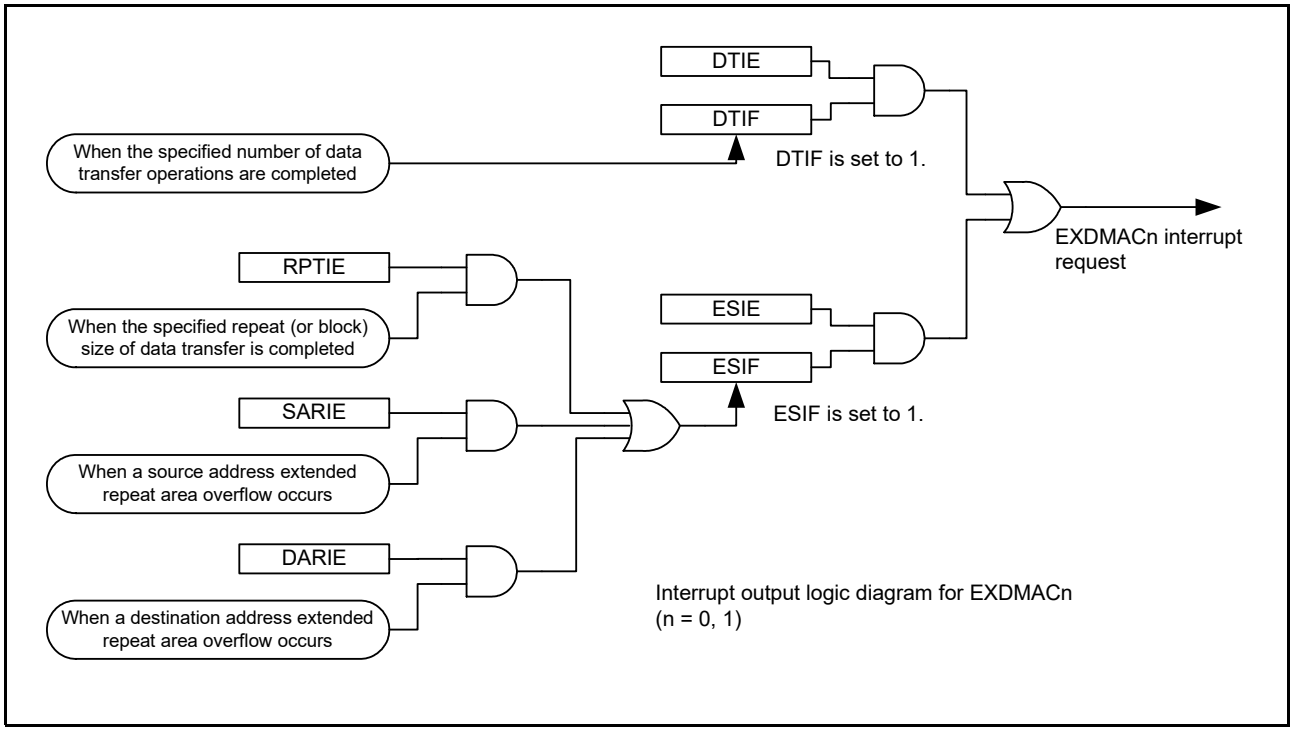


Figure 19.20 Schematic Logic Diagram of Interrupt Outputs

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the EDMSTS.DTIF flag to clear a transfer end interrupt, and to the EDMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The EXDMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the EDMCNT.DTE bit to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the EDMCNT.DTE bit. The EDMSTS.ESIF flag is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.

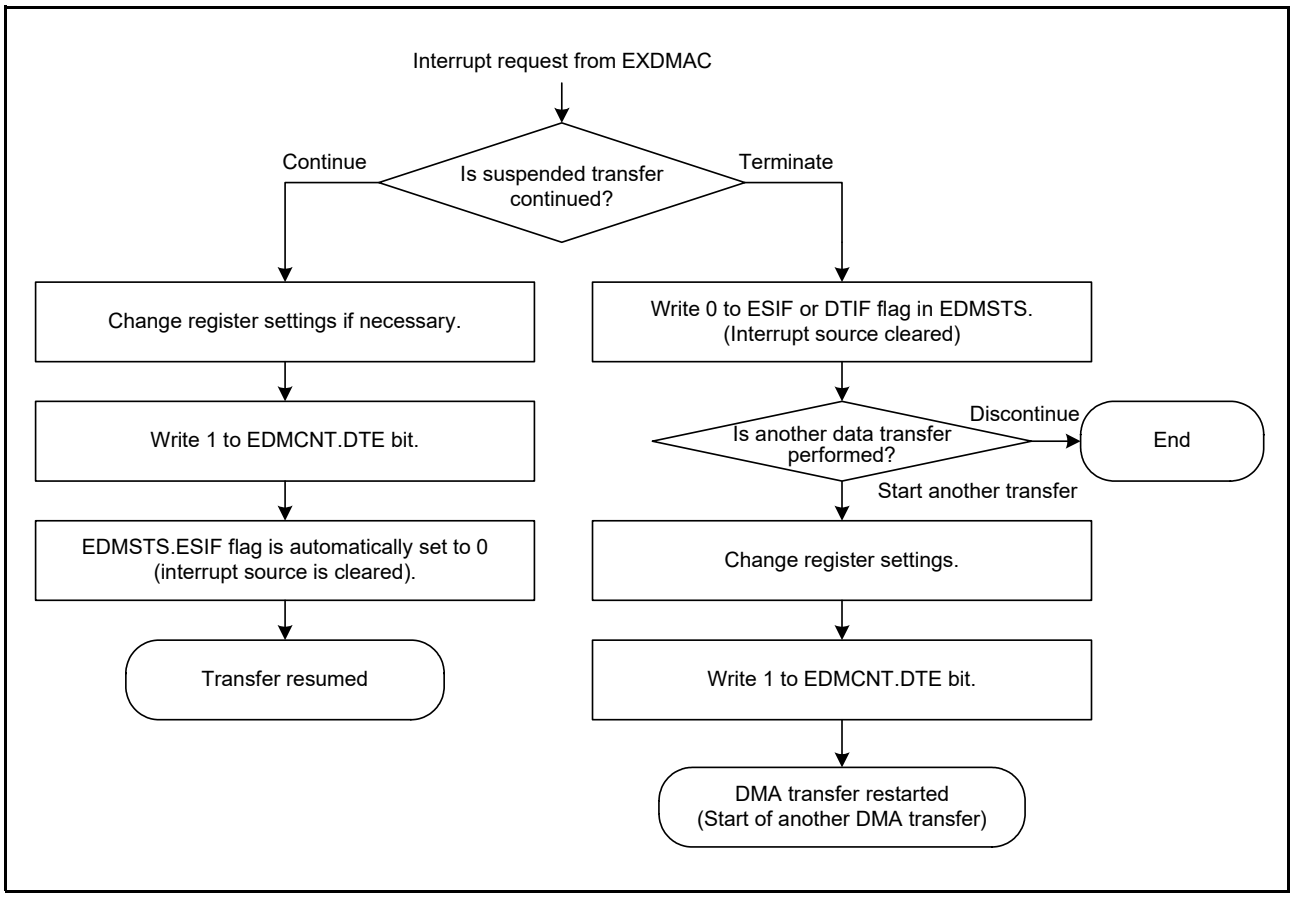


Figure 19.21 EXDMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

19.6.2 Transfer Error Interrupt

A transfer error occurring during DMA transfer is detected as a bus error. For details, see section 16.8, Bus Error Monitoring Section.

Table 19.18 Interrupt and Error Information Due to DMA Transfer Error Cause

Transfer Error Factor	Interrupt/Reset Selection	NMI	Interrupt	Bus Error Status	Error Channel Information
MSAU Error	BUS.MSAOAD.OAD	ICU.NMISR.BUSST	ICU.IR016.IR	BUS.BUSERRSTATEXDMAC.MSERRSTAT	EDMECHR
Illegal Access Error	BUS.BUSOAD.ILERROAD			BUS.BUSERRSTATEXDMAC.ILERRSTAT	
Slave Bus Error	BUS.BUSOAD.SLERROAD			BUS.BUSERRSTATEXDMAC.SLERRSTAT	
Protect Zone Access Error	BUS.BUSOAD.SLERROAD			BUS.BUSERRSTATEXDMAC.SLERRSTAT	

19.7 Low-Power Consumption Function

Before transition to the module-stop state, software standby mode, or deep software standby mode, set the EDMAST.DMST bit to 0 (the EXDMAC suspended), and then perform the following.

(1) Module-Stop Function

Writing 1 to the module-stop setting bit (transition to the module-stop state) enables the module-stop function of the EXDMAC. If DMA transfer is in progress at the time a 1 is written to the module-stop setting bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the module-stop setting bit is 1, accessing the EXDMAC registers are prohibited. Writing 0 to the module-stop setting bit releases the EXDMAC from the module-stop state.

(2) Software Standby and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.2.1, Transitioning to Software Standby Mode or section 11.6.3.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption. If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby or deep software standby follows the completion of DMA transfer.

(3) Note on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.8.6, Timing of the WAIT Instruction in section 11, Low Power Consumption. To perform DMA transfer after returning from low-power consumption mode, set the EDMAST.DMST bit to 1 again.

19.8 Usage Notes

19.8.1 Bus Response to DMA Write Access

The timing of the bus response to a DMA write access depends on the setting of the EDMBWR.BWE bit and the transfer destination. If the EDMBWR.BWE bit is set to 1, the bus response returns from the intermediate point described in section 16.2.5.2, Bufferable Write. Also, for writes to the destinations described in section 16.2.5.3, Notes on the Timing of Writing Completion, the bus response returns before the write is complete.

19.8.2 Access to the Registers during DMA Transfer

Do not write to the following registers while the EDMSTS.ACT flag of the same channel is set to 1 (EXDMAC active state) or the EDMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled): EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMTMD, EDMINT, EDMAMD, EDMOFR, EDMREQ, EDMSBS, EDMDBS, EDMSRR, EDMDRR, EDMBWR, EDMCTL, EDMBAC, and EDMPRF.

19.8.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

20. Data Transfer Controller (DTCA)

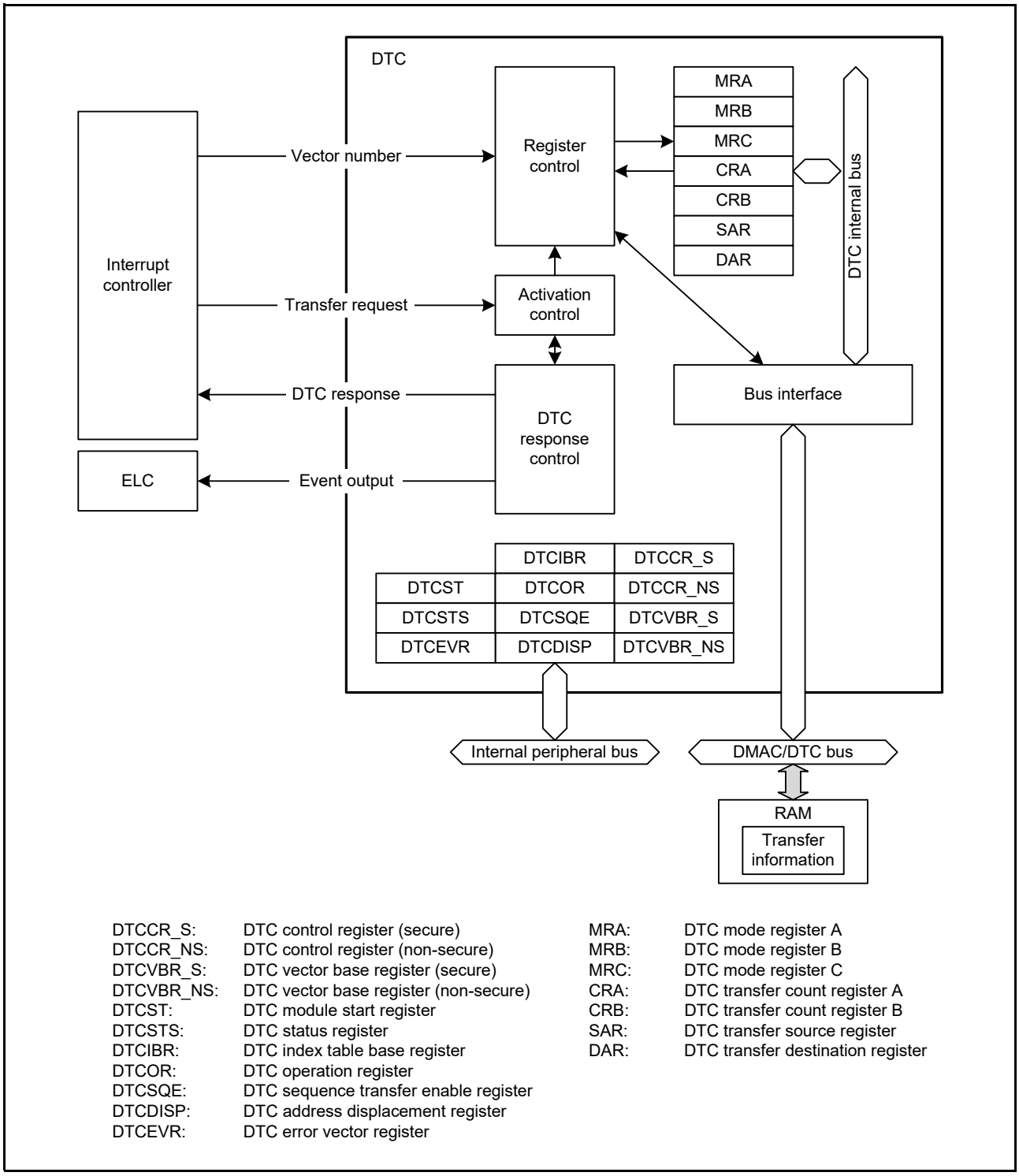
This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.
In addition to the conventional methods of DTC transfer (normal, repeat, block, and chain), DTCA supports sequential transfer, in which it handles a series of transfers made up of a combination of the other methods. In sequential transfer, the data that is initially transferred selects one from possible 256 sequences for execution. The DTCA can divide one sequence into several transfers depending on how the parts of the sequence are combined.

20.1 Overview

Table 20.1 lists the specifications of the DTC, and Figure 20.1 shows a block diagram of the DTC.

Table 20.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 64 bits, 2048 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 64 bits = 2048 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Sequence transfer	A series of complicated transfers can be registered as a sequence. Any sequence can be selected by the transfer data and executed. • Only one trigger source can be set at a time. • Up to 256 sequences for a single trigger source • The data that is initially transferred in response to a transfer request determines a sequence • The whole sequence can be executed on a single request, or be suspended in the middle of the sequence and resumed on the next transfer request (division of sequence).
Transfer space	• 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits), 1 double longword (64 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Transfer error handling	If a transfer error (bus error) occurs, the transfer is aborted and the IR flag of the transfer request source is cleared.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Write-back disable	Allows disabling the write-back of transfer information.
Displacement addition	The displacement value can be added to the transfer source address (for each transfer information)
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attributes can be set.



DTCCR_S: DTC control register (secure)
DTCCR_NS: DTC control register (non-secure)
DTCVBR_S: DTC vector base register (secure)
DTCVBR_NS: DTC vector base register (non-secure)
DTCST: DTC module start register
DTCSTS: DTC status register
DTCIBR: DTC index table base register
DTCOR: DTC operation register
DTCSQE: DTC sequence transfer enable register
DTCDISP: DTC address displacement register
DTCEVR: DTC error vector register

MRA: DTC mode register A
MRB: DTC mode register B
MRC: DTC mode register C
CRA: DTC transfer count register A
CRB: DTC transfer count register B
SAR: DTC transfer source register
DAR: DTC transfer destination register

Figure 20.1 DTC Block Diagram

20.2 Register Descriptions

Registers MRA, MRB, MRC, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

20.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	WBDIS
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	WBDIS	Write-Back Disable	0: Writes back the transfer information on completion of the data transfer 1: Does not write back the transfer information on completion of the data transfer	—
b1	—	Reserved	Set this bit to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b, +8 when 11b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b, –8 when 11b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Double longword (64-bit) transfer	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

WBDIS Bit (Write-Back Disable)

The WBDIS bit selects whether to write back the transfer information.

When the bit is 0, updated transfer information is written back.

When the bit is 1, updated transfer information is not written back even with the setting of that address is incremented after a transfer, and the same data transfer is executed every time for each transfer request. The transfer information can be stored in ROM because the transfer information is not written back.

While the WBDIS bit is 1, operation for each transfer mode is as follows:

(1) Normal transfer and repeat transfer modes
1 byte, 1 word, 1 longword, or 1 double longword of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same transfer is repeated on each transfer request. When the transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

(2) Block transfer mode
1-block of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same block transfer is repeated on each transfer request. When the block transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

When setting the MRC.DISPE bit to 1, set the WBDIS bit to 1 (does not write back the transfer information). If the value of the WBDIS bit in any transfer information is 1, set the DTCCR.RRS bit to 0 (so that reading of the transfer information is not skipped).

20.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		INDX	SQEND
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SQEND	Sequence Transfer End	0: Continue the sequence transfer 1: End the sequence transfer	—
b1	INDX	Index Table Reference	0: Does not refer to the index table 1: Refers the index table based on the transferred data*1	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b, +8 when 11b) 1 1: The DAR value is decremented after data transfer. (–1 when the MRA.SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b, –8 when 11b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

Note 1. Set the MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

SQEND Bit (Sequence Transfer End)

The SQEND bit selects whether to continue or end sequence transfer. Refer to Table 20.2 for details.
This bit can only be set to 1 for transfer information referred to by the DTC index table. Set this bit to 0 for transfer information referred to by the DTC vector table.

INDX Bit (Index Table Reference)

When the value of the INDX bit in transfer information that is read is 1, a sequence transfer proceeds. Refer to Table 20.2 for details.
Set this bit to 0 for transfer information which is not associated with sequence transfer or is not intended to start sequence transfer. Do not allow transfer requests to be generated by the sources different from that specified in the DTCSQE register but having the INDX bit set to 1.

Table 20.2 Values of Bits CHNE, SQEND, and INDX in the Sequence Transfer and DTC Operation

CHNE Bit	SQEND Bit	INDX Bit	Operation	Usage
0	0	1	Start sequence transfer	Use this setting for the transfer information that is first read in response to a transfer request from the source specified in the DTCSQE register.
1	0	0	Continue sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	0	0	Suspend sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	1	0	End sequence transfer	Use this setting with the last transfer information in a sequence.
0	1	1	End current sequence transfer and start new sequence transfer	Use this setting with the last transfer information in a sequence.

Note: Do not set the values other than listed above.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 20.4, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 20.4.6, Chain Transfer.

Refer to Table 20.2 for the setting value to be used in the sequence transfer.

20.2.3 DTC Mode Register C (MRC)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISPE
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	DISPE	Displacement Addition	0: The displacement value is not added to the transfer source address. 1: The displacement value is added to the transfer source address.	—
b7 to b1	—	Reserved	Set these bits to 0.	—

The MRC register is used to select DTC operating mode and cannot be accessed directly from the CPU.

DISPE Bit (Displacement Addition)

This bit specifies whether to use the SAR + DTCDISP value as the transfer source address.
When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information) and set the DTCCR.RRS bit to 0 (transfer information read is not skipped).

20.2.4 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

SAR register is used to set the transfer source start address.
SAR register cannot be accessed directly from the CPU.

20.2.5 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

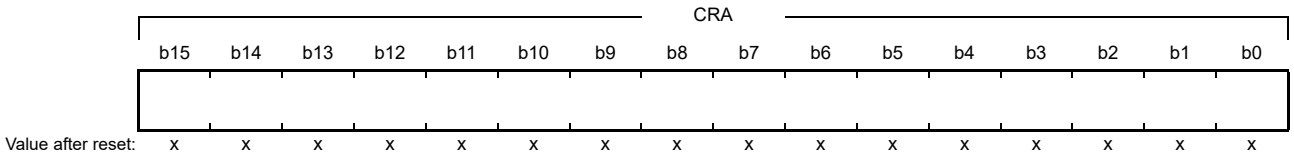
x: Undefined

DAR register is used to set the transfer destination start address.
DAR register cannot be accessed directly from the CPU.

20.2.6 DTC Transfer Count Register A (CRA)

- Normal transfer mode

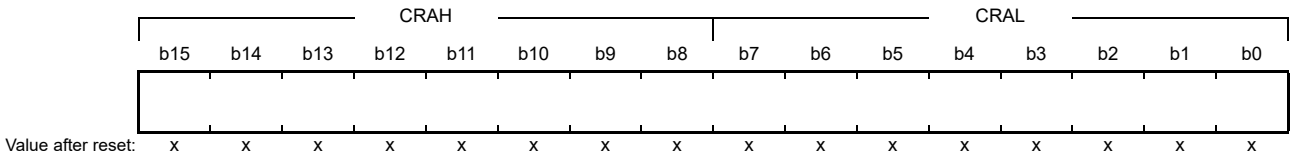
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

20.2.7 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

20.2.8 DTC Control Register for Secure Application (DTCCR_S), DTC Control Register for Non-secure Application (DTCCR_NS)

Address(es): DTC.DTCCR_S 8700 7C10h*1, DTC_NS.DTCCR_NS 0700 7C00h*2

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable*3	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 2. This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 3. Set this bit to 0 when using the sequence transfer.

DTCCR register is used to control the DTC operation. There are two registers: one for secure application developers (DTCCR_S) and one for non-secure application developers (DTCCR_NS).

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.
If the value of the MRA.WBDIS bit in any transfer information is 1, set the RRS bit to 0. Note that the MRA.WBDIS bit

should be set to 1 when the MRC.DISPE bit is set to 1.
Like chain transfer, sequence transfer handles sequences of multiple types of data transfer. When sequence transfer is to be used, set the RRS bit to 0 so that the previous data transfer will not be repeated.

20.2.9 DTC Vector Base Register for Secure Application (DTCVBR_S), DTC Vector Base Register for Non-secure Application (DTCVBR_NS)

Address(es): DTC.DTCVBR_S 8700 7C14h*1, DTC_NS.DTCVBR_NS 0700 7C04h*2

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1. This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 2. This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated.
The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
There are two registers: one for Secure application developers (DTCVBR_S) and one for Non-secure application developers (DTCVBR_NS).
The DTCVBR_NS register can be set in the range of 0000 0000h to 7FFF FC00h, and the DTCVBR_S register in the range of 8000 0000h to FFFF FC00h in 1-Kbyte units.

20.2.10 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 8700 7C0Ch, DTC_NS.DTCST 0700 7C0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, software standby mode, or deep software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state or software standby mode.

For details on transitions to the module stop state, software standby mode, and deep software standby mode, refer to section 20.9, Low Power Consumption Function, and section 11, Low Power Consumption.

20.2.11 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 8700 7C0Eh, DTC_NS.DTCSTS 0700 7C0Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.6, Interrupt Vector Table in section 15, Interrupt Controller (ICUM).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.
- When the sequence transfer is resumed.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.
- When the sequence transfer is suspended.

20.2.12 DTC Index Table Base Register (DTCIBR)

Address(es): DTC.DTCIBR 8700 7C24h, DTC_NS.DTCIBR 0700 7C24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCIBR register is used to set the base address for calculating the address to which the DTC index is allocated.
Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27.
The lower 10 bits (b9 to b0) are reserved bits and fixed to 0. When writing this register, set these bits to 0.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

20.2.13 DTC Operation Register (DTCOR)

Address(es): DTC.DTCOR 8700 7C28h, DTC_NS.DTCOR 0700 7C28h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SQTFRL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SQTFRL	Sequence Transfer Terminate	Writing 1 to this bit terminates the sequence transfer in progress. This bit is read as 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCOR register sets the operation of the DTC module.

SQTFRL Bit (Sequence Transfer Terminate)

Setting the SQTFRL bit to 1 terminates the sequence transfer in progress.
When the DTCSQE.ESPSEL bit is 1 (Sequence transfer is enabled), follow the procedure shown in Figure 20.2 to terminate the sequence transfer.
Writing 1 to the bit, while no sequence transfer is performed, have no effect.

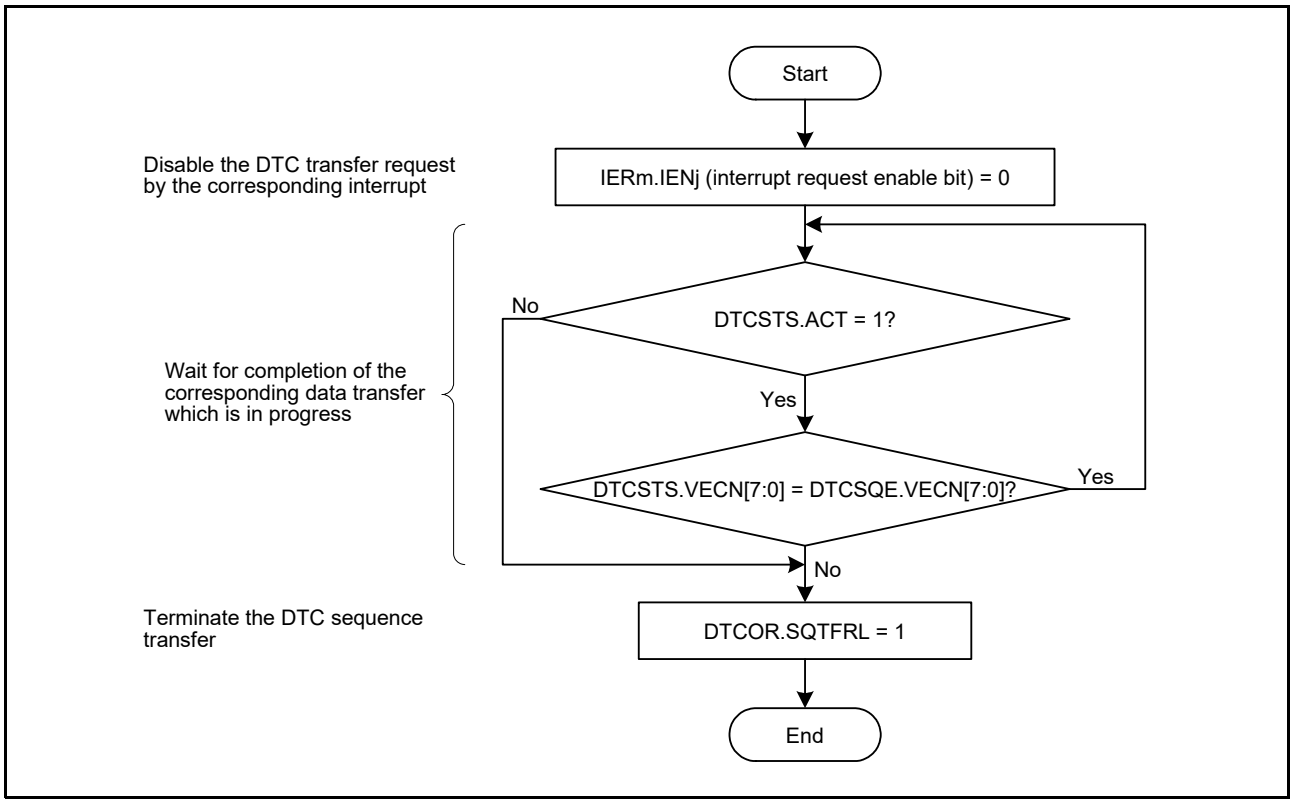


Figure 20.2 Procedure to Terminate Sequence Transfer

20.2.14 DTC Sequence Transfer Enable Register (DTCSQE)

Address(es): DTC.DTCSQE 8700 7C2Ch, DTC_NS.DTCSQE 0700 7C2Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESPSEL	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	Sequence Transfer Vector Number Setting	Specify the vector number by which a sequence transfer is enabled. The value is only valid when the ESPSEL bit is 1.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	ESPSEL	Sequence Transfer Enable	0: Sequence transfer is disabled. 1: Sequence transfer is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCSQE register is used to specify sequence transfer. Follow Figure 20.23 for details on the setting procedure.

VECN[7:0] Bit (Sequence Transfer Vector Number Setting)

This bit is used to specify for which vector number to perform sequence transfer. Sequence transfer can occur only for this trigger source.
Table 15.6, Interrupt Vector Table in section 15, Interrupt Controller (ICUM) shows the relationship between the trigger source and the vector number.

ESPSEL Bit (Sequence Transfer Enable)

The ESPSEL bit specifies whether sequence transfer is used.

20.2.15 DTC Address Displacement Register (DTCDISP)

Address(es): DTC.DTCDISP 8700 7C18h, DTC_NS.DTCDISP 0700 7C18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The DTCDISP register is used to specify the displacement value to add to the DTC transfer source address.

If MRC.DISPE bit is 1, the value SAR + DTCDISP is used as the transfer source address.

20.2.16 DTC Error Vector Register (DTCEVR)

Address(es): DTC.DTCEVR 8700 7C20h, DTC_NS.DTCEVR 0700 7C20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERRSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WBCF	ERRTI[2:0]			—	—	—	ERRSAF	ERRVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ERRVCT[7:0]	Error Occurrence Vector Number Indication	These bits indicate the vector number for the DTC trigger source when a transfer error occurs.	R*1
b8	ERRSAF	Error Occurrence Vector Security Attribute	0: Secure 1: Non-secure	R*1
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	ERRTI[2:0]	Error Occurrence Timing Indication	b14 b12 0 0 0: When reading the DTC vector 0 0 1: When reading the transfer information 0 1 0: When reading data from the source address 0 1 1: When writing data to the destination address 1 0 0: When writing back the transfer information 1 0 1: When referring the DTC index table Other than above: Reserved	R
b15	WBCF	Write-Back Completion Flag	0: Write-back of transfer information has not been completed. 1: Write-back of transfer information has been completed.	R*1
b16	ERRSTS	Error Status Flag	0: No transfer error 1: Transfer error occurred	R/W*2
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register can be read by both secure and non-secure transactions.
When the ERRSAF flag is set to 0 (secure), this register can only be written by secure transactions. Writes by non-secure transactions are ignored, and a Protect Zone access error is generated.
When the ERRSAF flag is set to 1 (non-secure), this register can be written by any transaction.
Note 1. Clearing the ERRSTS flag also clears these flags and bits.
Note 2. Writing 1 to this flag clears it.

The DTCEVR register indicates detailed information when a transfer error occurs. The information in this register is updated only when the ERRSTS flag changes from 0 to 1. It is not updated unless the ERRSTS flag is cleared.

ERRSAF Flag (Error Occurrence Vector Security Attribute)

This flag indicates the security attribute of the DTC trigger source that caused the transfer error. The security attribute of the DTC trigger source is the same as the security attribute of the ICU.DTCERn register.

WBCF Flag (Write-Back Completion Flag)

This flag indicates whether write-back of the transfer information has been completed when a transfer error occurs.

ERRSTS Flag (Error Status Flag)

This flag indicates that a bus error has occurred during DTC transfer.

[Setting condition]

- When a transfer error occurs when this flag is 0.

[Clearing condition]

- When 1 is written to this flag.

20.2.17 DTC Security Attribution Register (DTCSAR)

Address(es): CPSCU.DTCSAR 8701 8030h, CPSCU_NS.DTCSAR 0701 8030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SADTC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SADTC0	DTC Security Attribution 0	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the DTC module.

SADTC0 Bit (DTC Security Attribution 0)

This bit is used to set the security attributes of the DTC registers (DTCST, DTCSTS, DTCIBR, DTCOR, DTCSQE, and DTCDISP), except for the DTCCR, DTCVBR, and DTCEVR registers*1.

Note 1. The DTCCR and DTCVBR registers have two sets of registers, one for secure applications and one for non-secure applications, and their security attributes are fixed. The security attribute of the DTCEVR register is special.

20.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC. The security attribute of the DTC request source is the same as the security attribute of the ICU.DTCERn register and is set by the CPSCU.ICU0SAR0 to ICU0SAR7 registers.
For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.6, Interrupt Vector Table in section 15, Interrupt Controller (ICUM). For request by software, refer to section 15.2.5, Software Interrupt Generation Register (SWINTR) and section 15.2.6, Software Interrupt 2 Generation Register (SWINT2R) in section 15, Interrupt Controller (ICUM).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DMAC/DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

20.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address. There is one vector table each in the Secure world and Non-secure world, and the table that matches the security attribute of the request source is used.
The vector table should be located on a 1-Kbyte boundary so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register for Secure application (DTCVBR_S) to set the base address of the DTC vector table in the Secure world, and use the DTC vector base register for Non-secure application (DTCVBR_NS) to set the base address of the DTC vector table in the Non-secure world.
Transfer information is allocated in the RAM area. Transfer information can be allocated in the ROM area when the MRA.WBDIS bit is set to 1. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 16-byte boundary.
Figure 20.3 shows the relationship between the DTC vector table and transfer information.
Figure 20.4 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 20.10.2, Allocating Transfer Information.

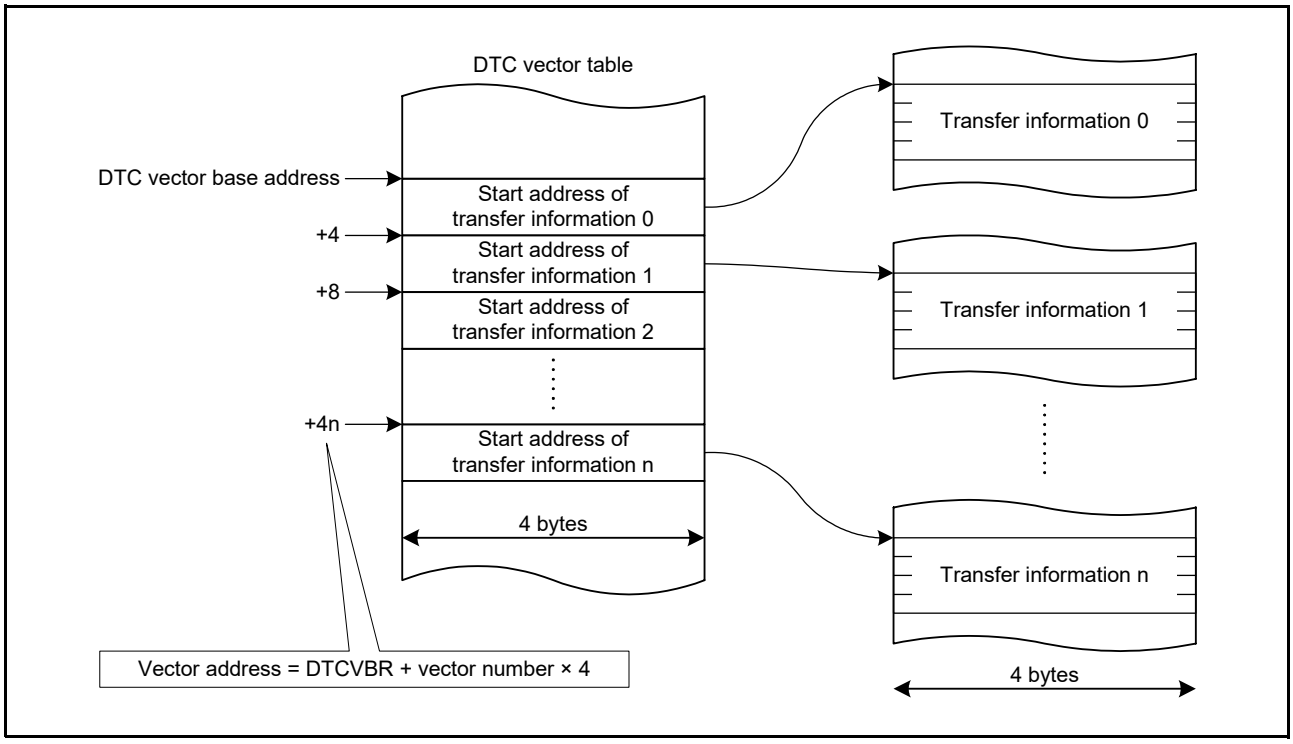


Figure 20.3 DTC Vector Table and Transfer Information

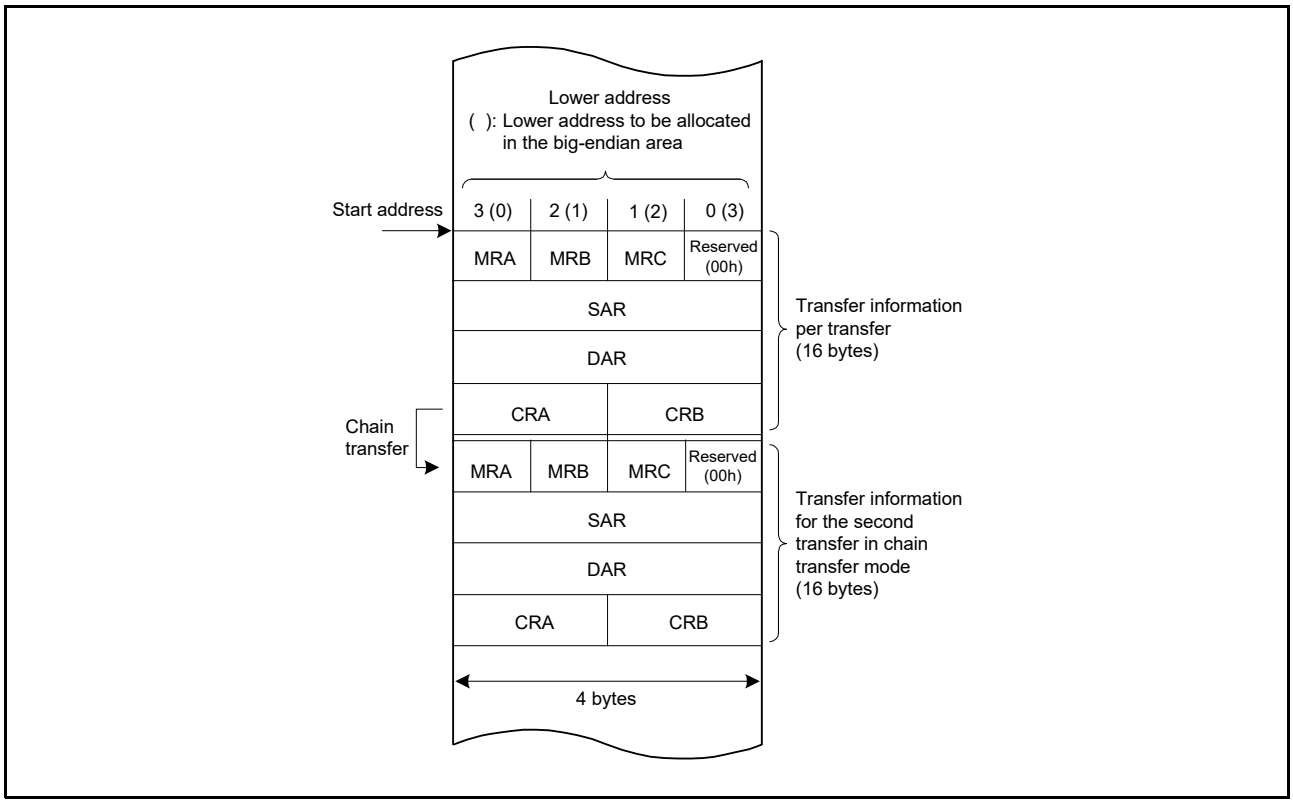


Figure 20.4 Allocation of Transfer Information in the RAM Area

20.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 20.3 lists transfer modes of the DTC.

Table 20.3 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword/1 double longword	Incremented/decremented by 1, 2, 4, or 8 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword/1 double longword	Incremented/decremented by 1, 2, 4, or 8 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords/1 to 256 double longwords)	Incremented/decremented by 1, 2, 4, or 8 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed.

Figure 20.5 shows the operation flowchart of the DTC. Table 20.4 lists chain transfer conditions.

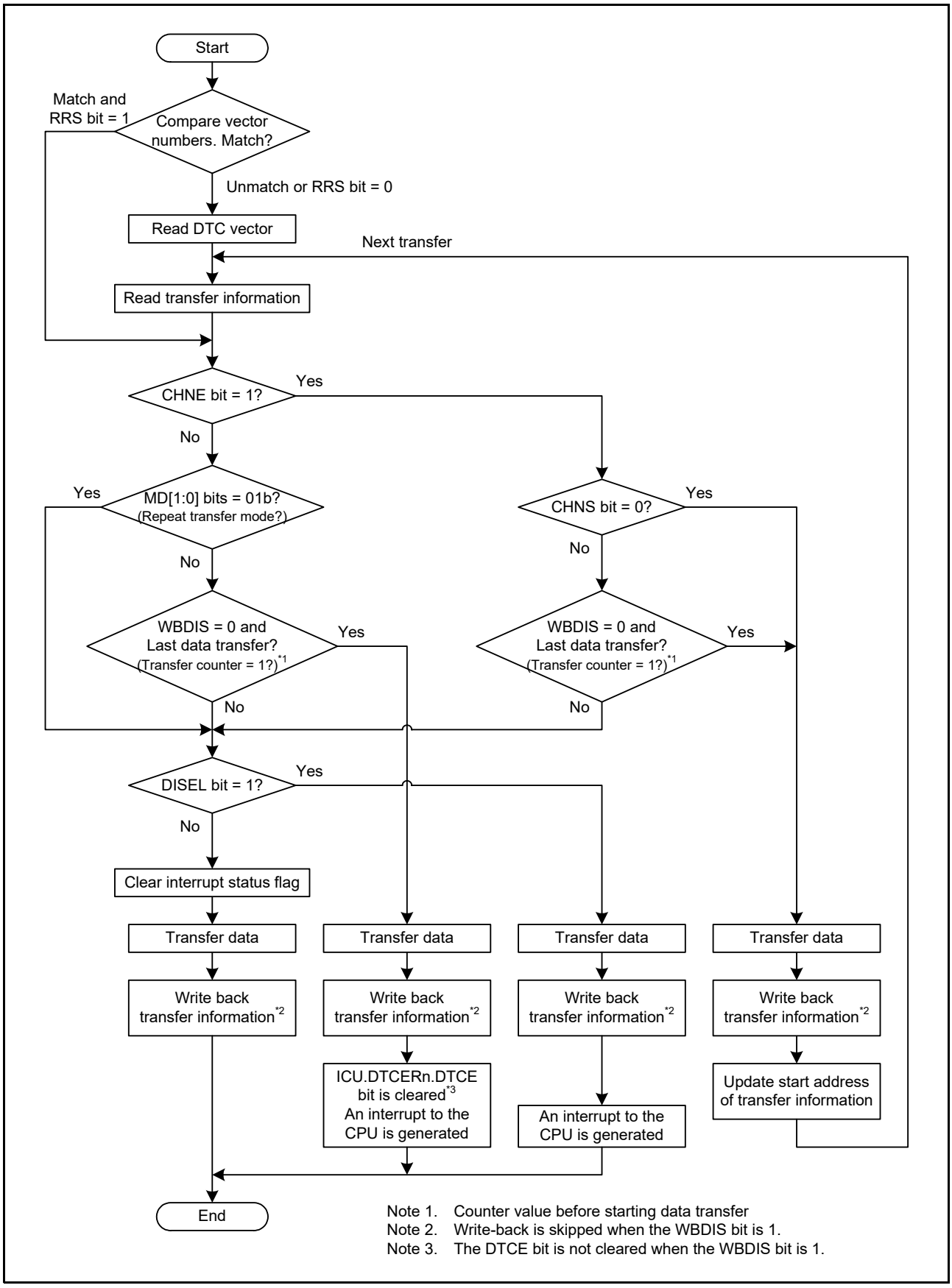


Figure 20.5 Operation Flowchart of the DTC

Table 20.4 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

20.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 20.13 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

20.4.2 Transfer Information Write-Back Skip Function

20.4.2.1 Write-Back Skip by Fixing Addresses

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back.

Table 20.5 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back, but write-back of registers MRA, MRB, and MRC is skipped.

Table 20.5 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

20.4.2.2 Write-Back Skip by the MRA.WBDIS Bit

When the MRA.WBDIS bit is 1, the transfer information (SAR, DAR, CRA, and CRB) is not written back regardless of the settings of the transfer information.

The transfer information on the memory is not updated, data can be transferred by the DTC without copying the transfer information from ROM to RAM. Skipping a write-back reduces time for post-processing of the data transfer.

20.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, 1-longword, or 1-double-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 20.6 lists register functions in normal transfer mode, and Figure 20.6 shows the memory map of normal transfer mode.

Table 20.6 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information*1
SAR	Transfer source address	Increment/decrement/fixed*2
DAR	Transfer destination address	Increment/decrement/fixed*2
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

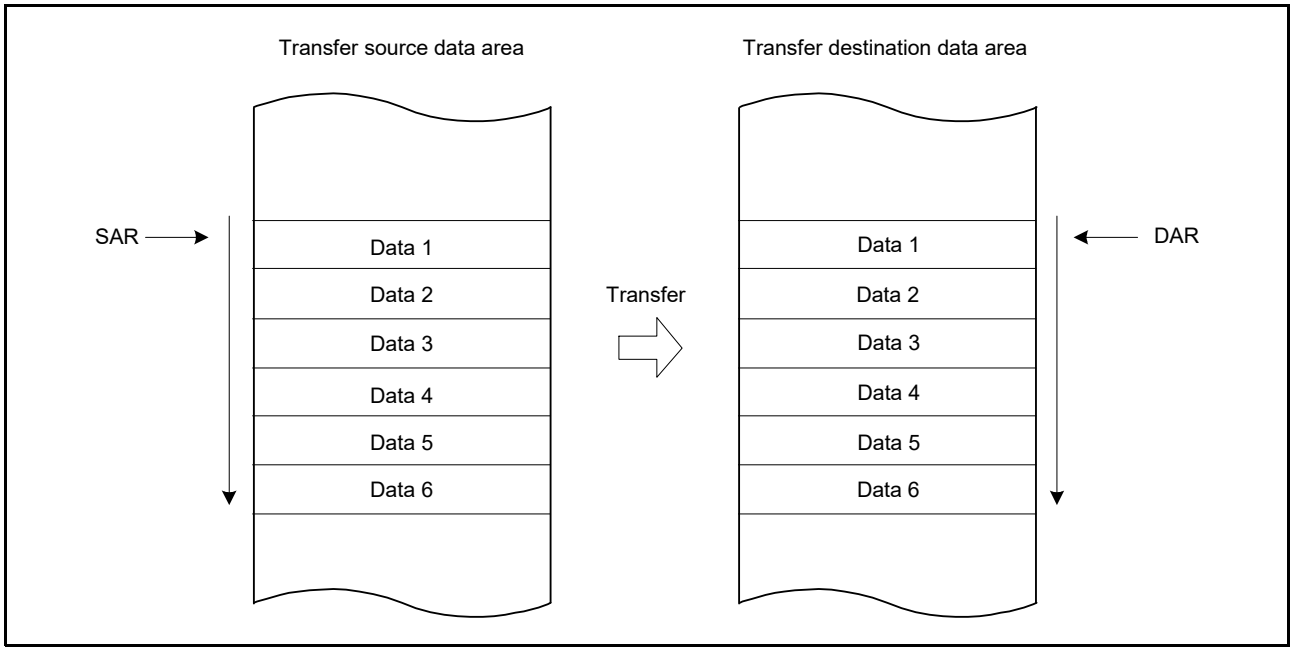


Figure 20.6 Memory Map of Normal Transfer Mode

20.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, 1-longword, or 1-double-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 20.7 lists the register functions in repeat transfer mode, and Figure 20.7 shows the memory map of repeat transfer mode.

Table 20.7 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information*1		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS Bit is 0**	**When the MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*2	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

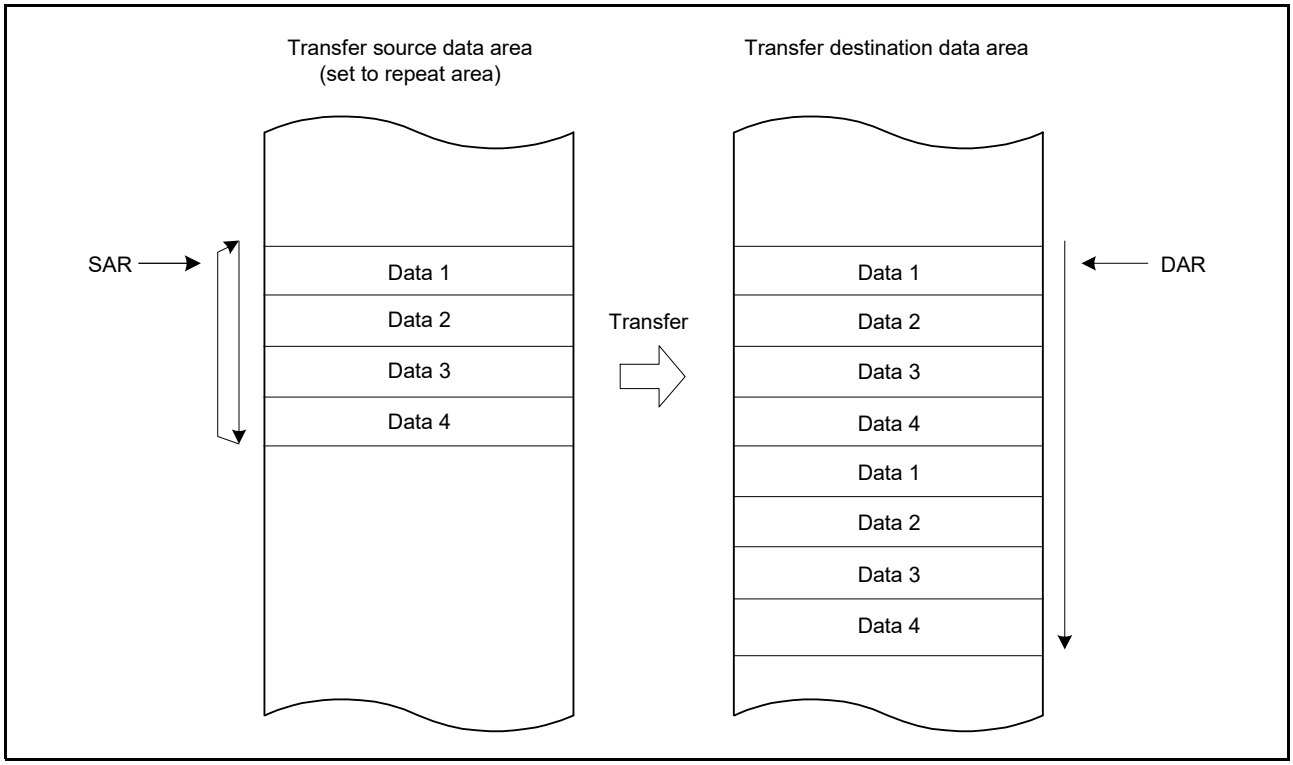


Figure 20.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

20.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, 1 to 256 longwords, or 1 to 256 double longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 20.8 lists register functions in block transfer mode, and Figure 20.8 shows the memory map of block transfer mode.

Table 20.8 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information*1	
Register	**Description**	**When MRB.DTS Bit is 0**	**When MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

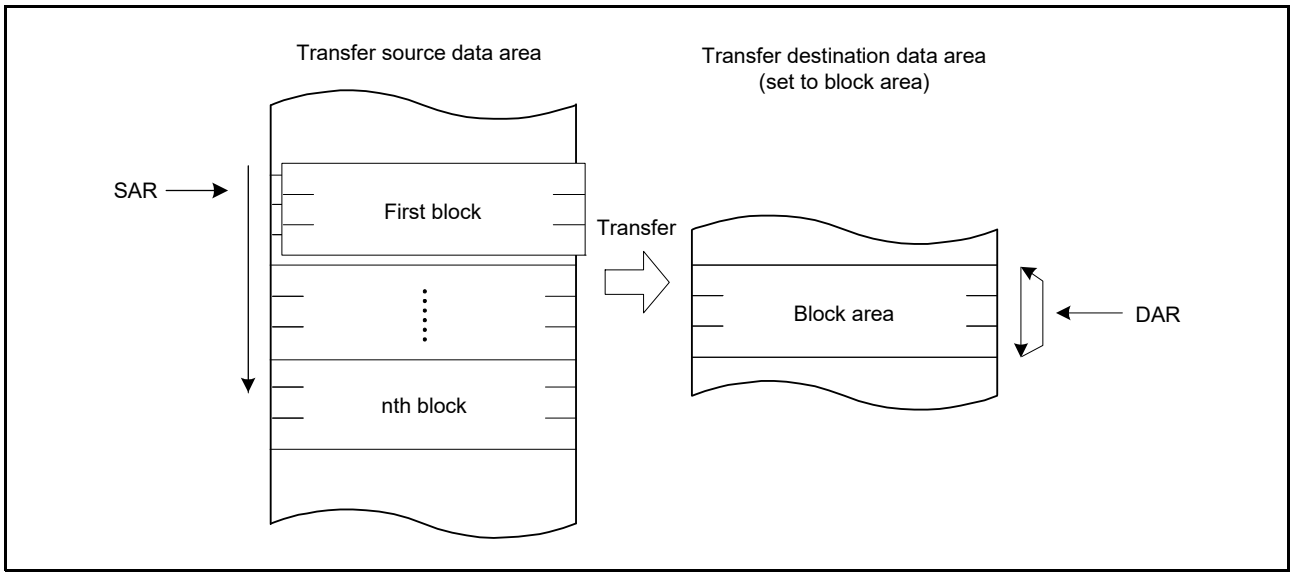


Figure 20.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

20.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request. If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source. The transfer information (SAR, DAR, CRA, CRB, MRA, MRB, and MRC) that define a data transfer can be specified independently of each other. Figure 20.9 shows chain transfer operation.

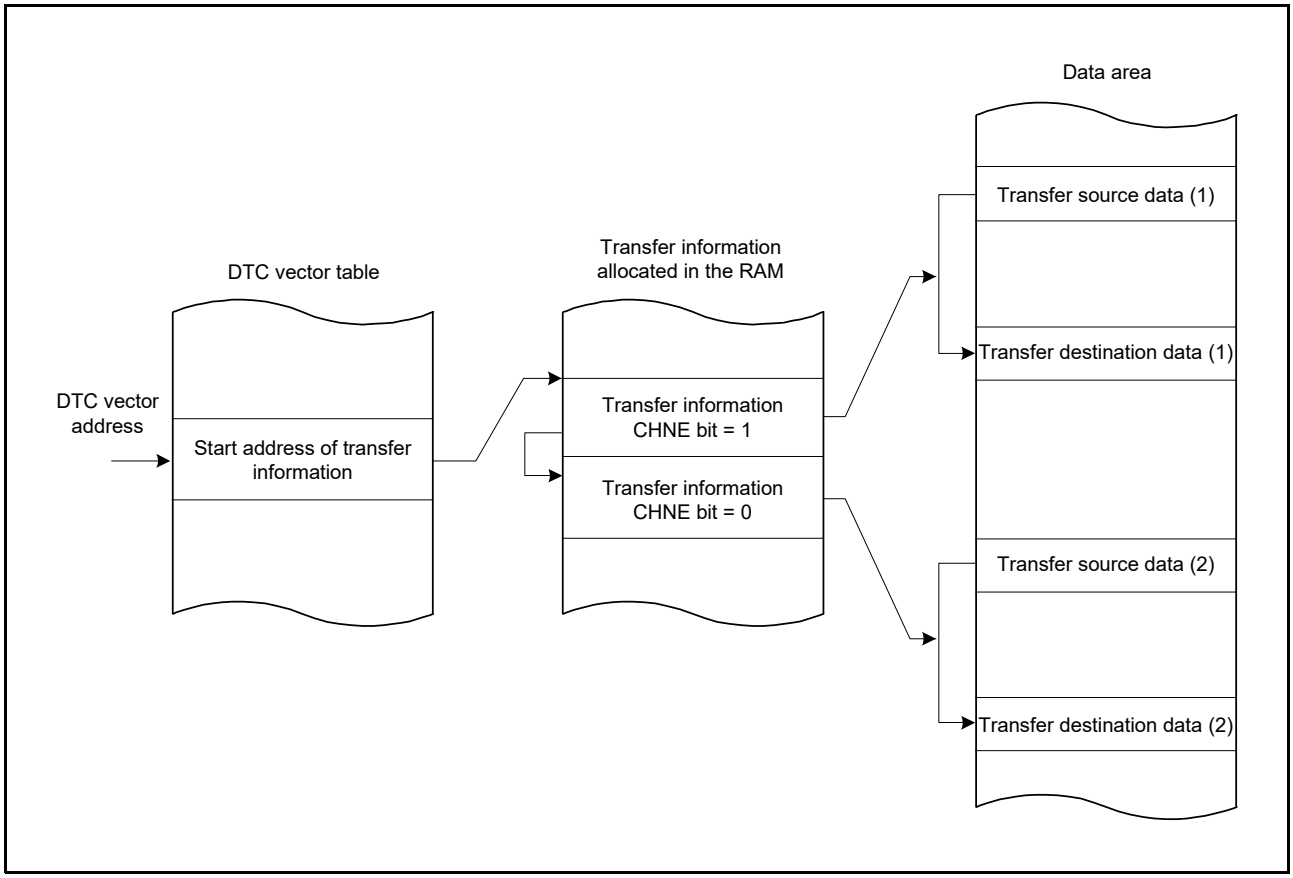


Figure 20.9 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.
For details on chain transfer conditions, refer to Table 20.4, Chain Transfer Conditions.

20.4.7 Operation Timing

Figure 20.10 to Figure 20.13 show examples of DTC operation timing.

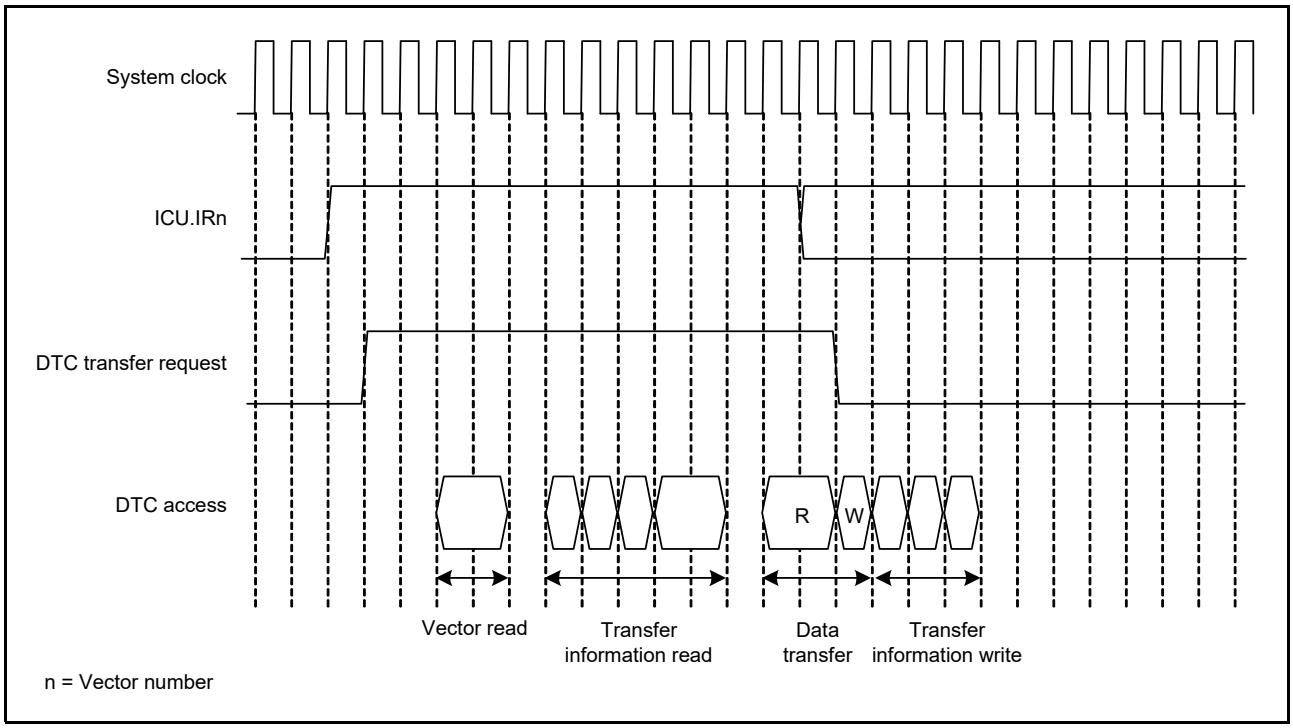


Figure 20.10 Example (1) of DTC Operation Timing (Normal Transfer Mode, Repeat Transfer Mode)

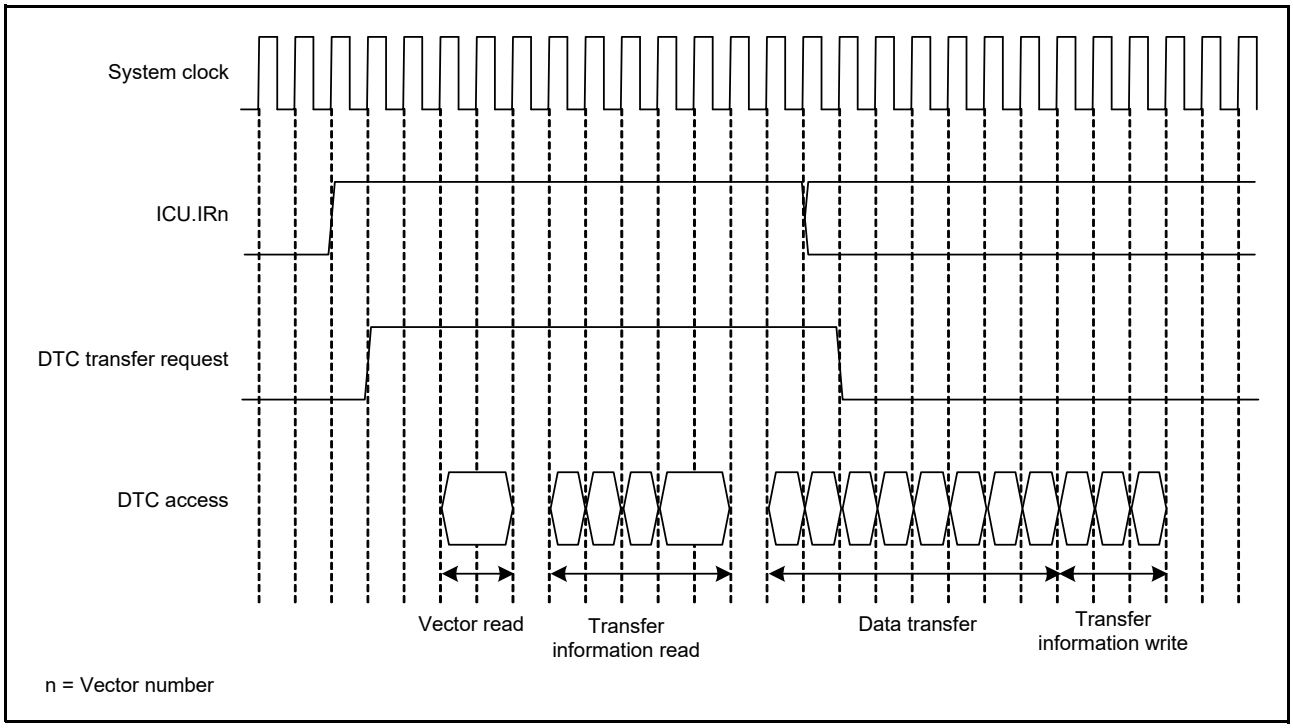


Figure 20.11 Example (2) of DTC Operation Timing (Block Transfer Mode, Block Size = 4)

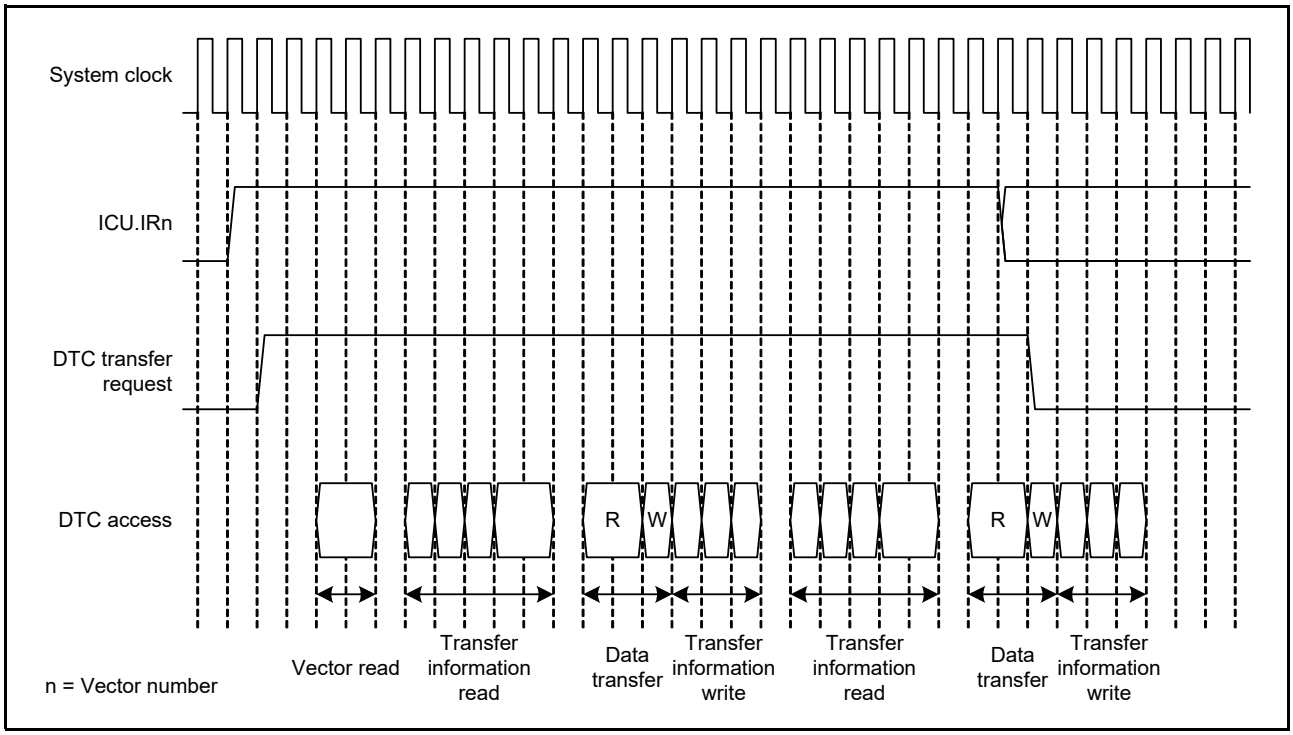
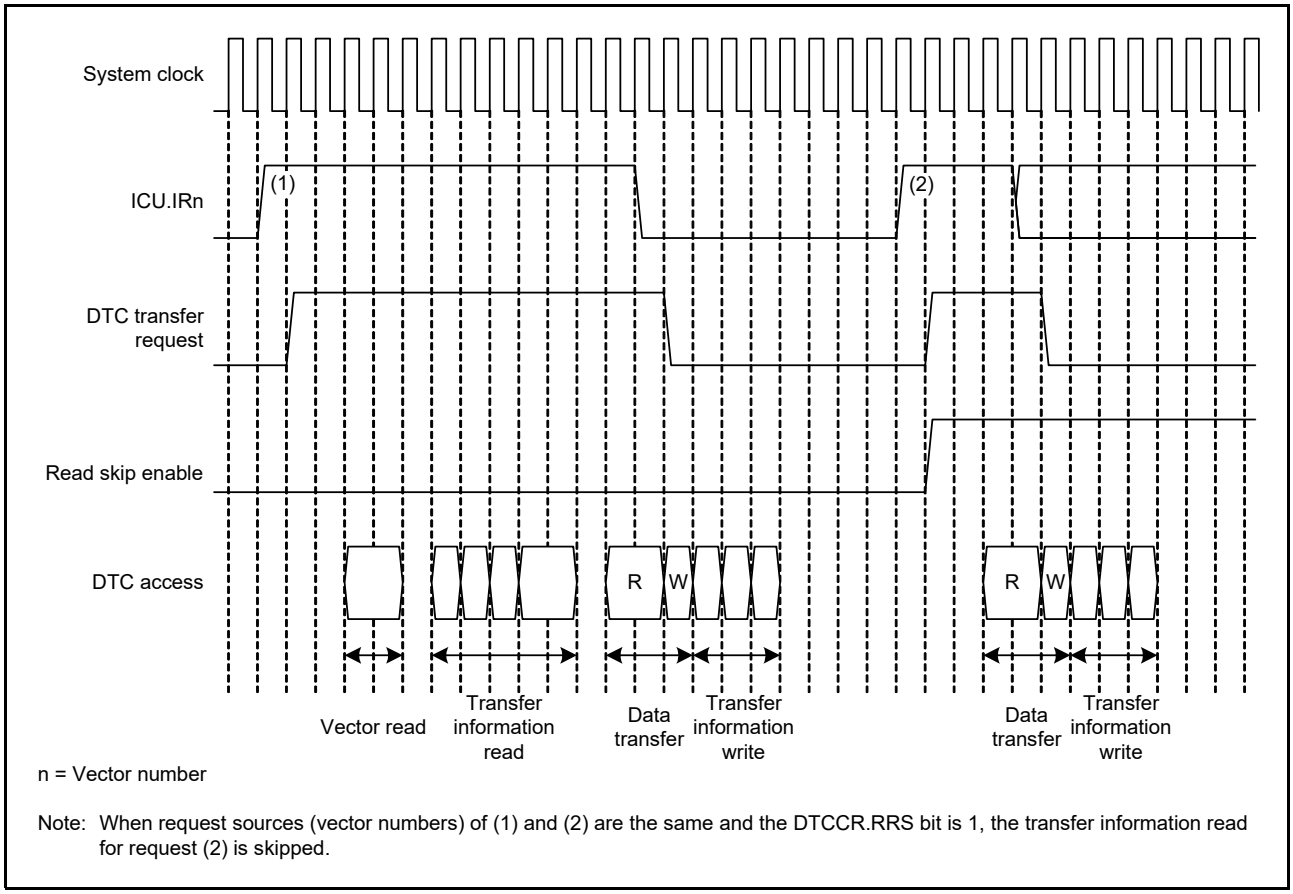


Figure 20.12 Example (3) of DTC Operation Timing (Chain Transfer)



Note: When request sources (vector numbers) of (1) and (2) are the same and the DTCCR.RRS bit is 1, the transfer information read for request (2) is skipped.

Figure 20.13 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

20.4.8 Sequence Transfer

A sequence transfer can be executed on the request source that is specified in the DTCSQE register. A sequence transfer is started by setting the MRB.INDX bit to 1 and ended by setting the MRB.SQEND bit to 1. Setting the DTCOR.SQTFRL bit to 1, even during a sequence transfer, forcibly ends the transfer and the next DTC transfer request starts new sequence transfer with reference to the index table.
A sequence transfer includes the following processing.

(1) The first data transfer is executed by referring to the DTC vector table in response to a DTC transfer request from the source specified in the DTCSQE register.
(2) The DTC index table is referred based on the value of the lower 8 bits of the first data transferred data in (1) (sequence number).
(3) The transfer information is read from the address obtained from the DTC index table.
(4) A data transfer is executed in accordance with the transfer information. On completion of the data transfer, either one of the following operations is performed by using the values of bits MRB.CHNE and MRB.SQEND.
- A chain transfer is executed when the CHNE bit is 1. The next transfer information is read. Go to (4).
- The sequence transfer is suspended when the CHNE bit is 0 and the SQEND bit is 0. Go to (5).
- The sequence transfer ends when the CHNE bit is 0 and the SQEND bit is 1.

(5) When a DTC transfer request from the source specified in the DTCSQE register is accepted, the suspended sequence transfer is resumed and the next transfer information is read. Go to (4).

Note 1. When the ICU.DTCERn.DTCE bit becomes 0 based on the result of the data transfer, a DTC transfer request is not generated. Set the DTCE bit to 1 to resume sequence transfer. Refer to Figure 20.5 or section 15, Interrupt Controller (ICUM) for the conditions where the DTCE bit becomes 0.

Figure 20.14 and Figure 20.15 shows a basic sequence transfer operation.

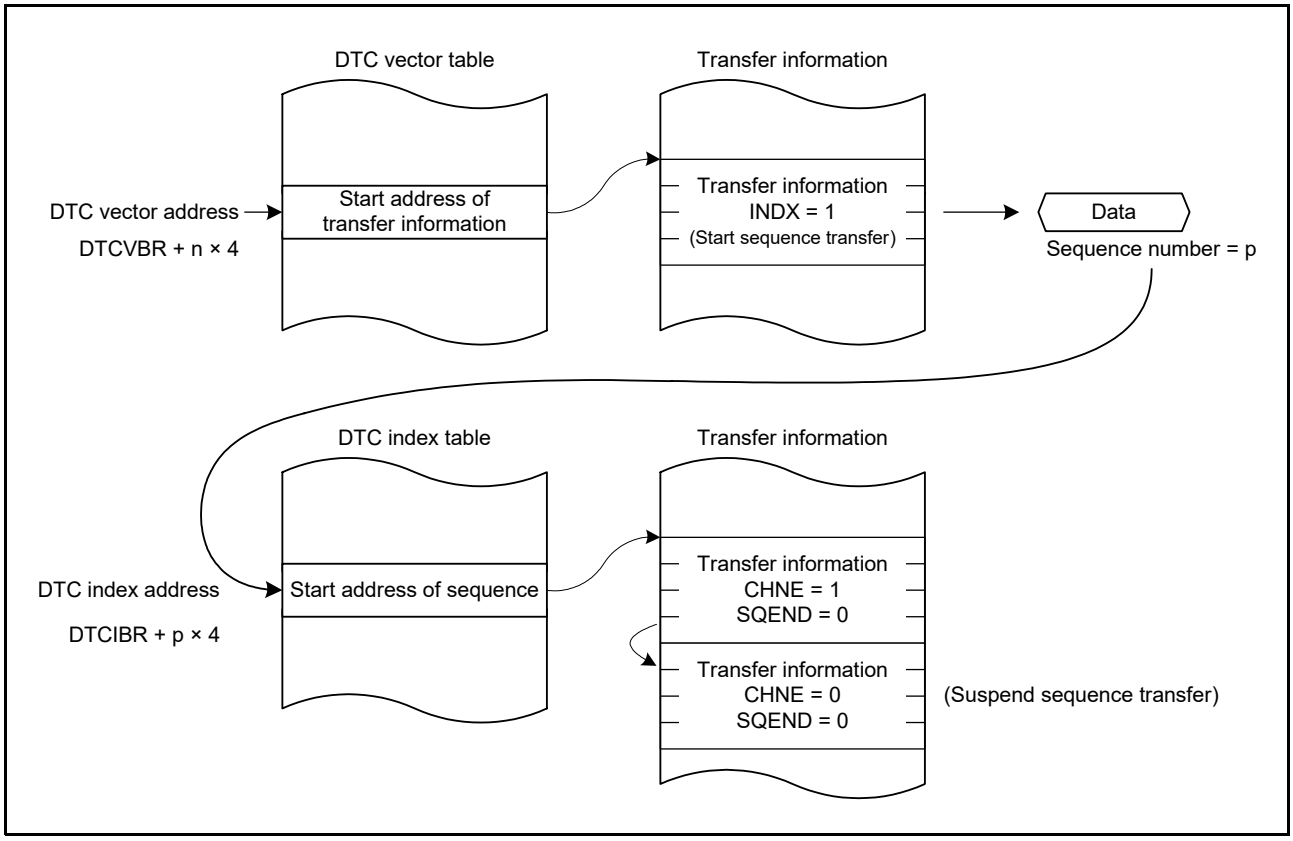


Figure 20.14 Start and Suspension of Sequence Transfer

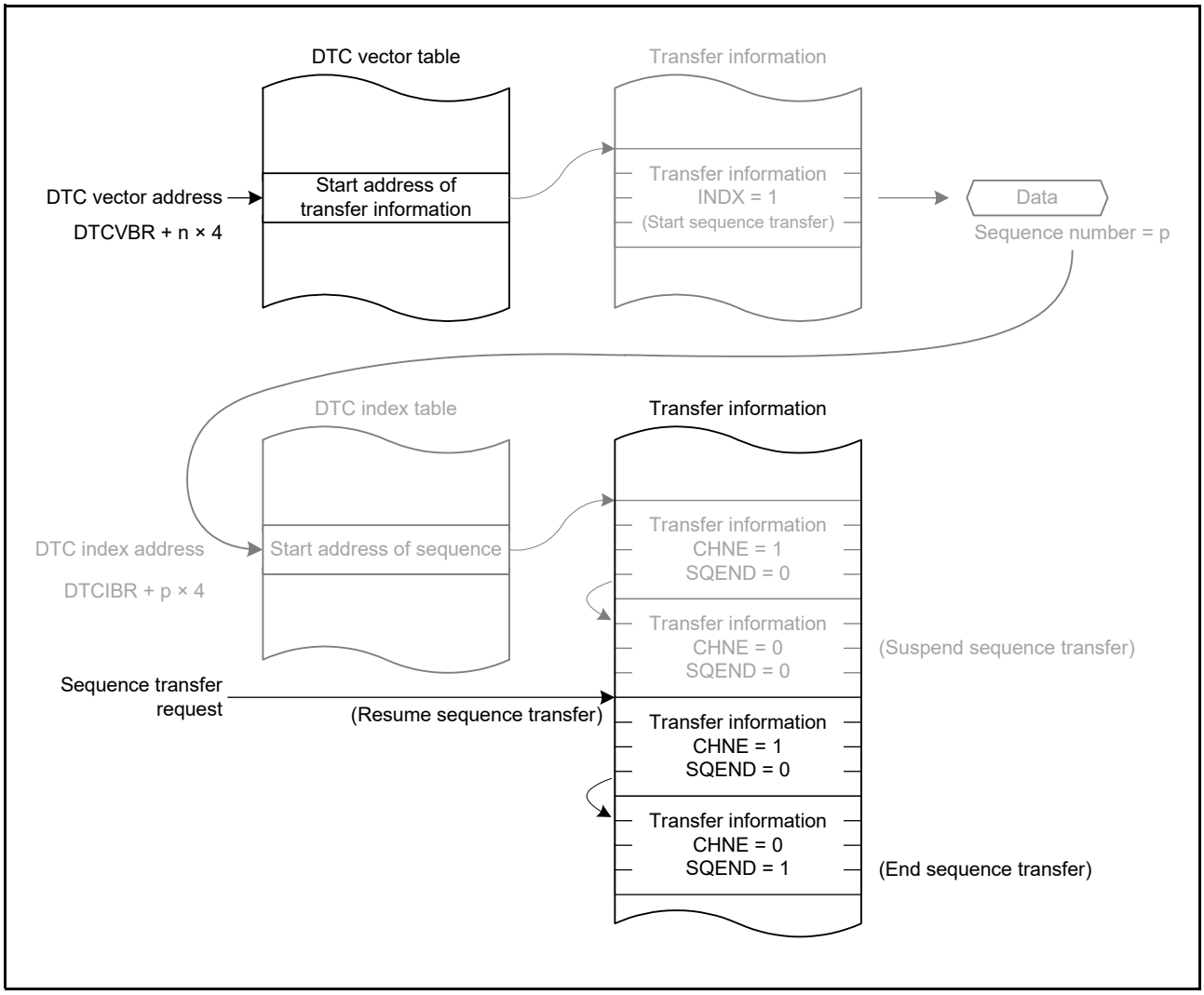


Figure 20.15 Resumption and End of Sequence Transfer

Table 20.9 lists the settings of bits CHNE, SQEND, and INDX during the sequence transfer.

Table 20.9 Sequence Transfer Process and Values of Bits CHNE, SQEND, and INDX

DTC Operations	CHNE Bit	SQEND Bit	INDX Bit
Start sequence transfer	0	0	1*1
Continue sequence transfer	1	0	0
Suspend sequence transfer*2	0	0	0
End sequence transfer	0	1	0
End current sequence transfer and start new sequence transfer	0	1	1*1
Some other transfer (not sequence transfer)	—	0	0

Note: Do not set the values other than listed above.
Note 1. Set MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.
Note 2. When a sequence transfer is suspended, the ICU.DTCERn.DTCE bit may become 0. Set the DTCE bit to 1 to resume sequence transfer.

Even when a sequence transfer is suspended, a new sequence transfer cannot start until the suspended sequence transfer is eventually completed. When a sequence transfer request is received during suspension of the sequence transfer, the suspended sequence transfer is resumed.

20.4.9 DTC Index Table

The DTC index table is allocated to the area where its start address is configured in the DTCIBR register.
Store the start address of transfer information table p for sequence number p in the address of DTCIBR + p × 4.
The upper 30 bits of the start address is set to the upper 30 bits of the DTC index. Set the CPUSEL bit to select either of reading the transfer information and starting the sequence, or output an interrupt request to the CPU without starting the sequence. For a complicated sequence that the DTC cannot handle, set the CPUSEL bit to 1 to allow the CPU to handle such a sequence.

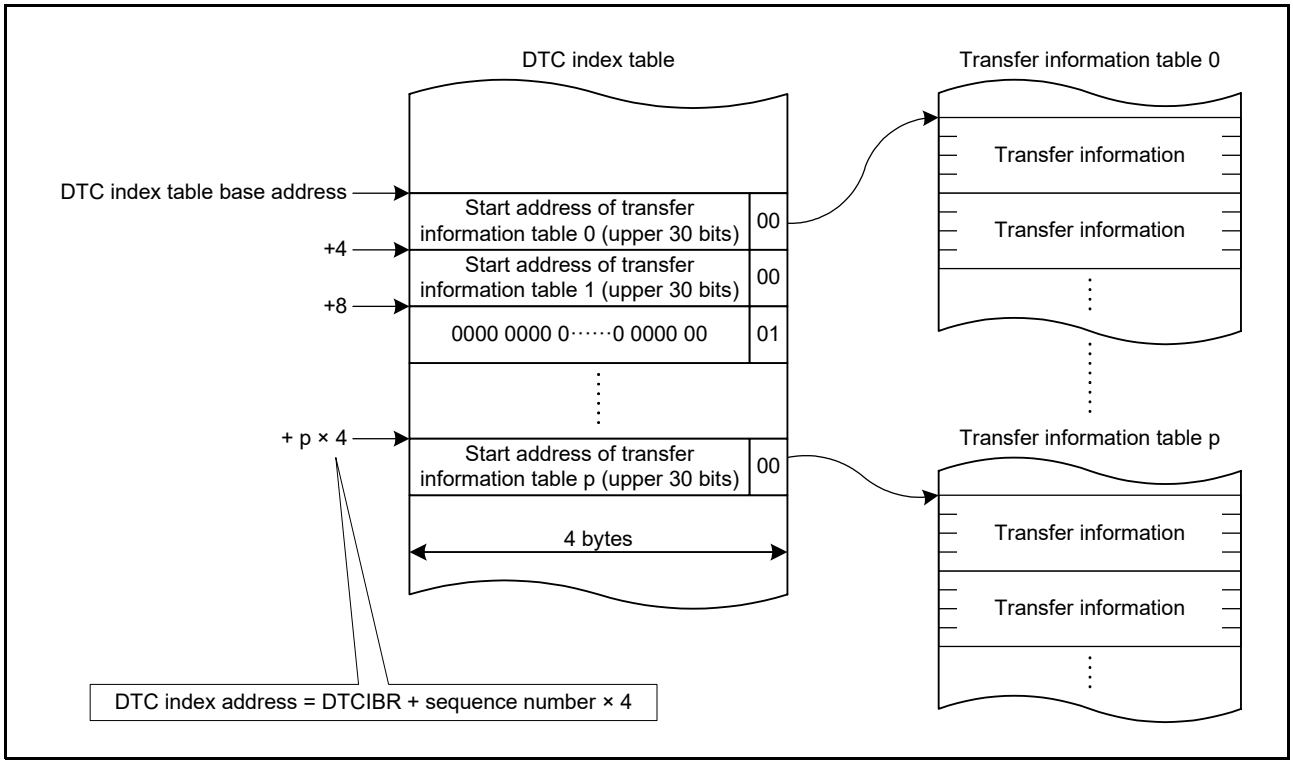
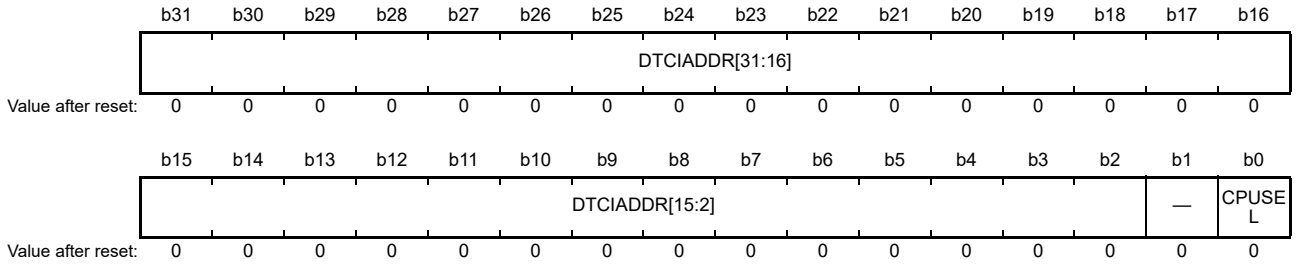


Figure 20.16 DTC Index Table

- DTC Index

Address(es): DTCIBR + p × 4



Bit	Symbol	Bit Name	Description	R/W
b0	CPUSEL	Sequence Transfer/CPU Interrupt Select	0: Continues the sequence transfer (starts the sequence) 1: Ends the sequence transfer and outputs an interrupt request to the CPU	—
b1	—	Reserved	Set this bit to 0.	—
b31 to b2	DTCIADDR[31:2]	Transfer Information Table Address	Set the upper 30 bits of the start address of the transfer information table to these bits. Writing to the upper 4 bits (b31 to b28) is ignored and the values in b31 to b28 become the same value as b27.	—

When the CPUSEL bit in the DTC index that the obtained sequence number indicates is 1, an interrupt request to the CPU is generated. At this time, the ICU.DTCERn.DTCE bit becomes 0. From this point, the interrupt request signal from the request source that is specified in the DTCSQE register is sent to the CPU, but not DTC. After completion of CPU interrupt processing, set the ICU.DTCERn.DTCE bit to 1 to enable DTC transfer request for starting the next sequence transfer.

20.4.10 Example of Sequence Transfer

Figure 20.17 shows a typical examples of a sequence transfer and Figure 20.18 to Figure 20.22 show configurations of the transfer information for the examples of the transfers in the figure.
In these examples, the interrupt source of vector number n is set as the source of the sequence transfer (DTCSQE.VECN[7:0] = n).
Once the DTC transfer request due to the interrupt source of vector number n (hereinafter referred to as “transfer request n”) is input, the DTC refers to the DTC vector table and reads the corresponding transfer information. The lower 8 bits that have been transferred based on the transfer information become a sequence number, selecting one sequence among possible 256 sequences.

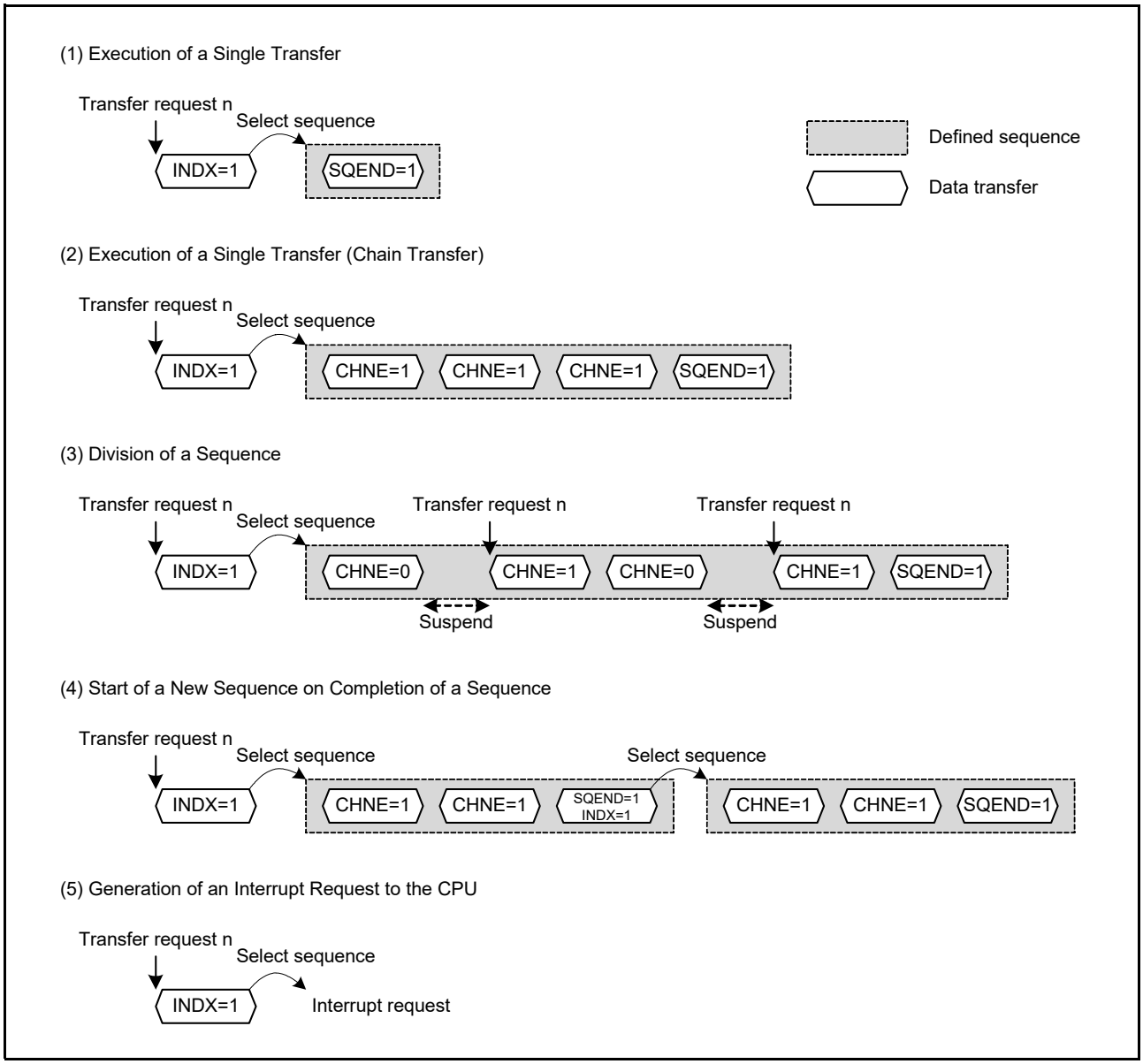


Figure 20.17 Examples of Sequence Transfers

(1) When Executing a Single Transfer

Figure 20.18 shows an example of a single transfer (normal, repeat, or block).
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number p.
Since the values of the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively, the sequence ends after the specified transfer is executed.

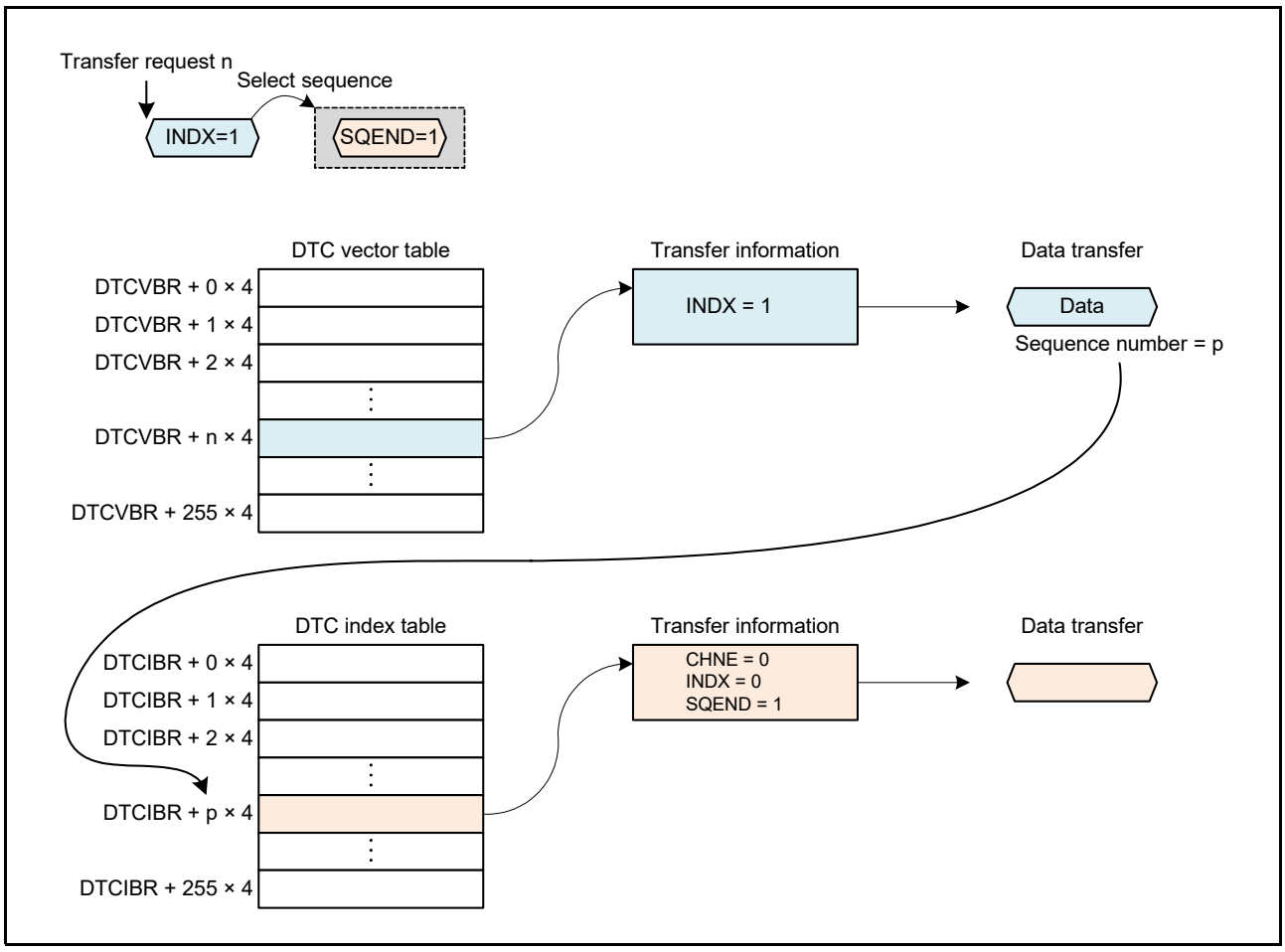


Figure 20.18 Example of a Sequence of Single Transfer

(2) When Executing a Single Chain Transfer

Figure 20.19 is an example of a sequence for a single chain transfer.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number q.
While the values of the CHNE, INDX, and SQEND bits are 1, 0, and 0 respectively, the specified chain transfer is executed. When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

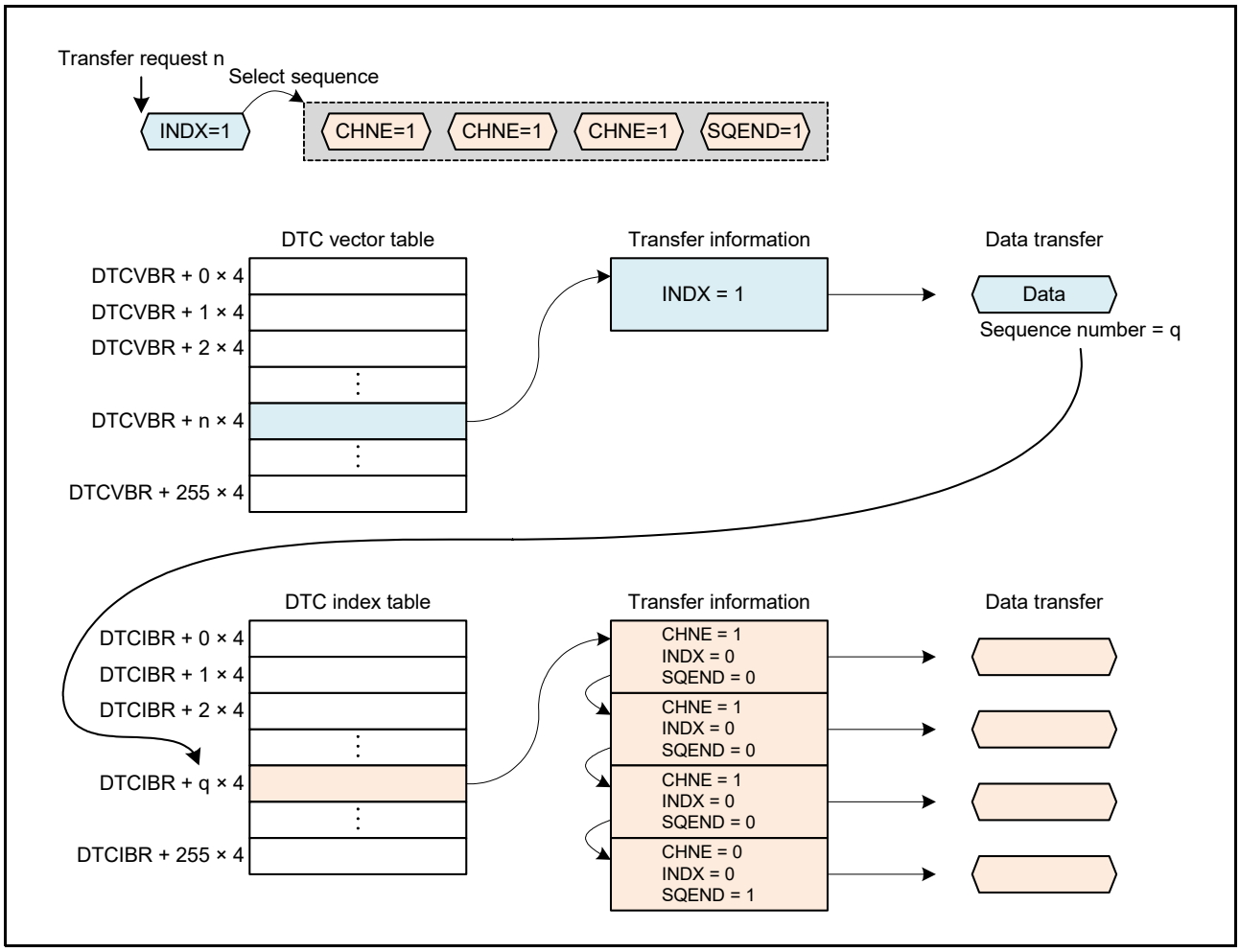


Figure 20.19 Example of Sequence of a Single Chain Transfer

(3) When Dividing a Sequence

Figure 20.20 is an example of the sequence that is divided into 3 parts.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number r.
Since the values of the CHNE, INDX, and SQEND bits in the transfer information are 0, 0, and 0 respectively, the sequence is suspended after the specified transfer is executed and the DTC waits for the next transfer request n.
When the transfer request n is input during a sequence transfer, the DTC vector table is not referred and the suspended sequence is resumed.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

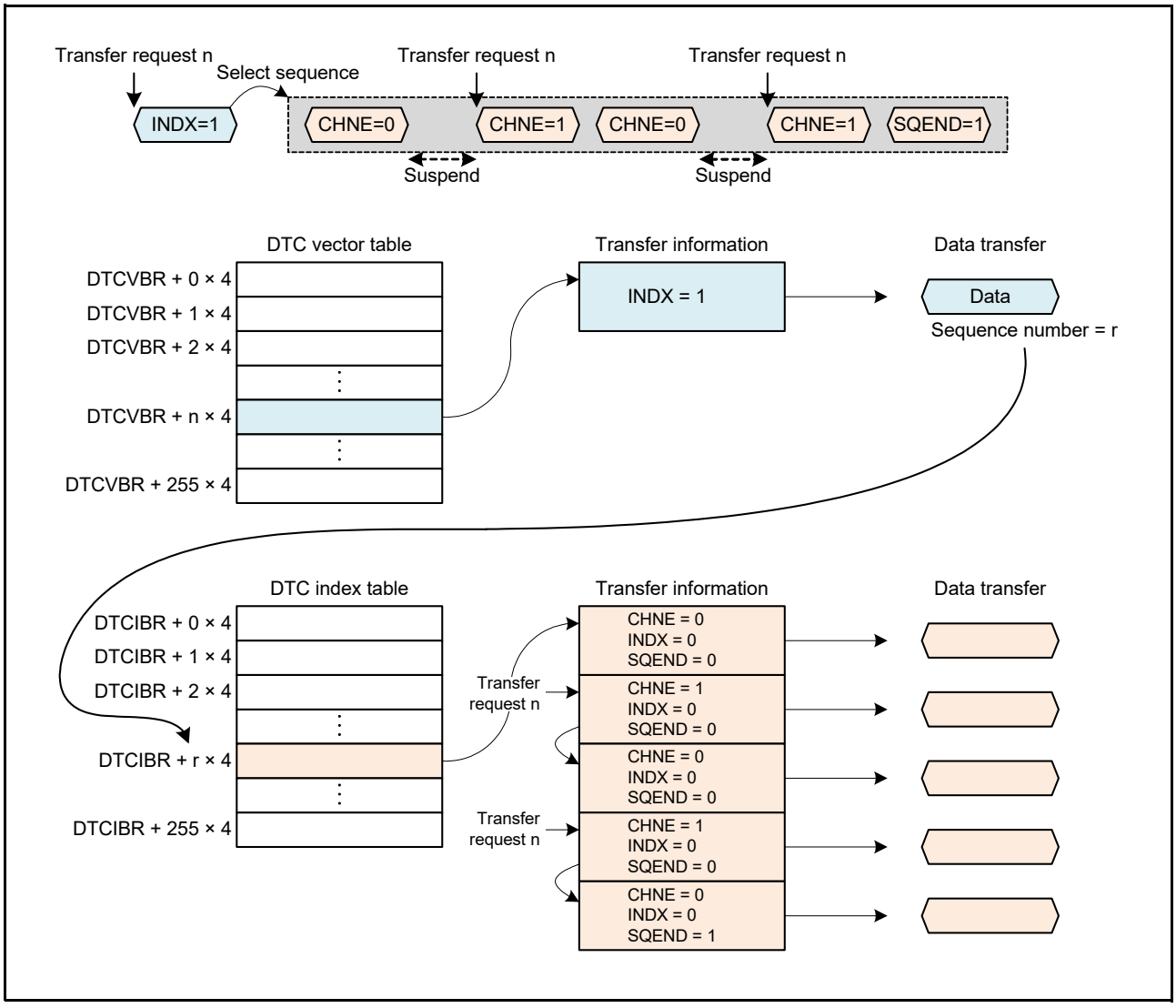


Figure 20.20 Example of Divided Sequence

(4) When Starting a New Sequence on completion of a Sequence

Figure 20.21 is an example for starting the next and new sequence on completion of the first sequence.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number s.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 1, and 1 respectively is read, the specified transfer is executed, then a new sequence number is obtained from the lower 8 bits of the transferred data.
The DTC again refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number k and then starts a new sequence.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

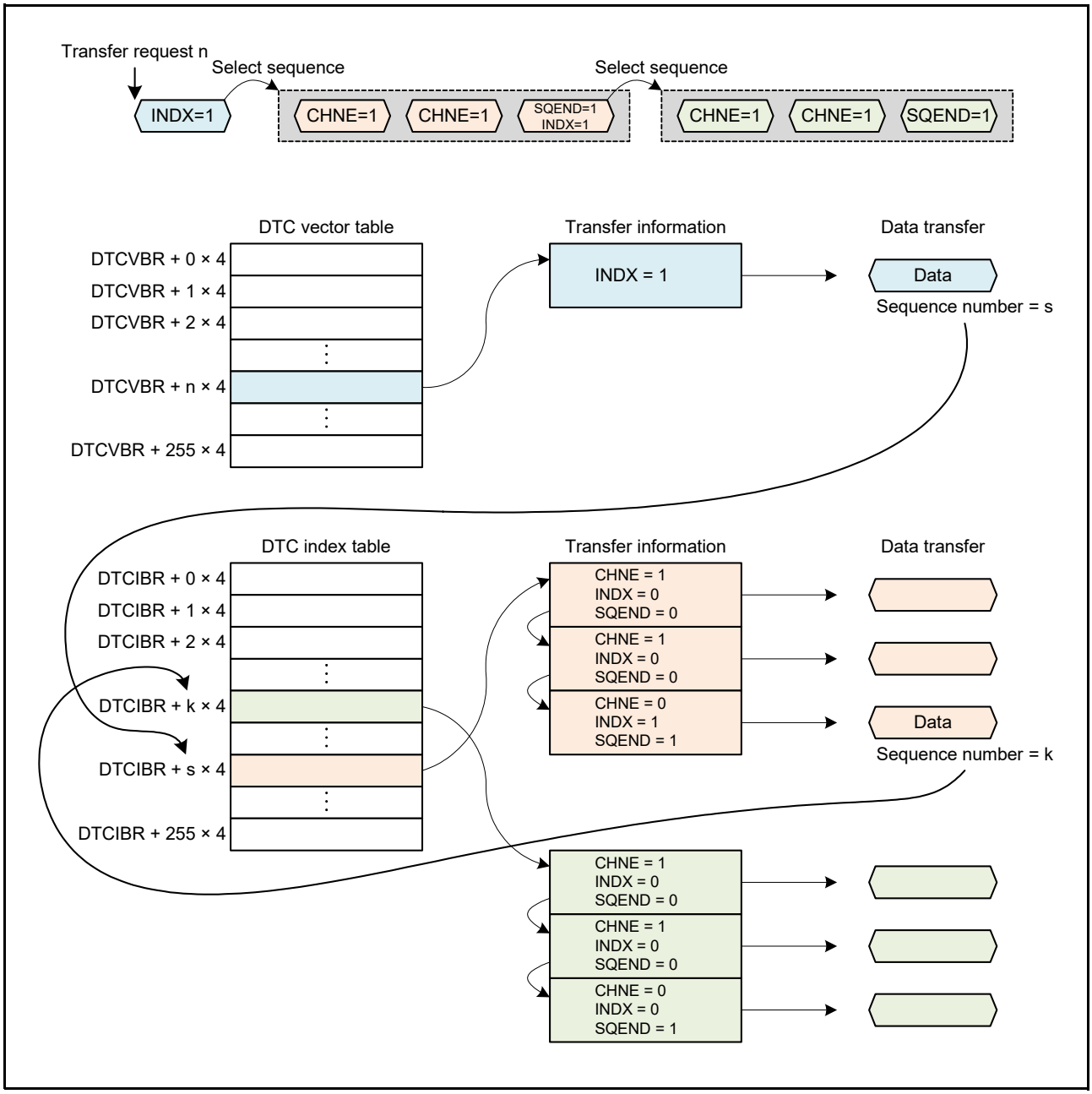


Figure 20.21 Example When Starting a New Sequence on Completion of a Sequence

(5) When Generating an Interrupt Request to the CPU

Figure 20.22 is an example of that an interrupt request is output to the CPU without starting of sequence.
The DTC obtains a DTC index that corresponds to the obtained sequence number t.
When the CPUSEL bit of the obtained DTC index is 1, the DTC ends the sequence transfer without starting the sequence, and then outputs an interrupt request to the CPU.

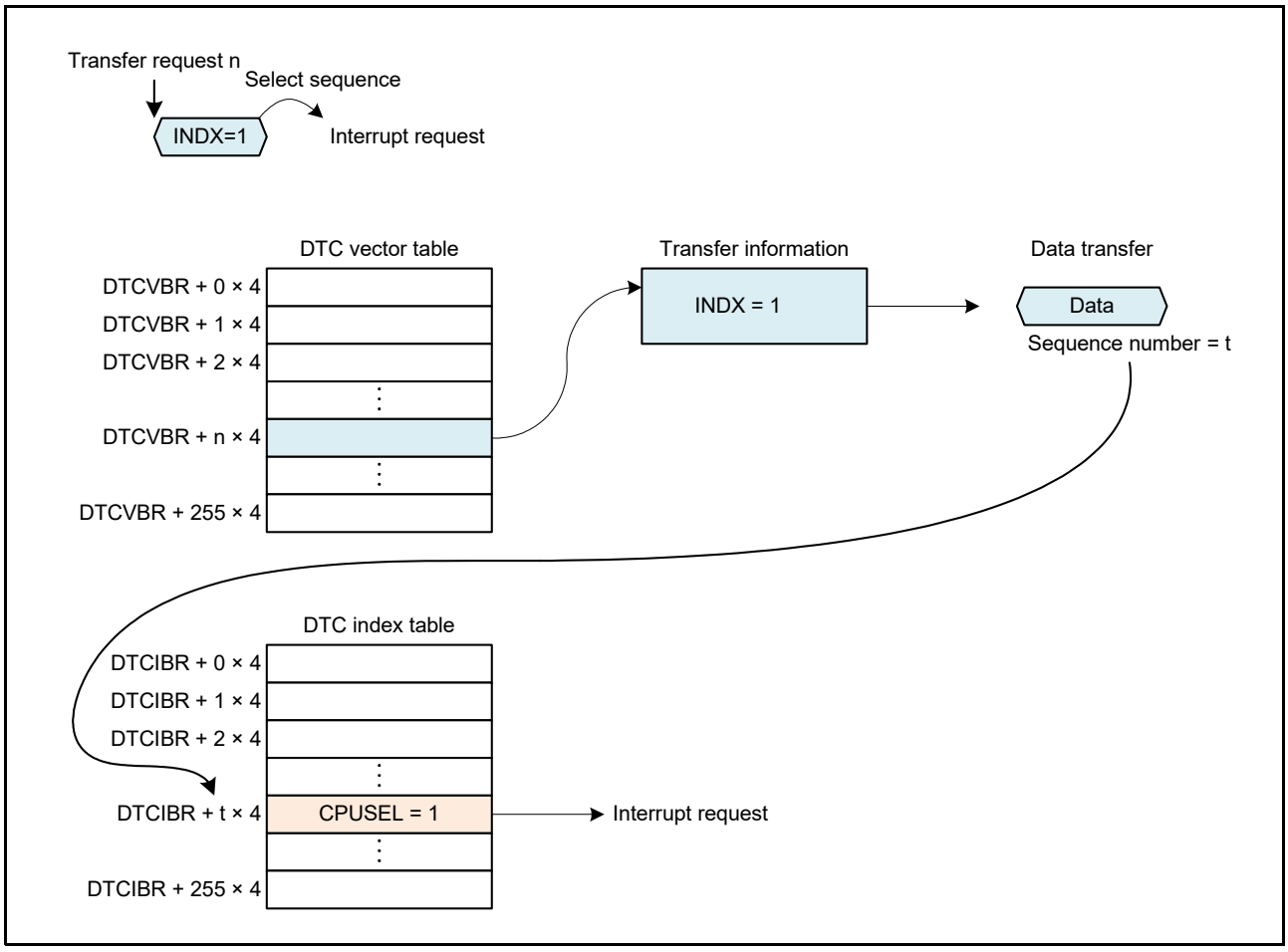


Figure 20.22 Example of Output of an Interrupt Request to the CPU

20.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR). When using sequence transfer, also set the DTC index table base register (DTCIBR).

Figure 20.23 shows the procedure to set the DTC.

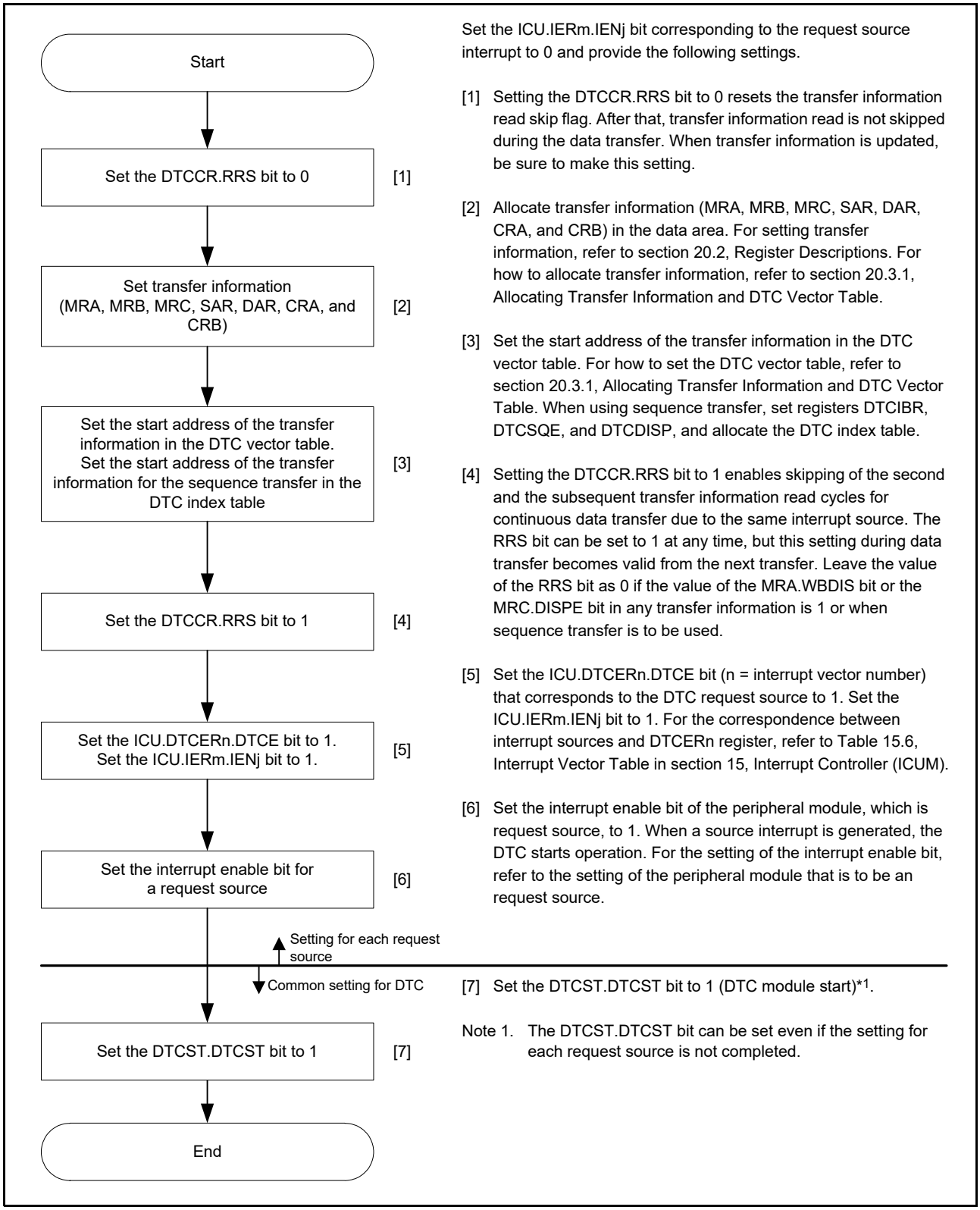


Figure 20.23 Procedure to Set the DTC

20.6 Examples of DTC Usage

20.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an RSCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the RSCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) RSCI Setting

Enable the RXI interrupt by setting the SCR0.RIE bit in the RSCI to 1. If a reception error occurs during the RSCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the RSCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the RSCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

20.6.2 Chain Transfer

As an example of chain transfer by the DTC, its employment in the output of pulses by a GPTW is described below. Chain transfer is used to transfer the compare value and period of the PWM timer and change the duty and period of the PWM timer. For the first chain transfer, specify normal transfer mode for the transfer to the GPTWn.GTCCRC register. For the second chain transfer, specify normal transfer mode for the transfer to the GPTWn.GTCCRE register. For the third chain transfer, specify normal transfer mode for the transfer to the GPTWn.GTPBR register.
An example of using the overflow interrupt of the GPTWn.GTCNT counter as a request source for the DTC is provided below.

(1) First Transfer Information Setting

Settings should be made for transfer to the GPTWn.GTCCRC register. Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 10b (longword transfer), and the MRA.SM[1:0] bits to 10b (SAR is incremented after data transfer). Set the MRB.CHNE bit to 1 (chain transfer is enabled), the MRB.CHNS bit to 0 (chain transfer is performed after each transfer), and the MRB.DM[1:0] bits to 00b (destination address is fixed). Set the SAR to the start address of the data table, the DAR register to the address of the GPTWn.GTCCRC register, and the CRA register to the size of the data table. The CRB register can be set to any value.

(2) Second Transfer Information Setting

Settings should be made for transfer to the GPTWn.GTCCRE register. Make the same settings as (1) except for setting the DAR register to the address of the GPTWn.GTCCRE register.

(3) Third Transfer Information Setting

Settings should be made for transfer to the GPTWn.GTPBR register. Make the same settings as (1) except for setting the MRB.CHNE bit to 0 (chain transfer is disabled), DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the DAR register to the address of the GPTWn.GTPBR register.

(4) Transfer Information Assignment

Place the second and third transfer information immediately after the first transfer information.

(5) DTC Vector Table

In the DTC vector table, set the address where the first transfer information starts.

(6) ICU Setting and DTC Module Activation

Set the ICU.DTCERn.DTCE bit corresponding to the GTCIVn interrupt and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(7) GPTW Setting

Set the GPTWn.GTIOR register so that the output changes on a compare match with the GPTWn.GTCCRA and GTCCRB registers. Set the compare match value for the first cycle in the GTCCRA and GTCCRB registers, and the compare match value for the second cycle in the GTCCRC and GTCCRE registers. Set the period of the first cycle of the PWM waveform in the GTPR register, and the period of the second cycle in the GTPBR register. Enable the output of the GTIOCnA and GTIOCnB pins using the PijPFS registers.

(8) GPTW Activation

Set the GPTWn.GTSTR.CSTARTn bit to 1 to start counting operation of the GPTWn.GTCNT counter.

(9) DTC Transfer

Every time the GPTWn.GTCNT counter overflows, the next compare match values are transferred to the GTCCRC and GTCCRE registers, and the next period value is transferred to the GTPBR register.

(10) Interrupt Handling

When the specified number of data transfers is completed (i.e. when the value of the CRA register for third transfer becomes 0), a GTCIVn interrupt request is issued for the CPU. Complete the process in the handling routine for this interrupt.

20.6.3 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.
The following shows an example of configuring a 128-Kbyte input buffer to addresses 0200 0000h to 0201 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 20.24 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
(2) Prepare the upper 16 bits (in this case, 0201h and 0200h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).
(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 16 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.
(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 16 bits of the transfer destination address of the first data transfer is set to 0201h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 16 bits of the transfer destination address of the first data transfer is set to 0200h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

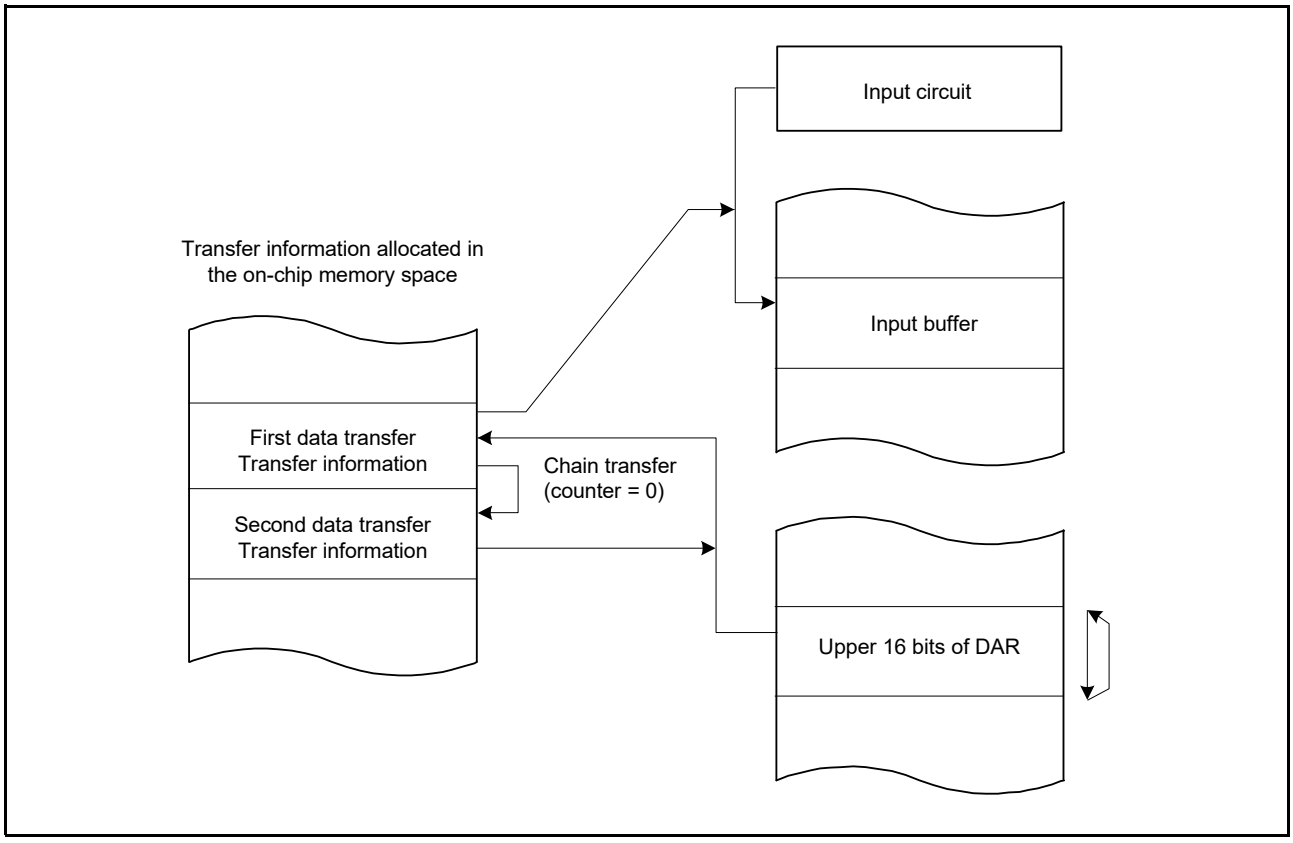


Figure 20.24 Chain Transfer When the Counter is 0

20.6.4 Sequence Transfer

The following is an example of using the RSCI receive interrupt as a request source of sequence transfer.

(1) Transfer Information Settings

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer), the MRB.INDX bit to 1 (start sequence transfer), and the MRB.SQEND bit to 0 (continue the sequence transfer). The MRB.DTS bit can be set to any value. Set the address of the RSCIk.RDR register in the SAR register and set the start address of the RAM area which stores the data in the DAR register.
When the MRA.WBDIS bit is set to 1 (Does not write back the transfer information), the values of registers CRA and CRB are ignored.

(2) DTC Vector Table Setting

Set the start address of the transfer information for the corresponding receive data full interrupt (RXI) in the DTC vector table.

(3) DTC Index Table Setting

Set the start address of the transfer information for each sequence in the DTC index table.

(4) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1. Set the DTCST.DTCST bit to 1.

(5) RSCI Setting

Set the RSCIk.SCR0.RIE bit to 1 to enable the RXI interrupt. If a reception error occurs during the RSCI receive operation, subsequent receptions are not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(6) Start of the Sequence Transfer

On completion of reception of 1-byte data by the RSCI, an RXI interrupt is generated to start the DTC. The DTC transfers the received data from the RSCIk.RDR register to the RAM. The DTC looks up the DTC index table by using the value from the received data (sequence number) and continues to transfer data corresponding to the that number. When the CPUSEL bit in the DTC index is 1, the DTC does not read the transfer information and sets the ICU.DTCERn.DTCE bit to 0. Then the DTC outputs an interrupt request to the CPU and ends the sequence transfer.

(7) During Suspension of the Sequence Transfer

Set the ICU.DTCERn.DTCE bit to 1 if the bit is 0. The DTC continues to transfer the data for every generation of the DTC transfer request in response to the corresponding RXI interrupt.

(8) End of the Sequence Transfer

Set the MRB.SQEND bit in the last transfer information of the sequence transfer to 1. After execution of this data transfer, the DTC ends the sequence transfer. The DTC starts to refer to the DTC vector table when a DTC transfer request is generated due to the next corresponding RXI interrupt.

20.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[4:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

20.8 Event Link

The DTC outputs an event signal on completing data transfer in response to one request.

20.9 Low Power Consumption Function

Before making a transition to the module stop state, software standby mode, or deep software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA22 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA22 bit releases the DTC from the module stop state.

(2) Software Standby and Deep Software Standby Modes

Make settings according to the procedure under section 11.6.2.1, Transitioning to Software Standby Mode or section 11.6.3.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode or deep software standby mode follows the completion of the data transfer.

(3) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.8.6, Timing of the WAIT Instruction in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 15.7.3.1, Interrupt Request Destination Setting Procedure in section 15, Interrupt Controller (ICUM), and then execute the WAIT instruction.

20.10 Usage Notes

20.10.1 Start Address of Transfer Information

Set multiples of 16 for the start addresses of the transfer information to be specified in the DTC vector table.

20.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 20.25. For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus Ch and the CRB setting to the address plus Eh. In little endian, write the CRB setting to the address plus Ch and the CRA setting to the address plus Eh. When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus Ch, regardless of endian.

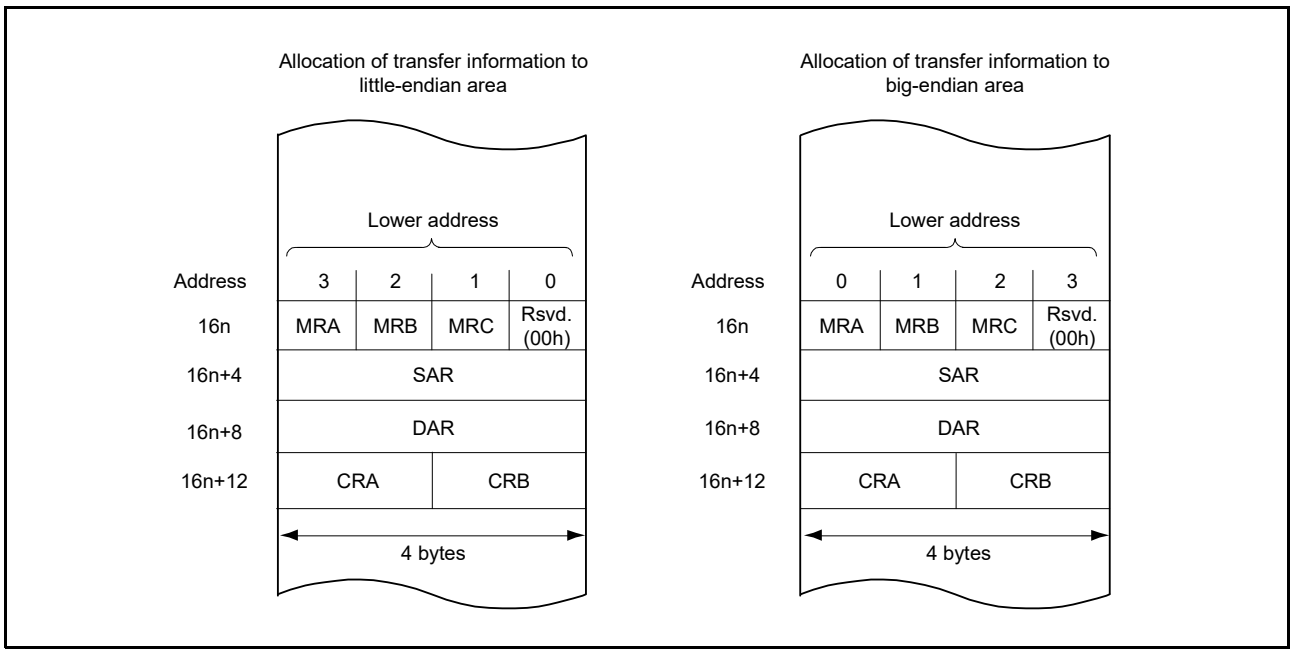


Figure 20.25 Allocation of Transfer Information

20.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn)

The DMA request should not be issued by setting the DMAC trigger select register (ICU.DMRSRm (m = DMAC channel number)) to the same vector number that has been specified by setting the ICU.DTCERn.DTCE bit to 1 (the corresponding interrupt source is selected as the DTC trigger). For details on the ICU.DTCERn and ICU.DMRSRm registers (m = DMAC channel number), refer to section 15, Interrupt Controller (ICUM).

20.10.4 Notes on Using the Sequence Transfer

When sequence transfer is to be used, make sure that the DTCCR.RRS bit is 0 (transfer information read is not skipped). In addition, set the MRB.CHNE bit to 0 (chain transfer is disabled) when setting the MRB.INDX bit to 1 (start sequence transfer and refer the index table) or the MRB.SQEND bit to 1 (end the sequence transfer).

21. Event Link Controller (ELC)

21.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, and interconnects (links) peripheral modules. As a result, peripheral modules can directly perform interlinked operation among them without using software.
Event signals can be output regardless of the settings of the corresponding interrupt request enable bits.

Table 21.1 lists the specifications of the ELC, and Figure 21.1 shows a block diagram of the ELC.

Table 21.1 ELC Specifications

Item	Description
Event link function	• 377 types of event signals can be directly interconnected to modules. • Operation for timer modules when inputting an event signal can be selected.
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attribute can be set for each register.

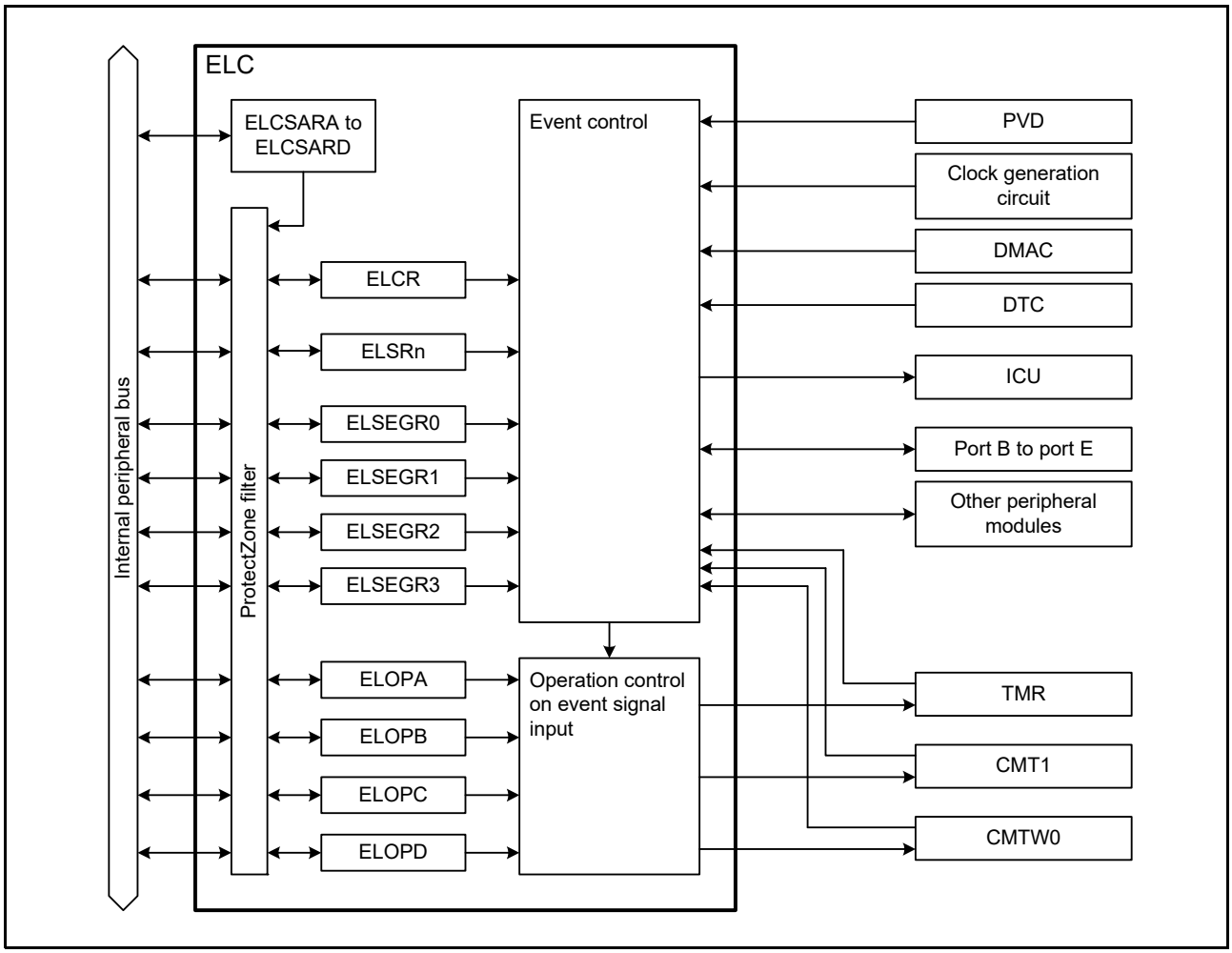


Figure 21.1 ELC Block Diagram

21.2 Register Descriptions

21.2.1 ELC Security Attribution Register A (ELCSARA)

Address(es): ELC.ELCSARA 8780 1030h, ELC_NS.ELCSARA 0780 1030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	ELOPD	ELOPC	ELOPB	ELOPA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ELSEGR3	ELSEGR2	ELSEGR1	ELSEGR0	ELCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ELCR	ELCR Security Attribution	0: Secure 1: Non-secure	R/W
b1	ELSEGR0	ELSEGR0 Security Attribution		R/W
b2	ELSEGR1	ELSEGR1 Security Attribution		R/W
b3	ELSEGR2	ELSEGR2 Security Attribution		R/W
b4	ELSEGR3	ELSEGR3 Security Attribution		R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ELOPA	ELOPA Security Attribution	0: Secure 1: Non-secure	R/W
b17	ELOPB	ELOPB Security Attribution		R/W
b18	ELOPC	ELOPC Security Attribution		R/W
b19	ELOPD	ELOPD Security Attribution		R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the ELCR, ELSEGR0 to ELSEGR3, and ELOPA to ELOPD registers.

21.2.2 ELC Security Attribution Register B (ELCSARB)

Address(es): ELC.ELCSARB 8780 1034h, ELC_NS.ELCSARB 0780 1034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	ELSR26	ELSR25	ELSR24	ELSR23	ELSR22	ELSR21	ELSR20	ELSR19	ELSR18	ELSR17	ELSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ELSR15	ELSR14	ELSR13	ELSR12	ELSR11	ELSR10	ELSR9	ELSR8	ELSR7	ELSR6	ELSR5	ELSR4	ELSR3	ELSR2	ELSR1	ELSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ELSR0	ELSR0 Security Attribution	0: Secure 1: Non-secure	R/W
b1	ELSR1	ELSR1 Security Attribution		R/W
b2	ELSR2	ELSR2 Security Attribution		R/W
b3	ELSR3	ELSR3 Security Attribution		R/W
b4	ELSR4	ELSR4 Security Attribution		R/W
b5	ELSR5	ELSR5 Security Attribution		R/W
b6	ELSR6	ELSR6 Security Attribution		R/W
b7	ELSR7	ELSR7 Security Attribution		R/W
b8	ELSR8	ELSR8 Security Attribution		R/W
b9	ELSR9	ELSR9 Security Attribution		R/W
b10	ELSR10	ELSR10 Security Attribution		R/W
b11	ELSR11	ELSR11 Security Attribution		R/W
b12	ELSR12	ELSR12 Security Attribution		R/W
b13	ELSR13	ELSR13 Security Attribution		R/W
b14	ELSR14	ELSR14 Security Attribution		R/W
b15	ELSR15	ELSR15 Security Attribution		R/W
b16	ELSR16	ELSR16 Security Attribution		R/W
b17	ELSR17	ELSR17 Security Attribution		R/W
b18	ELSR18	ELSR18 Security Attribution		R/W
b19	ELSR19	ELSR19 Security Attribution		R/W
b20	ELSR20	ELSR20 Security Attribution		R/W
b21	ELSR21	ELSR21 Security Attribution		R/W
b22	ELSR22	ELSR22 Security Attribution		R/W
b23	ELSR23	ELSR23 Security Attribution		R/W
b24	ELSR24	ELSR24 Security Attribution		R/W
b25	ELSR25	ELSR25 Security Attribution		R/W
b26	ELSR26	ELSR26 Security Attribution		R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the ELSR0 to ELSR26 registers.

21.2.3 ELC Security Attribution Register C (ELCSARC)

Address(es): ELC.ELCSARC 8780 1038h, ELC_NS.ELCSARC 0780 1038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	ELSR52	ELSR51	ELSR50	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ELSR47	ELSR46	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	ELSR46	ELSR46 Security Attribution	0: Secure 1: Non-secure	R/W
b15	ELSR47	ELSR47 Security Attribution		R/W
b17 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18	ELSR50	ELSR50 Security Attribution	0: Secure 1: Non-secure	R/W
b19	ELSR51	ELSR51 Security Attribution		R/W
b20	ELSR52	ELSR52 Security Attribution		R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the ELSR46, ELSR47, and ELSR50 to ELSR52 registers.

21.2.4 ELC Security Attribution Register D (ELCSARD)

Address(es): ELC.ELCSARD 8780 103Ch, ELC_NS.ELCSARD 0780 103Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	ELSR91	ELSR90	ELSR89	ELSR88	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ELSR78	ELSR77	ELSR76	ELSR75	ELSR74	ELSR73	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	ELSR73	ELSR73 Security Attribution	0: Secure 1: Non-secure	R/W
b10	ELSR74	ELSR74 Security Attribution		R/W
b11	ELSR75	ELSR75 Security Attribution		R/W
b12	ELSR76	ELSR76 Security Attribution		R/W
b13	ELSR77	ELSR77 Security Attribution		R/W
b14	ELSR78	ELSR78 Security Attribution		R/W
b23 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ELSR88	ELSR88 Security Attribution	0: Secure 1: Non-secure	R/W
b25	ELSR89	ELSR89 Security Attribution		R/W
b26	ELSR90	ELSR90 Security Attribution		R/W
b27	ELSR91	ELSR91 Security Attribution		R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is used to set the security attributes for the ELSR73 to ELSR78, and ELSR88 to ELSR91 registers.

21.2.5 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 8780 1000h, ELC_NS.ELCR 0780 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The ELCR register controls operation of the ELC.

21.2.6 Software Event Generation Register m (ELSEGRm) (m = 0 to 3)

Address(es): ELC.ELSEGR0 8780 1004h, ELC.ELSEGR1 8780 1008h, ELC.ELSEGR2 8780 100Ch, ELC.ELSEGR3 8780 1010h, ELC_NS.ELSEGR0 0780 1004h, ELC_NS.ELSEGR1 0780 1008h, ELC_NS.ELSEGR2 0780 100Ch, ELC_NS.ELSEGR3 0780 1010h

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	Writing 1 to this bit generates a software event. This bit is read as 0.	W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The written value is not retained. The read value is always 1.

These registers are used to generate software events 0 to 3, which are respectively assigned to event numbers 050h to 053h.

SEG Bit (Software Event Generation)

When 1 is written to this bit after setting the WE bit to 1, a software event is generated.

WE Bit (SEG Bit Write Enable)

This bit controls writing to the SEG bit. Set this bit to 1 before writing to the SEG bit.

WI Bit (ELSEGR Register Write Disable)

This bit controls writing to this register. Writing to this register is possible only when this register is written in 8-bit units with this bit set to 0.

21.2.7 Event Link Setting Register n (ELSRn) (n = 0 to 26, 46, 47, 50 to 52, 73 to 78, 88 to 91)

Address(es): ELC.ELSR0 8780 1100h, ELC.ELSR1 8780 1104h, ELC.ELSR2 8780 1108h, ELC.ELSR3 8780 110Ch, ELC.ELSR4 8780 1110h, ELC.ELSR5 8780 1114h, ELC.ELSR6 8780 1118h, ELC.ELSR7 8780 111Ch, ELC.ELSR8 8780 1120h, ELC.ELSR9 8780 1124h, ELC.ELSR10 8780 1128h, ELC.ELSR11 8780 112Ch, ELC.ELSR12 8780 1130h, ELC.ELSR13 8780 1134h, ELC.ELSR14 8780 1138h, ELC.ELSR15 8780 113Ch, ELC.ELSR16 8780 1140h, ELC.ELSR17 8780 1144h, ELC.ELSR18 8780 1148h, ELC.ELSR19 8780 114Ch, ELC.ELSR20 8780 1150h, ELC.ELSR21 8780 1154h, ELC.ELSR22 8780 1158h, ELC.ELSR23 8780 115Ch, ELC.ELSR24 8780 1160h, ELC.ELSR25 8780 1164h, ELC.ELSR26 8780 1168h, ELC.ELSR46 8780 11B8h, ELC.ELSR47 8780 11BCh, ELC.ELSR50 8780 11C8h, ELC.ELSR51 8780 11CCh, ELC.ELSR52 8780 11D0h, ELC.ELSR73 8780 1224h, ELC.ELSR74 8780 1228h, ELC.ELSR75 8780 122Ch, ELC.ELSR76 8780 1230h, ELC.ELSR77 8780 1234h, ELC.ELSR78 8780 1238h, ELC.ELSR88 8780 1260h, ELC.ELSR89 8780 1264h, ELC.ELSR90 8780 1268h, ELC.ELSR91 8780 126Ch,
ELC_NS.ELSR0 0780 1100h, ELC_NS.ELSR1 0780 1104h, ELC_NS.ELSR2 0780 1108h, ELC_NS.ELSR3 0780 110Ch, ELC_NS.ELSR4 0780 1110h, ELC_NS.ELSR5 0780 1114h, ELC_NS.ELSR6 0780 1118h, ELC_NS.ELSR7 0780 111Ch, ELC_NS.ELSR8 0780 1120h, ELC_NS.ELSR9 0780 1124h, ELC_NS.ELSR10 0780 1128h, ELC_NS.ELSR11 0780 112Ch, ELC_NS.ELSR12 0780 1130h, ELC_NS.ELSR13 0780 1134h, ELC_NS.ELSR14 0780 1138h, ELC_NS.ELSR15 0780 113Ch, ELC_NS.ELSR16 0780 1140h, ELC_NS.ELSR17 0780 1144h, ELC_NS.ELSR18 0780 1148h, ELC_NS.ELSR19 0780 114Ch, ELC_NS.ELSR20 0780 1150h, ELC_NS.ELSR21 0780 1154h, ELC_NS.ELSR22 0780 1158h, ELC_NS.ELSR23 0780 115Ch, ELC_NS.ELSR24 0780 1160h, ELC_NS.ELSR25 0780 1164h, ELC_NS.ELSR26 0780 1168h, ELC_NS.ELSR46 0780 11B8h, ELC_NS.ELSR47 0780 11BCh, ELC_NS.ELSR50 0780 11C8h, ELC_NS.ELSR51 0780 11CCh, ELC_NS.ELSR52 0780 11D0h, ELC_NS.ELSR73 0780 1224h, ELC_NS.ELSR74 0780 1228h, ELC_NS.ELSR75 0780 122Ch, ELC_NS.ELSR76 0780 1230h, ELC_NS.ELSR77 0780 1234h, ELC_NS.ELSR78 0780 1238h, ELC_NS.ELSR88 0780 1260h, ELC_NS.ELSR89 0780 1264h, ELC_NS.ELSR90 0780 1268h, ELC_NS.ELSR91 0780 126Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ELS[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	ELS[9:0]	Event Link Select	000h: Event signal output to the corresponding peripheral module is disabled. 021h to 33Bh: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, the behavior changes depending on the trusted event route setting.
When the trusted event route is disabled, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
When the trusted event route is enabled, it can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 21.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 21.3 shows the correspondence between values set in the ELSRn register and event signals.

Table 21.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR0	CMT1
ELSR1	CMTW0
ELSR2	TMR0
ELSR3	TMR1
ELSR4	TMR2
ELSR5	TMR3
ELSR6	GPTW event source A (common to all channels)
ELSR7	GPTW event source B (common to all channels)
ELSR8	GPTW event source C (common to all channels)
ELSR9	GPTW event source D (common to all channels)
ELSR10	GPTW event source E (common to all channels)
ELSR11	GPTW event source F (common to all channels)
ELSR12	GPTW event source G (common to all channels)
ELSR13	GPTW event source H (common to all channels)
ELSR14	Port B
ELSR15	Port C
ELSR16	Port D
ELSR17	Port E
ELSR18	ADC trigger 0 (common to all units)
ELSR19	ADC trigger 1 (common to all units)
ELSR20	ADC trigger 2 (common to all units)
ELSR21	ADC trigger 3 (common to all units)
ELSR22	ADC trigger 4 (common to all units)
ELSR23	ADC trigger 5 (common to all units)
ELSR24	ADC trigger 6 (common to all units)
ELSR25	ADC trigger 7 (common to all units)
ELSR26	ADC trigger 8 (common to all units)
ELSR46	DAC0
ELSR47	DAC1
ELSR50	RI3C0
ELSR51	GPTP (CAPTURE0)
ELSR52	GPTP (CAPTURE1)
ELSR73	PCIF0 (ELC_IN)
ELSR74	PCIF0 (PEK_IN)
ELSR75	PCIF0 (VLY_IN)
ELSR76	PCIF1 (ELC_IN)
ELSR77	PCIF1 (PEK_IN)
ELSR78	PCIF1 (VLY_IN)
ELSR88	ICU (Interrupt 1)*1
ELSR89	ICU (Interrupt 2)*1
ELSR90	ICU (Interrupt 3)*1
ELSR91	ICU (Interrupt 4)*1

Note 1. Specify an event number from among 050h to 053h, and 313h. Do not set other values.

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (1/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
021h	DMA controller	DMAC0 transfer end
022h		DMAC1 transfer end
023h		DMAC2 transfer end
024h		DMAC3 transfer end
025h		DMAC4 transfer end
026h		DMAC5 transfer end
027h		DMAC6 transfer end
028h		DMAC7 transfer end
031h	Data transfer controller	DTC transfer end
033h	Voltage detection circuit	PVD1 voltage detection
034h		PVD2 voltage detection
038h	Clock generation circuit	Oscillation stop detection with the main clock
04Ch	Independent watchdog timer	IWDT underflow or refresh error
04Dh	Watchdog timer	WDT underflow or refresh error
04Fh	Realtime clock	RTC periodic event (select 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 seconds)
050h	Event link controller	Software event 0
051h		Software event 1
052h		Software event 2
053h		Software event 3
054h	I/O ports	Port B event
055h		Port C event
056h		Port D event
057h		Port E event
058h	Compare match timer	CMT1 compare match
059h	Compare match timer W	CMTW0 compare match
05Ah	8-bit timers	TMR0 compare match A
05Bh		TMR0 compare match B
05Ch		TMR0 overflow
05Dh		TMR1 compare match A
05Eh		TMR1 compare match B
05Fh		TMR1 overflow
060h		TMR2 compare match A
061h		TMR2 compare match B
062h		TMR2 overflow
063h		TMR3 compare match A
064h		TMR3 compare match B
065h		TMR3 overflow

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (2/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
0C9h	General PWM timer	GPTW (OPS) U-/V-/W-phase input edge detected
0CAh		GPTW0 compare match A
0CBh		GPTW0 compare match B
0CCh		GPTW0 compare match C
0CDh		GPTW0 compare match D
0CEh		GPTW0 compare match E
0CFh		GPTW0 compare match F
0D0h		GPTW0 overflow
0D1h		GPTW0 underflow
0D2h		GPTW0 cycle count end
0D4h		GPTW0 A/D converter start request A
0D5h		GPTW0 A/D converter start request B
0D6h		GPTW1 compare match A
0D7h		GPTW1 compare match B
0D8h		GPTW1 compare match C
0D9h		GPTW1 compare match D
0DAh		GPTW1 compare match E
0DBh		GPTW1 compare match F
0DCh		GPTW1 overflow
0DDh		GPTW1 underflow
0DEh		GPTW1 cycle count end
0E0h		GPTW1 A/D converter start request A
0E1h		GPTW1 A/D converter start request B
0E2h		GPTW2 compare match A
0E3h		GPTW2 compare match B
0E4h		GPTW2 compare match C
0E5h		GPTW2 compare match D
0E6h		GPTW2 compare match E
0E7h		GPTW2 compare match F
0E8h		GPTW2 overflow
0E9h		GPTW2 underflow
0EAh		GPTW2 cycle count end
0ECh		GPTW2 A/D converter start request A
0EDh		GPTW2 A/D converter start request B
0EEh		GPTW3 compare match A
0EFh		GPTW3 compare match B
0F0h		GPTW3 compare match C
0F1h		GPTW3 compare match D
0F2h		GPTW3 compare match E
0F3h		GPTW3 compare match F
0F4h		GPTW3 overflow
0F5h		GPTW3 underflow
0F6h		GPTW3 cycle count end
0F8h		GPTW3 A/D converter start request A
0F9h		GPTW3 A/D converter start request B

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (3/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
0FAh	General PWM timer	GPTW4 compare match A
0FBh		GPTW4 compare match B
0FCh		GPTW4 compare match C
0FDh		GPTW4 compare match D
0FEh		GPTW4 compare match E
0FFh		GPTW4 compare match F
100h		GPTW4 overflow
101h		GPTW4 underflow
102h		GPTW4 cycle count end
104h		GPTW4 A/D converter start request A
105h		GPTW4 A/D converter start request B
106h		GPTW5 compare match A
107h		GPTW5 compare match B
108h		GPTW5 compare match C
109h		GPTW5 compare match D
10Ah		GPTW5 compare match E
10Bh		GPTW5 compare match F
10Ch		GPTW5 overflow
10Dh		GPTW5 underflow
10Eh		GPTW5 cycle count end
110h		GPTW5 A/D converter start request A
111h		GPTW5 A/D converter start request B
112h		GPTW6 compare match A
113h		GPTW6 compare match B
114h		GPTW6 compare match C
115h		GPTW6 compare match D
116h		GPTW6 compare match E
117h		GPTW6 compare match F
118h		GPTW6 overflow
119h		GPTW6 underflow
11Ah		GPTW6 cycle count end
11Ch		GPTW6 A/D converter start request A
11Dh		GPTW6 A/D converter start request B
11Eh		GPTW7 compare match A
11Fh		GPTW7 compare match B
120h		GPTW7 compare match C
121h		GPTW7 compare match D
122h		GPTW7 compare match E
123h		GPTW7 compare match F
124h		GPTW7 overflow
125h		GPTW7 underflow
126h		GPTW7 cycle count end
128h		GPTW7 A/D converter start request A
129h		GPTW7 A/D converter start request B

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (4/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
12Ah	General PWM timer	GPTW8 compare match A
12Bh		GPTW8 compare match B
12Ch		GPTW8 compare match C
12Dh		GPTW8 compare match D
12Eh		GPTW8 compare match E
12Fh		GPTW8 compare match F
130h		GPTW8 overflow
131h		GPTW8 underflow
132h		GPTW8 cycle count end
134h		GPTW8 A/D converter start request A
135h		GPTW8 A/D converter start request B
136h		GPTW9 compare match A
137h		GPTW9 compare match B
138h		GPTW9 compare match C
139h		GPTW9 compare match D
13Ah		GPTW9 compare match E
13Bh		GPTW9 compare match F
13Ch		GPTW9 overflow
13Dh		GPTW9 underflow
13Eh		GPTW9 cycle count end
140h		GPTW9 A/D converter start request A
141h		GPTW9 A/D converter start request B
142h		GPTW10 compare match A
143h		GPTW10 compare match B
144h		GPTW10 compare match C
145h		GPTW10 compare match D
146h		GPTW10 compare match E
147h		GPTW10 compare match F
148h		GPTW10 overflow
149h		GPTW10 underflow
14Ah		GPTW10 cycle count end
14Ch		GPTW10 A/D converter start request A
14Dh		GPTW10 A/D converter start request B
14Eh		GPTW11 compare match A
14Fh		GPTW11 compare match B
150h		GPTW11 compare match C
151h		GPTW11 compare match D
152h		GPTW11 compare match E
153h		GPTW11 compare match F
154h		GPTW11 overflow
155h		GPTW11 underflow
156h		GPTW11 cycle count end
158h		GPTW11 A/D converter start request A
159h		GPTW11 A/D converter start request B

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (5/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
15Ah	General PWM timer	GPTW12 compare match A
15Bh		GPTW12 compare match B
15Ch		GPTW12 compare match C
15Dh		GPTW12 compare match D
15Eh		GPTW12 compare match E
15Fh		GPTW12 compare match F
160h		GPTW12 overflow
161h		GPTW12 underflow
162h		GPTW12 cycle count end
164h		GPTW12 A/D converter start request A
165h		GPTW12 A/D converter start request B
166h		GPTW13 compare match A
167h		GPTW13 compare match B
168h		GPTW13 compare match C
169h		GPTW13 compare match D
16Ah		GPTW13 compare match E
16Bh		GPTW13 compare match F
16Ch		GPTW13 overflow
16Dh		GPTW13 underflow
16Eh		GPTW13 cycle count end
170h		GPTW13 A/D converter start request A
171h		GPTW13 A/D converter start request B
172h		GPTW14 compare match A
173h		GPTW14 compare match B
174h		GPTW14 compare match C
175h		GPTW14 compare match D
176h		GPTW14 compare match E
177h		GPTW14 compare match F
178h		GPTW14 overflow
179h		GPTW14 underflow
17Ah		GPTW14 cycle count end
17Ch		GPTW14 A/D converter start request A
17Dh		GPTW14 A/D converter start request B
17Eh		GPTW15 compare match A
17Fh		GPTW15 compare match B
180h		GPTW15 compare match C
181h		GPTW15 compare match D
182h		GPTW15 compare match E
183h		GPTW15 compare match F
184h		GPTW15 overflow
185h		GPTW15 underflow
186h		GPTW15 cycle count end
188h		GPTW15 A/D converter start request A
189h		GPTW15 A/D converter start request B

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (6/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
259h	General PTP timer	PTP pulse output 0
25Ah		PTP pulse output 1
25Dh		gPTP media clock recovery 0 (GPTP_MATCH0)
25Eh		gPTP media clock recovery 1 (GPTP_MATCH1)
26Dh	Serial communication interface	RSCI0 receive data full
26Eh		RSCI0 transmit data empty
26Fh		RSCI0 transmit end
270h		RSCI0 error (receive error or error signal detection)
271h		RSCI0 effective edge detected
272h		RSCI0 receive data matched
273h		RSCI0 receive data unmatched
274h		RSCI1 receive data full
275h		RSCI1 transmit data empty
276h		RSCI1 transmit end
277h		RSCI1 error (receive error or error signal detection)
278h		RSCI1 effective edge detected
279h		RSCI1 receive data matched
27Ah		RSCI1 receive data unmatched
27Bh		RSCI2 receive data full
27Ch		RSCI2 transmit data empty
27Dh		RSCI2 transmit end
27Eh		RSCI2 error (receive error or error signal detection)
27Fh		RSCI2 effective edge detected
280h		RSCI2 receive data matched
281h		RSCI2 receive data unmatched
282h		RSCI3 receive data full
283h		RSCI3 transmit data empty
284h		RSCI3 transmit end
285h		RSCI3 error (receive error or error signal detection)
286h		RSCI3 effective edge detected
287h		RSCI3 receive data matched
288h		RSCI3 receive data unmatched
289h		RSCI4 receive data full
28Ah		RSCI4 transmit data empty
28Bh		RSCI4 transmit end
28Ch		RSCI4 error (receive error or error signal detection)
28Dh		RSCI4 effective edge detected
28Eh		RSCI4 receive data matched
28Fh		RSCI4 receive data unmatched

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (7/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
290h	Serial communication interface	RSCI5 receive data full
291h		RSCI5 transmit data empty
292h		RSCI5 transmit end
293h		RSCI5 error (receive error or error signal detection)
294h		RSCI5 effective edge detected
295h		RSCI5 receive data matched
296h		RSCI5 receive data unmatched
297h		RSCI6 receive data full
298h		RSCI6 transmit data empty
299h		RSCI6 transmit end
29Ah		RSCI6 error (receive error or error signal detection)
29Bh		RSCI6 effective edge detected
29Ch		RSCI6 receive data matched
29Dh		RSCI6 receive data unmatched
29Eh		RSCI7 receive data full
29Fh		RSCI7 transmit data empty
2A0h		RSCI7 transmit end
2A1h		RSCI7 error (receive error or error signal detection)
2A2h		RSCI7 effective edge detected
2A3h		RSCI7 receive data matched
2A4h		RSCI7 receive data unmatched
2A5h		RSCI8 receive data full
2A6h		RSCI8 transmit data empty
2A7h		RSCI8 transmit end
2A8h		RSCI8 error (receive error or error signal detection)
2A9h		RSCI8 effective edge detected
2AAh		RSCI8 receive data matched
2ABh		RSCI8 receive data unmatched
2ACh		RSCI9 receive data full
2ADh		RSCI9 transmit data empty
2AEh		RSCI9 transmit end
2AFh		RSCI9 error (receive error or error signal detection)
2B0h		RSCI9 effective edge detected
2B1h		RSCI9 receive data matched
2B2h		RSCI9 receive data unmatched
2B3h		RSCI10 receive data full
2B4h		RSCI10 transmit data empty
2B5h		RSCI10 transmit end
2B6h		RSCI10 error (receive error or error signal detection)
2B7h		RSCI10 effective edge detected
2B8h		RSCI10 receive data matched
2B9h		RSCI10 receive data unmatched

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (8/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
2BAh	Serial communication interface	RSCI11 receive data full
2BBh		RSCI11 transmit data empty
2BCh		RSCI11 transmit end
2BDh		RSCI11 error (receive error or error signal detection)
2BEh		RSCI11 effective edge detected
2BFh		RSCI11 receive data matched
2C0h		RSCI11 receive data unmatched
2C1h		RSCI12 receive data full
2C2h		RSCI12 transmit data empty
2C3h		RSCI12 transmit end
2C4h		RSCI12 error (receive error or error signal detection)
2C5h		RSCI12 effective edge detected
2C6h		RSCI12 receive data matched
2C7h		RSCI12 receive data unmatched
2C8h	Serial peripheral interface	RSPI0 receive buffer full
2C9h		RSPI0 transmit buffer empty
2CAh		RSPI0 idle
2CBh		RSPI0 error (mode fault, overrun, underrun, or parity error)
2CCh		RSPI0 communication end
2CDh		RSPI1 receive buffer full
2CEh		RSPI1 transmit buffer empty
2CFh		RSPI1 idle
2D0h		RSPI1 error (mode fault, overrun, underrun, or parity error)
2D1h		RSPI1 communication end
2D2h		RSPI2 receive buffer full
2D3h		RSPI2 transmit buffer empty
2D4h		RSPI2 idle
2D5h		RSPI2 error (mode fault, overrun, underrun, or parity error)
2D6h		RSPI2 communication end
2DBh	High-speed I^2C-bus interface	RIICHS1 receive data full
2DCh		RIICHS1 transmit data empty
2DDh		RIICHS1 transmit end
2DEh		RIICHS1 communication error or communication event
2DFh		RIICHS2 receive data full
2E0h		RIICHS2 transmit data empty
2E1h		RIICHS2 transmit end
2E2h		RIICHS2 communication error or communication event
2E3h	I3C bus interface	RI3C0 response queue full
2E4h		RI3C0 command queue empty
2E5h		RI3C0 IBI queue empty/full
2E6h		RI3C0 receive data full
2E7h		RI3C0 transmit data empty
2E8h		RI3C0 receive status queue full
2E9h		RI3C0 high priority response queue full
2EAh		RI3C0 high priority command queue empty
2EBh		RI3C0 high priority receive data full

Table 21.3 Correspondence between Values Set in ELSRn.ELS[9:0] Bits and Event Signals (9/9)

ELS[9:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
2ECh	I3C bus interface	RI3C0 high priority transmit data empty
2EDh		RI3C0 transmit end (in I^2C mode)
2EEh		RI3C0 communication error or communication event
2EFh		RI3C0 synchronous timing
2F0h		RI3C0 C_REF counter overflow
2F1h		RI3C0 C_REF capture
2F2h		RI3C0 additional bus event
313h	Data Operation Circuit	DOC data operation condition met
314h	A/D converter	Group 0 A/D scan end
315h		Group 1 A/D scan end
316h		Group 2 A/D scan end
317h		Group 3 A/D scan end
318h		Group 4 A/D scan end
319h		Group 5 A/D scan end
31Ah		Group 6 A/D scan end
31Bh		Group 7 A/D scan end
31Ch		Group 8 A/D scan end
31Dh		Group 9 A/D scan end
31Eh		Group 10 A/D scan end
31Fh		Group 11 A/D scan end
324h		Composite condition compare match 0
325h		Composite condition compare match 1
326h		Composite condition compare match 2
327h		Composite condition compare match 3
328h		Composite condition compare match 4
329h		Composite condition compare match 5
32Ah		Composite condition compare match 6
32Bh		Composite condition compare match 7
334h		Composite condition compare unmatch 0
335h		Composite condition compare unmatch 1
336h		Composite condition compare unmatch 2
337h		Composite condition compare unmatch 3
338h		Composite condition compare unmatch 4
339h		Composite condition compare unmatch 5
33Ah		Composite condition compare unmatch 6
33Bh		Composite condition compare unmatch 7
Settings other than above are prohibited.		

21.2.8 Event Link Option Setting Register A (ELOPA)

Address(es): ELC.ELOPA 8780 1090h, ELC_NS.ELOPA 0780 1090h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMT1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The ELOPA register specifies the operations of CMT1 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.9 Event Link Option Setting Register B (ELOPB)

Address(es): ELC.ELOPB 8780 1094h, ELC_NS.ELOPB 0780 1094h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMTW0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMTW0MD[1:0]	CMTW0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The ELOPB register specifies the operation of CMTW0 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.10 Event Link Option Setting Register C (ELOPC)

Address(es): ELC.ELOPC 8780 1098h, ELC_NS.ELOPC 0780 1098h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	TMR1MD[1:0]		TMR0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b3, b2	TMR1MD[1:0]	TMR1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The ELOPC register specifies the operations of TMR0 and TMR1 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.11 Event Link Option Setting Register D (ELOPD)

Address(es): ELC.ELOPD 8780 109Ch, ELC_NS.ELOPD 0780 109Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	TMR3MD[1:0]		TMR2MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR2MD[1:0]	TMR2 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b3, b2	TMR3MD[1:0]	TMR3 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The ELOPD register specifies the operations of TMR2 and TMR3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.3 Operation

21.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the interrupt enable bits to enable/disable these interrupt requests. When an interrupt request is generated in a peripheral module, the corresponding interrupt request status flag becomes 1. If the corresponding interrupt request is enabled then, the interrupt is requested to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, interconnects (links) peripheral modules, and then, makes peripheral modules perform direct interlinked operation among them without using software. Event signals can be output regardless of the setting of the corresponding interrupt enable bit.

Figure 21.2 shows the relation between the interrupt handling and ELC.

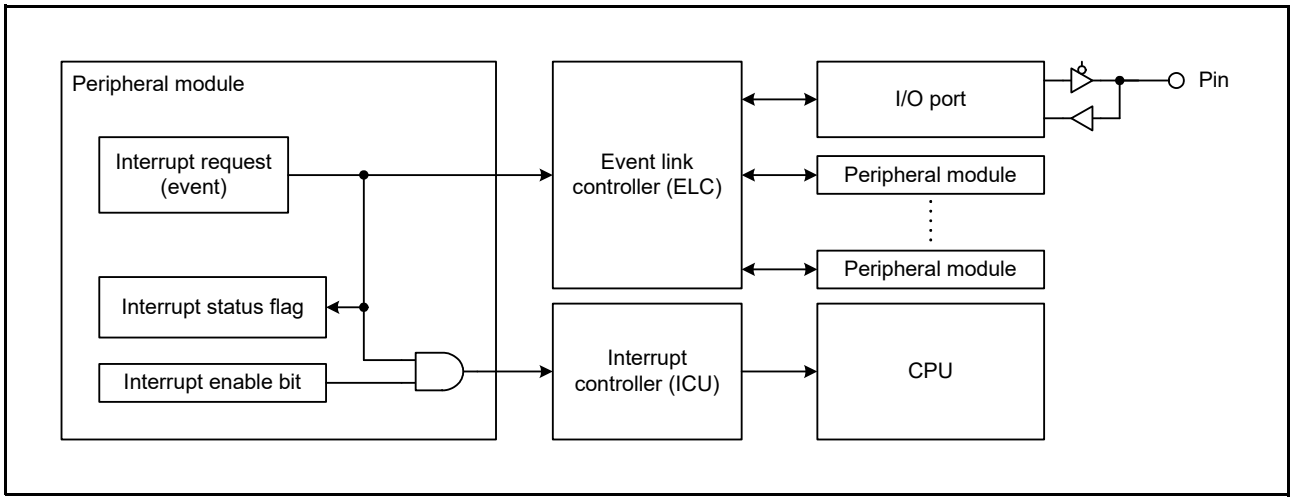


Figure 21.2 Relation between Interrupt Handling and ELC

21.3.2 Event Linkage

When events are specified in the ELSRn registers, the corresponding peripheral modules can be operated at generation of the specified events. A single peripheral module can link only with a single event. Set the ELSRn register after completing the initialization of the peripheral module to operate by an event. Table 21.4 lists the operations of peripheral modules when an event signal is input.

Table 21.4 Operations of Peripheral Modules When Event Signal is Input

Peripheral Module	Operations When Event Signal is Input
CMT CMTW TMR	The following operations can be selected by setting the ELOPA to ELOPD registers: • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events.
GPTW	The following operations can be selected by setting registers in GPTW: • Starts counting when an event signal is input. • Stops counting when an event signal is input. • Clears counter when an event signal is input. • Switches the direction of counting when an event signal is input. • Increments counter when an event signal is input. • Decrements counter when an event signal is input. • Performs input-capture operation when an event signal is input.
RI3C	Captures the T_C1 counter when an event signal is input.
GPTP	Generates the media clock capture event when an event signal is input.
PCIF	The following operations can be selected by setting registers in PCIF: • Event input signal for carrier period • Clears the internal counter when an event signal is input. • Z-phase output signal is generated when an event signal is input.
A/D converter	Starts A/D conversion when an event signal is input.
D/A converter	Starts D/A conversion when an event signal is input.
I/O ports	• Changes the signal levels of the output pins according to the settings of the PORTx.EORR and PORTx.EOSR registers. • Latches the signal levels of the input pins into the PORTx.EIDR register. • Ports B to E can be controlled by the ELC.
Interrupt controller	Request an interrupt to the CPU, starts DMA transfer, or starts DTC transfer when an event signal is input.

21.3.3 Operation of Peripheral Timer Modules When Event Signal is Input

For the timer modules except GPTW, set the ELOPA to ELOPD register to specify the operation for when an event signal is input.

(1) Count Start Operation

When an event signal is input, the timer starts counting and the count start bit*1 in each timer control register becomes 1. An event signal that is input while the count start bit is 1 is ignored.

(2) Count Restart Operation

When an event signal is input, the timer counter is cleared. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

(3) Event Counter Operation

Event signal is selected as the timer count source. When an event signal is input, the timer counter is incremented.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

21.3.4 Operation of GPTW When Event Signal is Input

Eight event signals specified by the ELSR6 to ELSR13 registers are connected to every channel of GPTW as GPTW event sources A to H. To set the operation of GPTW when inputting an event signal, enable any of the event sources A to H by a corresponding bit in registers listed in Table 21.5.

Table 21.5 Operations When Event Signal is Input and Corresponding Source Select Registers

Operations on Event	Register Symbol	Register Name
Count start	GTSSR	General PWM Timer Start Source Select Register
Count stop	GTPSR	General PWM Timer Stop Source Select Register
Counter clear	GTCSR	General PWM Timer Clear Source Select Register
Count switch	GTSWSR	General PWM Timer Switch Source Select Register
Count-up	GTUPSR	General PWM Timer Count-Up Source Select Register
Count-down	GTDNSR	General PWM Timer Count-Down Source Select Register
Input capture A	GTICASR	General PWM Timer Input Capture Source Select Register A
Input capture B	GTICBSR	General PWM Timer Input Capture Source Select Register B

21.3.5 Operation of A/D and D/A Converters When Event Signal is Input

When any of the event signal specified by the ELSR18 to ELSR26 registers is input, the A/D converter starts A/D conversion.
When an event signal specified by the ELSR46 or ELSR47 register is input, the D/A converter starts D/A conversion.
Refer to the A/D converter and D/A converter sections for details.

21.3.6 Example of Procedure for Linking Events

The following describes the procedure for linking events.

(1) Initialize the peripheral module (destination) that operates based on an event signal.
(2) Set the number of the event signal to the ELSRn register corresponding to the destination peripheral module.
(3) To link an event to a timer module, set any of the ELOPA to ELOPD registers corresponding to the timer as required.
(4) Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
(5) Set the operation of the peripheral module (source) from which an event signal is output, and activate the module. The preset operation of the destination peripheral module is started by the event signal that is output from the source peripheral module.
(6) To stop event linkage of independent peripheral module, set 00h to the ELSRn register corresponding to the peripheral module. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: If event signal output from the RTC is to be used, make the ELC settings after the RTC settings (initialization, time setting, etc.). Unintended events may be generated if RTC settings are made after the ELC settings.
Note: When using event signal output from the PVD, set the PVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the PVD.

21.4 Usage Notes

21.4.1 Setting ELSR88, ELSR89, ELSR90, and ELSR91 Registers

Specify an event number from among 050h to 053h, and 313h. Do not set the value other than preceding numbers.

21.4.2 Linking DMA/DTC Transfer End Signal as Event

When linking the DMA/DTC transfer end signal as an event signal, do not set the same peripheral module as the DMA/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMA/DTC transfer to the peripheral module is completed.

21.4.3 Clock Settings

To link events, make sure that the ELC and the related peripheral modules are in an operational condition. The peripheral modules cannot operate if they are in the module stop state or in mode which they stop (software standby mode or deep software standby mode).

21.4.4 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register C (MSTPCRC). After reset is released, the ELC function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

21.4.5 Internal Propagation Delay Times in the ELC

As shown in Figure 21.3, an event signal generated by a source peripheral module is propagated to a destination peripheral module via the ELC. The ELC includes synchronization (clock domain crossing, or CDC) circuits to ensure the transmission of event signals between modules operating with different clocks, and these circuits lead to propagation delays. Table 21.6 lists the internal propagation delay times of the ELC. Table 21.7 lists the clock domains of the peripheral modules.

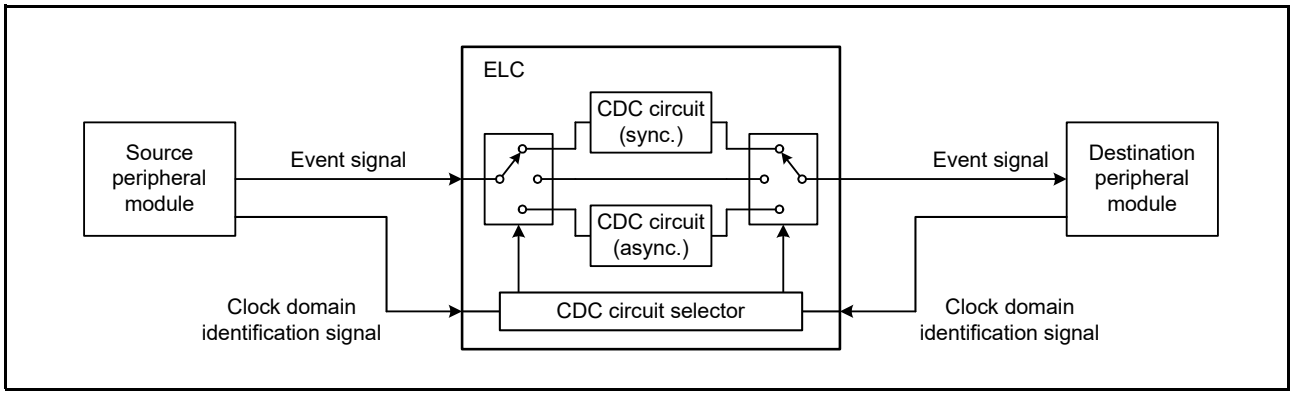


Figure 21.3 Event Signal Synchronization Circuits

Table 21.6 Internal Propagation Delay Times

Operating Clock (Clock Domain)		
Source Peripheral Module	**Destination Peripheral Module**	**Internal Propagation Delay Time**
PCLKA	PCLKA	0 cycles
	PCLKB or PCLKD	1 to 2 cycles of the slower clock of those for the source and destination modules
	Other than PCLK	5 PCLKA cycles + 4 cycles of the operating clock for the destination module
PCLKB	PCLKA or PCLKD	1 to 2 cycles of the slower clock of those for the source and destination modules
	PCLKB	0 cycles
	Other than PCLK	5 PCLKB cycles + 4 cycles of the operating clock for the destination module
PCLKD	PCLKA or PCLKB	1 to 2 cycles of the slower clock of those for the source and destination modules
	PCLKD	0 cycles
	Other than PCLK	5 PCLKD cycles + 4 cycles of the operating clock for the destination module
GPTCLK	PCLK	5 GPTCLK cycles + 4 PCLK cycles
	GPTCLK	0 cycles
	Other than PCLK and GPTCLK	5 GPTCLK cycles + 4 cycles of the operating clock for the destination module

Table 21.7 Clock Domains of Peripheral Modules

Peripheral Module		Clock Domain
Source	RSCI, RSPI, RI3C, ADC, DOC, GPTP, or RIICHS	PCLKA
	DMAC, DTC, PVD, Clock generation circuit, IWDT, WDT, RTC, ELC, I/O ports, CMT, CMTW, or TMR	PCLKB
	GPTW	GPTCLK or PCLKD
Destination	RI3C or GPTP	PCLKA
	I/O ports, DAC, CMT, CMTW, TMR, or ICU	PCLKB
	GPTW	GPTCLK or PCLKD
	ADC	ADCCK, GPTCLK, or PCLKA
	PCIF	GPTCLK or PCLKB

22. I/O Ports

22.1 Overview

The pins of an I/O port may function as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O pins, pins for the ELC port group function, or bus control pins. All pins except P302 (the TDO pin of the JTAG port by default) operate as input pins immediately after a reset and the I/O directions of the pins are switched by register settings. Whether each pin operates as an I/O port or a pin for a peripheral module can be specified by the registers corresponding to the pins. The configuration of the I/O ports depends on the package. Table 22.1 lists the specifications of the I/O ports and Table 22.2 lists the port functions.

Table 22.1 Specifications of I/O Ports

Port Symbol	Package 320-Pin	Number of Pins	Package 224-Pin	Number of Pins	Package 176-Pin	Number of Pins	Package 144-Pin	Number of Pins	Package 100-Pin	Number of Pins
PORT0	P000 to P015	16	P000 to P007	8	P000 to P007	8	P000 to P007	8	P001 to P007	7
PORT1	P100 to P115	16	P100 to P115	16	P100 to P107	8	P100 to P107	8	P101 to P107	7
PORT2	P200 to P215	16	P200 to P215	16	P200 to P207	8	P200 to P207	8	P200, P201	2
PORT3	P300 to P312, P315	14	P300 to P312, P315	14	P300 to P312, P315	14	P300 to P312, P315	14	P300 to P312, P315	14
PORT4	P400 to P415	16	P400 to P407, P414, P415	10	P400 to P407, P414, P415	10	P400 to P405, P414, P415	8	P400, P402, P404, P414	4
PORT5	P500 to P515	16	P500 to P507	8	P500 to P503	4	—	0	—	0
PORT6	P600 to P607	8	P600 to P607	8	P600 to P607	8	P600 to P605	6	P600, P601	2
PORT7	P700 to P715	16	P700 to P707	8	P700 to P707	8	P700 to P702	3	—	0
PORT8	P800 to P807	8	P800 to P807	8	P800 to P805	6	P800 to P803	4	—	0
PORT9	P900 to P907	8	P900 to P907	8	P900 to P903	4	P900, P901	2	—	0
PORTA	PA00 to PA07	8	PA00 to PA07	8	PA00 to PA07	8	PA00 to PA07	8	PA00 to PA07	8
PORTB	PB00 to PB07	8	PB00 to PB07	8	PB00 to PB07	8	PB00 to PB07	8	PB00 to PB06	7
PORTC	PC00 to PC07	8	PC00 to PC07	8	PC00 to PC07	8	PC00 to PC07	8	PC00 to PC06	7
PORTD	PD00 to PD07	8	PD00 to PD07	8	PD00 to PD07	8	PD00 to PD07	8	PD00 to PD07	8
PORTE	PE00 to PE07	8	PE00 to PE07	8	PE00 to PE07	8	PE00 to PE07	8	PE00 to PE06	7
PORTF	PF00 to PF09	10	PF00 to PF09	10	PF00, PF01	2	PF00, PF01	2	PF00, PF01	2
PORTG	PG00 to PG15	16	PG00 to PG07	8	PG00 to PG02	3	—	0	—	0
	Total Number of Pins	200	Total Number of Pins	162	Total Number of Pins	123	Total Number of Pins	103	Total Number of Pins	75

Table 22.2 Port Functions (1/2)

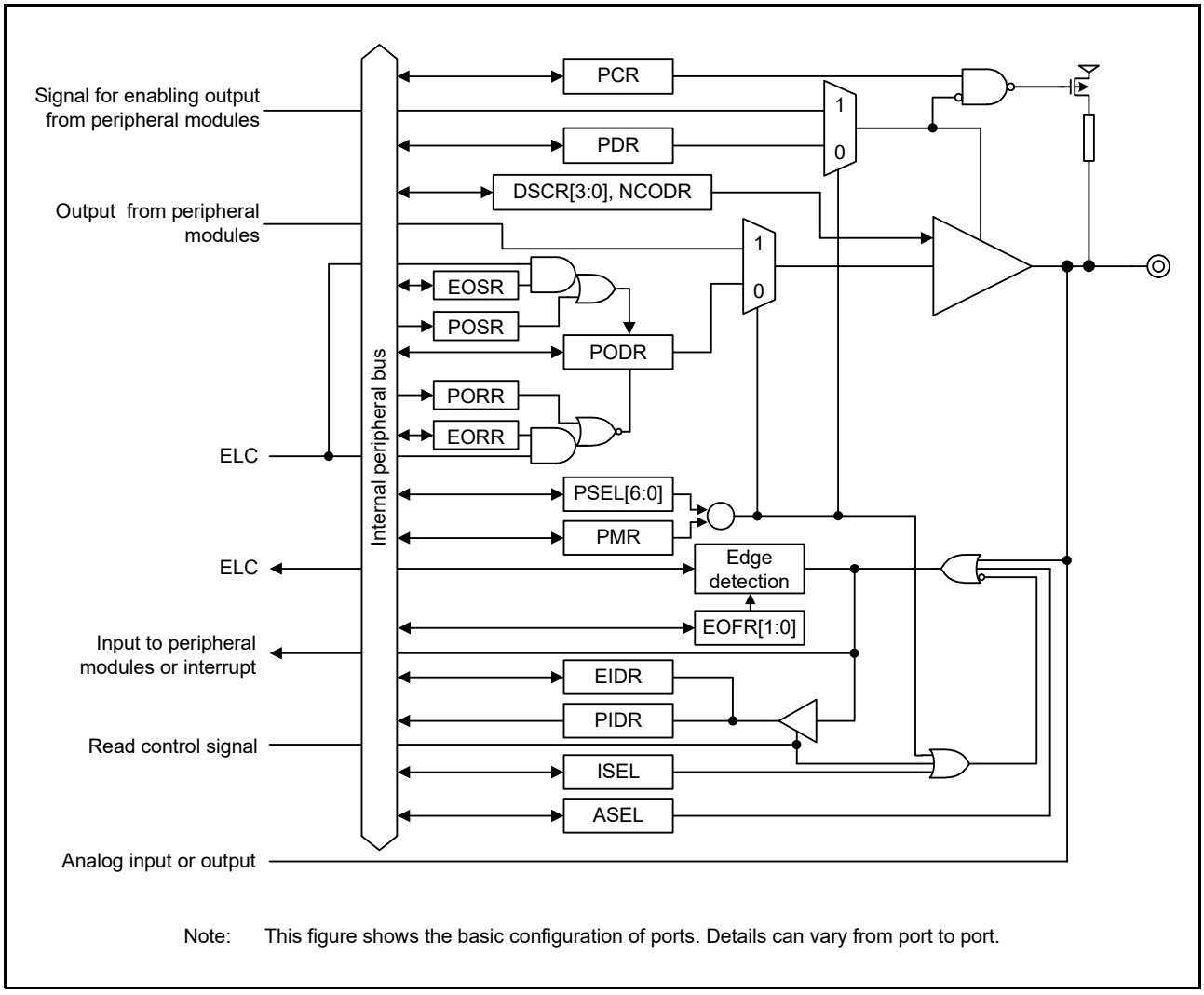
Port Symbol	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching*1	Tolerance	I/O	Power Supply
PORT0	P000 to P015	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORT1	P100, P101	✓	✓	DRV1, DRV2, DRV3, DRV4	5-V tolerant pins	I/O	VCC
	P102, P103	✓	✓	DRV1, DRV2, DRV3, DRV4*2	5-V tolerant pins	I/O	VCC
	P104, P105	✓	✓	DRV1, DRV2, DRV3, DRV4	5-V tolerant pins	I/O	VCC
	P106, P107	✓	✓	DRV1, DRV2, DRV3, DRV4*3	5-V tolerant pins	I/O	VCC
	P108 to P115	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORT2	P200, P201	✓	✓	DRV1, DRV2, DRV3, DRV4*3	5-V tolerant pins	I/O	VCC
	P202 to P207	✓	✓	DRV1, DRV2, DRV3, DRV4	5-V tolerant pins	I/O	VCC
	P208 to P215	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORT3	P300	✓	✓	DRV1, DRV2, DRV3, DRV4*4	—	I/O	VCC
	P301	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P302	✓	✓	DRV1, DRV2, DRV3, DRV4*4	—	I/O	VCC
	P303, P304	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P305	✓	—	—	—	Input	VCC
	P306, P307	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P308, P309	—	—	—	—	Input	VBATT
	P310 to P312	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P315	✓	✓	DRV1, DRV2, DRV3, DRV4*5	—	I/O	VCC
PORT4	P400 to P415	✓	✓	DRV1	—	I/O	AVCC0
PORT5	P500 to P511	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P512 to P515	✓	✓	DRV1, DRV2, DRV3, DRV4, DRV5, DRV6, DRV7	—	I/O	VCC
PORT6	P600 to P603	✓	✓	DRV1, DRV2, DRV3, DRV6, DRV8	—	I/O	VCC2
	P604 to P607	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC2
PORT7	P700	✓	✓	DRV1, DRV2, DRV3, DRV7	—	I/O	VCC
	P701 to P707	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P708 to P711	✓	✓	DRV1, DRV2, DRV3, DRV6, DRV8	—	I/O	VCC
	P712 to P715	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORT8	P800	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	P801	✓	✓	DRV1, DRV2, DRV3, DRV7	—	I/O	VCC
	P802 to P807	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORT9	P900 to P907	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC2
PORTA	PA00, PA01	✓	✓	DRV1, DRV2, DRV3, DRV4, DRV5, DRV6, DRV7	—	I/O	VCC
	PA02	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	PA03 to PA07	✓	✓	DRV1, DRV2, DRV3, DRV4, DRV5, DRV6, DRV7	—	I/O	VCC
PORTB	PB00 to PB07	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORTC	PC00	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	PC01 to PC07	✓	✓	DRV1, DRV2, DRV3, DRV4, DRV5, DRV6, DRV7	—	I/O	VCC
PORTD	PD00 to PD06	✓	✓	DRV1, DRV2, DRV3, DRV6, DRV8	—	I/O	VCC2
	PD07	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC2

Table 22.2 Port Functions (2/2)

Port Symbol	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching*1	Tolerance	I/O	Power Supply
PORTE	PE00 to PE06	✓	✓	DRV1, DRV2, DRV3, DRV6, DRV8	—	I/O	VCC
	PE07	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORTF	PF00, PF01	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC_USB
	PF02 to PF09	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
PORTG	PG00 to PG07	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC
	PG08 to PG11	✓	✓	DRV1, DRV2, DRV3, DRV4, DRV5, DRV6, DRV7	—	I/O	VCC
	PG12 to PG15	✓	✓	DRV1, DRV2, DRV3, DRV4	—	I/O	VCC

Note 1. Specifying a non-supported driving ability is prohibited.
Note 2. When an I3C function is selected, the function for switching the driving ability is disabled and the driving ability specialized for the I3C or I^2C is selected.
Note 3. When an I^2C function is selected, the function for switching the driving ability is disabled and the driving ability specialized for the I^2C is selected.
Note 4. When a JTAG function is selected, the function for switching the driving ability is disabled and the driving ability specialized for the JTAG is selected.
Note 5. When a FINE function is selected, the function for switching the driving ability is disabled and the driving ability specialized for the FINE is selected.

22.2 Configuration of I/O Ports



Note: This figure shows the basic configuration of ports. Details can vary from port to port.

Figure 22.1 Configuration of I/O Ports

22.3 Register Descriptions

22.3.1 Port Control Register 1 (PCNTR1)

Address(es): PORT0.PCNTR1 87C0 0000h, PORT1.PCNTR1 87C0 0020h, PORT2.PCNTR1 87C0 0040h, PORT3.PCNTR1 87C0 0060h, PORT4.PCNTR1 87C0 0080h, PORT5.PCNTR1 87C0 00A0h, PORT6.PCNTR1 87C0 00C0h, PORT7.PCNTR1 87C0 00E0h, PORT8.PCNTR1 87C0 0100h, PORT9.PCNTR1 87C0 0120h, PORTA.PCNTR1 87C0 0140h, PORTB.PCNTR1 87C0 0160h, PORTC.PCNTR1 87C0 0180h, PORTD.PCNTR1 87C0 01A0h, PORTE.PCNTR1 87C0 01C0h, PORTF.PCNTR1 87C0 01E0h, PORTG.PCNTR1 87C0 0200h,
PORT0_NS.PCNTR1 07C0 0000h, PORT1_NS.PCNTR1 07C0 0020h, PORT2_NS.PCNTR1 07C0 0040h, PORT3_NS.PCNTR1 07C0 0060h, PORT4_NS.PCNTR1 07C0 0080h, PORT5_NS.PCNTR1 07C0 00A0h, PORT6_NS.PCNTR1 07C0 00C0h, PORT7_NS.PCNTR1 07C0 00E0h, PORT8_NS.PCNTR1 07C0 0100h, PORT9_NS.PCNTR1 07C0 0120h, PORTA_NS.PCNTR1 07C0 0140h, PORTB_NS.PCNTR1 07C0 0160h, PORTC_NS.PCNTR1 07C0 0180h, PORTD_NS.PCNTR1 07C0 01A0h, PORTE_NS.PCNTR1 07C0 01C0h, PORTF_NS.PCNTR1 07C0 01E0h, PORTG_NS.PCNTR1 07C0 0200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDR00	Pm00 I/O Direction Select	0: Input (The pin functions as an input port pin.) 1: Output (The pin functions as an output port pin.)	R/W
b1	PDR01	Pm01 I/O Direction Select		R/W
b2	PDR02	Pm02 I/O Direction Select		R/W
b3	PDR03	Pm03 I/O Direction Select		R/W
b4	PDR04	Pm04 I/O Direction Select		R/W
b5	PDR05	Pm05 I/O Direction Select		R/W
b6	PDR06	Pm06 I/O Direction Select		R/W
b7	PDR07	Pm07 I/O Direction Select		R/W
b8	PDR08	Pm08 I/O Direction Select		R/W
b9	PDR09	Pm09 I/O Direction Select		R/W
b10	PDR10	Pm10 I/O Direction Select		R/W
b11	PDR11	Pm11 I/O Direction Select		R/W
b12	PDR12	Pm12 I/O Direction Select		R/W
b13	PDR13	Pm13 I/O Direction Select		R/W
b14	PDR14	Pm14 I/O Direction Select		R/W
b15	PDR15	Pm15 I/O Direction Select		R/W

Bit	Symbol	Bit Name	Description	R/W
b16	PODR00	Pm00 Output Data	0: Low-level output 1: High-level output	R/W
b17	PODR01	Pm01 Output Data		R/W
b18	PODR02	Pm02 Output Data		R/W
b19	PODR03	Pm03 Output Data		R/W
b20	PODR04	Pm04 Output Data		R/W
b21	PODR05	Pm05 Output Data		R/W
b22	PODR06	Pm06 Output Data		R/W
b23	PODR07	Pm07 Output Data		R/W
b24	PODR08	Pm08 Output Data		R/W
b25	PODR09	Pm09 Output Data		R/W
b26	PODR10	Pm10 Output Data		R/W
b27	PODR11	Pm11 Output Data		R/W
b28	PODR12	Pm12 Output Data		R/W
b29	PODR13	Pm13 Output Data		R/W
b30	PODR14	Pm14 Output Data		R/W
b31	PODR15	Pm15 Output Data		R/W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is readable and writable in 32-bit units and is used to select the input/output direction and output data for the individual pins of the corresponding port when the pins are configured as general I/O port pins.
A bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. When writing to such bits, write 0 to them. As the P305, P308, and P309 pins are input-only, the PDR05, PDR08, PDR09, PODR05, PODR08, and PODR09 bits in the PCNTR1 register for PORT3 are also reserved.

22.3.2 Port Direction Register (PDR)

Address(es): PORT0.PDR 87C0 0000h, PORT1.PDR 87C0 0020h, PORT2.PDR 87C0 0040h, PORT3.PDR 87C0 0060h, PORT4.PDR 87C0 0080h, PORT5.PDR 87C0 00A0h, PORT6.PDR 87C0 00C0h, PORT7.PDR 87C0 00E0h, PORT8.PDR 87C0 0100h, PORT9.PDR 87C0 0120h, PORTA.PDR 87C0 0140h, PORTB.PDR 87C0 0160h, PORTC.PDR 87C0 0180h, PORTD.PDR 87C0 01A0h, PORTE.PDR 87C0 01C0h, PORTF.PDR 87C0 01E0h, PORTG.PDR 87C0 0200h,
PORT0_NS.PDR 07C0 0000h, PORT1_NS.PDR 07C0 0020h, PORT2_NS.PDR 07C0 0040h, PORT3_NS.PDR 07C0 0060h, PORT4_NS.PDR 07C0 0080h, PORT5_NS.PDR 07C0 00A0h, PORT6_NS.PDR 07C0 00C0h, PORT7_NS.PDR 07C0 00E0h, PORT8_NS.PDR 07C0 0100h, PORT9_NS.PDR 07C0 0120h, PORTA_NS.PDR 07C0 0140h, PORTB_NS.PDR 07C0 0160h, PORTC_NS.PDR 07C0 0180h, PORTD_NS.PDR 07C0 01A0h, PORTE_NS.PDR 07C0 01C0h, PORTF_NS.PDR 07C0 01E0h, PORTG_NS.PDR 07C0 0200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDR00	Pm00 I/O Direction Select	0: Input (The pin functions as an input port pin.) 1: Output (The pin functions as an output port pin.)	R/W
b1	PDR01	Pm01 I/O Direction Select		R/W
b2	PDR02	Pm02 I/O Direction Select		R/W
b3	PDR03	Pm03 I/O Direction Select		R/W
b4	PDR04	Pm04 I/O Direction Select		R/W
b5	PDR05	Pm05 I/O Direction Select		R/W
b6	PDR06	Pm06 I/O Direction Select		R/W
b7	PDR07	Pm07 I/O Direction Select		R/W
b8	PDR08	Pm08 I/O Direction Select		R/W
b9	PDR09	Pm09 I/O Direction Select		R/W
b10	PDR10	Pm10 I/O Direction Select		R/W
b11	PDR11	Pm11 I/O Direction Select		R/W
b12	PDR12	Pm12 I/O Direction Select		R/W
b13	PDR13	Pm13 I/O Direction Select		R/W
b14	PDR14	Pm14 I/O Direction Select		R/W
b15	PDR15	Pm15 I/O Direction Select		R/W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is used to select the input/output direction for the individual pins of the corresponding port when the pins are configured as general I/O port pins and makes up the 16 lower-order bits of the PCNTR1 register. This register is accessible in 8- or 16-bit units.

A bit corresponding to a pin that does not exist is reserved. Make settings according to the description in section 22.6.11, Initialization of the Port Direction Register (PDR).

As the P305, P308, and P309 pins are input-only, the PORT3.PDR.PDR05, PORT3.PDR.PDR08, and PORT3.PDR.PDR09 bits are also reserved.

22.3.3 Port Output Data Register (PODR)

Address(es): PORT0.PODR 87C0 0002h, PORT1.PODR 87C0 0022h, PORT2.PODR 87C0 0042h, PORT3.PODR 87C0 0062h, PORT4.PODR 87C0 0082h, PORT5.PODR 87C0 00A2h, PORT6.PODR 87C0 00C2h, PORT7.PODR 87C0 00E2h, PORT8.PODR 87C0 0102h, PORT9.PODR 87C0 0122h, PORTA.PODR 87C0 0142h, PORTB.PODR 87C0 0162h, PORTC.PODR 87C0 0182h, PORTD.PODR 87C0 01A2h, PORTE.PODR 87C0 01C2h, PORTF.PODR 87C0 01E2h, PORTG.PODR 87C0 0202h,
PORT0_NS.PODR 07C0 0002h, PORT1_NS.PODR 07C0 0022h, PORT2_NS.PODR 07C0 0042h, PORT3_NS.PODR 07C0 0062h, PORT4_NS.PODR 07C0 0082h, PORT5_NS.PODR 07C0 00A2h, PORT6_NS.PODR 07C0 00C2h, PORT7_NS.PODR 07C0 00E2h, PORT8_NS.PODR 07C0 0102h, PORT9_NS.PODR 07C0 0122h, PORTA_NS.PODR 07C0 0142h, PORTB_NS.PODR 07C0 0162h, PORTC_NS.PODR 07C0 0182h, PORTD_NS.PODR 07C0 01A2h, PORTE_NS.PODR 07C0 01C2h, PORTF_NS.PODR 07C0 01E2h, PORTG_NS.PODR 07C0 0202h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PODR00	Pm00 Output Data	0: Low-level output 1: High-level output	R/W
b1	PODR01	Pm01 Output Data		R/W
b2	PODR02	Pm02 Output Data		R/W
b3	PODR03	Pm03 Output Data		R/W
b4	PODR04	Pm04 Output Data		R/W
b5	PODR05	Pm05 Output Data		R/W
b6	PODR06	Pm06 Output Data		R/W
b7	PODR07	Pm07 Output Data		R/W
b8	PODR08	Pm08 Output Data		R/W
b9	PODR09	Pm09 Output Data		R/W
b10	PODR10	Pm10 Output Data		R/W
b11	PODR11	Pm11 Output Data		R/W
b12	PODR12	Pm12 Output Data		R/W
b13	PODR13	Pm13 Output Data		R/W
b14	PODR14	Pm14 Output Data		R/W
b15	PODR15	Pm15 Output Data		R/W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is used to select the output data for the individual pins of the corresponding port when the pins are configured as general I/O port pins and makes up the 16 higher-order bits of the PCNTR1 register. This register is accessible in 8- or 16-bit units.

A bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. When writing to such bits, write 0 to them. As the P305, P308, and P309 pins are input-only, the PODR05, PODR08, and PODR09 bits in the PODR register for PORT3 are also reserved.

22.3.4 Port Control Register 2 (PCNTR2)

Address(es): PORT0.PCNTR2 87C0 0004h, PORT1.PCNTR2 87C0 0024h, PORT2.PCNTR2 87C0 0044h, PORT3.PCNTR2 87C0 0064h, PORT4.PCNTR2 87C0 0084h, PORT5.PCNTR2 87C0 00A4h, PORT6.PCNTR2 87C0 00C4h, PORT7.PCNTR2 87C0 00E4h, PORT8.PCNTR2 87C0 0104h, PORT9.PCNTR2 87C0 0124h, PORTA.PCNTR2 87C0 0144h, PORTB.PCNTR2 87C0 0164h, PORTC.PCNTR2 87C0 0184h, PORTD.PCNTR2 87C0 01A4h, PORTE.PCNTR2 87C0 01C4h, PORTF.PCNTR2 87C0 01E4h, PORTG.PCNTR2 87C0 0204h,
PORT0_NS.PCNTR2 07C0 0004h, PORT1_NS.PCNTR2 07C0 0024h, PORT2_NS.PCNTR2 07C0 0044h, PORT3_NS.PCNTR2 07C0 0064h, PORT4_NS.PCNTR2 07C0 0084h, PORT5_NS.PCNTR2 07C0 00A4h, PORT6_NS.PCNTR2 07C0 00C4h, PORT7_NS.PCNTR2 07C0 00E4h, PORT8_NS.PCNTR2 07C0 0104h, PORT9_NS.PCNTR2 07C0 0124h, PORTA_NS.PCNTR2 07C0 0144h, PORTB_NS.PCNTR2 07C0 0164h, PORTC_NS.PCNTR2 07C0 0184h, PORTD_NS.PCNTR2 07C0 01A4h, PORTE_NS.PCNTR2 07C0 01C4h, PORTF_NS.PCNTR2 07C0 01E4h, PORTG_NS.PCNTR2 07C0 0204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDR00	Pm00 Input Data	0: Low level 1: High level	R
b1	PIDR01	Pm01 Input Data		R
b2	PIDR02	Pm02 Input Data		R
b3	PIDR03	Pm03 Input Data		R
b4	PIDR04	Pm04 Input Data		R
b5	PIDR05	Pm05 Input Data		R
b6	PIDR06	Pm06 Input Data		R
b7	PIDR07	Pm07 Input Data		R
b8	PIDR08	Pm08 Input Data		R
b9	PIDR09	Pm09 Input Data		R
b10	PIDR10	Pm10 Input Data		R
b11	PIDR11	Pm11 Input Data		R
b12	PIDR12	Pm12 Input Data		R
b13	PIDR13	Pm13 Input Data		R
b14	PIDR14	Pm14 Input Data		R
b15	PIDR15	Pm15 Input Data		R
b16	EIDR00	Port Event Input Data 00*1	When an ELC_PORTm event has occurred: 0: Low-level input 1: High-level input	R
b17	EIDR01	Port Event Input Data 01*1		R
b18	EIDR02	Port Event Input Data 02*1		R
b19	EIDR03	Port Event Input Data 03*1		R
b20	EIDR04	Port Event Input Data 04*1		R
b21	EIDR05	Port Event Input Data 05*1		R
b22	EIDR06	Port Event Input Data 06*1		R
b23	EIDR07	Port Event Input Data 07*1		R
b31 to b24	—	Reserved	These bits are read as 0.	R

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only ports B to E have this bit.

This register is read-only in 32-bit units and is used to read the input level and port event input data on the Pmn pins (n = 00 to 15). A bit corresponding to a pin that does not exist is reserved. The value read from a reserved bit is undefined.

22.3.5 Port Input Data Register (PIDR)

Address(es): PORT0.PIDR 87C0 0004h, PORT1.PIDR 87C0 0024h, PORT2.PIDR 87C0 0044h, PORT3.PIDR 87C0 0064h, PORT4.PIDR 87C0 0084h, PORT5.PIDR 87C0 00A4h, PORT6.PIDR 87C0 00C4h, PORT7.PIDR 87C0 00E4h, PORT8.PIDR 87C0 0104h, PORT9.PIDR 87C0 0124h, PORTA.PIDR 87C0 0144h, PORTB.PIDR 87C0 0164h, PORTC.PIDR 87C0 0184h, PORTD.PIDR 87C0 01A4h, PORTE.PIDR 87C0 01C4h, PORTF.PIDR 87C0 01E4h, PORTG.PIDR 87C0 0204h, PORT0_NS.PIDR 07C0 0004h, PORT1_NS.PIDR 07C0 0024h, PORT2_NS.PIDR 07C0 0044h, PORT3_NS.PIDR 07C0 0064h, PORT4_NS.PIDR 07C0 0084h, PORT5_NS.PIDR 07C0 00A4h, PORT6_NS.PIDR 07C0 00C4h, PORT7_NS.PIDR 07C0 00E4h, PORT8_NS.PIDR 07C0 0104h, PORT9_NS.PIDR 07C0 0124h, PORTA_NS.PIDR 07C0 0144h, PORTB_NS.PIDR 07C0 0164h, PORTC_NS.PIDR 07C0 0184h, PORTD_NS.PIDR 07C0 01A4h, PORTE_NS.PIDR 07C0 01C4h, PORTF_NS.PIDR 07C0 01E4h, PORTG_NS.PIDR 07C0 0204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDR00	Pm00 Input Data	0: Low level 1: High level	R
b1	PIDR01	Pm01 Input Data		R
b2	PIDR02	Pm02 Input Data		R
b3	PIDR03	Pm03 Input Data		R
b4	PIDR04	Pm04 Input Data		R
b5	PIDR05	Pm05 Input Data		R
b6	PIDR06	Pm06 Input Data		R
b7	PIDR07	Pm07 Input Data		R
b8	PIDR08	Pm08 Input Data		R
b9	PIDR09	Pm09 Input Data		R
b10	PIDR10	Pm10 Input Data		R
b11	PIDR11	Pm11 Input Data		R
b12	PIDR12	Pm12 Input Data		R
b13	PIDR13	Pm13 Input Data		R
b14	PIDR14	Pm14 Input Data		R
b15	PIDR15	Pm15 Input Data		R

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register reflects the input level on the Pmn pins (n = 00 to 15) and makes up the 16 lower-order bits of the PCNTR2 register. This register is accessible in 8- or 16-bit units. The level on a pin can be read from the corresponding PIDRn bit regardless of the value of the PMR bit in the PmnPFS register or the PDRn bit in the PCNTR1 register for PORTm. If any of the following functions is selected for a pin, the level on the pin is not reflected in the PIDRn bit.

- RTC time capture input (RTCIC0 to RTCIC2) and tamper detection (TAMPI0 to TAMPI2)
- Analog function (the ASEL bit in the PmnPFS register is 1)
- JTAG or SWD interface (TCK or SWCLK, TMS or SWDIO, TDI, or TDO)

22.3.6 Port Event Input Data Register (EIDR)

Address(es): PORTB.EIDR 87C0 0166h, PORTC.EIDR 87C0 0186h, PORTD.EIDR 87C0 01A6h, PORTE.EIDR 87C0 01C6h, PORTB_NS.EIDR 07C0 0166h, PORTC_NS.EIDR 07C0 0186h, PORTD_NS.EIDR 07C0 01A6h, PORTE_NS.EIDR 07C0 01C6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EIDR00	Port Event Input Data 00	When an ELC_PORTm event has occurred: 0: Low-level input 1: High-level input	R
b1	EIDR01	Port Event Input Data 01		R
b2	EIDR02	Port Event Input Data 02		R
b3	EIDR03	Port Event Input Data 03		R
b4	EIDR04	Port Event Input Data 04		R
b5	EIDR05	Port Event Input Data 05		R
b6	EIDR06	Port Event Input Data 06		R
b7	EIDR07	Port Event Input Data 07		R
b15 to b8	—	Reserved	These bits are read as 0.	R

m = B to E

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is used to read the level latched on the individual pins when a corresponding ELC_PORTm event has occurred and makes up the 16 higher-order bits of the PCNTR2 register. This register is accessible in 8- or 16-bit units. An EIDRn bit only reflects the level on the corresponding pin when both the PMR bit in the PmnPFS register (n = 00 to 07) and the PDRn bit in the PDR register for PORTm are set to 0. If the ASEL bit in the PmnPFS register is set to 1, the EIDRn bit does not reflect the level on the corresponding pin. A bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0.

22.3.7 Port Control Register 3 (PCNTR3)

Address(es): PORT0.PCNTR3 87C0 0008h, PORT1.PCNTR3 87C0 0028h, PORT2.PCNTR3 87C0 0048h, PORT3.PCNTR3 87C0 0068h, PORT4.PCNTR3 87C0 0088h, PORT5.PCNTR3 87C0 00A8h, PORT6.PCNTR3 87C0 00C8h, PORT7.PCNTR3 87C0 00E8h, PORT8.PCNTR3 87C0 0108h, PORT9.PCNTR3 87C0 0128h, PORTA.PCNTR3 87C0 0148h, PORTB.PCNTR3 87C0 0168h, PORTC.PCNTR3 87C0 0188h, PORTD.PCNTR3 87C0 01A8h, PORTE.PCNTR3 87C0 01C8h, PORTF.PCNTR3 87C0 01E8h, PORTG.PCNTR3 87C0 0208h,
PORT0_NS.PCNTR3 07C0 0008h, PORT1_NS.PCNTR3 07C0 0028h, PORT2_NS.PCNTR3 07C0 0048h, PORT3_NS.PCNTR3 07C0 0068h, PORT4_NS.PCNTR3 07C0 0088h, PORT5_NS.PCNTR3 07C0 00A8h, PORT6_NS.PCNTR3 07C0 00C8h, PORT7_NS.PCNTR3 07C0 00E8h, PORT8_NS.PCNTR3 07C0 0108h, PORT9_NS.PCNTR3 07C0 0128h, PORTA_NS.PCNTR3 07C0 0148h, PORTB_NS.PCNTR3 07C0 0168h, PORTC_NS.PCNTR3 07C0 0188h, PORTD_NS.PCNTR3 07C0 01A8h, PORTE_NS.PCNTR3 07C0 01C8h, PORTF_NS.PCNTR3 07C0 01E8h, PORTG_NS.PCNTR3 07C0 0208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PORR15	PORR14	PORR13	PORR12	PORR11	PORR10	PORR09	PORR08	PORR07	PORR06	PORR05	PORR04	PORR03	PORR02	PORR01	PORR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POSR15	POSR14	POSR13	POSR12	POSR11	POSR10	POSR09	POSR08	POSR07	POSR06	POSR05	POSR04	POSR03	POSR02	POSR01	POSR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POSR00	Pm00 Output Set	0: No effect on output 1: High-level output	W
b1	POSR01	Pm01 Output Set		W
b2	POSR02	Pm02 Output Set		W
b3	POSR03	Pm03 Output Set		W
b4	POSR04	Pm04 Output Set		W
b5	POSR05	Pm05 Output Set		W
b6	POSR06	Pm06 Output Set		W
b7	POSR07	Pm07 Output Set		W
b8	POSR08	Pm08 Output Set		W
b9	POSR09	Pm09 Output Set		W
b10	POSR10	Pm10 Output Set		W
b11	POSR11	Pm11 Output Set		W
b12	POSR12	Pm12 Output Set		W
b13	POSR13	Pm13 Output Set		W
b14	POSR14	Pm14 Output Set		W
b15	POSR15	Pm15 Output Set		W

Bit	Symbol	Bit Name	Description	R/W
b16	PORR00	Pm00 Output Reset	0: No effect on output 1: Low-level output	W
b17	PORR01	Pm01 Output Reset		W
b18	PORR02	Pm02 Output Reset		W
b19	PORR03	Pm03 Output Reset		W
b20	PORR04	Pm04 Output Reset		W
b21	PORR05	Pm05 Output Reset		W
b22	PORR06	Pm06 Output Reset		W
b23	PORR07	Pm07 Output Reset		W
b24	PORR08	Pm08 Output Reset		W
b25	PORR09	Pm09 Output Reset		W
b26	PORR10	Pm10 Output Reset		W
b27	PORR11	Pm11 Output Reset		W
b28	PORR12	Pm12 Output Reset		W
b29	PORR13	Pm13 Output Reset		W
b30	PORR14	Pm14 Output Reset		W
b31	PORR15	Pm15 Output Reset		W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note: This register is write-only. It is always read as 0.

This register is writable in 32-bit units and is used to set or reset the data output through the Pmn pins (n = 00 to 15). Do not set both the PORRn and POSRn bits for a single pin to 1. A bit corresponding to a pin that does not exist is reserved. When writing to such bits, write 0 to them. As the P305, P308, and P309 pins are input-only, the POSR05, POSR08, POSR09, PORR05, PORR08, and PORR09 bits in the PCNTR3 register for PORT3 are also reserved.

22.3.8 Port Output Set Register (POSR)

Address(es): PORT0.POSR 87C0 0008h, PORT1.POSR 87C0 0028h, PORT2.POSR 87C0 0048h, PORT3.POSR 87C0 0068h, PORT4.POSR 87C0 0088h, PORT5.POSR 87C0 00A8h, PORT6.POSR 87C0 00C8h, PORT7.POSR 87C0 00E8h, PORT8.POSR 87C0 0108h, PORT9.POSR 87C0 0128h, PORTA.POSR 87C0 0148h, PORTB.POSR 87C0 0168h, PORTC.POSR 87C0 0188h, PORTD.POSR 87C0 01A8h, PORTE.POSR 87C0 01C8h, PORTF.POSR 87C0 01E8h, PORTG.POSR 87C0 0208h,
PORT0_NS.POSR 07C0 0008h, PORT1_NS.POSR 07C0 0028h, PORT2_NS.POSR 07C0 0048h, PORT3_NS.POSR 07C0 0068h, PORT4_NS.POSR 07C0 0088h, PORT5_NS.POSR 07C0 00A8h, PORT6_NS.POSR 07C0 00C8h, PORT7_NS.POSR 07C0 00E8h, PORT8_NS.POSR 07C0 0108h, PORT9_NS.POSR 07C0 0128h, PORTA_NS.POSR 07C0 0148h, PORTB_NS.POSR 07C0 0168h, PORTC_NS.POSR 07C0 0188h, PORTD_NS.POSR 07C0 01A8h, PORTE_NS.POSR 07C0 01C8h, PORTF_NS.POSR 07C0 01E8h, PORTG_NS.POSR 07C0 0208h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POSR15	POSR14	POSR13	POSR12	POSR11	POSR10	POSR09	POSR08	POSR07	POSR06	POSR05	POSR04	POSR03	POSR02	POSR01	POSR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POSR00	Pm00 Output Set	0: No effect on output 1: High-level output	W
b1	POSR01	Pm01 Output Set		W
b2	POSR02	Pm02 Output Set		W
b3	POSR03	Pm03 Output Set		W
b4	POSR04	Pm04 Output Set		W
b5	POSR05	Pm05 Output Set		W
b6	POSR06	Pm06 Output Set		W
b7	POSR07	Pm07 Output Set		W
b8	POSR08	Pm08 Output Set		W
b9	POSR09	Pm09 Output Set		W
b10	POSR10	Pm10 Output Set		W
b11	POSR11	Pm11 Output Set		W
b12	POSR12	Pm12 Output Set		W
b13	POSR13	Pm13 Output Set		W
b14	POSR14	Pm14 Output Set		W
b15	POSR15	Pm15 Output Set		W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is accessible in 8- or 16-bit units and is used to set the corresponding bits in the PODR register to 1. This register is always read as 0. Writing 1 to a POSRn bit (n = 00 to 15) sets the corresponding PODRn bit in the PODR register to 1. A bit corresponding to a pin that does not exist is reserved. When writing to such bits, write 0 to them. As the P305, P308, and P309 pins are input-only, the POSR05, POSR08, and POSR09 bits in the POSR register for PORT3 are also reserved.

22.3.9 Port Output Reset Register (PORR)

Address(es): PORT0.PORR 87C0 000Ah, PORT1.PORR 87C0 002Ah, PORT2.PORR 87C0 004Ah, PORT3.PORR 87C0 006Ah, PORT4.PORR 87C0 008Ah, PORT5.PORR 87C0 00AAh, PORT6.PORR 87C0 00CAh, PORT7.PORR 87C0 00EAh, PORT8.PORR 87C0 010Ah, PORT9.PORR 87C0 012Ah, PORTA.PORR 87C0 014Ah, PORTB.PORR 87C0 016Ah, PORTC.PORR 87C0 018Ah, PORTD.PORR 87C0 01AAh, PORTE.PORR 87C0 01CAh, PORTF.PORR 87C0 01EAh, PORTG.PORR 87C0 020Ah,
PORT0_NS.PORR 07C0 000Ah, PORT1_NS.PORR 07C0 002Ah, PORT2_NS.PORR 07C0 004Ah, PORT3_NS.PORR 07C0 006Ah, PORT4_NS.PORR 07C0 008Ah, PORT5_NS.PORR 07C0 00AAh, PORT6_NS.PORR 07C0 00CAh, PORT7_NS.PORR 07C0 00EAh, PORT8_NS.PORR 07C0 010Ah, PORT9_NS.PORR 07C0 012Ah, PORTA_NS.PORR 07C0 014Ah, PORTB_NS.PORR 07C0 016Ah, PORTC_NS.PORR 07C0 018Ah, PORTD_NS.PORR 07C0 01AAh, PORTE_NS.PORR 07C0 01CAh, PORTF_NS.PORR 07C0 01EAh, PORTG_NS.PORR 07C0 020Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PORR15	PORR14	PORR13	PORR12	PORR11	PORR10	PORR09	PORR08	PORR07	PORR06	PORR05	PORR04	PORR03	PORR02	PORR01	PORR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PORR00	Pm00 Output Reset	0: No effect on output 1: Low-level output	W
b1	PORR01	Pm01 Output Reset		W
b2	PORR02	Pm02 Output Reset		W
b3	PORR03	Pm03 Output Reset		W
b4	PORR04	Pm04 Output Reset		W
b5	PORR05	Pm05 Output Reset		W
b6	PORR06	Pm06 Output Reset		W
b7	PORR07	Pm07 Output Reset		W
b8	PORR08	Pm08 Output Reset		W
b9	PORR09	Pm09 Output Reset		W
b10	PORR10	Pm10 Output Reset		W
b11	PORR11	Pm11 Output Reset		W
b12	PORR12	Pm12 Output Reset		W
b13	PORR13	Pm13 Output Reset		W
b14	PORR14	Pm14 Output Reset		W
b15	PORR15	Pm15 Output Reset		W

m = 0 to 9 or A to G

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is accessible in 8- or 16-bit units and is used to set the corresponding bits in the PODR register to 0. This register is always read as 0. Writing 1 to a PORRn bit (n = 00 to 15) sets the corresponding PODRn bit in the PODR register to 0. A bit corresponding to a pin that does not exist is reserved. When writing to such bits, write 0 to them. As the P305, P308, and P309 pins are input-only, the PORR05, PORR08, and PORR09 bits in the PORR register for PORT3 are also reserved.

22.3.10 Port Control Register 4 (PCNTR4)

Address(es): PORTB.PCNTR4 87C0 016Ch, PORTC.PCNTR4 87C0 018Ch, PORTD.PCNTR4 87C0 01ACh, PORTE.PCNTR4 87C0 01CCh, PORTB_NS.PCNTR4 07C0 016Ch, PORTC_NS.PCNTR4 07C0 018Ch, PORTD_NS.PCNTR4 07C0 01ACh, PORTE_NS.PCNTR4 07C0 01CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	EORR07	EORR06	EORR05	EORR04	EORR03	EORR02	EORR01	EORR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EOSR07	EOSR06	EOSR05	EOSR04	EOSR03	EOSR02	EOSR01	EOSR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EOSR00	Pm00 Event Output Set	When an ELC_PORTm event has occurred: 0: No effect on output 1: High-level output	R/W
b1	EOSR01	Pm01 Event Output Set		R/W
b2	EOSR02	Pm02 Event Output Set		R/W
b3	EOSR03	Pm03 Event Output Set		R/W
b4	EOSR04	Pm04 Event Output Set		R/W
b5	EOSR05	Pm05 Event Output Set		R/W
b6	EOSR06	Pm06 Event Output Set		R/W
b7	EOSR07	Pm07 Event Output Set		R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EORR00	Pm00 Event Output Reset	When an ELC_PORTm event has occurred: 0: No effect on output 1: Low-level output	R/W
b17	EORR01	Pm01 Event Output Reset		R/W
b18	EORR02	Pm02 Event Output Reset		R/W
b19	EORR03	Pm03 Event Output Reset		R/W
b20	EORR04	Pm04 Event Output Reset		R/W
b21	EORR05	Pm05 Event Output Reset		R/W
b22	EORR06	Pm06 Event Output Reset		R/W
b23	EORR07	Pm07 Event Output Reset		R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = B to E

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

This register is readable and writable in 32-bit units and is used to set or reset the data output through the Pmn pins (n = 00 to 07) when an ELC_PORTm event has occurred. Do not set both the EORRn and EOSRn bits for a single pin to 1.

22.3.11 Event Output Set Register (EOSR)

Address(es): PORTB.EOSR 87C0 016Ch, PORTC.EOSR 87C0 018Ch, PORTD.EOSR 87C0 01ACh, PORTE.EOSR 87C0 01CCh, PORTB_NS.EOSR 07C0 016Ch, PORTC_NS.EOSR 07C0 018Ch, PORTD_NS.EOSR 07C0 01ACh, PORTE_NS.EOSR 07C0 01CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EOSR07	EOSR06	EOSR05	EOSR04	EOSR03	EOSR02	EOSR01	EOSR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EOSR00	Pm00 Event Output Set	When an ELC_PORTm event has occurred: 0: No effect on output 1: High-level output	R/W
b1	EOSR01	Pm01 Event Output Set		R/W
b2	EOSR02	Pm02 Event Output Set		R/W
b3	EOSR03	Pm03 Event Output Set		R/W
b4	EOSR04	Pm04 Event Output Set		R/W
b5	EOSR05	Pm05 Event Output Set		R/W
b6	EOSR06	Pm06 Event Output Set		R/W
b7	EOSR07	Pm07 Event Output Set		R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = B to E

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

After an EOSRn bit has been set to 1, the corresponding PODRn bit in the PODR register is set to 1 when an ELC_PORTm event occurs (n = 00 to 07). When the EORRn bit for a pin is set to 1 in the EORR register, the EOSRn bit for the pin in this register should be set to 0. This register is accessible in 8- or 16-bit units. A bit corresponding to a pin that does not exist is reserved. When writing to such bits, write 0 to them.

22.3.12 Event Output Reset Register (EORR)

Address(es): PORTB.EORR 87C0 016Eh, PORTC.EORR 87C0 018Eh, PORTD.EORR 87C0 01AEh, PORTE.EORR 87C0 01CEh, PORTB_NS.EORR 07C0 016Eh, PORTC_NS.EORR 07C0 018Eh, PORTD_NS.EORR 07C0 01AEh, PORTE_NS.EORR 07C0 01CEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EORR07	EORR06	EORR05	EORR04	EORR03	EORR02	EORR01	EORR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EORR00	Pm00 Event Output Reset	When an ELC_PORTm event has occurred: 0: No effect on output 1: Low-level output	R/W
b1	EORR01	Pm01 Event Output Reset		R/W
b2	EORR02	Pm02 Event Output Reset		R/W
b3	EORR03	Pm03 Event Output Reset		R/W
b4	EORR04	Pm04 Event Output Reset		R/W
b5	EORR05	Pm05 Event Output Reset		R/W
b6	EORR06	Pm06 Event Output Reset		R/W
b7	EORR07	Pm07 Event Output Reset		R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = B to E

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

After an EORRn bit has been set to 1, the corresponding PODRn bit in the PODR register is set to 0 when an ELC_PORTm event occurs (n = 00 to 07). When the EOSRn bit for a pin is set to 1 in the EOSR register, the EORRn bit for the pin in this register should be set to 0. This register is accessible in 8- or 16-bit units. A bit corresponding to a pin that does not exist is reserved. When writing to such bits, write 0 to them.

22.3.13 Pmn Pin Function Selection Register (PmnPFS) (m = 0 to 9, A to G; n = 00 to 15)

Address(es): PFS.P000PFS 87C0 0800h to PFS.P015PFS 87C0 083Ch,
PFS.P100PFS 87C0 0840h to PFS.P115PFS 87C0 087Ch,
PFS.P200PFS 87C0 0880h to PFS.P215PFS 87C0 08BCh,
PFS.P300PFS 87C0 08C0h to PFS.P312PFS 87C0 08F0h, PFS.P315PFS 87C0 08FCh,
PFS.P400PFS 87C0 0900h to PFS.P415PFS 87C0 093Ch,
PFS.P500PFS 87C0 0940h to PFS.P515PFS 87C0 097Ch,
PFS.P600PFS 87C0 0980h to PFS.P607PFS 87C0 099Ch,
PFS.P700PFS 87C0 09C0h to PFS.P715PFS 87C0 09FCh,
PFS.P800PFS 87C0 0A00h to PFS.P807PFS 87C0 0A1Ch,
PFS.P900PFS 87C0 0A40h to PFS.P907PFS 87C0 0A5Ch,
PFS.PA00PFS 87C0 0A80h to PFS.PA07PFS 87C0 0A9Ch,
PFS.PB00PFS 87C0 0AC0h to PFS.PB07PFS 87C0 0ADCh,
PFS.PC00PFS 87C0 0B00h to PFS.PC07PFS 87C0 0B1Ch,
PFS.PD00PFS 87C0 0B40h to PFS.PD07PFS 87C0 0B5Ch,
PFS.PE00PFS 87C0 0B80h to PFS.PE07PFS 87C0 0B9Ch,
PFS.PF00PFS 87C0 0BC0h to PFS.PF09PFS 87C0 0BE4h,
PFS.PG00PFS 87C0 0C00h to PFS.PG15PFS 87C0 0C3Ch,
PFS_NS.P000PFS 07C0 0800h to PFS_NS.P015PFS 07C0 083Ch,
PFS_NS.P100PFS 07C0 0840h to PFS_NS.P115PFS 07C0 087Ch,
PFS_NS.P200PFS 07C0 0880h to PFS_NS.P215PFS 07C0 08BCh,
PFS_NS.P300PFS 07C0 08C0h to PFS_NS.P312PFS 07C0 08F0h, PFS_NS.P315PFS 07C0 08FC,
PFS_NS.P400PFS 07C0 0900h to PFS_NS.P415PFS 07C0 093Ch,
PFS_NS.P500PFS 07C0 0940h to PFS_NS.P515PFS 07C0 097Ch,
PFS_NS.P600PFS 07C0 0980h to PFS_NS.P607PFS 07C0 099Ch,
PFS_NS.P700PFS 07C0 09C0h to PFS_NS.P715PFS 07C0 09FCh,
PFS_NS.P800PFS 07C0 0A00h to PFS_NS.P807PFS 07C0 0A1Ch,
PFS_NS.P900PFS 07C0 0A40h to PFS_NS.P907PFS 07C0 0A5Ch,
PFS_NS.PA00PFS 07C0 0A80h to PFS_NS.PA07PFS 07C0 0A9Ch,
PFS_NS.PB00PFS 07C0 0AC0h to PFS_NS.PB07PFS 07C0 0ADCh,
PFS_NS.PC00PFS 07C0 0B00h to PFS_NS.PC07PFS 07C0 0B1Ch,
PFS_NS.PD00PFS 07C0 0B40h to PFS_NS.PD07PFS 07C0 0B5Ch,
PFS_NS.PE00PFS 07C0 0B80h to PFS_NS.PE07PFS 07C0 0B9Ch,
PFS_NS.PF00PFS 07C0 0BC0h to PFS_NS.PF09PFS 07C0 0BE4h,
PFS_NS.PG00PFS 07C0 0C00h to PFS_NS.PG15PFS 07C0 0C3Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	PSEL[6:0]							—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0*1	0	0	0*1	0*1	0	0	0	0	0	0	0	0*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	EOFR[1:0]		DSCR[3:0]				—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0*1	0	0	x	0

Bit	Symbol	Bit Name	Description	R/W
b0	PODR	Port Output Data	0: Low-level output 1: High-level output	R/W
b1	PIDR	Port Input Data	0: Low level 1: High level	R
b2	PDR	Port I/O Direction Select	0: Input (The pin functions as an input port pin.) 1: Output (The pin functions as an output port pin.)	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	PCR	Input Pull-up Resistor Control	0: The input pull-up resistor is disabled. 1: The input pull-up resistor is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	NCODR	N-channel Open-Drain Output Select	0: CMOS output 1: N-channel open-drain output	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b11 to b8	DSCR[3:0]	Driving Ability Select*2	b11 b8 0 0 0 0: DRV1 0 0 0 1: DRV2 0 0 1 0: DRV3 0 0 1 1: DRV4 0 1 0 0: DRV5 0 1 0 1: DRV6 0 1 1 0: DRV7 0 1 1 1: DRV8 Settings other than above are prohibited.	R/W
b13, b12	EOFR[1:0]	Port Group Input Edge Detection Select *3	b13 b12 0 0: The pin is not assigned to a port group. 0 1: Rising edges are detected. 1 0: Falling edges are detected. 1 1: Both edges are detected.	R/W
b14	ISEL	Interrupt Input Function Select	0: The pin is not used as an IRQ input pin. 1: The pin is used as an IRQ input pin	R/W
b15	ASEL	Analog Function Select	0: The pin is not used as an analog signal pin. 1: The pin is used as an analog signal pin.	R/W
b16	PMR	Port Mode Select	0: The pin is used as a general I/O port pin. 1: The pin is used as an I/O pin for a peripheral function.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30 to b24	PSEL[6:0]	Peripheral Function Select	These bits are used to select a peripheral function. For the peripheral functions that are selectable for each pin, see the relevant tables in this section.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The bits are initialized to the following values.
- PSEL[6:0] bits in the PF00PFS and PF01PFS registers: 0010011b
- PMR bit in the P300PFS, P301PFS, P302PFS, P303PFS, P315PFS, PF00PFS, and PF01PFS registers: 1
- DSCR[3:0] bits in the P300PFS, P302PFS, and P315PFS registers: 0001b
- PCR bit in the P300PFS, P301PFS, P303PFS, and P315PFS registers: 1

Note 2. Specifying a non-supported driving ability is prohibited. For the driving abilities supported by the individual pins, see Table 22.2, Port Functions.

Note 3. Only ports B to E have these bits.

This register is readable and writable in 32-bit, 16-bit, or 8-bit units and is used to specify various functions of the Pmn pin. The PmnPFS register is protected by the PWPR register. Release the protection before modifying the value of this register.

PODR Bit (Port Output Data)

The function of this bit is the same as that of the corresponding PODRn bit in the PODR register for PORTm.

PIDR Bit (Port Input Data)

The function of this bit is the same as that of the corresponding PIDRn bit in the PIDR register for PORTm.

PDR Bit (Port I/O Direction Select)

The function of this bit is the same as that of the corresponding PDRn bit in the PDR register for PORTm.

PCR Bit (Input Pull-up Resistor Control)

This bit enables or disables the input pull-up resistor for the pin. When the pin is set as an input and this bit is set to 1, the pull-up resistor connected to the pin is enabled. When the pin is set as a general output port or an output pin for a peripheral function, the pull-up resistor is disabled regardless of the setting of this bit.

DSCR[3:0] Bits (Driving Ability Select)

These bits switch the driving ability of the output buffer for the pin. Specifying a driving ability that is not supported for the pin is prohibited. For the driving abilities supported by the individual pins, see Table 22.2. These bits are reserved for a pin that has a fixed driving ability. The write value should be 0000b in this case.

EOFR[1:0] Bits (Port Group Input Edge Detection Select)

These bits select the edges to be detected in the input signal for the pin in a port group. Rising edges, falling edges, or both edges can be selected. While these bits are set to 01b, 10b, or 11b, an event pulse is applied to the ELC when the level on the external pin changes in the corresponding way.

ISEL Bit (Interrupt Input Function Select)

This bit specifies the use of the pin as an IRQ input pin. The pin can be set to work as both an IRQ input pin and a peripheral function pin. Note that when a single IRQ signal is assignable to multiple pins, the ISEL bit for only one of the pins should be set to 1. This bit is reserved for a pin to which an IRQ signal is not assigned.

ASEL Bit (Analog Function Select)

This bit specifies the use of the pin as an analog I/O signal pin. When setting this bit to 1, set each of the corresponding PMR, PCR, and PDR bits to 0. This bit is reserved for a pin to which an analog I/O signal is not assigned.

PSEL[6:0] Bits (Peripheral Function Select)

These bits select the signal to be assigned to the pin. When a single peripheral function signal is assignable to multiple pins, the PSEL[6:0] bits for only one of the pins should be set to select the signal. For the settings of the individual signals for peripheral functions, see Table 22.3 to Table 22.28.

Table 22.3 Register Settings for I/O Pin Functions (PORT0, Pins P000 to P007)

PSEL[6:0] Setting	Function	Pin							
		P000	P001	P002	P003	P004	P005	P006	P007
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO7	GTCPPO0	GTCPPO1	GTCPPO2	GTCPPO3	GTCPPO4	GTCPPO5	GTCPPO6
0000011b	GPTW	GTIOC10A	GTIOC10B	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B
0000100b	GPTW	—	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B	—
0000101b	TMR	TMRI0	TMCI0	TMCI1	—	—	—	—	—
0001000b	POEG	POEGB	—	—	—	—	—	—	POEGA
0001001b	ADC	—	—	—	—	—	ADTRG2#	ADTRG1#	ADTRG0#
0001010b	RSCI*1	CTS000#	—	—	—	TXD000-B/ SMOSI000-B/SSDA000/ TXDA000	RXD000-B/ SMISO000-B/SSCL000	SCK000-B	CTS000#*2/ RTS000#*2/ SS000#
0001011b	RSCI	—	—	—	—	—	—	CTS000#	DE000
0001100b	RSCI	—	—	—	—	—	—	TXDB000	TXDB000
0001101b	RSCI*1	CTS007#*2/ RTS007#*2/ SS007#	SCK007-A	TXD007-A/ SMOSI007-A/SSDA007/ TXDA007	RXD007-A/ SMISO007-A/SSCL007	—	—	—	—
0001110b	RSCI	DE007	CTS007#	—	—	—	—	—	—
0001111b	RSCI	TXDB007	TXDB007	—	—	—	—	—	—
0010110b	PDMIF	—	—	—	—	—	—	PDMCLK0	PDMDAT0
0010111b	SSIE	AUDIO_CLK	SSIBCK0	SSIBCK1	SSIRXD1	SSITXD1	SSILRCK1	—	—
0011000b	PCIF	—	PCOUTZ0	PCOUTA0	PCOUTB0	PCOUTA1	PCOUTB1	PCOUTZ1	—
0100000b	ESWM*1 (GMII/MII)	ET0_INT	ET0_LINKSTA	ET0_MDC-D	ET0_MDIO-D	—	—	—	—
0100001b	ESWM*1 (RMII)	ET0_INT	ET0_LINKSTA	ET0_MDC-D	ET0_MDIO-D	—	—	—	—
0100010b	ESWM*1 (RGMII)	ET0_INT	ET0_LINKSTA	ET0_MDC-D	ET0_MDIO-D	—	—	—	—
1000000b	PIO	PIO_000	PIO_001	PIO_002	PIO_003	PIO_004	PIO_005	PIO_006	PIO_007
ASEL bit		AN016	AN017	AN018	AN019	AN020	AN021	AN022	AN023
ISEL bit		IRQ16-DS	IRQ17-DS	IRQ18-DS	IRQ19-DS	IRQ20-DS	IRQ21-DS	IRQ22-DS	IRQ23-DS
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		—	✓	✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.4 Register Settings for I/O Pin Functions (PORT0, Pins P008 to P015)

PSEL[6:0] Setting	Function	Pin							
		P008	P009	P010	P011	P012	P013	P014	P015
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO8	GTCPPO9	GTCPPO10	GTCPPO11	GTCPPO12	GTCPPO13	GTCPPO14	GTCPPO15
0000011b	GPTW	GTIOC14A	GTIOC14B	GTIOC15A	GTIOC15B	—	—	—	—
0001001b	ADC	ADTRG1#	ADTRG2#	—	—	—	—	—	—
0001010b	RSCI*1	TXD006-C/ SMOSI006-C/SSDA006/ TXDA006	RXD006-C/ SMISO006-C/SSCL006	SCK006-C	CTS006#*2/ RTS006#*2/ SS006#	CTS006#	—	—	—
0001011b	RSCI	—	—	CTS006#	DE006	—	—	—	—
0001100b	RSCI	—	—	TXDB006	TXDB006	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	—	ET0_LINKSTA	ET0_WOL	—	—	—	ET0_COL	ET0_CRS
0100001b	ESWM*1 (RMII)	—	ET0_LINKSTA	ET0_WOL	—	—	—	—	—
0100010b	ESWM*1 (RGMII)	—	ET0_LINKSTA	ET0_WOL	—	—	—	—	—
0100011b	ESWM	—	—	GPTP_CAPTURE0	GPTP_MATCH0	GPTP0_PPS	GPTP_CAPTURE1	GPTP_MATCH1	—
1000000b	PIO	PIO_032	PIO_033	PIO_034	PIO_035	PIO_036	PIO_037	PIO_038	PIO_039
ISEL bit		IRQ24	IRQ25	IRQ26	IRQ27	IRQ28	IRQ29	IRQ30	IRQ31
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		—	—	—	—	—	—	—	—
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.5 Register Settings for I/O Pin Functions (PORT1, Pins P100 to P107)

PSEL[6:0] Setting	Function	Pin							
		P100	P101	P102	P103	P104	P105	P106	P107
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTIU	GTETRGC	—	—	GTIW	GTIV	GTIU	GTIV
0000010b	GPTW	GTADSM0	GTADSM1	GTADSM0	GTADSM1	—	—	—	—
0000011b	GPTW	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B	GTIOC14A	GTIOC14B	GTIOC15A	GTIOC15B
0000101b	TMR	TMRI3	TMCI3	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
0000110b	RTC	—	—	—	—	—	—	RTCOUT	—
0000111b	CAC	CACREF	—	—	—	—	—	—	—
0001000b	POEG	—	—	—	—	—	—	POEGB	POEGA
0001001b	ADC	ADTRG2#	ADTRG1#	—	—	ADST0	ADST1	ADTRG0#	ADTRG1#
0001010b	RSCI*1	CTS002#*2/ RTS002#*2/ SS002#	SCK002-B	—	CTS002#	RXD002-B/ SMISO002-B/SSCL002	TXD002-B/ SMOSI002-B/SSDA002/ TXDA002	—	—
0001011b	RSCI	DE002	CTS002#	—	—	—	—	—	—
0001100b	RSCI	TXDB002	TXDB002	—	—	—	—	—	—
0001101b	RSCI*1	CTS005#	—	—	—	CTS003#*2/ RTS003#*2/ SS003#	SCK003-B	RXD003-B/ SMISO003-B/SSCL003	TXD003-B/ SMOSI003-B/SSDA003/ TXDA003
0001110b	RSCI	—	—	—	—	DE003	CTS003#	—	—
0001111b	RSCI	—	—	—	—	TXDB003	TXDB003	—	—
0010001b	RI3C*1/RIICHS*1	—	—	SCL00-A	SDA00-A	—	—	SCLHS2-A	SDAHS2-A
0010010b	CANFD	—	—	—	—	CTX2	CRX2	—	—
0010011b	USB0	USB0_OVRCURB-DS	USB0_VBUS-DS	—	—	USB0_OVRCURA-DS	USB0_VBUS	—	—
0010100b	USBHS	USBHS_VBUS	USBHS_EXICEN	—	—	USBHS_VBUS	USBHS_OVRCURB	—	—
0010111b	SSIE	—	—	—	—	—	SSILRCK1	—	SSITXD0
0011000b	PCIF	—	—	—	—	PCINA0	PCINB0	PCINA1	PCINB1
0011001b	SDHI*1	—	—	—	—	—	—	SD0_DAT2-C	SD0_DAT3-C
0011100b	PDC	—	—	—	—	PIXD0	PIXD1	PIXD2	PIXD3
0011111b	CLKOUT	—	—	—	—	—	—	CLKOUT	—
0100000b	ESWM*1 (GMII/MII)	ET0_TX_ER-B	ET0_TX_CLK-B	ET0_COL	ET0_CRS	ET0_TXD0-B	ET0_TXD1-B	ET0_TXD2-B	ET0_TXD3-B
0100001b	ESWM*1 (RMII)	—	—	—	—	RMII0_TXD0-B	RMII0_TXD1-B	—	—
0111101b	BUS*1	CS1#-B	CS0#-A	—	—	—	—	—	—
0111110b	BUS*1	WAIT#-A	ALE-B	—	—	—	—	—	—
1000000b	PIO	PIO_008	PIO_009	PIO_010	PIO_011	PIO_012	PIO_013	PIO_014	PIO_015
ISEL bit		IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ29-DS	IRQ30-DS	IRQ31-DS
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		—	✓	✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.
Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.6 Register Settings for I/O Pin Functions (PORT1, Pins P108 to P115)

PSEL[6:0] Setting	Function	Pin							
		P108	P109	P110	P111	P112	P113	P114	P115
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO8	GTCPPO9	GTCPPO10	GTCPPO11	GTCPPO12	GTCPPO13	GTCPPO14	GTCPPO15
0001001b	ADC	ADCNV0	ADCNV1	ADCNV2	—	ADST0	ADST1	ADST2	—
0001010b	RSCI*1	—	—	—	CTS012#	CTS012#*2/ RTS012#*2/ SS012#	SCK012-B	TXD012-B/ SMOSI012-B/SSDA012/ TXDA012	RXD012-B/ SMISO012-B/SSCL012
0001011b	RSCI	—	—	—	—	DE012	CTS012#	—	—
0001100b	RSCI	—	—	—	—	TXDB012	TXDB012	—	—
0001101b	RSCI*1	SCK009-C	RXD009-C/ SMISO009-C/SSCL009	TXD009-C/ SMOSI009-C/SSDA009/ TXDA009	CTS009#*2/ RTS009#*2/ SS009#	CTS009#	—	—	—
0001110b	RSCI	CTS009#	—	—	DE009	—	—	—	—
0001111b	RSCI	TXDB009	—	—	TXDB009	—	—	—	—
0010000b	RSPI*1	RSPCK2-C	MOSI2-C	MISO2-C	SSL20-C	SSL21-C	SSL22-C	SSL23-C	—
0010110b	PDMIF	—	—	—	—	PDMCLK0	PDMDAT0	—	—
0011001b	SDHI*1	SD0_DAT4-C	SD0_DAT5-C	SD0_DAT6-C	SD0_DAT7-C	—	—	—	—
0011100b	PDC	PIXD0	PIXD1	PIXD2	PIXD3	PIXD4	PIXD5	PIXD6	PIXD7
0011101b	CMTW	TIC0	TOC0	—	—	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_RX_DV-C	ET0_RX_CLK-C	ET0_RXD0-C	ET0_RXD1-C	ET0_RXD2-C	ET0_RXD3-C	—	—
0100001b	ESWM*1 (RMII)	RMII0_CRS_DV-C	RMII0_REF50CK-C	RMII0_RXD0-C	RMII0_RXD1-C	—	—	—	—
1000000b	PIO	PIO_040	PIO_041	PIO_042	PIO_043	PIO_044	PIO_045	PIO_046	PIO_047
ISEL bit		IRQ16	IRQ17	IRQ18	IRQ19	IRQ20	IRQ21	IRQ22	IRQ23
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.7 Register Settings for I/O Pin Functions (PORT2, Pins P200 to P207)

PSEL[6:0] Setting	Function	Pin							
		P200	P201	P202	P203	P204	P205	P206	P207
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGA	GTIW	GTOUUP	GTOULO	GTOVUP	GTOVLO	GTOWUP	GTOWLO
0000011b	GPTW	GTIOC3A	GTIOC3B	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC2A	GTIOC2B
0000100b	GPTW	GTIOC14A	GTIOC14B	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B
0000101b	TMR	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO3	TMRI3
0000110b	RTC	—	—	—	—	—	—	RTCOUT	—
0001001b	ADC	ADTRG2#	—	—	—	—	ADTRG0#	ADTRG1#	ADTRG2#
0001010b	RSCI*1	CTS006#*2/ RTS006#*2/ SS006#	SCK006-A	TXD006-A/ SMOSI006-A/SSDA006/ TXDA006	RXD006-A/ SMISO006-A/SSCL006	—	—	CTS006#	—
0001011b	RSCI	DE006	CTS006#	—	—	—	—	—	—
0001100b	RSCI	TXDB006	TXDB006	—	—	—	—	—	—
0001101b	RSCI*1	—	—	—	CTS003#	CTS001#*2/ RTS001#*2/ SS001#	SCK001-B	TXD001-B/ SMOSI001-B/SSDA001/ TXDA001	RXD001-B/ SMISO001-B/SSCL001
0001110b	RSCI	—	—	—	—	DE001	CTS001#	—	—
0001111b	RSCI	—	—	—	—	TXDB001	TXDB001	—	—
0010000b	RSPI*1	MOSI0-C	MISO0-C	RSPCK0-C	SSL00-C	SSL10-B	RSPCK1-B	MOSI1-B	MISO1-B
0010001b	RI3C*1/RIICHS*1	SDAHS1-A	SCLHS1-A	SCL00-B	SDA00-B	SCLHS1-B	SDAHS1-B	SCLHS2-B	SDAHS2-B
0010010b	CANFD	—	—	CTX2	CRX2	CTX0	CRX0	CTX1	CRX1
0010011b	USB0	USB0_ID	USB0_EXICEN	USB0_OVRCURA	USB0_OVRCURB	USB0_VBUSEN	USB0_VBUS	USB0_EXICEN	USB0_ID
0010100b	USBHS	—	—	USBHS_EXICEN	USBHS_ID	USBHS_VBUS-DS	USBHS_VBUSEN	USBHS_OVRCURA-DS	USBHS_OVRCURB-DS
0010110b	PDMIF	—	—	PDMCLK0	PDMDAT0	—	—	—	—
0010111b	SSIE	SSIRXD0	SSILRCK0	AUDIO_CLK	SSIBCK0	SSIBCK1	SSIRXD1	SSITXD1	—
0011000b	PCIF	PCINZ1	PCINZ0	—	—	PCOUTA1	PCOUTB1	PCOUTZ1	—
0011001b	SDHI*1	SD0_CMD-C	SD0_CLK-C	SD0_DAT0-C	SD0_DAT1-C	SD0_WP	SD0_CD	—	—
0011100b	PDC	PIXD4	PIXD5	PIXD6	PIXD7	PIXCLK	HSYNC	VSYNC	PCKO
0011111b	CLKOUT	—	—	—	—	ETHPHYCLK	CLKOUT	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_TX_EN-B	ET0_RX_ER-B	ET0_RX_DV-B	ET0_RX_CLK-B	ET0_RXD0-B	ET0_RXD1-B	ET0_RXD2-B	ET0_RXD3-B
0100001b	ESWM*1 (RMII)	RMII0_TX_EN-B	RMII0_RX_ER-B	RMII0_CRS_DV-B	RMII0_REF50CK-B	RMII0_RXD0-B	RMII0_RXD1-B	—	—
0111101b	BUS*1	—	—	CS3#-B	CS2#-B	A20-C	A21-C	A22-C	A23-C
0111110b	BUS*1	—	—	BREQ#-A	BACK#-A	CS7#-C	CS6#-C	CS5#-C	CS4#-C
1000000b	PIO	PIO_016	PIO_017	PIO_018	PIO_019	PIO_020	PIO_021	PIO_022	PIO_023
ISEL bit		IRQ8	IRQ9	IRQ15	IRQ3	IRQ12	IRQ5	IRQ2-DS	IRQ3-DS
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.
Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.8 Register Settings for I/O Pin Functions (PORT2, Pins P208 to P215)

PSEL[6:0] Setting	Function	Pin							
		P208	P209	P210	P211	P212	P213	P214	P215
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGA	GTETRGB	GTETRGC	GTETRGD	—	—	—	—
0000010b	GPTW	—	—	—	—	GTCPPO10	GTCPPO11	GTCPPO12	GTCPPO13
0001001b	ADC	—	—	—	—	ADTRG0#	ADTRG1#	ADTRG2#	—
0001010b	RSCI*1	SCK004-C	RXD004-C/ SMISO004-C/SSCL004	TXD004-C/ SMOSI004-C/SSDA004/ TXDA004	CTS004#*2/ RTS004#*2/ SS004#	—	—	—	CTS004#
0001011b	RSCI	CTS004#	—	—	DE004	—	—	—	—
0001100b	RSCI	TXDB004	—	—	TXDB004	—	—	—	—
0001101b	RSCI*1	—	—	—	CTS005#	SCK005-C	RXD005-C/ SMISO005-C/SSCL005	TXD005-C/ SMOSI005-C/SSDA005/ TXDA005	CTS005#*2/ RTS005#*2/ SS005#
0001110b	RSCI	—	—	—	—	CTS005#	—	—	DE005
0001111b	RSCI	—	—	—	—	TXDB005	—	—	TXDB005
0010000b	RSPI*1	RSPCK1-D	MOSI1-D	MISO1-D	SSL10-D	SSL11-D	SSL12-D	SSL13-D	—
0010100b	USBHS	—	—	—	USBHS_OVRCURB	USBHS_OVRCURA	USBHS_VBUSEN	USBHS_EXICEN	USBHS_ID
0010110b	PDMIF	—	—	—	—	—	—	PDMCLK0	PDMDAT0
0011000b	PCIF	PCINA1	PCINB1	PCINZ1	—	—	PCINA0	PCINB0	PCINZ0
0011101b	CMTW	TIC1	TOC1	—	—	—	—	—	—
1000000b	PIO	PIO_048	PIO_049	PIO_050	PIO_051	PIO_052	PIO_053	PIO_054	PIO_055
ISEL bit		IRQ24	IRQ25	IRQ26	IRQ27	IRQ28	IRQ29	IRQ30	IRQ31
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.
Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.9 Register Settings for I/O Pin Functions (PORT3, Pins P300 to P307)

PSEL[6:0] Setting	Function	Pin							
		P300	P301	P302	P303	P304	P305	P306	P307
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	TMS/ SWDIO	TCK/ SWCLK	TDO	TDI	Hi-z	Hi-z	Hi-z	Hi-z
0000001b	GPTW	—	—	—	—	GTETRGD	—	—	—
0000010b	GPTW	GTCPPO10	GTCPPO11	GTCPPO12	GTCPPO13	GTCPPO0	—	—	—
0000101b	TMR	TMO1	TMCI3	TMRI3	TMCI2	TMCI3	—	—	—
0001000b	POEG	—	—	—	—	POEGA	—	—	—
0001001b	ADC	—	—	—	—	ADTRG0#	—	—	—
0001010b	RSCI*1	CTS000#*2/ RTS000#*2/ SS000#	SCK000-A	TXD000-A/ SMOSI000-A/SSDA000/ TXDA000	RXD000-A/ SMISO000-A/SSCL000	CTS000#	—	—	—
0001011b	RSCI	DE000	CTS000#	—	—	—	—	—	—
0001100b	RSCI	TXDB000	TXDB000	—	—	—	—	—	—
0001101b	RSCI*1	CTS001#*2/ RTS001#*2/ SS001#	SCK001-A	TXD001-A/ SMOSI001-A/SSDA001/ TXDA001	RXD001-A/ SMISO001-A/SSCL001	CTS001#	—	—	—
0001110b	RSCI	DE001	CTS001#	—	—	—	—	—	—
0001111b	RSCI	TXDB001	TXDB001	—	—	—	—	—	—
0010000b	RSPI*1	SSL10-C	RSPCK1-C	MOSI1-C	MISO1-C	—	—	—	—
0010011b	USB0	—	—	—	—	USB0_VBUSEN	—	—	—
0010100b	USBHS	—	—	—	—	USBHS_ID	—	—	—
0011111b	CLKOUT	—	—	—	—	ETHPHYCLK	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_WOL	—	—	—	ET0_INT	—	—	—
0100001b	ESWM*1 (RMII)	ET0_WOL	—	—	—	ET0_INT	—	—	—
0100010b	ESWM*1 (RGMII)	ET0_WOL	—	—	—	ET0_INT	—	—	—
0111101b	BUS*1	—	—	—	—	ALE-C	—	—	—
0111110b	BUS*1	—	—	—	—	WAIT#-B	—	—	—
1000000b	PIO	PIO_024	PIO_025	PIO_026	PIO_027	PIO_028	—	—	—
ISEL bit		IRQ6	IRQ7	IRQ4	IRQ1	IRQ0-DS	IRQ0	—	—
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	—	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	—	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.10 Register Settings for I/O Pin Functions (PORT3, Pins P308 to P312, P315)

PSEL[6:0] Setting	Function	Pin					
		P308	P309	P310	P311	P312	P315
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	FINED
0000001b	GPTW	—	—	GTETRGB	GTETRGC	GTETRGD	—
0000010b	GPTW	—	—	GTCPPO1	GTCPPO14	GTCPPO15	—
0000111b	CAC	—	—	CACREF	—	—	—
0001000b	POEG	—	—	POEGC	POEGD	POEGA	—
0001001b	ADC	—	—	ADTRG0#	ADTRG1#	ADTRG2#	—
0001010b	RSCI	—	—	CTS000#	—	—	—
0001101b	RSCI	—	—	CTS001#	CTS007#	—	—
0010000b	RSPI*1	—	—	SSL11-B	SSL12-B	SSL13-B	—
0010111b	SSIE	—	—	SSITXD0	SSIRXD0	SSILRCK0	—
0100000b	ESWM*1 (GMII/MII)	—	—	—	ET0_WOL	—	—
0100001b	ESWM*1 (RMII)	—	—	—	ET0_WOL	—	—
0100010b	ESWM*1 (RGMII)	—	—	—	ET0_WOL	—	—
0111110b	BUS*1	—	—	—	BREQ#-B	BACK#-B	—
1000000b	PIO	—	—	PIO_058	PIO_059	PIO_060	—
ISEL bit		—	—	IRQ18	IRQ19	IRQ20	—
DSCR[3:0] bits	Drive capacity control	—	—	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	—
NCODR bit	N-channel open-drain	—	—	✓	✓	✓	✓
PCR bit	Pull-up resistor	—	—	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Table 22.11 Register Settings for I/O Pin Functions (PORT4, Pins P400 to P407)

PSEL[6:0] Setting	Function	Pin							
		P400	P401	P402	P403	P404	P405	P406	P407
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
1000000b	PIO	PIO_048	PIO_049	PIO_050	PIO_051	PIO_052	PIO_053	PIO_054	PIO_055
ASEL bit		AN000	AN001	AN002	AN003	AN004	AN005	AN006	AN007
ISEL bit		IRQ8-DS	IRQ9-DS	IRQ10-DS	IRQ11-DS	IRQ12-DS	IRQ13-DS	IRQ14-DS	IRQ15-DS
DSCR[3:0] bits	Drive capacity control	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	—	—
100-pin packages		✓	—	✓	—	✓	—	—	—

Table 22.12 Register Settings for I/O Pin Functions (PORT4, Pins P408 to P415)

PSEL[6:0] Setting	Function	Pin							
		P408	P409	P410	P411	P412	P413	P414	P415
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
1000000b	PIO	PIO_056	PIO_057	PIO_058	PIO_059	PIO_060	PIO_061	PIO_062	PIO_063
ASEL bit		AN008	AN009	AN010	AN011	AN012	AN013	AN014/DA0	AN015/DA1
ISEL bit		IRQ24-DS	IRQ25-DS	IRQ26-DS	IRQ27-DS	IRQ28-DS	IRQ29	IRQ30	IRQ31
DSCR[3:0] bits	Drive capacity control	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1	DRV1
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		—	—	—	—	—	—	✓	✓
176-pin packages		—	—	—	—	—	—	✓	✓
144-pin packages		—	—	—	—	—	—	✓	✓
100-pin packages		—	—	—	—	—	—	✓	—

Table 22.13 Register Settings for I/O Pin Functions (PORT5, Pins P500 to P507)

PSEL[6:0] Setting	Function	Pin							
		P500	P501	P502	P503	P504	P505	P506	P507
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	—	—	—	—	GTETRGA	GTETRGB	GTETRGC	GTETRGD
0000011b	GPTW	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC3B	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B
0000101b	TMR	—	—	—	—	TMCI1	TMO3	—	—
0001000b	POEG	—	—	—	—	—	—	POEGD	POEGC
0001001b	ADC	ADST0	ADST1	ADST2	—	—	—	—	—
0001010b	RSCI*1	CTS002#*2/ RTS002#*2/ SS002#	SCK002-C	TXD002-C/ SMOSI002-C/SSDA002/ TXDA002	RXD002-C/ SMISO002-C/SSCL002	CTS002#	—	—	—
0001011b	RSCI*1	DE002	CTS002#	—	—	—	—	—	—
0001100b	RSCI*1	TXDB002	TXDB002	—	—	—	—	—	—
0001101b	RSCI*1	CTS007#	—	—	—	RXD007-B/ SMISO007-B/SSCL007	TXD007-B/ SMOSI007-B/SSDA007/ TXDA007	SCK007-B	CTS007#*2/ RTS007#*2/ SS007#
0001110b	RSCI	—	—	—	—	—	—	CTS007#	DE007
0001111b	RSCI	—	—	—	—	—	—	TXDB007	TXDB007
0010000b	RSPI*1	SSL11-C	SSL12-C	SSL13-C	—	MOSI2-B	MISO2-B	RSPCK2-B	SSL20-B
0010010b	CANFD	CTX0	CRX0	CTX1	CRX1	—	—	—	—
0010011b	USB0*1	—	—	—	USB0_OVR CURB	—	—	—	—
0010111b	SSIE	—	—	—	SSIRXD0	—	—	—	—
0011001b	SDHI*1	SD0_DAT4-A	SD0_DAT5-A	SD0_DAT6-A	SD0_DAT7-A	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_LINKSTA	ET0_INT	ET0_WOL	—	ET0_TX_ER-C	ET0_TX_CLK-C	ET0_TXD2-C	ET0_TXD3-C
0100001b	ESWM*1 (RMII)	ET0_LINKSTA	ET0_INT	ET0_WOL	—	—	—	—	—
0100010b	ESWM*1 (RGMII)	ET0_LINKSTA	ET0_INT	ET0_WOL	—	—	—	—	—
0100011b	ESWM	—	—	—	—	GPTP_CAPTURE0	GPTP_MATCH0	GPTP0_PPS	—
0111101b	BUS*1	WR#-A/ WR0#-A	WR1#-A/ BC1#-A	WR2#-A/ BC2#-A	WR3#-A/ BC3#-A	—	—	—	—
0111110b	BUS*1	A16-C	A17-C	A18-C	A19-C	—	—	—	—
1000000b	PIO	PIO_032	PIO_033	PIO_034	PIO_035	PIO_036	PIO_037	PIO_038	PIO_039
ISEL bit		IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ10	IRQ6	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.14 Register Settings for I/O Pin Functions (PORT5, Pins P508 to P515)

PSEL[6:0] Setting	Function	Pin							
		P508	P509	P510	P511	P512	P513	P514	P515
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	—	—	GTETRGB	GTETRGA	—	—	—	—
0000011b	GPTW	GTIOC6A	GTIOC6B	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B
0001000b	POEG	—	—	POEGB	POEGC	—	—	—	—
0001010b	RSCI	—	—	—	—	—	—	CTS008#	CTS010#
0010000b	RSPI*1	RSPCK2-D	MOSI2-D	MISO2-D	SSL20-D	SSL21-D	SSL22-D	SSL23-D	—
0011000b	PCIF	PCOUTA0	PCOUTB0	PCOUTZ0	—	PCINZ0	PCINZ1	—	—
0011001b	SDHI*1	SD0_DAT2-D	SD0_DAT3-D	SD0_CMD-D	SD0_CLK-D	SD0_DAT0-D	SD0_DAT1-D	SD0_WP	SD0_CD
0011101b	CMTW	TIC2	TOC2	—	—	—	—	—	—
0011111b	CLKOUT	—	—	CLKOUT	—	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_RXD4-A	ET0_RXD5-A	ET0_RXD6-A	ET0_RXD7-A	ET0_TXD4-A	ET0_TXD5-A	ET0_TXD6-A	ET0_TXD7-A
0111101b	BUS*1	D23-A	D22-A	D21-A	D20-A	D19-A	D18-A	D17-A	D16-A
1000000b	PIO	PIO_000	PIO_001	PIO_002	PIO_003	PIO_004	PIO_005	PIO_006	PIO_007
ISEL bit		IRQ16	IRQ17	IRQ18	IRQ19	IRQ20	IRQ21	IRQ22	IRQ23
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		—	—	—	—	—	—	—	—
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Table 22.15 Register Settings for I/O Pin Functions (PORT6, Pins P600 to P607)

PSEL[6:0] Setting	Function	Pin							
		P600	P601	P602	P603	P604	P605	P606	P607
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	—	—	—	—	—	—	GTETRGA	GTETRGB
0000010b	GPTW	—	—	—	—	—	—	GTCPPO4	GTADSM0
0000011b	GPTW	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B	GTIOC15A	GTIOC15B
0001000b	POEG	—	—	—	—	—	—	POEGC	POEGA
0001001b	ADC	ADTRG0#	ADTRG1#	ADTRG2#	—	—	—	—	—
0001010b	RSCI*1	TXD008-B/ SMOSI008-B/SSDA008/ TXDA008	RXD008-B/ SMISO008-B/SSCL008	SCK008-B	CTS008#*2/ RTS008#*2/ SS008#	—	—	—	CTS008#
0001011b	RSCI	—	—	CTS008#	DE008	—	—	—	—
0001100b	RSCI	—	—	TXDB008	TXDB008	—	—	—	—
0001101b	RSCI*1	—	—	—	CTS009#	TXD009-B/ SMOSI009-B/SSDA009/ TXDA009	RXD009-B/ SMISO009-B/SSCL009	SCK009-B	CTS009#*2/ RTS009#*2/ SS009#
0001110b	RSCI	—	—	—	—	—	—	CTS009#	DE009
0001111b	RSCI	—	—	—	—	—	—	TXDB009	TXDB009
0010000b	RSPI*1	SSL00-D	RSPCK0-D	MOSI0-D	MISO0-D	SSL01-D	SSL02-D	SSL03-D	—
0010010b	CANFD	—	—	—	—	—	—	CTX2	CRX2
0011010b	XSPI*1	XSPI0_IO4	XSPI0_IO5	XSPI0_IO6	XSPI0_IO7	XSPI0_CS1#	XSPI0_INT0#	XSPI0_RESET0#	XSPI0_RSTO0#
0011011b	XSPI*1	—	—	—	—	XSPI0_RSTO0#	XSPI0_ECS0#	—	—
0011100b	PDC	PIXCLK	VSYNC	HSYNC	PCKO	—	—	—	—
0100011b	ESWM	GPTP_PTPOUT0	GPTP_PTPOUT1	—	—	—	—	—	—
0111101b	BUS*1	SDCS#-B	RAS#-B	CAS#-B	WE#-B	CS3#-C	CS2#-C	CS1#-C	CS0#-B
0111110b	BUS*1	D23-B	D22-B	D21-B	D20-B	D19-B	D18-B	D17-B	D16-B
1000000b	PIO	PIO_008	PIO_009	PIO_010	PIO_011	PIO_012	PIO_013	PIO_014	PIO_015
ISEL bit		IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ13	IRQ14	IRQ15
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	—	—
100-pin packages		✓	✓	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.16 Register Settings for I/O Pin Functions (PORT7, Pins P700 to P707)

PSEL[6:0] Setting	Function	Pin							
		P700	P701	P702	P703	P704	P705	P706	P707
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	—	—	GTETRGA	GTETRGB	—	—	—	—
0000011b	GPTW	GTIOC10A	GTIOC10B	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B
0000111b	CAC	—	—	—	CACREF	—	—	—	—
0001001b	ADC	ADST0	ADST1	ADST2	—	ADCNV0	ADCNV1	ADCNV2	—
0001010b	RSCI*1	TXD004-B/ SMOSI004-B/SSDA004/ TXDA004	RXD004-B/ SMISO004-B/SSCL004	SCK004-B	CTS004#*2/ RTS004#*2/ SS004#	CTS004#	—	—	—
0001011b	RSCI	—	—	CTS004#	DE004	—	—	—	—
0001100b	RSCI	—	—	TXDB004	TXDB004	—	—	—	—
0001101b	RSCI*1	—	—	—	—	CTS011#*2/ RTS011#*2/ SS011#	SCK011-B	RXD011-B/ SMISO011-B/SSCL011	TXD011-B/ SMOSI011-B/SSDA011/ TXDA011
0001110b	RSCI	—	—	—	—	DE011	CTS011#	—	—
0001111b	RSCI	—	—	—	—	TXDB011	TXDB011	—	—
0011010b	XSPI*1	XSPI1_INT0#	XSPI1_RSTO0#	XSPI1_WP0#	XSPI1_RESET0#	XSPI1_INT1#	XSPI1_RSTO1#	XSPI1_WP1#	XSPI1_RESET1#
0011011b	XSPI*1	XSPI1_ECS0#	XSPI1_RESET0#	XSPI1_RESET1#	—	XSPI1_ECS1#	—	—	—
0011100b	PDC	PIXCLK	VSYNC	HSYNC	PCKO	—	—	—	—
0011111b	CLKOUT	ETHPHYCLK	—	—	—	—	—	—	—
0111101b	BUS*1	SDCLK-A	CKE-A	RD#-A	ALE-A	SDCS#-A	RAS#-A	CAS#-A	WE#-A
0111110b	BUS*1	—	WR3#-B/ BC3#-B	DQM3-B	WR2#-B/ BC2#-B	CS4#-A	CS5#-A	CS6#-A	CS7#-A
1000000b	PIO	PIO_048	PIO_049	PIO_050	PIO_051	PIO_052	PIO_053	PIO_054	PIO_055
ISEL bit		IRQ0	IRQ1	IRQ10	IRQ8	IRQ12	IRQ13	IRQ14	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/7	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.17 Register Settings for I/O Pin Functions (PORT7, Pins P708 to P715)

PSEL[6:0] Setting	Function	Pin							
		P708	P709	P710	P711	P712	P713	P714	P715
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGC	GTETRGD	GTETRGA	GTETRGB	—	—	—	—
0000010b	GPTW	GTCPPO4	GTCPPO7	GTADSM0	GTADSM1	GTCPPO5	GTCPPO6	GTCPPO8	GTCPPO9
0000011b	GPTW	GTIOC10A	GTIOC10B	GTIOC11A	GTIOC11B	GTIOC14A	GTIOC14B	GTIOC15A	GTIOC15B
0001010b	RSCI*1	—	—	—	CTS010#	CTS010#*2/ RTS010#*2/ SS010#	SCK010-C	TXD010-C/ SMOSI010-C/SSDA010/ TXDA010	RXD010-C/ SMISO010-C/SSCL010
0001011b	RSCI	—	—	—	—	DE010	CTS010#	—	—
0001100b	RSCI	—	—	—	—	TXDB010	TXDB010	—	—
0001101b	RSCI*1	SCK011-C	RXD011-C/ SMISO011-C/SSCL011	TXD011-C/ SMOSI011-C/SSDA011/ TXDA011	CTS011#*2/ RTS011#*2/ SS011#	CTS011#	—	—	—
0001110b	RSCI*1	CTS011#	—	—	DE011	—	—	—	—
0001111b	RSCI*1	TXDB011	—	—	TXDB011	—	—	—	—
0011010b	XSPI*1	XSPI1_IO4	XSPI1_IO5	XSPI1_IO6	XSPI1_IO7	—	—	—	—
0011100b	PDC	—	—	—	—	PCKO	PIXCLK	VSYNC	HSYNC
0100000b	ESWM*1 (GMII/MII)	ET0_INT	—	—	—	—	—	—	—
0100001b	ESWM*1 (RMII)	ET0_INT	—	—	—	—	—	—	—
0100010b	ESWM*1 (RGMII)	ET0_INT	—	—	—	—	—	—	—
0111101b	BUS*1	A19-A	A18-A	A17-A	A16-A	A15-A	A14-A	A13-A	A12-A
1000000b	PIO	PIO_016	PIO_017	PIO_018	PIO_019	PIO_020	PIO_021	PIO_022	PIO_023
ISEL bit		IRQ16	IRQ17	IRQ18	IRQ19	IRQ20	IRQ21	IRQ22	IRQ23
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/6/ 8	DRV1/2/3/6/ 8	DRV1/2/3/6/ 8	DRV1/2/3/6/ 8	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		—	—	—	—	—	—	—	—
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.18 Register Settings for I/O Pin Functions (PORT8, Pins P800 to P807)

PSEL[6:0] Setting	Function	Pin							
		P800	P801	P802	P803	P804	P805	P806	P807
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTIU	GTIV	GTIW	—	—	—	—	—
0000010b	GPTW	—	—	—	—	GTADSM0	GTADSM1	GTCPPO7	GTCPPO4
0000011b	GPTW	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC15A	GTIOC15B	GTIOC14A	GTIOC14B
0001000b	POEG	POEGD	POEGC	POEGB	POEGA	—	—	—	POEGD
0001001b	ADC	ADCNV0	ADCNV1	ADCNV2	ADSYNC	—	ADSYNC	—	—
0001010b	RSCI*1	RXD010-B/ SMISO010-B/SSCL010	TXD010-B/ SMOSI010-B/SSDA010/ TXDA010	SCK010-B	CTS010#*2/ RTS010#*2/ SS010#	—	—	—	—
0001011b	RSCI	—	—	CTS010#	DE010	—	—	—	—
0001100b	RSCI	—	—	TXDB010	TXDB010	—	—	—	—
0001101b	RSCI*1	—	—	—	—	CTS005#*2/ RTS005#*2/ SS005#	SCK005-B	RXD005-B/ SMISO005-B/SSCL005	TXD005-B/ SMOSI005-B/SSDA005/ TXDA005
0001110b	RSCI	—	—	—	—	DE005	CTS005#	—	—
0001111b	RSCI	—	—	—	—	TXDB005	TXDB005	—	—
0010000b	RSPI*1	—	—	—	—	SSL21-B	SSL22-B	SSL23-B	—
0010010b	CANFD	—	—	CRX2	CTX2	CTX0	CRX0	CTX1	CRX1
0010011b	USB0	—	—	USB0_EXICEN	USB0_ID	USB0_OVRCURA	USB0_VBUSEN	—	—
0010100b	USBHS	—	—	—	—	USBHS_OVRCURA	USBHS_VBUSEN	—	—
0011000b	PCIF	—	—	—	—	PCINA0	PCINB0	PCINA1	PCINB1
0011100b	PDC	—	—	—	—	PCKO	HSYNC	VSYNC	PIXCLK
0011111b	CLKOUT	—	CLKOUT	—	—	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_COL	ET0_CRS	ET0_MDIO-B	ET0_MDC-B	ET0_TX_EN-C	ET0_TXD0-C	ET0_TXD1-C	ET0_RX_ER-C
0100001b	ESWM*1 (RMII)	—	—	ET0_MDIO-B	ET0_MDC-B	RMII0_TX_EN-C	RMII0_TXD0-C	RMII0_TXD1-C	RMII0_RX_ER-C
0100010b	ESWM*1 (RGMII)	—	—	ET0_MDIO-B	ET0_MDC-B	—	—	—	—
0111101b	BUS*1	CS1#-A	EBCLK-A	CS2#-A	CS3#-A	—	—	—	—
0111110b	BUS*1	DQM0-B	—	WR#-B/ WR0#-B	WR1#-B/ BC1#-B	—	—	—	—
1000000b	PIO	PIO_056	PIO_057	PIO_058	PIO_059	PIO_060	PIO_061	PIO_062	PIO_063
ISEL bit		IRQ8	IRQ9	IRQ2	IRQ3	IRQ4	IRQ5	IRQ14	IRQ15
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/7	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	—	—
144-pin packages		✓	✓	✓	✓	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.19 Register Settings for I/O Pin Functions (PORT9, Pins P900 to P907)

PSEL[6:0] Setting	Function	Pin							
		P900	P901	P902	P903	P904	P905	P906	P907
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGA	GTETRGB	GTETRGC	GTETRGD	—	—	—	—
0000010b	GPTW	GTCPPO5	GTCPPO6	GTCPPO8	GTCPPO9	GTCPPO0	GTCPPO1	GTCPPO2	GTCPPO3
0001000b	POEG	—	—	—	POEGD	POEGC	POEGB	POEGA	POEGD
0001010b	RSCI*1	—	—	—	CTS006#	TXD006-B/ SMOSI006-B/SSDA006/ TXDA006	RXD006-B/ SMISO006-B/SSCL006	SCK006-B	CTS006#*2/ RTS006#*2/ SS006#
0001011b	RSCI	—	—	—	—	—	—	CTS006#	DE006
0001100b	RSCI	—	—	—	—	—	—	TXDB006	TXDB006
0001101b	RSCI*1	RXD001-C/ SMISO001-C/SSCL001	TXD001-C/ SMOSI001-C/SSDA001/ TXDA001	SCK001-C	CTS001#*2/ RTS001#*2/ SS001#	—	—	—	CTS001#
0001110b	RSCI	—	—	CTS001#	DE001	—	—	—	—
0001111b	RSCI	—	—	TXDB001	TXDB001	—	—	—	—
0011010b	XSPI*1	XSPI0_INT1#	XSPI0_RSTO1#	XSPI0_RESET1#	XSPI0_WP1#	—	—	—	XSPI0_WP0#
0011011b	XSPI*1	XSPI0_ECS1#	XSPI0_RESET0#	—	—	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	—	—	—	—	ET0_COL	ET0_CRS	ET0_MDC-C	ET0_MDIO-C
0100001b	ESWM*1 (RMII)	—	—	—	—	—	—	ET0_MDC-C	ET0_MDIO-C
0100010b	ESWM*1 (RGMII)	—	—	—	—	—	—	ET0_MDC-C	ET0_MDIO-C
0100011b	ESWM	—	—	—	—	GPTP_PTPOUT0	GPTP_PTPOUT1	—	—
1000000b	PIO	PIO_000	PIO_001	PIO_002	PIO_003	PIO_004	PIO_005	PIO_006	PIO_007
ASEL bit		AN032	AN033	AN034	AN035	—	—	—	—
ISEL bit		IRQ0	IRQ9	IRQ10	IRQ11	IRQ12	IRQ13	IRQ14	IRQ15
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	—	—	—	—
144-pin packages		✓	✓	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.20 Register Settings for I/O Pin Functions (PORTA, Pins PA00 to PA07)

PSEL[6:0] Setting	Function	Pin							
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTIU	GTIV	GTIW	—	—	—	—	—
0000011b	GPTW	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC3B
0000100b	GPTW	GTIOC10A	GTIOC10B	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B
0000101b	TMR	—	—	—	—	TMRI0	—	TMCI3	—
0000111b	CAC	CACREF	—	—	—	—	—	—	—
0001001b	ADC	ADCNV0	ADCNV1	ADCNV2	ADSYNC	—	—	—	—
0001010b	RSCI*1	CTS010#	—	—	—	CTS010#*2/ RTS010#*2/ SS010#	SCK010-A	TXD010-A/ SMOSI010-A/SSDA010/ TXDA010	RXD010-A/ SMISO010-A/SSCL010
0001011b	RSCI	—	—	—	—	DE010	CTS010#	—	—
0001100b	RSCI	—	—	—	—	TXDB010	TXDB010	—	—
0001101b	RSCI*1	CTS005#*2/ RTS005#*2/ SS005#	SCK005-A	TXD005-A/ SMOSI005-A/SSDA005/ TXDA005	RXD005-A/ SMISO005-A/SSCL005	—	—	—	—
0001110b	RSCI	DE005	CTS005#	—	—	—	—	—	—
0001111b	RSCI	TXDB005	TXDB005	—	—	—	—	—	—
0010000b	RSPI*1	SSL01-B	SSL02-B	SSL03-B	—	SSL00-B	RSPCK0-B	MOSI0-B	MISO0-B
0011000b	PCIF	—	—	—	—	PCINZ0	PCINA0	PCINZ1	PCINB0
0011111b	CLKOUT	ETHPHYCLK	—	—	—	—	—	—	ETHPHYCLK
0111101b	BUS*1	A7-A	A6-A	A5-A	A4-A	A3-A	A2-A	A1-A	A0-A/BC0#-A
0111110b	BUS*1	—	—	—	—	—	—	DQM2-B	DQM1-B
1000000b	PIO	PIO_008	PIO_009	PIO_010	PIO_011	PIO_012	PIO_013	PIO_014	PIO_015
ISEL bit		IRQ1-DS	IRQ11	IRQ10	IRQ6-DS	IRQ5-DS	IRQ5	IRQ14	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.21 Register Settings for I/O Pin Functions (PORTB, Pins PB00 to PB07)

PSEL[6:0] Setting	Function	Pin							
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGA	GTETRGB	GTOUUP	GTOULO	GTOVUP	GTOVLO	GTOWUP	GTOWLO
0000010b	GPTW	GTADSM0	GTCPPO4	—	—	—	—	—	—
0000011b	GPTW	GTIOC14A	GTIOC14B	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B
0000101b	TMR	—	TMCI0	—	TMO0	—	TMRI1	—	—
0001000b	POEG	POEGC	POEGA	—	—	—	—	—	—
0001001b	ADC*1	ADST2	—	—	—	—	—	—	—
0001010b	RSCI*1	RXD004-A/ SMISO004-A/SSCL004	TXD004-A/ SMOSI004-A/SSDA004/ TXDA004	CTS004#*2/ RTS004#*2/ SS004#	SCK004-A	CTS004#	—	—	—
0001011b	RSCI	—	—	DE004	CTS004#	—	—	—	—
0001100b	RSCI	—	—	TXDB004	TXDB004	—	—	—	—
0001101b	RSCI*1	—	—	CTS009#	—	CTS009#*2/ RTS009#*2/ SS009#	SCK009-A	RXD009-A/ SMISO009-A/SSCL009	TXD009-A/ SMOSI009-A/SSDA009/ TXDA009
0001110b	RSCI	—	—	—	—	DE009	CTS009#	—	—
0001111b	RSCI	—	—	—	—	TXDB009	TXDB009	—	—
0011000b	PCIF	PCINA1	PCINB1	—	—	—	—	—	—
0100000b	ESWM*1 (GMII/MII)	ET0_RXD0-A	ET0_RXD1-A	ET0_RXD2-A	ET0_RXD3-A	ET0_RX_CLK-A	ET0_RX_DV-A	ET0_RX_ER-A	ET0_LINKSTA
0100001b	ESWM*1 (RMII)	RMII0_RXD0-A	RMII0_RXD1-A	—	—	RMII0_REF50CK-A	RMII0_CRS_DV-A	RMII0_RX_ER-A	ET0_LINKSTA
0100010b	ESWM*1 (RGMII)	RGMII0_RXD0-A	RGMII0_RXD1-A	RGMII0_RXD2-A	RGMII0_RXD3-A	RGMII0_RXC-A	RGMII0_RX_CTL-A	—	ET0_LINKSTA
0111101b	BUS*1	D15-A	D14-A	D13-A	D12-A	D11-A	D10-A	D9-A	D8-A
1000000b	PIO	PIO_016	PIO_017	PIO_018	PIO_019	PIO_020	PIO_021	PIO_022	PIO_023
ISEL bit		IRQ12	IRQ4-DS	IRQ2	IRQ3	IRQ4	IRQ13	IRQ6	IRQ15
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.22 Register Settings for I/O Pin Functions (PORTC, Pins PC00 to PC07)

PSEL[6:0] Setting	Function	Pin							
		PC00	PC01	PC02	PC03	PC04	PC05	PC06	PC07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTOUUP	GTOULO	GTOVUP	GTOVLO	GTOWUP	GTOWLO	GTETRGC	GTETRGD
0000010b	GPTW	—	—	—	—	—	—	GTCPPO7	GTADSM1
0000011b	GPTW	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B	GTIOC15A	GTIOC15B
0000101b	TMR	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
0000111b	CAC	—	—	—	—	—	—	—	CACREF
0001000b	POEG	—	—	—	—	—	—	POEGB	POEGA
0001010b	RSCI*1	—	—	—	—	CTS008#*2/ RTS008#*2/ SS008#	SCK008-A	RXD008-A/ SMISO008-A/SSCL008	TXD008-A/ SMOSI008-A/SSDA008/ TXDA008
0001011b	RSCI	—	—	—	—	DE008	CTS008#	—	—
0001100b	RSCI	—	—	—	—	TXDB008	TXDB008	—	—
0001101b	RSCI*1	CTS011#*2/ RTS011#*2/ SS011#	SCK011-A	RXD011-A/ SMISO011-A/SSCL011	TXD011-A/ SMOSI011-A/SSDA011/ TXDA011	CTS011#	—	—	—
0001110b	RSCI	DE011	CTS011#	—	—	—	—	—	—
0001111b	RSCI	TXDB011	TXDB011	—	—	—	—	—	—
0010000b	RSPI*1	SSL01-A	SSL02-A	SSL03-A	—	SSL00-A	RSPCK0-A	MOSI0-A	MISO0-A
0010001b	RI3C*1/RIICHS*1	SCL00-C	SDA00-C	SCLHS1-C	SDAHS1-C	—	—	—	—
0010110b	PDMIF	—	—	—	—	—	—	PDMCLK0	PDMDAT0
0010111b	SSIE	—	—	—	—	AUDIO_CLK	SSIBCK0	SSILRCK0	SSITXD0
0011000b	PCIF	PCOUTA0	PCOUTB0	PCOUTZ0	PCOUTA1	PCOUTB1	PCOUTZ1	—	—
0011001b	SDHI*1	SD0_DAT2-A	SD0_DAT3-A	SD0_CMD-A	SD0_CLK-A	SD0_DAT0-A	SD0_DAT1-A	SD0_WP	SD0_CD
0011101b	CMTW	—	—	—	—	—	—	TIC0	TOC0
0011111b	CLKOUT	—	—	—	—	—	—	—	ETHPHYCLK
0100000b	ESWM*1 (GMII/MII)	ET0_TX_CLK-A	ET0_TXD0-A	ET0_TXD1-A	ET0_TXD2-A	ET0_TXD3-A	ET0_TX_EN-A	ET0_TX_ER-A	ET0_GTX_CLK-A
0100001b	ESWM*1 (RMII)	—	RMII0_TXD0-A	RMII0_TXD1-A	—	—	RMII0_TX_EN-A	—	—
0100010b	ESWM*1 (RGMII)	—	RGMII0_TXD0-A	RGMII0_TXD1-A	RGMII0_TXD2-A	RGMII0_TXD3-A	RGMII0_TX_CTL-A	—	RGMII0_TXC-A
0111101b	BUS*1	D7-A	D6-A	D5-A	D4-A	D3-A	D2-A	D1-A	D0-A
1000000b	PIO	PIO_024	PIO_025	PIO_026	PIO_027	PIO_028	PIO_029	PIO_030	PIO_031
ISEL bit		IRQ14	IRQ12	IRQ10	IRQ11	IRQ12	IRQ5	IRQ13	IRQ14
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7	DRV1/2/3/4/ 5/6/7
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.23 Register Settings for I/O Pin Functions (PORTD, Pins PD00 to PD07)

PSEL[6:0] Setting	Function	Pin							
		PD00	PD01	PD02	PD03	PD04	PD05	PD06	PD07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTETRGD	GTETRGC	GTETRGB	—	GTETRGA	GTIU	GTIV	GTIW
0000011b	GPTW	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC3B
0001000b	POEG	POEGD	POEGC	POEGB	POEGA	POEGC	POEGD	POEGB	POEGA
0001010b	RSCI*1	CTS002#*2/ RTS002#*2/ SS002#	SCK002-A	RXD002-A/ SMISO002-A/SSCL002	TXD002-A/ SMOSI002-A/SSDA002/ TXDA002	CTS002#	—	—	—
0001011b	RSCI	DE002	CTS002#	—	—	—	—	—	—
0001100b	RSCI	TXDB002	TXDB002	—	—	—	—	—	—
0001101b	RSCI*1	CTS003#	—	—	—	CTS003#*2/ RTS003#*2/ SS003#	SCK003-A	RXD003-A/ SMISO003-A/SSCL003	TXD003-A/ SMOSI003-A/SSDA003/ TXDA003
0001110b	RSCI	—	—	—	—	DE003	CTS003#	—	—
0001111b	RSCI	—	—	—	—	TXDB003	TXDB003	—	—
0010000b	RSPI*1	SSL20-A	RSPCK2-A	MOSI2-A	MISO2-A	SSL21-A	SSL22-A	SSL23-A	—
0010010b	CANFD	—	—	—	—	CRX0	CTX0	CRX1	CTX1
0011001b	SDHI*1	SD0_DAT2-B	SD0_DAT3-B	SD0_CMD-B	SD0_CLK-B	SD0_DAT0-B	SD0_DAT1-B	SD0_WP	SD0_CD
0011010b	XSPI*1	XSPI0_CKN#	XSPI0_CK	XSPI0_IO0	XSPI0_IO1	XSPI0_IO2	XSPI0_IO3	XSPI0_DS	XSPI0_CS0#
0011011b	XSPI*1	XSPI0_CS1#	—	—	—	XSPI0_WP0#	—	—	XSPI0_RESET1#
0011100b	PDC	PIXD0	PIXD1	PIXD2	PIXD3	PIXD4	PIXD5	PIXD6	PIXD7
0011101b	CMTW	—	—	TIC2	TOC2	—	—	—	—
0111101b	BUS*1	A23-B	A22-B	A21-B	A20-B	A19-B	A18-B	A17-B	A16-B
0111110b	BUS*1	D31-B	D30-B	D29-B	D28-B	D27-B	D26-B	D25-B	D24-B
1000000b	PIO	PIO_032	PIO_033	PIO_034	PIO_035	PIO_036	PIO_037	PIO_038	PIO_039
ASEL bit		AN024	AN025	AN026	AN027	AN028	AN029	AN030	AN031
ISEL bit		IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	✓

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.24 Register Settings for I/O Pin Functions (PORTE, Pins PE00 to PE07)

PSEL[6:0] Setting	Function	Pin							
		PE00	PE01	PE02	PE03	PE04	PE05	PE06	PE07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000001b	GPTW	GTOUUP	GTOULO	GTOVUP	GTOVLO	GTOWUP	GTOWLO	GTETRGC	GTETRGD
0000010b	GPTW	—	—	—	—	—	—	GTADSM1	GTCPPO7
0000011b	GPTW	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B	GTIOC10A	GTIOC10B
0001000b	POEG	—	—	—	—	—	—	POEGB	POEGD
0001010b	RSCI*1	SCK012-A	TXD012-A/ SMOSI012-A/SSDA012/ TXDA012	RXD012-A/ SMISO012-A/SSCL012	CTS012#*2/ RTS012#*2/ SS012#	CTS012#	—	—	—
0001011b	RSCI	CTS012#	—	—	DE012	—	—	—	—
0001100b	RSCI	TXDB012	—	—	TXDB012	—	—	—	—
0010000b	RSPI*1	SSL10-A	RSPCK1-A	MOSI1-A	MISO1-A	SSL11-A	SSL12-A	SSL13-A	—
0011001b	SDHI*1	SD0_DAT4-B	SD0_DAT5-B	SD0_DAT6-B	SD0_DAT7-B	—	—	—	—
0011010b	XSPI*1	XSPI1_CKN#	XSPI1_CK	XSPI1_IO0	XSPI1_IO1	XSPI1_IO2	XSPI1_IO3	XSPI1_DS	XSPI1_CS0#
0011011b	XSPI*1	XSPI1_CS1#	—	—	—	XSPI1_WP0#	—	—	—
0011100b	PDC	PIXD0	PIXD1	PIXD2	PIXD3	PIXD4	PIXD5	PIXD6	PIXD7
0011101b	CMTW	—	—	TIC3	TOC3	—	—	TIC1	TOC1
0011111b	CLKOUT	ETHPHYCLK	CLKOUT	—	—	—	—	—	—
0111101b	BUS*1	A15-B	A14-B	A13-B	A12-B	A11-B	A10-B	A9-B	A8-B
0111110b	BUS*1	—	—	—	—	A23-D	A22-D	A21-D	A20-D
1000000b	PIO	PIO_040	PIO_041	PIO_042	PIO_043	PIO_044	PIO_045	PIO_046	PIO_047
ISEL bit		IRQ8	IRQ9	IRQ7-DS	IRQ11	IRQ12	IRQ5	IRQ6	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/6/8	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
144-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
100-pin packages		✓	✓	✓	✓	✓	✓	✓	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.25 Register Settings for I/O Pin Functions (PORTF, Pins PF00 to PF07)

PSEL[6:0] Setting	Function	Pin							
		PF00	PF01	PF02	PF03	PF04	PF05	PF06	PF07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO14	GTCPPO15	GTCPPO1	GTCPPO2	GTCPPO3	GTCPPO5	GTCPPO6	GTCPPO11
0000011b	GPTW	—	—	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B
0001000b	POEG	—	—	—	—	—	POEGD	POEGC	—
0001001b	ADC	—	—	ADSYNC	—	—	—	—	—
0001010b	RSCI*1	—	—	RXD000-C/ SMISO000-C/SSCL000	TXD000-C/ SMOSI000-C/SSDA000/ TXDA000	SCK000-C	—	CTS000#*2/ RTS000#*2/ SS000#	—
0001011b	RSCI	—	—	—	—	CTS000#	—	DE000	—
0001100b	RSCI	—	—	—	—	TXDB000	—	TXDB000	—
0001101b	RSCI*1	—	—	RXD007-C/ SMISO007-C/SSCL007	TXD007-C/ SMOSI007-C/SSDA007/ TXDA007	SCK007-C	CTS007#*2/ RTS007#*2/ SS007#	CTS007#	—
0001110b	RSCI	—	—	—	—	CTS007#	DE007	—	—
0001111b	RSCI	—	—	—	—	TXDB007	TXDB007	—	—
0010000b	RSPI*1	—	—	—	—	—	SSL01-C	SSL02-C	SSL03-C
0010011b	USB0	USB0_DP	USB0_DM	—	—	—	—	—	—
0011010b	XSPI*1	—	—	—	—	—	—	XSPI1_CS1#	XSPI1_RSTO0#
0011011b	XSPI*1	—	—	—	—	—	—	XSPI1_WP0#	—
0111101b	BUS*1	—	—	CS6#-B	CS7#-B	CS4#-B	CS5#-B	A23-A	A22-A
0111110b	BUS*1	—	—	BREQ#-C	BACK#-C	—	—	A19-D	A18-D
1000000b	PIO	PIO_048	PIO_049	PIO_050	PIO_051	PIO_052	PIO_053	PIO_054	PIO_055
ISEL bit		IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	—	—	—	—	—	—
144-pin packages		✓	✓	—	—	—	—	—	—
100-pin packages		✓	✓	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.26 Register Settings for I/O Pin Functions (PORTF, Pins PF08 to PF09)

PSEL[6:0] Setting	Function	Pin	
		PF08	PF09
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z	
0000010b	GPTW	GTCPPO12	—
0001000b	POEG	POEGA	—
0001010b	RSCI	—	CTS012#
0011010b	XSPI*1	XSPI1_RESET0#	XSPI1_INT0#
0011011b	XSPI*1	—	XSPI1_ECS0#
0100011b	ESWM	—	GPTP_CAPTURE1
0111101b	BUS*1	A21-A	A20-A
0111110b	BUS*1	A17-D	A16-D
1000000b	PIO	PIO_016	PIO_017
ISEL bit		IRQ16	IRQ17
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓
PCR bit	Pull-up resistor	✓	✓
320-pin packages		✓	✓
224-pin packages		✓	✓
176-pin packages		—	—
144-pin packages		—	—
100-pin packages		—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Table 22.27 Register Settings for I/O Pin Functions (PORTG, Pins PG00 to PG07)

PSEL[6:0] Setting	Function	Pin							
		PG00	PG01	PG02	PG03	PG04	PG05	PG06	PG07
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO0	GTCPPO1	—	—	—	—	—	—
0000011b	GPTW	—	—	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B
0001010b	RSCI*1	SCK008-C	RXD008-C/ SMISO008-C/SSCL008	TXD008-C/ SMOSI008-C/SSDA008/ TXDA008	CTS008#*2/ RTS008#*2/ SS008#	—	—	—	—
0001011b	RSCI	CTS008#	—	—	DE008	—	—	—	—
0001100b	RSCI	TXDB008	—	—	TXDB008	—	—	—	—
0001101b	RSCI*1	—	—	—	—	SCK003-C	RXD003-C/ SMISO003-C/SSCL003	TXD003-C/ SMOSI003-C/SSDA003/ TXDA003	CTS003#*2/ RTS003#*2/ SS003#
0001110b	RSCI	—	—	—	—	CTS003#	—	—	DE003
0001111b	RSCI	—	—	—	—	TXDB003	—	—	TXDB003
0100000b	ESWM*1 (GMII/MII)	—	—	—	ET0_INT	ET0_MDC-A	ET0_MDIO-A	ET0_COL	ET0_CRS
0100001b	ESWM*1 (RMII)	—	—	—	ET0_INT	ET0_MDC-A	ET0_MDIO-A	—	—
0100010b	ESWM*1 (RGMII)	—	—	—	ET0_INT	ET0_MDC-A	ET0_MDIO-A	—	—
0100011b	ESWM	—	—	—	GPTP_MATCH1	—	—	—	—
0111101b	BUS*1	DQM1-A	DQM0-A	DQM3-A	DQM2-A	A11-A	A10-A	A9-A	A8-A
0111110b	BUS*1	WR1#-C/ BC1#-C	WR#-C/ WR0#-C	WR3#-C/ BC3#-C	WR2#-C/ BC2#-C	—	—	—	—
1000000b	PIO	PIO_056	PIO_057	PIO_058	PIO_059	PIO_060	PIO_061	PIO_062	PIO_063
ISEL bit		IRQ8	IRQ9	IRQ10	IRQ11	IRQ12	IRQ13	IRQ14	IRQ15
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
176-pin packages		✓	✓	✓	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

Table 22.28 Register Settings for I/O Pin Functions (PORTG, Pins PG08 to PG15)

PSEL[6:0] Setting	Function	Pin							
		PG08	PG09	PG10	PG11	PG12	PG13	PG14	PG15
0000000b (Value after reset)	Hi-Z/JTAG/SWD/ FINE	Hi-Z							
0000010b	GPTW	GTCPPO10	GTCPPO11	GTCPPO12	GTCPPO13	GTCPPO12	GTCPPO13	GTCPPO14	GTCPPO15
0001010b	RSCI*1	SCK012-C	RXD012-C/ SMISO012-C/SSCL012	TXD012-C/ SMOSI012-C/SSDA012/ TXDA012	CTS012#*2/ RTS012#*2/ SS012#	CTS008#	—	—	—
0001011b	RSCI	CTS012#	—	—	DE012	—	—	—	—
0001100b	RSCI	TXDB012	—	—	TXDB012	—	—	—	—
0001101b	RSCI	—	—	—	—	CTS005#	CTS003#	—	CTS011#
0011000b	PCIF	—	PCOUTA1	PCOUTB1	PCOUTZ1	PCOUTA0	PCOUTB0	PCOUTZ0	—
0011001b	SDHI*1	—	—	—	—	SD0_DAT4-D	SD0_DAT5-D	SD0_DAT6-D	SD0_DAT7-D
0011010b	XSPI*1	—	—	—	—	XSPI1_INT1#	XSPI1_RESET1#	XSPI1_WP1#	XSPI1_RSTO1#
0011011b	XSPI*1	—	—	—	—	XSPI1_ECS1#	—	—	—
0011100b	PDC	PIXD0	PIXD1	PIXD2	PIXD3	PIXD4	PIXD5	PIXD6	PIXD7
0011101b	CMTW	TIC3	TOC3	—	—	—	—	—	—
0100011b	ESWM	—	—	—	—	GPTP_PTPOUT0	GPTP_PTPOUT1	—	—
0111101b	BUS*1	D31-A	D30-A	D29-A	D28-A	D27-A	D26-A	D25-A	D24-A
1000000b	PIO	PIO_024	PIO_025	PIO_026	PIO_027	PIO_028	PIO_029	PIO_030	PIO_031
ISEL bit		IRQ24	IRQ25	IRQ26	IRQ27	IRQ28	IRQ29	IRQ30	IRQ31
DSCR[3:0] bits	Drive capacity control	DRV1/2/3/4/5/6/7	DRV1/2/3/4/5/6/7	DRV1/2/3/4/5/6/7	DRV1/2/3/4/5/6/7	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4	DRV1/2/3/4
NCODR bit	N-channel open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up resistor	✓	✓	✓	✓	✓	✓	✓	✓
320-pin packages		✓	✓	✓	✓	✓	✓	✓	✓
224-pin packages		—	—	—	—	—	—	—	—
176-pin packages		—	—	—	—	—	—	—	—
144-pin packages		—	—	—	—	—	—	—	—
100-pin packages		—	—	—	—	—	—	—	—

Note 1. Suffixes such as “-A” or “-B” appended to pin names indicate pin group identifiers. Electrical characteristics were evaluated per pin group. AC electrical characteristics are only guaranteed when pins within the same group are used together.

Note 2. When the CRSEP bit in the SCR1 register is set to 1, this pin cannot be used for the CTSn# signal. It can be used for the RTSn# signal.

22.3.14 Non-secure World Write Protection Register (PWPR_NS)

Address(es): PFS_NS.PWPR_NS 07C0 0D0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PmnPFS Register Write Enable	0: Writing to the PmnPFS registers is disabled. 1: Writing to the PmnPFS registers is enabled.	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled. 1: Writing to the PFSWE bit is disabled.	R/W

m = 0 to 9 or A to G; n = 00 to 15

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: Access to this register is not controlled by any security attribute registers.

This register is used to protect the PmnPFS registers corresponding to the pins placed in the non-secure world by the PmSAR registers.

PFSWE Bit (PmnPFS Register Write Enable)

Writing to the PmnPFS registers is only enabled when this bit is set to 1. Set the B0WI bit to 0 before setting this bit to 1.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is only enabled when this bit is set to 0.

22.3.15 Secure World Write Protection Register (PWPR_S)

Address(es): PFS.PWPR_S 87C0 0D14h

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PmnPFS Register Write Enable	0: Writing to the PmnPFS registers is disabled. 1: Writing to the PmnPFS registers is enabled.	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled. 1: Writing to the PFSWE bit is disabled.	R/W

m = 0 to 9 or A to G; n = 00 to 15

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

This register is used to protect the PmnPFS registers corresponding to the pins placed in the secure world by the PmSAR registers

PFSWE Bit (PmnPFS Register Write Enable)

Writing to the PmnPFS registers is only enabled when this bit is set to 1. Set the B0WI bit to 0 before setting this bit to 1.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is only enabled when this bit is set to 0.

22.3.16 Port m Security Attribute Setting Register (PmSAR) (m = 0 to 9, A to G)

Address(es): PFS.P0SAR 87C0 0D30h, PFS.P1SAR 87C0 0D34h, PFS.P2SAR 87C0 0D38h, PFS.P3SAR 87C0 0D3Ch, PFS.P4SAR 87C0 0D40h, PFS.P5SAR 87C0 0D44h, PFS.P6SAR 87C0 0D48h, PFS.P7SAR 87C0 0D4Ch, PFS.P8SAR 87C0 0D50h, PFS.P9SAR 87C0 0D54h, PFS.PASAR 87C0 0D58h, PFS.PBSAR 87C0 0D5Ch, PFS.PCSAR 87C0 0D60h, PFS.PDSAR 87C0 0D64h, PFS.PESAR 87C0 0D68h, PFS.PFSAR 87C0 0D6Ch, PFS.PGSAR 87C0 0D70h,
PFS_NS.P0SAR 07C0 0D30h, PFS_NS.P1SAR 07C0 0D34h, PFS_NS.P2SAR 07C0 0D38h, PFS_NS.P3SAR 07C0 0D3Ch, PFS_NS.P4SAR 07C0 0D40h, PFS_NS.P5SAR 07C0 0D44h, PFS_NS.P6SAR 07C0 0D48h, PFS_NS.P7SAR 07C0 0D4Ch, PFS_NS.P8SAR 07C0 0D50h, PFS_NS.P9SAR 07C0 0D54h, PFS_NS.PASAR 07C0 0D58h, PFS_NS.PBSAR 07C0 0D5Ch, PFS_NS.PCSAR 07C0 0D60h, PFS_NS.PDSAR 07C0 0D64h, PFS_NS.PESAR 07C0 0D68h, PFS_NS.PFSAR 07C0 0D6Ch, PFS_NS.PGSAR 07C0 0D70h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Pm15SA	Pm14SA	Pm13SA	Pm12SA	Pm11SA	Pm10SA	Pm09SA	Pm08SA	Pm07SA	Pm06SA	Pm05SA	Pm04SA	Pm03SA	Pm02SA	Pm01SA	Pm00SA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	Pm00SA	Pm00 Security Attribute Setting	0: Secure 1: Non-secure	R/W
b1	Pm01SA	Pm01 Security Attribute Setting		R/W
b2	Pm02SA	Pm02 Security Attribute Setting		R/W
b3	Pm03SA	Pm03 Security Attribute Setting		R/W
b4	Pm04SA	Pm04 Security Attribute Setting		R/W
b5	Pm05SA	Pm05 Security Attribute Setting		R/W
b6	Pm06SA	Pm06 Security Attribute Setting		R/W
b7	Pm07SA	Pm07 Security Attribute Setting		R/W
b8	Pm08SA	Pm08 Security Attribute Setting		R/W
b9	Pm09SA	Pm09 Security Attribute Setting		R/W
b10	Pm10SA	Pm10 Security Attribute Setting		R/W
b11	Pm11SA	Pm11 Security Attribute Setting		R/W
b12	Pm12SA	Pm12 Security Attribute Setting		R/W
b13	Pm13SA	Pm13 Security Attribute Setting		R/W
b14	Pm14SA	Pm14 Security Attribute Setting		R/W
b15	Pm15SA	Pm15 Security Attribute Setting		R/W

m = 0 to 9 or A to G

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

This register is readable and writable in 8-bit or 16-bit units and is used to specify the security attributes for the individual pins.

22.3.17 Low-Voltage Operating Characteristics Control Register (LVOCR)

Address(es): SYSC.LVOCR 8701 9AB0h, SYSC_NS.LVOCR 0701 9AB0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LVO0E
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	LVO0E	Low-Voltage Operating Characteristics Control 0	0: VCC ≥ 2.7 V 1: VCC < 2.7 V	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register is used to control the operating characteristics of the input/output buffers.

LVO0E Bit (Low-Voltage Operating Characteristics Control 0)

Set this bit to 0 when the VCC voltage to be used is 2.7 V or higher. Leave this bit as 1 when the VCC voltage to be used is below 2.7 V.

22.4 Operation

22.4.1 General-Purpose I/O Port

All pins except P300 to P303, P315, PF00, or PF01 operate as general input port pins after a reset. Each general I/O port consists of 16 pins. The ports can be accessed in port units through the port control registers (PCNTRn, n = 1 to 4) or in pin units through the Pmn pin function selection registers (PmnPFS, m = 0 to 9 or A to G, n = 00 to 15). For details of the registers, refer to section 22.3, Register Descriptions.

22.4.2 Selecting Pin Functions

The following functions can be controlled for the individual pins through register settings.

- The security attribute is selectable per-pin as desired.
- Selection of CMOS or N-channel open-drain output, pull-up resistors, and driving ability can be controlled.
- The I/O directions of port pins can be selected, the data to be output can be specified, and the input data can be read.
- The I/O signals of peripheral functions are assignable to pins.

22.4.3 Port Group Function of the ELC

Ports B to E are assigned to the ELC port group function in this MCU.

22.4.3.1 Operation when the ELC_PORTn (n = B to E) Signal is Input from the ELC

This MCU has the following two functions to be activated in response to the input of an ELC_PORTn (n = B to E) signal from the ELC.

(1) Event input

When the ELC_PORTn (n = B to E) signal is generated by the ELC, the levels on the pins in the corresponding port group are transferred to the EIDR bits if the pins are general input pins (PDR = 0 and PMR = 0 in the PmnPFS register). See Figure 22.2. If the pins are general output pins (PDR = 1) or the I/O pins for peripheral functions (PMR = 1), the corresponding EIDR bits are set to 0 regardless of the levels on the pins.

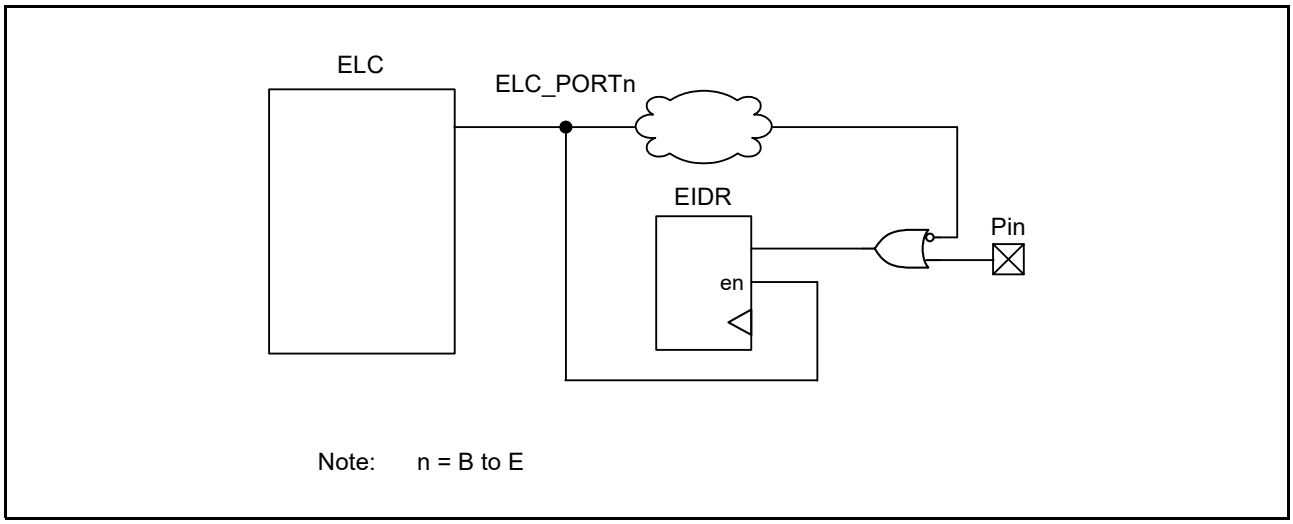


Figure 22.2 Data Input from an Event Port Group

(2) Setting the PODR Register through the EOSR and EORR Registers

When the ELC_PORTn (n = B to E) signal is generated, the PODR register is set according to the settings of the EOSR and EORR registers.

- If a bit in the EOSR register is set to 1, the corresponding bit in the PODR register is set to 1 in response to generation of the ELC_PORTn signal.
  If a bit in the EOSR register is set to 0, the corresponding bit in the PODR register retains its value.
- If a bit in the EORR register is set to 1, the corresponding bit in the PODR register is set to 0 in response to generation of the ELC_PORTn signal.
  If a bit in the EORR register is set to 0, the corresponding bit in the PODR register retains its value.

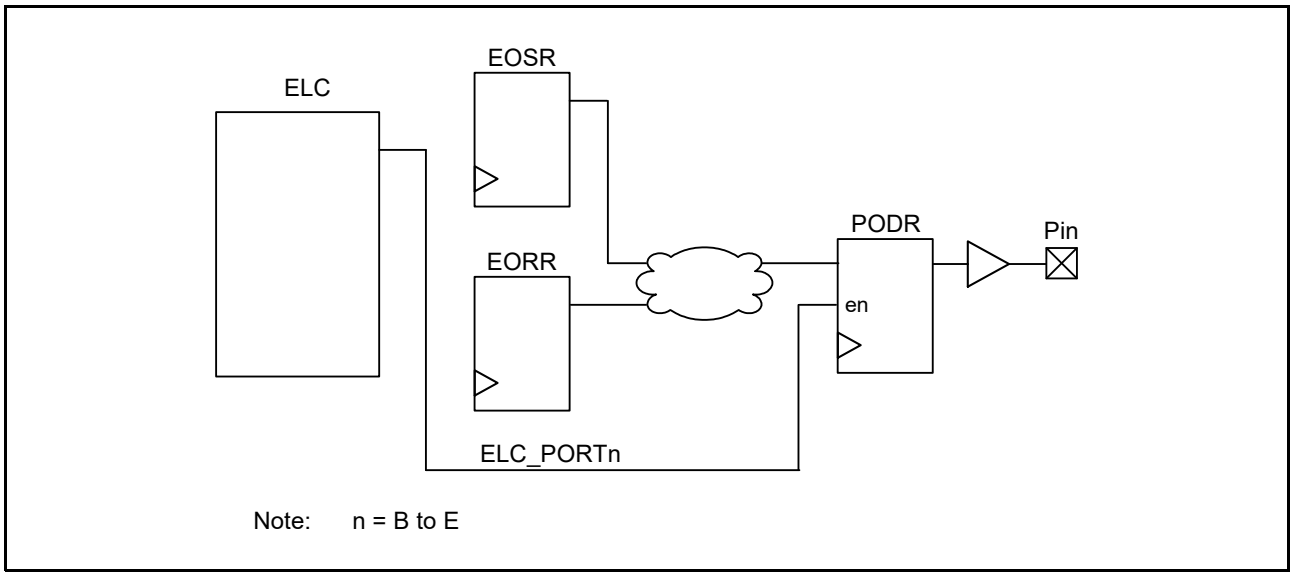


Figure 22.3 Data Output from an Event Port Group

22.4.3.2 Operation when an Event Pulse is Applied to the ELC

Use the EOFR[1:0] bits in the PmnPFS register to apply an event pulse from external pins to the ELC. For details, refer to section 22.3.13, Pmn Pin Function Selection Register (PmnPFS) (m = 0 to 9, A to G; n = 00 to 15). For example, when data are input to port B (PB00 to PB07), the logical OR of the data values from these eight pins are taken and the result is applied to the ELC as an event pulse. Ports C to E operate in the same way. See Figure 22.4.

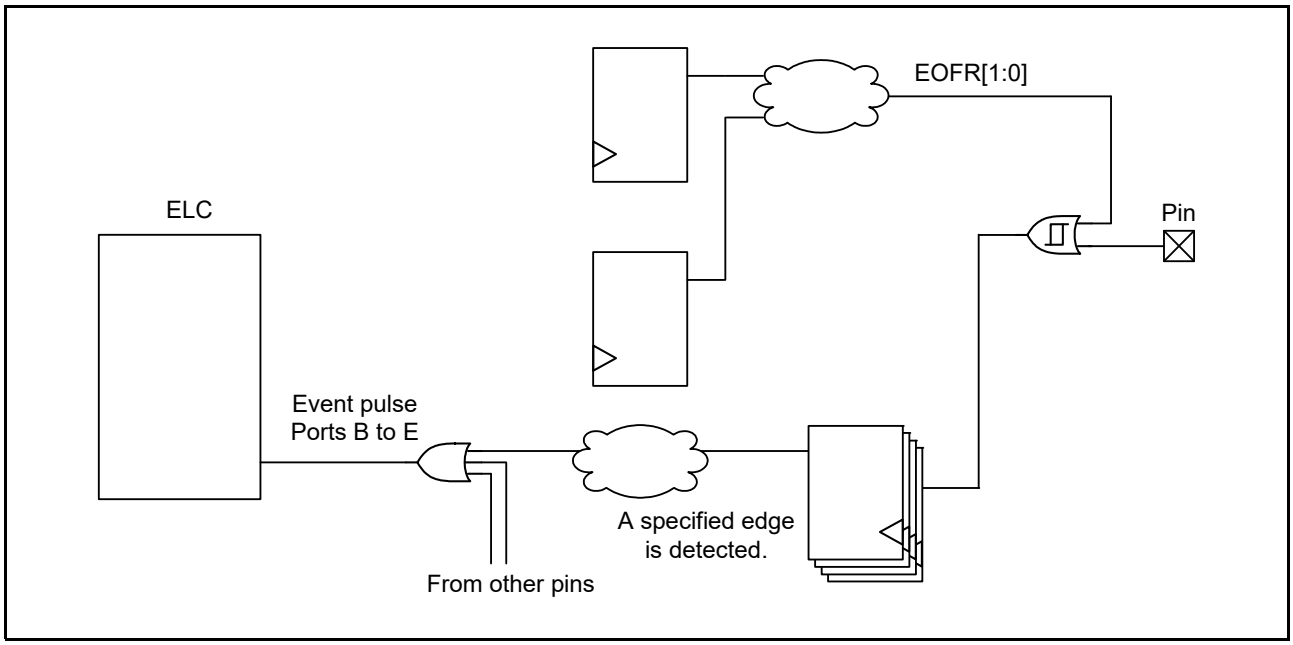


Figure 22.4 Generating an Event Pulse

22.5 Handling of Unused Pins

Details on the handling of unused pins are given in Table 22.29.

Table 22.29 Handling of Unused Pins

Pin Name	Handling
P315/MD	This pin must always be used as the mode pin.
RES#	Pull this pin up to VCC via a resistor.
PF00/USB0_DP, PF01/USB0_DM	Leave these pins open-circuit when the PMR bit is set to 1 in both the PF00PFS and PF01PFS registers. Set these pins as follows when the PMR bit is set to 0 in either the PF00PFS or PF01PFS register. • When the I/O direction is set as input (PDRn bit in the PCNTR1 register = 0), connect the corresponding pin to VCC_USB or VSS via a resistor.*1 • When the I/O direction is set as output (PDRn bit in the PCNTR1 register = 1), leave the corresponding pin open-circuit.*1
P305/NMI	Pull this pin up to VCC via a resistor.
P306/EXTAL	Set the MOSTP bit of the MOSCCR register in the SYSC to 1 (selecting general port pin P306) when the main clock oscillator is not to be used. If this pin is not used either for clock oscillation or as port pin P306, handle it in the way stated for “Other ports” below.
P307/XTAL	Set the MOSTP bit of the MOSCCR register in the SYSC to 1 (selecting general port pin P307) when the main clock oscillator is not to be used. The XTAL pin works as P307 when an external clock is input to the EXTAL pin. If this pin is not used either for clock oscillation or as port pin P307, handle it in the way stated for “Other ports” below.
P309/XCIN/EXCIN	Set the SOSTP bit of the SOSCCR register in the SYSC to 1 (selecting general port pin P309) when the sub-clock oscillator is not to be used. Set the SOSEL bit of the SOMCR register in the SYSC to 0 (selecting general port pin P309) when an external clock is not to be used. If this pin is not used either for clock oscillation or as port pin P309, pull it down to VSS via a resistor.
P308/XCOUT	Set the SOSTP bit of the SOSCCR register in the SYSC to 1 (selecting general port pin P308) when the sub-clock oscillator is not to be used. If this pin is not used either for clock oscillation or as port pin P308, pull it down to VSS via a resistor.
P300, P301, P303	Do not pull these pins down because the input pull-up resistors are enabled (PCR bit in the PmnPFS register = 1) after release from the reset state.
P302	Be careful since this pin operates as an output pin after release from the reset state.
P400 to P415	• When the I/O direction is set as input (PDRn bit in the PCNTR1 register = 0), connect the corresponding pin to AVCC0 or AVSS0 via a resistor.*1 • When the I/O direction is set as output (PDRn bit in the PCNTR1 register = 1), leave the corresponding pin open-circuit.*1
P600 to P607. P900 to P907, PD00 to PD07	• When the I/O direction is set as input (PDRn bit in the PCNTR1 register = 0), connect the corresponding pin to VCC2 or VSS via a resistor.*1 • When the I/O direction is set as output (PDRn bit in the PCNTR1 register = 1), leave the corresponding pin open-circuit.*1
Other ports	• When the I/O direction is set as input (PDRn bit in the PCNTR1 register = 0), connect the corresponding pin to VCC or VSS via a resistor.*1 • When the I/O direction is set as output (PDRn bit in the PCNTR1 register = 1), leave the corresponding pin open-circuit.*1
VREFH0	Connect this pin to AVCC0.
VREFL0	Connect this pin to AVSS0.
USBHS_DP, USBHS_DM, USBHS_RREF	Prerequisites • AVCC_USBHS = VCC_USBHS: Both are connected to VCC. • AVSS_USBHS = VSS_USBHS: Both are connected to VSS. - The USBHS is placed in the module-stop state (MSTPB12 bit in the MSTPCRB register = 1) Handling • Leave the USBHS_DP, USBHS_DM, and USBHS_RREF pins open-circuit.
VLO	Leave this pin open-circuit.*2
VBATT	Connect this pin to VCC.

Note 1. Set the PMR, ISEL, PCR, and ASEL bits in the relevant PmnPFS register to 0.

Note 2. For details, refer to section 66, Internal Voltage Regulator.

22.6 Notes

22.6.1 Procedure for Setting Pin Functions

Use the following procedure to specify the function of an I/O pin.

1. Set the B0WI bit in the PWPR_NS register to 0 to enable writing to the PFSWE bit in the PWPR_NS register.*1
2. Set the PFSWE bit in the PWPR_NS register to 1 to enable writing to the PmnPFS register.*1
3. Set the PMR bit in the PmnPFS register corresponding to the target pin to 0 to set the pin as a general I/O port pin.
4. Set the PSEL[6:0] bits in the PmnPFS register to select the peripheral module function pins.
5. Set the PMR bit in the PmnPFS register to 1 to switch the pin to the selected peripheral function as required.
6. Set the PFSWE bit in the PWPR_NS register to 0 to disable writing to the PmnPFS register.*1
7. Set the B0WI bit in the PWPR_NS register to 1 to disable writing to the PFSWE bit in the PWPR_NS register.*1

Note 1. When the security attribute of the Pmn pin is set to 0 in the PmSAR register, use the PWPR_S register to enable or disable writing to the PmnPFS register.

22.6.2 Procedure for Using the Event Output Function to ELC

Use the following procedure to use changes in the input to a port group (B to E) as an event.

1. Set the ELS[9:0] bits of the ELSRx register in the ELC to 000h to prevent malfunctions due to an unexpected event. Whether an unexpected event may be generated depends on the states of external pins.
2. Use the EOFR[1:0] bits in the PmnPFS register to select the input edges to be detected.
3. Wait for 100 ns.
4. Use the ELS[9:0] bits of the ELSRx register in the ELC to select the target for linking of the port event.

22.6.3 Method of Setting the Port Output Data Register (PODR)

The following methods can be used to specify a value in the PODR register.

- Set the EORR bit in the PCNTR4 register to 1 to set the corresponding bit in the PODR register to 0 when the ELC_PORTn (n = B to E) signal is generated.
- Set the EOSR bit in the PCNTR4 register to 1 to set the corresponding bit in the PODR register to 1 when the ELC_PORTn (n = B to E) signal is generated.
- Set the PORR bit in the PCNTR3 register to 1 to set the corresponding bit in the PODR register to 0.
- Set the POSR bit in the PCNTR3 register 1 to set the corresponding bit in the PODR register to 1.
- Set the PODRn bit in the PCNTR1 register to set the corresponding bit in the PODR register to 0 or 1.
- Set the PODRn bit in the PmnPFS register to set the corresponding bit in the PODR register to 0 or 1.

In case of simultaneous direct writing to and event-driven modification of a register, event-driven modification takes priority.

22.6.4 Note on Using Analog Functions

To use a pin for an analog signal, set both the pin mode select (PMR) bit in the PmnPFS register and the port I/O direction select (PDRn) bit in the PCNTR1 register to 0 to make the pin operate as a general input port pin. After that, set the analog function select (ASEL) bit in the PmnPFS register to 1.

22.6.5 Notes on Using the RTC Time-Capture Event Inputs and Tampering Detection

To use the P310, P311, and P312 pins for RTC time-capture event input and tampering detection, set the TAMPICR1.CHnEN (n = 0 to 2) bits of the SYSC to 1. When the setting of a TAMPICR1.CHnEN bit is 1, the settings of registers related to the P31n pins in this section are ignored. Note, however, that the settings of the P31nPFS.ISEL bits are valid.

The TAMPICR1 register is not initialized by a reset. Therefore, when RTC input and tampering detection are not to be used, the corresponding bits in the TAMPICR1 register require setting to 0 after a reset. For details of the TAMPICR1 register, refer to section 12.2.11, Tamper/RTCIC Input Control Register 1 (TAMPICR1).

22.6.6 Selecting the USB0_DP and USB0_DM Pins

The USB0_DP and USB0_DM pins are respectively multiplexed with the PF00 and PF01 pins. Use the PMR and PSEL[6:0] bits in the PF00PFS register to select the USB0_DP or PF00 pin function and use the PMR and PSEL[6:0] bits in the PF01PFS register to select the USB0_DM or PF01 pin function. Table 22.30 lists the PFS register settings and the corresponding functions selected for the PF00 and PF01 pins.

Table 22.30 Selection of the USB or Port Pins

PFS Register Settings				Pin Function	
PF00PFS		PF01PFS			
PMR	PSEL[6:0]	PMR	PSEL[6:0]	PF00 Pin	PF01 Pin
0	Don't care	0	Don't care	General I/O	General I/O
0	Don't care	1	Other than 0010011b	General I/O	Peripheral function other than USB
0	Don't care	1	0010011b	General I/O	This pin cannot be used.
1	Other than 0010011b	0	Don't care	Peripheral function other than USB	General I/O
1	Other than 0010011b	1	Other than 0010011b	Peripheral function other than USB	Peripheral function other than USB
1	Other than 0010011b	1	0010011b	Peripheral function other than USB	This pin cannot be used.
1	0010011b	0	Don't care	This pin cannot be used.	General I/O
1	0010011b	1	Other than 0010011b	This pin cannot be used.	Peripheral function other than USB
1	0010011b	1	0010011b	USB0_DP	USB0_DM

Note: When the PF00 and PF01 pins are to be used for general I/O signals (the PF00 and PF01 signals) or for a peripheral function other than the USB, use the registers related to the USB with their initial values.
Note: When the PF00 and PF01 pins are to be used for USB signals (the USB0_DP and USB0_DM signals), use the registers related to the PF00 and PF01 general I/O signals with their initial values.
Note: Only set the PF00PFS and PF01PFS registers for selecting the pin functions once after a reset.

22.6.7 Pulling the PF00 and PF01 Pins Used for USB or General I/O Signals Up or Down

When the PF00 and PF01 pins are used for general I/O signals, the pull-up and pull-down functions of the USB affect the PF00 and PF01 pins. Therefore, the pull-up or pull-down functions need to be disabled before the pins are used for general I/O signals. The pull-up and pull-down functions of the USB are controlled by the DPRPU and DRPD bits of the SYSCFG register in USB0.

22.6.8 Note on Using the NMI Pin

To use the P305 pin as an NMI pin, set the NMIER.NMIEN bit of the ICU to 1. Register settings for control of the pin as a port pin are not required. For details on the NMIER register, see section 15.2.11, Non-Maskable Interrupt Enable Register (NMIER).

22.6.9 Note on Using the Main Clock Oscillator

To use a resonator, set the MOMCR.MOSEL bit of the SYSC to 0, and set the P307 and P306 pins for use as general-purpose input port pins. To use the external clock input, set the MOMCR.MOSEL bit to 1, and set the P307 pin for use as a general-purpose input port pin. For details on the MOMCR register, see section 9.2.28, Main Clock Oscillator Mode Oscillation Control Register (MOMCR).

22.6.10 Note on Using the Sub-Clock Oscillator

To use a resonator, set the SOMCR.SOSEL bit of the SYSC to 0, and set the P309 and P308 pins for use as general-purpose input port pins. To use the external clock input, set the SOMCR.SOSEL bit to 1, and set the P309 pin for use as a general-purpose input port pin. For details on the SOMCR register, see section 9.2.29, Sub-Clock Oscillator Mode Control Register (SOMCR).

22.6.11 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 22.31 to Table 22.35.

- The blank columns in Table 22.31 to Table 22.35 indicate the bits corresponding to the pins listed in Table 22.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system. However, the corresponding PORT3.PDR.PDR05, PDR08, and PDR09 bits on the input-only P305, P308, and P309 terminals are reserved bits. These bits should be set to 0 (input).
- The columns other than the blank columns in Table 22.31 to Table 22.35 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 22.31 to Table 22.35.

Table 22.31 PDR Register Settings in 320-Pin Packages

Port Symbol	PDR Register															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT0																
PORT1																
PORT2																
PORT3		0	0				0	0			0					
PORT4																
PORT5																
PORT6	0	0	0	0	0	0	0	0								
PORT7																
PORT8	0	0	0	0	0	0	0	0								
PORT9	0	0	0	0	0	0	0	0								
PORTA	0	0	0	0	0	0	0	0								
PORTB	0	0	0	0	0	0	0	0								
PORTC	0	0	0	0	0	0	0	0								
PORTD	0	0	0	0	0	0	0	0								
PORTE	0	0	0	0	0	0	0	0								
PORTF	0	0	0	0	0	0										
PORTG																

Table 22.32 PDR Register Settings in 224-Pin Packages

Port Symbol	PDR Register															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1								
PORT1																
PORT2																
PORT3		0	0				0	0			0					
PORT4			1	1	1	1	1	1								
PORT5	1	1	1	1	1	1	1	1								
PORT6	0	0	0	0	0	0	0	0								
PORT7	1	1	1	1	1	1	1	1								
PORT8	0	0	0	0	0	0	0	0								
PORT9	0	0	0	0	0	0	0	0								
PORTA	0	0	0	0	0	0	0	0								
PORTB	0	0	0	0	0	0	0	0								
PORTC	0	0	0	0	0	0	0	0								
PORTD	0	0	0	0	0	0	0	0								
PORTE	0	0	0	0	0	0	0	0								
PORTF	0	0	0	0	0	0										
PORTG	1	1	1	1	1	1	1	1								

Table 22.33 PDR Register Settings in 176-Pin Packages

Port Symbol	PDR Register															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1								
PORT1	1	1	1	1	1	1	1	1								
PORT2	1	1	1	1	1	1	1	1								
PORT3		0	0				0	0			0					
PORT4			1	1	1	1	1	1								
PORT5	1	1	1	1	1	1	1	1	1	1	1	1				
PORT6	0	0	0	0	0	0	0	0								
PORT7	1	1	1	1	1	1	1	1								
PORT8	0	0	0	0	0	0	0	0	1	1						
PORT9	0	0	0	0	0	0	0	0	1	1	1	1				
PORTA	0	0	0	0	0	0	0	0								
PORTB	0	0	0	0	0	0	0	0								
PORTC	0	0	0	0	0	0	0	0								
PORTD	0	0	0	0	0	0	0	0								
PORTE	0	0	0	0	0	0	0	0								
PORTF	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
PORTG	1	1	1	1	1	1	1	1	1	1	1	1	1			

Table 22.34 PDR Register Settings in 144-Pin Packages

Port Symbol	PDR Register															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1								
PORT1	1	1	1	1	1	1	1	1								
PORT2	1	1	1	1	1	1	1	1								
PORT3		0	0				0	0			0					
PORT4			1	1	1	1	1	1	1	1						
PORT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PORT6	0	0	0	0	0	0	0	0	1	1						
PORT7	1	1	1	1	1	1	1	1	1	1	1	1	1			
PORT8	0	0	0	0	0	0	0	0	1	1	1	1				
PORT9	0	0	0	0	0	0	0	0	1	1	1	1	1	1		
PORTA	0	0	0	0	0	0	0	0								
PORTB	0	0	0	0	0	0	0	0								
PORTC	0	0	0	0	0	0	0	0								
PORTD	0	0	0	0	0	0	0	0								
PORTE	0	0	0	0	0	0	0	0								
PORTF	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
PORTG	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 22.35 PDR Register Settings in 100-Pin Packages

Port Symbol	PDR Register															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT0	1	1	1	1	1	1	1	1								1
PORT1	1	1	1	1	1	1	1	1								1
PORT2	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
PORT3		0	0				0	0			0					
PORT4	1		1	1	1	1	1	1	1	1	1		1		1	
PORT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PORT6	0	0	0	0	0	0	0	0	1	1	1	1	1	1		
PORT7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PORT8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
PORT9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
PORTA	0	0	0	0	0	0	0	0								
PORTB	0	0	0	0	0	0	0	0	1							
PORTC	0	0	0	0	0	0	0	0	1							
PORTD	0	0	0	0	0	0	0	0								
PORTE	0	0	0	0	0	0	0	0	1							
PORTF	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
PORTG	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

23. Software Configurable 32-Bit Programmable I/O Port (PIO)

23.1 Overview

The software configurable 32-bit programmable I/O port module (PIO) allows the selection of up to 32 out of 64 I/O port pins implemented as peripheral functions, and the result is usable just as if the pins were under the control of a general-purpose I/O port with up to 32 effective bits. This allows virtual bundling of the pins, so that simultaneous reflection of the port pin states in a register and simultaneous switching of the output signals on the selected port pins are both possible. Table 23.1 lists the specifications and Figure 23.1 shows a block diagram of the PIO.

Table 23.1 Specifications of the PIO

Items	Description
Operating clock	• PCLKB
Functions	• Of the 64 port pins denoted as PIO_x pins (x = 000 to 063), up to 32 can be selected for simultaneous reflection of the port pin states in a register and simultaneous switching of the output signals. • Input or output can be enabled, and for outputs, the high or low level can be specified through register settings. • The port pin states can be read through a register. • The output from any port pin can respectively be changed to the high or low level by using the port output setting or resetting register.
Protect Zone filter	• Security attribute can be set.

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

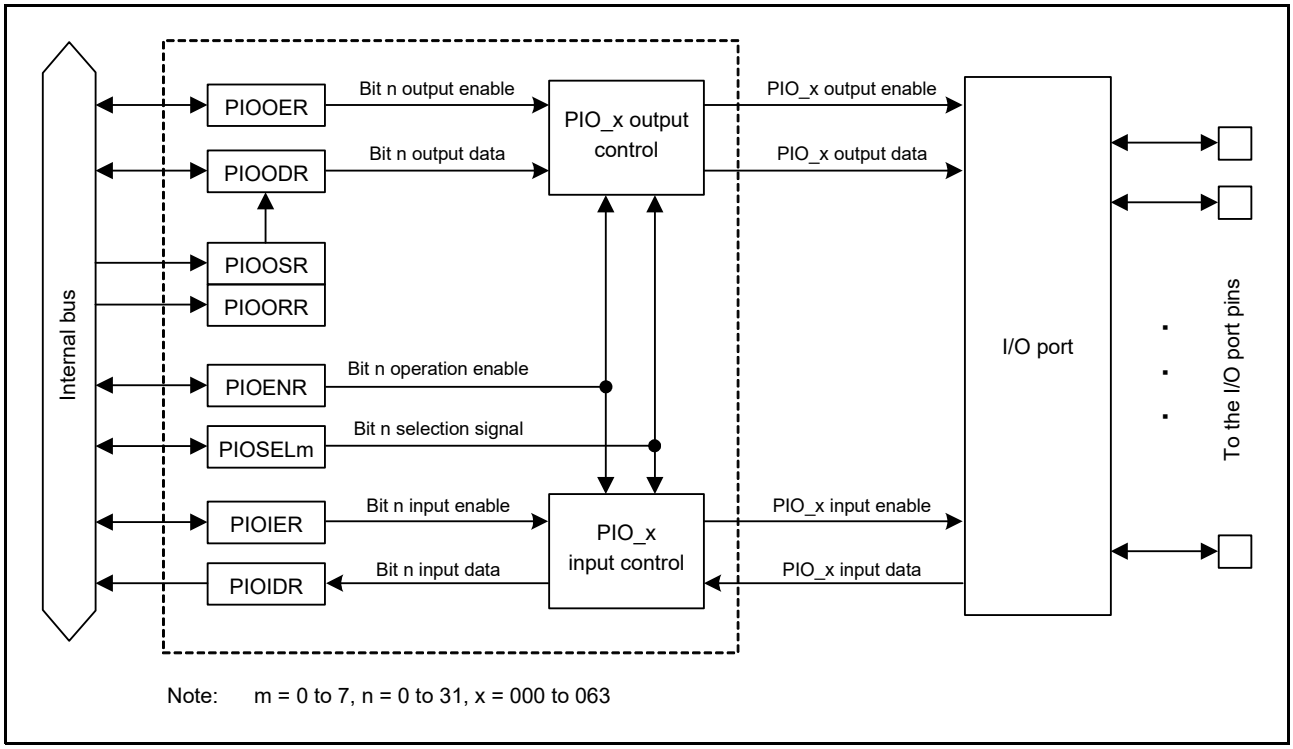


Figure 23.1 Block Diagram of the PIO

23.2 Register Descriptions

23.2.1 PIO Operation Enable Register (PIOENR)

Address(es): PIO.PIOENR 8780 6000h, PIO_NS.PIOENR 0780 6000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	PIO Operation Enable 0	0: Operation is disabled. 1: Operation is enabled.	R/W
b1	EN1	PIO Operation Enable 1		R/W
b2	EN2	PIO Operation Enable 2		R/W
b3	EN3	PIO Operation Enable 3		R/W
b4	EN4	PIO Operation Enable 4		R/W
b5	EN5	PIO Operation Enable 5		R/W
b6	EN6	PIO Operation Enable 6		R/W
b7	EN7	PIO Operation Enable 7		R/W
b8	EN8	PIO Operation Enable 8		R/W
b9	EN9	PIO Operation Enable 9		R/W
b10	EN10	PIO Operation Enable 10		R/W
b11	EN11	PIO Operation Enable 11		R/W
b12	EN12	PIO Operation Enable 12		R/W
b13	EN13	PIO Operation Enable 13		R/W
b14	EN14	PIO Operation Enable 14		R/W
b15	EN15	PIO Operation Enable 15		R/W
b16	EN16	PIO Operation Enable 16		R/W
b17	EN17	PIO Operation Enable 17		R/W
b18	EN18	PIO Operation Enable 18		R/W
b19	EN19	PIO Operation Enable 19		R/W
b20	EN20	PIO Operation Enable 20		R/W
b21	EN21	PIO Operation Enable 21		R/W
b22	EN22	PIO Operation Enable 22		R/W
b23	EN23	PIO Operation Enable 23		R/W
b24	EN24	PIO Operation Enable 24		R/W
b25	EN25	PIO Operation Enable 25		R/W
b26	EN26	PIO Operation Enable 26		R/W
b27	EN27	PIO Operation Enable 27		R/W
b28	EN28	PIO Operation Enable 28		R/W
b29	EN29	PIO Operation Enable 29		R/W
b30	EN30	PIO Operation Enable 30		R/W
b31	EN31	PIO Operation Enable 31		R/W

The PIOENR register is used to enable or disable operation of the PIO target port pins. For details, see section 23.3.1, Input and section 23.3.2, Output.

23.2.2 PIO Port Input Enable Register (PIOIER)

Address(es): PIO.PIOIER 8780 6010h, PIO_NS.PIOIER 0780 6010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IE31	IE30	IE29	IE28	IE27	IE26	IE25	IE24	IE23	IE22	IE21	IE20	IE19	IE18	IE17	IE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IE0	Port Input Enable 0	0: Input is disabled. 1: Input is enabled.	R/W
b1	IE1	Port Input Enable 1		R/W
b2	IE2	Port Input Enable 2		R/W
b3	IE3	Port Input Enable 3		R/W
b4	IE4	Port Input Enable 4		R/W
b5	IE5	Port Input Enable 5		R/W
b6	IE6	Port Input Enable 6		R/W
b7	IE7	Port Input Enable 7		R/W
b8	IE8	Port Input Enable 8		R/W
b9	IE9	Port Input Enable 9		R/W
b10	IE10	Port Input Enable 10		R/W
b11	IE11	Port Input Enable 11		R/W
b12	IE12	Port Input Enable 12		R/W
b13	IE13	Port Input Enable 13		R/W
b14	IE14	Port Input Enable 14		R/W
b15	IE15	Port Input Enable 15		R/W
b16	IE16	Port Input Enable 16		R/W
b17	IE17	Port Input Enable 17		R/W
b18	IE18	Port Input Enable 18		R/W
b19	IE19	Port Input Enable 19		R/W
b20	IE20	Port Input Enable 20		R/W
b21	IE21	Port Input Enable 21		R/W
b22	IE22	Port Input Enable 22		R/W
b23	IE23	Port Input Enable 23		R/W
b24	IE24	Port Input Enable 24		R/W
b25	IE25	Port Input Enable 25		R/W
b26	IE26	Port Input Enable 26		R/W
b27	IE27	Port Input Enable 27		R/W
b28	IE28	Port Input Enable 28		R/W
b29	IE29	Port Input Enable 29		R/W
b30	IE30	Port Input Enable 30		R/W
b31	IE31	Port Input Enable 31		R/W

The PIOIER register is used to enable or disable input to the PIO target port pins. For details, see section 23.3.1, Input.

23.2.3 PIO Port Output Enable Register (PIOOER)

Address(es): PIO.PIOOER 8780 6014h, PIO_NS.PIOOER 0780 6014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	OE31	OE30	OE29	OE28	OE27	OE26	OE25	OE24	OE23	OE22	OE21	OE20	OE19	OE18	OE17	OE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE0	Port Output Enable 0	0: Output is disabled. 1: Output is enabled.	R/W
b1	OE1	Port Output Enable 1		R/W
b2	OE2	Port Output Enable 2		R/W
b3	OE3	Port Output Enable 3		R/W
b4	OE4	Port Output Enable 4		R/W
b5	OE5	Port Output Enable 5		R/W
b6	OE6	Port Output Enable 6		R/W
b7	OE7	Port Output Enable 7		R/W
b8	OE8	Port Output Enable 8		R/W
b9	OE9	Port Output Enable 9		R/W
b10	OE10	Port Output Enable 10		R/W
b11	OE11	Port Output Enable 11		R/W
b12	OE12	Port Output Enable 12		R/W
b13	OE13	Port Output Enable 13		R/W
b14	OE14	Port Output Enable 14		R/W
b15	OE15	Port Output Enable 15		R/W
b16	OE16	Port Output Enable 16		R/W
b17	OE17	Port Output Enable 17		R/W
b18	OE18	Port Output Enable 18		R/W
b19	OE19	Port Output Enable 19		R/W
b20	OE20	Port Output Enable 20		R/W
b21	OE21	Port Output Enable 21		R/W
b22	OE22	Port Output Enable 22		R/W
b23	OE23	Port Output Enable 23		R/W
b24	OE24	Port Output Enable 24		R/W
b25	OE25	Port Output Enable 25		R/W
b26	OE26	Port Output Enable 26		R/W
b27	OE27	Port Output Enable 27		R/W
b28	OE28	Port Output Enable 28		R/W
b29	OE29	Port Output Enable 29		R/W
b30	OE30	Port Output Enable 30		R/W
b31	OE31	Port Output Enable 31		R/W

The PIOOER register is used to enable or disable output from the PIO target port pins. For details, see section 23.3.2, Output.

23.2.4 PIO Port Input Data Register (PIOIDR)

Address(es): PIO.PIOIDR 8780 6020h, PIO_NS.PIOIDR 0780 6020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDR31	IDR30	IDR29	IDR28	IDR27	IDR26	IDR25	IDR24	IDR23	IDR22	IDR21	IDR20	IDR19	IDR18	IDR17	IDR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IDR0	Port Input Data 0	0: Low level 1: High level	R
b1	IDR1	Port Input Data 1		R
b2	IDR2	Port Input Data 2		R
b3	IDR3	Port Input Data 3		R
b4	IDR4	Port Input Data 4		R
b5	IDR5	Port Input Data 5		R
b6	IDR6	Port Input Data 6		R
b7	IDR7	Port Input Data 7		R
b8	IDR8	Port Input Data 8		R
b9	IDR9	Port Input Data 9		R
b10	IDR10	Port Input Data 10		R
b11	IDR11	Port Input Data 11		R
b12	IDR12	Port Input Data 12		R
b13	IDR13	Port Input Data 13		R
b14	IDR14	Port Input Data 14		R
b15	IDR15	Port Input Data 15		R
b16	IDR16	Port Input Data 16		R
b17	IDR17	Port Input Data 17		R
b18	IDR18	Port Input Data 18		R
b19	IDR19	Port Input Data 19		R
b20	IDR20	Port Input Data 20		R
b21	IDR21	Port Input Data 21		R
b22	IDR22	Port Input Data 22		R
b23	IDR23	Port Input Data 23		R
b24	IDR24	Port Input Data 24		R
b25	IDR25	Port Input Data 25		R
b26	IDR26	Port Input Data 26		R
b27	IDR27	Port Input Data 27		R
b28	IDR28	Port Input Data 28		R
b29	IDR29	Port Input Data 29		R
b30	IDR30	Port Input Data 30		R
b31	IDR31	Port Input Data 31		R

The PIOIDR register is used to read the port pin states from the PIO target port pins. For details, see section 23.3.1, Input.

23.2.5 PIO Port Output Data Register (PIOODR)

Address(es): PIO.PIOODR 8780 6024h, PIO_NS.PIOODR 0780 6024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ODR31	ODR30	ODR29	ODR28	ODR27	ODR26	ODR25	ODR24	ODR23	ODR22	ODR21	ODR20	ODR19	ODR18	ODR17	ODR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ODR0	Port Output Data 0	0: Low-level output 1: High-level output	R/W
b1	ODR1	Port Output Data 1		R/W
b2	ODR2	Port Output Data 2		R/W
b3	ODR3	Port Output Data 3		R/W
b4	ODR4	Port Output Data 4		R/W
b5	ODR5	Port Output Data 5		R/W
b6	ODR6	Port Output Data 6		R/W
b7	ODR7	Port Output Data 7		R/W
b8	ODR8	Port Output Data 8		R/W
b9	ODR9	Port Output Data 9		R/W
b10	ODR10	Port Output Data 10		R/W
b11	ODR11	Port Output Data 11		R/W
b12	ODR12	Port Output Data 12		R/W
b13	ODR13	Port Output Data 13		R/W
b14	ODR14	Port Output Data 14		R/W
b15	ODR15	Port Output Data 15		R/W
b16	ODR16	Port Output Data 16		R/W
b17	ODR17	Port Output Data 17		R/W
b18	ODR18	Port Output Data 18		R/W
b19	ODR19	Port Output Data 19		R/W
b20	ODR20	Port Output Data 20		R/W
b21	ODR21	Port Output Data 21		R/W
b22	ODR22	Port Output Data 22		R/W
b23	ODR23	Port Output Data 23		R/W
b24	ODR24	Port Output Data 24		R/W
b25	ODR25	Port Output Data 25		R/W
b26	ODR26	Port Output Data 26		R/W
b27	ODR27	Port Output Data 27		R/W
b28	ODR28	Port Output Data 28		R/W
b29	ODR29	Port Output Data 29		R/W
b30	ODR30	Port Output Data 30		R/W
b31	ODR31	Port Output Data 31		R/W

The PIOODR register holds the data for output from the PIO target port pins. For details, see section 23.3.2, Output.

23.2.6 PIO Port Output Setting Register (PIOOSR)

Address(es): PIO.PIOOSR 8780 6030h, PIO_NS.PIOOSR 0780 6030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	OSR31	OSR30	OSR29	OSR28	OSR27	OSR26	OSR25	OSR24	OSR23	OSR22	OSR21	OSR20	OSR19	OSR18	OSR17	OSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSR15	OSR14	OSR13	OSR12	OSR11	OSR10	OSR9	OSR8	OSR7	OSR6	OSR5	OSR4	OSR3	OSR2	OSR1	OSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSR0	Port Output Setting 0	0: No effect 1: High-level output	W
b1	OSR1	Port Output Setting 1		W
b2	OSR2	Port Output Setting 2		W
b3	OSR3	Port Output Setting 3		W
b4	OSR4	Port Output Setting 4		W
b5	OSR5	Port Output Setting 5		W
b6	OSR6	Port Output Setting 6		W
b7	OSR7	Port Output Setting 7		W
b8	OSR8	Port Output Setting 8		W
b9	OSR9	Port Output Setting 9		W
b10	OSR10	Port Output Setting 10		W
b11	OSR11	Port Output Setting 11		W
b12	OSR12	Port Output Setting 12		W
b13	OSR13	Port Output Setting 13		W
b14	OSR14	Port Output Setting 14		W
b15	OSR15	Port Output Setting 15		W
b16	OSR16	Port Output Setting 16		W
b17	OSR17	Port Output Setting 17		W
b18	OSR18	Port Output Setting 18		W
b19	OSR19	Port Output Setting 19		W
b20	OSR20	Port Output Setting 20		W
b21	OSR21	Port Output Setting 21		W
b22	OSR22	Port Output Setting 22		W
b23	OSR23	Port Output Setting 23		W
b24	OSR24	Port Output Setting 24		W
b25	OSR25	Port Output Setting 25		W
b26	OSR26	Port Output Setting 26		W
b27	OSR27	Port Output Setting 27		W
b28	OSR28	Port Output Setting 28		W
b29	OSR29	Port Output Setting 29		W
b30	OSR30	Port Output Setting 30		W
b31	OSR31	Port Output Setting 31		W

The PIOOSR register is used to set the data for output from the PIO target port pins.

OSRn Bits (Port Output Setting n) (n = 0 to 31)

Setting an OSRn bit to 1 sets the corresponding ODRn (n = 0 to 31) bit in the PIOODR register to 1. Setting of 0 has no effect on the PIOODR register. Setting of this register is not usable with bits to which port pins have not been assigned by the PIOSELm (m = 0 to 7) registers. All bits are read as 0. For details, see section 23.3.2, Output.

23.2.7 PIO Port Output Resetting Register (PIOORR)

Address(es): PIO.PIOORR 8780 6034h, PIO_NS.PIOORR 0780 6034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ORR31	ORR30	ORR29	ORR28	ORR27	ORR26	ORR25	ORR24	ORR23	ORR22	ORR21	ORR20	ORR19	ORR18	ORR17	ORR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ORR15	ORR14	ORR13	ORR12	ORR11	ORR10	ORR9	ORR8	ORR7	ORR6	ORR5	ORR4	ORR3	ORR2	ORR1	ORR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ORR0	Port Output Resetting 0	0: No effect 1: Low-level output	W
b1	ORR1	Port Output Resetting 1		W
b2	ORR2	Port Output Resetting 2		W
b3	ORR3	Port Output Resetting 3		W
b4	ORR4	Port Output Resetting 4		W
b5	ORR5	Port Output Resetting 5		W
b6	ORR6	Port Output Resetting 6		W
b7	ORR7	Port Output Resetting 7		W
b8	ORR8	Port Output Resetting 8		W
b9	ORR9	Port Output Resetting 9		W
b10	ORR10	Port Output Resetting 10		W
b11	ORR11	Port Output Resetting 11		W
b12	ORR12	Port Output Resetting 12		W
b13	ORR13	Port Output Resetting 13		W
b14	ORR14	Port Output Resetting 14		W
b15	ORR15	Port Output Resetting 15		W
b16	ORR16	Port Output Resetting 16		W
b17	ORR17	Port Output Resetting 17		W
b18	ORR18	Port Output Resetting 18		W
b19	ORR19	Port Output Resetting 19		W
b20	ORR20	Port Output Resetting 20		W
b21	ORR21	Port Output Resetting 21		W
b22	ORR22	Port Output Resetting 22		W
b23	ORR23	Port Output Resetting 23		W
b24	ORR24	Port Output Resetting 24		W
b25	ORR25	Port Output Resetting 25		W
b26	ORR26	Port Output Resetting 26		W
b27	ORR27	Port Output Resetting 27		W
b28	ORR28	Port Output Resetting 28		W
b29	ORR29	Port Output Resetting 29		W
b30	ORR30	Port Output Resetting 30		W
b31	ORR31	Port Output Resetting 31		W

The PIOORR register is used to reset the data for output from the PIO target port pins.

ORRn Bits (Port Output Resetting n) (n = 0 to 31)

Setting an ORRn bit to 1 sets the corresponding ODRn (n = 0 to 31) bit in the PIOODR register to 0. Setting of 0 has no effect on the PIOODR register. Setting of this register is not usable with bits to which port pins have not been assigned by the PIOSELm (m = 0 to 7) registers. All bits are read as 0. For details, see section 23.3.2, Output.

23.2.8 PIO Port Selection Register 0 (PIOSEL0)

Address(es): PIO.PIOSEL0 8780 6040h, PIO_NS.PIOSEL0 0780 6040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR3SEL[7:0]								DR2SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR1SEL[7:0]								DR0SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR0SEL[7:0]	DR0 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR1SEL[7:0]	DR1 I/O Port Selection		R/W
b23 to b16	DR2SEL[7:0]	DR2 I/O Port Selection		R/W
b31 to b24	DR3SEL[7:0]	DR3 I/O Port Selection		R/W

The PIOSEL0 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 3 to 0 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.9 PIO Port Selection Register 1 (PIOSEL1)

Address(es): PIO.PIOSEL1 8780 6044h, PIO_NS.PIOSEL1 0780 6044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR7SEL[7:0]								DR6SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR5SEL[7:0]								DR4SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR4SEL[7:0]	DR4 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR5SEL[7:0]	DR5 I/O Port Selection		R/W
b23 to b16	DR6SEL[7:0]	DR6 I/O Port Selection		R/W
b31 to b24	DR7SEL[7:0]	DR7 I/O Port Selection		R/W

The PIOSEL1 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 7 to 4 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.10 PIO Port Selection Register 2 (PIOSEL2)

Address(es): PIO.PIOSEL2 8780 6048h, PIO_NS.PIOSEL2 0780 6048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR11SEL[7:0]								DR10SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR9SEL[7:0]								DR8SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR8SEL[7:0]	DR8 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR9SEL[7:0]	DR9 I/O Port Selection		R/W
b23 to b16	DR10SEL[7:0]	DR10 I/O Port Selection		R/W
b31 to b24	DR11SEL[7:0]	DR11 I/O Port Selection		R/W

The PIOSEL2 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 11 to 8 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.11 PIO Port Selection Register 3 (PIOSEL3)

Address(es): PIO.PIOSEL3 8780 604Ch, PIO_NS.PIOSEL3 0780 604Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR15SEL[7:0]								DR14SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR13SEL[7:0]								DR12SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR12SEL[7:0]	DR12 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR13SEL[7:0]	DR13 I/O Port Selection		R/W
b23 to b16	DR14SEL[7:0]	DR14 I/O Port Selection		R/W
b31 to b24	DR15SEL[7:0]	DR15 I/O Port Selection		R/W

The PIOSEL3 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 15 to 12 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.12 PIO Port Selection Register 4 (PIOSEL4)

Address(es): PIO.PIOSEL4 8780 6050h, PIO_NS.PIOSEL4 0780 6050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR19SEL[7:0]								DR18SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR17SEL[7:0]								DR16SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR16SEL[7:0]	DR16 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR17SEL[7:0]	DR17 I/O Port Selection		R/W
b23 to b16	DR18SEL[7:0]	DR18 I/O Port Selection		R/W
b31 to b24	DR19SEL[7:0]	DR19 I/O Port Selection		R/W

The PIOSEL4 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 19 to 16 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.13 PIO Port Selection Register 5 (PIOSEL5)

Address(es): PIO.PIOSEL5 8780 6054h, PIO_NS.PIOSEL5 0780 6054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR23SEL[7:0]								DR22SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR21SEL[7:0]								DR20SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR20SEL[7:0]	DR20 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR21SEL[7:0]	DR21 I/O Port Selection		R/W
b23 to b16	DR22SEL[7:0]	DR22 I/O Port Selection		R/W
b31 to b24	DR23SEL[7:0]	DR23 I/O Port Selection		R/W

The PIOSEL5 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 23 to 20 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.14 PIO Port Selection Register 6 (PIOSEL6)

Address(es): PIO.PIOSEL6 8780 6058h, PIO_NS.PIOSEL6 0780 6058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR27SEL[7:0]								DR26SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR25SEL[7:0]								DR24SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR24SEL[7:0]	DR24 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR25SEL[7:0]	DR25 I/O Port Selection		R/W
b23 to b16	DR26SEL[7:0]	DR26 I/O Port Selection		R/W
b31 to b24	DR27SEL[7:0]	DR27 I/O Port Selection		R/W

The PIOSEL6 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 27 to 24 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.2.15 PIO Port Selection Register 7 (PIOSEL7)

Address(es): PIO.PIOSEL7 8780 605Ch, PIO_NS.PIOSEL7 0780 605Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DR31SEL[7:0]								DR30SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DR29SEL[7:0]								DR28SEL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DR28SEL[7:0]	DR28 I/O Port Selection	Specify the PIO_x pins for allocation for control by each of the bits n of the 32-bit port. The pins are selected from among those with x in the range from 0 to 63. Note that setting a value of or greater than 64 leads to no allocation of port pins.	R/W
b15 to b8	DR29SEL[7:0]	DR29 I/O Port Selection		R/W
b23 to b16	DR30SEL[7:0]	DR30 I/O Port Selection		R/W
b31 to b24	DR31SEL[7:0]	DR31 I/O Port Selection		R/W

The PIOSEL7 register is used to specify the PIO_x pins (x = 000 to 063) to be controlled by bits 31 to 28 of the 32-bit port. Setting the same value in multiple PIOSELm.DRnSEL[7:0] bits (m = 0 to 7, n = 0 to 31) is prohibited.

23.3 Operation

23.3.1 Input

The states of the PIO target port pins are reflected in the PIOIDR register by making the following settings.

- Set the PIOSELm.DRnSEL[7:0] bits to select the PIO target port pins from among the PIO_x pins (x = 000 to 063).
- Set the PIOENR register to enable the operation of the PIO target port pins.
- Set the PIOIER register to enable input to the PIO target port pins.

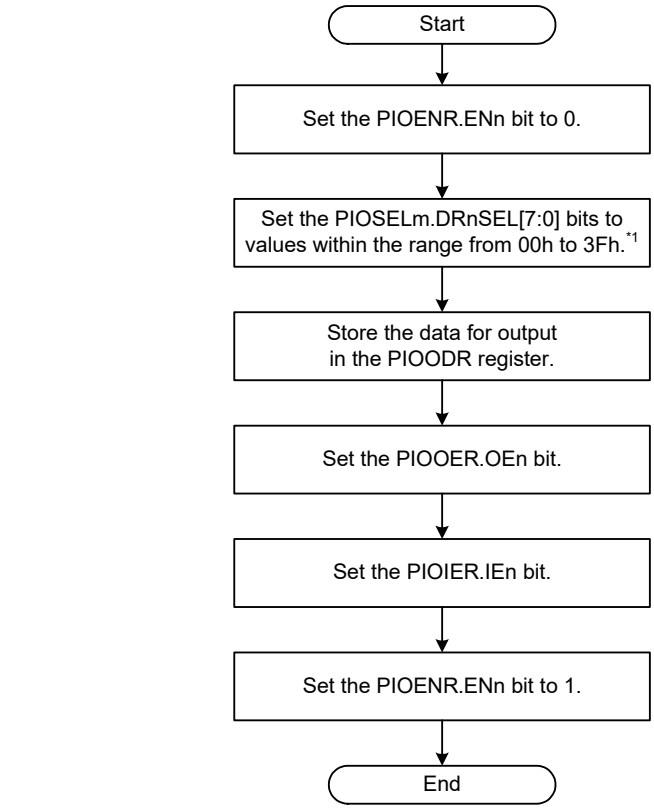
23.3.2 Output

The data stored in the PIOODR register are output from the PIO target port pins by making the following settings. The PIOODR register can also be set by using the PIOOSR and PIOORR registers.

- Set the PIOSELm.DRnSEL[7:0] bits to select the PIO target port pins from among the PIO_x pins (x = 000 to 063).
- Set the PIOENR register to enable the operation of the PIO target port pins.
- Set the PIOOER register to enable output from the PIO target port pins.

23.4 Initial Settings

Figure 23.2 shows the procedure for initial settings of PIO pins. The initial settings are required to change the assignment of the PIO target port pins or to change the direction (input/output) of the PIO target port pins.



Note 1. Setting the same value to multiple PIOSELm.DRnSEL[7:0] bits is prohibited.

Figure 23.2 Example Flowchart for PIO Initial Settings

24. General PWM Timer (GPTW)

This MCU has a general PWM timer (GPTW) consisting of sixteen 32-bit timer channels.

24.1 Overview

Table 24.1 lists the specifications for the GPTW, and Table 24.2 shows the functions of the GPTW. Figure 24.1, Figure 24.2 and Figure 24.3 show block diagrams of the GPTW.

Table 24.1 GPTW Specifications

Item	Description
Functions	• 32 bits × 16 channels • Up-counting or down-counting (sawtooth waves) or up/down-counting (triangle waves) for each counter. • Clock sources independently selectable for each channel • Two I/O pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can take place at crests or troughs, enabling the generation of laterally asymmetrical PWM waveforms. • Registers for setting up frame cycles in each channel (with capability for generating interrupts at overflow or underflow) • Generation of dead times in PWM operation • The direction of counting can be switched in triangle-wave PWM mode. • Complementary PWM control with use of the sawtooth-wave one-shot pulse mode or the triangle-wave PWM mode • Generation of high-accuracy duty cycles in the vicinity of 0% and 100% for complementary PWM output in the complementary PWM mode • In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times • Output with 0% or 100% duty cycle • Simultaneous start/stop/clearing/switch of desired channel counters • Operation of count start/count stop/counter clearing/count switch/up-counting/down-counting/input capture by maximum of eight ELC events based on the ELC setting • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by detecting two input signal conditions • Operation of count start/count stop/counter clearing/count switch/up-counting/down-counting/input capture by maximum of four external triggers • Function to control output negation by dead time error and requests for disabling of output from the POEG • A/D conversion start trigger generation function • Event signals for compare match A to F and for overflow/underflow can be output to the ELC • Input capture input can select noise filter function • Cycle count function • External pulse width measuring function • Logical operation between the channel output • Synchronous counter clearing/counter setting/input capture among channels • The timer-linked synchronization signal output function enables synchronous operation between channels synchronized with the timer period, and synchronous operation with other IPs synchronized with the timer period. • GTIOCnA/GTIOCnB pin input/output polarity inversion • Bus clock: PCLKA, GPTW count reference clock: GTCLK*1 Frequency ratio between PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64) when the synchronous clock is in use; PCLKA ≤ GPTCLK when the asynchronous clock is in use • Low power consumption function Module stop state can be set. • Protect Zone filter Security attribute can be set for each register*2

Note 1. Selection of the synchronous clock leads to the selection of PCLKD and selection of the asynchronous clock leads to the selection of GPTCLK as the GPTW count reference clock (GTCLK). Refer to Figure 24.5, Select GPTW Count Reference Clock.

Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 24.2 GPTW Functions (1/2)

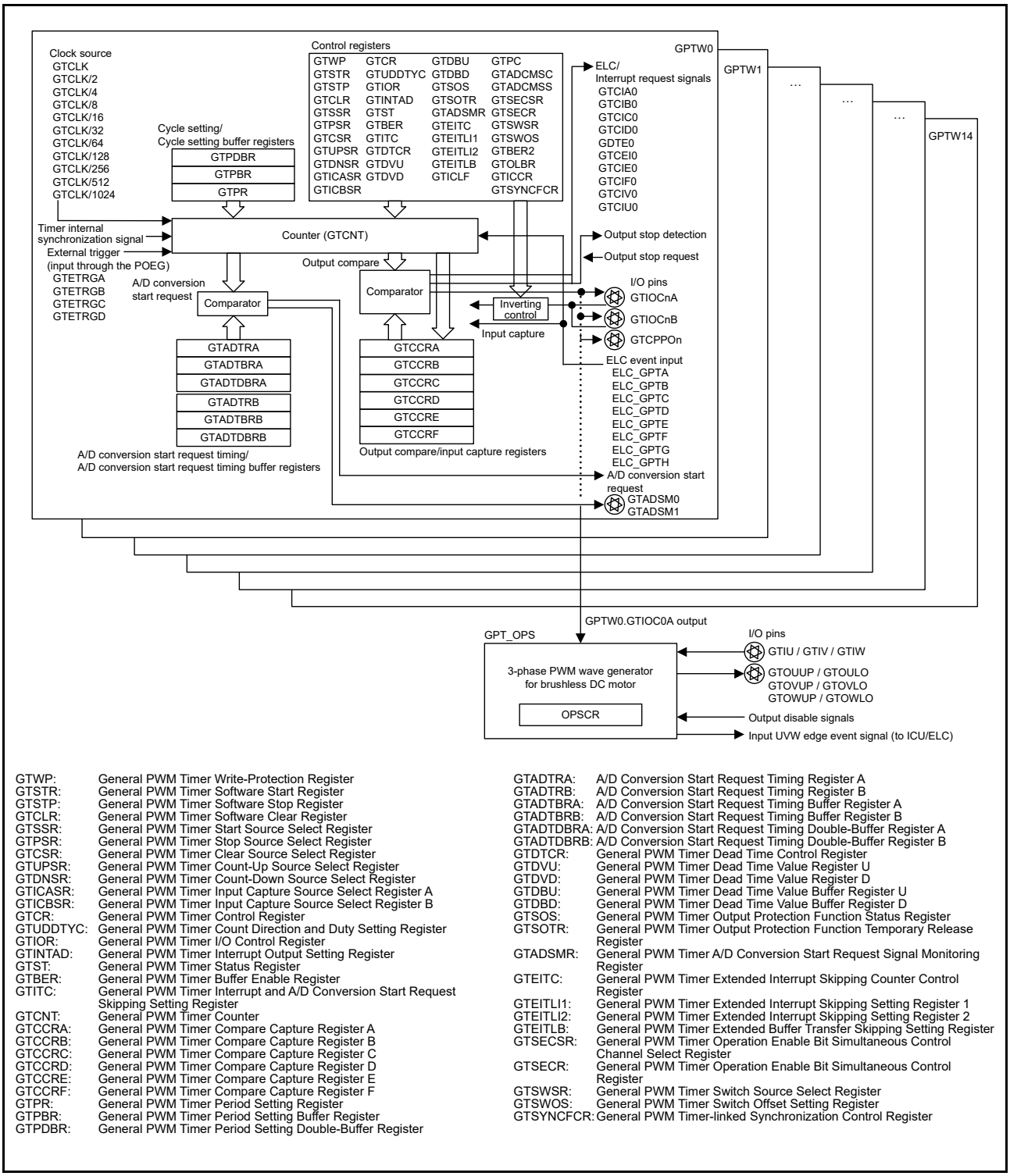
Item		GPTW0 to GPTW3	GPTW4 to GPTW9	GPTW10 to GPTW14	GPTW15
Timer counter		32-bit			
Count clock synchronous/asynchronous selection function		✓(Selected for all channels)			
Count clocks		GTCLK, GTCLK/2, GTCLK/4, GTCLK/8, GTCLK/16, GTCLK/32, GTCLK/64, GTCLK/128, GTCLK/256, GTCLK/512, GTCLK/1024, GTETRGA, GTETRGB, GTETRGC, GTETRGD			
Output compare/input capture registers (GTCCR)		GTCCRA, GTCCRB			
Compare/buffer registers		GTCCRC, GTCCRD, GTCCRE, GTCCRF			
PWM period setting register		GTPR			
PWM period setting buffer registers		GTPBR, GTPDBR			
Input capture input/ compare match output/ PWM output pin		GTIOCnA, GTIOCnB			
External trigger input pin (via the POEG)		GTETRGA, GTETRGB, GTETRGC, GTETRGD			
Counter clear sources		Compare match for the GTPR register, input capture, ELC input, the state of input pins, external trigger input, GTCCR register compare match, and Other channel's counter clear sources			
Cycle count function		✓			
Compare match output	Low output	✓			
	High output	✓			
	Toggle output	✓			
Input capture function		✓			
Automatic addition of dead time		✓			
PWM mode	Sawtooth-wave PWM mode 1	✓	✓	✓	✓
	Sawtooth-wave PWM mode 2	—	✓	—	—
	Sawtooth-wave one-shot pulse mode	✓	✓	✓	✓
	Triangle-wave PWM mode 1/2/3	✓	✓	✓	✓
	Complementary PWM mode 1/2/3/4	—	✓*1	—	—
Phase counting function		✓			
External pulse width measuring function		✓			
Buffer operation	Double buffer	✓			
	Simultaneous operation disable control for multiple channels	✓			
	Buffer transfer by counter clear / compare match	✓			
One-shot operation		✓			
DMAC/DTC activation		All interrupt sources			
A/D conversion start trigger		Compare match of GTADTRA or GTADTRB register			
3-phase PWM wave generator for brushless DC motor		✓(One function overall)			

Table 24.2 GPTW Functions (2/2)

Item		GPTW0 to GPTW3	GPTW4 to GPTW9	GPTW10 to GPTW14	GPTW15
Interrupt sources		10 sources • GTCCRA register compare match/input capture (GTCIAn) • GTCCRB register compare match/input capture (GTCIBn) • GTCCRC register compare match (GTCICn) • GTCCRD register compare match (GTCIDn) • Dead time error (GDTEn) • GTCCRE register compare match (GTCIEn) • GTCCRF register compare match (GTCIFn) • GTCNT counter overflow (GTPR register compare match) (GTCIVn) • GTCNT counter underflow (GTCIUn) • Cycle count end (GTCEIn)			
Interrupt skipping function	24.4.3.1 Interrupt Skipping Function by GTITC Register: Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GTCIVn) and GTCNT counter underflow (GTCIUn) (interlocked with other interrupts and A/D conversion start requests)	✓	✓	✓	✓
	24.4.3.2 Extended Interrupt Skipping Function (buffer transfer skipping function)	✓	✓	✓	✓
	24.4.3.3 A/D conversion Start Request Compare Match Skipping Function: Skipping of GTADTRA and GTADTRB register compare match	—	✓	—	—
Event operation by the ELC		✓			
Noise filter function		✓			
Logical operation between the channel output		✓			
Synchronous counter clearing/counter setting/ input capture		✓			
Timer-linked synchronization signal output		Synchronization target channels	Synchronization target channels	Synchronization target channels	Synchronization control channel
Hardware counter switching		✓			
High resolution PWM waveform generation		✓	—	—	—

✓: Possible

Note 1. The complementary PWM mode enables the output of 3-phase PWM waveforms with the use of three consecutive channels. GPTW4 to GPTW6 and GPTW7 to GPTW9 are used to realize this mode.



GTWP: General PWM Timer Write-Protection Register
GTSTR: General PWM Timer Software Start Register
GTSTP: General PWM Timer Software Stop Register
GTCLR: General PWM Timer Software Clear Register
GTSSR: General PWM Timer Start Source Select Register
GTPSR: General PWM Timer Stop Source Select Register
GTCSR: General PWM Timer Clear Source Select Register
GTUPSR: General PWM Timer Count-Up Source Select Register
GTDNSR: General PWM Timer Count-Down Source Select Register
GTICASR: General PWM Timer Input Capture Source Select Register A
GTICBSR: General PWM Timer Input Capture Source Select Register B
GTCR: General PWM Timer Control Register
GTUDDTYC: General PWM Timer Count Direction and Duty Setting Register
GTIOR: General PWM Timer I/O Control Register
GTINTAD: General PWM Timer Interrupt Output Setting Register
GTST: General PWM Timer Status Register
GTBER: General PWM Timer Buffer Enable Register
GTITC: General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register
GTCNT: General PWM Timer Counter
GTCCRA: General PWM Timer Compare Capture Register A
GTCCRB: General PWM Timer Compare Capture Register B
GTCCRC: General PWM Timer Compare Capture Register C
GTCCRD: General PWM Timer Compare Capture Register D
GTCCRE: General PWM Timer Compare Capture Register E
GTCCRF: General PWM Timer Compare Capture Register F
GTPR: General PWM Timer Period Setting Register
GTPBR: General PWM Timer Period Setting Buffer Register
GTPDBR: General PWM Timer Period Setting Double-Buffer Register

GTADTRA: A/D Conversion Start Request Timing Register A
GTADTRB: A/D Conversion Start Request Timing Register B
GTADTBRA: A/D Conversion Start Request Timing Buffer Register A
GTADTBRB: A/D Conversion Start Request Timing Buffer Register B
GTADTDBRA: A/D Conversion Start Request Timing Double-Buffer Register A
GTADTDBRB: A/D Conversion Start Request Timing Double-Buffer Register B
GTDTCR: General PWM Timer Dead Time Control Register
GTDVU: General PWM Timer Dead Time Value Register U
GTDVD: General PWM Timer Dead Time Value Register D
GTDBU: General PWM Timer Dead Time Value Buffer Register U
GTDBD: General PWM Timer Dead Time Value Buffer Register D
GTSOS: General PWM Timer Output Protection Function Status Register
GTSOTR: General PWM Timer Output Protection Function Temporary Release Register
GTADSMR: General PWM Timer A/D Conversion Start Request Signal Monitoring Register
GTEITC: General PWM Timer Extended Interrupt Skipping Counter Control Register
GTEITLI1: General PWM Timer Extended Interrupt Skipping Setting Register 1
GTEITLI2: General PWM Timer Extended Interrupt Skipping Setting Register 2
GTEITLB: General PWM Timer Extended Buffer Transfer Skipping Setting Register
GTSECSR: General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register
GTSECR: General PWM Timer Operation Enable Bit Simultaneous Control Register
GTSWSR: General PWM Timer Switch Source Select Register
GTSWOS: General PWM Timer Switch Offset Setting Register
GTSYNCFCR: General PWM Timer-linked Synchronization Control Register

Figure 24.1 GPTW Block Diagram (Sawtooth-wave PWM mode, Sawtooth-wave one-shot pulse mode, and Triangle-wave PWM mode 1/2/3)

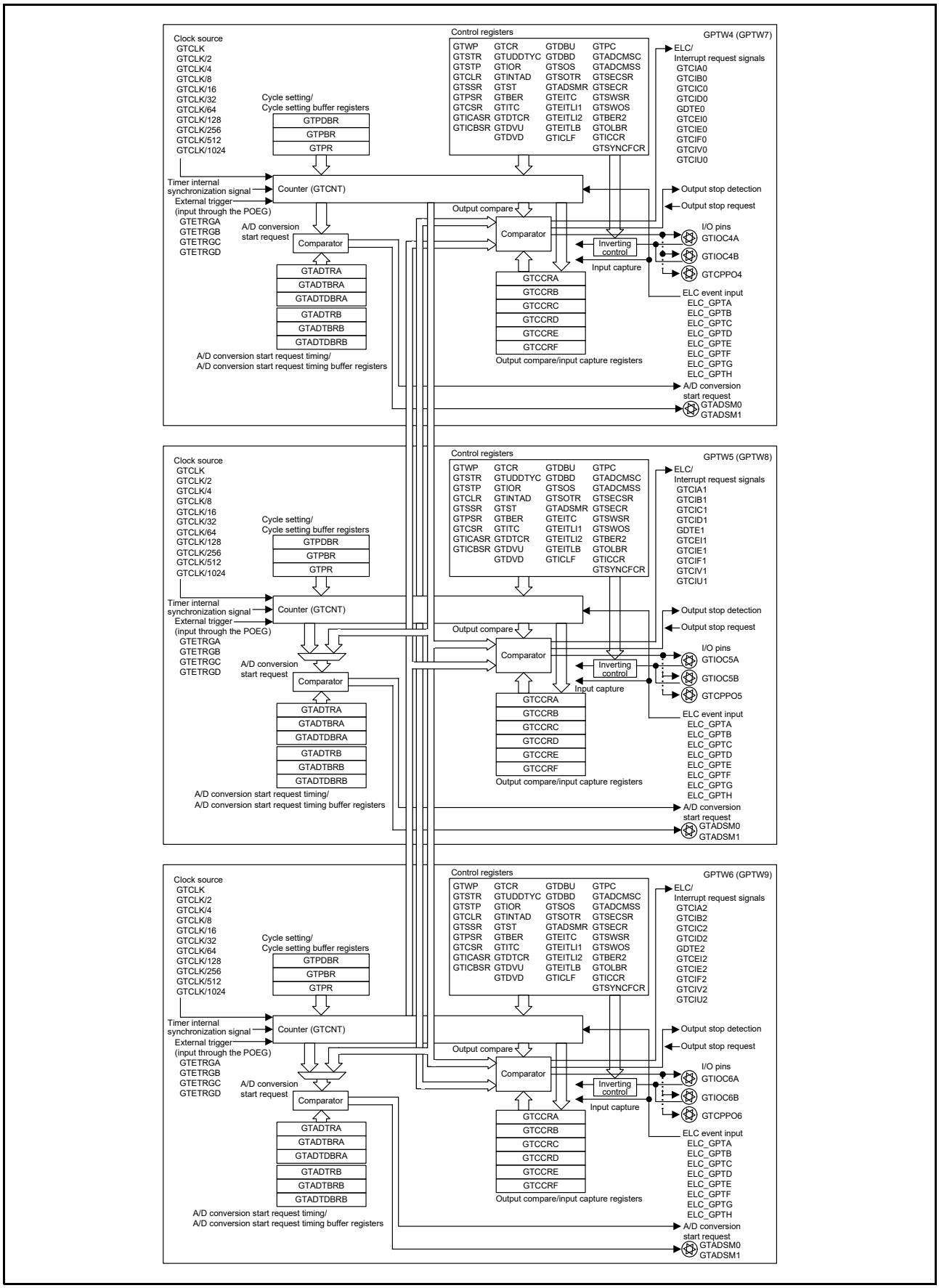
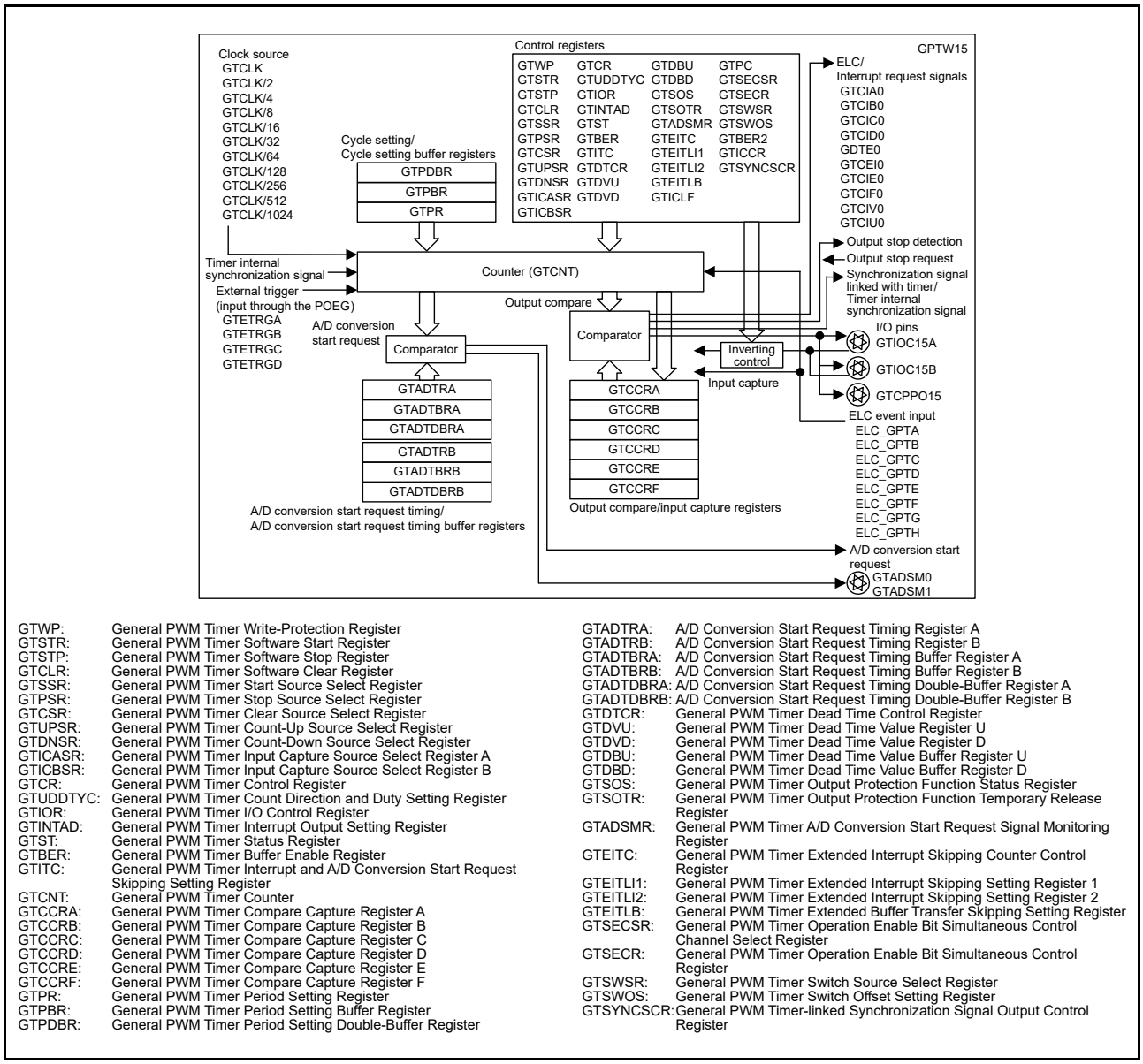


Figure 24.2 GPTW Block Diagram (Sawtooth-wave PWM mode 1/2, Sawtooth-wave one-shot pulse mode, Triangle-wave PWM mode 1/2/3, and Complementary PWM 1/2/3/4)



GTWP: General PWM Timer Write-Protection Register
GTSTR: General PWM Timer Software Start Register
GTSTP: General PWM Timer Software Stop Register
GTCLR: General PWM Timer Software Clear Register
GTSSR: General PWM Timer Start Source Select Register
GTPSR: General PWM Timer Stop Source Select Register
GTCSR: General PWM Timer Clear Source Select Register
GTUPSR: General PWM Timer Count-Up Source Select Register
GTDNSR: General PWM Timer Count-Down Source Select Register
GTICASR: General PWM Timer Input Capture Source Select Register A
GTICBSR: General PWM Timer Input Capture Source Select Register B
GTCR: General PWM Timer Control Register
GTUDDTYC: General PWM Timer Count Direction and Duty Setting Register
GTIOR: General PWM Timer I/O Control Register
GTINTAD: General PWM Timer Interrupt Output Setting Register
GTST: General PWM Timer Status Register
GTBER: General PWM Timer Buffer Enable Register
GTITC: General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register
GTCNT: General PWM Timer Counter
GTCCRA: General PWM Timer Compare Capture Register A
GTCCRB: General PWM Timer Compare Capture Register B
GTCCRC: General PWM Timer Compare Capture Register C
GTCCRD: General PWM Timer Compare Capture Register D
GTCCRE: General PWM Timer Compare Capture Register E
GTCCRF: General PWM Timer Compare Capture Register F
GTPR: General PWM Timer Period Setting Register
GTPBR: General PWM Timer Period Setting Buffer Register
GTPDBR: General PWM Timer Period Setting Double-Buffer Register

GTADTRA: A/D Conversion Start Request Timing Register A
GTADTRB: A/D Conversion Start Request Timing Register B
GTADTBRA: A/D Conversion Start Request Timing Buffer Register A
GTADTBRB: A/D Conversion Start Request Timing Buffer Register B
GTADTDBRA: A/D Conversion Start Request Timing Double-Buffer Register A
GTADTDBRB: A/D Conversion Start Request Timing Double-Buffer Register B
GTDTCR: General PWM Timer Dead Time Control Register
GTDVU: General PWM Timer Dead Time Value Register U
GTDVD: General PWM Timer Dead Time Value Register D
GTDBU: General PWM Timer Dead Time Value Buffer Register U
GTDBD: General PWM Timer Dead Time Value Buffer Register D
GTSOS: General PWM Timer Output Protection Function Status Register
GTSOTR: General PWM Timer Output Protection Function Temporary Release Register
GTADSMR: General PWM Timer A/D Conversion Start Request Signal Monitoring Register
GTEITC: General PWM Timer Extended Interrupt Skipping Counter Control Register
GTEITLI1: General PWM Timer Extended Interrupt Skipping Setting Register 1
GTEITLI2: General PWM Timer Extended Interrupt Skipping Setting Register 2
GTEITLB: General PWM Timer Extended Buffer Transfer Skipping Setting Register
GTSECSR: General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register
GTSECR: General PWM Timer Operation Enable Bit Simultaneous Control Register
GTSWSR: General PWM Timer Switch Source Select Register
GTSWOS: General PWM Timer Switch Offset Setting Register
GTSYNCSCR: General PWM Timer-linked Synchronization Signal Output Control Register

Figure 24.3 GPTW15 Block Diagram (Sawtooth-wave PWM mode, Sawtooth-wave one-shot pulse mode, and Triangle-wave PWM mode 1/2/3)

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

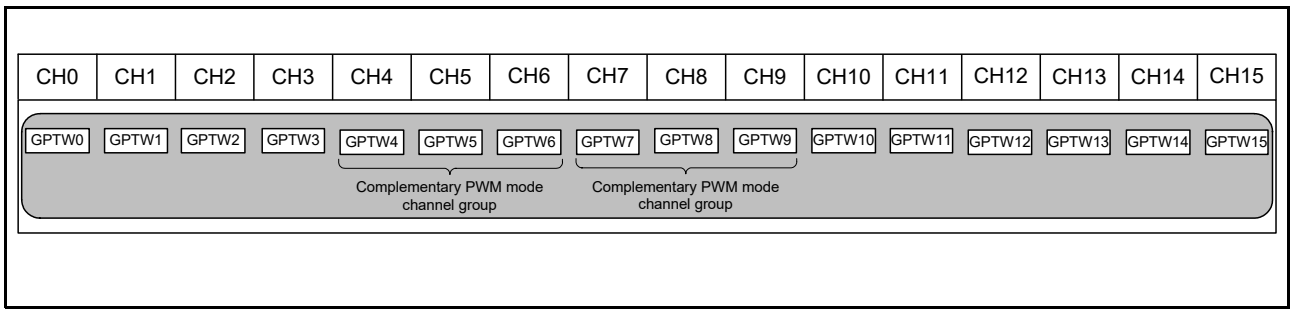


Figure 24.4 Association between GPTW channels and module names

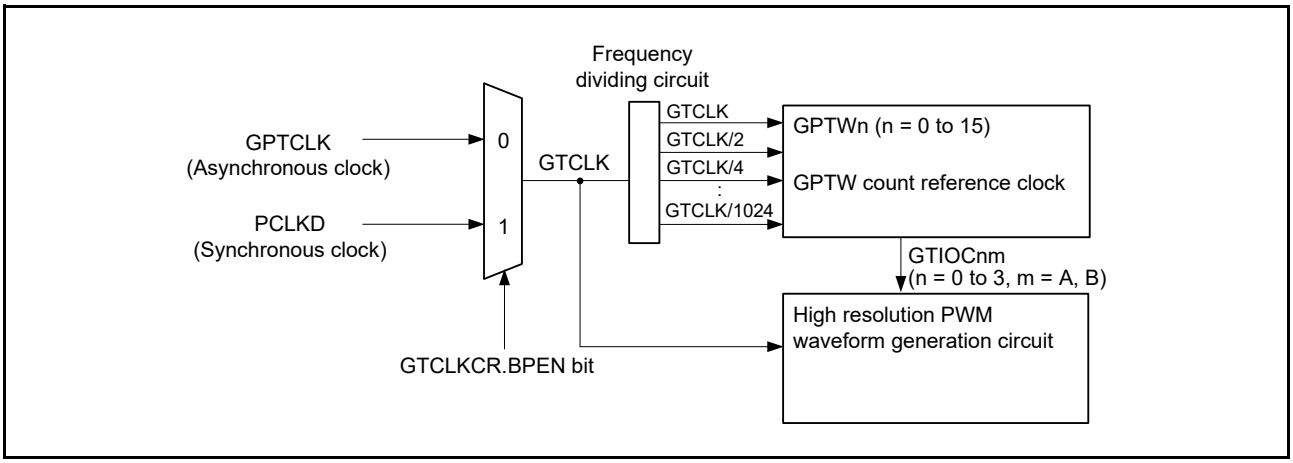


Figure 24.5 Select GPTW Count Reference Clock

Table 24.3 lists the I/O pins used in the GPTW.

Table 24.3 GPTW I/O Pins (n = 0 to 15)

Channel	Pin Name	I/O	Function
GPTW	GTETRGA	Input	External trigger input pin A (input via the POEG)
	GTETRGB	Input	External trigger input pin B (input via the POEG)
	GTETRGC	Input	External trigger input pin C (input via the POEG)
	GTETRGD	Input	External trigger input pin D (input via the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPTWn	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPOn	Output	Toggle output synchronized with PWM period*1
OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note 1. In complementary PWM mode, only GTCPPO4 and GTCPPO7 are valid.

24.2 Register Descriptions

24.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPTW0.GTWP 8742 C000h, GPTW1.GTWP 8742 C100h, GPTW2.GTWP 8742 C200h, GPTW3.GTWP 8742 C300h, GPTW4.GTWP 8742 C400h, GPTW5.GTWP 8742 C500h, GPTW6.GTWP 8742 C600h, GPTW7.GTWP 8742 C700h, GPTW8.GTWP 8742 C800h, GPTW9.GTWP 8742 C900h, GPTW10.GTWP 8742 CA00h, GPTW11.GTWP 8742 CB00h, GPTW12.GTWP 8742 CC00h, GPTW13.GTWP 8742 CD00h, GPTW14.GTWP 8742 CE00h, GPTW15.GTWP 8742 CF00h, GPTW0_NS.GTWP 0742 C000h, GPTW1_NS.GTWP 0742 C100h, GPTW2_NS.GTWP 0742 C200h, GPTW3_NS.GTWP 0742 C300h, GPTW4_NS.GTWP 0742 C400h, GPTW5_NS.GTWP 0742 C500h, GPTW6_NS.GTWP 0742 C600h, GPTW7_NS.GTWP 0742 C700h, GPTW8_NS.GTWP 0742 C800h, GPTW9_NS.GTWP 0742 C900h, GPTW10_NS.GTWP 0742 CA00h, GPTW11_NS.GTWP 0742 CB00h, GPTW12_NS.GTWP 0742 CC00h, GPTW13_NS.GTWP 0742 CD00h, GPTW14_NS.GTWP 0742 CE00h, GPTW15_NS.GTWP 0742 CF00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WP	Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b1	STRWP	GTSTR.CSTRT Bit Write Disable	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b2	STPWP	GTSTP.CSTOP Bit Write Disable	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b3	CLRWP	GTCLR.CCLR Bit Write Disable	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b4	CMNWP	Common Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	GTWP Key Code	These bits are read as 0. To modify the WP, STRWP, STPWP, CLRWP, and CMNWP bits, write A5h.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTWP register enables or disables writing to registers to prevent accidental modification.
Protection by the GTWP register only covers the CPU write operation.
Such protection does not cover a register updating generated in coordination with the CPU write.
For registers that are write enabled or disabled depending on the setting of the GTWP register, see section 24.8.1, Write-Protection for Registers.

WP Bit (Register Write Disable)

This bit enables or disables writing to the registers of GPTW.
Registers which are targets of write enable/disable are as follows:
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC*1, GTADCMSC*1, GTADCMSS, GTSWSR, GTSWOS, GTBER2*1, GTOLBR, GTICCR, GTSYNCFCR, GTSYNCSCR

Note 1. The register write protection constraint applies only to channels GPTW4 to GPTW9.

STRWP Bit (GTSTR.CSTRT Bit Write Disable)

This bit enables or disables starting the update of counter values by writing to the CSTRTn bit corresponding to the channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR registers is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel in the case of simultaneous writing to all channels.

Thus, in the case of writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel will not be updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter starts running. When the setting of the GPTW0.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not start running.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP Bit (GTSTP.CSTOP Bit Write Disable)

This bit enables or disables stopping the counter by writing to the CSTOPn bit corresponding to the channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel in the case of simultaneous writing to all channels.

Thus, in the case of writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel will not be updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is stopped. When the setting of the GPTW0.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not be stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP Bit (GTCLR.CCLR Bit Write Disable)

This bit enables or disables updating the value of the CCLRn bit by writing to the bit corresponding to the channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel in the case of simultaneous writing to all channels.

Thus, in the case of writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel will not be updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTCLR.CCLR0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP Bit (Common Register Write Disable)

This bit enables or disables updating of values by writing to the SECSELn bit (n = 0 to 15) corresponding to the channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel in the case of simultaneous writing to all channels.

Thus, in the case of writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel will not be updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) will be updated.

For example, when the setting of the GPTW0.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will update the value of the GPTW0.GTSECR register. When the setting of the GPTW0.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will not lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will not lead to updating of the value of the GPTW0.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

PRKEY[7:0] Bits (GTWP Key Code)

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

24.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPTW0.GTSTR 8742 C004h, GPTW1.GTSTR 8742 C104h, GPTW2.GTSTR 8742 C204h, GPTW3.GTSTR 8742 C304h, GPTW4.GTSTR 8742 C404h, GPTW5.GTSTR 8742 C504h, GPTW6.GTSTR 8742 C604h, GPTW7.GTSTR 8742 C704h, GPTW8.GTSTR 8742 C804h, GPTW9.GTSTR 8742 C904h, GPTW10.GTSTR 8742 CA04h, GPTW11.GTSTR 8742 CB04h, GPTW12.GTSTR 8742 CC04h, GPTW13.GTSTR 8742 CD04h, GPTW14.GTSTR 8742 CE04h, GPTW15.GTSTR 8742 CF04h, GPTW0_NS.GTSTR 0742 C004h, GPTW1_NS.GTSTR 0742 C104h, GPTW2_NS.GTSTR 0742 C204h, GPTW3_NS.GTSTR 0742 C304h, GPTW4_NS.GTSTR 0742 C404h, GPTW5_NS.GTSTR 0742 C504h, GPTW6_NS.GTSTR 0742 C604h, GPTW7_NS.GTSTR 0742 C704h, GPTW8_NS.GTSTR 0742 C804h, GPTW9_NS.GTSTR 0742 C904h, GPTW10_NS.GTSTR 0742 CA04h, GPTW11_NS.GTSTR 0742 CB04h, GPTW12_NS.GTSTR 0742 CC04h, GPTW13_NS.GTSTR 0742 CD04h, GPTW14_NS.GTSTR 0742 CE04h, GPTW15_NS.GTSTR 0742 CF04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSTRT15	CSTRT14	CSTRT13	CSTRT12	CSTRT11	CSTRT10	CSTRT9	CSTRT8	CSTRT7	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTRT0	Channel 0 Count Start	When reading 0: The counter is stopped. 1: The counter is running. When writing 0: Ignored. 1: The counter starts running.	R/W
b1	CSTRT1	Channel 1 Count Start		R/W
b2	CSTRT2	Channel 2 Count Start		R/W
b3	CSTRT3	Channel 3 Count Start		R/W
b4	CSTRT4	Channel 4 Count Start		R/W
b5	CSTRT5	Channel 5 Count Start		R/W
b6	CSTRT6	Channel 6 Count Start		R/W
b7	CSTRT7	Channel 7 Count Start		R/W
b8	CSTRT8	Channel 8 Count Start		R/W
b9	CSTRT9	Channel 9 Count Start		R/W
b10	CSTRT10	Channel 10 Count Start		R/W
b11	CSTRT11	Channel 11 Count Start		R/W
b12	CSTRT12	Channel 12 Count Start		R/W
b13	CSTRT13	Channel 13 Count Start		R/W
b14	CSTRT14	Channel 14 Count Start		R/W
b15	CSTRT15	Channel 15 Count Start		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTR register starts the GTCNT counter operation.

A bit position of the GTSTR register indicates a channel number. The GTSTR register of each channel is a common register, and updating the GTSTR register of any channel by setting bits to 1 starts the operation of the GTCNT counters in all the channels corresponding to those bit positions. No change in counter operation or register value occurs when writing 0 to the bit.

In complementary PWM mode, only writing to the CSTRTn bit of the master channel is valid. The bit on the slave channels reflects the bit value of the master channel.

CSTRTn Bit (Channel n Count Start) (n = 0 to 15)

This bit starts the GTCNT counter for channel n.

A read value indicates the state of the counter operation in a corresponding channel (GTCR.CST bit). The bit with 0 indicates that the counter is stopped, while the bit with 1 indicates that the counter is in operation.

24.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPTW0.GTSTP 8742 C008h, GPTW1.GTSTP 8742 C108h, GPTW2.GTSTP 8742 C208h, GPTW3.GTSTP 8742 C308h, GPTW4.GTSTP 8742 C408h, GPTW5.GTSTP 8742 C508h, GPTW6.GTSTP 8742 C608h, GPTW7.GTSTP 8742 C708h, GPTW8.GTSTP 8742 C808h, GPTW9.GTSTP 8742 C908h, GPTW10.GTSTP 8742 CA08h, GPTW11.GTSTP 8742 CB08h, GPTW12.GTSTP 8742 CC08h, GPTW13.GTSTP 8742 CD08h, GPTW14.GTSTP 8742 CE08h, GPTW15.GTSTP 8742 CF08h, GPTW0_NS.GTSTP 0742 C008h, GPTW1_NS.GTSTP 0742 C108h, GPTW2_NS.GTSTP 0742 C208h, GPTW3_NS.GTSTP 0742 C308h, GPTW4_NS.GTSTP 0742 C408h, GPTW5_NS.GTSTP 0742 C508h, GPTW6_NS.GTSTP 0742 C608h, GPTW7_NS.GTSTP 0742 C708h, GPTW8_NS.GTSTP 0742 C808h, GPTW9_NS.GTSTP 0742 C908h, GPTW10_NS.GTSTP 0742 CA08h, GPTW11_NS.GTSTP 0742 CB08h, GPTW12_NS.GTSTP 0742 CC08h, GPTW13_NS.GTSTP 0742 CD08h, GPTW14_NS.GTSTP 0742 CE08h, GPTW15_NS.GTSTP 0742 CF08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSTOP15	CSTOP14	CSTOP13	CSTOP12	CSTOP11	CSTOP10	CSTOP9	CSTOP8	CSTOP7	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CSTOP0	Channel 0 Count Stop	When reading 0: The counter is operating. 1: The counter is stopped. When writing 0: Ignored 1: Stops the counter	R/W
b1	CSTOP1	Channel 1 Count Stop		R/W
b2	CSTOP2	Channel 2 Count Stop		R/W
b3	CSTOP3	Channel 3 Count Stop		R/W
b4	CSTOP4	Channel 4 Count Stop		R/W
b5	CSTOP5	Channel 5 Count Stop		R/W
b6	CSTOP6	Channel 6 Count Stop		R/W
b7	CSTOP7	Channel 7 Count Stop		R/W
b8	CSTOP8	Channel 8 Count Stop		R/W
b9	CSTOP9	Channel 9 Count Stop		R/W
b10	CSTOP10	Channel 10 Count Stop		R/W
b11	CSTOP11	Channel 11 Count Stop		R/W
b12	CSTOP12	Channel 12 Count Stop		R/W
b13	CSTOP13	Channel 13 Count Stop		R/W
b14	CSTOP14	Channel 14 Count Stop		R/W
b15	CSTOP15	Channel 15 Count Stop		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTP register stops the GTCNT counter operation.

A bit position of the GTSTP register indicates a channel number. The GTSTP register of each channel is a common register, and updating the GTSTP register of any channel by setting bits to 1 stops the operation of the GTCNT counters in all the channels corresponding to those bit positions. No change in counter operation or register value occurs when writing 0 to the bit.

In complementary PWM mode, only writing to the CSTOPn bit of the master channel is valid. The bit on the slave channels reflects the bit value of the master channel.

CSTOPn Bit (Channel n Count Stop) (n = 0 to 15)

This bit stops the GTCNT counter for channel n.

A read value indicates the state of the counter operation in a corresponding channel (inverted GTCR.CST bit). The bit with 0 indicates that the counter is in operation, while the bit with 1 indicates that the counter is stopped.

24.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPTW0.GTCLR 8742 C00Ch, GPTW1.GTCLR 8742 C10Ch, GPTW2.GTCLR 8742 C20Ch, GPTW3.GTCLR 8742 C30Ch, GPTW4.GTCLR 8742 C40Ch, GPTW5.GTCLR 8742 C50Ch, GPTW6.GTCLR 8742 C60Ch, GPTW7.GTCLR 8742 C70Ch, GPTW8.GTCLR 8742 C80Ch, GPTW9.GTCLR 8742 C90Ch, GPTW10.GTCLR 8742 CA0Ch, GPTW11.GTCLR 8742 CB0Ch, GPTW12.GTCLR 8742 CC0Ch, GPTW13.GTCLR 8742 CD0Ch, GPTW14.GTCLR 8742 CE0Ch, GPTW15.GTCLR 8742 CF0Ch, GPTW0_NS.GTCLR 0742 C00Ch, GPTW1_NS.GTCLR 0742 C10Ch, GPTW2_NS.GTCLR 0742 C20Ch, GPTW3_NS.GTCLR 0742 C30Ch, GPTW4_NS.GTCLR 0742 C40Ch, GPTW5_NS.GTCLR 0742 C50Ch, GPTW6_NS.GTCLR 0742 C60Ch, GPTW7_NS.GTCLR 0742 C70Ch, GPTW8_NS.GTCLR 0742 C80Ch, GPTW9_NS.GTCLR 0742 C90Ch, GPTW10_NS.GTCLR 0742 CA0Ch, GPTW11_NS.GTCLR 0742 CB0Ch, GPTW12_NS.GTCLR 0742 CC0Ch, GPTW13_NS.GTCLR 0742 CD0Ch, GPTW14_NS.GTCLR 0742 CE0Ch, GPTW15_NS.GTCLR 0742 CF0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR15	CCLR14	CCLR13	CCLR12	CCLR11	CCLR10	CCLR9	CCLR8	CCLR7	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCLR0	Channel 0 Counter Clear	0: Ignored. 1: The counter is to be cleared.	W
b1	CCLR1	Channel 1 Counter Clear		W
b2	CCLR2	Channel 2 Counter Clear		W
b3	CCLR3	Channel 3 Counter Clear		W
b4	CCLR4	Channel 4 Counter Clear		W
b5	CCLR5	Channel 5 Counter Clear		W
b6	CCLR6	Channel 6 Counter Clear		W
b7	CCLR7	Channel 7 Counter Clear		W
b8	CCLR8	Channel 8 Counter Clear		W
b9	CCLR9	Channel 9 Counter Clear		W
b10	CCLR10	Channel 10 Counter Clear		W
b11	CCLR11	Channel 11 Counter Clear		W
b12	CCLR12	Channel 12 Counter Clear		W
b13	CCLR13	Channel 13 Counter Clear		W
b14	CCLR14	Channel 14 Counter Clear		W
b15	CCLR15	Channel 15 Counter Clear		W
b31 to b16	—	Reserved	The write value should be 0.	W

The GTCLR register is a write-only register that clears the GTCNT counter.

A bit position of the GTCLR register indicates a channel number. The GTCLR register of each channel is a common register, and updating the GTCLR register of any channel by setting bits to 1 clears the GTCNT counters to 0 in all the channels corresponding to those bit positions. No change in counter operation or register value occurs when writing 0 to the bit.

In complementary PWM mode, only writing to the CCLRn bit of the master channel is valid. The bit on the slave channels reflects the bit value of the master channel.

CCLRn Bit (Channel n Counter Clear) (n = 0 to 15)

When the counting direction flag is set for decrementation (GTST.TUCF flag = 0) with sawtooth-wave mode selected in the GTCR.MD[2:0] or the GTCR.MD[3:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0000 0000h with other settings.

24.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPTW0.GTSSR 8742 C010h, GPTW1.GTSSR 8742 C110h, GPTW2.GTSSR 8742 C210h, GPTW3.GTSSR 8742 C310h, GPTW4.GTSSR 8742 C410h, GPTW5.GTSSR 8742 C510h, GPTW6.GTSSR 8742 C610h, GPTW7.GTSSR 8742 C710h, GPTW8.GTSSR 8742 C810h, GPTW9.GTSSR 8742 C910h, GPTW10.GTSSR 8742 CA10h, GPTW11.GTSSR 8742 CB10h, GPTW12.GTSSR 8742 CC10h, GPTW13.GTSSR 8742 CD10h, GPTW14.GTSSR 8742 CE10h, GPTW15.GTSSR 8742 CF10h, GPTW0_NS.GTSSR 0742 C010h, GPTW1_NS.GTSSR 0742 C110h, GPTW2_NS.GTSSR 0742 C210h, GPTW3_NS.GTSSR 0742 C310h, GPTW4_NS.GTSSR 0742 C410h, GPTW5_NS.GTSSR 0742 C510h, GPTW6_NS.GTSSR 0742 C610h, GPTW7_NS.GTSSR 0742 C710h, GPTW8_NS.GTSSR 0742 C810h, GPTW9_NS.GTSSR 0742 C910h, GPTW10_NS.GTSSR 0742 CA10h, GPTW11_NS.GTSSR 0742 CB10h, GPTW12_NS.GTSSR 0742 CC10h, GPTW13_NS.GTSSR 0742 CD10h, GPTW14_NS.GTSSR 0742 CE10h, GPTW15_NS.GTSSR 0742 CF10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGA signal. 1 0: The counter starts at a falling edge of the GTETRGA signal. 1 1: The counter starts at both edges of the GTETRGA signal.	R/W *1
b1	SSGTRGAF			R/W *1
b2	SSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGB signal. 1 0: The counter starts at a falling edge of the GTETRGB signal. 1 1: The counter starts at both edges of the GTETRGB signal.	R/W *1
b3	SSGTRGBF			R/W *1
b4	SSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGC signal. 1 0: The counter starts at a falling edge of the GTETRGC signal. 1 1: The counter starts at both edges of the GTETRGC signal.	R/W *1
b5	SSGTRGCF			R/W *1
b6	SSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGD signal. 1 0: The counter starts at a falling edge of the GTETRGD signal. 1 1: The counter starts at both edges of the GTETRGD signal.	R/W *1
b7	SSGTRGDF			R/W *1
b8	SSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnA signal.	R/W
b9	SSCARBH			R/W
b10	SSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnA signal.	R/W
b11	SSCAFBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b12	SSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnB signal.	R/W
b13	SSCBRAH			R/W
b14	SSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnB signal.	R/W
b15	SSCBFAH			R/W
b16	SSELCA	ELCA Event Source Count Start Enable	0: Disables count start by the ELCA event input 1: Enables count start by the ELCA event input	R/W *1
b17	SSELCB	ELCB Event Source Count Start Enable	0: Disables count start by the ELCB event input 1: Enables count start by the ELCB event input	R/W *1
b18	SSELCC	ELCC Event Source Count Start Enable	0: Disables count start by the ELCC event input 1: Enables count start by the ELCC event input	R/W *1
b19	SSELCD	ELCD Event Source Count Start Enable	0: Disables count start by the ELCD event input 1: Enables count start by the ELCD event input	R/W *1
b20	SSELCE	ELCE Event Source Count Start Enable	0: Disables count start by the ELCE event input 1: Enables count start by the ELCE event input	R/W *1
b21	SSELCF	ELCF Event Source Count Start Enable	0: Disables count start by the ELCF event input 1: Enables count start by the ELCF event input	R/W *1
b22	SSELCG	ELCG Event Source Count Start Enable	0: Disables count start by the ELCG event input 1: Enables count start by the ELCG event input	R/W *1
b23	SSELCH	ELCH Event Source Count Start Enable	0: Disables count start by the ELCH event input 1: Enables count start by the ELCH event input	R/W *1
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTRT	Software Source Count Start Enable	0: Disables count start by the GTSTR register 1: Enables count start by the GTSTR register	R/W *1

n = 0 to 15

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW4 to GPTW9.

The GTSSR register sets a count start source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPTW0.GTPSR 8742 C014h, GPTW1.GTPSR 8742 C114h, GPTW2.GTPSR 8742 C214h, GPTW3.GTPSR 8742 C314h, GPTW4.GTPSR 8742 C414h, GPTW5.GTPSR 8742 C514h, GPTW6.GTPSR 8742 C614h, GPTW7.GTPSR 8742 C714h, GPTW8.GTPSR 8742 C814h, GPTW9.GTPSR 8742 C914h, GPTW10.GTPSR 8742 CA14h, GPTW11.GTPSR 8742 CB14h, GPTW12.GTPSR 8742 CC14h, GPTW13.GTPSR 8742 CD14h, GPTW14.GTPSR 8742 CE14h, GPTW15.GTPSR 8742 CF14h, GPTW0_NS.GTPSR 0742 C014h, GPTW1_NS.GTPSR 0742 C114h, GPTW2_NS.GTPSR 0742 C214h, GPTW3_NS.GTPSR 0742 C314h, GPTW4_NS.GTPSR 0742 C414h, GPTW5_NS.GTPSR 0742 C514h, GPTW6_NS.GTPSR 0742 C614h, GPTW7_NS.GTPSR 0742 C714h, GPTW8_NS.GTPSR 0742 C814h, GPTW9_NS.GTPSR 0742 C914h, GPTW10_NS.GTPSR 0742 CA14h, GPTW11_NS.GTPSR 0742 CB14h, GPTW12_NS.GTPSR 0742 CC14h, GPTW13_NS.GTPSR 0742 CD14h, GPTW14_NS.GTPSR 0742 CE14h, GPTW15_NS.GTPSR 0742 CF14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGA signal. 1 0: The counter stops at a falling edge of the GTETRGA signal. 1 1: The counter stops at both edges of the GTETRGA signal.	R/W *1
b1	PSGTRGAF			**R/W *1**
b2	PSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGB signal. 1 0: The counter stops at a falling edge of the GTETRGB signal. 1 1: The counter stops at both edges of the GTETRGB signal.	**R/W *1**
b3	PSGTRGBF			**R/W *1**
b4	PSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGC signal. 1 0: The counter stops at a falling edge of the GTETRGC signal. 1 1: The counter stops at both edges of the GTETRGC signal.	**R/W *1**
b5	PSGTRGCF			**R/W *1**
b6	PSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGD signal. 1 0: The counter stops at a falling edge of the GTETRGD signal. 1 1: The counter stops at both edges of the GTETRGD signal.	**R/W *1**
b7	PSGTRGDF			**R/W *1**
b8	PSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnA signal.	R/W
b9	PSCARBH			R/W
b10	PSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnA signal.	R/W
b11	PSCAFBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b12	PSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnB signal.	R/W
b13	PSCBRAH			R/W
b14	PSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnB signal.	R/W
b15	PSCBFAH			R/W
b16	PSELCA	ELCA Event Source Count Stop Enable	0: Disables count stop by the ELCA event input 1: Enables count stop by the ELCA event input	**R/W** *1
b17	PSELCB	ELCB Event Source Count Stop Enable	0: Disables count stop by the ELCB event input 1: Enables count stop by the ELCB event input	**R/W** *1
b18	PSELCC	ELCC Event Source Count Stop Enable	0: Disables count stop by the ELCC event input 1: Enables count stop by the ELCC event input	**R/W** *1
b19	PSELCD	ELCD Event Source Count Stop Enable	0: Disables count stop by the ELCD event input 1: Enables count stop by the ELCD event input	**R/W** *1
b20	PSELCE	ELCE Event Source Count Stop Enable	0: Disables count stop by the ELCE event input 1: Enables count stop by the ELCE event input	**R/W** *1
b21	PSELCF	ELCF Event Source Count Stop Enable	0: Disables count stop by the ELCF event input 1: Enables count stop by the ELCF event input	**R/W** *1
b22	PSELCG	ELCG Event Source Count Stop Enable	0: Disables count stop by the ELCG event input 1: Enables count stop by the ELCG event input	**R/W** *1
b23	PSELCH	ELCH Event Source Count Stop Enable	0: Disables count stop by the ELCH event input 1: Enables count stop by the ELCH event input	**R/W** *1
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTOP	Software Source Count Stop Enable	0: Disables count stop by the GTSTP register 1: Enables count stop by the GTSTP register	**R/W** *1

n = 0 to 15

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW4 to GPTW9.

The GTPSR register sets a count stop source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPTW0.GTCSR 8742 C018h, GPTW1.GTCSR 8742 C118h, GPTW2.GTCSR 8742 C218h, GPTW3.GTCSR 8742 C318h, GPTW4.GTCSR 8742 C418h, GPTW5.GTCSR 8742 C518h, GPTW6.GTCSR 8742 C618h, GPTW7.GTCSR 8742 C718h, GPTW8.GTCSR 8742 C818h, GPTW9.GTCSR 8742 C918h, GPTW10.GTCSR 8742 CA18h, GPTW11.GTCSR 8742 CB18h, GPTW12.GTCSR 8742 CC18h, GPTW13.GTCSR 8742 CD18h, GPTW14.GTCSR 8742 CE18h, GPTW15.GTCSR 8742 CF18h, GPTW0_NS.GTCSR 0742 C018h, GPTW1_NS.GTCSR 0742 C118h, GPTW2_NS.GTCSR 0742 C218h, GPTW3_NS.GTCSR 0742 C318h, GPTW4_NS.GTCSR 0742 C418h, GPTW5_NS.GTCSR 0742 C518h, GPTW6_NS.GTCSR 0742 C618h, GPTW7_NS.GTCSR 0742 C718h, GPTW8_NS.GTCSR 0742 C818h, GPTW9_NS.GTCSR 0742 C918h, GPTW10_NS.GTCSR 0742 CA18h, GPTW11_NS.GTCSR 0742 CB18h, GPTW12_NS.GTCSR 0742 CC18h, GPTW13_NS.GTCSR 0742 CD18h, GPTW14_NS.GTCSR 0742 CE18h, GPTW15_NS.GTCSR 0742 CF18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGA signal. 1 0: The counter is cleared at a falling edge of the GTETRGA signal. 1 1: The counter is cleared at both edges of the GTETRGA signal.	R/W *1
b1	CSGTRGAF			R/W *1
b2	CSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGB signal. 1 0: The counter is cleared at a falling edge of the GTETRGB signal. 1 1: The counter is cleared at both edges of the GTETRGB signal.	R/W *1
b3	CSGTRGBF			R/W *1
b4	CSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGC signal. 1 0: The counter is cleared at a falling edge of the GTETRGC signal. 1 1: The counter is cleared at both edges of the GTETRGC signal.	R/W *1
b5	CSGTRGCF			R/W *1
b6	CSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGD signal. 1 0: The counter is cleared at a falling edge of the GTETRGD signal. 1 1: The counter is cleared at both edges of the GTETRGD signal.	R/W *1
b7	CSGTRGDF			R/W *1

Bit	Symbol	Bit Name	Description	R/W
b8	CSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnA signal.	R/W
b9	CSCARBH			R/W
b10	CSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnA signal.	R/W
b11	CSCAFBH			R/W
b12	CSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnB signal.	R/W
b13	CSCBRAH			R/W
b14	CSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnB signal.	R/W
b15	CSCBFAH			R/W
b16	CSELCA	ELCA Event Source Counter Clear Enable	0: Disables counter clear by the ELCA event input 1: Enables counter clear by the ELCA event input	R/W *1
b17	CSELCB	ELCB Event Source Counter Clear Enable	0: Disables counter clear by the ELCB event input 1: Enables counter clear by the ELCB event input	R/W *1
b18	CSELCC	ELCC Event Source Counter Clear Enable	0: Disables counter clear by the ELCC event input 1: Enables counter clear by the ELCC event input	R/W *1
b19	CSELCD	ELCD Event Source Counter Clear Enable	0: Disables counter clear by the ELCD event input 1: Enables counter clear by the ELCD event input	R/W *1
b20	CSELCE	ELCE Event Source Counter Clear Enable	0: Disables counter clear by the ELCE event input 1: Enables counter clear by the ELCE event input	R/W *1
b21	CSELCF	ELCF Event Source Counter Clear Enable	0: Disables counter clear by the ELCF event input 1: Enables counter clear by the ELCF event input	R/W *1
b22	CSELCG	ELCG Event Source Counter Clear Enable	0: Disables counter clear by the ELCG event input 1: Enables counter clear by the ELCG event input	R/W *1
b23	CSELCH	ELCH Event Source Counter Clear Enable	0: Disables counter clear by the ELCH event input 1: Enables counter clear by the ELCH event input	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b26 to b24	CSCMSC[2:0]	Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable	b26 b25 b24 0 0 0: Counter clear disabled by Compare match/Input capture/Synchronous counter clearing group. 0 0 1: Counter clear enabled at the GTCCRA register compare match/Input capture. 0 1 0: Counter clear enabled at the GTCCRB register compare match/Input capture. 0 1 1: Counter clear enabled at the GTCCRC register compare match. 1 0 0: Counter clear enabled at the GTCCRD register compare match. 1 0 1: Counter clear enabled at the GTCCRE register compare match. 1 1 0: Counter clear enabled at the GTCCRF register compare match. 1 1 1: Counter clear enabled at the synchronous counter clearing group.	R/W
b27	CP1CCE	Complementary PWM Mode 1 Crest Source Counter Clear Enable*3	0: Counter clear disabled at the crest of complementary PWM mode 1 1: Counter clear enabled at the crest of complementary PWM mode 1	R/W *1
b30 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CCLR	Software Source Counter Clear Enable	0: Disables counter clear by the GTCLR register 1: Enables counter clear by the GTCLR register	R/W *1

n = 0 to 15

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW4 to GPTW9.

Note 3. This bit in each of GPTW0 to GPTW3 and GPTW10 to GPTW15 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTCSR register sets a counter clear source for the GTCNT counter.

Counter clearing can be executed whether the counter is running (the GTCR.CST bit = 1) or stopped (the GTCR.CST bit = 0).

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

CSCMSC[2:0] Bits (Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable)

Select whether to enable or disable for the counter clear of the GTCNT counter by compare match/input capture/ synchronous counter clearing group. When counter clearing in response to a match in comparison or input capture is enabled, either can be used as a source for synchronous clearing by inter channel cooperation as described in section 24.3.9.3, Synchronous Clear Operation by Inter Channel Cooperation.

In addition, an input capture triggered by a source from another channel can be used as a source for counter clearing (by setting the GTICASR.ASOC bit or GTICBSR.BSOC bit to 1). Input capture triggered by a source from another channel cannot be used as the synchronous clearing source in interlinked channel operation but can be used as a counter-clearing source for the channel itself.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE Bit (Complementary PWM Mode 1 Crest Source Counter Clear Enable)

Select enable or disable for the counter clear at the crest of complementary PWM mode 1.

To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.

It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counters of the slave channels 1 and 2 in complementary PWM mode.

24.2.8 General PWM Timer Count-Up Source Select Register (GTUPSR)

Address(es): GPTW0.GTUPSR 8742 C01Ch, GPTW1.GTUPSR 8742 C11Ch, GPTW2.GTUPSR 8742 C21Ch, GPTW3.GTUPSR 8742 C31Ch, GPTW4.GTUPSR 8742 C41Ch, GPTW5.GTUPSR 8742 C51Ch, GPTW6.GTUPSR 8742 C61Ch, GPTW7.GTUPSR 8742 C71Ch, GPTW8.GTUPSR 8742 C81Ch, GPTW9.GTUPSR 8742 C91Ch, GPTW10.GTUPSR 8742 CA1Ch, GPTW11.GTUPSR 8742 CB1Ch, GPTW12.GTUPSR 8742 CC1Ch, GPTW13.GTUPSR 8742 CD1Ch, GPTW14.GTUPSR 8742 CE1Ch, GPTW15.GTUPSR 8742 CF1Ch, GPTW0_NS.GTUPSR 0742 C01Ch, GPTW1_NS.GTUPSR 0742 C11Ch, GPTW2_NS.GTUPSR 0742 C21Ch, GPTW3_NS.GTUPSR 0742 C31Ch, GPTW4_NS.GTUPSR 0742 C41Ch, GPTW5_NS.GTUPSR 0742 C51Ch, GPTW6_NS.GTUPSR 0742 C61Ch, GPTW7_NS.GTUPSR 0742 C71Ch, GPTW8_NS.GTUPSR 0742 C81Ch, GPTW9_NS.GTUPSR 0742 C91Ch, GPTW10_NS.GTUPSR 0742 CA1Ch, GPTW11_NS.GTUPSR 0742 CB1Ch, GPTW12_NS.GTUPSR 0742 CC1Ch, GPTW13_NS.GTUPSR 0742 CD1Ch, GPTW14_NS.GTUPSR 0742 CE1Ch, GPTW15_NS.GTUPSR 0742 CF1Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGA signal. 1 0: The counter is incremented at a falling edge of the GTETRGA signal. 1 1: The counter is incremented at both edges of the GTETRGA signal.	R/W
b1	USGTRGAF			R/W
b2	USGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGB signal. 1 0: The counter is incremented at a falling edge of the GTETRGB signal. 1 1: The counter is incremented at both edges of the GTETRGB signal.	R/W
b3	USGTRGBF			R/W
b4	USGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGC signal. 1 0: The counter is incremented at a falling edge of the GTETRGC signal. 1 1: The counter is incremented at both edges of the GTETRGC signal.	R/W
b5	USGTRGCF			R/W
b6	USGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGD signal. 1 0: The counter is incremented at a falling edge of the GTETRGD signal. 1 1: The counter is incremented at both edges of the GTETRGD signal.	R/W
b7	USGTRGDF			R/W

Bit	Symbol	Bit Name	Description	R/W
b8	USCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnA signal.	R/W
b9	USCARBH			R/W
b10	USCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnA signal.	R/W
b11	USCAFBH			R/W
b12	USCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnB signal.	R/W
b13	USCBRAH			R/W
b14	USCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnB signal.	R/W
b15	USCBFAH			R/W
b16	USELCA	ELCA Event Source Count-Up Enable	0: Disables count-up by the ELCA event input 1: Enables count-up by the ELCA event input	R/W
b17	USELCB	ELCB Event Source Count-Up Enable	0: Disables count-up by the ELCB event input 1: Enables count-up by the ELCB event input	R/W
b18	USELCC	ELCC Event Source Count-Up Enable	0: Disables count-up by the ELCC event input 1: Enables count-up by the ELCC event input	R/W
b19	USELCD	ELCD Event Source Count-Up Enable	0: Disables count-up by the ELCD event input 1: Enables count-up by the ELCD event input	R/W
b20	USELCE	ELCE Event Source Count-Up Enable	0: Disables count-up by the ELCE event input 1: Enables count-up by the ELCE event input	R/W
b21	USELCF	ELCF Event Source Count-Up Enable	0: Disables count-up by the ELCF event input 1: Enables count-up by the ELCF event input	R/W
b22	USELCG	ELCG Event Source Count-Up Enable	0: Disables count-up by the ELCG event input 1: Enables count-up by the ELCG event input	R/W
b23	USELCH	ELCH Event Source Count-Up Enable	0: Disables count-up by the ELCH event input 1: Enables count-up by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27to b24	USILVL[3:0]	External Input Level Source Count-Up Enable	b27 b24 0 0 0 0: Disable count-up by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-up by GTIOCnA pin input level Low. 0 0 1 1: Enable count-up by GTIOCnA pin input level High. 0 1 0 0: Enable count-up by GTIOCnB pin input level Low. 0 1 0 1: Enable count-up by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-up by GTETRGA pin input level Low. 1 0 0 1: Enable count-up by GTETRGA pin input level High. 1 0 1 0: Enable count-up by GTETRGB pin input level Low. 1 0 1 1: Enable count-up by GTETRGB pin input level High. 1 1 0 0: Enable count-up by GTETRGC pin input level Low. 1 1 0 1: Enable count-up by GTETRGC pin input level High. 1 1 1 0: Enable count-up by GTETRGD pin input level Low. 1 1 1 1: Enable count-up by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 15

The GTUPSR register sets a count-up source for the GTCNT counter.

The GTUPSR register is primarily used as described in 24.3.12, Phase Counting Mode, and 24.3.13, External Pulse Width Measuring Function.

When at least one bit among bits in the GTUPSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-up by a source set as 1 by the GTUPSR register is executed.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.9 General PWM Timer Count-Down Source Select Register (GTDNSR)

Address(es): GPTW0.GTDNSR 8742 C020h, GPTW1.GTDNSR 8742 C120h, GPTW2.GTDNSR 8742 C220h, GPTW3.GTDNSR 8742 C320h, GPTW4.GTDNSR 8742 C420h, GPTW5.GTDNSR 8742 C520h, GPTW6.GTDNSR 8742 C620h, GPTW7.GTDNSR 8742 C720h, GPTW8.GTDNSR 8742 C820h, GPTW9.GTDNSR 8742 C920h, GPTW10.GTDNSR 8742 CA20h, GPTW11.GTDNSR 8742 CB20h, GPTW12.GTDNSR 8742 CC20h, GPTW13.GTDNSR 8742 CD20h, GPTW14.GTDNSR 8742 CE20h, GPTW15.GTDNSR 8742 CF20h, GPTW0_NS.GTDNSR 0742 C020h, GPTW1_NS.GTDNSR 0742 C120h, GPTW2_NS.GTDNSR 0742 C220h, GPTW3_NS.GTDNSR 0742 C320h, GPTW4_NS.GTDNSR 0742 C420h, GPTW5_NS.GTDNSR 0742 C520h, GPTW6_NS.GTDNSR 0742 C620h, GPTW7_NS.GTDNSR 0742 C720h, GPTW8_NS.GTDNSR 0742 C820h, GPTW9_NS.GTDNSR 0742 C920h, GPTW10_NS.GTDNSR 0742 CA20h, GPTW11_NS.GTDNSR 0742 CB20h, GPTW12_NS.GTDNSR 0742 CC20h, GPTW13_NS.GTDNSR 0742 CD20h, GPTW14_NS.GTDNSR 0742 CE20h, GPTW15_NS.GTDNSR 0742 CF20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGA signal. 1 0: The counter is decremented at a falling edge of the GTETRGA signal. 1 1: The counter is decremented at both edges of the GTETRGA signal.	R/W
b1	DSGTRGAF			R/W
b2	DSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGB signal. 1 0: The counter is decremented at a falling edge of the GTETRGB signal. 1 1: The counter is decremented at both edges of the GTETRGB signal.	R/W
b3	DSGTRGBF			R/W
b4	DSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGC signal. 1 0: The counter is decremented at a falling edge of the GTETRGC signal. 1 1: The counter is decremented at both edges of the GTETRGC signal.	R/W
b5	DSGTRGCF			R/W
b6	DSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGD signal. 1 0: The counter is decremented at a falling edge of the GTETRGD signal. 1 1: The counter is decremented at both edges of the GTETRGD signal.	R/W
b7	DSGTRGDF			R/W

Bit	Symbol	Bit Name	Description	R/W
b8	DSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnA signal.	R/W
b9	DSCARBH			R/W
b10	DSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnA signal.	R/W
b11	DSCAFBH			R/W
b12	DSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnB signal.	R/W
b13	DSCBRAH			R/W
b14	DSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnB signal.	R/W
b15	DSCBFAH			R/W
b16	DSELCA	ELCA Event Source Count-Down Enable	0: Disables count-down by the ELCA event input 1: Enables count-down by the ELCA event input	R/W
b17	DSELCB	ELCB Event Source Count-Down Enable	0: Disables count-down by the ELCB event input 1: Enables count-down by the ELCB event input	R/W
b18	DSELCC	ELCC Event Source Count-Down Enable	0: Disables count-down by the ELCC event input 1: Enables count-down by the ELCC event input	R/W
b19	DSELCD	ELCD Event Source Count-Down Enable	0: Disables count-down by the ELCD event input 1: Enables count-down by the ELCD event input	R/W
b20	DSELCE	ELCE Event Source Count-Down Enable	0: Disables count-down by the ELCE event input 1: Enables count-down by the ELCE event input	R/W
b21	DSELCF	ELCF Event Source Count-Down Enable	0: Disables count-down by the ELCF event input 1: Enables count-down by the ELCF event input	R/W
b22	DSELCG	ELCG Event Source Count-Down Enable	0: Disables count-down by the ELCG event input 1: Enables count-down by the ELCG event input	R/W
b23	DSELCH	ELCH Event Source Count-Down Enable	0: Disables count-down by the ELCH event input 1: Enables count-down by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	DSILVL[3:0]	External Input Level Source Count-Down Enable	b27 b24 0 0 0 0: Disable count-down by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-down by GTIOCnA pin input level Low. 0 0 1 1: Enable count-down by GTIOCnA pin input level High. 0 1 0 0: Enable count-down by GTIOCnB pin input level Low. 0 1 0 1: Enable count-down by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-down by GTETRGA pin input level Low. 1 0 0 1: Enable count-down by GTETRGA pin input level High. 1 0 1 0: Enable count-down by GTETRGB pin input level Low. 1 0 1 1: Enable count-down by GTETRGB pin input level High. 1 1 0 0: Enable count-down by GTETRGC pin input level Low. 1 1 0 1: Enable count-down by GTETRGC pin input level High. 1 1 1 0: Enable count-down by GTETRGD pin input level Low. 1 1 1 1: Enable count-down by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 15

The GTDNSR register sets a count-down source for the GTCNT counter.

The GTDNSR register is primarily used in 24.3.12 Phase Counting Mode and 24.3.13 External Pulse Width Measuring Function.

When at least one bit among bits in the GTDNSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-down by a source set as 1 by the GTDNSR register is executed.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR)

Address(es): GPTW0.GTICASR 8742 C024h, GPTW1.GTICASR 8742 C124h, GPTW2.GTICASR 8742 C224h, GPTW3.GTICASR 8742 C324h, GPTW4.GTICASR 8742 C424h, GPTW5.GTICASR 8742 C524h, GPTW6.GTICASR 8742 C624h, GPTW7.GTICASR 8742 C724h, GPTW8.GTICASR 8742 C824h, GPTW9.GTICASR 8742 C924h, GPTW10.GTICASR 8742 CA24h, GPTW11.GTICASR 8742 CB24h, GPTW12.GTICASR 8742 CC24h, GPTW13.GTICASR 8742 CD24h, GPTW14.GTICASR 8742 CE24h, GPTW15.GTICASR 8742 CF24h, GPTW0_NS.GTICASR 0742 C024h, GPTW1_NS.GTICASR 0742 C124h, GPTW2_NS.GTICASR 0742 C224h, GPTW3_NS.GTICASR 0742 C324h, GPTW4_NS.GTICASR 0742 C424h, GPTW5_NS.GTICASR 0742 C524h, GPTW6_NS.GTICASR 0742 C624h, GPTW7_NS.GTICASR 0742 C724h, GPTW8_NS.GTICASR 0742 C824h, GPTW9_NS.GTICASR 0742 C924h, GPTW10_NS.GTICASR 0742 CA24h, GPTW11_NS.GTICASR 0742 CB24h, GPTW12_NS.GTICASR 0742 CC24h, GPTW13_NS.GTICASR 0742 CD24h, GPTW14_NS.GTICASR 0742 CE24h, GPTW15_NS.GTICASR 0742 CF24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ASGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGA signal.	R/W
b1	ASGTRGAF			R/W
b2	ASGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGB signal.	R/W
b3	ASGTRGBF			R/W
b4	ASGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGC signal.	R/W
b5	ASGTRGCF			R/W
b6	ASGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGD signal.	R/W
b7	ASGTRGDF			R/W

Bit	Symbol	Bit Name	Description	R/W
b8	ASCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal.	R/W
b9	ASCARBH			R/W
b10	ASCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal.	R/W
b11	ASCAFBH			R/W
b12	ASCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal.	R/W
b13	ASCBRAH			R/W
b14	ASCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal.	R/W
b15	ASCBFAH			R/W
b16	ASELCA	ELCA Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCA event input 1: Enables GTCCRA input capture by the ELCA event input	R/W
b17	ASELCB	ELCB Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCB event input 1: Enables GTCCRA input capture by the ELCB event input	R/W
b18	ASELCC	ELCC Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCC event input 1: Enables GTCCRA input capture by the ELCC event input	R/W
b19	ASELCD	ELCD Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCD event input 1: Enables GTCCRA input capture by the ELCD event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	ASELCE	ELCE Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCE event input 1: Enables GTCCRA input capture by the ELCE event input	R/W
b21	ASELCF	ELCF Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCF event input 1: Enables GTCCRA input capture by the ELCF event input	R/W
b22	ASELCG	ELCG Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCG event input 1: Enables GTCCRA input capture by the ELCG event input	R/W
b23	ASELCH	ELCH Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCH event input 1: Enables GTCCRA input capture by the ELCH event input	R/W
b24	ASOC	Other Channel Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the other channel factor 1: Enables GTCCRA input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 15

The GTICASR register sets an input capture source for the GTCCRA register.
When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

ASOC Bit (Other Channel Source GTCCRA Input Capture Enable)

This bit enables or disables triggering of input capture of a timer value by the GTCCRA register through interlinked channel operation with the trigger source specified in the GTICCR register in another channel. Input capture triggered by a source in another channel enabled by this bit cannot be used as a source to trigger input capture in another channel through specification by the ICAFA and ICBFA bits in the GTICCR register in its own channel.

24.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR)

Address(es): GPTW0.GTICBSR 8742 C028h, GPTW1.GTICBSR 8742 C128h, GPTW2.GTICBSR 8742 C228h, GPTW3.GTICBSR 8742 C328h, GPTW4.GTICBSR 8742 C428h, GPTW5.GTICBSR 8742 C528h, GPTW6.GTICBSR 8742 C628h, GPTW7.GTICBSR 8742 C728h, GPTW8.GTICBSR 8742 C828h, GPTW9.GTICBSR 8742 C928h, GPTW10.GTICBSR 8742 CA28h, GPTW11.GTICBSR 8742 CB28h, GPTW12.GTICBSR 8742 CC28h, GPTW13.GTICBSR 8742 CD28h, GPTW14.GTICBSR 8742 CE28h, GPTW15.GTICBSR 8742 CF28h, GPTW0_NS.GTICBSR 0742 C028h, GPTW1_NS.GTICBSR 0742 C128h, GPTW2_NS.GTICBSR 0742 C228h, GPTW3_NS.GTICBSR 0742 C328h, GPTW4_NS.GTICBSR 0742 C428h, GPTW5_NS.GTICBSR 0742 C528h, GPTW6_NS.GTICBSR 0742 C628h, GPTW7_NS.GTICBSR 0742 C728h, GPTW8_NS.GTICBSR 0742 C828h, GPTW9_NS.GTICBSR 0742 C928h, GPTW10_NS.GTICBSR 0742 CA28h, GPTW11_NS.GTICBSR 0742 CB28h, GPTW12_NS.GTICBSR 0742 CC28h, GPTW13_NS.GTICBSR 0742 CD28h, GPTW14_NS.GTICBSR 0742 CE28h, GPTW15_NS.GTICBSR 0742 CF28h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGA signal.	R/W
b1	BSGTRGAF			R/W
b2	BSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGB signal.	R/W
b3	BSGTRGBF			R/W
b4	BSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGC signal.	R/W
b5	BSGTRGCF			R/W
b6	BSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGD signal.	R/W
b7	BSGTRGDF			R/W

Bit	Symbol	Bit Name	Description	R/W
b8	BSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal.	R/W
b9	BSCARBH			R/W
b10	BSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal.	R/W
b11	BSCAFBH			R/W
b12	BSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal.	R/W
b13	BSCBRAH			R/W
b14	BSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal.	R/W
b15	BSCBFAH			R/W
b16	BSELCA	ELCA Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCA event input 1: Enables GTCCRB input capture by the ELCA event input	R/W
b17	BSELCB	ELCB Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCB event input 1: Enables GTCCRB input capture by the ELCB event input	R/W
b18	BSELCC	ELCC Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCC event input 1: Enables GTCCRB input capture by the ELCC event input	R/W
b19	BSELCD	ELCD Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCD event input 1: Enables GTCCRB input capture by the ELCD event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	BSELCE	ELCE Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCE event input 1: Enables GTCCRB input capture by the ELCE event input	R/W
b21	BSELCF	ELCF Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCF event input 1: Enables GTCCRB input capture by the ELCF event input	R/W
b22	BSELCG	ELCG Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCG event input 1: Enables GTCCRB input capture by the ELCG event input	R/W
b23	BSELCH	ELCH Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCH event input 1: Enables GTCCRB input capture by the ELCH event input	R/W
b24	BSOC	Other Channel Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the other channel factor 1: Enables GTCCRB input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 15

The GTICBSR register sets an input capture source for the GTCCRB register.
When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

BSOC Bit (Other Channel Source GTCCRB Input Capture Enable)

This bit enables or disables triggering of input capture of a timer value by the GTCCRB register through interlinked channel operation with the trigger source specified in the GTICCR register in another channel. Input capture triggered by a source in another channel enabled by this bit cannot be used as a source to trigger input capture in another channel through specification by the ICAFB and ICBFB bits in the GTICCR register in its own channel.

24.2.12 General PWM Timer Switch Source Select Register (GTSWSR)

Address(es): GPTW0.GTSWSR 8742 C0D8h, GPTW1.GTSWSR 8742 C1D8h, GPTW2.GTSWSR 8742 C2D8h, GPTW3.GTSWSR 8742 C3D8h, GPTW4.GTSWSR 8742 C4D8h, GPTW5.GTSWSR 8742 C5D8h, GPTW6.GTSWSR 8742 C6D8h, GPTW7.GTSWSR 8742 C7D8h, GPTW8.GTSWSR 8742 C8D8h, GPTW9.GTSWSR 8742 C9D8h, GPTW10.GTSWSR 8742 CAD8h, GPTW11.GTSWSR 8742 CBD8h, GPTW12.GTSWSR 8742 CCD8h, GPTW13.GTSWSR 8742 CDD8h, GPTW14.GTSWSR 8742 CED8h, GPTW15.GTSWSR 8742 CFD8h,
GPTW0_NS.GTSWSR 0742 C0D8h, GPTW1_NS.GTSWSR 0742 C1D8h, GPTW2_NS.GTSWSR 0742 C2D8h, GPTW3_NS.GTSWSR 0742 C3D8h, GPTW4_NS.GTSWSR 0742 C4D8h, GPTW5_NS.GTSWSR 0742 C5D8h, GPTW6_NS.GTSWSR 0742 C6D8h, GPTW7_NS.GTSWSR 0742 C7D8h, GPTW8_NS.GTSWSR 0742 C8D8h, GPTW9_NS.GTSWSR 0742 C9D8h, GPTW10_NS.GTSWSR 0742 CAD8h, GPTW11_NS.GTSWSR 0742 CBD8h, GPTW12_NS.GTSWSR 0742 CCD8h, GPTW13_NS.GTSWSR 0742 CDD8h, GPTW14_NS.GTSWSR 0742 CED8h, GPTW15_NS.GTSWSR 0742 CFD8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	WSELCH	WSELCG	WSELCF	WSELCE	WSELCD	WSELCC	WSELCB	WSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WSGTRGDF	WSGTRGDR	WSGTRGCF	WSGTRGCR	WSGTRGBF	WSGTRGBR	WSGTRGAF	WSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to switch the direction of counting. 0 1: The direction of counting switches on the rising edge of the GTETRGA signal. 1 0: The direction of counting switches on the falling edge of the GTETRGA signal. 1 1: The direction of counting switches on both edges of the GTETRGA signal.	R/W
b1	WSGTRGAF			R/W
b2	WSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to switch the direction of counting. 0 1: The direction of counting switches on the rising edge of the GTETRGB signal. 1 0: The direction of counting switches on the falling edge of the GTETRGB signal. 1 1: The direction of counting switches on both edges of the GTETRGB signal.	R/W
b3	WSGTRGBF			R/W
b4	WSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to switch the direction of counting. 0 1: The direction of counting switches on the rising edge of the GTETRGC signal. 1 0: The direction of counting switches on the falling edge of the GTETRGC signal. 1 1: The direction of counting switches on both edges of the GTETRGC signal.	R/W
b5	WSGTRGCF			R/W
b6	WSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to switch the direction of counting. 0 1: The direction of counting switches on the rising edge of the GTETRGD signal. 1 0: The direction of counting switches on the falling edge of the GTETRGD signal. 1 1: The direction of counting switches on both edges of the GTETRGD signal.	R/W
b7	WSGTRGDF			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	WSELCA	ELCA Event Source Count Switch Enable	0: Disables count switch by the ELCA event input 1: Enables count switch by the ELCA event input	R/W
b17	WSELCB	ELCB Event Source Count Switch Enable	0: Disables count switch by the ELCB event input 1: Enables count switch by the ELCB event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b18	WSELCC	ELCC Event Source Count Switch Enable	0: Disables count switch by the ELCC event input 1: Enables count switch by the ELCC event input	R/W
b19	WSELCD	ELCD Event Source Count Switch Enable	0: Disables count switch by the ELCD event input 1: Enables count switch by the ELCD event input	R/W
b20	WSELCE	ELCE Event Source Count Switch Enable	0: Disables count switch by the ELCE event input 1: Enables count switch by the ELCE event input	R/W
b21	WSELCF	ELCF Event Source Count Switch Enable	0: Disables count switch by the ELCF event input 1: Enables count switch by the ELCF event input	R/W
b22	WSELCG	ELCG Event Source Count Switch Enable	0: Disables count switch by the ELCG event input 1: Enables count switch by the ELCG event input	R/W
b23	WSELCH	ELCH Event Source Count Switch Enable	0: Disables count switch by the ELCH event input 1: Enables count switch by the ELCH event input	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSWSR register sets a count switch source for the GTCNT counter.
Count switch can be executed whether the counter is running (the GTCR.CST bit = 1) or stopped (the GTCR.CST bit = 0).
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTWn via the POEG. Set the polarity of these signals with the POEG.

24.2.13 General PWM Timer Switch Offset Setting Register (GTSWOS)

Address(es): GPTW0.GTSWOS 8742 C0DCh, GPTW1.GTSWOS 8742 C1DCh, GPTW2.GTSWOS 8742 C2DCh, GPTW3.GTSWOS 8742 C3DCh, GPTW4.GTSWOS 8742 C4DCh, GPTW5.GTSWOS 8742 C5DCh, GPTW6.GTSWOS 8742 C6DCh, GPTW7.GTSWOS 8742 C7DCh, GPTW8.GTSWOS 8742 C8DCh, GPTW9.GTSWOS 8742 C9DCh, GPTW10.GTSWOS 8742 CADCh, GPTW11.GTSWOS 8742 CBDCh, GPTW12.GTSWOS 8742 CCDCh, GPTW13.GTSWOS 8742 CDDCh, GPTW14.GTSWOS 8742 CEDCh, GPTW15.GTSWOS 8742 CFDCh,
GPTW0_NS.GTSWOS 0742 C0DCh, GPTW1_NS.GTSWOS 0742 C1DCh, GPTW2_NS.GTSWOS 0742 C2DCh, GPTW3_NS.GTSWOS 0742 C3DCh, GPTW4_NS.GTSWOS 0742 C4DCh, GPTW5_NS.GTSWOS 0742 C5DCh, GPTW6_NS.GTSWOS 0742 C6DCh, GPTW7_NS.GTSWOS 0742 C7DCh, GPTW8_NS.GTSWOS 0742 C8DCh, GPTW9_NS.GTSWOS 0742 C9DCh, GPTW10_NS.GTSWOS 0742 CADCh, GPTW11_NS.GTSWOS 0742 CBDCh, GPTW12_NS.GTSWOS 0742 CCDCh, GPTW13_NS.GTSWOS 0742 CDDCh, GPTW14_NS.GTSWOS 0742 CEDCh, GPTW15_NS.GTSWOS 0742 CFDCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTSWOS register is a readable/writable register that sets the offset value of GTCNT counter which is reloaded by count switch operation.

Access in 8-bit or 16-bit units to the GTSWOS register is prohibited, and it should be accessed in 32-bit units.

The GTSWOS register is writable even while the GTCNT counter operation. If the GTSWOS register writing and the event input of count switch occur at the same time, the GTSWOS register setting before writing is reloaded to GTCNT counter. Write the GTSWOS register earlier than the event input of count switch to ensure that the written GTSWOS register value is reloaded by the count switch operation.

24.2.14 General PWM Timer Control Register (GTCR)

- GPTW0.GTCR, GPTW1.GTCR, GPTW2.GTCR, GPTW3.GTCR, GPTW10.GTCR, GPTW11.GTCR, GPTW12.GTCR, GPTW13.GTCR, GPTW14.GTCR, GPTW15.GTCR

Address(es): GPTW0.GTCR 8742 C02Ch, GPTW1.GTCR 8742 C12Ch, GPTW2.GTCR 8742 C22Ch, GPTW3.GTCR 8742 C32Ch, GPTW10.GTCR 8742 CA2Ch, GPTW11.GTCR 8742 CB2Ch, GPTW12.GTCR 8742 CC2Ch, GPTW13.GTCR 8742 CD2Ch, GPTW14.GTCR 8742 CE2Ch, GPTW15.GTCR 8742 CF2Ch, GPTW0_NS.GTCR 0742 C02Ch, GPTW1_NS.GTCR 0742 C12Ch, GPTW2_NS.GTCR 0742 C22Ch, GPTW3_NS.GTCR 0742 C32Ch, GPTW10_NS.GTCR 0742 CA2Ch, GPTW11_NS.GTCR 0742 CB2Ch, GPTW12_NS.GTCR 0742 CC2Ch, GPTW13_NS.GTCR 0742 CD2Ch, GPTW14_NS.GTCR 0742 CE2Ch, GPTW15_NS.GTCR 0742 CF2Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SWMD[2:0]			CKEG[1:0]		TPCS[3:0]				—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	BINV	AINV	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	AINV	GTIOCnA Input/Output Pin Polarity Reversal Control	0: Not inverted 1: Inverted	R/W
b5	BINV	GTIOCnB Input/Output Pin Polarity Reversal Control	0: Not inverted 1: Inverted	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC Input Source Synchronous Clear Enable	0: Disables use of the counter clear by GTIOC input as the clear factor for other channels 1: Enables use of the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Sawtooth-wave PWM mode (single buffer or double buffer possible) 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
b22 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: GTCLK 0 0 0 1: GTCLK/2 0 0 1 0: GTCLK/4 0 0 1 1: GTCLK/8 0 1 0 0: GTCLK/16 0 1 0 1: GTCLK/32 0 1 1 0: GTCLK/64 0 1 1 1: GTCLK/128 1 0 0 0: GTCLK/256 1 0 0 1: GTCLK/512 1 0 1 0: GTCLK/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W
b31 to b29	SWMD[2:0]	Switch Mode Select	b31 b29 0 0 0: Hardware switch operation is disabled. 0 0 1: Switch to up-count with offset reload. 0 1 0: Switch to down-count with offset reload. 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Switch to up-count without offset reload. 1 1 0: Switch to down-count without offset reload. 1 1 1: Setting prohibited	R/W

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 0 to 3, 10 to 15), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- A cycle-counting operation is completed while the GTPC.ASTP bit is 1
- 0 is written by software

AINV Bit (GTIOCnA Input/Output Pin Polarity Reversal Control) (n = 0 to 3, 10 to 15)

This bit selects GTIOCnA input/output pin polarity reversal control.

BINV Bit (GTIOCnB Input/Output Pin Polarity Reversal Control) (n = 0 to 3, 10 to 15)

This bit selects GTIOCnB input/output pin polarity reversal control.

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC Input Source Synchronous Clear Enable)

This bit enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit enables or disables synchronous set/clear.
In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[2:0] Bits (Mode Select)

These bits select the GPTW operating mode.
Only the MD[2] bit is valid at input capture.
Counting in sawtooth-wave mode is performed with 0 for the MD[2] bit, counting in triangle-wave mode is performed with 1 for the MD[2] bit.
Set the MD[2:0] bits while the GTCNT counter operation is stopped.
During the event count operation (at least one of the bits in the GTUPSR or GTDNSR register is 1), set the MD[2:0] bits to the initial value (000b).

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by the TPCS[3:0] bits, the CKEG[1:0] bits select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

SWMD[2:0] Bits (Switch Mode Select)

These bits select the switch mode.
Set the SWMD[2:0] bits while the GTCNT counter operation is stopped.
The setting is invalid except in triangle-wave PWM mode.

- GPTW4.GTCR, GPTW5.GTCR, GPTW6.GTCR, GPTW7.GTCR, GPTW8.GTCR, GPTW9.GTCR

Address(es): GPTW4.GTCR 8742 C42Ch, GPTW5.GTCR 8742 C52Ch, GPTW6.GTCR 8742 C62Ch, GPTW7.GTCR 8742 C72Ch, GPTW8.GTCR 8742 C82Ch, GPTW9.GTCR 8742 C92Ch, GPTW4_NS.GTCR 0742 C42Ch, GPTW5_NS.GTCR 0742 C52Ch, GPTW6_NS.GTCR 0742 C62Ch, GPTW7_NS.GTCR 0742 C72Ch, GPTW8_NS.GTCR 0742 C82Ch, GPTW9_NS.GTCR 0742 C92Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SWMD[2:0]			CKEG[1:0]		TPCS[3:0]				—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	BINV	AINV	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	AINV	GTIOCnA Input/Output Pin Polarity Reversal Control	0: Not inverted 1: Inverted	R/W
b5	BINV	GTIOCnB Input/Output Pin Polarity Reversal Control	0: Not inverted 1: Inverted	R/W
b7 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC Input Source Synchronous Clear Enable	0: Disables use of the counter clear by GTIOC input as the clear factor for other channels 1: Enables use of the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W *1
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b19 to b16	MD[3:0]	Mode Select	b19 b16 0 0 0 0: Sawtooth-wave PWM mode 1 (single buffer or double buffer possible) 0 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 0 1 0: Sawtooth-wave PWM mode 2 (single buffer or double buffer possible) 0 0 1 1: Setting prohibited 0 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0 1 1 1: Setting prohibited 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Setting prohibited 1 0 1 1: Setting prohibited 1 1 0 0: Complementary PWM mode 1 (transfer at crest) 1 1 0 1: Complementary PWM mode 2 (transfer at trough) 1 1 1 0: Complementary PWM mode 3 (transfer at crest and trough) 1 1 1 1: Complementary PWM mode 4 (immediate transfer)	R/W *1
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: GTCLK 0 0 0 1: GTCLK/2 0 0 1 0: GTCLK/4 0 0 1 1: GTCLK/8 0 1 0 0: GTCLK/16 0 1 0 1: GTCLK/32 0 1 1 0: GTCLK/64 0 1 1 1: GTCLK/128 1 0 0 0: GTCLK/256 1 0 0 1: GTCLK/512 1 0 1 0: GTCLK/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W *1
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W *1
b31 to b29	SWMD[2:0]	Switch Mode Select	b31 b29 0 0 0: Hardware switch operation is disabled. 0 0 1: Switch to up-count with offset reload. 0 1 0: Switch to down-count with offset reload. 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Switch to up-count without offset reload. 1 1 0: Switch to down-count without offset reload. 1 1 1: Setting prohibited	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 4 to 9), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- A cycle-counting operation is completed while the GTPC.ASTP bit is 1
- 0 is written by software

AINV Bit (GTIOCnA Input/Output Pin Polarity Reversal Control) (n = 4 to 9)

This bit selects GTIOCnA input/output pin polarity reversal control.

BINV Bit (GTIOCnB Input/Output Pin Polarity Reversal Control) (n = 4 to 9)

This bit selects GTIOCnB input/output pin polarity reversal control.

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC Input Source Synchronous Clear Enable)

This bit enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.
In complementary PWM mode, slave channels are also controlled by setting the SSCGRP[1:0] bits of the master channel.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit enables or disables synchronous set/clear.
In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] Bits (Mode Select)

These bits select the GPTW operating mode.
In complementary PWM mode, the master channel setting of the MD[3:0] bits is also used to control the slave channels.
Only the MD[3:2] bits are valid at input capture.
Counting in sawtooth-wave mode is performed with 00b for the MD[3:2] bits, counting in triangle-wave mode is performed with 01b for the MD[3:2] bits.
Set the MD[3:0] bits while the GTCNT counter operation is stopped.
During the event count operation (at least one of the bits in the GTUPSR or GTDNSR register is 1), set the MD[3:0] bits to the initial value (0000b).

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by the TPCS[3:0] bits, the CKEG[1:0] bits select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

SWMD[2:0] Bits (Switch Mode Select)

These bits select the switch mode.
Set the SWMD[2:0] bits while the GTCNT counter operation is stopped.
The setting is invalid except in triangle-wave PWM mode.

24.2.15 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPTW0.GTUDDTYC 8742 C030h, GPTW1.GTUDDTYC 8742 C130h, GPTW2.GTUDDTYC 8742 C230h, GPTW3.GTUDDTYC 8742 C330h, GPTW4.GTUDDTYC 8742 C430h, GPTW5.GTUDDTYC 8742 C530h, GPTW6.GTUDDTYC 8742 C630h, GPTW7.GTUDDTYC 8742 C730h, GPTW8.GTUDDTYC 8742 C830h, GPTW9.GTUDDTYC 8742 C930h, GPTW10.GTUDDTYC 8742 CA30h, GPTW11.GTUDDTYC 8742 CB30h, GPTW12.GTUDDTYC 8742 CC30h, GPTW13.GTUDDTYC 8742 CD30h, GPTW14.GTUDDTYC 8742 CE30h, GPTW15.GTUDDTYC 8742 CF30h,
GPTW0_NS.GTUDDTYC 0742 C030h, GPTW1_NS.GTUDDTYC 0742 C130h, GPTW2_NS.GTUDDTYC 0742 C230h, GPTW3_NS.GTUDDTYC 0742 C330h, GPTW4_NS.GTUDDTYC 0742 C430h, GPTW5_NS.GTUDDTYC 0742 C530h, GPTW6_NS.GTUDDTYC 0742 C630h, GPTW7_NS.GTUDDTYC 0742 C730h, GPTW8_NS.GTUDDTYC 0742 C830h, GPTW9_NS.GTUDDTYC 0742 C930h, GPTW10_NS.GTUDDTYC 0742 CA30h, GPTW11_NS.GTUDDTYC 0742 CB30h, GPTW12_NS.GTUDDTYC 0742 CC30h, GPTW13_NS.GTUDDTYC 0742 CD30h, GPTW14_NS.GTUDDTYC 0742 CE30h, GPTW15_NS.GTUDDTYC 0742 CF30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	OABDTYT	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down. 1: GTCNT counts up.	R/W
b1	UDF	Forcible Count Direction Setting	0: Count direction is not forcibly set. 1: Count direction is forcibly set.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	OADTY[1:0]	GTIOCnA Pin Output Duty Setting	b17b16 0 x: Compare matches determine the duty cycle of the output on the GTIOCnA pin. 1 0: The duty cycle of the output on the GTIOCnA pin is 0%. 1 1: The duty cycle of the output on the GTIOCnA pin is 100%.	R/W
b18	OADTYF	GTIOCnA Pin Output Duty Forced Setting	0: Duty of the GTIOCnA pin output is not forcibly set. 1: Duty of the GTIOCnA pin output is forcibly set.	R/W
b19	OADTYR	Output after Release of GTIOCnA Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCnB Pin Output Duty Setting	b25b24 0 x: Compare matches determine the duty cycle of the output on the GTIOCnB pin 1 0: The duty cycle of the output on the GTIOCnB pin is 0%. 1 1: The duty cycle of the output on the GTIOCnB pin is 100%.	R/W
b26	OBDTYF	GTIOCnB Pin Output Duty Forced Setting	0: Duty of the GTIOCnB pin output is not forcibly set. 1: Duty of the GTIOCnB pin output is forcibly set.	R/W
b27	OBDTYR	Output after Release of GTIOCnB Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b28	OABDTYT	GTIOCnm Pin Output 0%/ 100% Duty Cycle Setting Reflection Timing	0: A new output duty-cycle setting is reflected in a trough 1: A new output duty-cycle setting is reflected in a trough or crest The setting “1” is prohibited except in the triangle-wave mode.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 15; m = A, B

The GTUDDTYC register sets the count direction (up-counting/down-counting) for the GTCNT counter and configures duty setting for output from the GTIOCnA and GTIOCnB pins.
The setting is invalid during the event count operation, sawtooth-wave PWM mode 2, and complementary PWM mode.

(1) Setting of Count Direction

- In sawtooth-wave mode

When the UD bit value is set to 0 during up-counting, the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value). When the UD bit value is set to 1 during down-counting, the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
If the UD bit value is changed from 1 to 0 with the UDF bit being 0 and while counting is stopped, the counter starts up-counting and the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value).
If the UD bit value is changed from 0 to 1 with the UDF bit being 0 and while counting is stopped, the counter starts down-counting and the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
When the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

- In triangle-wave mode

Even if the UD bit value is changed during counting, the change will not be reflected in the count direction.
If the UD bit value is modified while the UDF bit is 0 and counting is stopped, the change will not be reflected in the count direction when counting starts. If the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

UD Bit (Count Direction Setting)

This bit sets the count direction (up-counting or down-counting) for the GTCNT counter.

UDF Bit (Forcible Count Direction Setting)

This bit forcibly sets the count direction when the GTCNT counter starts operation as the UD bit value.
Only 0 should be written to this bit during count operation.
When 1 has been written to this bit while counting is stopped, this bit should be returned to 0 before counting starts.

(2) Setting of Output Duty

- In sawtooth-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during the up-count operation, the updated duty-cycle setting applied at an overflow.
When the OADTY[1:0]/OBDTY[1:0] bits are changed during the down-count operation, the updated duty-cycle setting applied at an underflow.
If the OADTYF/OBDTYF bit is 0 and the count operation is stopped when the OADTY[1:0]/OBDTY[1:0] bits are changed, the updated duty-cycle setting is not applied in the first count operation, but it is applied at an overflow when counting up and at an underflow when counting down.
If the OADTYF/OBDTYF bit is 1 and the count operation is stopped when the OADTY[1:0]/OBDTY[1:0] bits are changed, the updated duty-cycle setting is immediately applied in the first count operation.

- In triangle-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during the count operation, the updated duty-cycle setting is applied at an underflow.
If the OADTYF/OBDTYF bit is 0 and the count operation is stopped when the OADTY[1:0]/OBDTY[1:0] bits are changed, the updated duty-cycle setting is not applied in the first count operation, but it is applied at an underflow.

If the OADTYF/OBDTYF bit is 1 and the count operation is stopped when the OADTY[1:0]/OBDTY[1:0] bits are changed, the updated duty-cycle setting is immediately applied in the first count operation.

In both sawtooth-wave mode and triangle-wave mode, if the duty of the first cycle after the count starts is configured using the OADTY[1:0]/OBDTY[1:0] bits while the OADTYF/OBDTYF bit is 1 and the count operation is stopped, then even if the OADTYF/OBDTYF bit is changed to 0 and the OADTY[1:0]/OBDTY[1:0] bits are reconfigured, after the count starts the duty for the first and second cycles will remain as the values set during the count stop.

OmDTY[1:0] Bits (GTIOCnm Pin Output Duty Setting) (n = 0 to 15; m = A, B)

These bits set the output of the duty cycle (0%, 100% or compare match control) on the GTIOCnm pin.

OmDTYF Bit (GTIOCnm Pin Output Duty Forced Setting) (n = 0 to 15; m = A, B)

This bit forcibly specifies the duty cycle at the start of the GTCNT counter in the OmDTY[1:0] bits.
Always write 0 to the bit during count operation.
When writing 1 to the bit while the count operation is stopped, reset the bit to 0 by the end of the first cycle after the count operation starts and specify the next cycle setting.

OmDTYR Bit (Output after Release of GTIOCnm Pin Output 0%/100% Duty Cycle Settings) (n = 0 to 15; m = A, B)

When the GTIOCnm pin setting is changed from 0% or 100% duty cycle to compare match and the GTIOR.GTIOm[3:2] bits are set to 00b (output maintained at the end of the cycle) or 11b (output toggled), this bit specifies the value to maintain or toggle.
The GPTW internally continues to perform compare match operation during duty-cycle 0% or 100% operation. If the OmDTYR bit is 1, the value at the end of a cycle of compare match operation is applied to the GTIOm[3:2] bits setting.

OABDTYT Bit (GTIOCnm Pin Output 0%/100% Duty Cycle Setting Reflection Timing) (n = 0 to 15; m = A, B)

Select the timing of reflection of the 0% or 100% duty-cycle setting for the GTIOCnm pin in the triangle-wave mode. When the setting of the OABDTYT bit is 0, a new output duty-cycle setting in the GTUDDTYC.OADTY[1:0] or OBDTY[1:0] bits is reflected in response to an underflow. When the setting of the OABDTYT bit is 1, a new output duty-cycle setting in the GTUDDTYC.OADTY[1:0] or OBDTY[1:0] bits is reflected in response to an underflow or at the peak of the triangle wave.

24.2.16 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPTW0.GTIOR 8742 C034h, GPTW1.GTIOR 8742 C134h, GPTW2.GTIOR 8742 C234h, GPTW3.GTIOR 8742 C334h, GPTW4.GTIOR 8742 C434h, GPTW5.GTIOR 8742 C534h, GPTW6.GTIOR 8742 C634h, GPTW7.GTIOR 8742 C734h, GPTW8.GTIOR 8742 C834h, GPTW9.GTIOR 8742 C934h, GPTW10.GTIOR 8742 CA34h, GPTW11.GTIOR 8742 CB34h, GPTW12.GTIOR 8742 CC34h, GPTW13.GTIOR 8742 CD34h, GPTW14.GTIOR 8742 CE34h, GPTW15.GTIOR 8742 CF34h, GPTW0_NS.GTIOR 0742 C034h, GPTW1_NS.GTIOR 0742 C134h, GPTW2_NS.GTIOR 0742 C234h, GPTW3_NS.GTIOR 0742 C334h, GPTW4_NS.GTIOR 0742 C434h, GPTW5_NS.GTIOR 0742 C534h, GPTW6_NS.GTIOR 0742 C634h, GPTW7_NS.GTIOR 0742 C734h, GPTW8_NS.GTIOR 0742 C834h, GPTW9_NS.GTIOR 0742 C934h, GPTW10_NS.GTIOR 0742 CA34h, GPTW11_NS.GTIOR 0742 CB34h, GPTW12_NS.GTIOR 0742 CC34h, GPTW13_NS.GTIOR 0742 CD34h, GPTW14_NS.GTIOR 0742 CE34h, GPTW15_NS.GTIOR 0742 CF34h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	GTIOA[4:0]	GTIOCnA Pin Function Select	See Table 24.4 and Table 24.5.	R/W
b5	CPSCIR	Complementary PWM Mode Initial Output at Synchronous Clear Disable*1	0: Output the initial value set by the GTIOR.GTIOA[4:0] and GTIOB[4:0] bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disable output the initial value	R/W
b6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop	0: The GTIOCnA pin outputs low when counting is stopped. 1: The GTIOCnA pin outputs high when counting is stopped.	R/W
b7	OAHLD	GTIOCnA Pin Output Retention at the Start/Stop Count	0: The GTIOCnA pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnA pin output level is retained at start/stop of counting.	R/W
b8	OAE	GTIOCnA Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W
b10, b9	OADF[1:0]	GTIOCnA Pin Negate Value Setting	b10 b9 0 0: None of the following sources is specified 0 1: GTIOCnA pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnA pin is set to 0 in response to control for output negation. 1 1: GTIOCnA pin is set to 1 in response to control for output negation.	R/W
b11	OAEOCD	GTCCRA Register Compare Match Cycle End Output Invalidate*1	0: Validate GTIOA[3:0] setting 1: Invalidate GTIOA[3:0] setting (GTIOCnA pin output is retained)	R/W
b12	PSYE	PWM Synchronous Output Enable	0: Disable GTCPPOn pins output 1: Enable GTCPPOn pins output	R/W
b13	NFAEN	GTIOCnA Pin Input Noise Filter Enable	0: The noise filter for GTIOCnA pin input is disabled 1: The noise filter for GTIOCnA pin input is enabled	R/W
b15, b14	NFCSA[1:0]	GTIOCnA Pin Input Noise Filter Sampling Clock Select	b15b14 0 0: GTCLK 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W
b20 to b16	GTIOB[4:0]	GTIOCnB Pin Function Select	See Table 24.4.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop	0: The GTIOCnB pin outputs low when counting is stopped. 1: The GTIOCnB pin outputs high when counting is stopped.	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	OBHLD	GTIOCnB Pin Output Retention at the Start/Stop Count	0: The GTIOCnB pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnB pin output level is retained at start/stop of counting.	R/W
b24	OBE	GTIOCnB Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W
b26, b25	OBDF[1:0]	GTIOCnB Pin Negate Value Setting	b26b25 0 0: None of the following sources is specified 0 1: GTIOCnB pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnB pin is set to 0 in response to control for output negation. 1 1: GTIOCnB pin is set to 1 in response to control for output negation.	R/W
b27	OBEOCD	GTCCRB Register Compare Match Cycle End Output Invalidate*1	• When sawtooth-wave PWM mode 1 0: Validate GTIOB[3:2] setting 1: Invalidate GTIOB[3:2] setting (GTIOCnB pin output is retained) • When sawtooth-wave PWM mode 2 0: Validate GTIOA[3:2] setting 1: Invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
b28	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29	NFBEN	GTIOCnB Pin Input Noise Filter Enable	0: The noise filter for GTIOCnB pin input is disabled 1: The noise filter for GTIOCnB pin input is enabled	R/W
b31, b30	NFCSB[1:0]	GTIOCnB Pin Input Noise Filter Sampling Clock Select	b31b30 0 0: GTCLK 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

n = 0 to 15

Note 1. This bit in each of GPTW0 to GPTW3 and GPTW10 to GPTW15 is reserved, and is read as 0. When writing, write 0 to the bit.

The GTIOR register configures the functions of the GTIOCnA, GTIOCnB, and GTCPPOn pins.

GTIOA[4:0] Bits (GTIOCnA Pin Function Select) (n = 0 to 15)

These bits select the GTIOCnA pin function. For details, see Table 24.4.

CPSCIR Bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

This bit selects the output waveform to be generated when synchronous clear occurs in complementary PWM mode. The initial output is disabled by this function only when synchronous clear occurs in the trough in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4]/GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4]/GTIOB[4] bits is output even when the synchronous clear occurs in the trough immediately after the count starts.

OADFLT Bit (GTIOCnA Pin Output Value Setting at the Count Stop) (n = 0 to 15)

This bit sets whether the GTIOCnA pin outputs high or low when counting is stopped.

OAHLD Bit (GTIOCnA Pin Output Retention at the Start/Stop Count) (n = 0 to 15)

This bit specifies whether the GTIOCnA pin output level is retained or the level depends on the register setting when counting is started or stopped.

[When the OAHLD bit is set to 0]

- The value specified by the GTIOA[4] bit is output when counting starts.
- The value specified by the OADFLT bit is output when counting stops.
- If the OADFLT bit is modified while counting is stopped, it is immediately reflected in the output.

[When the OAHLD bit is set to 1]

- The output is retained when counting starts or stops.

OAE Bit (GTIOCnA Pin Output Enable) (n = 0 to 15)

This bit enables or disables output on the GTIOCnA pin.
When the GTCCRA register is used as an input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit setting.

OADF[1:0] Bits (GTIOCnA Pin Negate Value Setting) (n = 0 to 15)

These bits specify the value for output from the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD Bit (GTCCRA Register Compare Match Cycle End Output Invalidate)

This bit selects whether to disable or enable the GTIOA[3:2] bit setting when the cycle end coincides with the timing of the compare match of the GTCCRA register in sawtooth-wave PWM mode 1 and 2.
If this bit is 1 (setting not applied), the GTIOCnA pin maintains the output level when the cycle end and the GTCCRA compare match timing coincide.

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables output from GTCPPOn pins synchronized with the PWM cycle. The output signal toggles at crest/trough/GTCNT counter clear in complementary PWM mode or in triangle-wave mode, and at the end of the cycle in sawtooth-wave mode. The initial output of the GTCPPOn pins is low, and changes to high at count start.

NFAEN Bit (GTIOCnA Pin Input Noise Filter Enable) (n = 0 to 15)

This bit sets enable or disable of the noise filter function for the GTIOCnA pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSA[1:0] Bits (GTIOCnA Pin Input Noise Filter Sampling Clock Select) (n = 0 to 15)

These bits set a sampling clock for the noise filter to the GTIOCnA pin input. Set the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

GTIOB[4:0] Bits (GTIOCnB Pin Function Select) (n = 0 to 15)

These bits select the GTIOCnB pin function. For details, see Table 24.4.
In sawtooth-wave PWM mode 2, only the GTIOB[1:0] bits take effect, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT Bit (GTIOCnB Pin Output Value Setting at the Count Stop) (n = 0 to 15)

This bit sets whether the GTIOCnB pin outputs high or low when counting is stopped.

OBHLD Bit (GTIOCnB Pin Output Retention at the Start/Stop Count) (n = 0 to 15)

This bit specifies whether the GTIOCnB pin output level is retained or the level depends on the register setting when counting is started or stopped.
[When the OBHLD bit is set to 0]

- The value specified by the GTIOB[4] bit is output when counting starts.
- The value specified by the OBDFLT bit is output when counting stops.
- If the OBDFLT bit is modified while counting is stopped, it is immediately reflected in the output.

[When the OBHLD bit is set to 1]

- The output is retained when counting starts or stops.

OBE Bit (GTIOCnB Pin Output Enable) (n = 0 to 15)

This bit enables or disables output on the GTIOCnB pin.
When the GTCCRB register is used as an input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit setting.

OBDF[1:0] Bits (GTIOCnB Pin Negate Value Setting) (n = 0 to 15)

These bits specify the value for output from the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD Bit (GTCCRB Register Compare Match Cycle End Output Invalidate)

This bit selects whether to disable or enable the GTIOB[3:2] bit setting for sawtooth-wave PWM mode 1 or the GTIOA[3:2] bit setting for sawtooth-wave PWM mode 2 when the cycle end coincides with the timing of the compare match of the GTCCRB register.
If this bit is 1 (setting not applied), the GTIOCnB pin or GTIOCnA pin maintains the output level when the cycle end and the GTCCRB compare match timing coincide in sawtooth-wave PWM mode 1 or 2 respectively.

NFBEN Bit (GTIOCnB Pin Input Noise Filter Enable) (n = 0 to 15)

This bit sets enable or disable of the noise filter function for the GTIOCnB pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSB[1:0] Bits (GTIOCnB Pin Input Noise Filter Sampling Clock Select) (n = 0 to 15)

These bits set a sampling clock for the noise filter to the GTIOCnB pin input. Set the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

Table 24.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (1/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	0	0	0	Initial output is Low.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Table 24.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (2/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
1	0	0	0	0	Initial output is High.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note: In sawtooth-wave mode, “end of a cycle” refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0000 0000h in up-counting), an underflow (the value of the GTCNT counter changing from 0000 0000h to that of the GTPR register in down-counting), or counter clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0000 0000h to 0000 0001h)).

Note: When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB register compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in sawtooth-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note: During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), the b3 and b2 settings become invalid.

Note: In sawtooth-wave PWM mode 2, the GTIOB[4:2] bits are invalid. Since only the GTIOCnA pin is the output pin, set the GTIOA[4] bit for the initial output. Set the GTIOA[3:2] bits for output at the end of the cycle.

Table 24.5 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Complementary PWM mode)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
					Initial Output/Active Level	Compare Match Output (Up-counting)	Compare Match Output (Down-counting)
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note: In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.

Note: In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.

Note: In complementary PWM mode, setting the GTIOB[4:0] bits does not use compare match of the GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see section 24.3.3.7, Complementary PWM mode 1, 2, 3.

24.2.17 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPTW0.GTINTAD 8742 C038h, GPTW1.GTINTAD 8742 C138h, GPTW2.GTINTAD 8742 C238h, GPTW3.GTINTAD 8742 C338h, GPTW4.GTINTAD 8742 C438h, GPTW5.GTINTAD 8742 C538h, GPTW6.GTINTAD 8742 C638h, GPTW7.GTINTAD 8742 C738h, GPTW8.GTINTAD 8742 C838h, GPTW9.GTINTAD 8742 C938h, GPTW10.GTINTAD 8742 CA38h, GPTW11.GTINTAD 8742 CB38h, GPTW12.GTINTAD 8742 CC38h, GPTW13.GTINTAD 8742 CD38h, GPTW14.GTINTAD 8742 CE38h, GPTW15.GTINTAD 8742 CF38h, GPTW0_NS.GTINTAD 0742 C038h, GPTW1_NS.GTINTAD 0742 C138h, GPTW2_NS.GTINTAD 0742 C238h, GPTW3_NS.GTINTAD 0742 C338h, GPTW4_NS.GTINTAD 0742 C438h, GPTW5_NS.GTINTAD 0742 C538h, GPTW6_NS.GTINTAD 0742 C638h, GPTW7_NS.GTINTAD 0742 C738h, GPTW8_NS.GTINTAD 0742 C838h, GPTW9_NS.GTINTAD 0742 C938h, GPTW10_NS.GTINTAD 0742 CA38h, GPTW11_NS.GTINTAD 0742 CB38h, GPTW12_NS.GTINTAD 0742 CC38h, GPTW13_NS.GTINTAD 0742 CD38h, GPTW14_NS.GTINTAD 0742 CE38h, GPTW15_NS.GTINTAD 0742 CF38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	GTINTPC	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GTINTA	GTCCRA Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	GTINTB	GTCCRB Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b2	GTINTC	GTCCRC Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b3	GTINTD	GTCCRD Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b4	GTINTE	GTCCRE Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b5	GTINTF	GTCCRF Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b7, b6	GTINTPR[1:0]	GTPR Register Compare Match Interrupt Enable	b7 b6 0 0: Interrupt request is disabled. 0 1: In sawtooth-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests. 1 0: In sawtooth-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs. 1 1: In sawtooth-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs.	R/W
b8	SCFA	GTCCRA Register Compare Match/ Input Capture Source Synchronous Clear Enable	0: Disables the GTCCRA register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRA register comparison match/ input capture as a source for other channel clears.	R/W
b9	SCFB	GTCCRB Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRB register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRB register comparison match/ input capture as a source for other channel clears.	R/W
b10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRC register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRC register comparison match/ input capture as a source for other channel clears.	R/W

Bit	Symbol	Bit Name	Description	R/W
b11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRD register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRD register comparison match/ input capture as a source for other channel clears.	R/W
b12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRE register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRE register comparison match/ input capture as a source for other channel clears.	R/W
b13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRF register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRF register comparison match/ input capture as a source for other channel clears.	R/W
b14	SCFPO	Overflow Source Synchronous Clear Enable	0: Disables Overflow as a source for other channel clears. 1: Enables Overflow as a source for other channel clears.	R/W
b15	SCFPU	Underflow Source Synchronous Clear Enable	0: Disables Underflow as a source for other channel clears. 1: Enables Underflow as a source for other channel clears.	R/W
b16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Stop Group Select	b25b24 0 0: Group A is selected 0 1: Group B is selected 1 0: Group C is selected 1 1: Group D is selected	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	GRPDTE	Dead Time Error Output Stop Detection Enable	0: Dead time error output stop detection is disabled 1: Dead time error output stop detection is enabled	R/W
b29	GRPABH	Simultaneous High Output Stop Detection Enable	0: Simultaneous high output stop detection is disabled 1: Simultaneous high output stop detection is enabled	R/W
b30	GRPABL	Simultaneous Low Output Stop Detection Enable	0: Simultaneous low output stop detection is disabled 1: Simultaneous low output stop detection is enabled	R/W
b31	GTINTPC	Cycle Count End Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W

The GTINTAD register enables or disables interrupt request, A/D conversion start request, and output stop detection.

GTINTA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRA register compare match/input capture (GTCIA).

GTINTB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRB register compare match/input capture (GTCIB).

GTINTC Bit (GTCCRC Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRC register compare match (GTCIC).

GTINTD Bit (GTCCRD Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRD register compare match (GTCID).

GTINTE Bit (GTCCRE Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRE register compare match (GTCIE).

GTINTF Bit (GTCCRF Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRF register compare match (GTCIF).

GTINTPR[1:0] Bits (GTPR Register Compare Match Interrupt Enable)

These bits enable or disable interrupt requests by a GTPR register compare match (GTCNT counter overflow) and those by a GTCNT counter underflow (GTCIV/GTCIU).

SCFA Bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables the GTCCRA register compare match/input capture as a clearing factor for other channels.

SCFB Bit (GTCCRB Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRB register compare match/input capture as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFC Bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRC register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFD Bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRD register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFE Bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRE register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFF Bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRF register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFPO Bit (Overflow Source Synchronous Clear Enable)

This bit enables or disables overflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

SCFPU Bit (Underflow Source Synchronous Clear Enable)

This bit enables or disables underflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

ADTRAUEN Bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter up-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRADEN Bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter down-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRBUEN Bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter up-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRBDEN Bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter down-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

GRP[1:0] Bits (Output Stop Group Select)

Select the group to detect disabling of output from the GPTW to the POEG and to request of disabling of output from the POEG to the GPTW.
Each signal indicating detection of disabling in response to a dead-time error or simultaneous driving of outputs to the high or low level is output to the group selected in the GRP[1:0] bits while the respective output stop detection enable bits are 1.
Requests to disabling output from the POEG for the group selected in the GRP[1:0] bits can be monitored with the GTST.ODF flag.
Set the GRP[1:0] bits while both of the OAE and OBE bits in the GTIOR register are set to 0.

GRPDTE Bit (Dead Time Error Output Stop Detection Enable)

This bit enables or disables output stop detection due to a dead time error.
The dead time error output stop detection is not generated during the event count operation.

GRPABH Bit (Simultaneous High Output Stop Detection Enable)

This bit enables or disables output stop detection, which occurs when both GTIOCnA and GTIOCnB pins become high simultaneously.

GRPABL Bit (Simultaneous Low Output Stop Detection Enable)

This bit enables or disables output stop detection, which occurs when both GTIOCnA and GTIOCnB pins become low simultaneously.

GTINTPC Bit (Cycle Count End Interrupt Enable)

This bit enables or disables the request of the cycle count end interrupt (GTCEI).

24.2.18 General PWM Timer Status Register (GTST)

Address(es): GPTW0.GTST 8742 C03Ch, GPTW1.GTST 8742 C13Ch, GPTW2.GTST 8742 C23Ch, GPTW3.GTST 8742 C33Ch, GPTW4.GTST 8742 C43Ch, GPTW5.GTST 8742 C53Ch, GPTW6.GTST 8742 C63Ch, GPTW7.GTST 8742 C73Ch, GPTW8.GTST 8742 C83Ch, GPTW9.GTST 8742 C93Ch, GPTW10.GTST 8742 CA3Ch, GPTW11.GTST 8742 CB3Ch, GPTW12.GTST 8742 CC3Ch, GPTW13.GTST 8742 CD3Ch, GPTW14.GTST 8742 CE3Ch, GPTW15.GTST 8742 CF3Ch, GPTW0_NS.GTST 0742 C03Ch, GPTW1_NS.GTST 0742 C13Ch, GPTW2_NS.GTST 0742 C23Ch, GPTW3_NS.GTST 0742 C33Ch, GPTW4_NS.GTST 0742 C43Ch, GPTW5_NS.GTST 0742 C53Ch, GPTW6_NS.GTST 0742 C63Ch, GPTW7_NS.GTST 0742 C73Ch, GPTW8_NS.GTST 0742 C83Ch, GPTW9_NS.GTST 0742 C93Ch, GPTW10_NS.GTST 0742 CA3Ch, GPTW11_NS.GTST 0742 CB3Ch, GPTW12_NS.GTST 0742 CC3Ch, GPTW13_NS.GTST 0742 CD3Ch, GPTW14_NS.GTST 0742 CE3Ch, GPTW15_NS.GTST 0742 CF3Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PCF	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCFA	Input Capture/Compare Match Flag A	0: No input capture/compare match of GTCCRA is generated. 1: An input capture/compare match of GTCCRA is generated.	R/(W) *1
b1	TCFB	Input Capture/Compare Match Flag B	0: No input capture/compare match of GTCCRB is generated. 1: An input capture/compare match of GTCCRB is generated.	R/(W) *1
b2	TCFC	Compare Match Flag C	0: No compare match of GTCCRC is generated. 1: An compare match of GTCCRC is generated.	R/(W) *1
b3	TCFD	Compare Match Flag D	0: No compare match of GTCCRD is generated. 1: An compare match of GTCCRD is generated.	R/(W) *1
b4	TCFE	Compare Match Flag E	0: No compare match of GTCCRE is generated. 1: An compare match of GTCCRE is generated.	R/(W) *1
b5	TCFF	Compare Match Flag F	0: No compare match of GTCCRF is generated. 1: An compare match of GTCCRF is generated.	R/(W) *1
b6	TCFPO	Overflow Flag	0: No overflow (crest) occurred. 1: An overflow (crest) occurred.	R/(W) *1
b7	TCFPU	Underflow Flag	0: No underflow (trough) occurred. 1: An underflow (trough) occurred.	R/(W) *1
b10 to b8	ITCNT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Counter	Counter for counting the number of times a timer interrupt has been skipped.	R
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: The GTCNT counter counts downward. 1: The GTCNT counter counts upward.	R
b16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag	0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/(W) *1
b17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag	0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/(W) *1
b18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag	0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/(W) *1
b19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag	0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ODF	Output Stop Request Flag	0: No output stop request has occurred. 1: An output stop request has occurred.	R
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	DTEF	Dead Time Error Flag	0: No dead time error has occurred. 1: A dead time error has occurred.	R
b29	OABHF	Simultaneous High Output Flag	0: No simultaneous generation of 1 both for the GTIOCnA and GTIOCnB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCnA and GTIOCnB pins has occurred.	R
b30	OABLF	Simultaneous Low Output Flag	0: No simultaneous generation of 0 both for the GTIOCnA and GTIOCnB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCnA and GTIOCnB pins has occurred.	R
b31	PCF	Cycle Count End Flag	0: No cycle count end has occurred. 1: A cycle count end has occurred.	R/(W) *1

n = 0 to 15

Note 1. Only 0 can be written to clear the flag. When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

The GTST register indicates the status of the GPTW.

TCFA Flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of the GTCCRA register.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRA register by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB Flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of the GTCCRB register.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRB register by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC Flag (Compare Match Flag C)

The TCFC flag indicates the status for the compare match of the GTCCRC register.

When the GTCCRC register performs buffer operation, the GTCCRC register doesn't perform compare match.

[Setting conditions]

- GTCNT = GTCCRC

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)

- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRC register performs buffer operation).

TCFD Flag (Compare Match Flag D)

The TCFD flag indicates the status for the compare match of the GTCCRD register.
When the GTCCRD register performs buffer operation, the GTCCRD register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRD

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRD register performs buffer operation).

TCFE Flag (Compare Match Flag E)

The TCFE flag indicates the status for the compare match of the GTCCRE register.
When the GTCCRE register performs buffer operation, the GTCCRE register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRE

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (the GTCCRE register performs buffer operation).

TCFF Flag (Compare Match Flag F)

The TCFF flag indicates the status for the compare match of the GTCCRF register.
When the GTCCRF register performs buffer operation, the GTCCRF register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRF

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode).
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (the GTCCRF register performs buffer operation).

TCFPO Flag (Overflow Flag)

The TCFPO flag indicates whether an overflow or crest has occurred.
[Setting conditions]

- In sawtooth-wave mode, an overflow (the GTCNT counter changes from the GTPR register value to 0 in up-counting) has occurred.
- In triangle-wave mode, a crest (the GTCNT counter changes from the GTPR register value to GTPR – 1) has occurred.
- In counting by hardware sources, an overflow (the GTCNT counter changes from the GTPR register value to 0 in

up-counting) has occurred.
[Clearing condition]

- 0 is written to this flag.

TCFPU Flag (Underflow Flag)

The TCFPU flag indicates whether an underflow or trough has occurred.
[Setting conditions]

- In sawtooth-wave mode, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred.
- In triangle-wave mode, a trough (the GTCNT counter changes from 0 to 1) has occurred.
- In counting by hardware sources, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

ITCNT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Counter)

When the GTCIV/GTCIU interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GTCIV/GTCIU interrupt source is generated.
These bits are operated independently of the extended interrupt skipping by the GTEITC register.
[Clearing conditions]

- The GTCIV/GTCIU interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GTCIV/GTCIU interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped.

TUCF Flag (Count Direction Flag)

This flag indicates the count direction of the GTCNT counter.
This flag indicates 1 when in up-counting and indicates 0 when in down-counting during the event count operation.

ADTRAUF Flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting

[Clearing conditions]

- 0 is written to the ADTRAUF flag

ADTRADF Flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting

[Clearing conditions]

- 0 is written to the ADTRADF flag

ADTRBUF Flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting

[Clearing conditions]

- 0 is written to the ADTRBUF flag

ADTRBDF Flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting

[Clearing conditions]

- 0 is written to the ADTRBDF flag

ODF Flag (Output Stop Request Flag)

This flag monitors the output stop request for the group selected by the GTINTAD.GRP[1:0] bits.
Even if a request to disable output is de-asserted after the request is made, releasing the control of negation on the PWM pins has to wait until the end of the PWM cycle.

DTEF Flag (Dead Time Error Flag)

This flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.
This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period. This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When an interrupt by the DTEF flag is enabled (when the GTINTAD.GRPDTE bit is 1), the DTEF flag is output as the output stop detection to the POEG, and an GDTE interrupt is generated whenever the DTEF flag changes from 0 to 1.
[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period (in the following cases)
  Up-counting in triangle-wave mode: GTCCRA register – GTDVU register ≤ 0
  Down-counting in triangle-wave mode: GTCCRA register – GTDVD register < 0
  Up-counting in sawtooth-wave one-shot pulse mode:
  When GTCCRA register – GTDVU register < 0, or GTCCRA register + GTDVD register > GTPR register
  Down-counting in sawtooth-wave one-shot pulse mode:
  When GTCCRA register + GTDVU register > GTPR register, or GTCCRA register – GTDVD register < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

OABHF Flag (Simultaneous High Output Flag)

This flag indicates that both the GTIOCnA and GTIOCnB pins output 1 simultaneously.
When either the GTIOCnA or GTIOCnB pin changes to 0, the flag is set to 0.
This flag is read-only. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABHF flag is enabled (the GTINTAD.GRPABH bit is 1), the OABHF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt triggered by simultaneous High outputs. Use the interrupt function in the POEG if this is required.
[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.
- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

OABLF Flag (Simultaneous Low Output Flag)

This flag indicates that both the GTIOCnA and GTIOCnB pins output 0 simultaneously.
When either the GTIOCnA or GTIOCnB pin changes to 1, the flag is set to 0.
This flag is read-only. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABLF flag is enabled (the GTINTAD.GRPABL bit is 1), the OABLF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt triggered by simultaneous Low outputs. Use the interrupt function in the POEG if this is required.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.
- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

Comparison for the output value to generate the OABHF or OABLF flag uses the value before the compare match output (PWM output) is masked by the output negate function. Even during the output negate condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF Flag (Cycle Count End Flag)

This status flag indicates the end of the cycle count operation.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing conditions]

- 0 is written to the PCF flag

24.2.19 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPTW0.GTBER 8742 C040h, GPTW1.GTBER 8742 C140h, GPTW2.GTBER 8742 C240h, GPTW3.GTBER 8742 C340h, GPTW4.GTBER 8742 C440h, GPTW5.GTBER 8742 C540h, GPTW6.GTBER 8742 C640h, GPTW7.GTBER 8742 C740h, GPTW8.GTBER 8742 C840h, GPTW9.GTBER 8742 C940h, GPTW10.GTBER 8742 CA40h, GPTW11.GTBER 8742 CB40h, GPTW12.GTBER 8742 CC40h, GPTW13.GTBER 8742 CD40h, GPTW14.GTBER 8742 CE40h, GPTW15.GTBER 8742 CF40h, GPTW0_NS.GTBER 0742 C040h, GPTW1_NS.GTBER 0742 C140h, GPTW2_NS.GTBER 0742 C240h, GPTW3_NS.GTBER 0742 C340h, GPTW4_NS.GTBER 0742 C440h, GPTW5_NS.GTBER 0742 C540h, GPTW6_NS.GTBER 0742 C640h, GPTW7_NS.GTBER 0742 C740h, GPTW8_NS.GTBER 0742 C840h, GPTW9_NS.GTBER 0742 C940h, GPTW10_NS.GTBER 0742 CA40h, GPTW11_NS.GTBER 0742 CB40h, GPTW12_NS.GTBER 0742 CC40h, GPTW13_NS.GTBER 0742 CD40h, GPTW14_NS.GTBER 0742 CE40h, GPTW15_NS.GTBER 0742 CF40h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD[3]	BD[2]	BD[1]	BD[0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BD[0]	GTCCRA/GTCCRB Registers Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b1	BD[1]	GTPR Register Buffer Operation Disable		R/W *1
b2	BD[2]	GTADTRA/GTADTRB Registers Buffer Operation Disable		R/W
b3	BD[3]	GTDVU/GTDVD Registers Buffer Operation Disable		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable	0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable	0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CCRA[1:0]	GTCCRA Register Buffer Operation	b17b16 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA register ⇔ GTCCRC register) 1 x: Double buffer operation (GTCCRA register ⇔ GTCCRC register ⇔ GTCCRD register)	R/W
b19, b18	CCRB[1:0]	GTCCRB Register Buffer Operation	b19b18 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB register ⇔ GTCCRE register) 1 x: Double buffer operation (GTCCRB register ⇔ GTCCRE register ⇔ GTCCRF register)	R/W

Bit	Symbol	Bit Name	Description	R/W
b21, b20	PR[1:0]	GTPR Register Buffer Operation	b21b20 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR register ⇒ GTPR register) 1 x: Double buffer operation (GTPDBR register ⇒ GTPBR register ⇒ GTPR register)	R/W
b22	CCRSWT	GTCCRA and GTCCRB Registers Forcible Buffer Operation	Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25, b24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select	• Triangle waves or in complementary PWM mode b25b24 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b25b24 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
b26	ADTDA	GTADTRA Register Double Buffer Operation	0: Single buffer operation (GTADTBRA register ⇒ GTADTRA register) 1: Double buffer operation (GTADTDBRA register ⇒ GTADTBRA register ⇒ GTADTRA register)	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29, b28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select	• Triangle waves or in complementary PWM mode b29b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b29b28 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
b30	ADTDB	GTADTRB Register Double Buffer Operation	0: Single buffer operation (GTADTBRB register ⇒ GTADTRB register) 1: Double buffer operation (GTADTDBRB register ⇒ GTADTBRB register ⇒ GTADTRB register)	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTBER register makes settings for buffer operation.
Set the GTBER register except the BD[3:0] bits while the GTCNT counter is stopped.

BD[0] Bit (GTCCRA/GTCCRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTCCRA, GTCCRC, and GTCCRD registers together and buffer operation using the GTCCRB, GTCCRE, and GTCCRF registers together.
Even though the BD[0] bit is set to 0, buffer operation in the GTCCRB register is not performed if the GTDTCR.TDE bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode. Instead, the compare match value for negative-phase waveform with dead time is set automatically.
In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. No buffer transfer to GTCCRB is performed in

complementary PWM mode. Buffer operation of GTCCRE register and GTCCRF register is enabled/disabled by GTBER2.CP3DB bit.
If 1 is written to the SBDCE or SBDCD bit in the GTSECR register, a value can be set for the BD[0] bit of the channel whose corresponding bit in the GTSECSR register is 1.
When the DBRTECm (m = A, B) bit is 1, setting the BD[0] bit to 1 while the mode of operation is sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 results in transfer from the intermediate buffer to the GTCCRm register.

BD[1] Bit (GTPR Register Buffer Operation Disable)

This bit disables buffer operation using the GTPR, GTPBR, and GTPDBR registers together.
A value for the BD[1] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDPE or SBDPD bit in the GTSECR register.
In complementary PWM mode, the slave channel is also controlled by setting the BD[1] bit of the master channel.

BD[2] Bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.
The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.
A value for the BD[2] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

BD[3] Bit (GTDVU/GTDVD Registers Buffer Operation Disable)

In sawtooth-wave PWM mode 1, sawtooth-wave one-shot pulse mode or triangle-wave PWM mode, this bit disables buffer operation using the GTDVU and GTDBU registers together and buffer operation using the GTDVD and GTDBD registers together.
Even though the BD[3] bit is set to 0, buffer operation in the GTDVD register is not performed if the GTDTCR.TDFER bit is set to 1. Instead, the value in the GTDVU register is set automatically.
In sawtooth-wave PWM mode 2 or complementary PWM mode, this bit is invalid and GTDVU and GTDVD registers don’t perform buffer operation.
The setting is invalid during the event count operation, and the buffer operation using the GTDVU and GTDVD registers is not performed.
A value for the BD[3] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDDE or SBDDD bit in the GTSECR register.

DBRTECm Bit (GTCCRm Register Double Buffer Repeat Operation Enable) (m = A, B)

This setting enables the operation to repeat a transfer to the GTCCRm register from the intermediate buffer by cycle during the buffer transfer disable period when performing the double buffer operation using the GTCCRm register. It is valid in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3.
The disabling period of buffer transfer indicates the period that buffer transfer is stopped by setting of the BD[0] bit (CPU writing or simultaneous buffer operation control by the GTSECSR register) and the period for buffer transfer extended skipping (except the case for skipping by counting both crests and troughs) by the GTEITLB register.
When the DBRTECm bit is 1, writing by the CPU to the GTCCRm register sets the same value in temporary register x (x = C, E). The value of the GTCCRx (x = C, E) register is also transferred to the temporary register x (x = C, E) by forcible buffer transfer.

CCRA[1:0] Bits (GTCCRA Register Buffer Operation)

These bits set buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

CCRB[1:0] Bits (GTCCRB Register Buffer Operation)

These bits set buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

PR[1:0] Bits (GTPR Register Buffer Operation)

These bits set buffer operation with the GTPR, GTPBR, and GTPDBR registers combined.
The setting is invalid in complementary PWM mode. Buffer operation unique to complementary PWM mode is performed regardless of the setting value of the PR[1:0] bits.

CCRSWT Bit (GTCCRA and GTCCRB Registers Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs buffer transfer of the GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.
This bit is valid only when counting is stopped with compare match operation specified.
The setting is invalid in complementary PWM mode.

ADTTA[1:0] Bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.
The setting is invalid during the event count operation.

ADTDA Bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.
The setting is invalid during the event count operation.

ADTTB[1:0] Bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.
The setting is invalid during the event count operation.

ADTDB Bit (GTADTRB Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.
The setting is invalid during the event count operation.

Note 1. The buffer operation mode is fixed in sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3 (64-bit transfer at trough), or complementary PWM mode.

24.2.20 General PWM Timer Buffer Enable Register 2 (GTBER2)

Address(es): GPTW0.GTBER2 8742 C0E0h, GPTW1.GTBER2 8742 C1E0h, GPTW2.GTBER2 8742 C2E0h, GPTW3.GTBER2 8742 C3E0h, GPTW4.GTBER2 8742 C4E0h, GPTW5.GTBER2 8742 C5E0h, GPTW6.GTBER2 8742 C6E0h, GPTW7.GTBER2 8742 C7E0h, GPTW8.GTBER2 8742 C8E0h, GPTW9.GTBER2 8742 C9E0h, GPTW10.GTBER2 8742 CAE0h, GPTW11.GTBER2 8742 CBE0h, GPTW12.GTBER2 8742 CCE0h, GPTW13.GTBER2 8742 CDE0h, GPTW14.GTBER2 8742 CEE0h, GPTW15.GTBER2 8742 CFE0h, GPTW0_NS.GTBER2 0742 C0E0h, GPTW1_NS.GTBER2 0742 C1E0h, GPTW2_NS.GTBER2 0742 C2E0h, GPTW3_NS.GTBER2 0742 C3E0h, GPTW4_NS.GTBER2 0742 C4E0h, GPTW5_NS.GTBER2 0742 C5E0h, GPTW6_NS.GTBER2 0742 C6E0h, GPTW7_NS.GTBER2 0742 C7E0h, GPTW8_NS.GTBER2 0742 C8E0h, GPTW9_NS.GTBER2 0742 C9E0h, GPTW10_NS.GTBER2 0742 CAE0h, GPTW11_NS.GTBER2 0742 CBE0h, GPTW12_NS.GTBER2 0742 CCE0h, GPTW13_NS.GTBER2 0742 CDE0h, GPTW14_NS.GTBER2 0742 CEE0h, GPTW15_NS.GTBER2 0742 CFE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	OLTTB[1:0]		OLTTA[1:0]		CPBTD	CP3DB	—	—	CPTDV	CPTADB	CPTADA	CPTPR	CPTCB	CPTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMTADB	CMTADA	—	CMTCB[1:0]		CMTCA[1:0]		—	—	CCTDV	CCTADB	CCTADA	CCTPR	CCTCB	CCTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCTCA	Counter Clear Source GTCCRA Register Buffer Transfer Disable	0: Enable GTCCRA register buffer transfer by counter clear 1: Disable GTCCRA register buffer transfer by counter clear	R/W
b1	CCTCB	Counter Clear Source GTCCRB Register Buffer Transfer Disable	0: Enable GTCCRB register buffer transfer by counter clear 1: Disable GTCCRB register buffer transfer by counter clear	R/W
b2	CCTPR	Counter Clear Source GTPR Register Buffer Transfer Disable	0: Enable GTPR register buffer transfer by counter clear 1: Disable GTPR register buffer transfer by counter clear	R/W
b3	CCTADA	Counter Clear Source GTADTRA Register Buffer Transfer Disable	0: Enable GTADTRA register buffer transfer by counter clear 1: Disable GTADTRA register buffer transfer by counter clear	R/W
b4	CCTADB	Counter Clear Source GTADTRB Register Buffer Transfer Disable	0: Enable GTADTRB register buffer transfer by counter clear 1: Disable GTADTRB register buffer transfer by counter clear	R/W
b5	CCTDV	Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable	0: Enable GTDVU/GTDVD register buffer transfer by counter clear 1: Disable GTDVU/GTDVD register buffer transfer by counter clear	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	CMTCA[1:0]	Compare Match Source GTCCRA Register Buffer Transfer Enable	b9 b8 0 0: Disable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRA register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
b11, b10	CMTCB[1:0]	Compare Match Source GTCCRB Register Buffer Transfer Enable	b11 b10 0 0: Disable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRB register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13	CMTADA	Compare Match Source GTADTRA Register Buffer Transfer Enable	0: Disable GTADTRA register buffer transfer by compare match of GTADTRA register 1: Enable GTADTRA register buffer transfer by compare match of GTADTRA register	R/W
b14	CMTADB	Compare Match Source GTADTRB Register Buffer Transfer Enable	0: Disable GTADTRB register buffer transfer by compare match of GTADTRB register 1: Enable GTADTRB register buffer transfer by compare match of GTADTRB register	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	CPTCA	Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable	0: Enable GTCCRA register buffer transfer by overflow/underflow 1: Disable GTCCRA register buffer transfer by overflow/underflow	R/W
b17	CPTCB	Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable	0: Enable GTCCRB register buffer transfer by overflow/underflow 1: Disable GTCCRB register buffer transfer by overflow/underflow	R/W
b18	CPTPR	Overflow/Underflow Source GTPR Register Buffer Transfer Disable	0: Enable GTPR register buffer transfer by overflow/underflow 1: Disable GTPR register buffer transfer by overflow/underflow	R/W
b19	CPTADA	Overflow/Underflow Source GTADTRA Register Buffer Transfer Disable	0: Enable GTADTRA register buffer transfer by overflow/ underflow 1: Disable GTADTRA register buffer transfer by overflow/ underflow	R/W
b20	CPTADB	Overflow/Underflow Source GTADTRB Register Buffer Transfer Disable	0: Enable GTADTRB register buffer transfer by overflow/ underflow 1: Disable GTADTRB register buffer transfer by overflow/ underflow	R/W
b21	CPTDV	Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable	0: Enable GTDVU/GTDVD register buffer transfer by overflow/ underflow 1: Disable GTDVU/GTDVD register buffer transfer by overflow/ underflow	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	CP3DB	Complementary PWM Mode 3,4 Double Buffer Select*1	0: Disable double buffer function in complementary PWM mode 3, 4 1: Enable double buffer function in complementary PWM mode 3, 4	R/W
b25	CPBTD	Complementary PWM Mode Buffer Transfer Disable*1	0: Enable buffer transfer from the temporary register to the GTCCRC and GTPBR register 1: Disable buffer transfer from the temporary register to the GTCCRC and GTPBR register	R/W
b27, b26	OLTTA[1:0]	GTIOCnA Output Level Buffer Transfer Timing Select*1	• Triangle waves, complementary PWM mode b27 b26 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b27 b26 0 0: No transfer 0 1: Transfer at the end of period 1 0: Transfer by compare match of the GTCCRA register 1 1: Setting prohibited	R/W
b29, b28	OLTTB[1:0]	GTIOCnB Output Level Buffer Transfer Timing Select*1	• Triangle waves, complementary PWM mode b29 b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b29 b28 0 0: No transfer 0 1: Transfer at the end of period 1 0: Transfer by compare match of the GTCCRB register 1 1: Setting prohibited	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits in each of GPTW0 to GPTW3, and GPTW10 to GPTW15 are reserved, and are read as 0. When writing, write 0 to the bits.

The GTBER2 register makes settings for buffer operation.
Set the CP3DB bit and OLTTm[1:0] (m = A, B) bits when the GTCNT counter is stopped.

CCTCm Bit (Counter Clear Source GTCCRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by counter clear using the GTCCRm, GTCCRn, and GTCCRx registers together (m = A, B; n = C, E; x = D, F).
This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRm[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCm bit is 0, buffer transfer is not performed by counter clear.
If there is a conflict with the CMTCm bit setting, the CCTCm bit setting has priority.
The setting is invalid during the event count operation.

CCTPR Bit (Counter Clear Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTPR, GTPBR, and GTPDBR registers together.
This bit is effective when the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTPR bit is 0, buffer transfer is not performed by counter clear.
The setting is invalid during the event count operation.

CCTADm Bit (Counter Clear Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by counter clear using the GTADTRm, GTADTBRm and GTADTDBRm registers together.
This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.ADTTm[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTADm bit is 0, buffer transfer is not performed by counter clear.
If there is a conflict with the CMTADm bit setting, the CCTADm bit setting has priority.
The setting is invalid during the event count operation.

CCTDV Bit (Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTDVU and GTDBU or GTDVD and GTDBD registers together.
This bit is effective when the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTDV bit is 0, buffer transfer is not performed by counter clear.
The setting is invalid during the event count operation.

CMTCA[1:0] Bits (Compare Match Source GTCCRA Register Buffer Transfer Enable)

These bits enable buffer transfer by compare match of GTCCRA using the GTCCRA, GTCCRC and GTCCRD registers together.
These bits are effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected

by the GTBER.CCRA[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
If there is a conflict with the CCTCA bit setting, the CCTCA bit setting has priority.
The setting is invalid during the event count operation.

CMTCB[1:0] Bits (Compare Match Source GTCCRB Register Buffer Transfer Enable)

These bits enable buffer transfer by compare match of GTCCRB using the GTCCRB, GTCCRE and GTCCRF registers together.
These bits are effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
If there is a conflict with the CCTCB bit setting, the CCTCB bit setting has priority.
The setting is invalid during the event count operation.

CMTADm Bit (Compare Match Source GTADTRm Register Buffer Transfer Enable) (m = A, B)

This bit enables buffer transfer by compare match of the GTADTRm using the GTADTRm, GTADTBRm and GTADTDBRm registers together.
This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.ADTTm[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
If there is a conflict with the CCTADm bit setting, the CCTADm bit setting has priority.
The setting is invalid during the event count operation.

CPTCA Bit (Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in sawtooth-waves using the GTCCRA, GTCCRC and GTCCRD registers together.
This bit is effective when the CCTCA bit is 0 (GTCCRA register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
The setting is invalid during the event count operation.

CPTCB Bit (Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in sawtooth-waves using the GTCCRB, GTCCRE and GTCCRF registers together.
This bit is effective when the CCTCB bit is 0 (GTCCRB register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
The setting is invalid during the event count operation.

CPTPR Bit (Overflow/Underflow Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in sawtooth-waves using the GTPR, GTPBR and GTPDBR registers together.
This bit is effective when the CCTPR bit is 0 (GTPR register buffer transfer by counter clear is enabled), the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode.
The setting is invalid during the event count operation.

CPTADm Bit (Overflow/Underflow Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by overflow/underflow in sawtooth-waves using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the CCTADm bit is 0 (GTADTRm register buffer transfer by counter clear is enabled), the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.ADTTm[1:0] bits with sawtooth-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTDV Bit (Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in sawtooth-waves using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the CCTDV bit is 0 (GTDVU/GTDVD register buffer transfer by counter clear is enabled), the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with sawtooth-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CP3DB Bit (Complementary PWM Mode 3,4 Double Buffer Select)

This bit enables buffer transfer using the GTCCRA, GTCCRE and GTCCRF registers together in complementary PWM mode 3,4.

CPBTD Bit (Complementary PWM Mode Buffer Transfer Disable)

This bit disables buffer transfer from temporary register (temporary register A and temporary register P) to the GTCCRC and GTPBR registers during timer counting in complementary PWM mode 1, 2 and 3. When CP3DB bit is 1, the buffer transfer from temporary register B to GTCCRE register is also disabled. The setting is invalid in complementary PWM mode 4.

OLTTm[1:0] Bits (GTIOCnm Output Level Buffer Transfer Timing Select) (m = A, B)

These bits set the timing of buffer transfer from GTOLBR.GTIOmB[4:0] bits to GTIOR.GTIOm[4:0] bits.

The setting is invalid during the event count operation.

24.2.21 General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register (GTITC)

Address(es): GPTW0.GTITC 8742 C044h, GPTW1.GTITC 8742 C144h, GPTW2.GTITC 8742 C244h, GPTW3.GTITC 8742 C344h, GPTW4.GTITC 8742 C444h, GPTW5.GTITC 8742 C544h, GPTW6.GTITC 8742 C644h, GPTW7.GTITC 8742 C744h, GPTW8.GTITC 8742 C844h, GPTW9.GTITC 8742 C944h, GPTW10.GTITC 8742 CA44h, GPTW11.GTITC 8742 CB44h, GPTW12.GTITC 8742 CC44h, GPTW13.GTITC 8742 CD44h, GPTW14.GTITC 8742 CE44h, GPTW15.GTITC 8742 CF44h, GPTW0_NS.GTITC 0742 C044h, GPTW1_NS.GTITC 0742 C144h, GPTW2_NS.GTITC 0742 C244h, GPTW3_NS.GTITC 0742 C344h, GPTW4_NS.GTITC 0742 C444h, GPTW5_NS.GTITC 0742 C544h, GPTW6_NS.GTITC 0742 C644h, GPTW7_NS.GTITC 0742 C744h, GPTW8_NS.GTITC 0742 C844h, GPTW9_NS.GTITC 0742 C944h, GPTW10_NS.GTITC 0742 CA44h, GPTW11_NS.GTITC 0742 CB44h, GPTW12_NS.GTITC 0742 CC44h, GPTW13_NS.GTITC 0742 CD44h, GPTW14_NS.GTITC 0742 CE44h, GPTW15_NS.GTITC 0742 CF44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b2	ITLC	GTCCRC Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b3	ITLD	GTCCRD Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b4	ITLE	GTCCRE Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b5	ITLF	GTCCRF Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b7, b6	IVTC[1:0]	GTCIV/GTCIU Interrupt Skipping Function Select*1	b7 b6 0 0: Skipping is not performed 0 1: Both overflow and underflow for sawtooth waves and crest for triangle waves and complementary PWM mode are counted and skipped. 1 0: Both overflow and underflow for sawtooth waves and trough for triangle waves and complementary PWM mode are counted and skipped. 1 1: Both overflow and underflow for sawtooth waves and both crest and trough for triangle waves and complementary PWM mode are counted and skipped	R/W
b10 to b8	IVTT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Select	b10 b8 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADTAL	GTADTRA Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b14	ADTBL	GTADTRB Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This function is not usable with GPTW4 to GPTW9 in the sawtooth-wave PWM mode 2.

The GTITC register configures the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GTCIV)/underflow interrupt (GTCIU), and determines whether other interrupts and A/D conversion start requests are linked to the GTCIV/GTCIU interrupt skipping function. Note that output stop detection to the POEG and dead-time error interrupts cannot be linked with the GTCIV or GTCIU interrupt skipping function. Additionally, when the interrupt skipping function is performed, the change in the status flag is also skipped.
The setting is invalid during the event count operation.
The setting is operated independently of the extended interrupt skipping by the GTEITC register.

ITLA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GTCIA) with the GTCIV/GTCIU interrupt skipping function.

ITLB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GTCIB) with the GTCIV/GTCIU interrupt skipping function.

ITLC Bit (GTCCRC Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GTCIC) with the GTCIV/GTCIU interrupt skipping function.

ITLD Bit (GTCCRD Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GTCID) with the GTCIV/GTCIU interrupt skipping function.

ITLE Bit (GTCCRE Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GTCIE) with the GTCIV/GTCIU interrupt skipping function.

ITLF Bit (GTCCRF Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GTCIF) with the GTCIV/GTCIU interrupt skipping function.

IVTC[1:0] Bits (GTCIV/GTCIU Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/GTCNT counter underflow interrupt (GTCIU).

IVTT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/GTCNT counter underflow interrupt (GTCIU).
When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL Bit (GTADTRA Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match

with the GTCNT counter and the GTADTRA register, with the GTCIV/GTCIU interrupt skipping function.

ADTBL Bit (GTADTRB Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GTCIV/GTCIU interrupt skipping function.

24.2.22 General PWM Timer Counter (GTCNT)

Address(es): GPTW0.GTCNT 8742 C048h, GPTW1.GTCNT 8742 C148h, GPTW2.GTCNT 8742 C248h, GPTW3.GTCNT 8742 C348h, GPTW4.GTCNT 8742 C448h, GPTW5.GTCNT 8742 C548h, GPTW6.GTCNT 8742 C648h, GPTW7.GTCNT 8742 C748h, GPTW8.GTCNT 8742 C848h, GPTW9.GTCNT 8742 C948h, GPTW10.GTCNT 8742 CA48h, GPTW11.GTCNT 8742 CB48h, GPTW12.GTCNT 8742 CC48h, GPTW13.GTCNT 8742 CD48h, GPTW14.GTCNT 8742 CE48h, GPTW15.GTCNT 8742 CF48h, GPTW0_NS.GTCNT 0742 C048h, GPTW1_NS.GTCNT 0742 C148h, GPTW2_NS.GTCNT 0742 C248h, GPTW3_NS.GTCNT 0742 C348h, GPTW4_NS.GTCNT 0742 C448h, GPTW5_NS.GTCNT 0742 C548h, GPTW6_NS.GTCNT 0742 C648h, GPTW7_NS.GTCNT 0742 C748h, GPTW8_NS.GTCNT 0742 C848h, GPTW9_NS.GTCNT 0742 C948h, GPTW10_NS.GTCNT 0742 CA48h, GPTW11_NS.GTCNT 0742 CB48h, GPTW12_NS.GTCNT 0742 CC48h, GPTW13_NS.GTCNT 0742 CD48h, GPTW14_NS.GTCNT 0742 CE48h, GPTW15_NS.GTCNT 0742 CF48h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTCNT counter is a 32-bit readable/writable counter. There is one GTCNT counter for each channel. Writing is available only when counting is stopped, and thus writing is disabled while counting is in progress (when CST = 1). Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units.
When the sawtooth or triangle waves are to be used in GPTW4 to GPTW9, set the GTCNT counter within the range from 0 up to the value of the GTPR register.

24.2.23 General PWM Timer Compare Capture Register m (GTCCRm) (m = A to F)

Address(es): GPTW0.GTCCRA 8742 C04Ch, GPTW1.GTCCRA 8742 C14Ch, GPTW2.GTCCRA 8742 C24Ch, GPTW3.GTCCRA 8742 C34Ch, GPTW4.GTCCRA 8742 C44Ch, GPTW5.GTCCRA 8742 C54Ch, GPTW6.GTCCRA 8742 C64Ch, GPTW7.GTCCRA 8742 C74Ch, GPTW8.GTCCRA 8742 C84Ch, GPTW9.GTCCRA 8742 C94Ch, GPTW10.GTCCRA 8742 CA4Ch, GPTW11.GTCCRA 8742 CB4Ch, GPTW12.GTCCRA 8742 CC4Ch, GPTW13.GTCCRA 8742 CD4Ch, GPTW14.GTCCRA 8742 CE4Ch, GPTW15.GTCCRA 8742 CF4Ch, GPTW0.GTCCRB 8742 C050h, GPTW1.GTCCRB 8742 C150h, GPTW2.GTCCRB 8742 C250h, GPTW3.GTCCRB 8742 C350h, GPTW4.GTCCRB 8742 C450h, GPTW5.GTCCRB 8742 C550h, GPTW6.GTCCRB 8742 C650h, GPTW7.GTCCRB 8742 C750h, GPTW8.GTCCRB 8742 C850h, GPTW9.GTCCRB 8742 C950h, GPTW10.GTCCRB 8742 CA50h, GPTW11.GTCCRB 8742 CB50h, GPTW12.GTCCRB 8742 CC50h, GPTW13.GTCCRB 8742 CD50h, GPTW14.GTCCRB 8742 CE50h, GPTW15.GTCCRB 8742 CF50h, GPTW0.GTCCRC 8742 C054h, GPTW1.GTCCRC 8742 C154h, GPTW2.GTCCRC 8742 C254h, GPTW3.GTCCRC 8742 C354h, GPTW4.GTCCRC 8742 C454h, GPTW5.GTCCRC 8742 C554h, GPTW6.GTCCRC 8742 C654h, GPTW7.GTCCRC 8742 C754h, GPTW8.GTCCRC 8742 C854h, GPTW9.GTCCRC 8742 C954h, GPTW10.GTCCRC 8742 CA54h, GPTW11.GTCCRC 8742 CB54h, GPTW12.GTCCRC 8742 CC54h, GPTW13.GTCCRC 8742 CD54h, GPTW14.GTCCRC 8742 CE54h, GPTW15.GTCCRC 8742 CF54h, GPTW0.GTCCRE 8742 C058h, GPTW1.GTCCRE 8742 C158h, GPTW2.GTCCRE 8742 C258h, GPTW3.GTCCRE 8742 C358h, GPTW4.GTCCRE 8742 C458h, GPTW5.GTCCRE 8742 C558h, GPTW6.GTCCRE 8742 C658h, GPTW7.GTCCRE 8742 C758h, GPTW8.GTCCRE 8742 C858h, GPTW9.GTCCRE 8742 C958h, GPTW10.GTCCRE 8742 CA58h, GPTW11.GTCCRE 8742 CB58h, GPTW12.GTCCRE 8742 CC58h, GPTW13.GTCCRE 8742 CD58h, GPTW14.GTCCRE 8742 CE58h, GPTW15.GTCCRE 8742 CF58h, GPTW0.GTCCRD 8742 C05Ch, GPTW1.GTCCRD 8742 C15Ch, GPTW2.GTCCRD 8742 C25Ch, GPTW3.GTCCRD 8742 C35Ch, GPTW4.GTCCRD 8742 C45Ch, GPTW5.GTCCRD 8742 C55Ch, GPTW6.GTCCRD 8742 C65Ch, GPTW7.GTCCRD 8742 C75Ch, GPTW8.GTCCRD 8742 C85Ch, GPTW9.GTCCRD 8742 C95Ch, GPTW10.GTCCRD 8742 CA5Ch, GPTW11.GTCCRD 8742 CB5Ch, GPTW12.GTCCRD 8742 CC5Ch, GPTW13.GTCCRD 8742 CD5Ch, GPTW14.GTCCRD 8742 CE5Ch, GPTW15.GTCCRD 8742 CF5Ch, GPTW0.GTCCRF 8742 C060h, GPTW1.GTCCRF 8742 C160h, GPTW2.GTCCRF 8742 C260h, GPTW3.GTCCRF 8742 C360h, GPTW4.GTCCRF 8742 C460h, GPTW5.GTCCRF 8742 C560h, GPTW6.GTCCRF 8742 C660h, GPTW7.GTCCRF 8742 C760h, GPTW8.GTCCRF 8742 C860h, GPTW9.GTCCRF 8742 C960h, GPTW10.GTCCRF 8742 CA60h, GPTW11.GTCCRF 8742 CB60h, GPTW12.GTCCRF 8742 CC60h, GPTW13.GTCCRF 8742 CD60h, GPTW14.GTCCRF 8742 CE60h, GPTW15.GTCCRF 8742 CF60h, GPTW0_NS.GTCCRA 0742 C04Ch, GPTW1_NS.GTCCRA 0742 C14Ch, GPTW2_NS.GTCCRA 0742 C24Ch, GPTW3_NS.GTCCRA 0742 C34Ch, GPTW4_NS.GTCCRA 0742 C44Ch, GPTW5_NS.GTCCRA 0742 C54Ch, GPTW6_NS.GTCCRA 0742 C64Ch, GPTW7_NS.GTCCRA 0742 C74Ch, GPTW8_NS.GTCCRA 0742 C84Ch, GPTW9_NS.GTCCRA 0742 C94Ch, GPTW10_NS.GTCCRA 0742 CA4Ch, GPTW11_NS.GTCCRA 0742 CB4Ch, GPTW12_NS.GTCCRA 0742 CC4Ch, GPTW13_NS.GTCCRA 0742 CD4Ch, GPTW14_NS.GTCCRA 0742 CE4Ch, GPTW15_NS.GTCCRA 0742 CF4Ch,
GPTW0_NS.GTCCRB 0742 C050h, GPTW1_NS.GTCCRB 0742 C150h, GPTW2_NS.GTCCRB 0742 C250h, GPTW3_NS.GTCCRB 0742 C350h, GPTW4_NS.GTCCRB 0742 C450h, GPTW5_NS.GTCCRB 0742 C550h, GPTW6_NS.GTCCRB 0742 C650h, GPTW7_NS.GTCCRB 0742 C750h, GPTW8_NS.GTCCRB 0742 C850h, GPTW9_NS.GTCCRB 0742 C950h, GPTW10_NS.GTCCRB 0742 CA50h, GPTW11_NS.GTCCRB 0742 CB50h, GPTW12_NS.GTCCRB 0742 CC50h, GPTW13_NS.GTCCRB 0742 CD50h, GPTW14_NS.GTCCRB 0742 CE50h, GPTW15_NS.GTCCRB 0742 CF50h,
GPTW0_NS.GTCCRC 0742 C054h, GPTW1_NS.GTCCRC 0742 C154h, GPTW2_NS.GTCCRC 0742 C254h, GPTW3_NS.GTCCRC 0742 C354h, GPTW4_NS.GTCCRC 0742 C454h, GPTW5_NS.GTCCRC 0742 C554h, GPTW6_NS.GTCCRC 0742 C654h, GPTW7_NS.GTCCRC 0742 C754h, GPTW8_NS.GTCCRC 0742 C854h, GPTW9_NS.GTCCRC 0742 C954h, GPTW10_NS.GTCCRC 0742 CA54h, GPTW11_NS.GTCCRC 0742 CB54h, GPTW12_NS.GTCCRC 0742 CC54h, GPTW13_NS.GTCCRC 0742 CD54h, GPTW14_NS.GTCCRC 0742 CE54h, GPTW15_NS.GTCCRC 0742 CF54h,
GPTW0_NS.GTCCRE 0742 C058h, GPTW1_NS.GTCCRE 0742 C158h, GPTW2_NS.GTCCRE 0742 C258h, GPTW3_NS.GTCCRE 0742 C358h, GPTW4_NS.GTCCRE 0742 C458h, GPTW5_NS.GTCCRE 0742 C558h, GPTW6_NS.GTCCRE 0742 C658h, GPTW7_NS.GTCCRE 0742 C758h, GPTW8_NS.GTCCRE 0742 C858h, GPTW9_NS.GTCCRE 0742 C958h, GPTW10_NS.GTCCRE 0742 CA58h, GPTW11_NS.GTCCRE 0742 CB58h, GPTW12_NS.GTCCRE 0742 CC58h, GPTW13_NS.GTCCRE 0742 CD58h, GPTW14_NS.GTCCRE 0742 CE58h, GPTW15_NS.GTCCRE 0742 CF58h,
GPTW0_NS.GTCCRD 0742 C05Ch, GPTW1_NS.GTCCRD 0742 C15Ch, GPTW2_NS.GTCCRD 0742 C25Ch, GPTW3_NS.GTCCRD 0742 C35Ch, GPTW4_NS.GTCCRD 0742 C45Ch, GPTW5_NS.GTCCRD 0742 C55Ch, GPTW6_NS.GTCCRD 0742 C65Ch, GPTW7_NS.GTCCRD 0742 C75Ch, GPTW8_NS.GTCCRD 0742 C85Ch, GPTW9_NS.GTCCRD 0742 C95Ch, GPTW10_NS.GTCCRD 0742 CA5Ch, GPTW11_NS.GTCCRD 0742 CB5Ch, GPTW12_NS.GTCCRD 0742 CC5Ch, GPTW13_NS.GTCCRD 0742 CD5Ch, GPTW14_NS.GTCCRD 0742 CE5Ch, GPTW15_NS.GTCCRD 0742 CF5Ch,
GPTW0_NS.GTCCRF 0742 C060h, GPTW1_NS.GTCCRF 0742 C160h, GPTW2_NS.GTCCRF 0742 C260h, GPTW3_NS.GTCCRF 0742 C360h, GPTW4_NS.GTCCRF 0742 C460h, GPTW5_NS.GTCCRF 0742 C560h, GPTW6_NS.GTCCRF 0742 C660h, GPTW7_NS.GTCCRF 0742 C760h, GPTW8_NS.GTCCRF 0742 C860h, GPTW9_NS.GTCCRF 0742 C960h, GPTW10_NS.GTCCRF 0742 CA60h, GPTW11_NS.GTCCRF 0742 CB60h, GPTW12_NS.GTCCRF 0742 CC60h, GPTW13_NS.GTCCRF 0742 CD60h, GPTW14_NS.GTCCRF 0742 CE60h, GPTW15_NS.GTCCRF 0742 CF60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTCCRm register is a 32-bit readable/writable register.

Access in 8-bit or 16-bit units to the GTCCRm register is prohibited, and it should be accessed in 32-bit units.

The GTCCRA and GTCCRB registers are registers used for both output compare and input capture.

The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers.

The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers).

24.2.24 General PWM Timer Period Setting Register (GTPR)

Address(es): GPTW0.GTPR 8742 C064h, GPTW1.GTPR 8742 C164h, GPTW2.GTPR 8742 C264h, GPTW3.GTPR 8742 C364h, GPTW4.GTPR 8742 C464h, GPTW5.GTPR 8742 C564h, GPTW6.GTPR 8742 C664h, GPTW7.GTPR 8742 C764h, GPTW8.GTPR 8742 C864h, GPTW9.GTPR 8742 C964h, GPTW10.GTPR 8742 CA64h, GPTW11.GTPR 8742 CB64h, GPTW12.GTPR 8742 CC64h, GPTW13.GTPR 8742 CD64h, GPTW14.GTPR 8742 CE64h, GPTW15.GTPR 8742 CF64h, GPTW0_NS.GTPR 0742 C064h, GPTW1_NS.GTPR 0742 C164h, GPTW2_NS.GTPR 0742 C264h, GPTW3_NS.GTPR 0742 C364h, GPTW4_NS.GTPR 0742 C464h, GPTW5_NS.GTPR 0742 C564h, GPTW6_NS.GTPR 0742 C664h, GPTW7_NS.GTPR 0742 C764h, GPTW8_NS.GTPR 0742 C864h, GPTW9_NS.GTPR 0742 C964h, GPTW10_NS.GTPR 0742 CA64h, GPTW11_NS.GTPR 0742 CB64h, GPTW12_NS.GTPR 0742 CC64h, GPTW13_NS.GTPR 0742 CD64h, GPTW14_NS.GTPR 0742 CE64h, GPTW15_NS.GTPR 0742 CF64h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPR register is a 32-bit readable/writable register that sets the maximum count value of the GTCNT counter. Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. In sawtooth-wave mode except the sawtooth-wave PWM mode 2, the value of (GTPR register + 1) is the count period. In triangle-wave mode or complementary PWM mode, the value of (GTPR register value × 2) is the count period.
When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.25 General PWM Timer Period Setting Buffer Register (GTPBR)

Address(es): GPTW0.GTPBR 8742 C068h, GPTW1.GTPBR 8742 C168h, GPTW2.GTPBR 8742 C268h, GPTW3.GTPBR 8742 C368h, GPTW4.GTPBR 8742 C468h, GPTW5.GTPBR 8742 C568h, GPTW6.GTPBR 8742 C668h, GPTW7.GTPBR 8742 C768h, GPTW8.GTPBR 8742 C868h, GPTW9.GTPBR 8742 C968h, GPTW10.GTPBR 8742 CA68h, GPTW11.GTPBR 8742 CB68h, GPTW12.GTPBR 8742 CC68h, GPTW13.GTPBR 8742 CD68h, GPTW14.GTPBR 8742 CE68h, GPTW15.GTPBR 8742 CF68h, GPTW0_NS.GTPBR 0742 C068h, GPTW1_NS.GTPBR 0742 C168h, GPTW2_NS.GTPBR 0742 C268h, GPTW3_NS.GTPBR 0742 C368h, GPTW4_NS.GTPBR 0742 C468h, GPTW5_NS.GTPBR 0742 C568h, GPTW6_NS.GTPBR 0742 C668h, GPTW7_NS.GTPBR 0742 C768h, GPTW8_NS.GTPBR 0742 C868h, GPTW9_NS.GTPBR 0742 C968h, GPTW10_NS.GTPBR 0742 CA68h, GPTW11_NS.GTPBR 0742 CB68h, GPTW12_NS.GTPBR 0742 CC68h, GPTW13_NS.GTPBR 0742 CD68h, GPTW14_NS.GTPBR 0742 CE68h, GPTW15_NS.GTPBR 0742 CF68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPR register. Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.26 General PWM Timer Period Setting Double-Buffer Register (GTPDBR)

Address(es): GPTW0.GTPDBR 8742 C06Ch, GPTW1.GTPDBR 8742 C16Ch, GPTW2.GTPDBR 8742 C26Ch, GPTW3.GTPDBR 8742 C36Ch, GPTW4.GTPDBR 8742 C46Ch, GPTW5.GTPDBR 8742 C56Ch, GPTW6.GTPDBR 8742 C66Ch, GPTW7.GTPDBR 8742 C76Ch, GPTW8.GTPDBR 8742 C86Ch, GPTW9.GTPDBR 8742 C96Ch, GPTW10.GTPDBR 8742 CA6Ch, GPTW11.GTPDBR 8742 CB6Ch, GPTW12.GTPDBR 8742 CC6Ch, GPTW13.GTPDBR 8742 CD6Ch, GPTW14.GTPDBR 8742 CE6Ch, GPTW15.GTPDBR 8742 CF6Ch, GPTW0_NS.GTPDBR 0742 C06Ch, GPTW1_NS.GTPDBR 0742 C16Ch, GPTW2_NS.GTPDBR 0742 C26Ch, GPTW3_NS.GTPDBR 0742 C36Ch, GPTW4_NS.GTPDBR 0742 C46Ch, GPTW5_NS.GTPDBR 0742 C56Ch, GPTW6_NS.GTPDBR 0742 C66Ch, GPTW7_NS.GTPDBR 0742 C76Ch, GPTW8_NS.GTPDBR 0742 C86Ch, GPTW9_NS.GTPDBR 0742 C96Ch, GPTW10_NS.GTPDBR 0742 CA6Ch, GPTW11_NS.GTPDBR 0742 CB6Ch, GPTW12_NS.GTPDBR 0742 CC6Ch, GPTW13_NS.GTPDBR 0742 CD6Ch, GPTW14_NS.GTPDBR 0742 CE6Ch, GPTW15_NS.GTPDBR 0742 CF6Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPDBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register).

Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units.

The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.27 A/D Conversion Start Request Timing Register m (GTADTRm) (m = A, B)

Address(es): GPTW0.GTADTRA 8742 C070h, GPTW1.GTADTRA 8742 C170h, GPTW2.GTADTRA 8742 C270h, GPTW3.GTADTRA 8742 C370h, GPTW4.GTADTRA 8742 C470h, GPTW5.GTADTRA 8742 C570h, GPTW6.GTADTRA 8742 C670h, GPTW7.GTADTRA 8742 C770h, GPTW8.GTADTRA 8742 C870h, GPTW9.GTADTRA 8742 C970h, GPTW10.GTADTRA 8742 CA70h, GPTW11.GTADTRA 8742 CB70h, GPTW12.GTADTRA 8742 CC70h, GPTW13.GTADTRA 8742 CD70h, GPTW14.GTADTRA 8742 CE70h, GPTW15.GTADTRA 8742 CF70h,
GPTW0.GTADTRB 8742 C07Ch, GPTW1.GTADTRB 8742 C17Ch, GPTW2.GTADTRB 8742 C27Ch, GPTW3.GTADTRB 8742 C37Ch, GPTW4.GTADTRB 8742 C47Ch, GPTW5.GTADTRB 8742 C57Ch, GPTW6.GTADTRB 8742 C67Ch, GPTW7.GTADTRB 8742 C77Ch, GPTW8.GTADTRB 8742 C87Ch, GPTW9.GTADTRB 8742 C97Ch, GPTW10.GTADTRB 8742 CA7Ch, GPTW11.GTADTRB 8742 CB7Ch, GPTW12.GTADTRB 8742 CC7Ch, GPTW13.GTADTRB 8742 CD7Ch, GPTW14.GTADTRB 8742 CE7Ch, GPTW15.GTADTRB 8742 CF7Ch,
GPTW0_NS.GTADTRA 0742 C070h, GPTW1_NS.GTADTRA 0742 C170h, GPTW2_NS.GTADTRA 0742 C270h, GPTW3_NS.GTADTRA 0742 C370h, GPTW4_NS.GTADTRA 0742 C470h, GPTW5_NS.GTADTRA 0742 C570h, GPTW6_NS.GTADTRA 0742 C670h, GPTW7_NS.GTADTRA 0742 C770h, GPTW8_NS.GTADTRA 0742 C870h, GPTW9_NS.GTADTRA 0742 C970h, GPTW10_NS.GTADTRA 0742 CA70h, GPTW11_NS.GTADTRA 0742 CB70h, GPTW12_NS.GTADTRA 0742 CC70h, GPTW13_NS.GTADTRA 0742 CD70h, GPTW14_NS.GTADTRA 0742 CE70h, GPTW15_NS.GTADTRA 0742 CF70h,
GPTW0_NS.GTADTRB 0742 C07Ch, GPTW1_NS.GTADTRB 0742 C17Ch, GPTW2_NS.GTADTRB 0742 C27Ch, GPTW3_NS.GTADTRB 0742 C37Ch, GPTW4_NS.GTADTRB 0742 C47Ch, GPTW5_NS.GTADTRB 0742 C57Ch, GPTW6_NS.GTADTRB 0742 C67Ch, GPTW7_NS.GTADTRB 0742 C77Ch, GPTW8_NS.GTADTRB 0742 C87Ch, GPTW9_NS.GTADTRB 0742 C97Ch, GPTW10_NS.GTADTRB 0742 CA7Ch, GPTW11_NS.GTADTRB 0742 CB7Ch, GPTW12_NS.GTADTRB 0742 CC7Ch, GPTW13_NS.GTADTRB 0742 CD7Ch, GPTW14_NS.GTADTRB 0742 CE7Ch, GPTW15_NS.GTADTRB 0742 CF7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTRm register is a 32-bit readable/writable register that sets the timing of A/D conversion start request generation.

Access in 8-bit or 16-bit units to the GTADTRm register is prohibited, and it should be accessed in 32-bit units.

When the GTADTRm register value matches the GTCNT counter value, an A/D conversion start request is generated. In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.

24.2.28 A/D Conversion Start Request Timing Buffer Register m (GTADTBRm) (m = A, B)

Address(es): GPTW0.GTADTBRA 8742 C074h, GPTW1.GTADTBRA 8742 C174h, GPTW2.GTADTBRA 8742 C274h, GPTW3.GTADTBRA 8742 C374h, GPTW4.GTADTBRA 8742 C474h, GPTW5.GTADTBRA 8742 C574h, GPTW6.GTADTBRA 8742 C674h, GPTW7.GTADTBRA 8742 C774h, GPTW8.GTADTBRA 8742 C874h, GPTW9.GTADTBRA 8742 C974h, GPTW10.GTADTBRA 8742 CA74h, GPTW11.GTADTBRA 8742 CB74h, GPTW12.GTADTBRA 8742 CC74h, GPTW13.GTADTBRA 8742 CD74h, GPTW14.GTADTBRA 8742 CE74h, GPTW15.GTADTBRA 8742 CF74h,
GPTW0.GTADTBRB 8742 C080h, GPTW1.GTADTBRB 8742 C180h, GPTW2.GTADTBRB 8742 C280h, GPTW3.GTADTBRB 8742 C380h, GPTW4.GTADTBRB 8742 C480h, GPTW5.GTADTBRB 8742 C580h, GPTW6.GTADTBRB 8742 C680h, GPTW7.GTADTBRB 8742 C780h, GPTW8.GTADTBRB 8742 C880h, GPTW9.GTADTBRB 8742 C980h, GPTW10.GTADTBRB 8742 CA80h, GPTW11.GTADTBRB 8742 CB80h, GPTW12.GTADTBRB 8742 CC80h, GPTW13.GTADTBRB 8742 CD80h, GPTW14.GTADTBRB 8742 CE80h, GPTW15.GTADTBRB 8742 CF80h,
GPTW0_NS.GTADTBRA 0742 C074h, GPTW1_NS.GTADTBRA 0742 C174h, GPTW2_NS.GTADTBRA 0742 C274h, GPTW3_NS.GTADTBRA 0742 C374h, GPTW4_NS.GTADTBRA 0742 C474h, GPTW5_NS.GTADTBRA 0742 C574h, GPTW6_NS.GTADTBRA 0742 C674h, GPTW7_NS.GTADTBRA 0742 C774h, GPTW8_NS.GTADTBRA 0742 C874h, GPTW9_NS.GTADTBRA 0742 C974h, GPTW10_NS.GTADTBRA 0742 CA74h, GPTW11_NS.GTADTBRA 0742 CB74h, GPTW12_NS.GTADTBRA 0742 CC74h, GPTW13_NS.GTADTBRA 0742 CD74h, GPTW14_NS.GTADTBRA 0742 CE74h, GPTW15_NS.GTADTBRA 0742 CF74h,
GPTW0_NS.GTADTBRB 0742 C080h, GPTW1_NS.GTADTBRB 0742 C180h, GPTW2_NS.GTADTBRB 0742 C280h, GPTW3_NS.GTADTBRB 0742 C380h, GPTW4_NS.GTADTBRB 0742 C480h, GPTW5_NS.GTADTBRB 0742 C580h, GPTW6_NS.GTADTBRB 0742 C680h, GPTW7_NS.GTADTBRB 0742 C780h, GPTW8_NS.GTADTBRB 0742 C880h, GPTW9_NS.GTADTBRB 0742 C980h, GPTW10_NS.GTADTBRB 0742 CA80h, GPTW11_NS.GTADTBRB 0742 CB80h, GPTW12_NS.GTADTBRB 0742 CC80h, GPTW13_NS.GTADTBRB 0742 CD80h, GPTW14_NS.GTADTBRB 0742 CE80h, GPTW15_NS.GTADTBRB 0742 CF80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTBRm register is a 32-bit readable/writable register that functions as a buffer register for the GTADTRm register.

Access in 8-bit or 16-bit units to the GTADTBRm register is prohibited, and it should be accessed in 32-bit units.

24.2.29 A/D Conversion Start Request Timing Double-Buffer Register m (GTADTDBRm) (m = A, B)

Address(es): GPTW0.GTADTDBRA 8742 C078h, GPTW1.GTADTDBRA 8742 C178h, GPTW2.GTADTDBRA 8742 C278h, GPTW3.GTADTDBRA 8742 C378h, GPTW4.GTADTDBRA 8742 C478h, GPTW5.GTADTDBRA 8742 C578h, GPTW6.GTADTDBRA 8742 C678h, GPTW7.GTADTDBRA 8742 C778h, GPTW8.GTADTDBRA 8742 C878h, GPTW9.GTADTDBRA 8742 C978h, GPTW10.GTADTDBRA 8742 CA78h, GPTW11.GTADTDBRA 8742 CB78h, GPTW12.GTADTDBRA 8742 CC78h, GPTW13.GTADTDBRA 8742 CD78h, GPTW14.GTADTDBRA 8742 CE78h, GPTW15.GTADTDBRA 8742 CF78h,
GPTW0.GTADTDBRB 8742 C084h, GPTW1.GTADTDBRB 8742 C184h, GPTW2.GTADTDBRB 8742 C284h, GPTW3.GTADTDBRB 8742 C384h, GPTW4.GTADTDBRB 8742 C484h, GPTW5.GTADTDBRB 8742 C584h, GPTW6.GTADTDBRB 8742 C684h, GPTW7.GTADTDBRB 8742 C784h, GPTW8.GTADTDBRB 8742 C884h, GPTW9.GTADTDBRB 8742 C984h, GPTW10.GTADTDBRB 8742 CA84h, GPTW11.GTADTDBRB 8742 CB84h, GPTW12.GTADTDBRB 8742 CC84h, GPTW13.GTADTDBRB 8742 CD84h, GPTW14.GTADTDBRB 8742 CE84h, GPTW15.GTADTDBRB 8742 CF84h,
GPTW0_NS.GTADTDBRA 0742 C078h, GPTW1_NS.GTADTDBRA 0742 C178h, GPTW2_NS.GTADTDBRA 0742 C278h, GPTW3_NS.GTADTDBRA 0742 C378h, GPTW4_NS.GTADTDBRA 0742 C478h, GPTW5_NS.GTADTDBRA 0742 C578h, GPTW6_NS.GTADTDBRA 0742 C678h, GPTW7_NS.GTADTDBRA 0742 C778h, GPTW8_NS.GTADTDBRA 0742 C878h, GPTW9_NS.GTADTDBRA 0742 C978h, GPTW10_NS.GTADTDBRA 0742 CA78h, GPTW11_NS.GTADTDBRA 0742 CB78h, GPTW12_NS.GTADTDBRA 0742 CC78h, GPTW13_NS.GTADTDBRA 0742 CD78h, GPTW14_NS.GTADTDBRA 0742 CE78h, GPTW15_NS.GTADTDBRA 0742 CF78h,
GPTW0_NS.GTADTDBRB 0742 C084h, GPTW1_NS.GTADTDBRB 0742 C184h, GPTW2_NS.GTADTDBRB 0742 C284h, GPTW3_NS.GTADTDBRB 0742 C384h, GPTW4_NS.GTADTDBRB 0742 C484h, GPTW5_NS.GTADTDBRB 0742 C584h, GPTW6_NS.GTADTDBRB 0742 C684h, GPTW7_NS.GTADTDBRB 0742 C784h, GPTW8_NS.GTADTDBRB 0742 C884h, GPTW9_NS.GTADTDBRB 0742 C984h, GPTW10_NS.GTADTDBRB 0742 CA84h, GPTW11_NS.GTADTDBRB 0742 CB84h, GPTW12_NS.GTADTDBRB 0742 CC84h, GPTW13_NS.GTADTDBRB 0742 CD84h, GPTW14_NS.GTADTDBRB 0742 CE84h, GPTW15_NS.GTADTDBRB 0742 CF84h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTDBRm register is 32-bit readable/writable register that function as buffer register for the GTADTBRm register (double buffer register for the GTADTRm register).
Access in 8-bit or 16-bit units to the GTADTDBRm register is prohibited, and it should be accessed in 32-bit units.

24.2.30 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPTW0.GTDTCR 8742 C088h, GPTW1.GTDTCR 8742 C188h, GPTW2.GTDTCR 8742 C288h, GPTW3.GTDTCR 8742 C388h, GPTW4.GTDTCR 8742 C488h, GPTW5.GTDTCR 8742 C588h, GPTW6.GTDTCR 8742 C688h, GPTW7.GTDTCR 8742 C788h, GPTW8.GTDTCR 8742 C888h, GPTW9.GTDTCR 8742 C988h, GPTW10.GTDTCR 8742 CA88h, GPTW11.GTDTCR 8742 CB88h, GPTW12.GTDTCR 8742 CC88h, GPTW13.GTDTCR 8742 CD88h, GPTW14.GTDTCR 8742 CE88h, GPTW15.GTDTCR 8742 CF88h, GPTW0_NS.GTDTCR 0742 C088h, GPTW1_NS.GTDTCR 0742 C188h, GPTW2_NS.GTDTCR 0742 C288h, GPTW3_NS.GTDTCR 0742 C388h, GPTW4_NS.GTDTCR 0742 C488h, GPTW5_NS.GTDTCR 0742 C588h, GPTW6_NS.GTDTCR 0742 C688h, GPTW7_NS.GTDTCR 0742 C788h, GPTW8_NS.GTDTCR 0742 C888h, GPTW9_NS.GTDTCR 0742 C988h, GPTW10_NS.GTDTCR 0742 CA88h, GPTW11_NS.GTDTCR 0742 CB88h, GPTW12_NS.GTDTCR 0742 CC88h, GPTW13_NS.GTDTCR 0742 CD88h, GPTW14_NS.GTDTCR 0742 CE88h, GPTW15_NS.GTDTCR 0742 CF88h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: GTCCRB register is set without using GTDVU and GTDVD registers. 1: GTDVU and GTDVD registers are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB register.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TDBUE	GTDVU Register Buffer Operation Enable	0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
b5	TDBDE	GTDVD Register Buffer Operation Enable	0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TDFER	GTDVD Register Setting	0: GTDVU and GTDVD registers are set separately. 1: The value written to GTDVU register is automatically set to GTDVD register.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. The setting is invalid during the event count operation.
This register is invalid in sawtooth-wave PWM mode 1, 2 or complementary PWM mode.

TDE Bit (Negative-Phase Waveform Setting)

This bit sets whether to use the GTDVU and GTDVD registers. When the GTDVU and GTDVD registers are used, the compare match value for a negative-phase waveform with dead time that was obtained by the compare match value of a positive-phase waveform (GTCCRA register) and the dead time value (GTDVU and GTDVD registers) is automatically set in the GTCCRB register.
The TDE bit setting is ignored in sawtooth-wave PWM mode, and the GTCCRB register is not automatic setting.
The automatically set value of the GTCCRB register has the following upper and lower limits:

- Triangle wave PWM mode
  Upper limit: the GTPR register value – 1
  Lower limit: 0000 0001h in up-counting, 0000 0000h in down-counting
- Sawtooth-wave one-shot pulse mode
  Upper limit: the GTPR register value
  Lower limit: 0000 0000h

If the obtained GTCCRB register value is not within the range between the upper and lower limits, the upper or lower limit is set in the GTCCRB register, and the GTST.DTEF flag becomes 1. However, if the obtained GTCCRB register value exceeds the upper limit in triangle-wave PWM mode, the DTEF flag becomes 0.

TDBUE Bit (GTDVU Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVU and GTDBU registers combined.
The timing of buffer transfer is at troughs in triangle-wave mode, and at overflows or underflows in sawtooth-wave mode.

TDBDE Bit (GTDVD Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVD and GTDBD registers combined.
The buffer transfer timing is the trough in triangle-wave mode, and at an overflow or underflow in sawtooth-wave mode.
When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

TDFER Bit (GTDVD Register Setting)

This bit specifies whether the value written to the GTDVU register is automatically set to the GTDVD register.

24.2.31 General PWM Timer Dead Time Value Register m (GTDVm) (m = U, D)

Address(es): GPTW0.GTDVU 8742 C08Ch, GPTW1.GTDVU 8742 C18Ch, GPTW2.GTDVU 8742 C28Ch, GPTW3.GTDVU 8742 C38Ch, GPTW4.GTDVU 8742 C48Ch, GPTW5.GTDVU 8742 C58Ch, GPTW6.GTDVU 8742 C68Ch, GPTW7.GTDVU 8742 C78Ch, GPTW8.GTDVU 8742 C88Ch, GPTW9.GTDVU 8742 C98Ch, GPTW10.GTDVU 8742 CA8Ch, GPTW11.GTDVU 8742 CB8Ch, GPTW12.GTDVU 8742 CC8Ch, GPTW13.GTDVU 8742 CD8Ch, GPTW14.GTDVU 8742 CE8Ch, GPTW15.GTDVU 8742 CF8Ch, GPTW0.GTDVD 8742 C090h, GPTW1.GTDVD 8742 C190h, GPTW2.GTDVD 8742 C290h, GPTW3.GTDVD 8742 C390h, GPTW4.GTDVD 8742 C490h, GPTW5.GTDVD 8742 C590h, GPTW6.GTDVD 8742 C690h, GPTW7.GTDVD 8742 C790h, GPTW8.GTDVD 8742 C890h, GPTW9.GTDVD 8742 C990h, GPTW10.GTDVD 8742 CA90h, GPTW11.GTDVD 8742 CB90h, GPTW12.GTDVD 8742 CC90h, GPTW13.GTDVD 8742 CD90h, GPTW14.GTDVD 8742 CE90h, GPTW15.GTDVD 8742 CF90h, GPTW0_NS.GTDVU 0742 C08Ch, GPTW1_NS.GTDVU 0742 C18Ch, GPTW2_NS.GTDVU 0742 C28Ch, GPTW3_NS.GTDVU 0742 C38Ch, GPTW4_NS.GTDVU 0742 C48Ch, GPTW5_NS.GTDVU 0742 C58Ch, GPTW6_NS.GTDVU 0742 C68Ch, GPTW7_NS.GTDVU 0742 C78Ch, GPTW8_NS.GTDVU 0742 C88Ch, GPTW9_NS.GTDVU 0742 C98Ch, GPTW10_NS.GTDVU 0742 CA8Ch, GPTW11_NS.GTDVU 0742 CB8Ch, GPTW12_NS.GTDVU 0742 CC8Ch, GPTW13_NS.GTDVU 0742 CD8Ch, GPTW14_NS.GTDVU 0742 CE8Ch, GPTW15_NS.GTDVU 0742 CF8Ch, GPTW0_NS.GTDVD 0742 C090h, GPTW1_NS.GTDVD 0742 C190h, GPTW2_NS.GTDVD 0742 C290h, GPTW3_NS.GTDVD 0742 C390h, GPTW4_NS.GTDVD 0742 C490h, GPTW5_NS.GTDVD 0742 C590h, GPTW6_NS.GTDVD 0742 C690h, GPTW7_NS.GTDVD 0742 C790h, GPTW8_NS.GTDVD 0742 C890h, GPTW9_NS.GTDVD 0742 C990h, GPTW10_NS.GTDVD 0742 CA90h, GPTW11_NS.GTDVD 0742 CB90h, GPTW12_NS.GTDVD 0742 CC90h, GPTW13_NS.GTDVD 0742 CD90h, GPTW14_NS.GTDVD 0742 CE90h, GPTW15_NS.GTDVD 0742 CF90h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTDVm register is a 32-bit readable/writable register that sets the dead time for generating PWM waveforms with dead time.

Access in 8-bit or 16-bit units to the GTDVm register is prohibited, and it should be accessed in 32-bit units.

The setting is invalid in sawtooth-wave PWM mode 1, 2.

In complementary PWM mode, the GTDVD register is invalid and the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

In triangle-wave mode, the GTDVU register is used for up-counting, and the GTDVD register is used for down-counting.

In sawtooth-wave mode, the GTDVU register controls the first-half dead time, and the GTDVD register controls the second-half dead time, both in up-counting and down-counting.

Do not set this register to a value equal to or larger than that of the GTPR register.

- GTDVU > 0
- GTDVU < GTPR/2
- GTDVU + GTPR ≤ FFFF FFFFh

In addition, when using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When using the GTDVm register, writing to the GTCCRB register is prohibited. In modes other than complementary PWM, setting this register to 0 leads to the output of waveforms without dead time.

When the GTDTCR.TDFER bit is 1, writing to the GTDVD register has no effect. In this case, reading the GTDVD register returns the value of the GTDVU register.

24.2.32 General PWM Timer Dead Time Value Buffer Register m (GTDBm) (m = U, D)

Address(es): GPTW0.GTDBU 8742 C094h, GPTW1.GTDBU 8742 C194h, GPTW2.GTDBU 8742 C294h, GPTW3.GTDBU 8742 C394h, GPTW4.GTDBU 8742 C494h, GPTW5.GTDBU 8742 C594h, GPTW6.GTDBU 8742 C694h, GPTW7.GTDBU 8742 C794h, GPTW8.GTDBU 8742 C894h, GPTW9.GTDBU 8742 C994h, GPTW10.GTDBU 8742 CA94h, GPTW11.GTDBU 8742 CB94h, GPTW12.GTDBU 8742 CC94h, GPTW13.GTDBU 8742 CD94h, GPTW14.GTDBU 8742 CE94h, GPTW15.GTDBU 8742 CF94h, GPTW0.GTDBD 8742 C098h, GPTW1.GTDBD 8742 C198h, GPTW2.GTDBD 8742 C298h, GPTW3.GTDBD 8742 C398h, GPTW4.GTDBD 8742 C498h, GPTW5.GTDBD 8742 C598h, GPTW6.GTDBD 8742 C698h, GPTW7.GTDBD 8742 C798h, GPTW8.GTDBD 8742 C898h, GPTW9.GTDBD 8742 C998h, GPTW10.GTDBD 8742 CA98h, GPTW11.GTDBD 8742 CB98h, GPTW12.GTDBD 8742 CC98h, GPTW13.GTDBD 8742 CD98h, GPTW14.GTDBD 8742 CE98h, GPTW15.GTDBD 8742 CF98h, GPTW0_NS.GTDBU 0742 C094h, GPTW1_NS.GTDBU 0742 C194h, GPTW2_NS.GTDBU 0742 C294h, GPTW3_NS.GTDBU 0742 C394h, GPTW4_NS.GTDBU 0742 C494h, GPTW5_NS.GTDBU 0742 C594h, GPTW6_NS.GTDBU 0742 C694h, GPTW7_NS.GTDBU 0742 C794h, GPTW8_NS.GTDBU 0742 C894h, GPTW9_NS.GTDBU 0742 C994h, GPTW10_NS.GTDBU 0742 CA94h, GPTW11_NS.GTDBU 0742 CB94h, GPTW12_NS.GTDBU 0742 CC94h, GPTW13_NS.GTDBU 0742 CD94h, GPTW14_NS.GTDBU 0742 CE94h, GPTW15_NS.GTDBU 0742 CF94h, GPTW0_NS.GTDBD 0742 C098h, GPTW1_NS.GTDBD 0742 C198h, GPTW2_NS.GTDBD 0742 C298h, GPTW3_NS.GTDBD 0742 C398h, GPTW4_NS.GTDBD 0742 C498h, GPTW5_NS.GTDBD 0742 C598h, GPTW6_NS.GTDBD 0742 C698h, GPTW7_NS.GTDBD 0742 C798h, GPTW8_NS.GTDBD 0742 C898h, GPTW9_NS.GTDBD 0742 C998h, GPTW10_NS.GTDBD 0742 CA98h, GPTW11_NS.GTDBD 0742 CB98h, GPTW12_NS.GTDBD 0742 CC98h, GPTW13_NS.GTDBD 0742 CD98h, GPTW14_NS.GTDBD 0742 CE98h, GPTW15_NS.GTDBD 0742 CF98h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTDBm register is a 32-bit readable/writable register that functions as a buffer register for the GTDVm register. Access in 8-bit or 16-bit units to the GTDBm register is prohibited, and it should be accessed in 32-bit units.

24.2.33 General PWM Timer Output Protection Function Status Register (GTSOS)

Address(es): GPTW0.GTSOS 8742 C09Ch, GPTW1.GTSOS 8742 C19Ch, GPTW2.GTSOS 8742 C29Ch, GPTW3.GTSOS 8742 C39Ch, GPTW4.GTSOS 8742 C49Ch, GPTW5.GTSOS 8742 C59Ch, GPTW6.GTSOS 8742 C69Ch, GPTW7.GTSOS 8742 C79Ch, GPTW8.GTSOS 8742 C89Ch, GPTW9.GTSOS 8742 C99Ch, GPTW10.GTSOS 8742 CA9Ch, GPTW11.GTSOS 8742 CB9Ch, GPTW12.GTSOS 8742 CC9Ch, GPTW13.GTSOS 8742 CD9Ch, GPTW14.GTSOS 8742 CE9Ch, GPTW15.GTSOS 8742 CF9Ch, GPTW0_NS.GTSOS 0742 C09Ch, GPTW1_NS.GTSOS 0742 C19Ch, GPTW2_NS.GTSOS 0742 C29Ch, GPTW3_NS.GTSOS 0742 C39Ch, GPTW4_NS.GTSOS 0742 C49Ch, GPTW5_NS.GTSOS 0742 C59Ch, GPTW6_NS.GTSOS 0742 C69Ch, GPTW7_NS.GTSOS 0742 C79Ch, GPTW8_NS.GTSOS 0742 C89Ch, GPTW9_NS.GTSOS 0742 C99Ch, GPTW10_NS.GTSOS 0742 CA9Ch, GPTW11_NS.GTSOS 0742 CB9Ch, GPTW12_NS.GTSOS 0742 CC9Ch, GPTW13_NS.GTSOS 0742 CD9Ch, GPTW14_NS.GTSOS 0742 CE9Ch, GPTW15_NS.GTSOS 0742 CF9Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SOS[1:0]	Output Protection Function Status	b1 b0 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b9, b8	—	Reserved	The read value is undefined.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (the GTDTCR.TDE bit is 1) in triangle-wave mode.

SOS[1:0] Bits (Output Protection Function Status)

These bits indicate the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 15; m = A, B).

24.2.34 General PWM Timer Output Protection Function Temporary Release Register (GTSOTR)

Address(es): GPTW0.GTSOTR 8742 C0A0h, GPTW1.GTSOTR 8742 C1A0h, GPTW2.GTSOTR 8742 C2A0h, GPTW3.GTSOTR 8742 C3A0h, GPTW4.GTSOTR 8742 C4A0h, GPTW5.GTSOTR 8742 C5A0h, GPTW6.GTSOTR 8742 C6A0h, GPTW7.GTSOTR 8742 C7A0h, GPTW8.GTSOTR 8742 C8A0h, GPTW9.GTSOTR 8742 C9A0h, GPTW10.GTSOTR 8742 CAA0h, GPTW11.GTSOTR 8742 CBA0h, GPTW12.GTSOTR 8742 CCA0h, GPTW13.GTSOTR 8742 CDA0h, GPTW14.GTSOTR 8742 CEA0h, GPTW15.GTSOTR 8742 CFA0h, GPTW0_NS.GTSOTR 0742 C0A0h, GPTW1_NS.GTSOTR 0742 C1A0h, GPTW2_NS.GTSOTR 0742 C2A0h, GPTW3_NS.GTSOTR 0742 C3A0h, GPTW4_NS.GTSOTR 0742 C4A0h, GPTW5_NS.GTSOTR 0742 C5A0h, GPTW6_NS.GTSOTR 0742 C6A0h, GPTW7_NS.GTSOTR 0742 C7A0h, GPTW8_NS.GTSOTR 0742 C8A0h, GPTW9_NS.GTSOTR 0742 C9A0h, GPTW10_NS.GTSOTR 0742 CAA0h, GPTW11_NS.GTSOTR 0742 CBA0h, GPTW12_NS.GTSOTR 0742 CCA0h, GPTW13_NS.GTSOTR 0742 CDA0h, GPTW14_NS.GTSOTR 0742 CEA0h, GPTW15_NS.GTSOTR 0742 CFA0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOTR	Output Protection Function Temporary Release	0: Protected state is not released 1: Protected state is released	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output (n = 0 to 15) when output protection has been set.

The protected state can be released only for the case of the GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough). The protected state cannot be released for any other case.

SOTR Bit (Output Protection Function Temporary Release)

This bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state. After the SOTR bit has been set to 1, the output protection function is canceled from the first trough. After the SOTR bit has been set to 0, output protection is resumed from the first trough.

24.2.35 General PWM Timer A/D Conversion Start Request Signal Monitoring Register (GTADSMR)

Address(es): GPTW0.GTADSMR 8742 C0A4h, GPTW1.GTADSMR 8742 C1A4h, GPTW2.GTADSMR 8742 C2A4h, GPTW3.GTADSMR 8742 C3A4h, GPTW4.GTADSMR 8742 C4A4h, GPTW5.GTADSMR 8742 C5A4h, GPTW6.GTADSMR 8742 C6A4h, GPTW7.GTADSMR 8742 C7A4h, GPTW8.GTADSMR 8742 C8A4h, GPTW9.GTADSMR 8742 C9A4h, GPTW10.GTADSMR 8742 CAA4h, GPTW11.GTADSMR 8742 CBA4h, GPTW12.GTADSMR 8742 CCA4h, GPTW13.GTADSMR 8742 CDA4h, GPTW14.GTADSMR 8742 CEA4h, GPTW15.GTADSMR 8742 CFA4h, GPTW0_NS.GTADSMR 0742 C0A4h, GPTW1_NS.GTADSMR 0742 C1A4h, GPTW2_NS.GTADSMR 0742 C2A4h, GPTW3_NS.GTADSMR 0742 C3A4h, GPTW4_NS.GTADSMR 0742 C4A4h, GPTW5_NS.GTADSMR 0742 C5A4h, GPTW6_NS.GTADSMR 0742 C6A4h, GPTW7_NS.GTADSMR 0742 C7A4h, GPTW8_NS.GTADSMR 0742 C8A4h, GPTW9_NS.GTADSMR 0742 C9A4h, GPTW10_NS.GTADSMR 0742 CAA4h, GPTW11_NS.GTADSMR 0742 CBA4h, GPTW12_NS.GTADSMR 0742 CCA4h, GPTW13_NS.GTADSMR 0742 CDA4h, GPTW14_NS.GTADSMR 0742 CEA4h, GPTW15_NS.GTADSMR 0742 CFA4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection	b1 b0 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling	0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection	b17b16 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling	0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the monitoring of the A/D conversion start request signal that is synchronized with a frame.

ADSMSk[1:0] Bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

These bits are used to select an A/D conversion start request signal synchronized with a frame to be monitored by the GTADSMk pin.

In triangle-wave PWM mode or complementary PWM mode, the following settings are prohibited:

- Set the ADSMSk[1:0] bits to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set the ADSMSk[1:0] bits to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set the ADSMSk[1:0] bits to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set the ADSMSk[1:0] bits to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk Bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

This bit enables or disables the monitor output to the GTADSMk pin.
When the output is disabled, the GTADSMk pin goes to the low level.

When this bit is 1, the GTADSMk pin goes high on assertion of the A/D conversion start request signal selected by the ADSMSk[1:0] bits, and returns low at the end of the timer cycle for the channel that generated the request. If the counter stops, the output retains the level at the time the counter stopped. Set the ADSMENk bit to 0 to output a low level.
When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.
When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPTW.

24.2.36 General PWM Timer Extended Interrupt Skipping Counter Control Register (GTEITC)

Address(es): GPTW0.GTEITC 8742 C0A8h, GPTW1.GTEITC 8742 C1A8h, GPTW2.GTEITC 8742 C2A8h, GPTW3.GTEITC 8742 C3A8h, GPTW4.GTEITC 8742 C4A8h, GPTW5.GTEITC 8742 C5A8h, GPTW6.GTEITC 8742 C6A8h, GPTW7.GTEITC 8742 C7A8h, GPTW8.GTEITC 8742 C8A8h, GPTW9.GTEITC 8742 C9A8h, GPTW10.GTEITC 8742 CAA8h, GPTW11.GTEITC 8742 CBA8h, GPTW12.GTEITC 8742 CCA8h, GPTW13.GTEITC 8742 CDA8h, GPTW14.GTEITC 8742 CEA8h, GPTW15.GTEITC 8742 CFA8h, GPTW0_NS.GTEITC 0742 C0A8h, GPTW1_NS.GTEITC 0742 C1A8h, GPTW2_NS.GTEITC 0742 C2A8h, GPTW3_NS.GTEITC 0742 C3A8h, GPTW4_NS.GTEITC 0742 C4A8h, GPTW5_NS.GTEITC 0742 C5A8h, GPTW6_NS.GTEITC 0742 C6A8h, GPTW7_NS.GTEITC 0742 C7A8h, GPTW8_NS.GTEITC 0742 C8A8h, GPTW9_NS.GTEITC 0742 C9A8h, GPTW10_NS.GTEITC 0742 CAA8h, GPTW11_NS.GTEITC 0742 CBA8h, GPTW12_NS.GTEITC 0742 CCA8h, GPTW13_NS.GTEITC 0742 CDA8h, GPTW14_NS.GTEITC 0742 CEA8h, GPTW15_NS.GTEITC 0742 CFA8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select	b1 b0 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting	Skipping count for the extended interrupt skipping 1	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1	The extended interrupt skipping counter 1	R
b17, b16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source Select	b17b16 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting	Skipping count for the extended interrupt skipping 2	R/W
b27 to b24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value	The extended interrupt skipping counter 2 initial value	R/W *1
b31 to b28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2	The extended interrupt skipping counter 2	R

Note 1. The EITCNT2IV[3:0] bits are only writable when the value other than 00b is written to the EIVTC2[1:0] bits that have been 00b.

The GTEITC register sets the extended interrupt skipping function to skip the interrupts, A/D conversion start requests, and buffer transfers independently by counting at overflow and underflow in the GTCNT counter.
The setting is operated independently of the interrupt skipping by the GTITC register, or the GTADCMSC register in GPTW4 to GPTW9.
The setting is invalid during the event count operation.
Access in 8-bit units to GTEITC is prohibited.

EIVTCk[1:0] Bits (Extended Interrupt Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the extended interrupt skipping counter k. Setting only with these bits does not skip the interrupts, A/D conversion start requests, and buffer transfers.
Skipping function for the interrupt, A/D conversion start request, and buffer transfer, all of which are a target of skipping, is set individually with the GTEITLI1, GTEITLI2, and GTEITLB registers.

EIVTTk[3:0] Bits (Extended Interrupt Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the EIVTCk[1:0] bits to the next generation of the count source. When the count source is generated while the EIVTTk[3:0] bits match the EITCNTk[3:0] bits, the EITCNTk[3:0] bits are cleared.
When these bits are 0000b, skipping is not performed.

EITCNT1[3:0] Bits (Extended Interrupt Skipping Counter 1)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the EIVTT1[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
[Clearing conditions]

- 00b is written to the EIVTC1[1:0] bits
- 0000b is written to the EIVTT1[3:0] bits
- A count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated when the extended interrupt skipping 1 skipping count set by the EIVTT1[3:0] bits match the value for the EITCNT1[3:0]

EITCNT2IV[3:0] Bits (Extended Interrupt Skipping Counter 2 Initial Value)

These bits are the initial value for the extended interrupt skipping counter 2. Writing to the EITCNT2IV[3:0] is performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the EITCNT2[3:0] bits are set not to count (EIVTC2[1:0] bits are 00b). When the EITCNT2IV[3:0] bits are written, the value written to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits simultaneously.
Writing to the EITCNT2IV[3:0] bits is ignored when the EITCNT2[3:0] bits are set to count (EIVTC2[1:0] bits are other than 00b) or when attempt to write 00b (do not count) to the EIVTC2[1:0] bits.
The EITCNT2IV[3:0] bits are not reset by writing 00b to the EIVTC2[1:0] bits.

EITCNT2[3:0] Bits (Extended Interrupt Skipping Counter 2)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the EIVTT2[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
Setting of the initial value for the EITCNT2[3:0] bits are performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the extended interrupt skipping counter 2 is set as not to count (00b is written to the EIVTC2[1:0] bits).

When the initial value is set, the written value to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits as the initial value.
[Clearing conditions]

- 00b is written to the EIVTC2[1:0] bits
- 0000b is written to the EIVTT2[3:0] bits
- When the EIVTC2[1:0] bits are 00b, a value other than 00b is written to the EIVTC2[1:0] bits and 0000b is written to the EITCNT2IV[3:0] bits at the same time.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the EIVTT2[3:0] bits match the value for the EITCNT2[3:0]

24.2.37 General PWM Timer Extended Interrupt Skipping Setting Register 1 (GTEITLI1)

Address(es): GPTW0.GTEITLI1 8742 C0ACh, GPTW1.GTEITLI1 8742 C1ACh, GPTW2.GTEITLI1 8742 C2ACh, GPTW3.GTEITLI1 8742 C3ACh, GPTW4.GTEITLI1 8742 C4ACh, GPTW5.GTEITLI1 8742 C5ACh, GPTW6.GTEITLI1 8742 C6ACh, GPTW7.GTEITLI1 8742 C7ACh, GPTW8.GTEITLI1 8742 C8ACh, GPTW9.GTEITLI1 8742 C9ACh, GPTW10.GTEITLI1 8742 CAACh, GPTW11.GTEITLI1 8742 CBACh, GPTW12.GTEITLI1 8742 CCACh, GPTW13.GTEITLI1 8742 CDACh, GPTW14.GTEITLI1 8742 CEACh, GPTW15.GTEITLI1 8742 CFACh,
GPTW0_NS.GTEITLI1 0742 C0ACh, GPTW1_NS.GTEITLI1 0742 C1ACh, GPTW2_NS.GTEITLI1 0742 C2ACh, GPTW3_NS.GTEITLI1 0742 C3ACh, GPTW4_NS.GTEITLI1 0742 C4ACh, GPTW5_NS.GTEITLI1 0742 C5ACh, GPTW6_NS.GTEITLI1 0742 C6ACh, GPTW7_NS.GTEITLI1 0742 C7ACh, GPTW8_NS.GTEITLI1 0742 C8ACh, GPTW9_NS.GTEITLI1 0742 C9ACh, GPTW10_NS.GTEITLI1 0742 CAACh, GPTW11_NS.GTEITLI1 0742 CBACh, GPTW12_NS.GTEITLI1 0742 CCACh, GPTW13_NS.GTEITLI1 0742 CDACh, GPTW14_NS.GTEITLI1 0742 CEACh, GPTW15_NS.GTEITLI1 0742 CFACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EITLD[2:0]			—	EITLC[2:0]			—	EITLB[2:0]			—	EITLA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EITLA[2:0]	GTCCRA Register Compare Match/ Input Capture Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EITLB[2:0]	GTCCRB Register Compare Match/ Input Capture Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	EITLC[2:0]	GTCCRC Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	EITLD[2:0]	GTCCRD Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	EITLE[2:0]	GTCCRE Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	EITLF[2:0]	GTCCRF Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow, and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding

extended interrupt skipping counter operates the counting.
The setting is operated independently of the interrupt skipping by the GTITC register or the GTADCMSC register.
The setting is invalid during the event count operation.

EITLm[2:0] Bits (GTCCRm Register Compare Match/Input Capture Interrupt Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip the compare match/input capture interrupt (GTCIm) in the GTCCRm register.
Refer to Table 24.6.

EITLx[2:0] Bits (GTCCRx Register Compare Match Interrupt Extended Skipping Function Select) (x = C, D, E, F)

These bits select the extended interrupt skipping function to skip the compare match interrupt (GTCIx) in the GTCCRx register.
Refer to Table 24.6.

EITLV[2:0] Bits (Overflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at overflow.
Refer to Table 24.6.

EITLU[2:0] Bits (Underflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at underflow.
Refer to Table 24.6.

Table 24.6 Setting the Function Select for the GTEITLI1 Register

EITLy[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An interrupt is output in the period of EITCNT1 bit = 0)
0 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An interrupt is output in the period of EITCNT2 bit = 0)
0 1 1	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An interrupt is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An interrupt is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An interrupt is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An interrupt is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

y = A, B, C, D, E, F, V, U; k = 1, 2
Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
Note: When the EITLy[2:0] bits are set to 011b or 111b, and when one of the skipping counter1 or the skipping counter 2 is set as not to count, skipping is not performed.

24.2.38 General PWM Timer Extended Interrupt Skipping Setting Register 2 (GTEITLI2)

Address(es): GPTW0.GTEITLI2 8742 C0B0h, GPTW1.GTEITLI2 8742 C1B0h, GPTW2.GTEITLI2 8742 C2B0h, GPTW3.GTEITLI2 8742 C3B0h, GPTW4.GTEITLI2 8742 C4B0h, GPTW5.GTEITLI2 8742 C5B0h, GPTW6.GTEITLI2 8742 C6B0h, GPTW7.GTEITLI2 8742 C7B0h, GPTW8.GTEITLI2 8742 C8B0h, GPTW9.GTEITLI2 8742 C9B0h, GPTW10.GTEITLI2 8742 CAB0h, GPTW11.GTEITLI2 8742 CBB0h, GPTW12.GTEITLI2 8742 CCB0h, GPTW13.GTEITLI2 8742 CDB0h, GPTW14.GTEITLI2 8742 CEB0h, GPTW15.GTEITLI2 8742 CFB0h, GPTW0_NS.GTEITLI2 0742 C0B0h, GPTW1_NS.GTEITLI2 0742 C1B0h, GPTW2_NS.GTEITLI2 0742 C2B0h, GPTW3_NS.GTEITLI2 0742 C3B0h, GPTW4_NS.GTEITLI2 0742 C4B0h, GPTW5_NS.GTEITLI2 0742 C5B0h, GPTW6_NS.GTEITLI2 0742 C6B0h, GPTW7_NS.GTEITLI2 0742 C7B0h, GPTW8_NS.GTEITLI2 0742 C8B0h, GPTW9_NS.GTEITLI2 0742 C9B0h, GPTW10_NS.GTEITLI2 0742 CAB0h, GPTW11_NS.GTEITLI2 0742 CBB0h, GPTW12_NS.GTEITLI2 0742 CCB0h, GPTW13_NS.GTEITLI2 0742 CDB0h, GPTW14_NS.GTEITLI2 0742 CEB0h, GPTW15_NS.GTEITLI2 0742 CFB0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	EADTBL[2:0]			—	EADTAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EADTAL[2:0]	GTADTRA A/D Conversion Start Request Extended Skipping Function Select	See Table 24.7.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EADTBL[2:0]	GTADTRB A/D Conversion Start Request Extended Skipping Function Select	See Table 24.7.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D conversion start requests.
Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.
The setting is operated independently of the interrupt skipping by the GTITC register or the GTADCMSC register.
The setting is invalid during the event count operation.

EADTmL[2:0] Bits (GTADTRm A/D Conversion Start Request Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip A/D conversion start requests for the compare match in the GTADTRm register.
Refer to Table 24.7.

Table 24.7 Setting the Function Select for the GTEITLI2 Register

EADTmL[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of EITCNT1 bit = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of EITCNT2 bit = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

m = A, B; k = 1, 2

Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.

Note: When the EADTmL[2:0] bits are set to 011b or 111b, and when one of the skipping counter1 or the skipping counter 2 is set as not to count, skipping is not performed.

24.2.39 General PWM Timer Extended Buffer Transfer Skipping Setting Register (GTEITLB)

Address(es): GPTW0.GTEITLB 8742 C0B4h, GPTW1.GTEITLB 8742 C1B4h, GPTW2.GTEITLB 8742 C2B4h, GPTW3.GTEITLB 8742 C3B4h, GPTW4.GTEITLB 8742 C4B4h, GPTW5.GTEITLB 8742 C5B4h, GPTW6.GTEITLB 8742 C6B4h, GPTW7.GTEITLB 8742 C7B4h, GPTW8.GTEITLB 8742 C8B4h, GPTW9.GTEITLB 8742 C9B4h, GPTW10.GTEITLB 8742 CAB4h, GPTW11.GTEITLB 8742 CBB4h, GPTW12.GTEITLB 8742 CCB4h, GPTW13.GTEITLB 8742 CDB4h, GPTW14.GTEITLB 8742 CEB4h, GPTW15.GTEITLB 8742 CFB4h, GPTW0_NS.GTEITLB 0742 C0B4h, GPTW1_NS.GTEITLB 0742 C1B4h, GPTW2_NS.GTEITLB 0742 C2B4h, GPTW3_NS.GTEITLB 0742 C3B4h, GPTW4_NS.GTEITLB 0742 C4B4h, GPTW5_NS.GTEITLB 0742 C5B4h, GPTW6_NS.GTEITLB 0742 C6B4h, GPTW7_NS.GTEITLB 0742 C7B4h, GPTW8_NS.GTEITLB 0742 C8B4h, GPTW9_NS.GTEITLB 0742 C9B4h, GPTW10_NS.GTEITLB 0742 CAB4h, GPTW11_NS.GTEITLB 0742 CBB4h, GPTW12_NS.GTEITLB 0742 CCB4h, GPTW13_NS.GTEITLB 0742 CDB4h, GPTW14_NS.GTEITLB 0742 CEB4h, GPTW15_NS.GTEITLB 0742 CFB4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	EBTLPR[2:0]			—	EBTLCB[2:0]			—	EBTLCA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W *1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. In the complementary PWM mode, writing to these bits in any of the registers in the master channel, slave channel 1, or slave channel 2 simultaneously applies to all three channels.

The GTEITLB register sets the extended skipping function for buffer transfers.
Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.
The setting is operated independently of the interrupt skipping by the GTITC register.
The buffer transfer from the GTOLBR register to the GTIOR.GTIOA[4:0], GTIOB[4:0] bits is not target of the extended buffer transfer skipping function.
The setting is invalid during the event count operation.

EBTLCA[2:0] Bits (GTCCRA Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRA, GTCCRC, GTCCRD registers and a temporary register A) in the GTCCRA register.
An extended skipping of buffer transfers in the GTCCRA register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRA register should be performed in the condition of not performing the extended buffer transfer skipping.
The buffer transfer between the GTCCRC, GTCCRE register, and GTCCRA register in complementary PWM mode cannot be skipped.

EBTLCB[2:0] Bits (GTCCRB Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRB, GTCCRE, GTCCRF registers and a temporary register B) in the GTCCRB register.
An extended skipping of buffer transfers in the GTCCRB register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRB register should be performed in the condition of not performing the extended buffer transfer skipping.

EBTLPR[2:0] Bits (GTPR Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTPR, GTPBR, and GTPDBR registers) in the GTPR register.
When skipping GTPR buffer transfers in complementary PWM mode, GTEITC register setting of slave channel should be matched to the master channel so that the buffer transfer timing of the slave channel matches the master channel.

EBTLADm[2:0] Bits (GTADTRm Register Buffer Transfer Extended Skipping Function Select) (m = A, B)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTADTRm, GTADTBRm, and GTADTDBRm registers) in the GTADTRm register.

EBTLDVm[2:0] Bits (GTDVm Register Buffer Transfer Extended Skipping Function Select) (m = U, D)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers between the GTDVm and GTDBm registers) in the GTDVm register.

Table 24.8 Setting the Function Select for the GTEITLB Register

EBTLx[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0 (A buffer transfer is output in the period of EITCNT1 bit = 0)
0 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0 (A buffer transfer is output in the period of EITCNT2 bit = 0)
0 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (A buffer transfer is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (A buffer transfer is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (A buffer transfer is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (A buffer transfer is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

x = CA, CB, PR, ADA, ADB, DVU, DVD; k = 1, 2

Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
Note: When the EBTLx[2:0] bits are set to 011b or 111b, and when one of the skipping counter1 or the skipping counter 2 is set as not to count, skipping is not performed.

24.2.40 General PWM Timer Inter Channel Logical Operation Function Setting Register (GTICLF)

Address(es): GPTW0.GTICLF 8742 C0B8h, GPTW1.GTICLF 8742 C1B8h, GPTW2.GTICLF 8742 C2B8h, GPTW3.GTICLF 8742 C3B8h, GPTW4.GTICLF 8742 C4B8h, GPTW5.GTICLF 8742 C5B8h, GPTW6.GTICLF 8742 C6B8h, GPTW7.GTICLF 8742 C7B8h, GPTW8.GTICLF 8742 C8B8h, GPTW9.GTICLF 8742 C9B8h, GPTW10.GTICLF 8742 CAB8h, GPTW11.GTICLF 8742 CBB8h, GPTW12.GTICLF 8742 CCB8h, GPTW13.GTICLF 8742 CDB8h, GPTW14.GTICLF 8742 CEB8h, GPTW15.GTICLF 8742 CFB8h, GPTW0_NS.GTICLF 0742 C0B8h, GPTW1_NS.GTICLF 0742 C1B8h, GPTW2_NS.GTICLF 0742 C2B8h, GPTW3_NS.GTICLF 0742 C3B8h, GPTW4_NS.GTICLF 0742 C4B8h, GPTW5_NS.GTICLF 0742 C5B8h, GPTW6_NS.GTICLF 0742 C6B8h, GPTW7_NS.GTICLF 0742 C7B8h, GPTW8_NS.GTICLF 0742 C8B8h, GPTW9_NS.GTICLF 0742 C9B8h, GPTW10_NS.GTICLF 0742 CAB8h, GPTW11_NS.GTICLF 0742 CBB8h, GPTW12_NS.GTICLF 0742 CCB8h, GPTW13_NS.GTICLF 0742 CDB8h, GPTW14_NS.GTICLF 0742 CEB8h, GPTW15_NS.GTICLF 0742 CFB8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	ICLFSELD[5:0]						—	ICLFB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ICLFSELC[5:0]						—	ICLFA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select	b2 b0 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1 GTCLK delay) 0 1 1: NOT C (1 GTCLK delay) 1 0 0: A AND C (1 GTCLK delay)*2 1 0 1: A OR C (1 GTCLK delay)*2 1 1 0: A EXOR C (1 GTCLK delay)*2 1 1 1: A NOR C (1 GTCLK delay)*2	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9 to b4	ICLFSELC[5:0]	Inter Channel Signal C Select*1*2	b9 b4 0 0 0 0 0 0: GTIOC0A 0 0 0 0 0 1: GTIOC0B 0 0 0 0 1 0: GTIOC1A 0 0 0 0 1 1: GTIOC1B 0 0 0 1 0 0: GTIOC2A 0 0 0 1 0 1: GTIOC2B 0 0 0 1 1 0: GTIOC3A 0 0 0 1 1 1: GTIOC3B 0 0 1 0 0 0: GTIOC4A 0 0 1 0 0 1: GTIOC4B 0 0 1 0 1 0: GTIOC5A 0 0 1 0 1 1: GTIOC5B 0 0 1 1 0 0: GTIOC6A 0 0 1 1 0 1: GTIOC6B 0 0 1 1 1 0: GTIOC7A 0 0 1 1 1 1: GTIOC7B 0 1 0 0 0 0: GTIOC8A 0 1 0 0 0 1: GTIOC8B 0 1 0 0 1 0: GTIOC9A 0 1 0 0 1 1: GTIOC9B 0 1 0 1 0 0: GTIOC10A 0 1 0 1 0 1: GTIOC10B 0 1 0 1 1 0: GTIOC11A 0 1 0 1 1 1: GTIOC11B 0 1 1 0 0 0: GTIOC12A 0 1 1 0 0 1: GTIOC12B 0 1 1 0 1 0: GTIOC13A 0 1 1 0 1 1: GTIOC13B 0 1 1 1 0 0: GTIOC14A 0 1 1 1 0 1: GTIOC14B 0 1 1 1 1 0: GTIOC15A 0 1 1 1 1 1: GTIOC15B Settings other than above are prohibited.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select	b18 b16 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1 GTCLK delay) 0 1 1: NOT D (1 GTCLK delay) 1 0 0: B AND D (1 GTCLK delay)*3 1 0 1: B OR D (1 GTCLK delay)*3 1 1 0: B EXOR D (1 GTCLK delay)*3 1 1 1: B NOR D (1 GTCLK delay)*3	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b25 to b20	ICLFSELD[5:0]	Inter Channel Signal D Select*1*3	b25 b20 / b25 b20 0 0 0 0 0 0: GTIOC0A / 0 1 0 0 0 0: GTIOC8A 0 0 0 0 0 1: GTIOC0B / 0 1 0 0 0 1: GTIOC8B 0 0 0 0 1 0: GTIOC1A / 0 1 0 0 1 0: GTIOC9A 0 0 0 0 1 1: GTIOC1B / 0 1 0 0 1 1: GTIOC9B 0 0 0 1 0 0: GTIOC2A / 0 1 0 1 0 0: GTIOC10A 0 0 0 1 0 1: GTIOC2B / 0 1 0 1 0 1: GTIOC10B 0 0 0 1 1 0: GTIOC3A / 0 1 0 1 1 0: GTIOC11A 0 0 0 1 1 1: GTIOC3B / 0 1 0 1 1 1: GTIOC11B 0 0 1 0 0 0: GTIOC4A / 0 1 1 0 0 0: GTIOC12A 0 0 1 0 0 1: GTIOC4B / 0 1 1 0 0 1: GTIOC12B 0 0 1 0 1 0: GTIOC5A / 0 1 1 0 1 0: GTIOC13A 0 0 1 0 1 1: GTIOC5B / 0 1 1 0 1 1: GTIOC13B 0 0 1 1 0 0: GTIOC6A / 0 1 1 1 0 0: GTIOC14A 0 0 1 1 0 1: GTIOC6B / 0 1 1 1 0 1: GTIOC14B 0 0 1 1 1 0: GTIOC7A / 0 1 1 1 1 0: GTIOC15A 0 0 1 1 1 1: GTIOC7B / 0 1 1 1 1 1: GTIOC15B Settings other than above are prohibited.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The signal before performing output disable control is selected.
Note 2. When channel's own GTIOCnA is selected, C is treated as 1.
Note 3. When channel's own GTIOCnB is selected, D is treated as 1.

The GTICLF register sets the logical operation function between compare match outputs.
The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control.
The output disable control is performed with the signal after logical operation.
Access in 8-bit units to the GTICLF register is prohibited.

ICLFm[2:0] Bits (GTIOCnm Output Logical Operation Function Select) (n = 0 to 15; m = A, B)

These bits select the logical operation function between signals before performing output disable control for GTIOCnm. Logical operations involving the C or D signal take an additional clock cycle of GTCLK. When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as 1.

ICLFSELk[5:0] Bits (Inter Channel Signal k Select) (k = C, D)

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

24.2.41 General PWM Timer Cycle Count Register (GTPC)

Address(es): GPTW0.GTPC 8742 C0BCh, GPTW1.GTPC 8742 C1BCh, GPTW2.GTPC 8742 C2BCh, GPTW3.GTPC 8742 C3BCh, GPTW4.GTPC 8742 C4BCh, GPTW5.GTPC 8742 C5BCh, GPTW6.GTPC 8742 C6BCh, GPTW7.GTPC 8742 C7BCh, GPTW8.GTPC 8742 C8BCh, GPTW9.GTPC 8742 C9BCh, GPTW10.GTPC 8742 CABCh, GPTW11.GTPC 8742 CBBCh, GPTW12.GTPC 8742 CCBCh, GPTW13.GTPC 8742 CDBCh, GPTW14.GTPC 8742 CEBCh, GPTW15.GTPC 8742 CFBCh, GPTW0_NS.GTPC 0742 C0BCh, GPTW1_NS.GTPC 0742 C1BCh, GPTW2_NS.GTPC 0742 C2BCh, GPTW3_NS.GTPC 0742 C3BCh, GPTW4_NS.GTPC 0742 C4BCh, GPTW5_NS.GTPC 0742 C5BCh, GPTW6_NS.GTPC 0742 C6BCh, GPTW7_NS.GTPC 0742 C7BCh, GPTW8_NS.GTPC 0742 C8BCh, GPTW9_NS.GTPC 0742 C9BCh, GPTW10_NS.GTPC 0742 CABCh, GPTW11_NS.GTPC 0742 CBBCh, GPTW12_NS.GTPC 0742 CCBCh, GPTW13_NS.GTPC 0742 CDBCh, GPTW14_NS.GTPC 0742 CEBCh, GPTW15_NS.GTPC 0742 CFBCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ASTP	—	—	—	—	—	—	—	PCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCEN	Cycle Count Function Enable	0: Cycle count function is disabled. 1: Cycle count function is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ASTP	Automatic Stop Function Enable	0: Automatic function is disabled. 1: Automatic function is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	PCNT[11:0]	Cycle Counter	Counter for number of cycles.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN Bit (Cycle Count Function Enable)

This bit enables or disables cycle count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP Bit (Automatic Stop Function Enable)

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] Bits (Cycle Counter)

This counter counts the number of cycles.

When the PCEN bis is 0, writing the number of cycles is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of cycle. In sawtooth-wave mode, the end of cycle refers to overflow, underflow, or GTCNT counter clearing. In triangle-wave mode or complementary PWM mode, it refers to trough.

When the PCNT counter is 1 at the end of cycle, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while cycle count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value. When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

24.2.42 General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register (GTADCMSC)

Address(es): GPTW4.GTADCMSC 8742 C4C0h, GPTW5.GTADCMSC 8742 C5C0h, GPTW6.GTADCMSC 8742 C6C0h, GPTW7.GTADCMSC 8742 C7C0h, GPTW8.GTADCMSC 8742 C8C0h, GPTW9.GTADCMSC 8742 C9C0h, GPTW4_NS.GTADCMSC 0742 C4C0h, GPTW5_NS.GTADCMSC 0742 C5C0h, GPTW6_NS.GTADCMSC 0742 C6C0h, GPTW7_NS.GTADCMSC 0742 C7C0h, GPTW8_NS.GTADCMSC 0742 C8C0h, GPTW9_NS.GTADCMSC 0742 C9C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADCMSCNT2[3:0]				ADCMSCNT2IV[3:0]				ADCMST2[3:0]				—	—	ADCMSC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADCMSCNT1[3:0]				ADCMSCNT1IV[3:0]				ADCMST1[3:0]				—	—	ADCMSC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADCMSC1[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Count Source Select	b1 b0 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	ADCMST1[3:0]	A/D Conversion Start Request Compare Match Skipping 1 Skipping Count Setting	Skipping count for the A/D conversion start request compare match skipping 1.	R/W
b11 to b8	ADCMSCNT1IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	R/W *1
b15 to b12	ADCMSCNT1[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1	A/D Conversion Start Request Compare Match Skipping Counter 1	R/W
b17, b16	ADCMSC2[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Count Source Select	b17 b16 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	ADCMST2[3:0]	A/D Conversion Start Request Compare Match Skipping 2 Skipping Count Setting	Skipping count for the A/D conversion start request compare match skipping 2	R/W
b27 to b24	ADCMSCNT2IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	R/W *2
b31 to b28	ADCMSCNT2[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2	A/D Conversion Start Request Compare Match Skipping Counter 2	R/W

Note 1. Writing is possible only when the ADCMSC1[1:0] bits are 00b and a value other than 00b is written to the ADCMSC1[1:0] bits.
Note 2. Writing is possible only when the ADCMSC2[1:0] bits are 00b and a value other than 00b is written to the ADCMSC2[1:0] bits.

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.
This register setting is operated independently of the interrupt skipping by the GTITC register or GTEITC register.
The setting is invalid during the event count operation.

Access in 8-bit units to GTADCMSC is prohibited.

ADCMSCk[1:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the A/D conversion start request compare match skipping counter k.
Setting only with these bits does not skip the A/D conversion start requests and buffer transfers. Skipping function for the A/D conversion start request and buffer transfer which are a target of skipping is set individually with the GTADCMSS register.

ADCMSTk[3:0] Bits (A/D Conversion Start Request Compare Match Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the ADCMSCk[1:0] bits to the next generation of the count source.
When the count source is generated while the ADCMSTk[3:0] bits match the ADCMSCNTk[3:0] bits, the ADCMSCNTk[3:0] bits are cleared.
When these bits are 0000b, skipping is not performed.

ADCMSCNTkIV[3:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k Initial Value) (k = 1, 2)

These bits are the value of the initial value for the A/D conversion start request compare match skipping counter k.
Writing to the ADCMSCNTkIV[3:0] is performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the ADCMSCNTk[3:0] bits are set not to count (ADCMSCk[1:0] bits are 00b). When the ADCMSCNTkIV[3:0] bits are written, the value written to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits simultaneously.
The writing to the ADCMSCNTkIV[3:0] bits are ignored when the ADCMSCNTk[3:0] bits are set to count (ADCMSCk[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the ADCMSCk[1:0] bits).
The ADCMSCNTkIV[3:0] bits are not reset by the writing 00b to the ADCMSCk[1:0] bits.

ADCMSCNTk[3:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k) (k = 1, 2)

The counting is incremented by 1 every time a count source selected by the ADCMSCk[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the ADCMSTk[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
Setting of the initial value for the ADCMSCNTk[3:0] bits are performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the A/D conversion start request compare match skipping counter k is set as not to count (ADCMSCk[1:0] bits are 00b).
When the initial value is set, the written value to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits as the initial value.
[Clearing condition]

- 00b is written to the ADCMSCk[1:0] bits.
- 0000b is written to the ADCMSTk[3:0] bits.
- When the ADCMSCk[1:0] bits are 00b, a value other than 00b is written to the ADCMSCk[1:0] bits and 0000b is written to the ADCMSCNTkIV[3:0] bits at the same time.
- A count source selected by the ADCMSCk[1:0] bits is generated when the extended interrupt skipping k skipping count set by the ADCMSTk[3:0] bits match the value for the ADCMSCNTk[3:0].

24.2.43 General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register (GTADCMSS)

Address(es): GPTW4.GTADCMSS 8742 C4C4h, GPTW5.GTADCMSS 8742 C5C4h, GPTW6.GTADCMSS 8742 C6C4h, GPTW7.GTADCMSS 8742 C7C4h, GPTW8.GTADCMSS 8742 C8C4h, GPTW9.GTADCMSS 8742 C9C4h, GPTW4_NS.GTADCMSS 0742 C4C4h, GPTW5_NS.GTADCMSS 0742 C5C4h, GPTW6_NS.GTADCMSS 0742 C6C4h, GPTW7_NS.GTADCMSS 0742 C7C4h, GPTW8_NS.GTADCMSS 0742 C8C4h, GPTW9_NS.GTADCMSS 0742 C9C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	ADCMBSB[2:0]			—	ADCMBSA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ADCMSBL[2:0]			—	ADCMSAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADCMSAL[2:0]	GTADTRA Register A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.9.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	ADCMSBL[2:0]	GTADTRB Register A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.9.	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	ADCMBSA[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.10.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	ADCMBSB[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.10.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTADCMSS register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.
This register setting is operated independently of the interrupt skipping by the GTITC register or GTEITC register.
The setting is invalid during the event count operation.

ADCMSmL[2:0] Bits (GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request compare match skipping function of the GTADTRm (m = A, B) register.
Refer to Table 24.9.

Table 24.9 Setting of GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit

ADCMSmL[2:0] Bits	Function
0 0 0	Do not perform an A/D conversion start request compare match skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0000b)
0 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = 0000b)
0 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0000b and the ADCMSCNT2[3:0] bits = 0000b)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

m = A, B

Note: When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0000b), skipping is not performed. (k = 1, 2)

Note: When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counter1 or the skipping counter 2 is set as not to count, skipping is not performed.

ADCMBSm[2:0] Bits (GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request comparison match skip function for buffer transfers in the GTADTRm register (transfers between the GTADTRm register, GTADTBRm register, and GTADTDBRm register).
Refer to Table 24.10.

Table 24.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit

ADCMBSm[2:0] Bits	Function
0 0 0	Do not perform an GTADTRm register buffer transfer by A/D conversion start request compare match skipping
0 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0000b)
0 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = 0000b)
0 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0000b and the ADCMSCNT2[3:0] bits = 0000b)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (A Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

m = A, B

Note: When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0000b), skipping is not performed. (k = 1, 2)

Note: When the ADCMBSm[2:0] bits are set to 011b or 111b, and when one of the skipping counter1 or the skipping counter 2 is set as not to count, skipping is not performed.

24.2.44 General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register (GTSECSR)

Address(es): GPTW0.GTSECSR 8742 C0D0h, GPTW1.GTSECSR 8742 C1D0h, GPTW2.GTSECSR 8742 C2D0h, GPTW3.GTSECSR 8742 C3D0h, GPTW4.GTSECSR 8742 C4D0h, GPTW5.GTSECSR 8742 C5D0h, GPTW6.GTSECSR 8742 C6D0h, GPTW7.GTSECSR 8742 C7D0h, GPTW8.GTSECSR 8742 C8D0h, GPTW9.GTSECSR 8742 C9D0h, GPTW10.GTSECSR 8742 CAD0h, GPTW11.GTSECSR 8742 CBD0h, GPTW12.GTSECSR 8742 CCD0h, GPTW13.GTSECSR 8742 CDD0h, GPTW14.GTSECSR 8742 CED0h, GPTW15.GTSECSR 8742 CFD0h, GPTW0_NS.GTSECSR 0742 C0D0h, GPTW1_NS.GTSECSR 0742 C1D0h, GPTW2_NS.GTSECSR 0742 C2D0h, GPTW3_NS.GTSECSR 0742 C3D0h, GPTW4_NS.GTSECSR 0742 C4D0h, GPTW5_NS.GTSECSR 0742 C5D0h, GPTW6_NS.GTSECSR 0742 C6D0h, GPTW7_NS.GTSECSR 0742 C7D0h, GPTW8_NS.GTSECSR 0742 C8D0h, GPTW9_NS.GTSECSR 0742 C9D0h, GPTW10_NS.GTSECSR 0742 CAD0h, GPTW11_NS.GTSECSR 0742 CBD0h, GPTW12_NS.GTSECSR 0742 CCD0h, GPTW13_NS.GTSECSR 0742 CDD0h, GPTW14_NS.GTSECSR 0742 CED0h, GPTW15_NS.GTSECSR 0742 CFD0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECSEL15	SECSEL14	SECSEL13	SECSEL12	SECSEL11	SECSEL10	SECSEL9	SECSEL8	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select	0: Disable simultaneous control. 1: Enable simultaneous control.	R/W
b1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select		R/W
b2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select		R/W
b3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select		R/W
b4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select		R/W
b5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select		R/W
b6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select		R/W
b7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select		R/W
b8	SECSEL8	Channel 8 Operation Enable Bit Simultaneous Control Channel Select		R/W
b9	SECSEL9	Channel 9 Operation Enable Bit Simultaneous Control Channel Select		R/W
b10	SECSEL10	Channel 10 Operation Enable Bit Simultaneous Control Channel Select		R/W
b11	SECSEL11	Channel 11 Operation Enable Bit Simultaneous Control Channel Select		R/W
b12	SECSEL12	Channel 12 Operation Enable Bit Simultaneous Control Channel Select		R/W
b13	SECSEL13	Channel 13 Operation Enable Bit Simultaneous Control Channel Select		R/W
b14	SECSEL14	Channel 14 Operation Enable Bit Simultaneous Control Channel Select		R/W
b15	SECSEL15	Channel 15 Operation Enable Bit Simultaneous Control Channel Select		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects the channels whose operation enable bit is to be updated using the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.
Access in 8-bit or 16-bit units to the GTSECSR register is prohibited, and it should be accessed in 32-bit units.

SECSELn Bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 15)

This bit enables or disables the simultaneous control of operation enable in channel n.
When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

24.2.45 General PWM Timer Operation Enable Bit Simultaneous Control Register (GTSECR)

Address(es): GPTW0.GTSECR 8742 C0D4h, GPTW1.GTSECR 8742 C1D4h, GPTW2.GTSECR 8742 C2D4h, GPTW3.GTSECR 8742 C3D4h, GPTW4.GTSECR 8742 C4D4h, GPTW5.GTSECR 8742 C5D4h, GPTW6.GTSECR 8742 C6D4h, GPTW7.GTSECR 8742 C7D4h, GPTW8.GTSECR 8742 C8D4h, GPTW9.GTSECR 8742 C9D4h, GPTW10.GTSECR 8742 CAD4h, GPTW11.GTSECR 8742 CBD4h, GPTW12.GTSECR 8742 CCD4h, GPTW13.GTSECR 8742 CDD4h, GPTW14.GTSECR 8742 CED4h, GPTW15.GTSECR 8742 CFD4h, GPTW0_NS.GTSECR 0742 C0D4h, GPTW1_NS.GTSECR 0742 C1D4h, GPTW2_NS.GTSECR 0742 C2D4h, GPTW3_NS.GTSECR 0742 C3D4h, GPTW4_NS.GTSECR 0742 C4D4h, GPTW5_NS.GTSECR 0742 C5D4h, GPTW6_NS.GTSECR 0742 C6D4h, GPTW7_NS.GTSECR 0742 C7D4h, GPTW8_NS.GTSECR 0742 C8D4h, GPTW9_NS.GTSECR 0742 C9D4h, GPTW10_NS.GTSECR 0742 CAD4h, GPTW11_NS.GTSECR 0742 CBD4h, GPTW12_NS.GTSECR 0742 CCD4h, GPTW13_NS.GTSECR 0742 CDD4h, GPTW14_NS.GTSECR 0742 CED4h, GPTW15_NS.GTSECR 0742 CFD4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SSCD	SPCD	—	—	—	—	—	—	SSCE	SPCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
b1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W *1
b2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
b3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
b9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W *1
b10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
b11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SPCE	Cycle Count Function Simultaneous Enable	0: Disable simultaneous enabling cycle count function 1: Enable cycle count function simultaneously	R/W
b17	SSCE	Synchronous Set/Clear Simultaneous Enable	0: Disable simultaneous enabling synchronous set/clear 1: Enable synchronous set/clear simultaneously	R/W *1
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	SPCD	Cycle Count Function Simultaneous Disable	0: Disable simultaneous disabling cycle count function 1: Disable cycle count function simultaneously	R/W
b25	SSCD	Synchronous Set/Clear Simultaneous Disable	0: Disable simultaneous disabling synchronous set/clear 1: Disable synchronous set/clear simultaneously	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers. Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.
A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.
Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE Bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE Bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are enabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE Bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDE Bit (GTDV Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are enabled.
Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SBDCD Bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD Bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are disabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD Bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDD Bit (GTDV Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR

register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are disabled.
Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SPCE Bit (Cycle Count Function Simultaneous Enable)

Writing 1 to this bit simultaneously sets any GTPC.PCEN bits to 1 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and enables the cycle-counting function.
Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCE Bit (Synchronous Set/Clear Simultaneous Enable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 1 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and enables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SPCD Bit (Cycle Count Function Simultaneous Disable)

Writing 1 to this bit simultaneously sets any GTPC.PCEN bits to 0 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and disables the cycle-counting function.
Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCD Bit (Synchronous Set/Clear Simultaneous Disable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 0 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and disables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

24.2.46 General PWM Timer Output Level Buffer Register (GTOLBR)

Address(es): GPTW4.GTOLBR 8742 C4E4h, GPTW5.GTOLBR 8742 C5E4h, GPTW6.GTOLBR 8742 C6E4h, GPTW7.GTOLBR 8742 C7E4h, GPTW8.GTOLBR 8742 C8E4h, GPTW9.GTOLBR 8742 C9E4h, GPTW4_NS.GTOLBR 0742 C4E4h, GPTW5_NS.GTOLBR 0742 C5E4h, GPTW6_NS.GTOLBR 0742 C6E4h, GPTW7_NS.GTOLBR 0742 C7E4h, GPTW8_NS.GTOLBR 0742 C8E4h, GPTW9_NS.GTOLBR 0742 C9E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	GTIOBB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	GTIOAB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	GTIOAB[4:0]	GTIOA Buffer	GTIOA[4:0] Register Buffer	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	GTIOBB[4:0]	GTIOB Buffer	GTIOB[4:0] Register Buffer	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTOLBR register is a buffer register for the GTIOR.GTIOA[4:0] bits and GTIOR.GTIOB[4:0] bits.

GTIOmB[4:0] Bits (GTIOm Buffer) (m = A, B)

These bits are buffer bits of GTIOR.GTIOm[4:0] bits.

These bits are transferred to the GTIOR.GTIOm[4:0] bits at the transfer timing selected by the GTBER2.OLTTm[1:0] (m = A, B) bits.

24.2.47 General PWM Timer Inter Channel Cooperation Input Capture Control Register (GTICCR)

Address(es): GPTW0.GTICCR 8742 C0ECh, GPTW1.GTICCR 8742 C1ECh, GPTW2.GTICCR 8742 C2ECh, GPTW3.GTICCR 8742 C3ECh, GPTW4.GTICCR 8742 C4ECh, GPTW5.GTICCR 8742 C5ECh, GPTW6.GTICCR 8742 C6ECh, GPTW7.GTICCR 8742 C7ECh, GPTW8.GTICCR 8742 C8ECh, GPTW9.GTICCR 8742 C9ECh, GPTW10.GTICCR 8742 CAECh, GPTW11.GTICCR 8742 CBECh, GPTW12.GTICCR 8742 CCECh, GPTW13.GTICCR 8742 CDECh, GPTW14.GTICCR 8742 CEECh, GPTW15.GTICCR 8742 CFECh, GPTW0_NS.GTICCR 0742 C0ECh, GPTW1_NS.GTICCR 0742 C1ECh, GPTW2_NS.GTICCR 0742 C2ECh, GPTW3_NS.GTICCR 0742 C3ECh, GPTW4_NS.GTICCR 0742 C4ECh, GPTW5_NS.GTICCR 0742 C5ECh, GPTW6_NS.GTICCR 0742 C6ECh, GPTW7_NS.GTICCR 0742 C7ECh, GPTW8_NS.GTICCR 0742 C8ECh, GPTW9_NS.GTICCR 0742 C9ECh, GPTW10_NS.GTICCR 0742 CAECh, GPTW11_NS.GTICCR 0742 CBECh, GPTW12_NS.GTICCR 0742 CCECh, GPTW13_NS.GTICCR 0742 CDECh, GPTW14_NS.GTICCR 0742 CEECh, GPTW15_NS.GTICCR 0742 CFECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICAFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b1	ICAFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b2	ICAFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels	R/W
b3	ICAFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels	R/W
b4	ICAFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels	R/W
b5	ICAFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels	R/W
b6	ICAFPO	Enable Overflow as Source of Other Channel GTCCRA Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W

Bit	Symbol	Bit Name	Description	R/W
b7	ICAFPU	Enable Underflow as Source of Other Channel GTCCRA Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W
b8	ICACLK	Enable Count Clock as Source of Other Channel GTCCRA Input Capture	0: Disable count clock, which are the GTCCRA input capture source of other channels 1: Enable count clock, which are the GTCCRA input capture source of other channels	R/W
b13 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ICAGRP[1:0]	GTCCRA Input Capture Group Select	b15b14 0 0: Group A 0 1: Group B 1 0: Group C 1 1: Group D	R/W
b16	ICBFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b17	ICBFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b18	ICBFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels	R/W
b19	ICBFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels	R/W
b20	ICBFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels	R/W
b21	ICBFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels	R/W
b22	ICBFPO	Enable Overflow as Source of Other Channel GTCCRB Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b23	ICBFPU	Enable Underflow as Source of Other Channel GTCCRB Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b24	ICBCLK	Enable Count Clock as Source of Other Channel GTCCRB Input Capture	0: Disable count clock, which are the GTCCRB input capture source of other channels 1: Enable count clock, which are the GTCCRB input capture source of other channels	R/W
b29 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	ICBGRP[1:0]	GTCCRB Input Capture Group Select	b31b30 0 0: Group A 0 1: Group B 1 0: Group C 1 1: Group D	R/W

The GTICCR register is a register that controls input capture by inter channel link.
For channels that perform input capture in cooperation, the input capture factor set in the GTCCRA and GTCCRB registers is invalid.

ICAFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture) (m = A, B)

This bit enables or disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRA Input Capture) (n = C, D, E, F)

This bit enables or disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRA register.

ICAFPO Bit (Enable Overflow as Source of Other Channel GTCCRA Input Capture)

This bit enables or disables the use of overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU Bit (Enable Underflow as Source of Other Channel GTCCRA Input Capture)

This bit enables or disables the use of underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK Bit (Enable Count Clock as Source of Other Channel GTCCRA Input Capture)

This bit enables or disables the use of count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] Bits (GTCCRA Input Capture Group Select)

These bits select the group of input capture by inter channel cooperation for GTCCRA register.
For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture) (m = A, B)

This bit enables or disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRB Input Capture) (n = C, D, E, F)

This bit enables or disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRB register.

ICBFPO Bit (Enable Overflow as Source of Other Channel GTCCRB Input Capture)

This bit enables or disables the use of overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM

mode as the input capture source of other channel's GTCCRB register.

ICBFPU Bit (Enable Underflow as Source of Other Channel GTCCRB Input Capture)

This bit enables or disables the use of underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK Bit (Enable Count Clock as Source of Other Channel GTCCRB Input Capture)

This bit enables or disables the use of count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] Bits (GTCCRB Input Capture Group Select)

These buts select the group of input capture by inter channel cooperation for GTCCRB register.
For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

24.2.48 General PWM Timer-linked Synchronization Control Register (GTSYNCFCR)

Address(es): GPTW0.GTSYNCFCR 8742 C0F0h, GPTW1.GTSYNCFCR 8742 C1F0h, GPTW2.GTSYNCFCR 8742 C2F0h, GPTW3.GTSYNCFCR 8742 C3F0h, GPTW4.GTSYNCFCR 8742 C4F0h, GPTW5.GTSYNCFCR 8742 C5F0h, GPTW6.GTSYNCFCR 8742 C6F0h, GPTW7.GTSYNCFCR 8742 C7F0h, GPTW8.GTSYNCFCR 8742 C8F0h, GPTW9.GTSYNCFCR 8742 C9F0h, GPTW10.GTSYNCFCR 8742 CAF0h, GPTW11.GTSYNCFCR 8742 CBF0h, GPTW12.GTSYNCFCR 8742 CCF0h, GPTW13.GTSYNCFCR 8742 CDF0h, GPTW14.GTSYNCFCR 8742 CEF0h, GPTW0_NS.GTSYNCFCR 0742 C0F0h, GPTW1_NS.GTSYNCFCR 0742 C1F0h, GPTW2_NS.GTSYNCFCR 0742 C2F0h, GPTW3_NS.GTSYNCFCR 0742 C3F0h, GPTW4_NS.GTSYNCFCR 0742 C4F0h, GPTW5_NS.GTSYNCFCR 0742 C5F0h, GPTW6_NS.GTSYNCFCR 0742 C6F0h, GPTW7_NS.GTSYNCFCR 0742 C7F0h, GPTW8_NS.GTSYNCFCR 0742 C8F0h, GPTW9_NS.GTSYNCFCR 0742 C9F0h, GPTW10_NS.GTSYNCFCR 0742 CAF0h, GPTW11_NS.GTSYNCFCR 0742 CBF0h, GPTW12_NS.GTSYNCFCR 0742 CCF0h, GPTW13_NS.GTSYNCFCR 0742 CDF0h, GPTW14_NS.GTSYNCFCR 0742 CEF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SYST	Timer-Internal Synchronization Signal Synchronization Control	0: Disable starting, stopping, and clearing of the counter in synchronization with the timer-internal synchronization signal 1: Enable starting, stopping, and clearing of the counter in synchronization with the timer-internal synchronization signal	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSYNCFCR register is used to enable or disable starting, stopping, and clearing of the counter in synchronization with the timer-internal synchronization signal.

For the timer-internal synchronization signal and synchronous operation, refer to section 24.3.17, Timer-Linked Synchronization Signal Output Function.

24.2.49 General PWM Timer-linked Synchronization Signal Output Control Register (GTSYNCSCR)

Address(es): GPTW15.GTSYNCSCR 8742 CFF0h, GPTW15_NS.GTSYNCSCR 0742 CFF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYSG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SYSG[1:0]	Timer-Linked Synchronization Signal and Timer-Internal Synchronization Signal Output Enable	b1 b0 0 0: Disable output of the timer-linked synchronization signal or timer-internal synchronization signal 0 1: Enable output of the timer-linked synchronization signal or timer-internal synchronization signal Settings other than above are prohibited.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSYNCSCR register controls enabling and disabling of output of the timer-linked synchronization signal and the timer-internal synchronization signal.
For the timer-linked synchronization signal and the timer-internal synchronization signal, refer to section 24.3.17, Timer-Linked Synchronization Signal Output Function.

SYSG[1:0] Bits (Timer-Linked Synchronization Signal and Timer-Internal Synchronization Signal Output Enable)

These bits enable or disable timer-linked synchronization signal output in response to compare match and timer-internal synchronization signal output on overflow in sawtooth-wave PWM mode 1.

24.2.50 Output Phase Switching Control Register (OPSCR)

Address(es): GPTW.OPSCR 8742 BF00h, GPTW_NS.OPSCR 0742 BF00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Input Phase Software Setting	These bits set the input phase from software settings. Setting these bits is valid when FB = 1.	R/W
b1	VF			R/W
b2	WF			R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	U	Input U-Phase Monitor	This bit monitors the state of the input phase. • FB = 0 External input that are synchronized by GTCLK • FB = 1 The software settings (UF, VF, WF bit)	R
b5	V	Input V-Phase Monitor		R
b6	W	Input W-Phase Monitor		R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	EN	Output Phase Enable	0: Do not output (Hi-Z external pin) 1: Output*1	R/W
b15-b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from software settings and external input. 0: The external input 1: The software setting (UF, VF, WF bit)	R/W
b17	P	Positive-Phase Output (P) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b18	N	Negative-Phase Output (N) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b19	INV	Output Phase Invert Control	0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
b20	RV	Output Phase Rotation Direction Reversal Control	0: Positive rotation 1: Reverse rotation	R/W
b21	ALIGN	Input Phase Alignment	0: Input phase aligned to GTCLK 1: Input phase aligned to the falling edge of PWM	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disabled Source Selection	b25b24 0 0: Group A output disable source 0 1: Group B output disable source 1 0: Group C output disable source 1 1: Group D output disable source	R/W
b26	GODF	Group Output Disable Function	0: This bit function is ignored 1: When the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is cleared	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFEN	External Input Noise Filter Enable	0: The noise filter for the external input is disabled. 1: The noise filter for the external input is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	NFCS[1:0]	External Input Noise Filter Clock Select	b31b30 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. When the GODF bit is 1 and the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF, VF, WF Bits (Input Phase Software Setting)

The UF, VF, and WF bits set the input phase from the software settings. When the FB bit is 1, these bits are valid. The set value of the UF, VF, and WF bit takes the place of the U, V, W external input.

U, V, W Bits (Input m-Phase Monitor) (m = U, V, W)

When the FB bit is 0, external inputs that are synchronized by GTCLK are monitored by these bits. When the FB bit is 1, the U, V, and W bits can read the UF, VF, and WF bits.

EN Bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).
When the EN bit is 1, the signal waveform is output.
When the EN bit is 0, first set FB, UF, VF, and WF (software setting is selected), P, N, INV, RV, ALIGN, GRP[1:0], GODF, NFEN, and NFCS[1:0] bits. Then, set this bit to 1. The EN bit should be set when no output disable request is issued by POEG. When the GODF bit is 1 and the signal value selected in the GRP[1:0] bits is high, the EN bit is set to 0. To reenable output, clear the output request in software and then set the EN bit to 1.

- EN bit priority (conflict)

When a conflict occurs between setting the EN bit by software and clearing it due to an output disable request, the EN bit is set to 0.

FB Bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (UF, VF, and WF bits) and external input such as a Hall element.

P Bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N Bit (Negative-Phase Output (N) Control)

The P bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV Bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV Bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN Bit (Input Phase Alignment)

The ALIGN bit selects the GTCLK or PWM for the sampling of the input phase (input phase is specified in the FB bit).

When ALIGN bit is 0, input phase is aligned to GTCLK.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] Bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source.

Set the GRP[1:0] bits while the GODF bit is 0.

GODF Bit (Group Output Disable Function)

When the GODF bit is 1 and the signal selected by the GRP[1:0] bits is high, the EN bit is set to 0.

When the GODF bit is 0, this bit is ignored.

Set the GODF bit while there is no output disable request from POEG.

NFEN Bit (External Input Noise Filter Enable)

The NFEN bit enables or disables the external input noise filter.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by this bit switching.

NFCS[1:0] Bits (External Input Noise Filter Clock Select)

The NFCS[1:0] bits select the clock for the external input noise filter.

1. Set the NFCS[1:0] bits.
2. Wait for 2 cycles.
3. Set the EN bit to 1.

24.2.51 General PWM Timer Clock Control Register (GTCLKCR)

Address(es): GPTW.GTCLKCR 8742 BF10h, GPTW_NS.GTCLKCR 0742 BF10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BPEN	Synchronization Circuit Bypass Enable	0: The bus clock and GPTW counter reference clock are asynchronously in use. 1: The bus clock and GPTW counter reference clock are synchronously in use.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTCLKCR register controls supply of the clock to the GPTW module.
Set this register before the GPTW module is released from the module stop state after a reset. Changing the setting of the effective bit in the register is prohibited after release from the module stop state.

BPEN Bit (Synchronization Circuit Bypass Enable)

This bit enables or disables bypassing of the synchronization between the bus clock (PCLKA) and GPTW counter reference clock (GTCLK). Set this bit to 0 or 1 according to whether the asynchronous or synchronous clock is to be used as the GPTW counter reference clock, respectively.

24.3 Operation

24.3.1 Basic Operation

The timer in each channel performs periodic count operation by count clock or hardware source. The GTCNT counter performs up-counting, or down-counting. The timer period is controlled by the GTPR or GTCCRm (m = A to F) registers. When the GTCNT counter value matches the values for GTCCRA or GTCCRB registers, the GTIOCnA or GTIOCnB pin outputs (n = 0 to 15) can be changed respectively. Also, the GTCCRA or GTCCRB registers can be used as the input capture registers by hardware source. The GTCCRC and GTCCRD registers can be performed as buffer registers for GTCCRA register, and the GTCCRE and GTCCRF registers can be performed as buffer registers for GTCCRB register.

24.3.1.1 Counter Operation

(1) Count Start/Count Stop

The counter in each channel starts counting operation when the corresponding GTCR.CST bit is set to 1, and stops counting when the bit is set to 0.
CST bit value can be changed by the following sources:

- Writing to the GTCR register
- Writing 1 to the bit corresponding to the channel number for the GTSTR register while the GTSSR.CSTRT bit is 1
- Writing 1 to the bit corresponding to the channel number for the GTSTP register while the GTPSR.CSTOP bit is 1
- Hardware source specified by the GTSSR register
- Hardware source specified by the GTPSR register
- Completion of the cycle count operation while the GTPC.ASTP bit is 1.

(2) Periodic Count Operation (In Up-Counting by the Count Clock)

The counter in each channel starts up-counting when the corresponding GTCR.CST bit is set to 1 while the GTUPSR register and the GTDNSR register are 0000 0000h. When the value of the GTCNT counter changes from the GTPR register value to 0000 0000h (an overflow), or the GTCCRm (m = A to F) register value selected by the GTCSR.CSCMSC[2:0] bits changes to 0000 0000h in sawtooth-wave PWM mode 2, the GTST.TCFPO flag is set to 1, and if the setting of the GTINTAD.GTINTPR[0] bit is 1, a GTCIV interrupt request is also issued. After the GTCNT counter has overflowed, counting up is resumed from 0000 0000h.
Figure 24.6 shows an example of periodic count operation in up-counting by count clock.

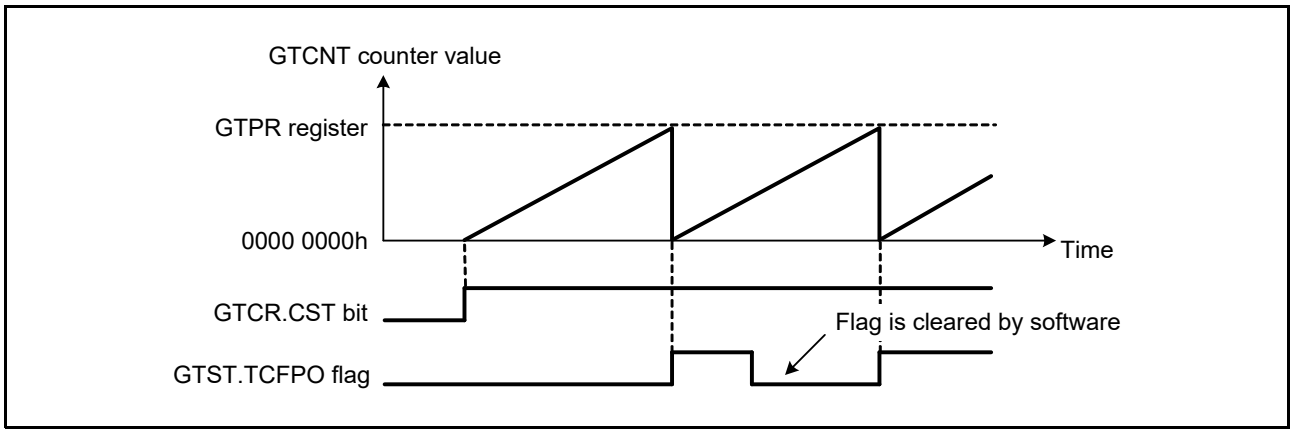


Figure 24.6 Example of Periodic Count Operation (In Up-Counting by the Count Clock)

Figure 24.7 shows an example for setting periodic count operation in up-counting.

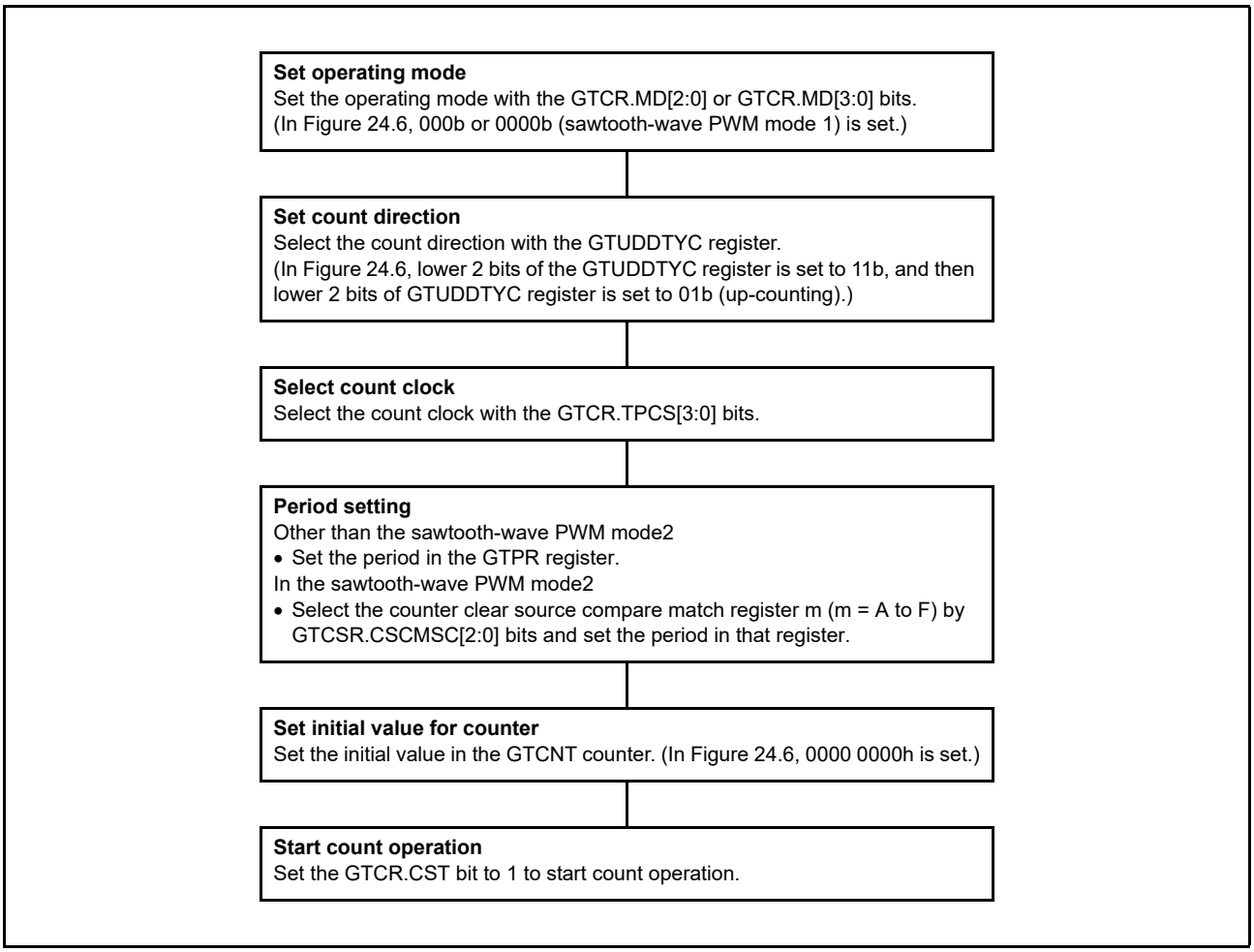


Figure 24.7 Example for Setting Periodic Count Operation (In Up-Counting by the Count Clock)

(3) Periodic Count Operation (In Down-Counting by the Count Clock)

The counter in each channel can start counting down when the corresponding GTUDDTYC.UD bit is set to 0 while the settings of the GTUPSR and GTDNSR registers are both 0000 0000h. When the value of the GTCNT counter changes from that of the GTPR register to 0000 0000h (an underflow), the GTST.TCFPU flag is set to 1, and if the setting of the GTINTAD.GTINTPR[1] bit is 1, a GTCIU interrupt request is also issued. After the GTCNT counter underflows, down-counting is resumed from the GTPR register value.

Figure 24.8 shows an example of periodic count operation in down-counting by count clock.

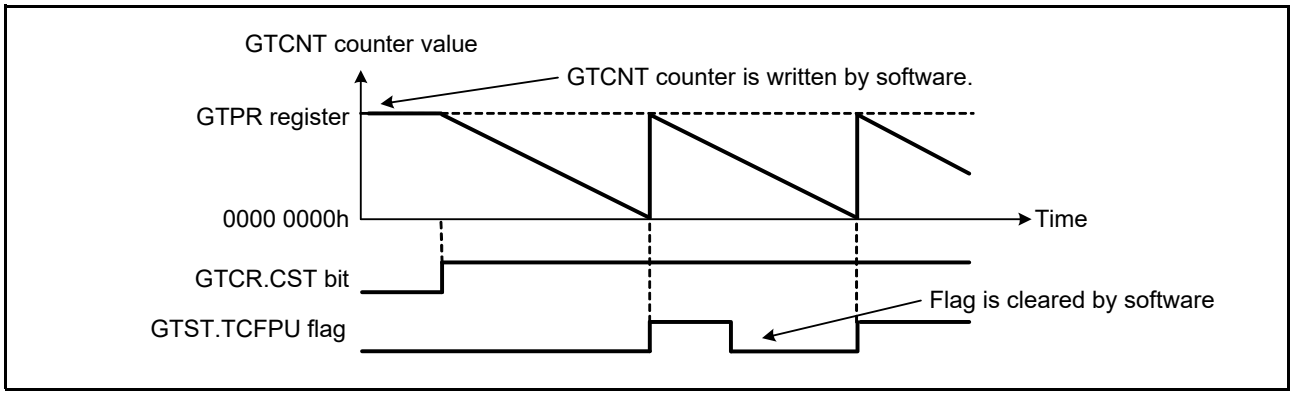


Figure 24.8 Example of Periodic Count Operation (In Down-Counting by the Count Clock)

Figure 24.9 shows an example for setting periodic count operation in down-counting by count clock.

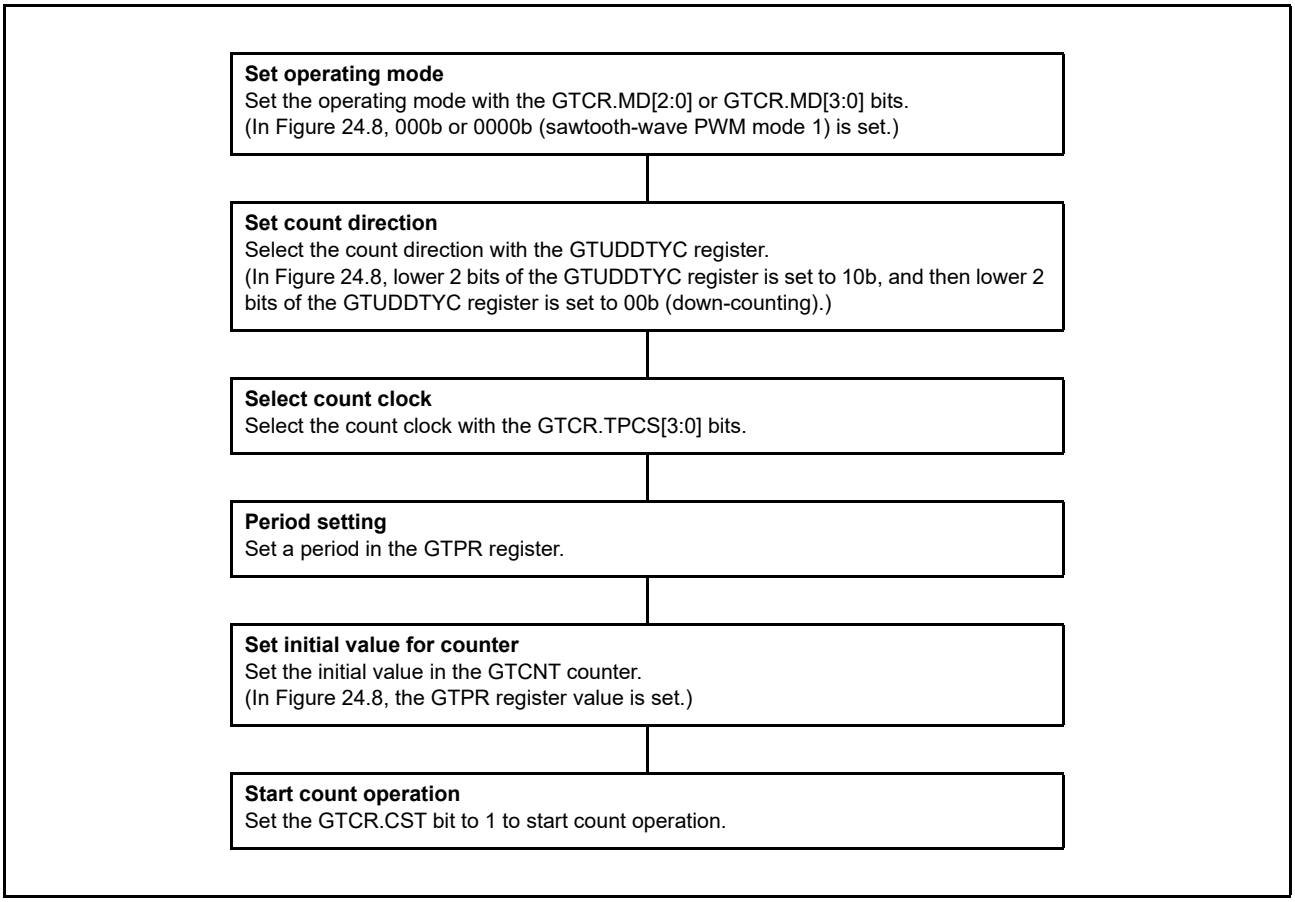


Figure 24.9 Example for Setting Periodic Count Operation (In Down-Counting by the Count Clock)

(4) Event Count Operation (In Up-Counting by Hardware Source)

The counter in each channel can start up-counting by hardware source by setting the GTUPSR register. When the GTUPSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to overflow in up-counting by hardware source is the same as the periodic count operation by the count clock.

To perform up-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start up-counting operation. To start up-counting after one GTCLK from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 24.10 and Figure 24.11 show examples of up-counting operation by hardware source (the rising edge of GTETRGA pin input and the rising edge of GTIOCnA pin input).

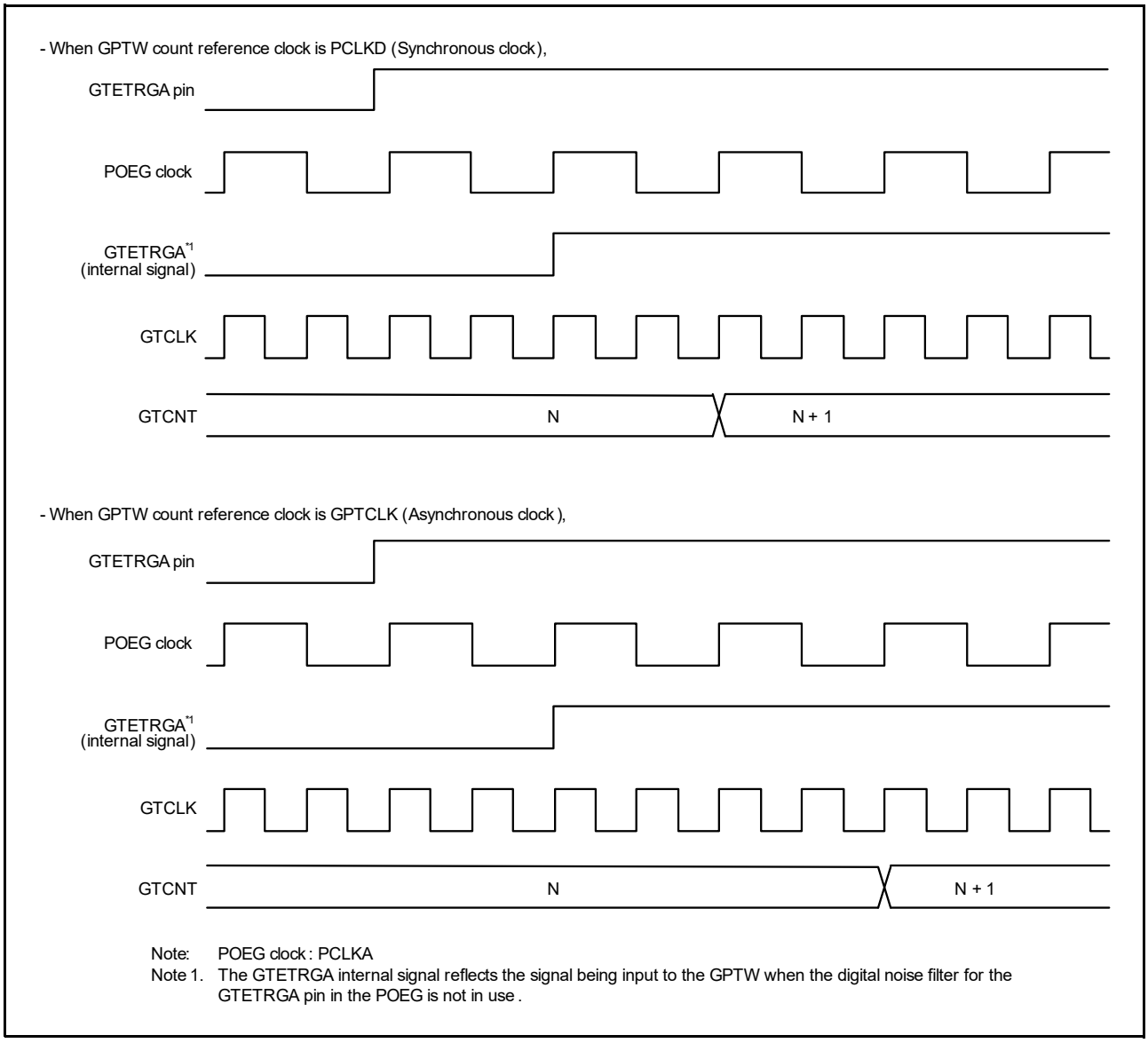


Figure 24.10 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTETRGA Pin)

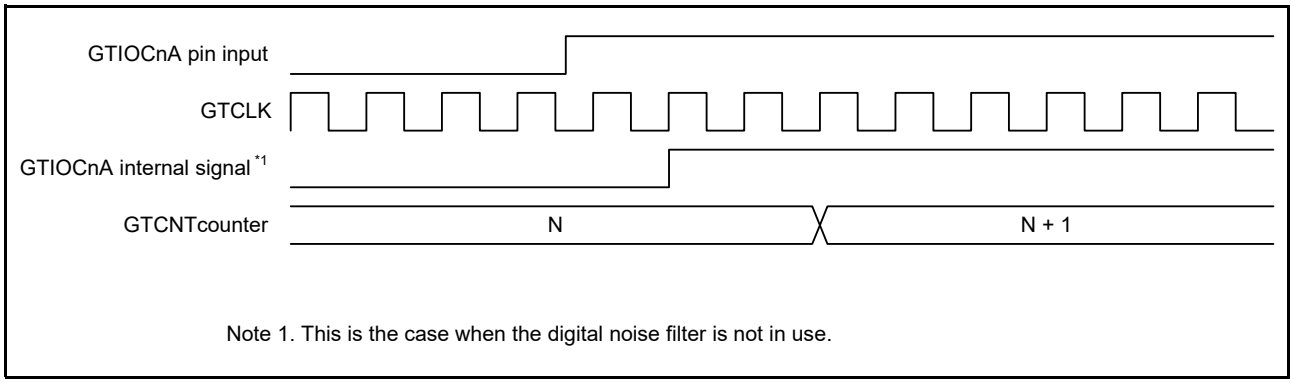


Figure 24.11 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTIOCnA Pin)

Figure 24.12 shows an example of event counting when the hardware source is an ELC event input.
This example shows the following signal flow:

- A compare match with the GPTW0.GTCCRA register outputs an event signal to the ELC.
- The ELC routes this signal to the GPTW as event source A, causing GPTW1.GTCNT to count events.

The ELC receives the GPTW0 compare match A signal, which is synchronized with GTCLK, and re-synchronizes it with PCLKB. After one PCLKB cycle, the ELC outputs the signal as GPTW event source A.

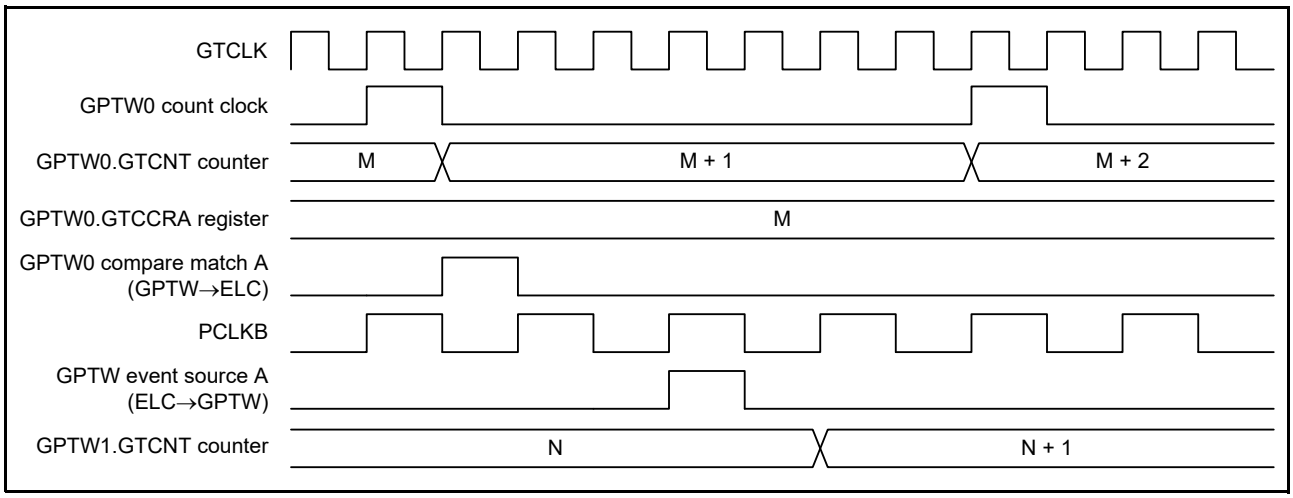


Figure 24.12 Example of Event Count Operation (Up-Counting the Number of Event Signals Input from the ELCA)

Figure 24.13 shows an example for setting event count operation in up-counting by hardware source.

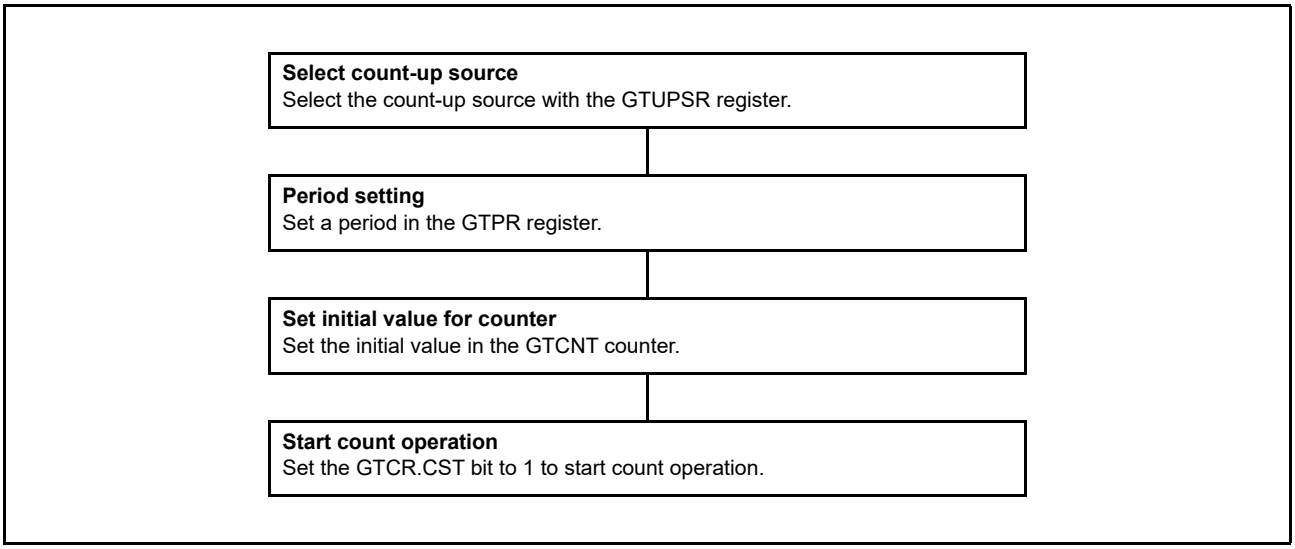


Figure 24.13 Example for Setting Event Count Operation (In Up-Counting by Hardware Source)

(5) Event Count Operation (In Down-Counting by Hardware Source)

The counter in each channel can start down-counting by hardware source by setting the GTDNSR register. When the GTDNSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to underflow in down-counting by hardware source is the same as the periodic count operation by the count clock.

To perform down-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start down-counting operation. To start down-counting after one GTCLK from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 24.14 shows an example of down-counting operation by hardware source.

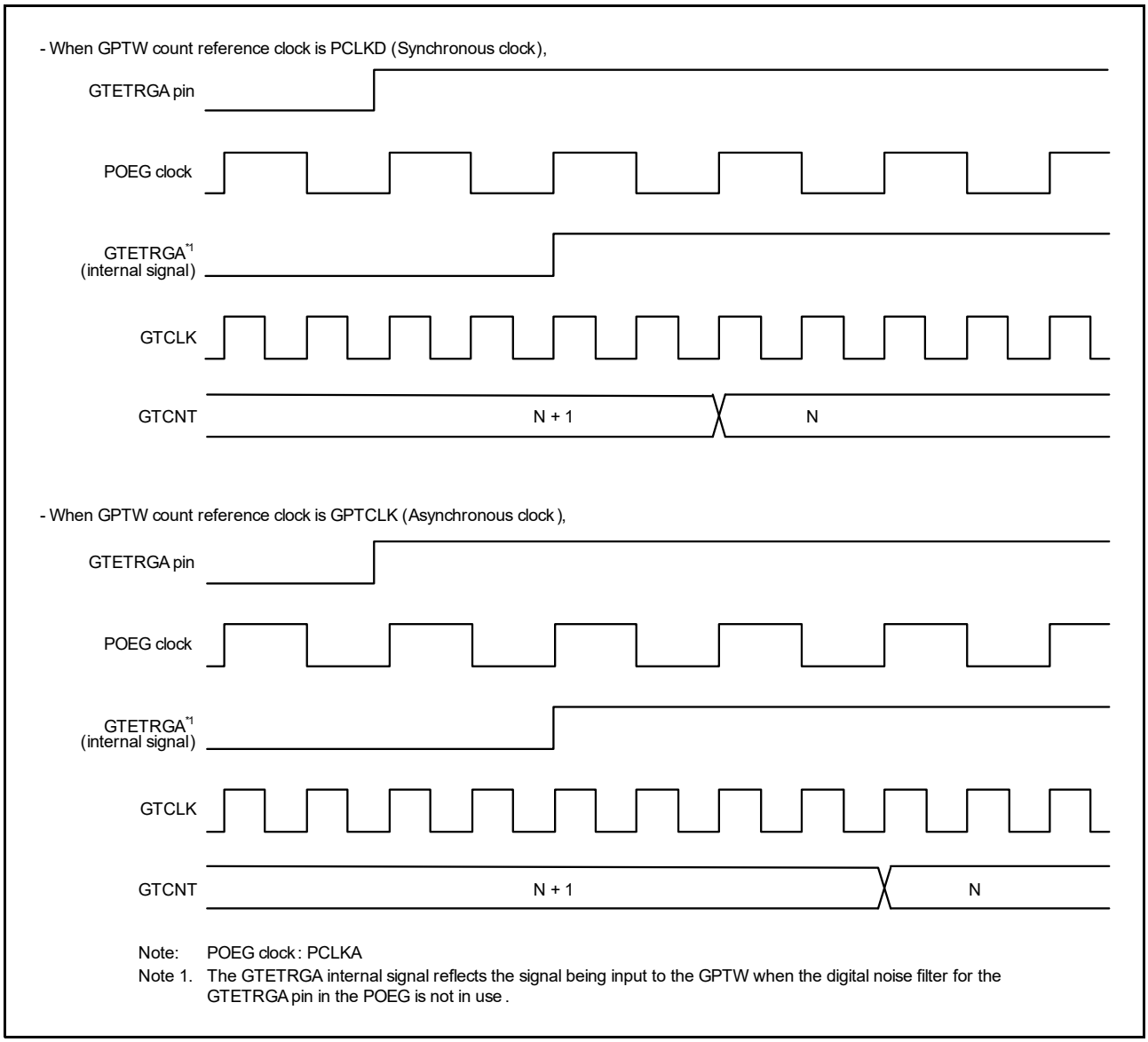


Figure 24.14 Example of Event Count Operation (Down-Counting of Rising Edges of the Input on the GTETRGA Pin)

Figure 24.15 shows an example for setting event count operation in down-counting by hardware source.

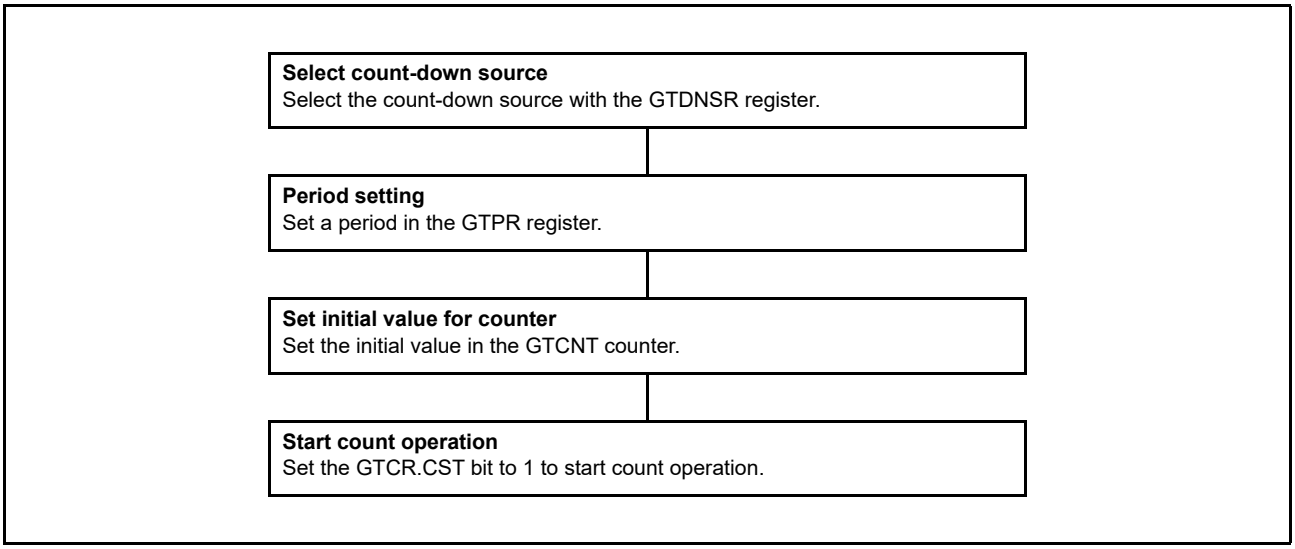


Figure 24.15 Example for Setting Event Count Operation (In Down-Counting by Hardware Source)

(6) Counter Clearing Operation

The counter in each channel can be cleared by the following sources:

- Writing to the GTCNT counter
- Writing 1 to the bit corresponding to the channel number for the GTCLR register while the GTCSR.CCLR bit is 1
- Hardware source specified by the GTCSR register

Do not write to the GTCNT register during counting.
GTCLR register can be written to and cleared by hardware source whether the count operation is performed (GTCR.CST bit = 1) or is stopped (CST bit = 0). When the count direction flag is set as decrement (GTST.TUCF flag = 0) in sawtooth-wave mode (except sawtooth-wave PWM mode 2) selected with GTCR.MD[2:0] or GTCR.MD[3:0] bits, the GTCNT counter value for writing to the GTCLR register and for clearing by hardware source becomes the GTPR register value whether the counter is in operation or is stopped. The GTCNT counter value becomes 0000 0000h for other settings.
When the event count operation is set (at least one of bits for the GTUPSR or the GTDNSR register is set as 1), writing to the GTCLR register and clearing by hardware source are performed right after the clear source is generated (the operation is based on GTCLK). For other settings, clearing operation is synchronized with the count clock selected with the GTCR.TPCS[3:0] bits.

24.3.1.2 Waveform Output by Compare Match

Compare match refers to when the GTCNT counter value matches the GTCCRA or GTCCRB register value. The output of the corresponding GTIOCnA or GTIOCnB pin (n = 0 to 15) can be set to be driven low, high, or toggled in synchronization with the counter clock (including in the case of event counting) after the match.
In addition, the GTIOCnA or GTIOCnB pin output can be driven low, high, or toggled at the “end of the cycle” which is determined by the GTPR register or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in sawtooth-wave PWM mode 2 (n = 0 to 7, m = A to F). The end of the cycle is as follows:

- For sawtooth waves (except sawtooth-wave PWM mode 2) in up-counting: When the GTCNT counter value changes from the GTPR register value to 0000 0000h (overflow)
- For sawtooth waves (except sawtooth-wave PWM mode 2) in down-counting: When the GTCNT counter value changes from 0000 0000h to the GTPR register value (underflow)
- For sawtooth-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits. When the GTCNT counter value changes from the GTCCRm register value to 0000 0000h
- For sawtooth waves with the GTCNT counter clear
- For triangle waves or complementary PWM mode: When the GTCNT counter value changes from 0000 0000h to 0000 0001h (trough)

(1) Low Output and High Output

Figure 24.16 shows an example of low output and high output operation by a compare match of the GTCNT counter and GTCCRA register and of the GTCNT counter and GTCCRB register.
In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that high is output from the GTIOC0A pin by a GPTW0.GTCCRA register compare match, and low is output from the GTIOC0B pin by a GPTW0.GTCCRB register compare match. The pin level does not change when the specified level and pin level match.

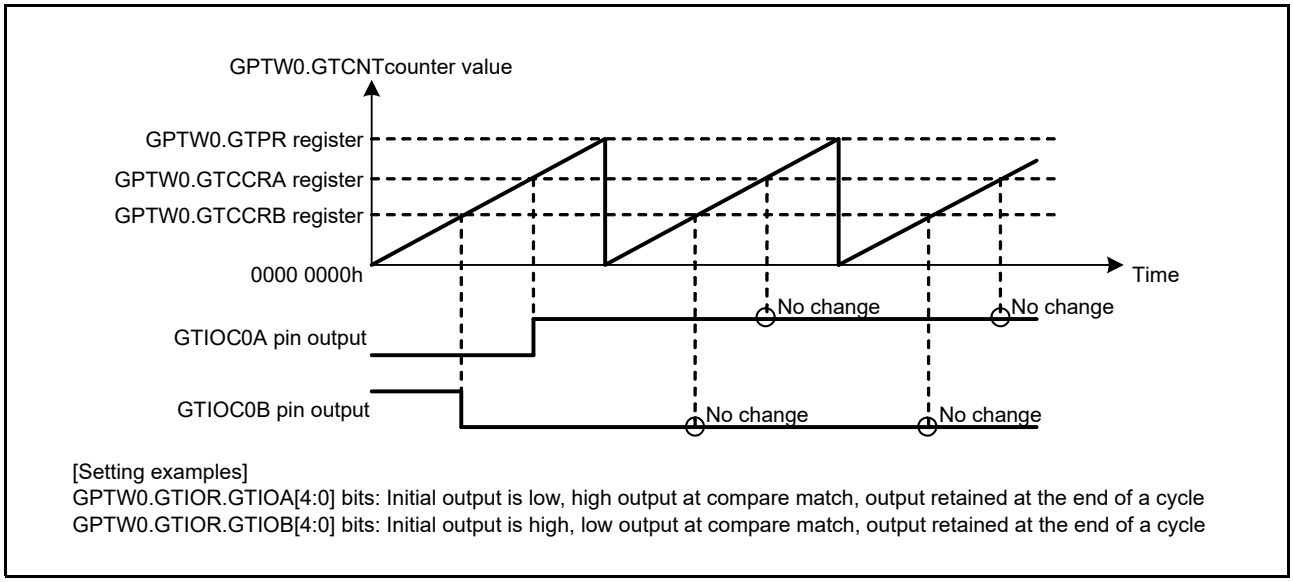
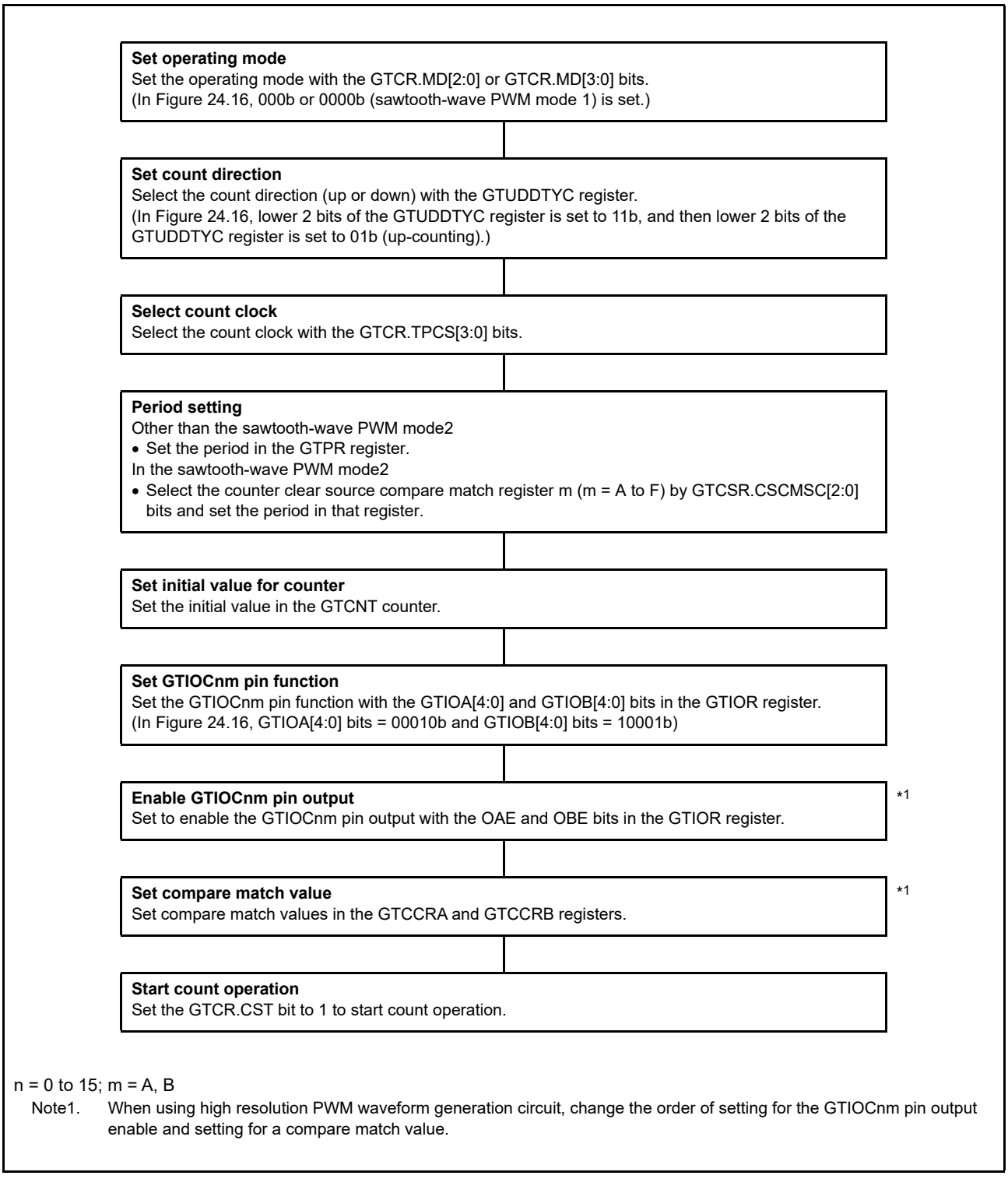


Figure 24.16 Example of Low Output and High Output Operation

Figure 24.17 shows an example for setting Low/High output operation.



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.17 Example for Setting Low/High Output Operation

(2) Toggled Output

Figure 24.18 and Figure 24.19 show examples of toggle output operations by compare match between the GPTW0.GTCNT counter and the GTCCRA and GTCCRB registers.

In Figure 24.18, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output by a GPTW0.GTCCRA register compare match and GTIOC0B pin output by a GPTW0.GTCCRB register compare match are toggled.

In Figure 24.19, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output is toggled by a compare match of GPTW0.GTCCRA register and the GTIOC0B pin output is toggled at the end of the cycle.

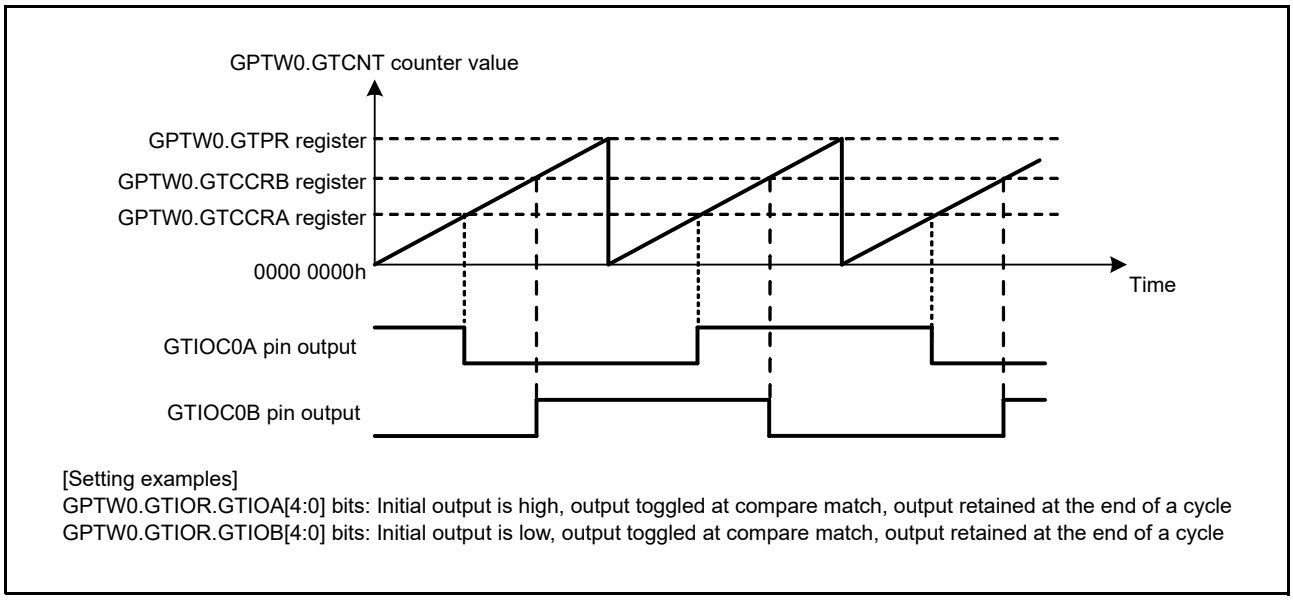


Figure 24.18 Example of Toggled Output Operation (1)

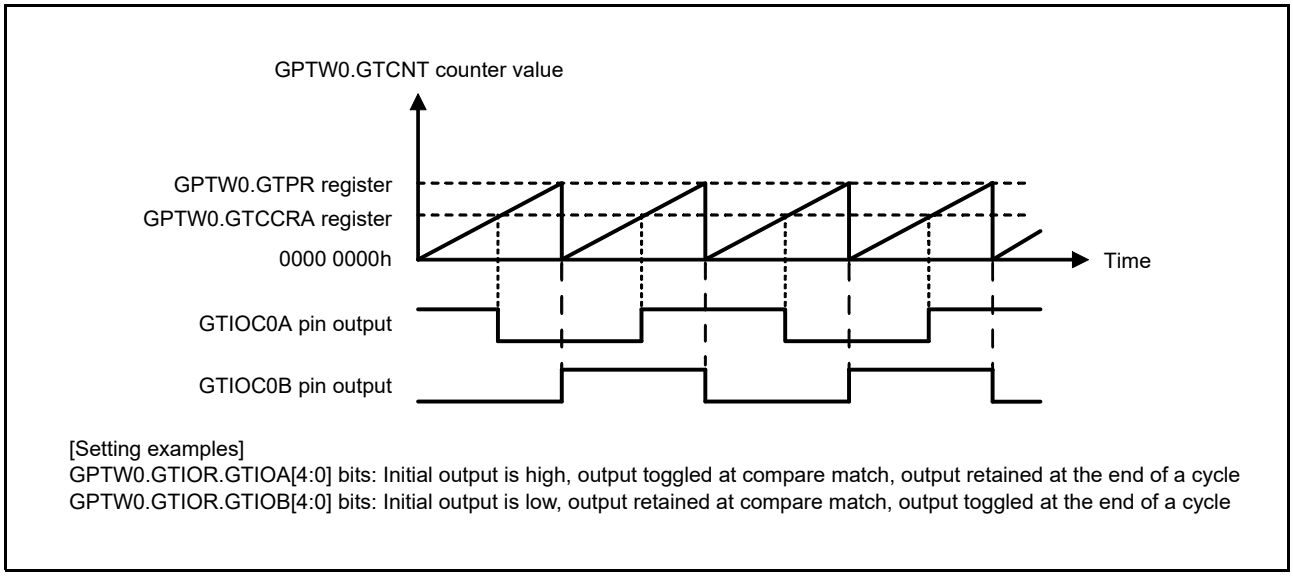
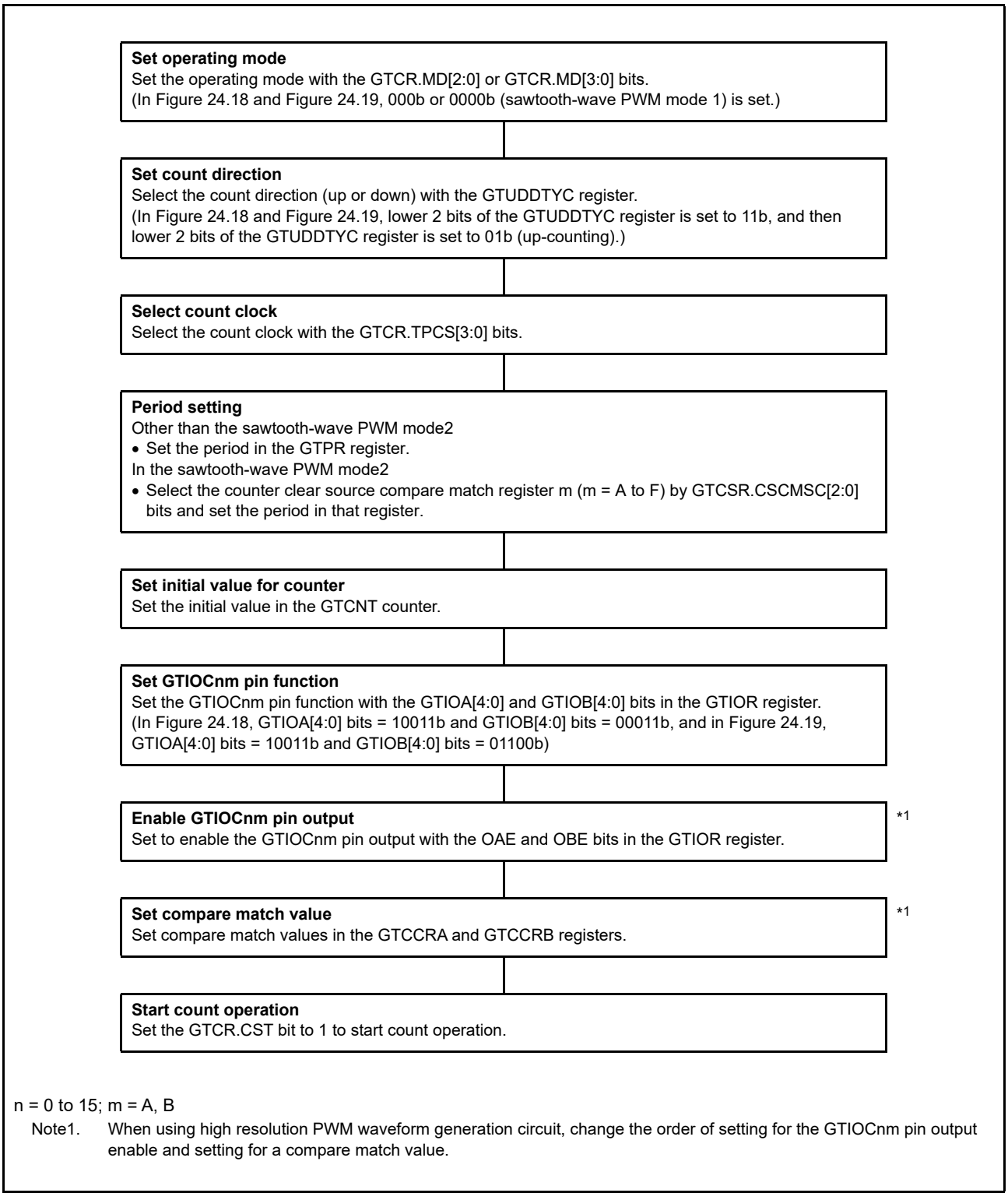


Figure 24.19 Example of Toggled Output Operation (2)

Figure 24.20 shows an example for setting toggled output operation.



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.20 Example for Setting Toggled Output Operation

24.3.1.3 Input Capture Function

Through detecting hardware sources selected by the GTICASR and GTICBSR registers, the GTCNT counter values can be transferred to the GTCCRA and GTCCRB registers respectively.

In complementary PWM mode, the GTCCRA and GTCCRB registers do not function as input capture registers.

Figure 24.21 shows an example of input capture function operation.

In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that an input capture is performed at both edges of the GTIOC0A pin input and at the rising edge of the GTIOC0B pin input.

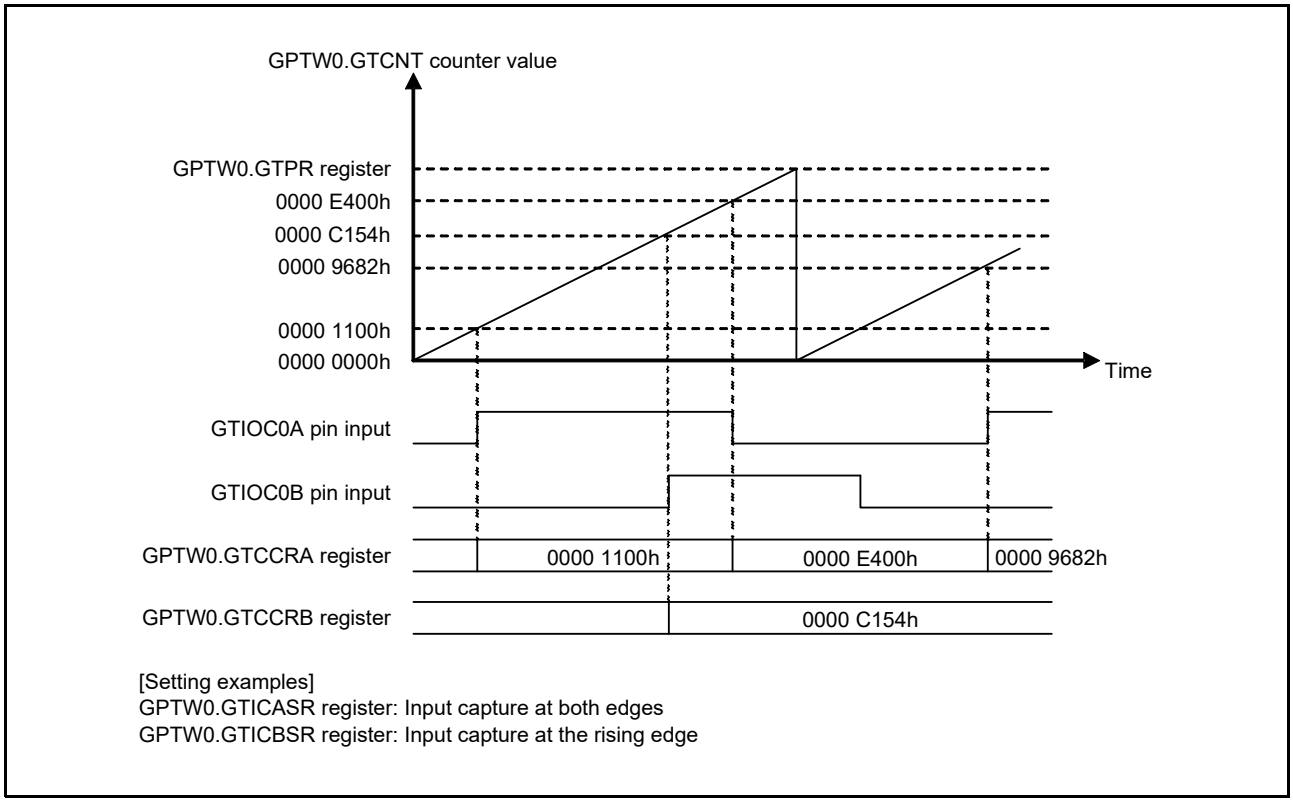


Figure 24.21 Example of Input Capture Operation

Figure 24.22 shows an example for setting input capture operation.

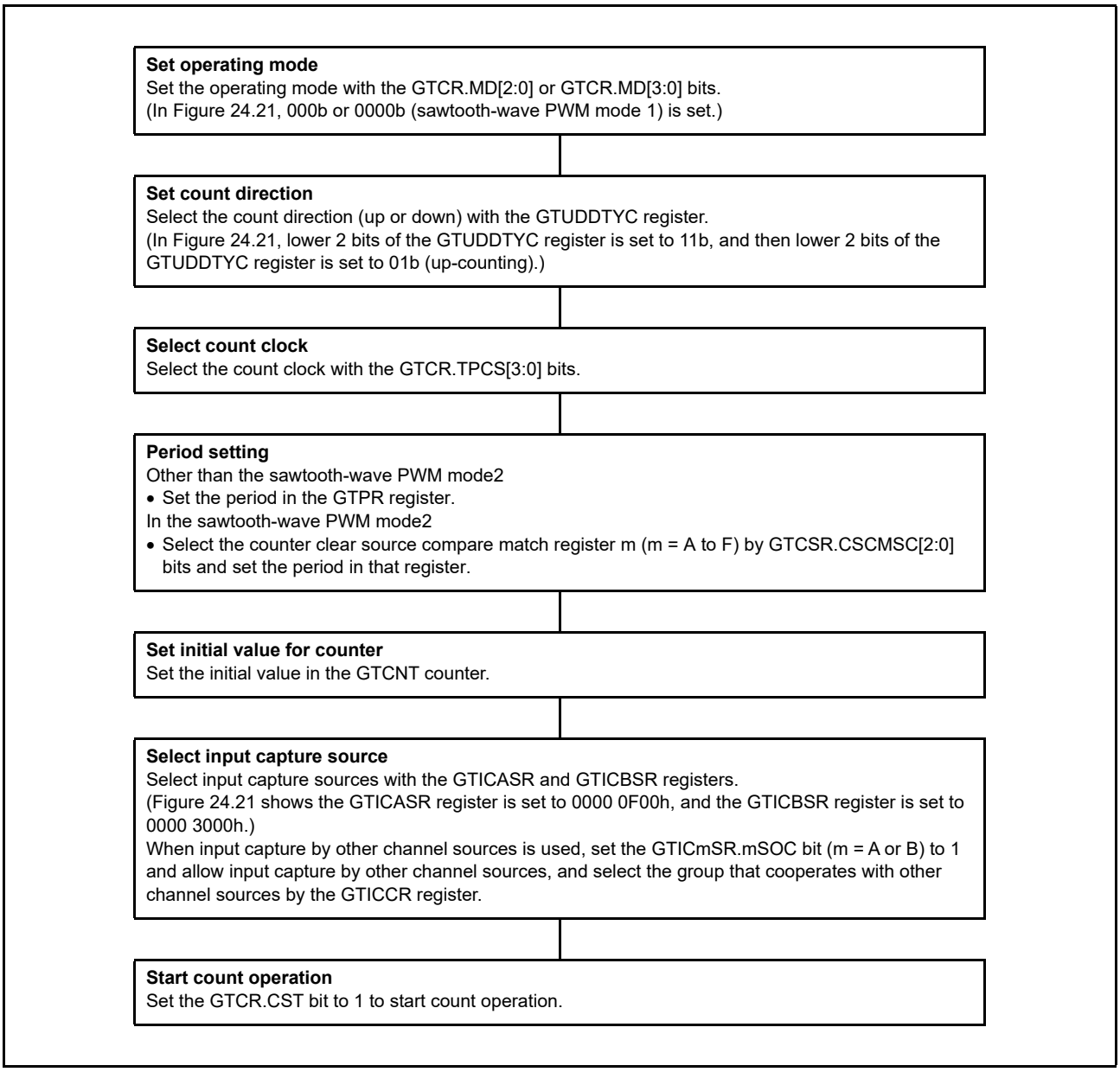


Figure 24.22 Example for Setting Input Capture Operation

Figure 24.23 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin.

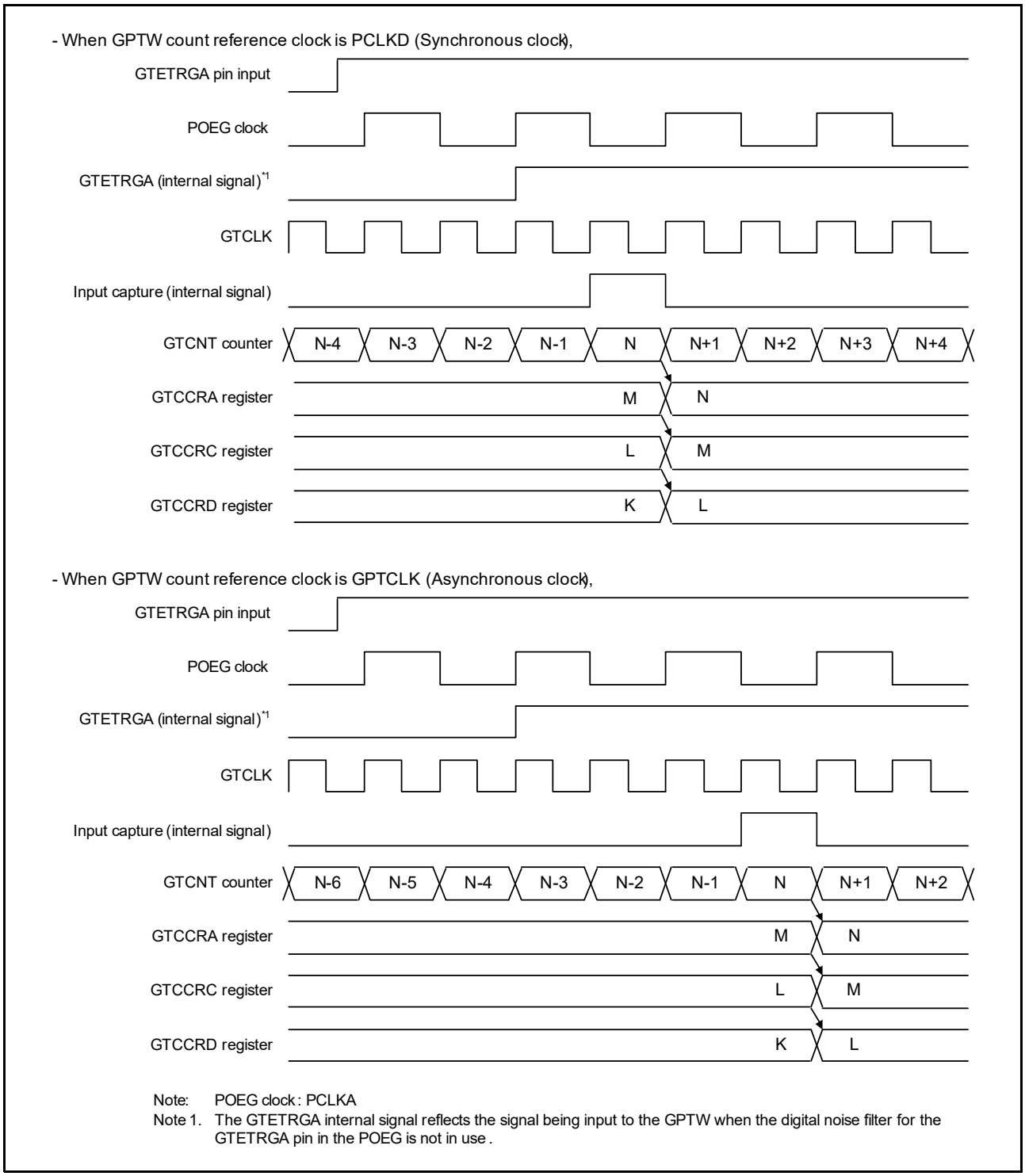


Figure 24.23 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.24 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin.

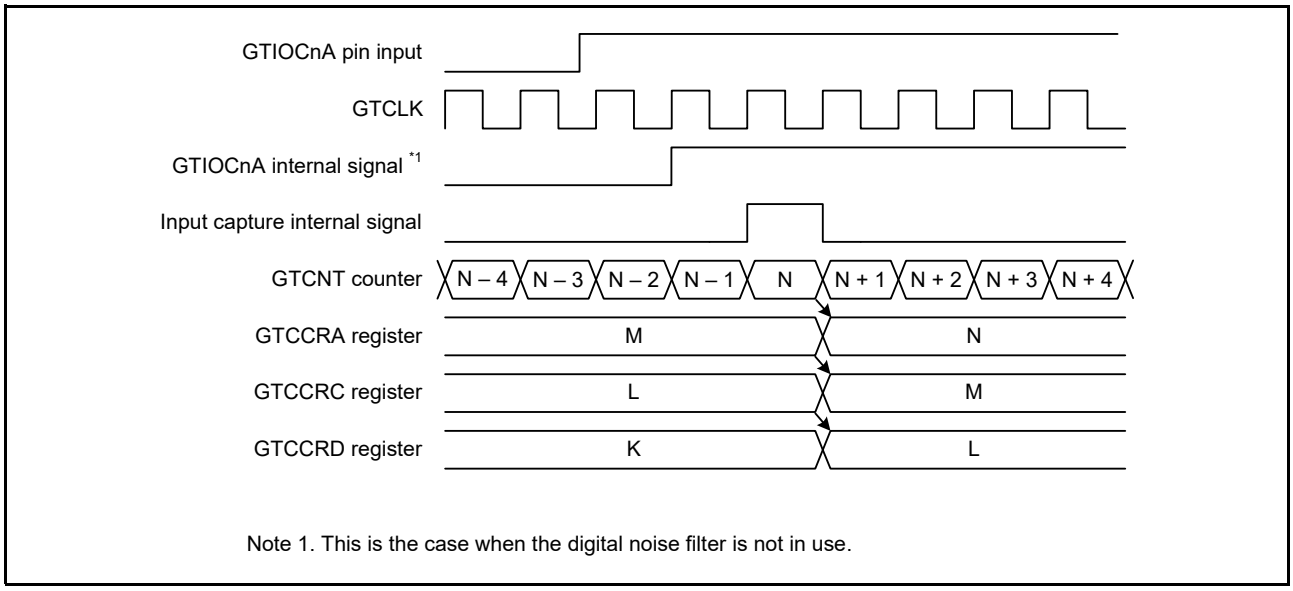


Figure 24.24 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.25 shows an example of the timing of operations for input capture in response to event input from the ELCA. This is an example of capture of the counter value by the GPTW1.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

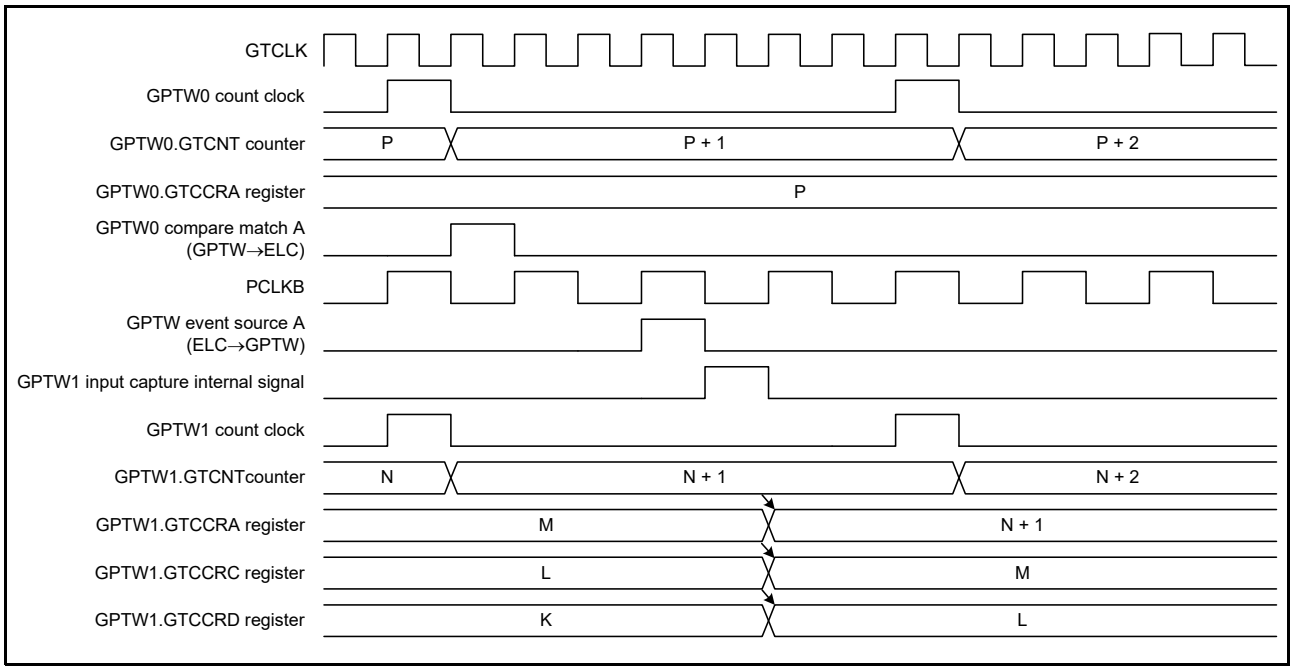


Figure 24.25 Example of the Timing of Operations for Input Capture in Response to Event Input from the ELCA

Figure 24.26 shows the input capture operation timing by the counter clock of other channels.

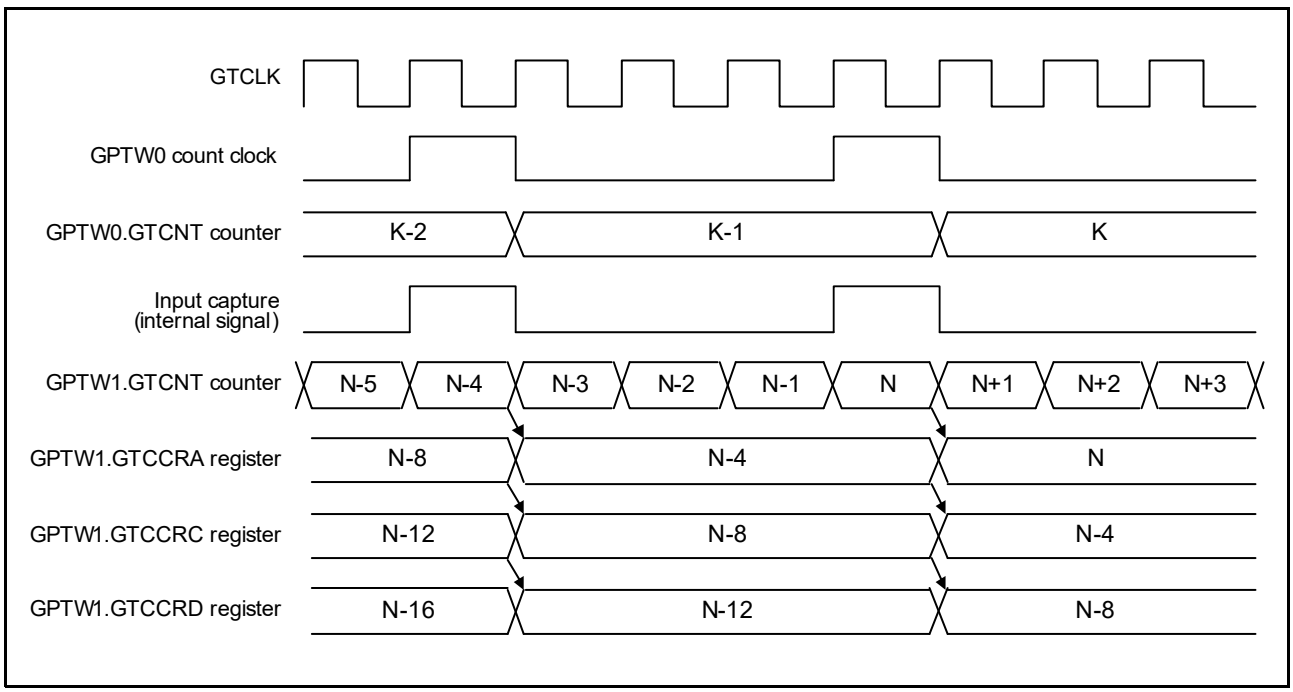


Figure 24.26 Example of the Timing of Input Capture Operation in Response to Count Clock from Other Channel

24.3.2 Buffer Operation

The following buffer operation can be set with the GTBER register.

- Buffer operation with the GTPR, GTPBR, and GTPDBR registers used together
- Buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers used together
- Buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers used together
- Buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- Buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

The following buffer operation can be set with the GTDTCR register.

- Buffer operation with the GTDVU and GTDBU registers used together
- Buffer operation with the GTDVD and GTDBD registers used together

The following buffer operation can be set with the GTBER2 register.

- Buffer operation with the GTCCRA, GTCCRE, and GTCCRF registers (in complementary PWM mode 3, 4)
- Buffer operation with the GTOLBR.GTIOAB[4:0] bits and GTIOR.GTIOA[4:0] bits
- Buffer operation with the GTOLBR.GTIOBB[4:0] bits and GTIOR.GTIOB[4:0] bits

24.3.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register).
In complementary PWM mode, the buffer transfer from the GTPDBR register to the temporary register P is performed only in the master channel (GPTWn). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPTWn+1), and slave channel 2 (GPTWn+2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the period of the GTCNT counters of the three channels. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.
The setting is invalid in the sawtooth-wave PWM mode 2.
The buffer transfer is performed at an overflow (in up-counting) or underflow (in down-counting) in sawtooth-wave mode or in event counting, or at a trough in triangle-wave mode.
When in sawtooth-wave mode and in event counting, the buffer transfer is performed even when the counter clearing shown below occurs during the counting operation.

- Hardware source clearing

(Clear source selected with GTCSR.CSGTRGAR, CSGTRGAF, CSGTRGBR, CSGTRGBF, CSGTRGCR, CSGTRGCF, CSGTRGDR, CSGTRGDF, CSCARBL, CSCARBH, CSCAFBL, CSCAFBH, CSCBRAL, CSCBRAH, CSCBFAL, CSCBFAH, CSELCA, CSELCB, CSELCC, CSELCD, CSELCE, CSELCF, CSELCG, CSELCH, CSCMSC[2:0], or CP1CCE bit)

- Clearing by software

(When the GTCLR.CCLRn bit is written to 1 during the GTCSR.CCLR bit is set to 1) (n = 0 to 15)
In the case of sawtooth waves, the buffer transfer by clearing counter can be prohibited by the GTBER2.CCTPR bit.
Figure 24.12 shows the buffer transfer timing in the complementary PWM mode.

Table 24.11 The GTPR Buffer Transfer Timing in Complementary PWM Mode

Mode	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3 Complementary PWM mode 4
GTPDBR register ↓ Temporary register P	Counter running • One cycle of GTCLK after writing to the GTCCRD register for slave channel 2 (GPTWn+2) Counter stopped • One cycle of GTCLK after writing to the GTPDBR register	Counter running • One cycle of GTCLK after writing to the GTCCRD register for slave channel 2 (GPTWn+2) Counter stopped • One cycle of GTCLK after writing to the GTPDBR register	Counter running • One cycle of GTCLK after writing to the GTCCRD register for slave channel 2 (GPTWn+2) Counter stopped • One cycle of GTCLK after writing to the GTPDBR register
Temporary register P ↓ GTPBR register	When data is transferred to the temporary register P during up-counting middle • One cycle of GTCLK after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than up-counting middle • At the end of trough Counter stopped • One cycle of GTCLK after transfer of data to the temporary register P	When data is transferred to the temporary register P during down-counting middle • One cycle of GTCLK after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than down-counting middle • At the end of crest Counter stopped • One cycle of GTCLK after transfer of data to the temporary register P	When data is transferred to the temporary register P middle • One cycle of GTCLK after transfer of data to the temporary register P When data is transferred to the temporary register P during a section other than middle • At the end of crest/trough Counter stopped • One cycle of GTCLK after transfer of data to the temporary register P
GTPBR register ↓ GTPR register	At the end of crest • Counter clearing in an up-counting middle or crest section, including counter clearing enabled by setting the GTCSR.CP1CCE bit	At the end of trough • Counter clearing in a down-counting middle or trough section	At the end of crest At the end of trough • Counter clearing

To enable double buffer operation for the GTPR register, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. To disable buffering, set 00b.

In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bits setting.

Figure 24.27 to Figure 24.29 show examples of the GTPR register buffer operation and Figure 24.33 shows an example for setting the GTPR register buffer operation.

For details of operation settings in complementary PWM mode, refer to section 24.3.3, PWM Output Operating Mode.

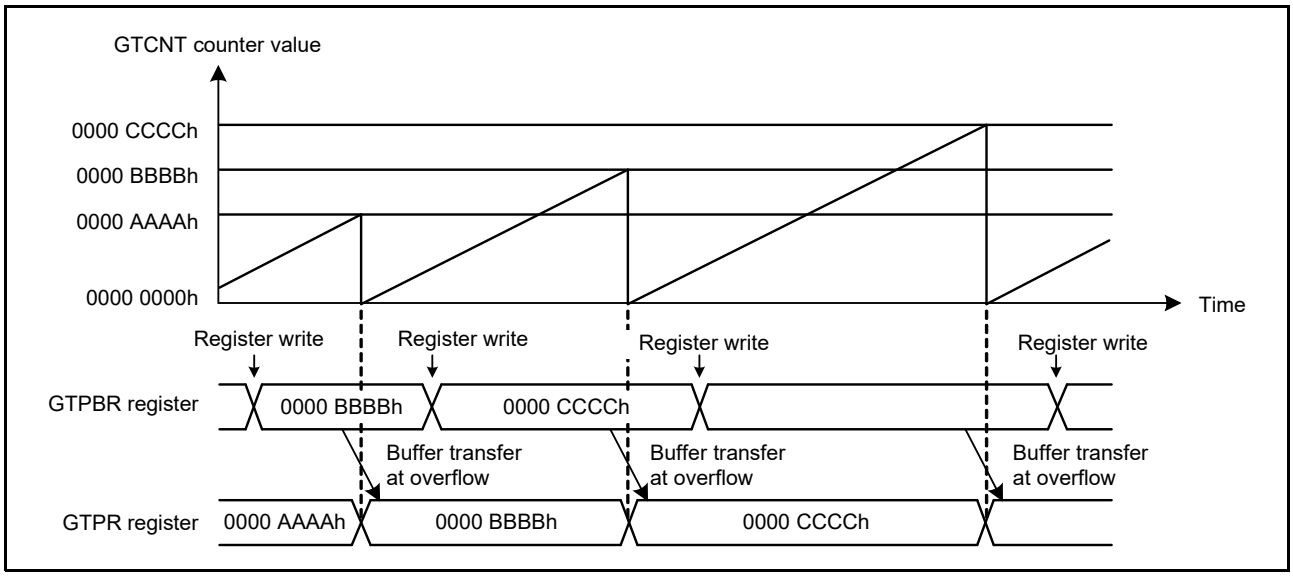


Figure 24.27 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Up-Counting)

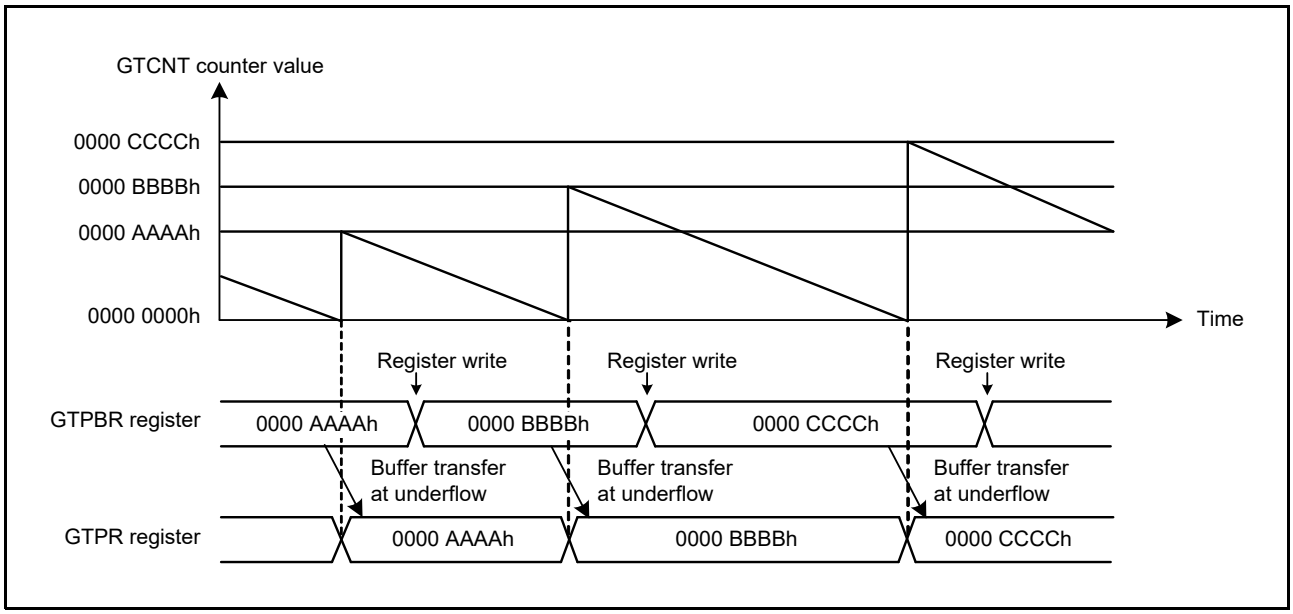


Figure 24.28 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Down-Counting)

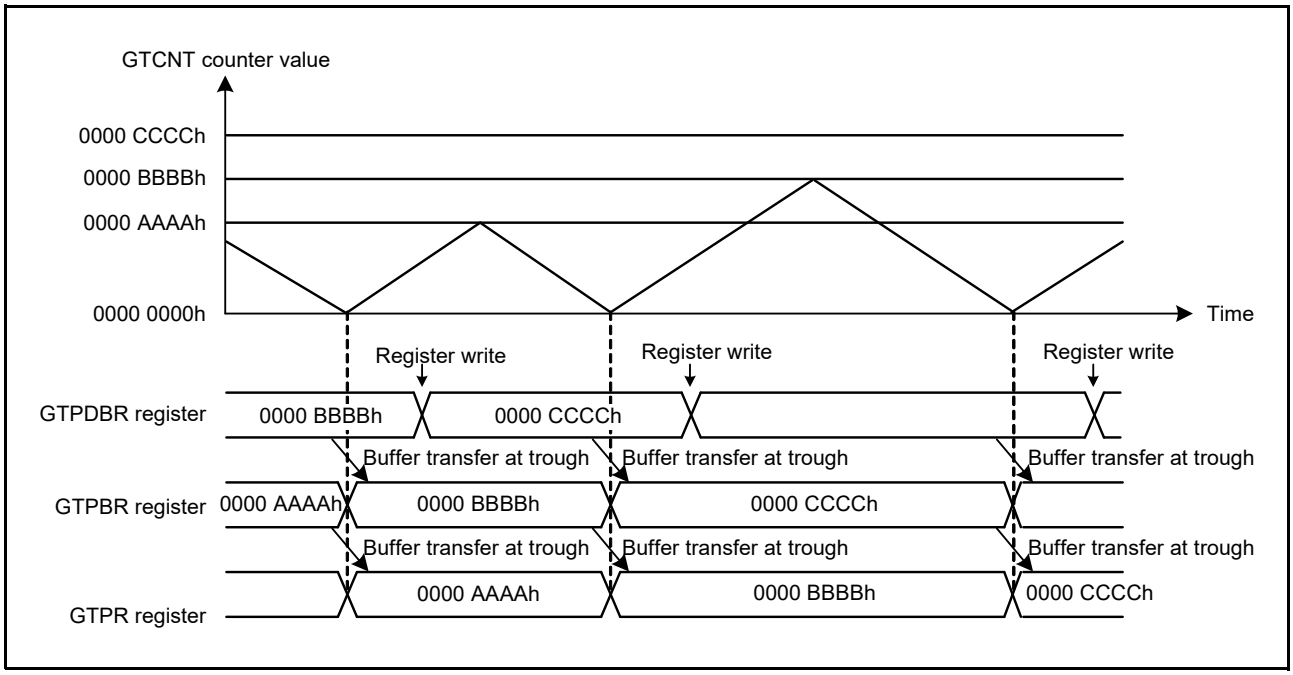


Figure 24.29 Example of the GTPR Register Double Buffer Operation (Triangle Waves)

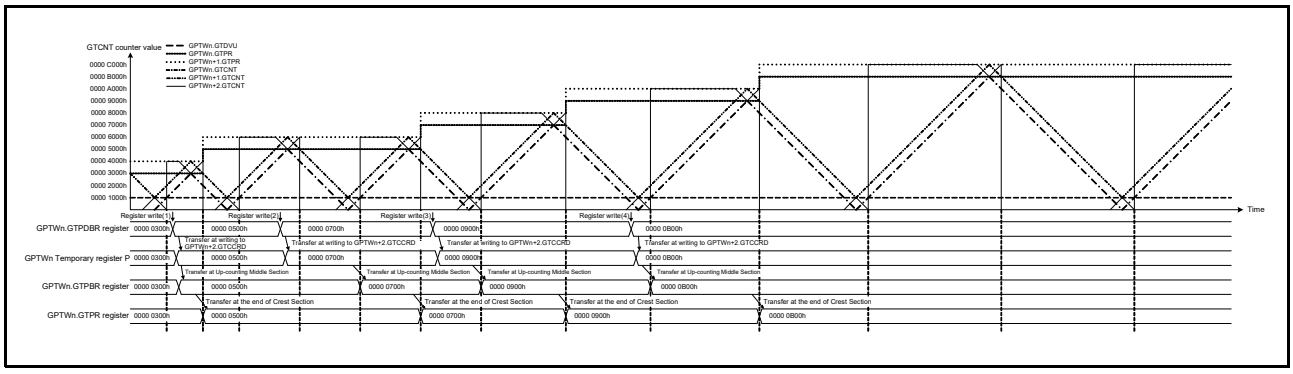


Figure 24.30 Example of GTPR Double Buffer Operation (Complementary PWM Mode 1)

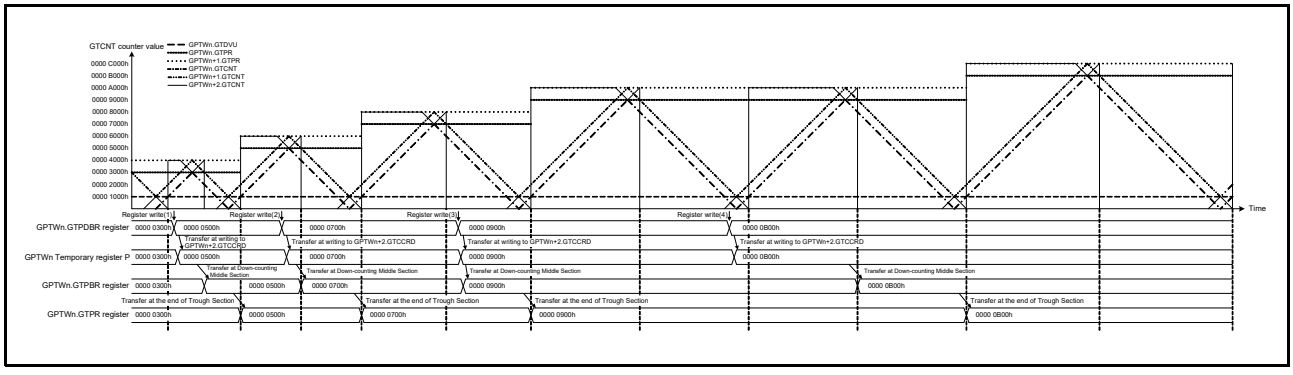


Figure 24.31 Example of GTPR Double Buffer Operation (Complementary PWM Mode 2)

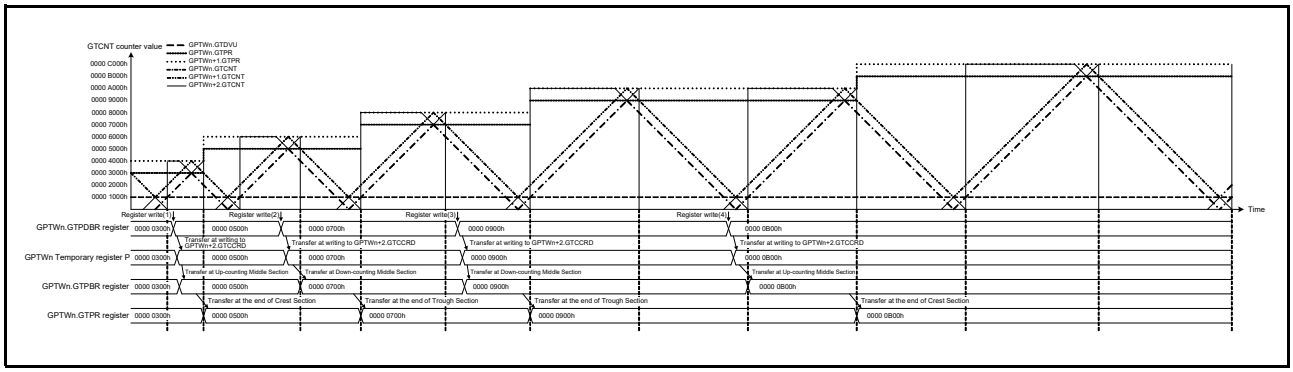


Figure 24.32 Example of GTPR Double Buffer Operation (Complementary PWM Mode 3, 4)

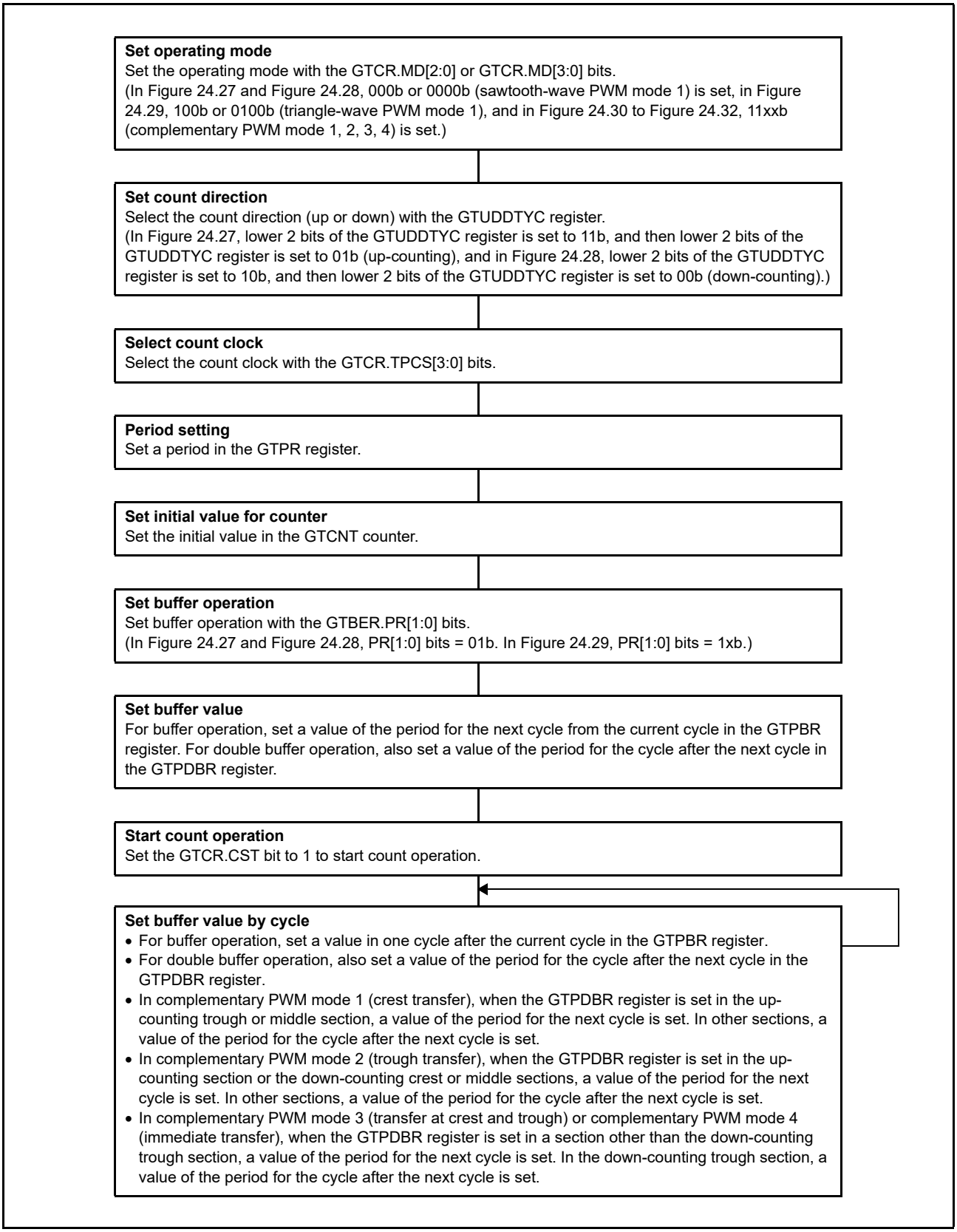


Figure 24.33 Example for Setting the GTPR Register Buffer Operation

24.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).
To enable double buffer operation for the GTCCRA or GTCCRB register, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b, respectively. For single buffer operation, set 01b. To disable buffering, set 00b.
In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or complementary PWM mode, the buffer operations specific to each PWM output operating mode are performed regardless of the settings of the GTBER.CCRA[1:0] and GTBER.CCRB[1:0] bits. For details on the buffer operations in sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, and complementary PWM mode, refer to section 24.3.3, PWM Output Operating Mode.
The following describes buffer operation during output compare and input capture operation.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or complementary PWM mode, the buffer operation is performed as appropriate for the PWM output operating mode regardless of the setting of the GTBER.CCRA[1:0] and GTBER.CCRB[1:0] bits. For details, refer to section 24.3.3, PWM Output Operating Mode. Other than the above PWM output operating mode, buffer transfer is performed in the following four cases:

- Buffer transfer at an overflow (in up-counting) and underflow (in down-counting)
  Buffer transfer is performed at an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode and in event counting, and at a trough (triangle-wave PWM mode 1) or at a crest/trough (triangle-wave PWM mode 2) in triangle-wave mode.
- Buffer transfer by counter clearing
  In sawtooth-wave mode and in event counting, the buffer transfer by the same counter clearing sources described in section 24.3.2.1, GTPR Register Buffer Operation, is performed during counting operation, in the same way as at an overflow (in up-counting) or an underflow (in down-counting).
  In sawtooth-wave, the buffer transfer of GTCCRm register by counter clearing can be disabled with the GTBER2.CCTCm bit (m = A, B).
  In triangle-wave mode, buffer transfer by counter clearing is not performed.
- Buffer transfer by compare match
  In sawtooth-wave, the buffer transfer by the compare match of GTCCRm register enabled by GTBER2.CMTCm (m = A, B) bit is performed.
- Forcible buffer transfer
  In sawtooth-wave mode, triangle-wave mode, and event counting, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while the counting is stopped.
  In sawtooth-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B is also performed using the forcible buffer operation.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.
Figure 24.34 to Figure 24.37 show examples of the GTCCRA and GTCCRB registers buffer operations, and Figure 24.38 shows an example for setting the GTCCRA and GTCCRB registers buffer operations.

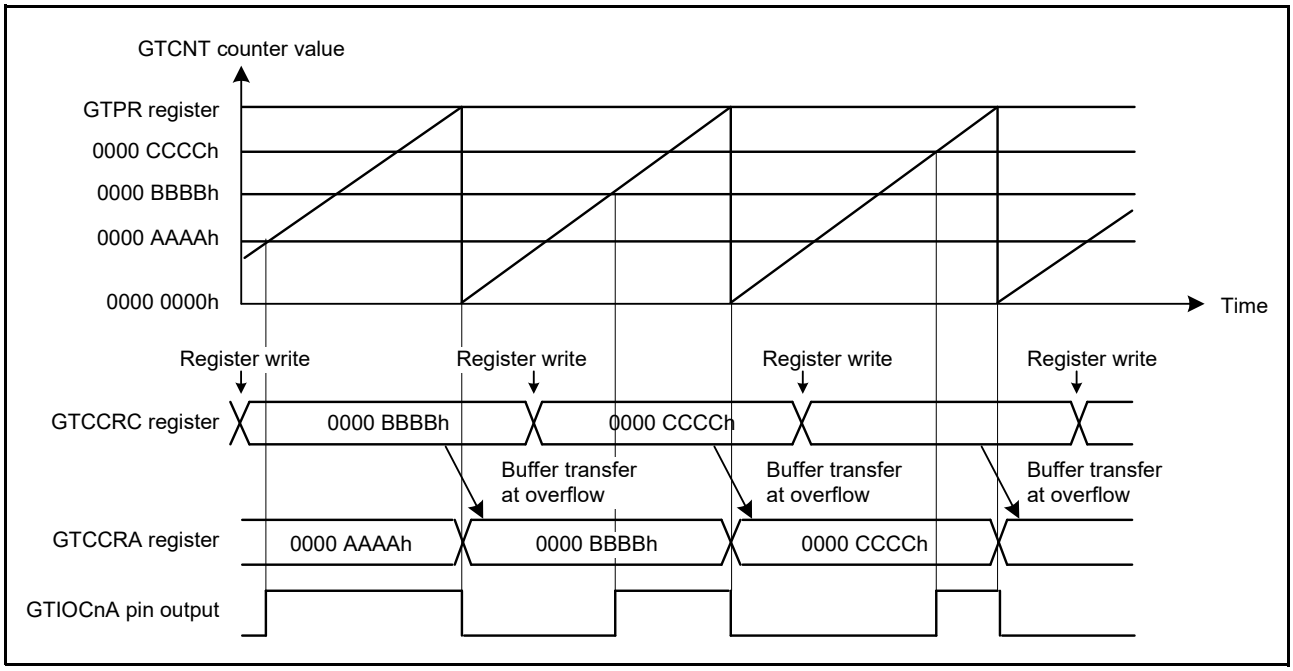


Figure 24.34 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Sawtooth-Waves, Sawtooth-Wave PWM mode 1 in Up-Counting, High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) (n = 0 to 15)

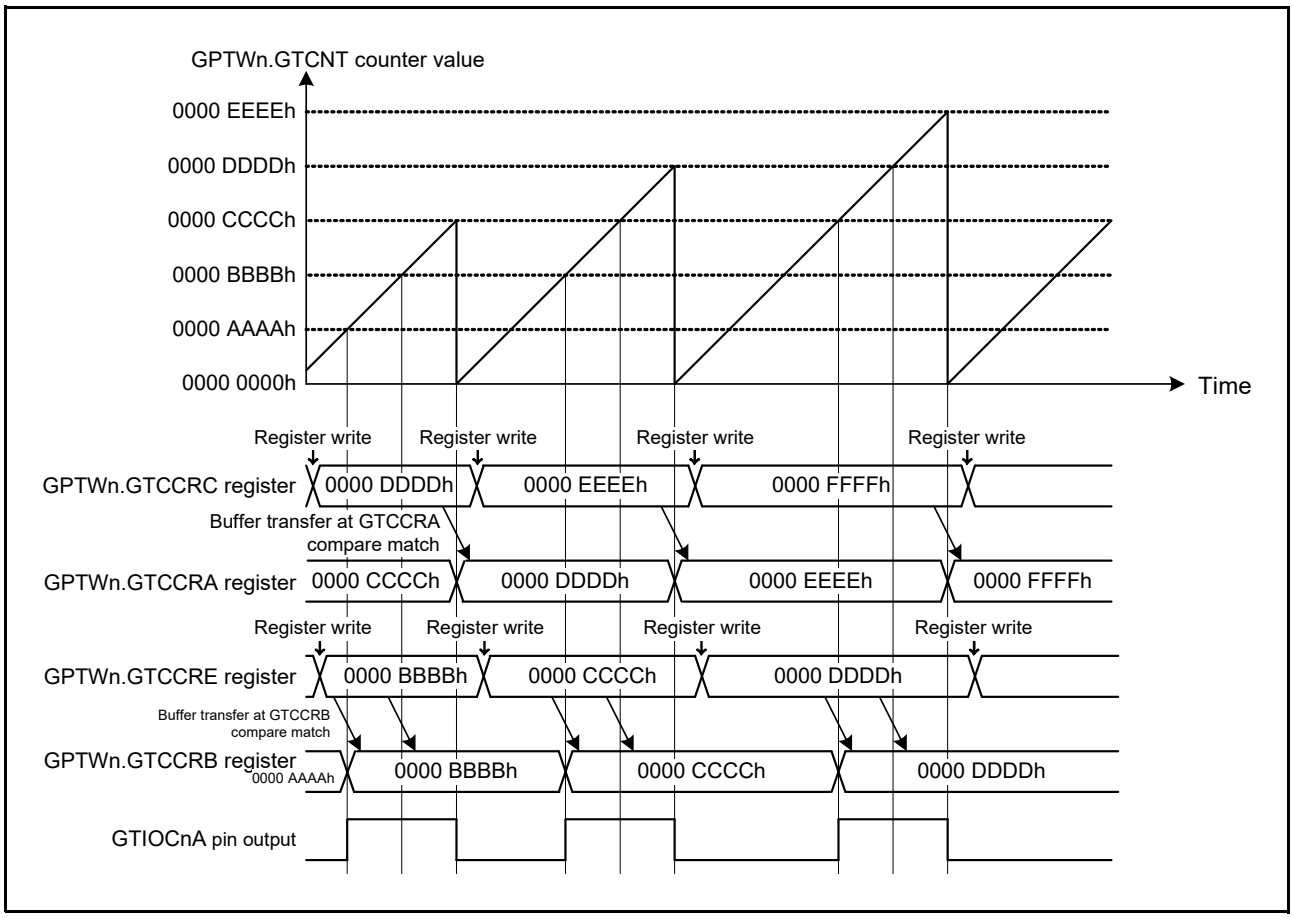


Figure 24.35 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Sawtooth-Wave PWM mode 2, Buffer Transfer at GTCCRA Register Compare Match, Counter Clearing, Low Output, Buffer Transfer at GTCCRB Register Compare Match, High Output) (n = 0 to 15)

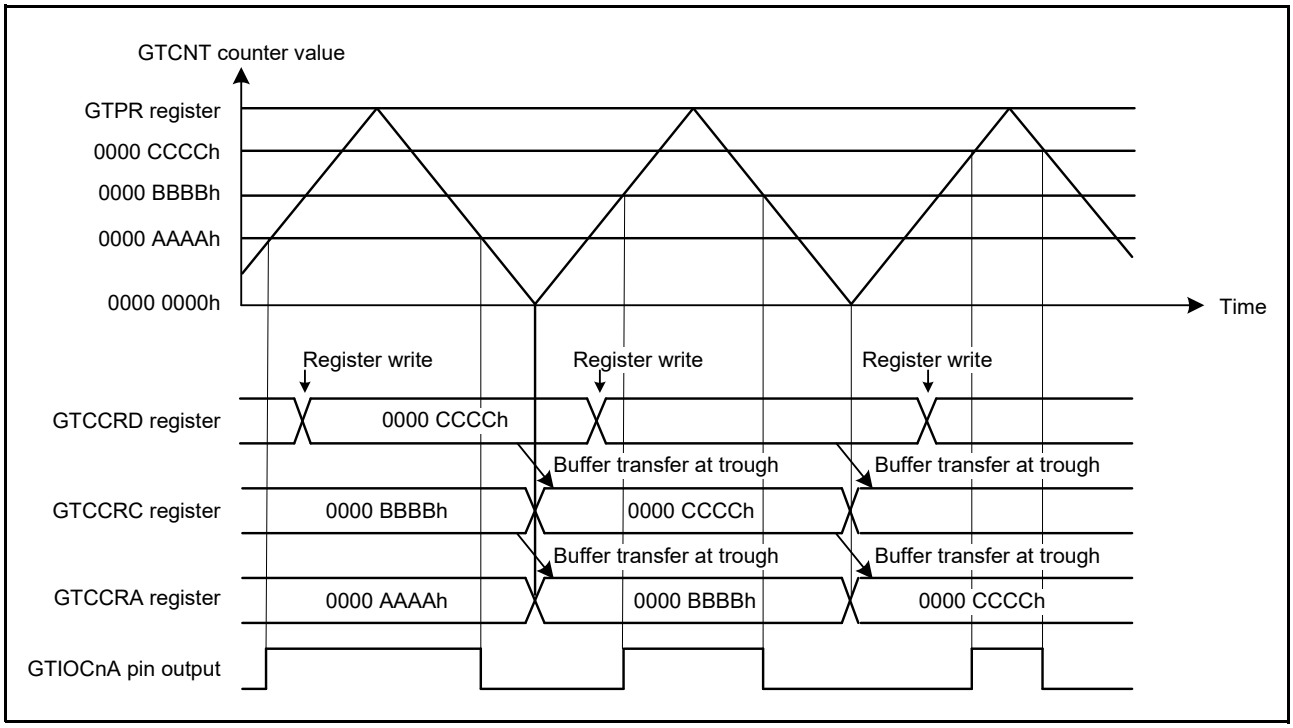


Figure 24.36 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Trough, Output Toggled at GTCCRA Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 15)

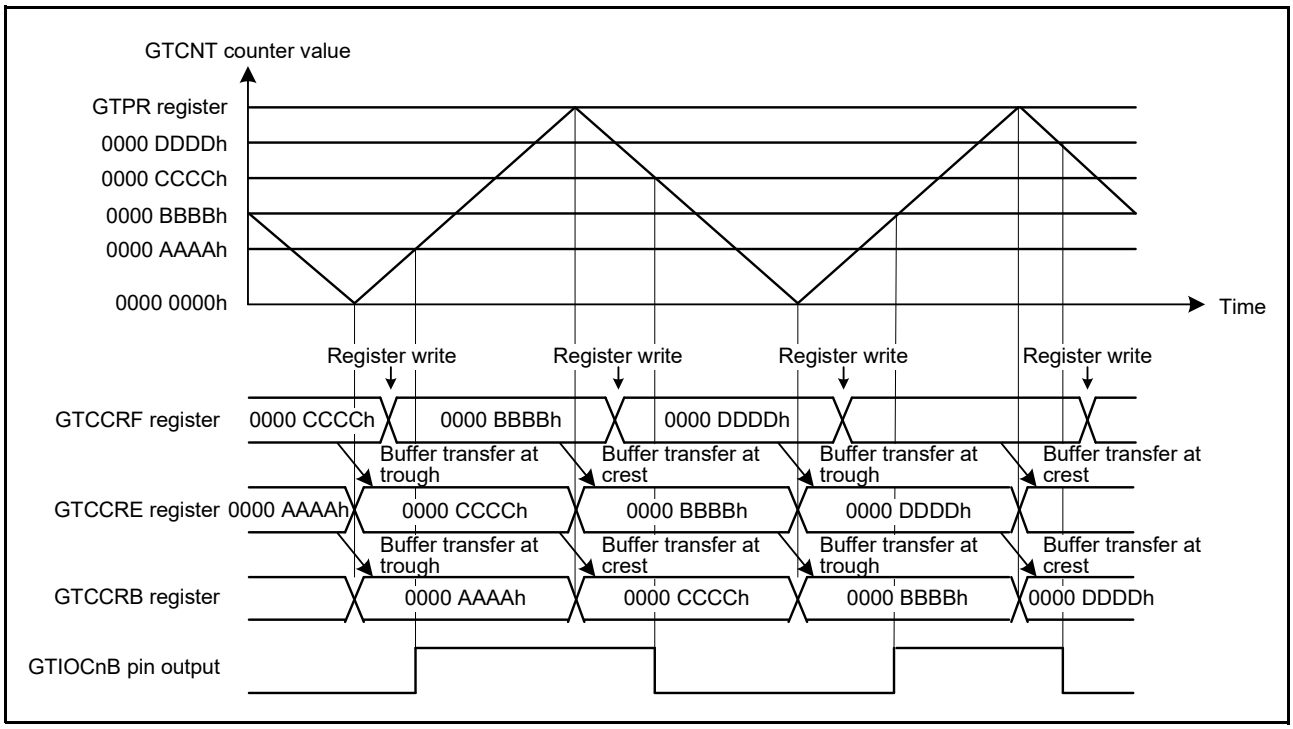
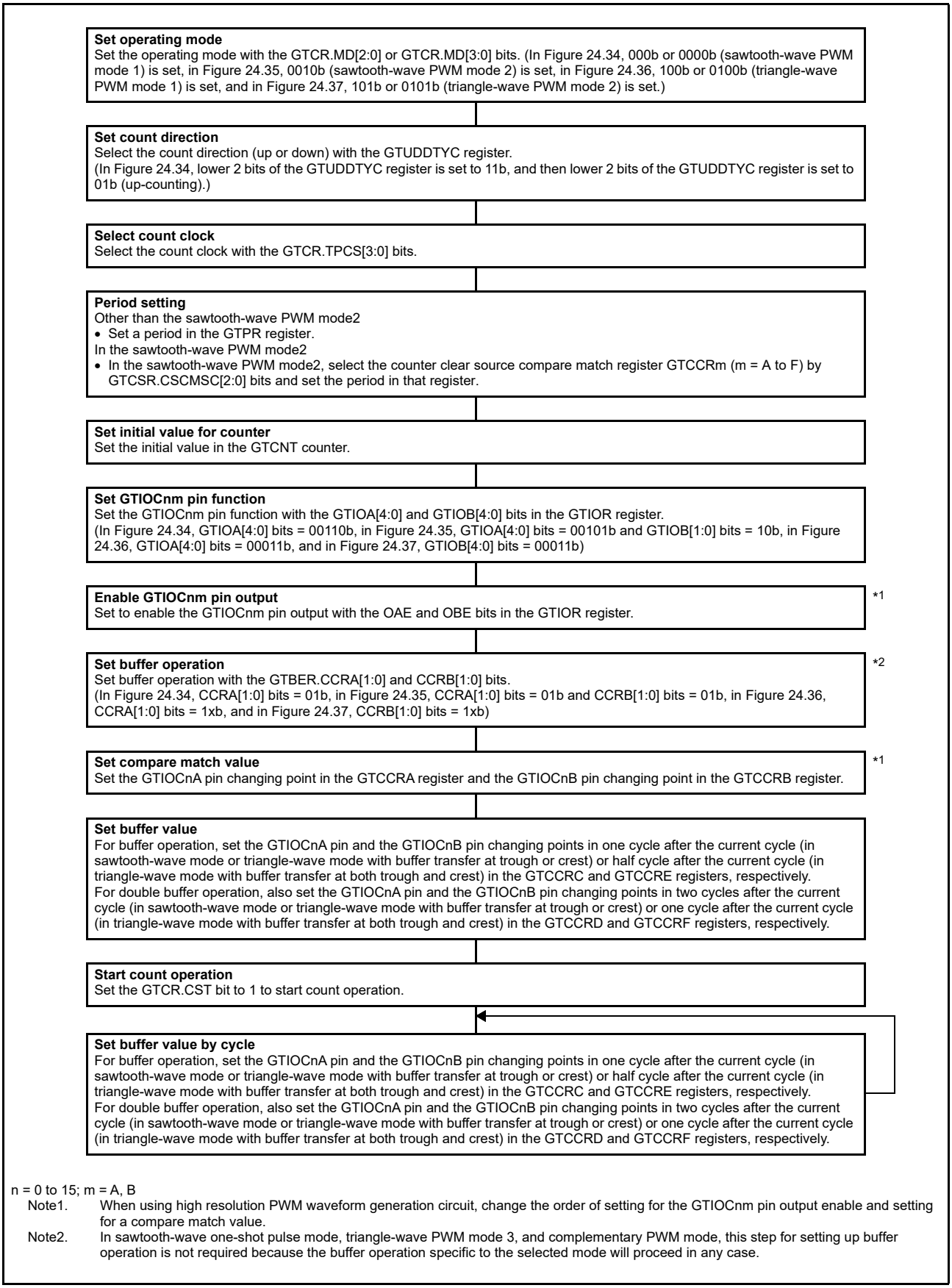


Figure 24.37 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Both Troughs and Crests, Output Toggled at GTCCRB Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 15)



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Note2. In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, and complementary PWM mode, this step for setting up buffer operation is not required because the buffer operation specific to the selected mode will proceed in any case.

Figure 24.38 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Output Compare)

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

The transfer to the buffers is performed when an input capture occurs. Upon input capture, the GTCNT counter value is transferred to the GTCCRA and GTCCRB registers and the values previously stored in the GTCCRA and GTCCRB registers are transferred to the buffer registers. Buffer transfer by counter clearing does not occur when input capture is in use.

Figure 24.39 and Figure 24.40 show examples of buffer operation of the GTCCRA and GTCCRB registers, and Figure 24.41 shows an example for setting buffer operation of the GTCCRA and GTCCRB registers.

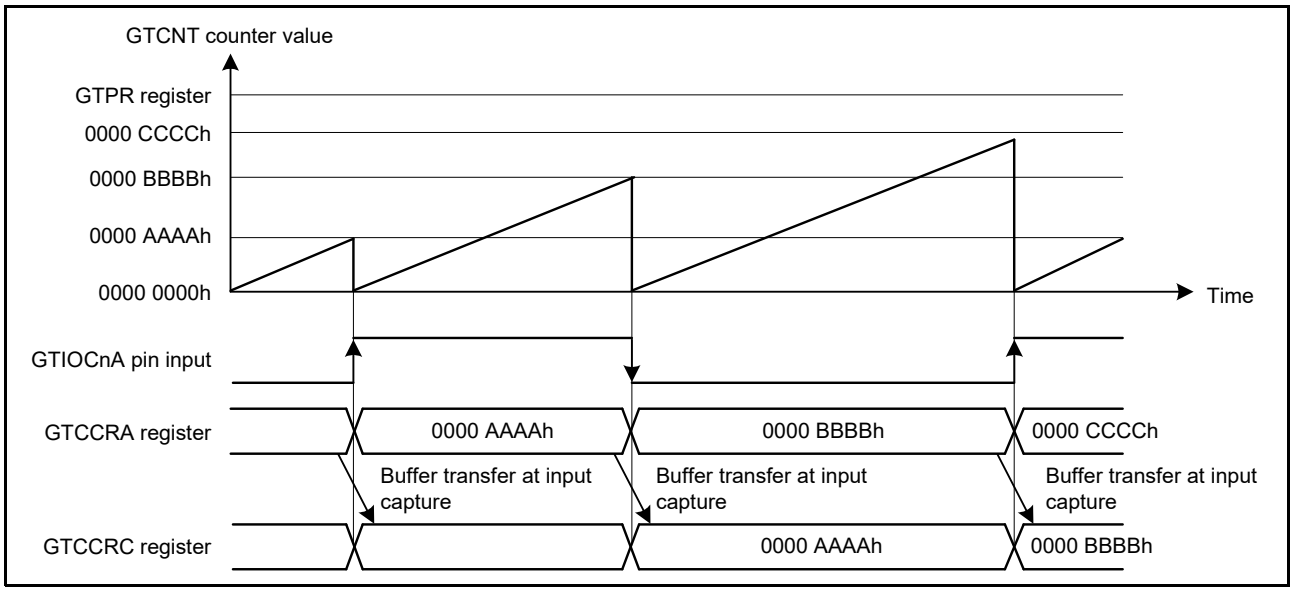


Figure 24.39 Example of Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnA Pin) (n = 0 to 15)

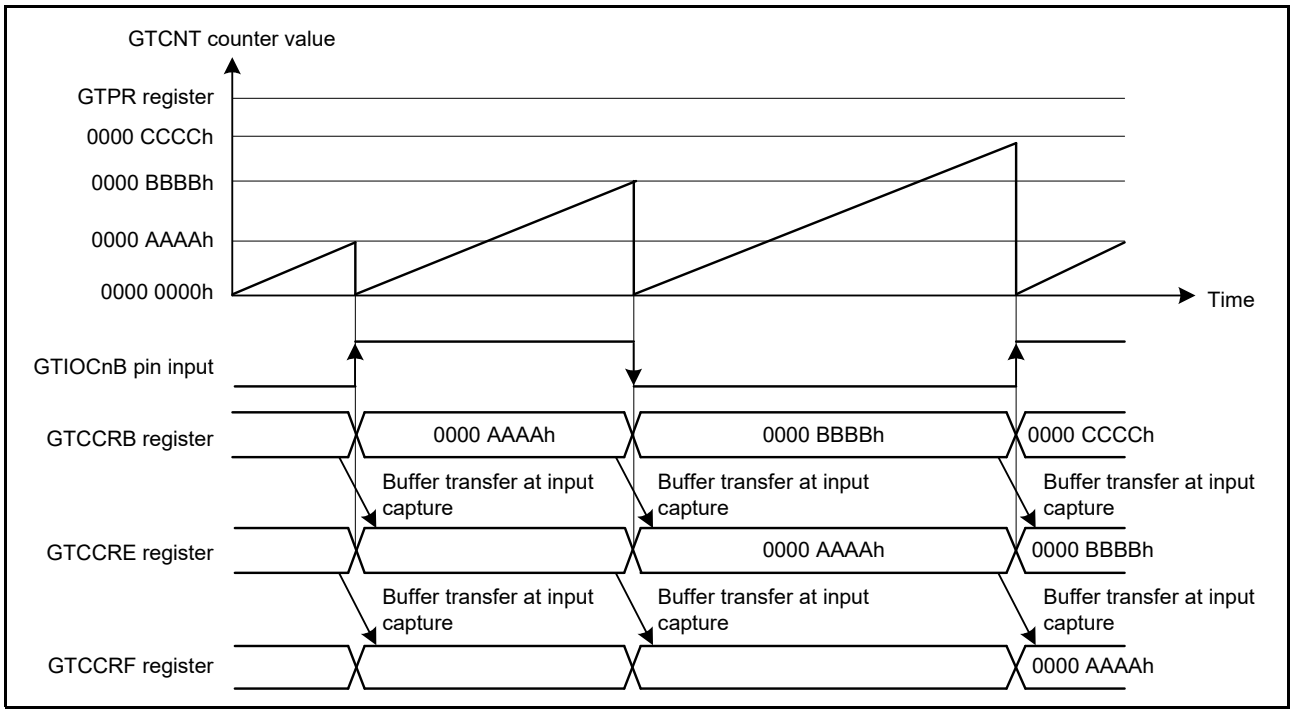


Figure 24.40 Example of Double Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnB Pin) (n = 0 to 15)

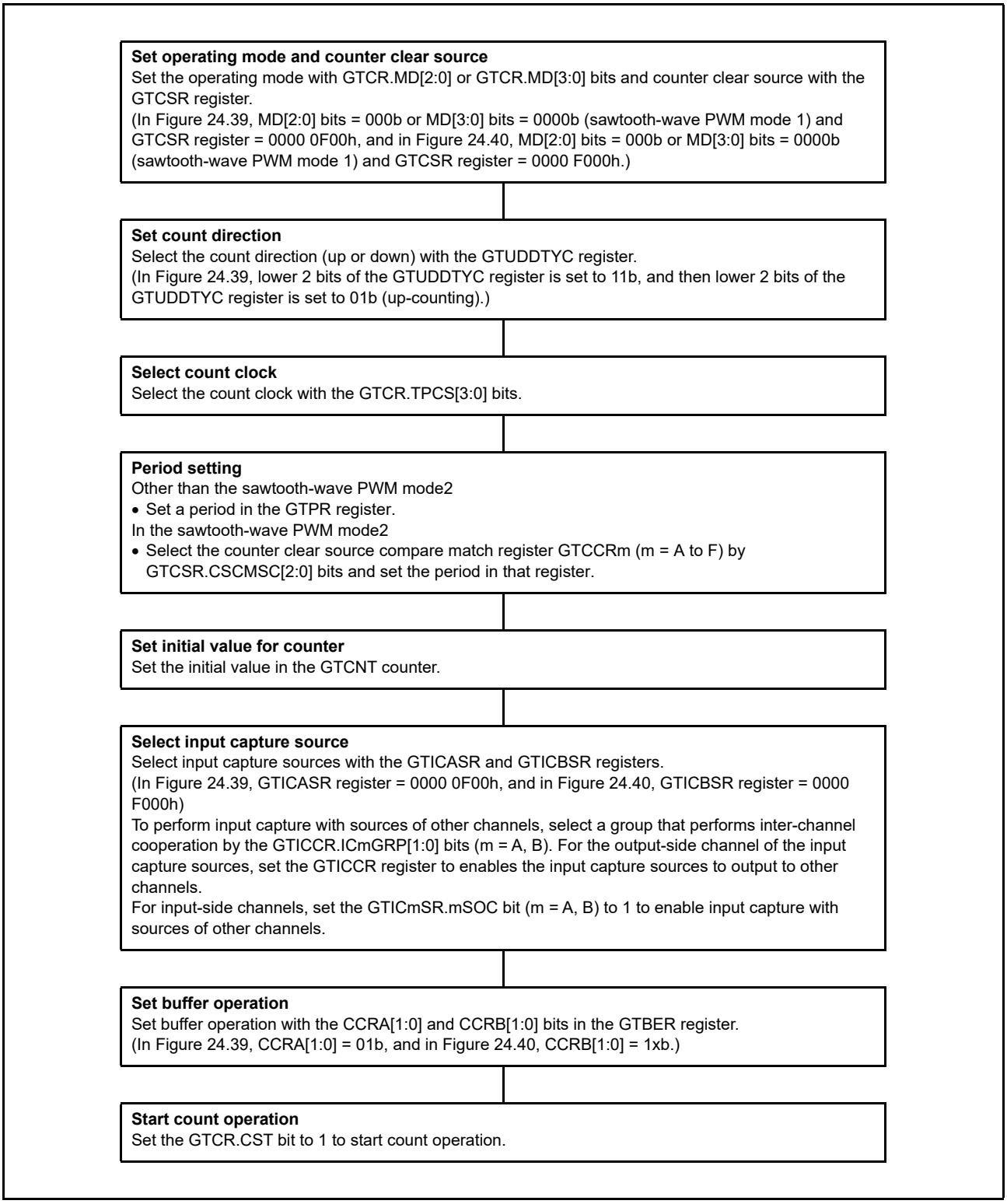


Figure 24.41 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Input Capture)

24.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To enable double buffer operation for the GTADTRA or GTADTRB register, set the GTBER.ADTDA or ADTDB bit to 1, respectively. For single buffer operation, set 0. To disable buffering, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In sawtooth-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to a value other than 00b and in count operation, the buffer transfer by the same counter clearing sources described in section 24.3.2.1, GTPR Register Buffer Operation, is performed in the same way as at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one GTCLK cycle from the GTCCRD register write of slave channel 2.

In sawtooth-wave mode, the buffer transfer of GTADTRm register by counter clearing can be disabled by GTBER2.CCTADm (m = A, B) bit setting.

In sawtooth-wave mode, the buffer transfer of GTADTRm register by its own compare match can be enabled by the GTBER2.CMTADm (m = A, B) bit setting.

Figure 24.42 to Figure 24.46 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Figure 24.47 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

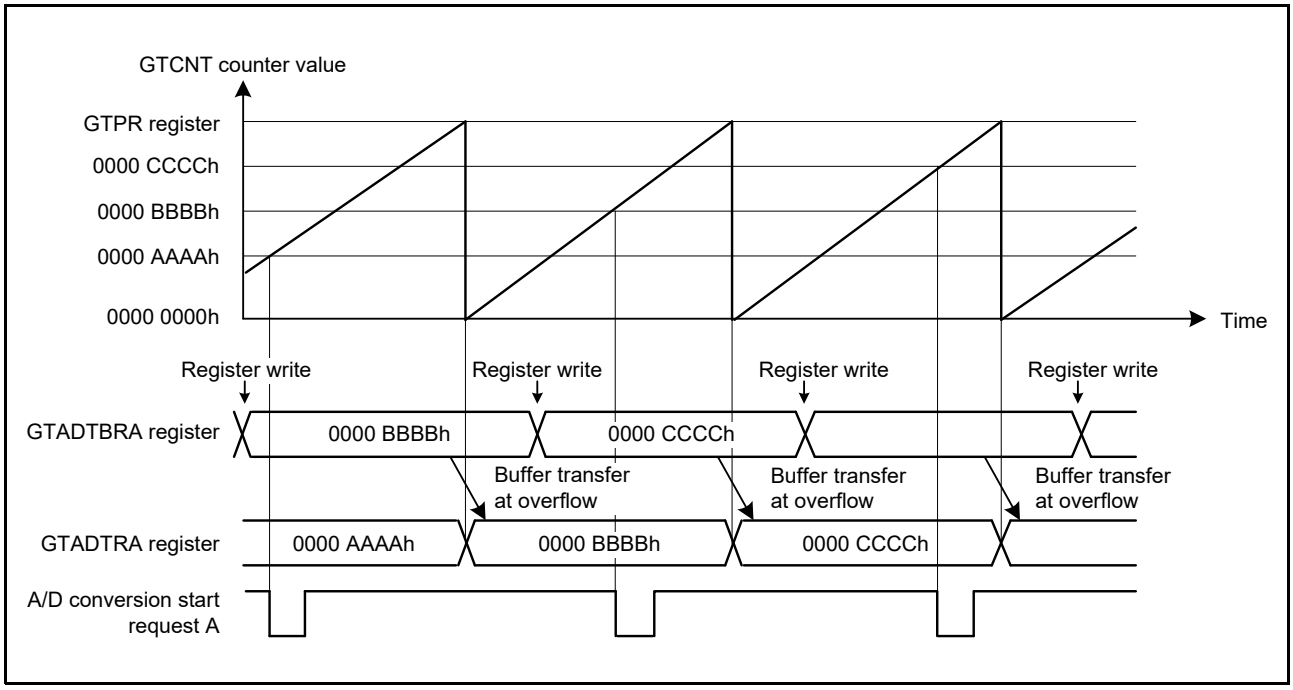


Figure 24.42 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

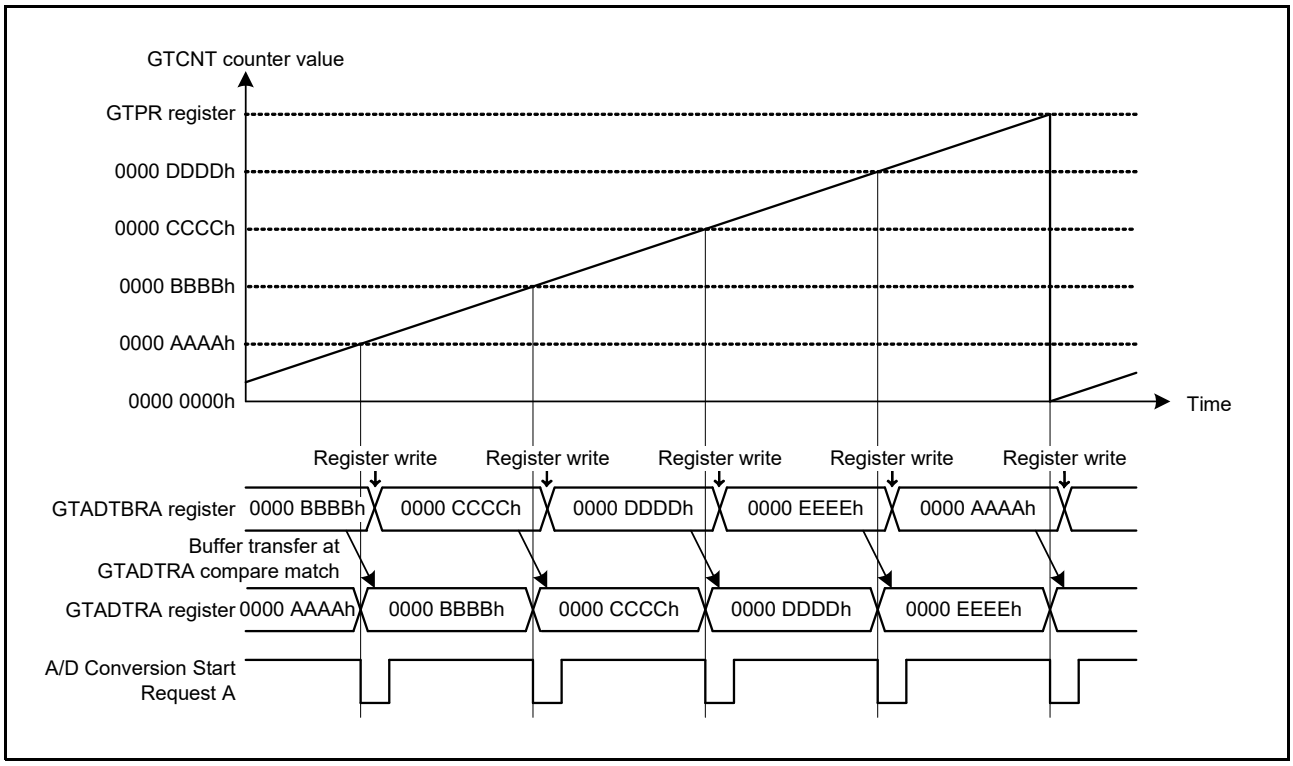


Figure 24.43 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, GTADTRA Register Compare Match for buffer transfer, A/D Conversion Start Request Generated by Up-Counting)

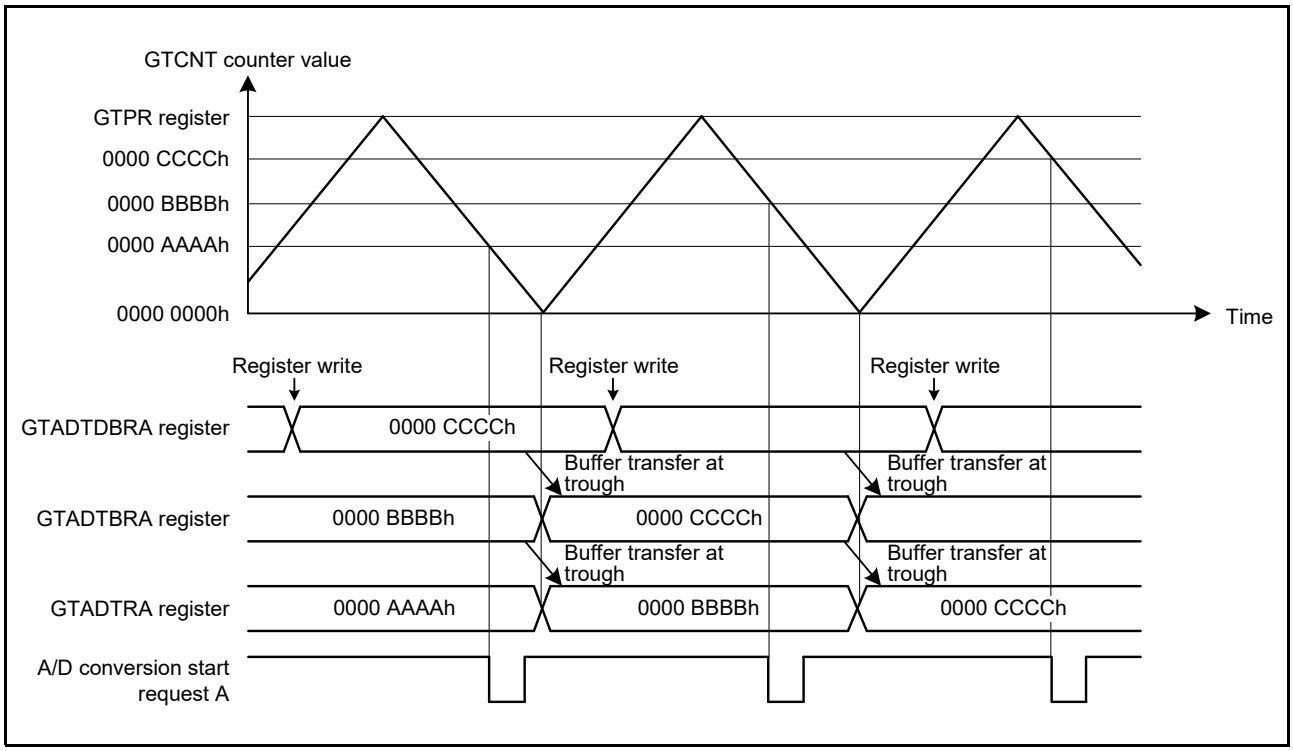


Figure 24.44 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Operation at Trough, A/D Conversion Start Request Generated by Down-Counting)

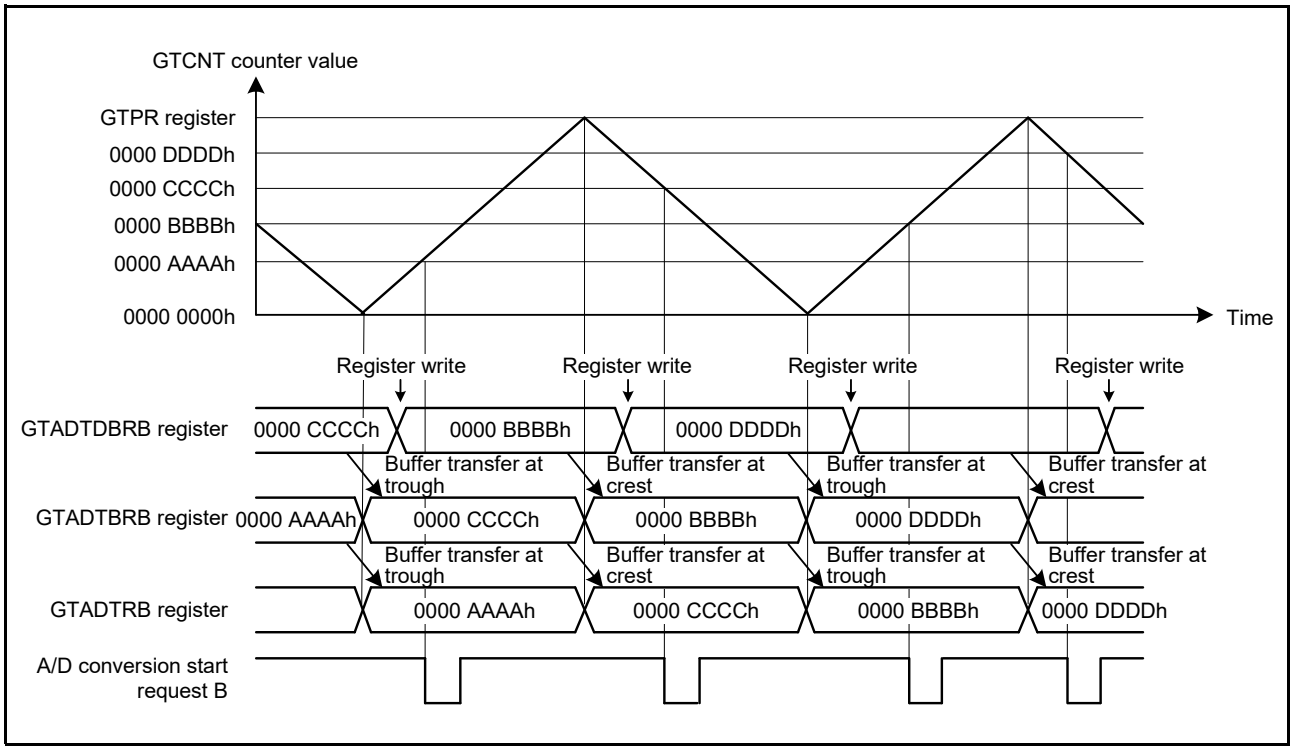


Figure 24.45 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)

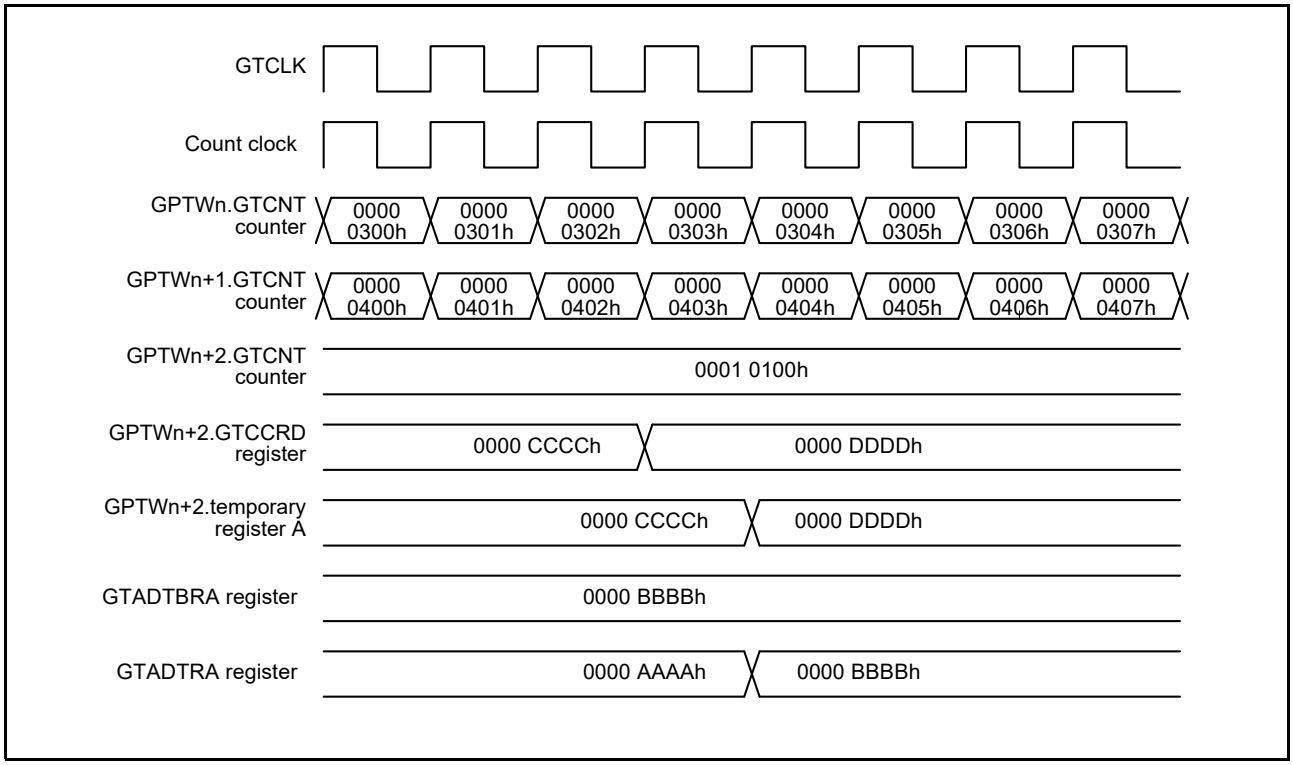


Figure 24.46 Example of Buffer Operation of the GTADTRA and GTADTRB Registers at the GTCCRD Register of Slave Channel 2 Updating in Complementary PWM Mode

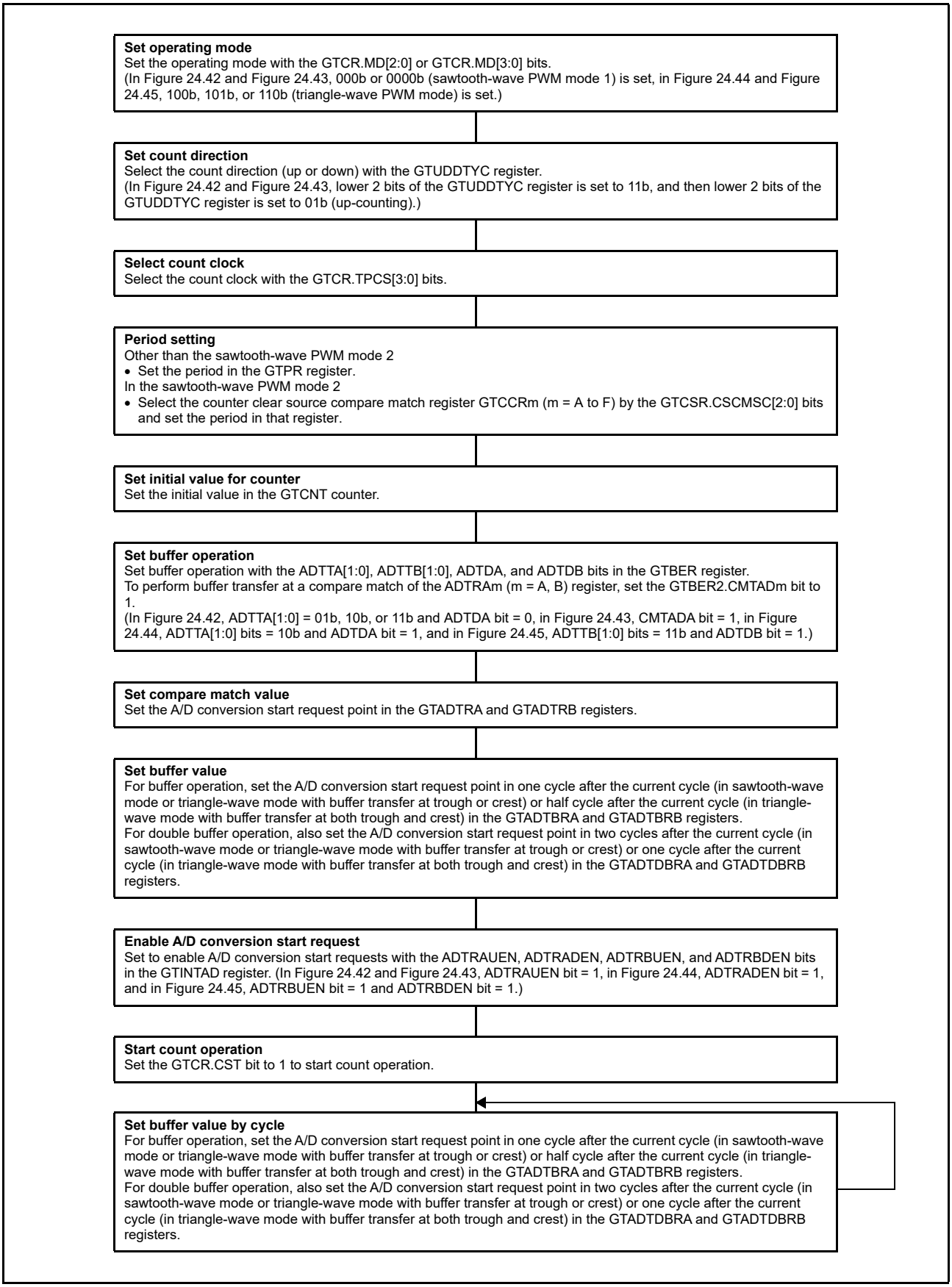


Figure 24.47 Example for Setting Buffer Operation of the GTADTRA and GTADTRB Registers

24.3.2.4 Buffer Operation for the GTIOA[4:0] and GTIOB[4:0] Bits

The GTOLBR.GTIOAB[4:0] bits can function as the buffer register of the GTIOR.GTIOA[4:0] bits and the GTOLBR.GTIOBB[4:0] bits can function as the buffer register of the GTIOR.GTIOB[4:0] bits.
Buffer transfer timing can be set with the GTBER2.OLTTm[1:0] bits (m = A, B). It can be selected from the end of cycle or GTCCR register compare match (in sawtooth-wave mode), from crest, trough, or both crest and trough (in triangle-wave mode and complementary PWM mode). In the case that the GTBER2.OLTTm[1:0] bits are 00b, buffer transfer is not performed.
In complementary PWM mode, the setting to change GTIOA and GTIOB by buffer transfer during the dead time of complementary PWM is prohibited. Therefore, when changing the GTIOA[4:0] and GTIOB[4:0] bits by buffer transfer, it is necessary to set the timings of the crest and trough such that they do not overlap with the dead time and change the GTOLBR.GTIOAB[4:0] and GTOLBR.GTIOBB[4:0] bits. When the timing of buffer transfer is at the crest, set the GTCCRm (m = A, C, E) register to satisfy GTCCRm < GTPR. When the timing of buffer transfer is in the trough, set the GTCCRm register to satisfy GTDVU < GTCCRm.
Figure 24.48 to Figure 24.50 show examples of buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits. Figure 24.51 shows an example for setting buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits.

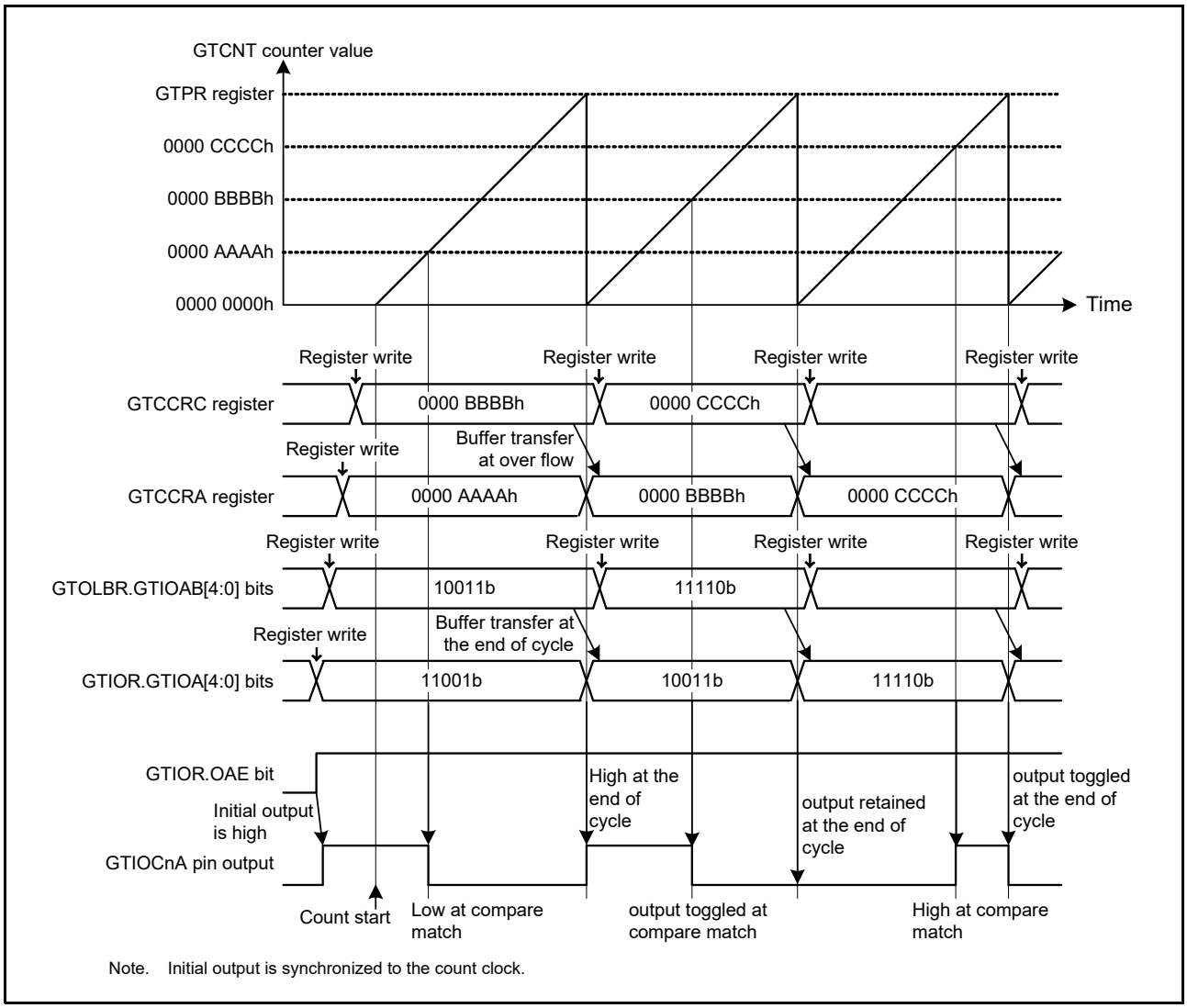
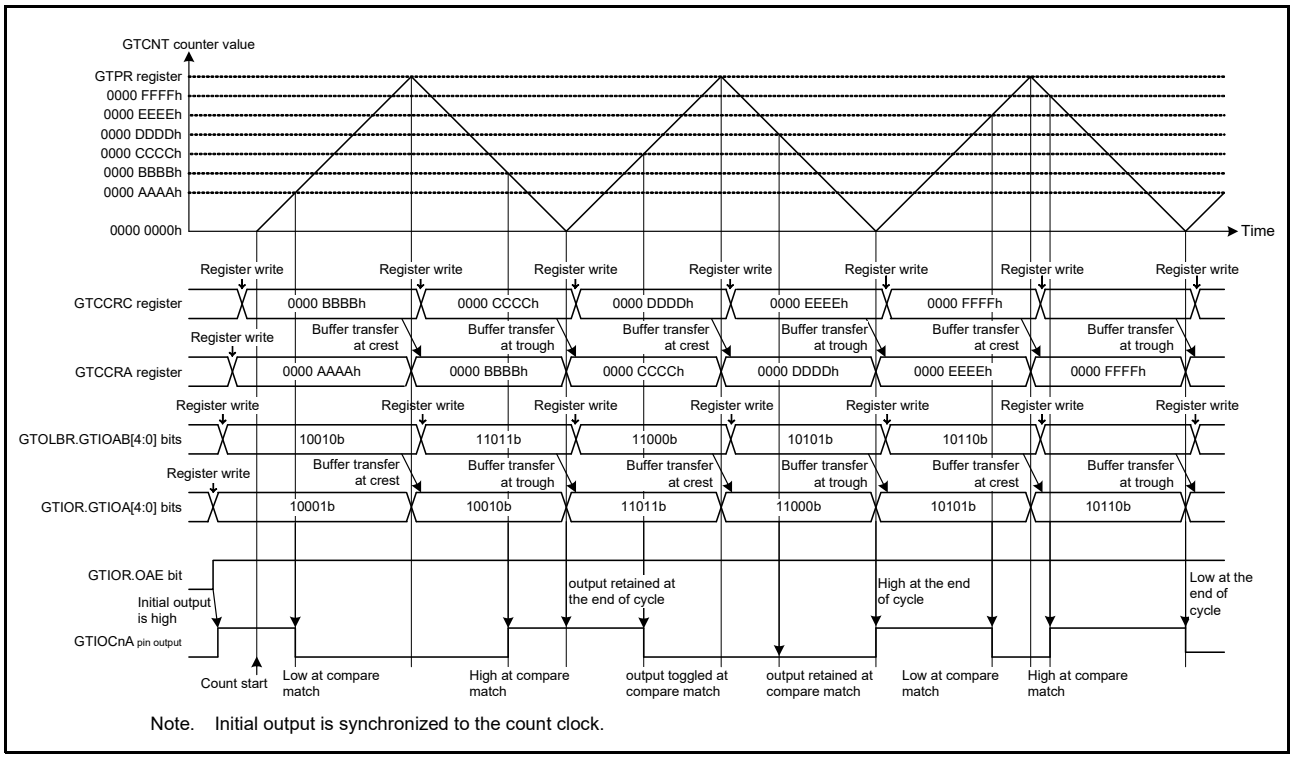
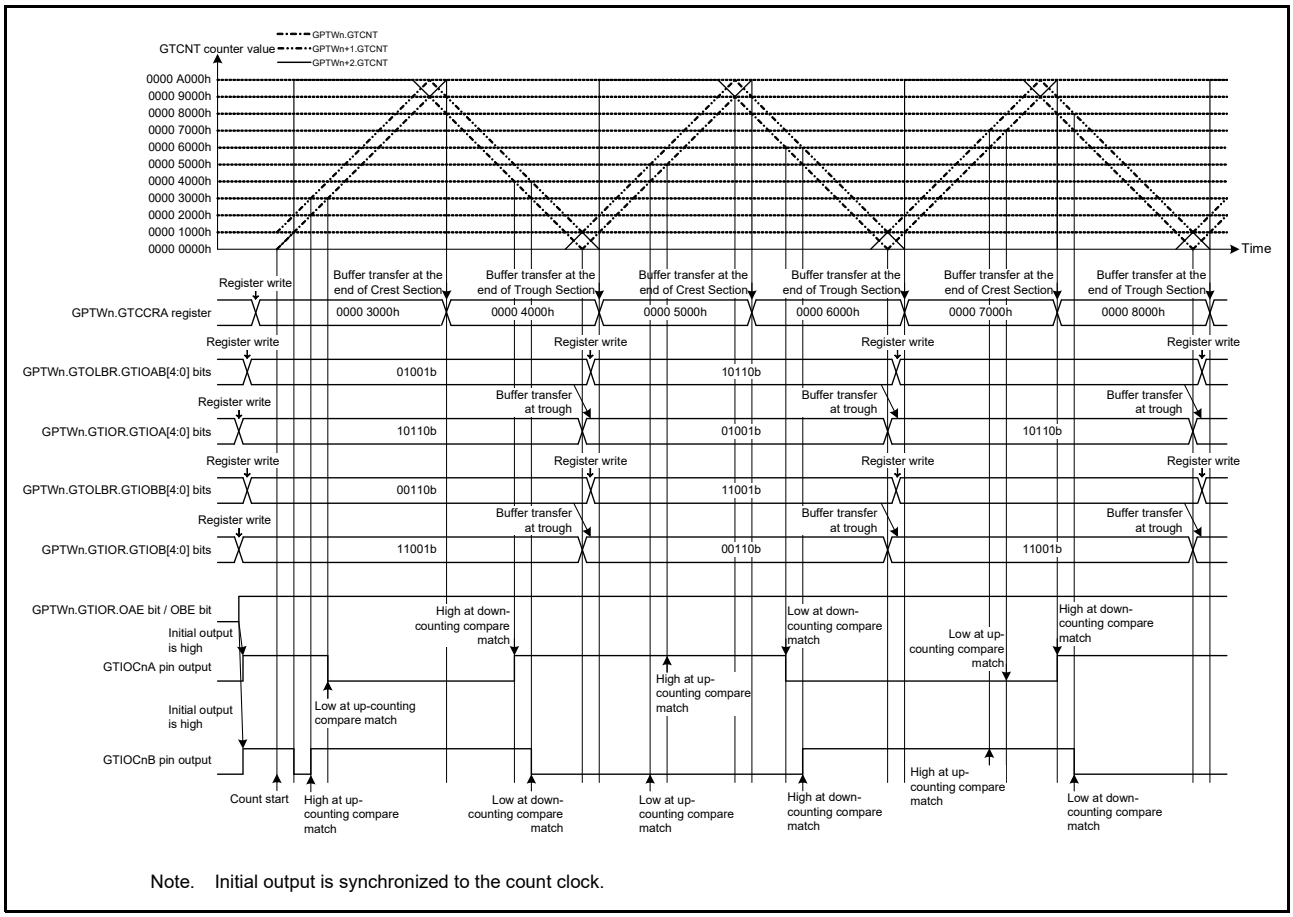


Figure 24.48 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Up-Counting in Sawtooth-Wave PWM Mode 1, Buffer Transfer at the End of Period)



Note. Initial output is synchronized to the count clock.

Figure 24.49 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Triangle-Waves PWM Mode 2, Buffer Transfer at Both Troughs and Crests)



Note. Initial output is synchronized to the count clock.

Figure 24.50 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Complementary PWM Mode 3, Buffer Transfer at Troughs)

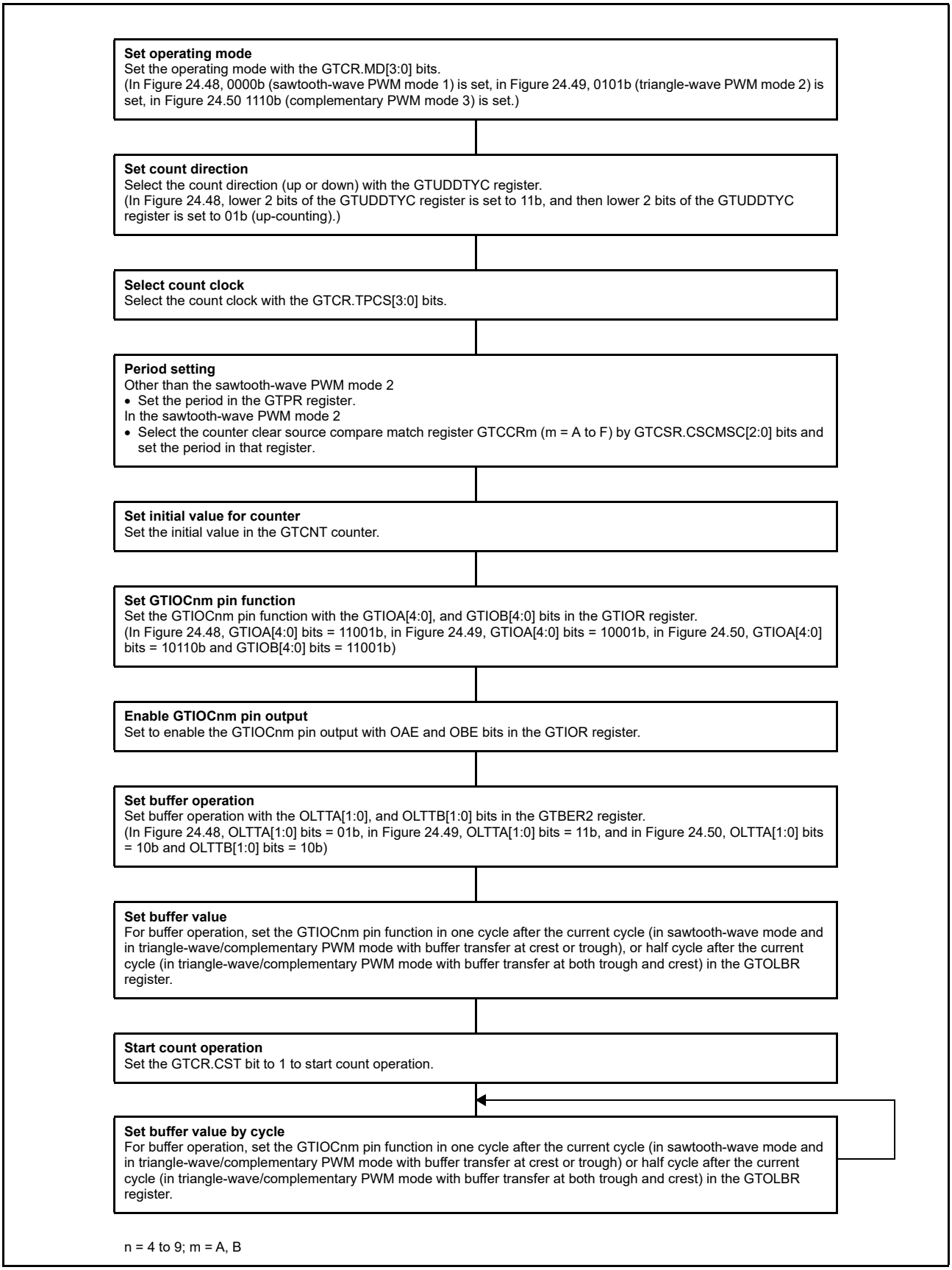


Figure 24.51 Example for Setting Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits

24.3.3 PWM Output Operating Mode

The GPTW can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin (n = 0 to 15) by a compare match between the GTCNT counter and the GTCCRA or GTCCRB register.
By setting the GTDTCR, GTDVU, and GTDVD registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.
In sawtooth-wave modes other than sawtooth-wave PWM mode 2, in triangle-wave mode, or in the master channel of complementary PWM mode, a signal synchronized with the PWM cycle is output from the GTCPPOn terminal when the GTIOR.PSYE bit is set to 1. The GTCPPOn output is toggled at the end of the cycle in sawtooth-wave mode, or at the crest/trough/GTCNT counter clearing timing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPOn is low, and it becomes high when the count starts.

24.3.3.1 Sawtooth-Wave PWM Mode 1

In sawtooth-wave PWM mode 1, the GTCNT counter performs sawtooth-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
When 0 is set in the GTIOR.OmEOCD (m = A, B) bit and the timing of the end of cycle and the timing of the GTCCRm register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOm[3:2] bits.
When 1 is set in the GTIOR.OmEOCD bit, GTIOCnm output is retained.
Figure 24.52 shows an example of sawtooth-wave PWM mode 1 operation, and Figure 24.53 shows an example for setting sawtooth-wave PWM mode 1.

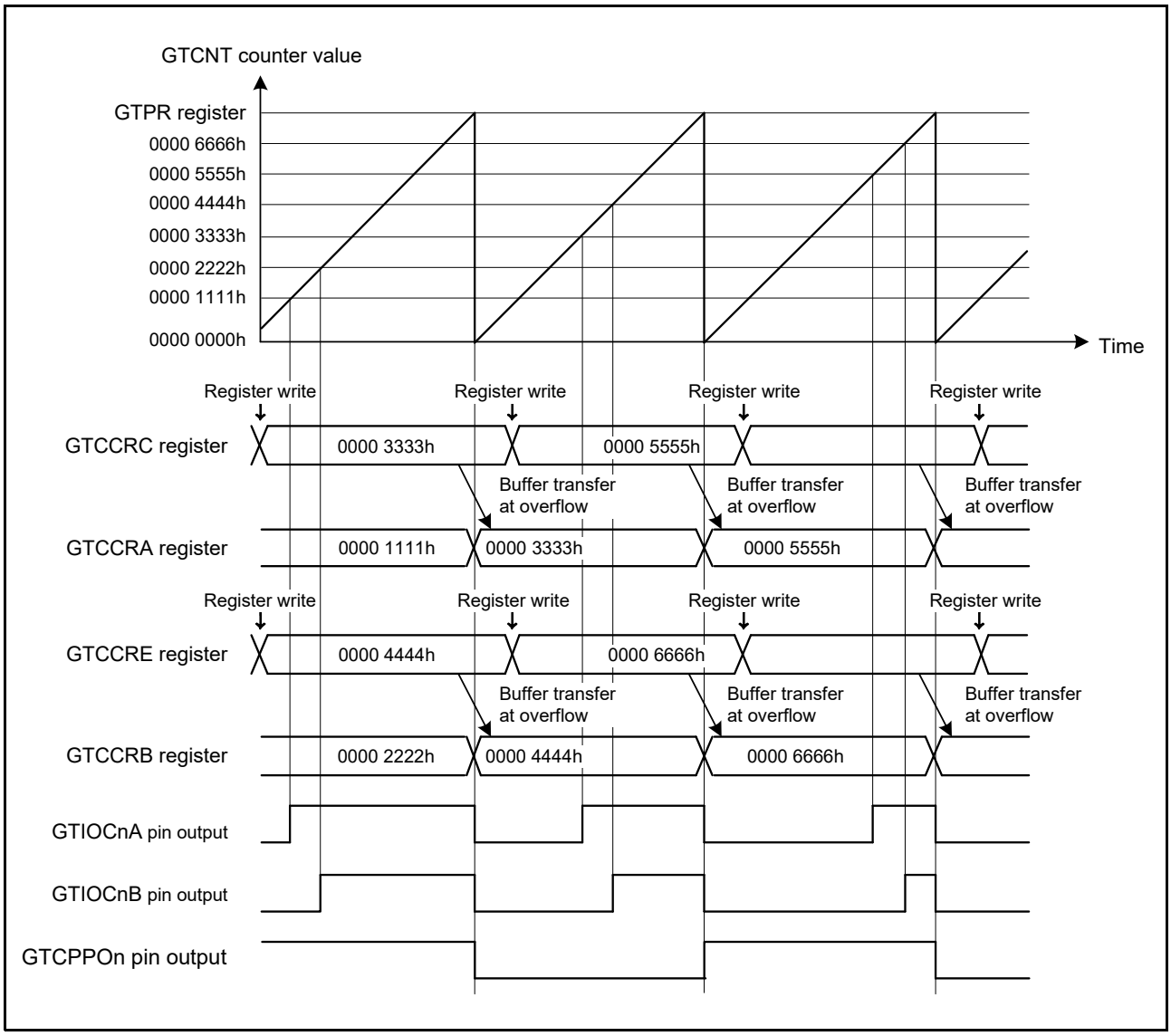
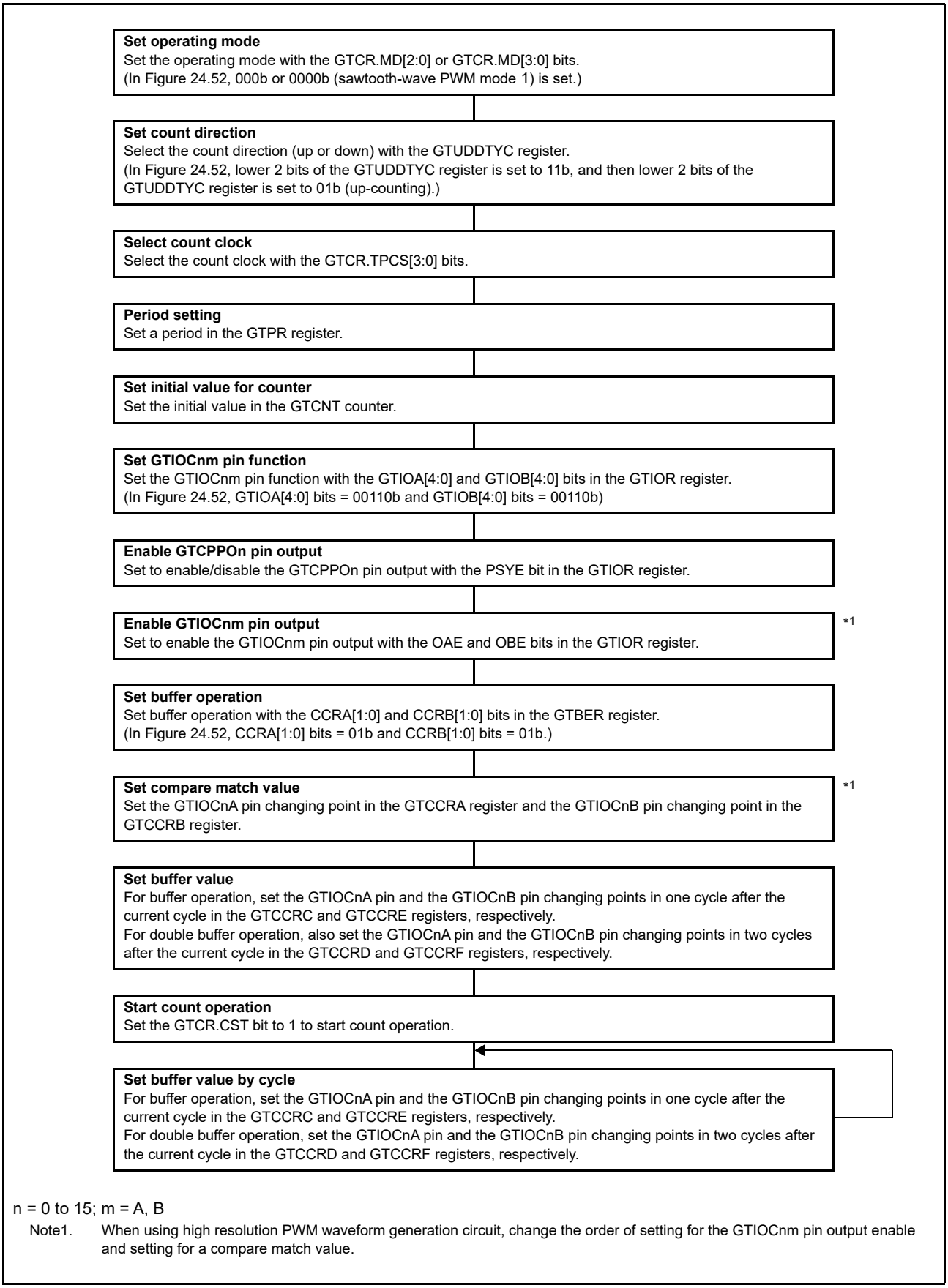


Figure 24.52 Example of Sawtooth-Wave PWM Mode 1 Operation (Up-Counting, Buffer Operation, High Output on Compare Match between the GTCCRA and GTCCRB Registers, Low Output at the End of the Cycle, GTIOR.PSYE bit = 1 (n = 0 to 15))



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.53 Example for Setting Sawtooth-Wave PWM Mode 1

24.3.3.2 Sawtooth-Wave PWM Mode 2

The sawtooth-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a sawtooth wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers.

The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA (n = 0 to 2, 4 to 6) pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed.

If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRm (m = A, B) register compare match, PWM output operation is performed at the end of cycle (in the case of the GTIOR.OmEOCD bit = 0) or the PWM output is retained (in the case of GTIOR.OmEOCD bit = 1).

Figure 24.54 to Figure 24.56 show examples of sawtooth-wave PWM mode 2 operation. Figure 24.57 shows an example for setting sawtooth-wave PWM mode 2.

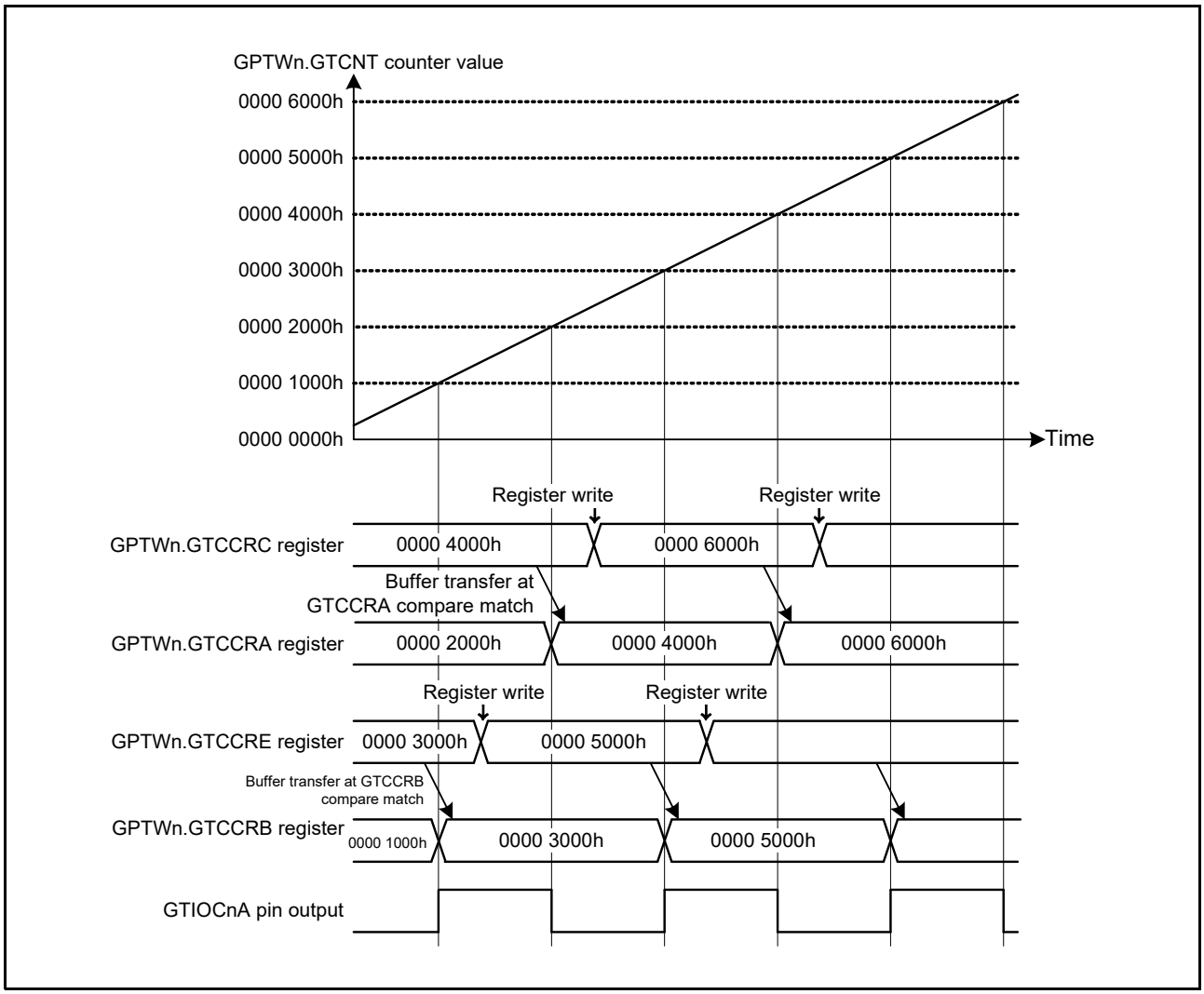


Figure 24.54 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Single Buffer Operation, and No Clear Setting) (n = 4 to 9)

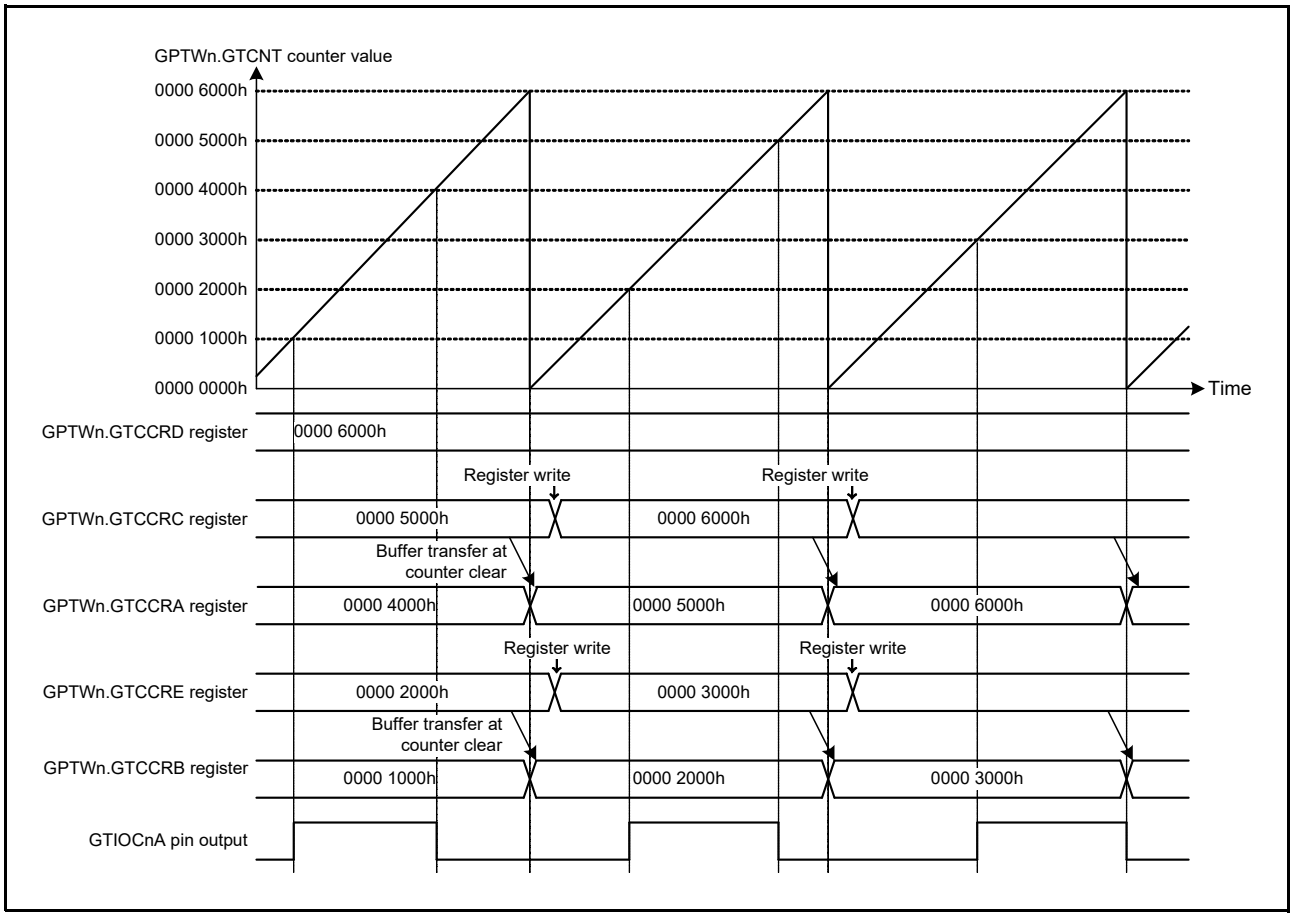


Figure 24.55 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of the Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and the GTIOR.OAEOCD Bit = 0) (n = 4 to 9)

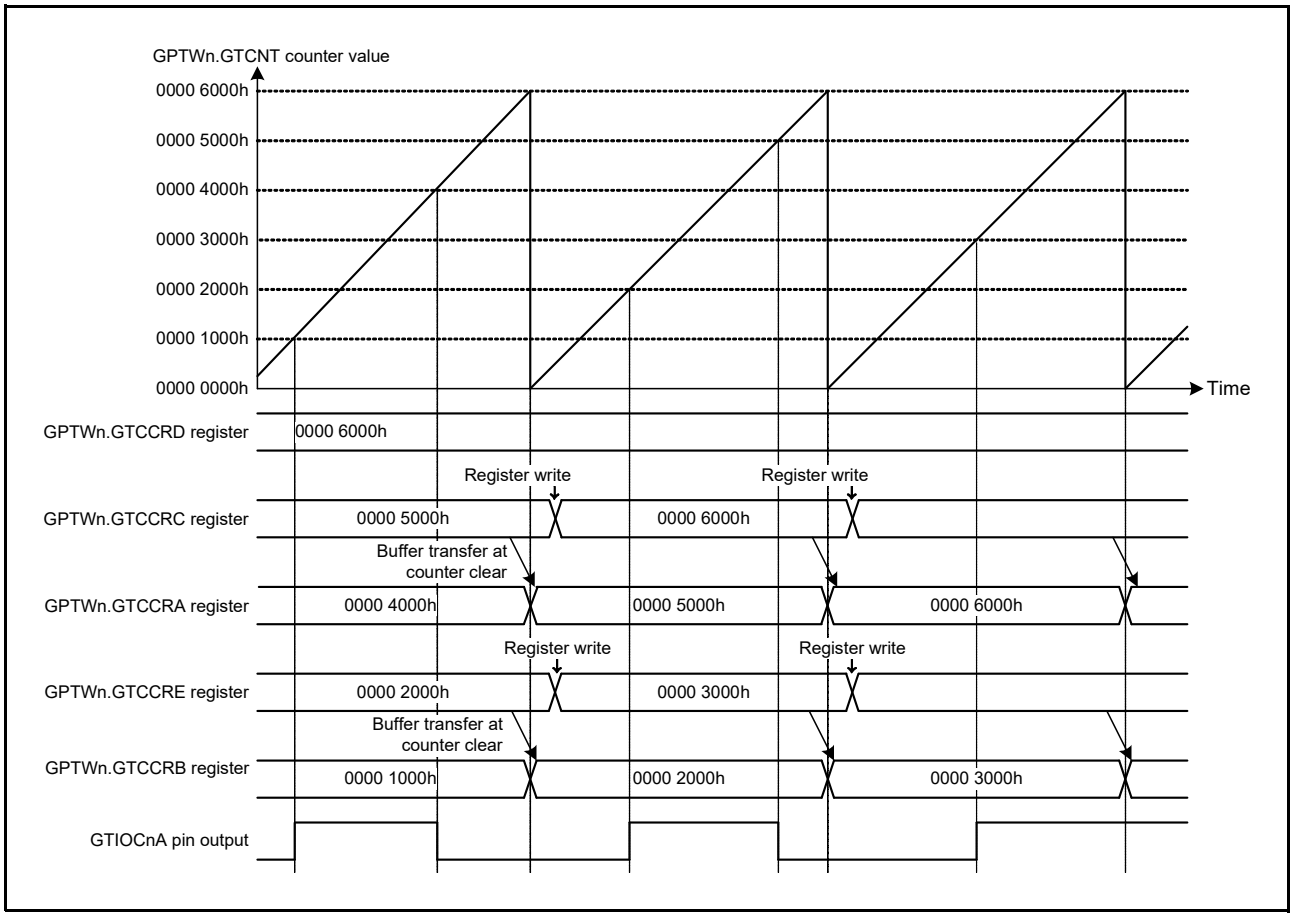
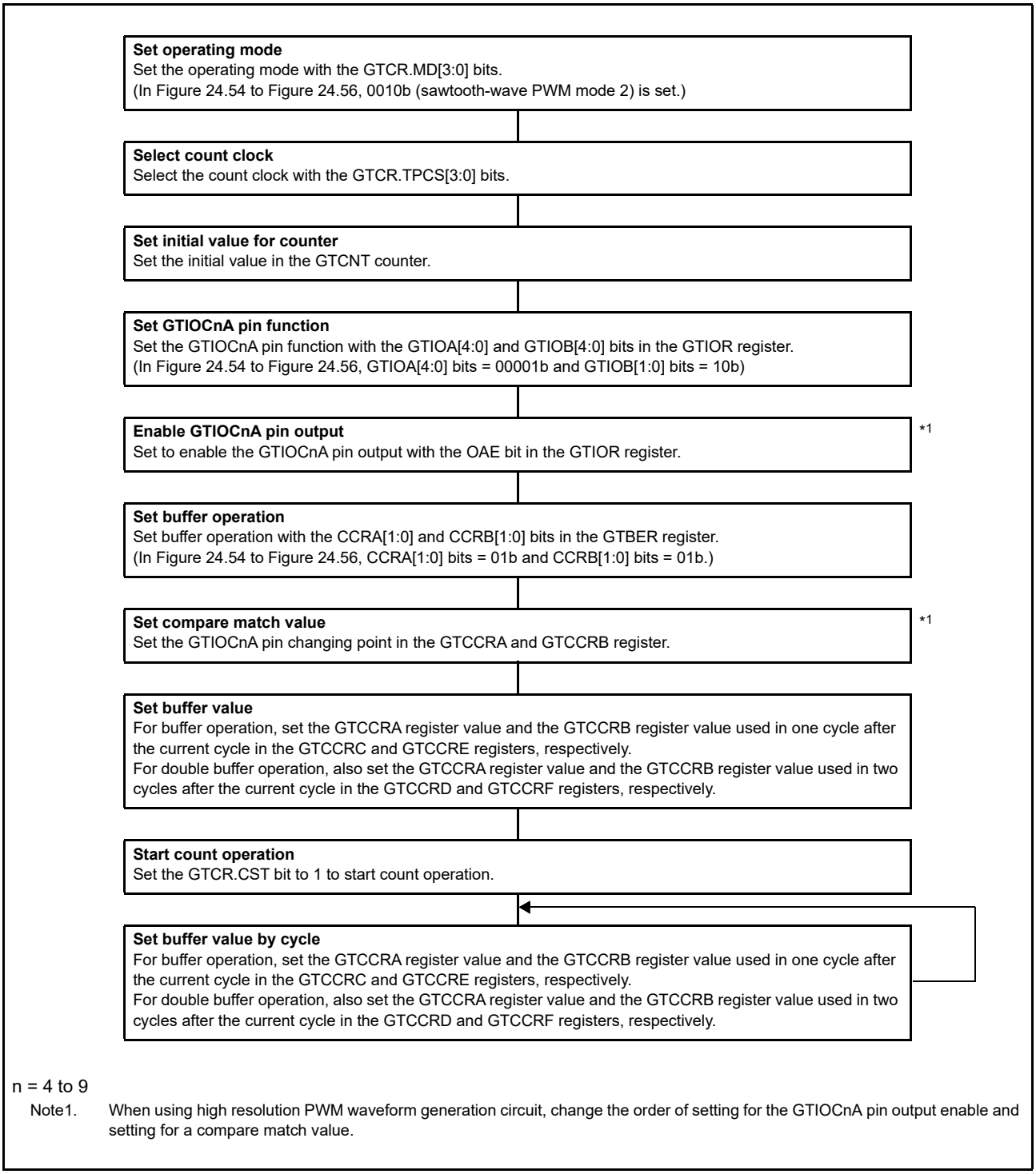


Figure 24.56 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and GTIOR.OAEOCD Bit = 1) (n = 4 to 9)



n = 4 to 9

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnA pin output enable and setting for a compare match value.

Figure 24.57 Example for Setting Sawtooth-Wave PWM Mode 2

24.3.3.3 Sawtooth-Wave One-Shot Pulse Mode

The sawtooth-wave one-shot pulse mode is a mode in which the period is set in the GTPR register, the GTCNT counter performs sawtooth-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 15).

Buffer operation in sawtooth-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the end of the cycle, and from the temporary register A to the GTCCRA register at a GTCCRA register compare match and from the temporary register B to the GTCCRB register at a GTCCRB register compare match. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR register setting.

The temporary register A and the temporary register B can perform forcible buffer transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B respectively by writing 1 to the GTBER.CCRSWT bit while the count is stopped.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm (m = A, B) registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 24.58 shows an example of sawtooth-wave one-shot pulse mode operation, and Figure 24.59 shows an example for setting sawtooth-wave one-shot pulse mode.

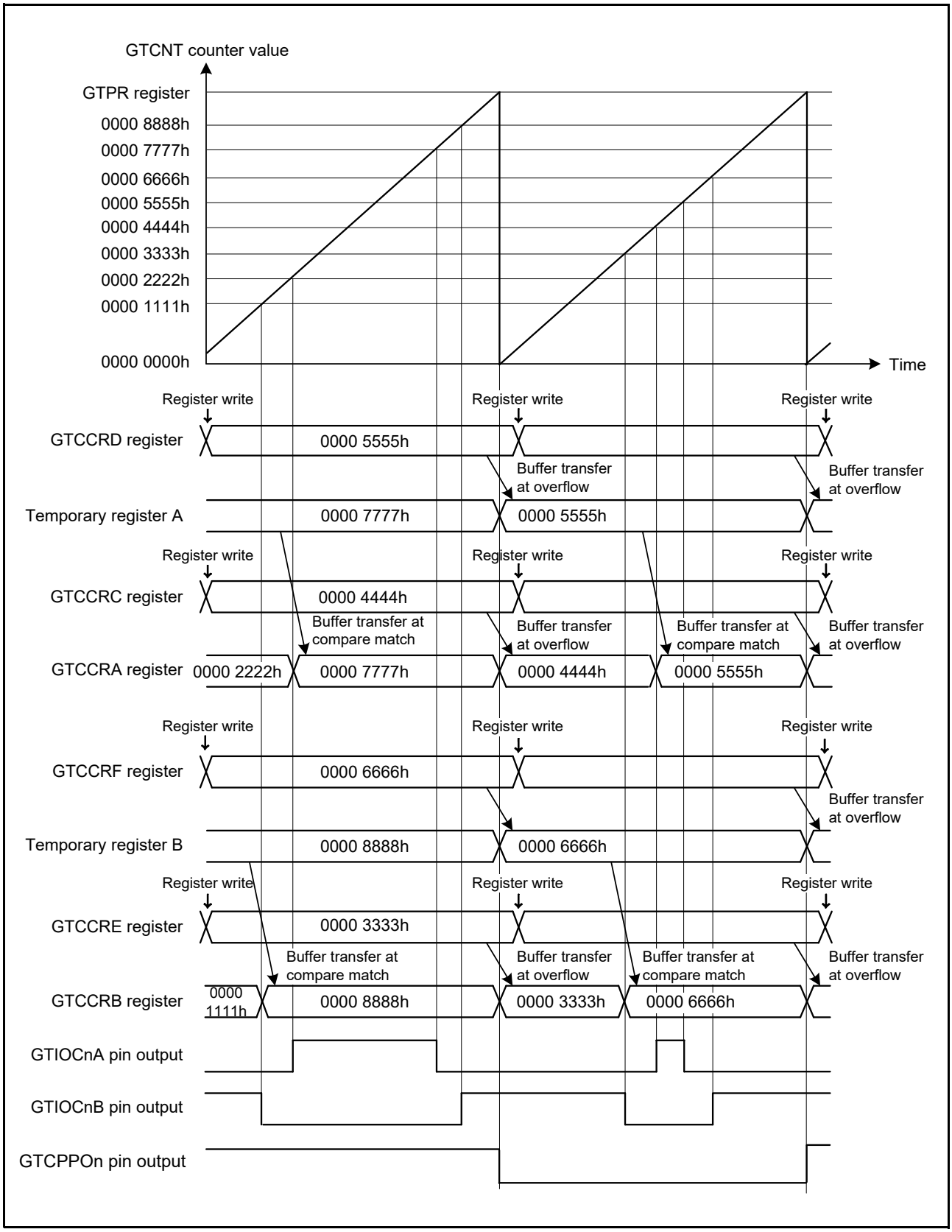
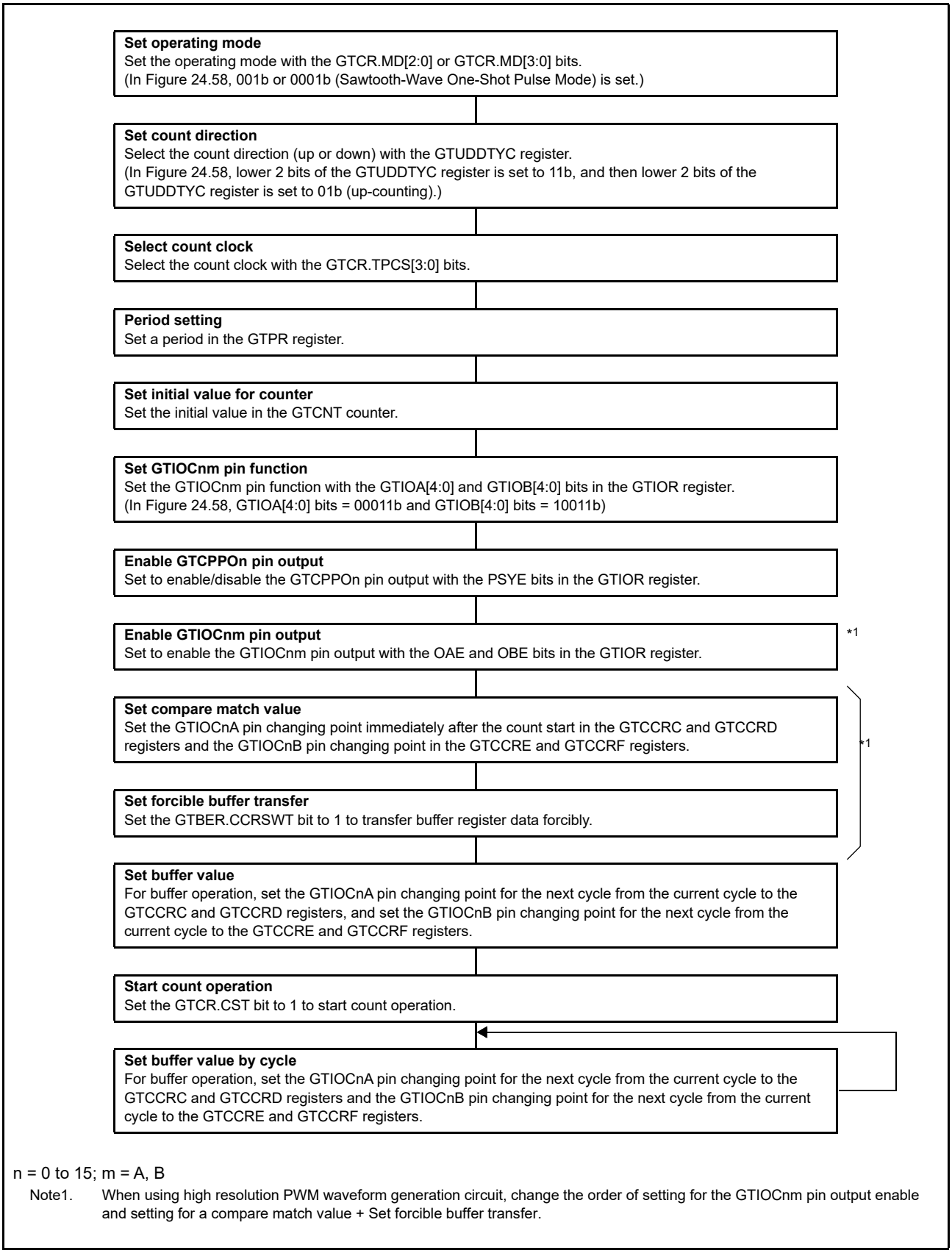


Figure 24.58 Example of Sawtooth-Wave One-Shot Pulse Mode Operation (Up-Counting, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE bit = 1 (n = 0 to 15))



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value + Set forcible buffer transfer.

Figure 24.59 Example for Setting Sawtooth-Wave One-Shot Pulse Mode

24.3.3.4 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 15).
Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
Figure 24.60 shows an example of triangle-wave PWM mode 1 operation, and Figure 24.61 shows an example for setting triangle-wave PWM mode 1.

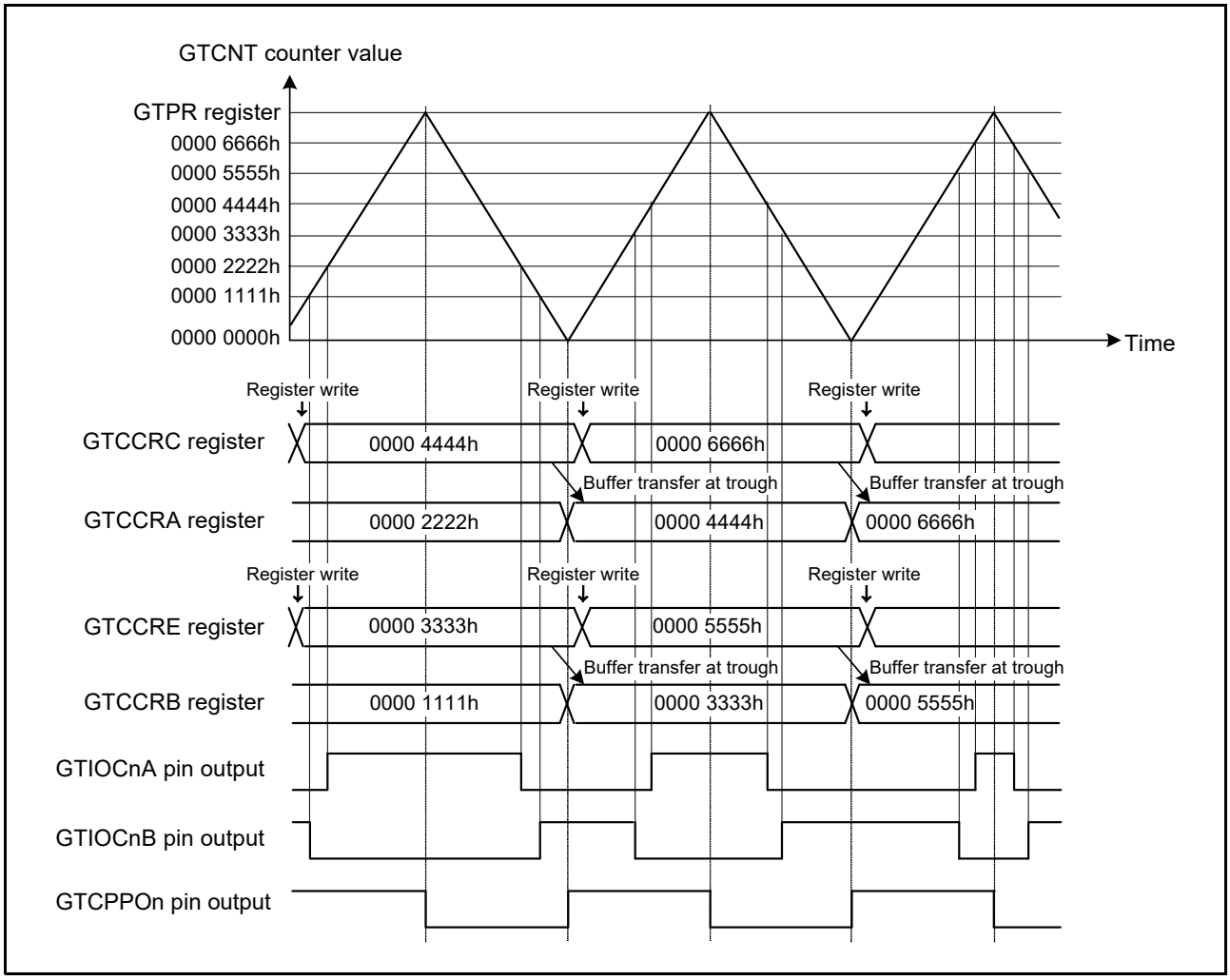
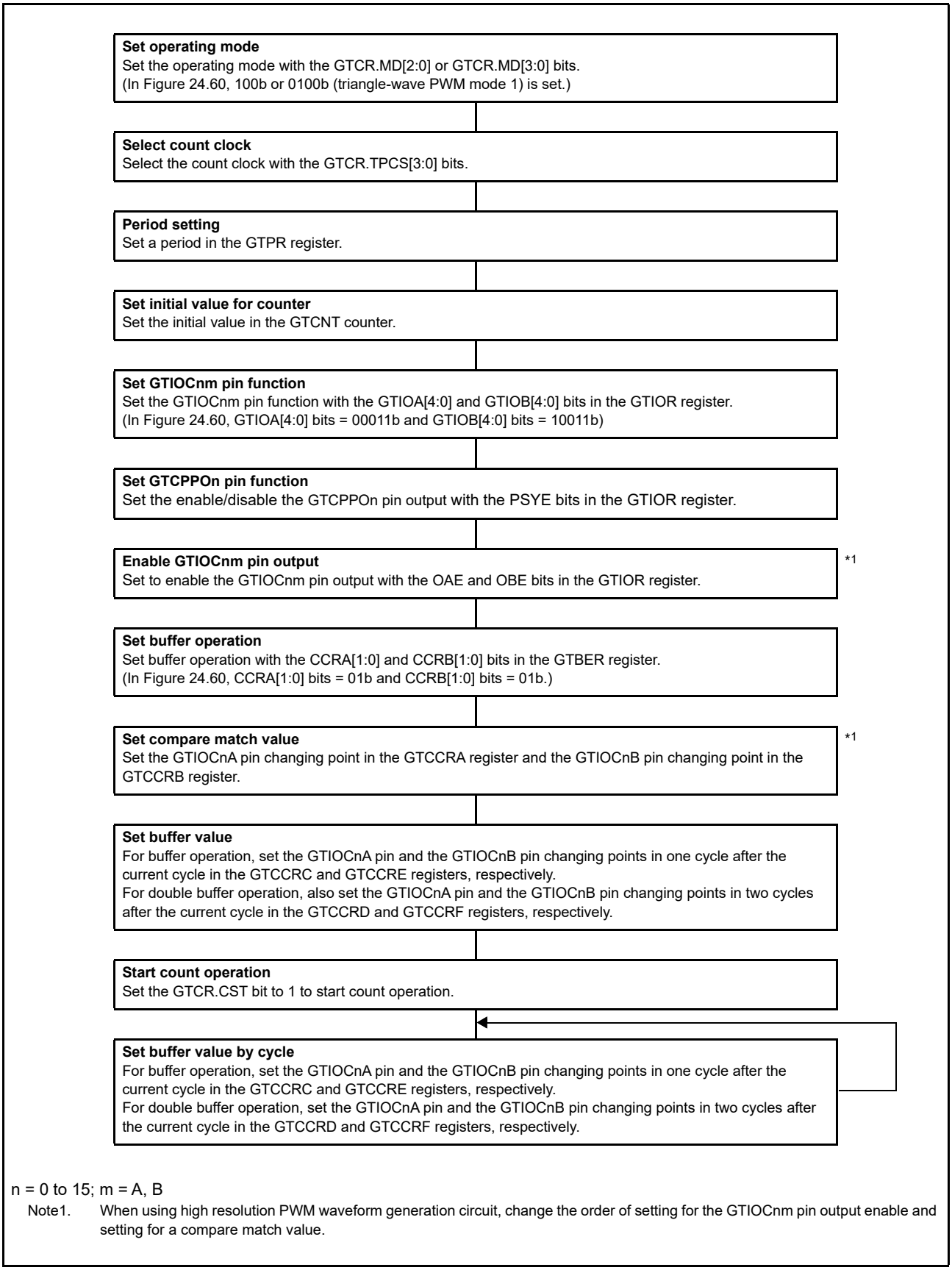


Figure 24.60 Example of Triangle-Wave PWM Mode 1 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE bit = 1 (n = 0 to 15))



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.61 Example for Setting Triangle-Wave PWM Mode 1

24.3.3.5 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 15). The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
Figure 24.62 shows an example of triangle-wave PWM mode 2 operation, and Figure 24.63 shows an example for setting triangle-wave PWM mode 2.

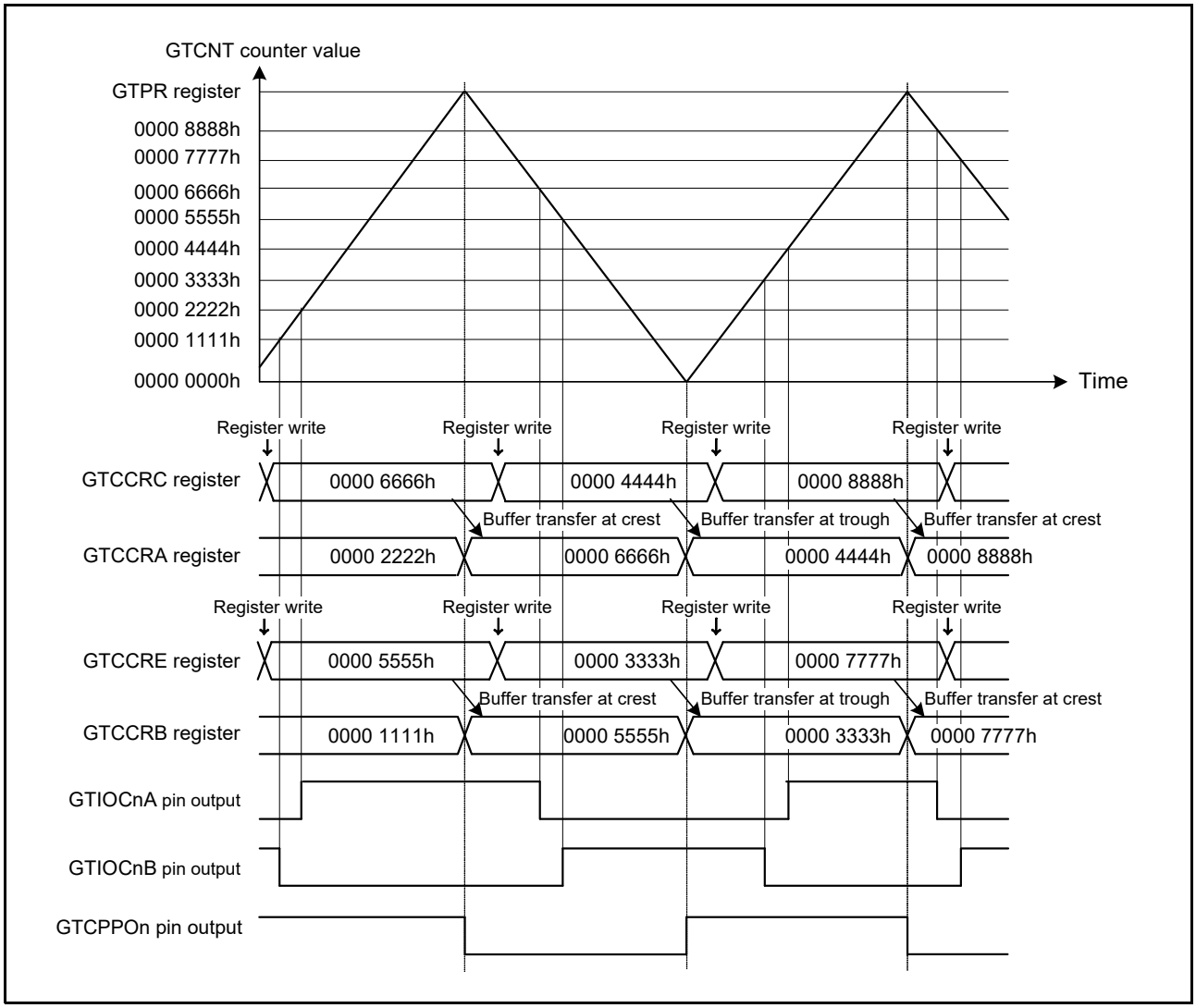
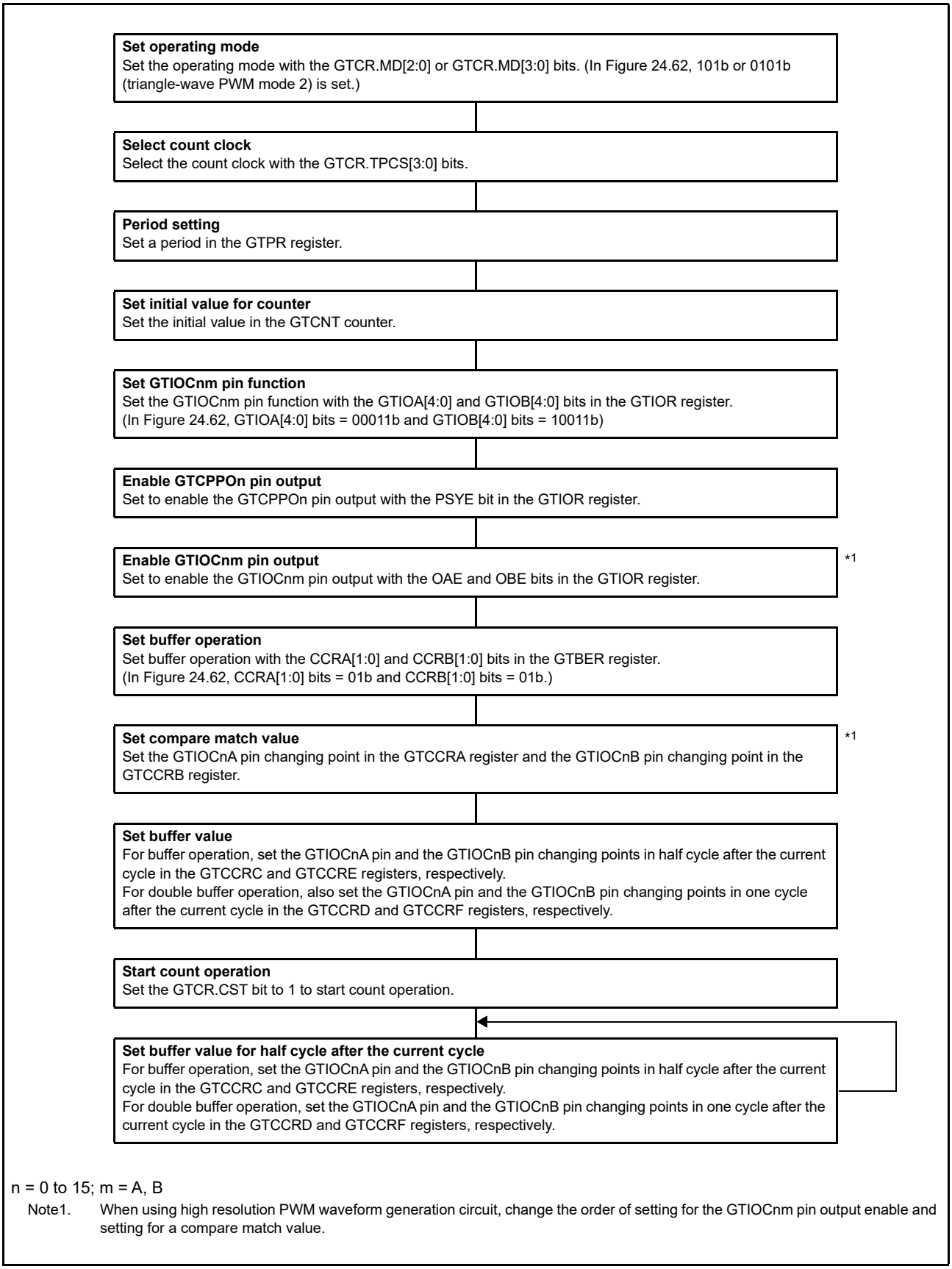


Figure 24.62 Example of Triangle-Wave PWM Mode 2 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE bit = 1 (n = 0 to 15))



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.63 Example for Setting Triangle-Wave PWM Mode 2

24.3.3.6 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 15). Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the trough, and from the temporary register A to the GTCCRA register and from the temporary register B to the GTCCRB register at the crest. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm (m = A, B) registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 24.64 shows an example of triangle-wave PWM mode 3 operation, and Figure 24.65 shows an example for setting triangle-wave PWM mode 3.

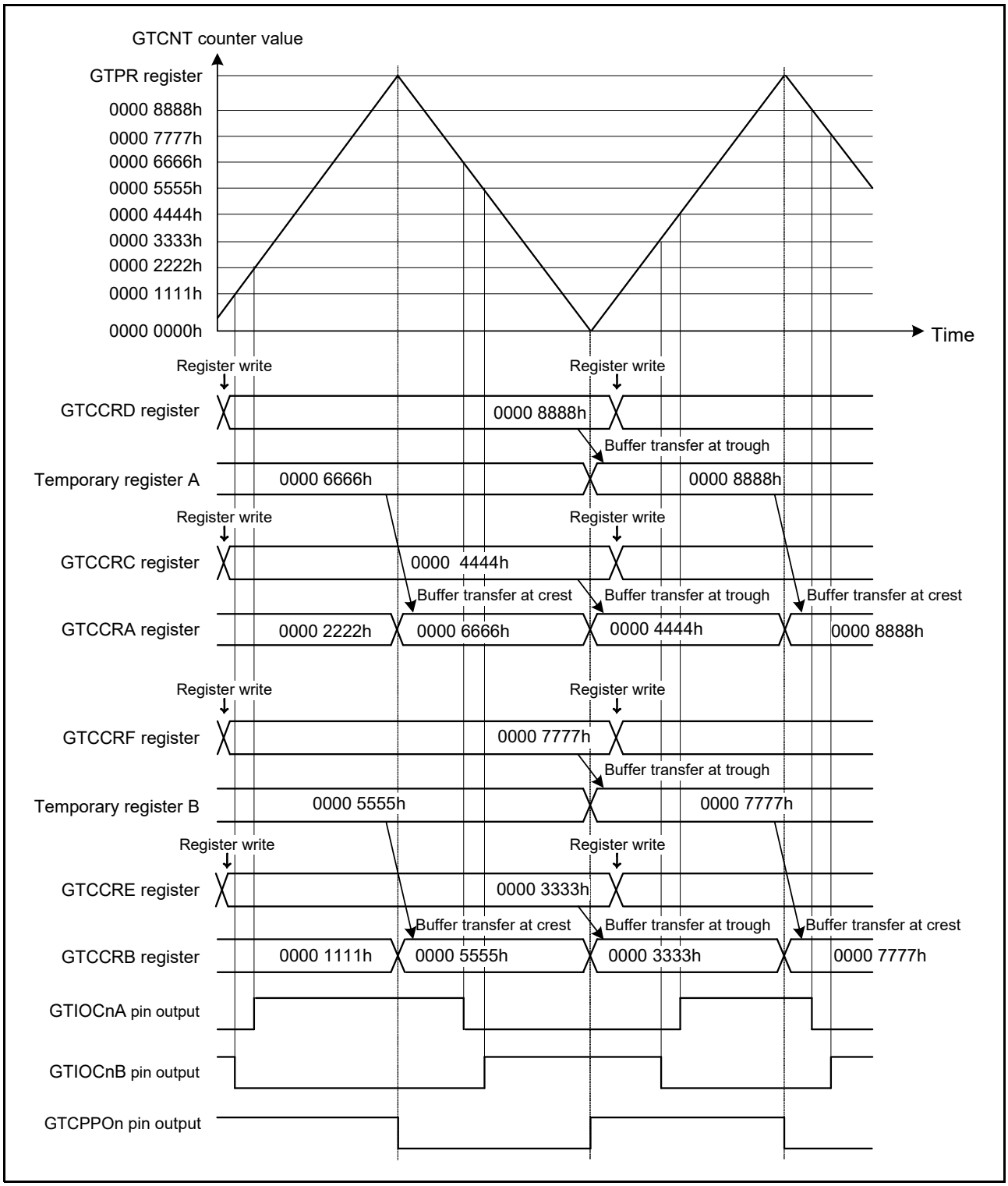
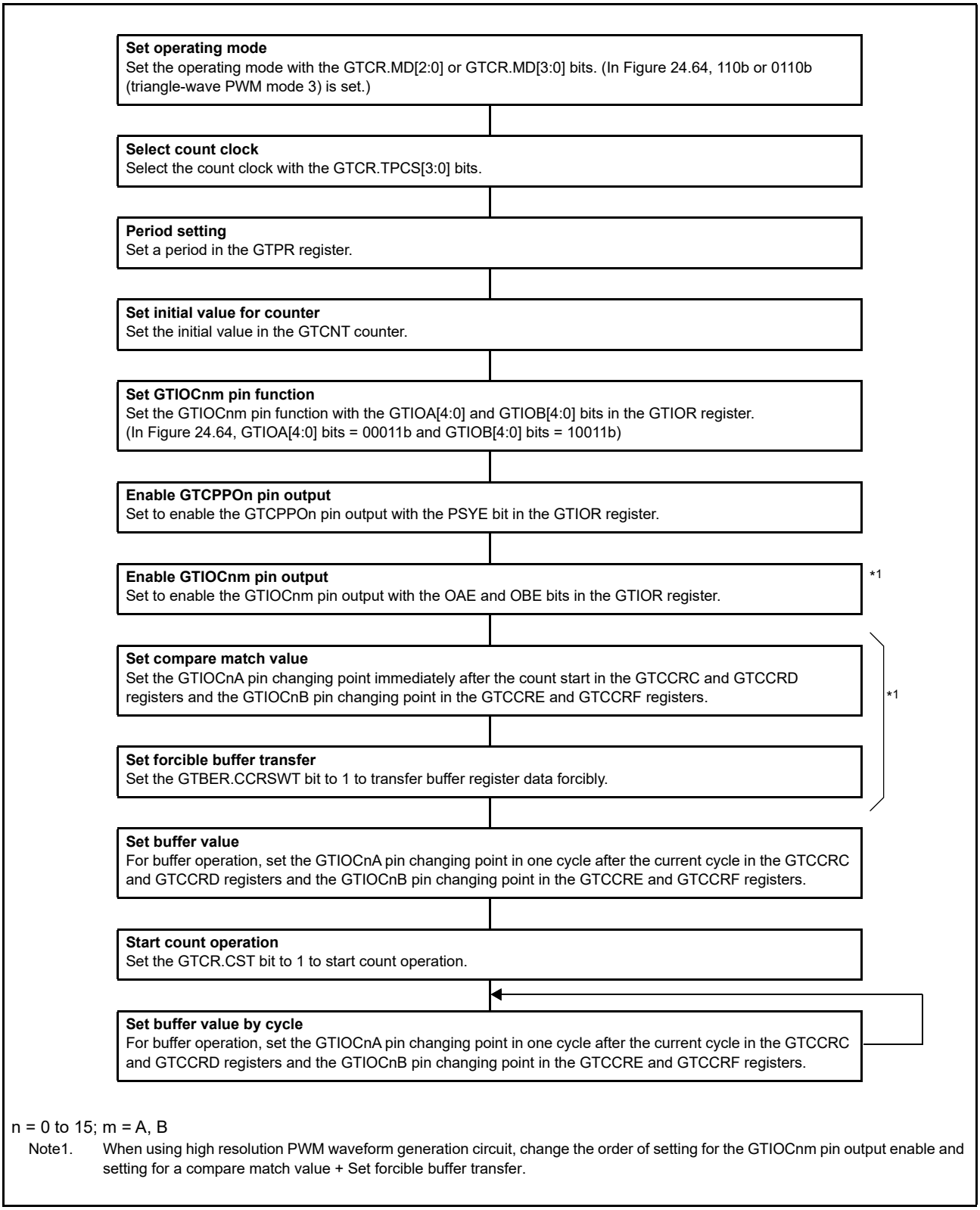


Figure 24.64 Example of Triangle-Wave PWM Mode 3 Operation (Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE bit = 1 (n = 0 to 15))



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value + Set forcible buffer transfer.

Figure 24.65 Example for Setting Triangle-Wave PWM Mode 3

24.3.3.7 Complementary PWM mode 1, 2, 3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on differences in buffer operation: (1) complementary PWM mode 1 (transfer at crests), (2) complementary PWM mode 2 (transfer at troughs), (3) complementary PWM mode 3 (transfer at crests and troughs), and (4) complementary PWM mode 4 (immediate transfer).

Figure 24.66 shows the block diagram in complementary PWM modes 1 to 3.

Among consecutive three channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel.

In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCnA pin (n = 4 to 9) and the GTIOCnB pin (n = 4 to 9) respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register. In complementary PWM mode 3, setting the GTBER2.CP3DB bit to 1 also enables buffer operation of the GTCCRA register by the GTCCRE register, temporary register B, and GTCCRF register, allowing double buffer operation.

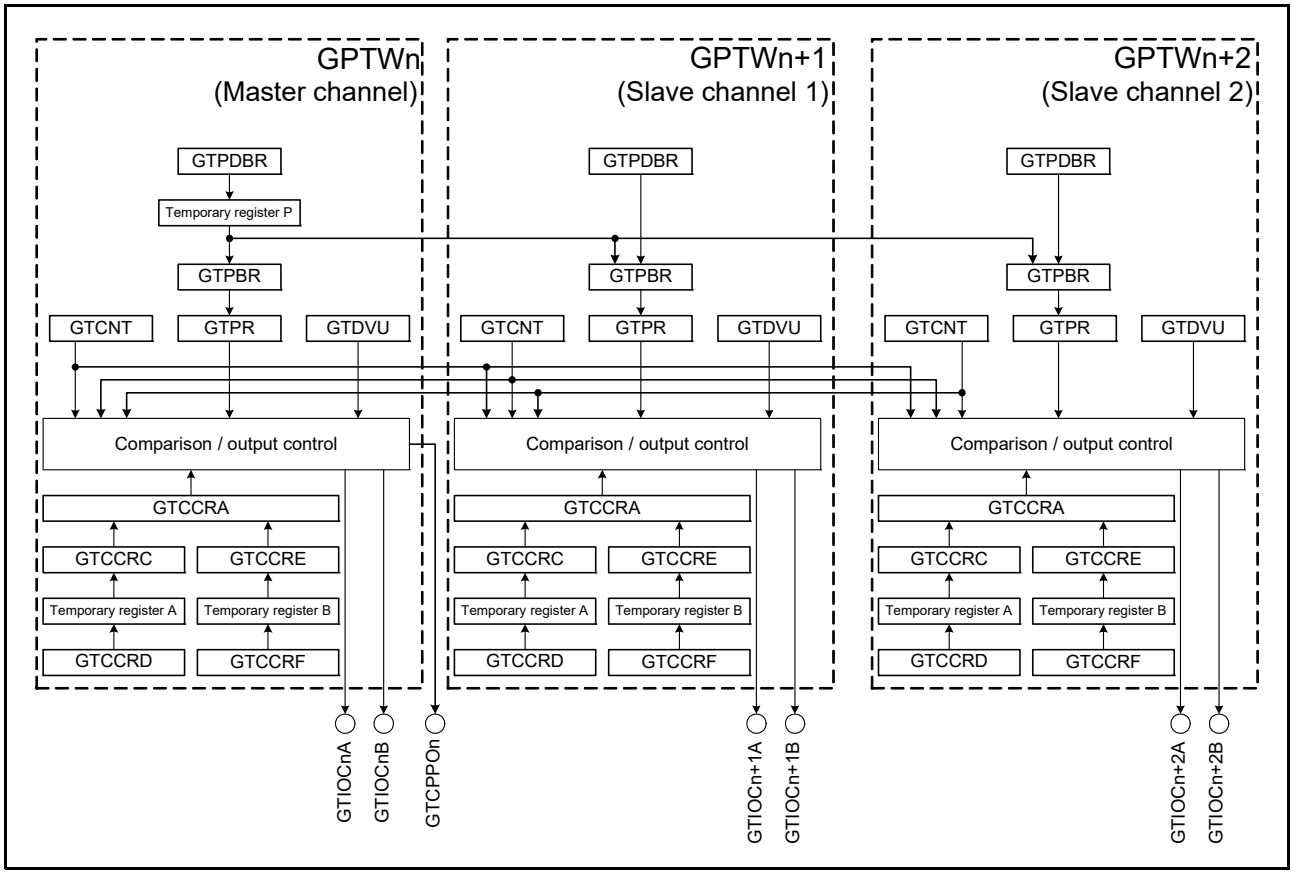


Figure 24.66 Block Diagram in Complementary PWM Mode 1, 2, 3 (n = 4, 7)

The GPTWn.GTCNT counter performs count operation for triangle waves using the GPTWn.GTPR register as a cycle register. A section where the GPTWn.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPTWn+1.GTCNT counter performs count operation with the value (GPTWn.GTCNT counter value + dead time value set in the GPTWn.GTDVU register). A section where the GPTWn+1.GTCNT counter value is larger than the

GPTWn.GTPR register value is referred to as crest section.
Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction.
A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.
The GPTWn+2.GTCNT counter functions as a counter to ensure the linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPTWn.GTPR register value + dead time value) as an initial value and the GPTWn.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest.
This counter becomes the value (GPTWn.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPTWn.GTPR register value + dead time value).
If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.
Table 24.12 and Table 24.13 show count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 24.12 Count Operation in Complementary PWM Mode (1) (n = 4, 7)

Counter	Initial Value	Initial Output Section (After Start)	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section
GPTWn.GTCNT	0	Up-Counting 0 to GTDVU	Up-Counting GTDVU + 1 to GTPR – GTDVU	Up-Counting GTPR – GTDVU + 1 to GTPR	Down-Counting GTPR – 1 to GTPR – GTDVU
GPTWn+1.GTCNT	GTDVU	Up-Counting GTDVU to GTDVU × 2	Up-Counting GTDVU × 2 + 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR
GPTWn+2.GTCNT	0	Up-Counting 0 to GTDVU	Stop GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GTDVU

Table 24.13 Count Operation in Complementary PWM Mode (2) (n = 4, 7)

Counter	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
GPTWn.GTCNT	Down-Counting GTPR – GTDVU – 1 to GTDVU	Down-Counting GTDVU – 1 to 0	Up-Counting 1 to GTDVU
GPTWn+1.GTCNT	Down-Counting GTPR – 1 to GTDVU × 2	Down-Counting GTDVU × 2 – 1 to GTDVU	Up-Counting GTDVU + 1 to GTDVU × 2
GPTWn+2.GTCNT	Stop 0	Up-Counting 1 to GTDVU	Down-Counting GTDVU – 1 to 0

In complementary PWM mode, the GTCCRA register buffer operation is different from normal buffer operation.
Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPTWn+2 channel.
Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).
Buffer operation of the GTPR register in complementary PWM mode is described in section 24.3.2.1, GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.

Table 24.14 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3. Table 24.15 shows buffer transfer timing during double buffer operation in complementary PWM mode 3.

Table 24.14 Single Buffer Transfer Timing in Complementary PWM Mode 1, 2, 3 (n = 4, 7)

Buffer Transfer	Complementary PWM Mode 1	Complementary PWM Mode 2	Complementary PWM Mode 3
GTCCRD ↓ Temporary register A	After one GTCLK cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)	After one GTCLK cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)	After one GTCLK cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)
Temporary register A ↓ GTCCRC	(1) When transferring to the temporary register A at the middle section of the up-count • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the up-count • The end of the trough (3) Counter stopped • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A	(1) When transferring to the temporary register A at the middle section of the down-count • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the down-count • The end of the crest (3) Counter stopped • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A	(1) When transferring to the temporary register A at the middle section • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A (2) When transferring to the temporary register A at other than the middle section • The end of the crest/trough (3) Counter stopped • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A
GTCCRC ↓ GTCCRA	• The end of the trough • Counter clear in the up-count intermediate section and crest section	• The end of the trough section excluding the initial output section • Counter clear in the down-count intermediate section and trough section	• The end of the trough • The end of the trough section excluding the initial output section • Counter clear

Table 24.15 Double Buffer Transfer Timing in Complementary PWM Mode 3 (n = 4, 7)

Buffer Transfer	Transfer Timing
GTCCRD ↓ Temporary register A	After 1 GTCLK after slave channel 2 (GPTWn+2) writes to the GTCCRD register
Temporary register A ↓ GTCCRC	(1) When transferring data to the temporary register A at the middle section • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A (2) When transferring data to the temporary register A at other than the middle section • At the end of crest and trough sections (3) Counter stopped • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register A
GTCCRC ↓ GTCCRA	• The end of the crest • Counter clear

Buffer Transfer	Transfer Timing
GTCCRF ↓ Temporary register B	After 1 GTCLK after slave channel 2 (GPTWn+2) writes to the GTCCRD register
Temporary register B ↓ GTCCRE	(1) When transferring data to the temporary register B at the middle section • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register B (2) When transferring data to the temporary register B at other than the middle section • At the end of crest and trough sections (3) Counter stopped • One cycle of PCLKCA/GTCLK after transfer of data to the temporary register B
GTCCRE ↓ GTCCRA	• The end of the trough section excluding the initial output section

The change in the output level of the positive phase waveform output from the GTIOCnA pin (n = 4 to 9) and the negative phase waveform output from the GTIOCnB pin (n = 4 to 9) is determined by the compare match by the combination of the counter and register in each operation section. In the middle section, the compare match between the GPTWn.GTCNT counter and the GTCCRA register changes the positive phase waveform output level, and the compare match between the GPTWn+1.GTCNT counter and the GTCCRA register changes the negative phase waveform output level. The crest and trough sections use the GPTWn+2.GTCNT counter, GTCCRC register, and GTCCRE register to

perform a compare match operation to ensure linearity near 0% and 100% duty.
When the compare match value is greater than or equal to the GPTWn.GTPR register value, the duty is 0% (positive phase waveform OFF, negative phase waveform ON). When the compare match value is 0, the duty is 100% (positive phase waveform ON, negative phase waveform OFF).
Table 24.16 shows the counter and register combinations used in the comparative match operation to generate a positive-phase waveform and a negative-phase waveform in each operating section.

Table 24.16 Combinations of Counters and Registers for Compare Match Operation in Complementary PWM Mode

	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
Negative-Phase OFF	GPTWn+1.GTCNT	GPTWn+1.GTCNT	—	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT	GPTWn+1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-Phase ON	GPTWn.GTCNT	GPTWn.GTCNT	GPTWn+2.GTCNT	—	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Negative-Phase OFF	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT	GPTWn.GTCNT	GPTWn.GTCNT	GPTWn.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Positive-Phase ON	—	GPTWn+1.GTCNT*1	GPTWn+1.GTCNT	GPTWn+1.GTCNT	GPTWn+1.GTCNT	GPTWn+2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

In the case of normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.
In the initial output section, the initial output set in the GTIOR register is retained. In the case that the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.
As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 24.67 and Figure 24.68 show complementary PWM mode 1, Figure 24.69 and Figure 24.70 show complementary PWM mode 2, Figure 24.71 and Figure 24.72 show single buffer complementary PWM mode 3, and Figure 24.73 and Figure 24.74 show double buffer complementary PWM mode 3.
Figure 24.75 to Figure 24.86 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.
Figure 24.87 and Figure 24.88 show examples of initial output operation according to the GTCCRA register value.
Figure 24.89 shows an example for setting complementary PWM modes 1 to 3.

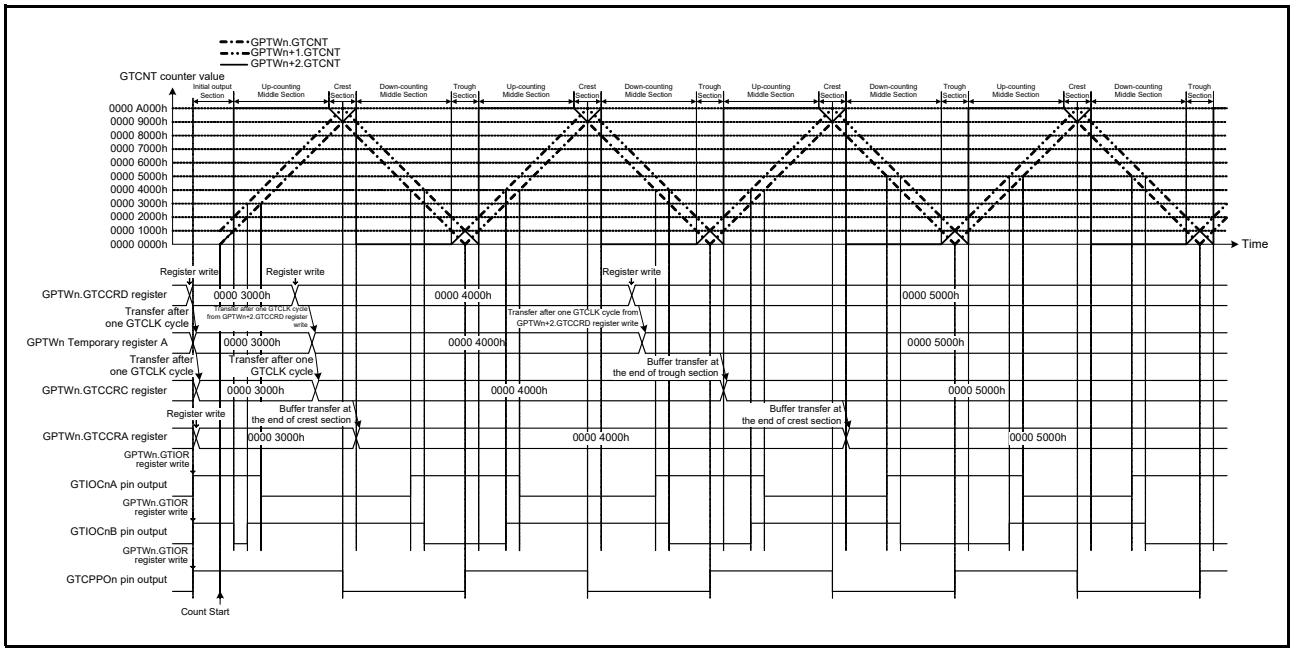


Figure 24.67 Example of Complementary PWM Mode 1 Operation (1)
(GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 4, 7)

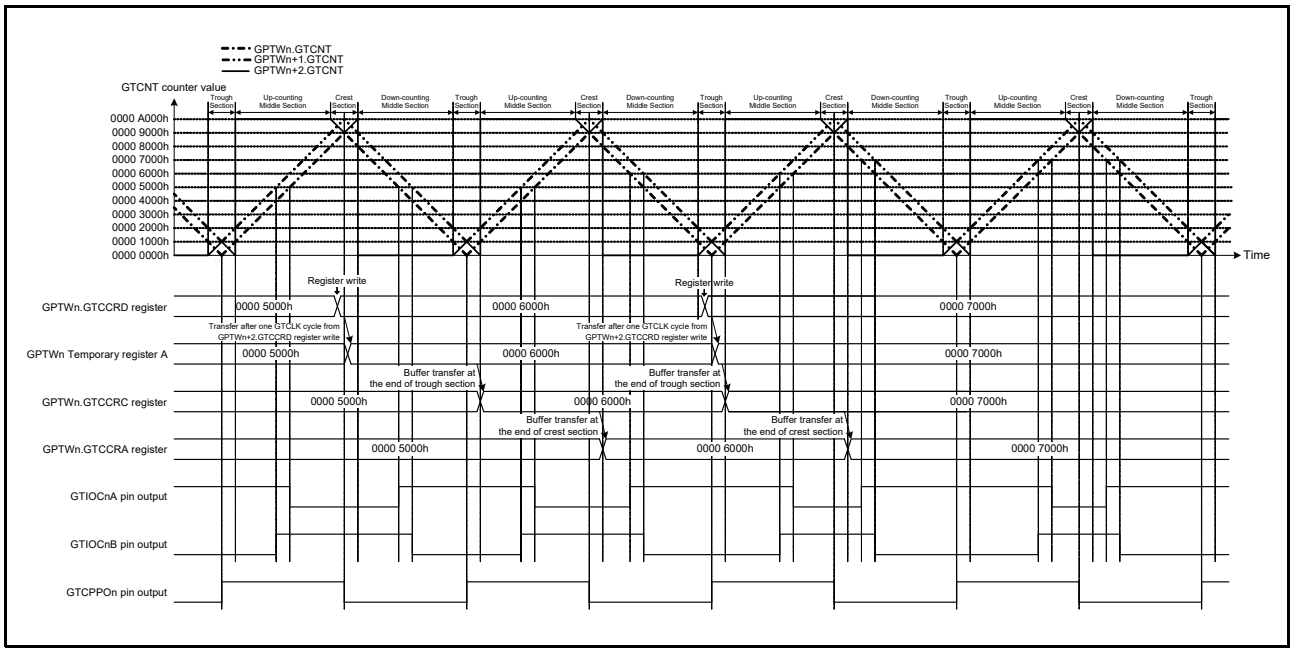


Figure 24.68 Example of Complementary PWM Mode 1 Operation (2)
(GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 4, 7)

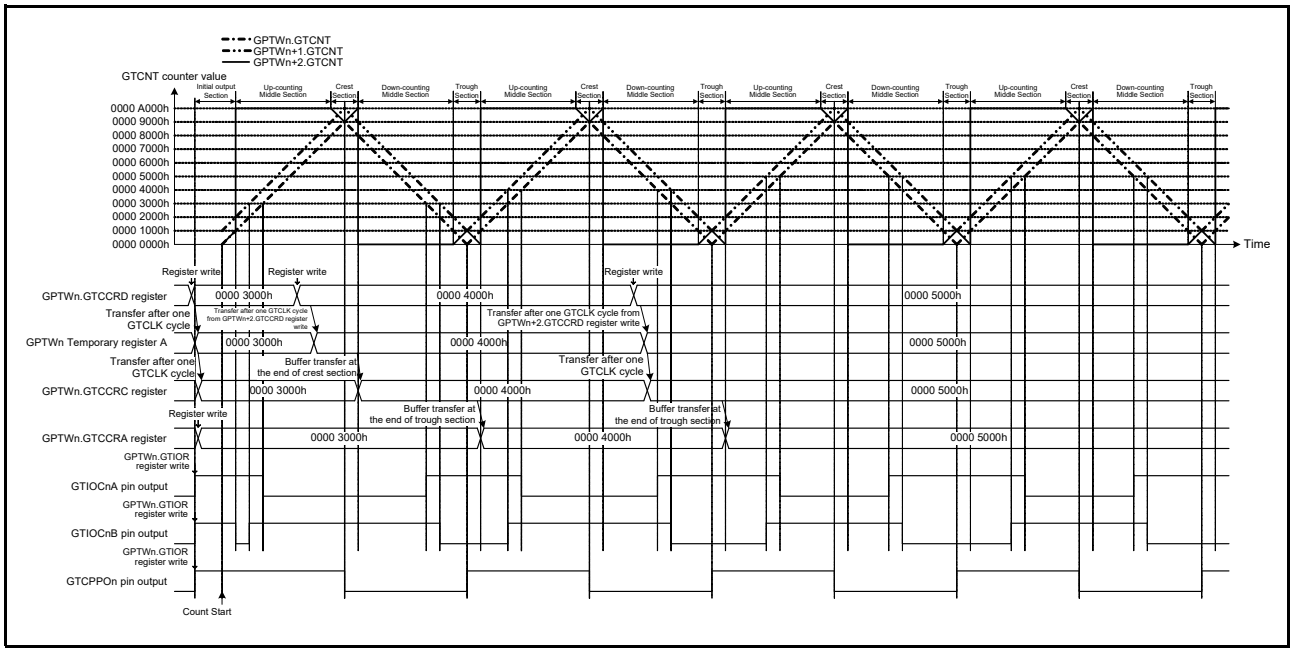


Figure 24.69 Example of Complementary PWM Mode 2 Operation (1)
(GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 4, 7)

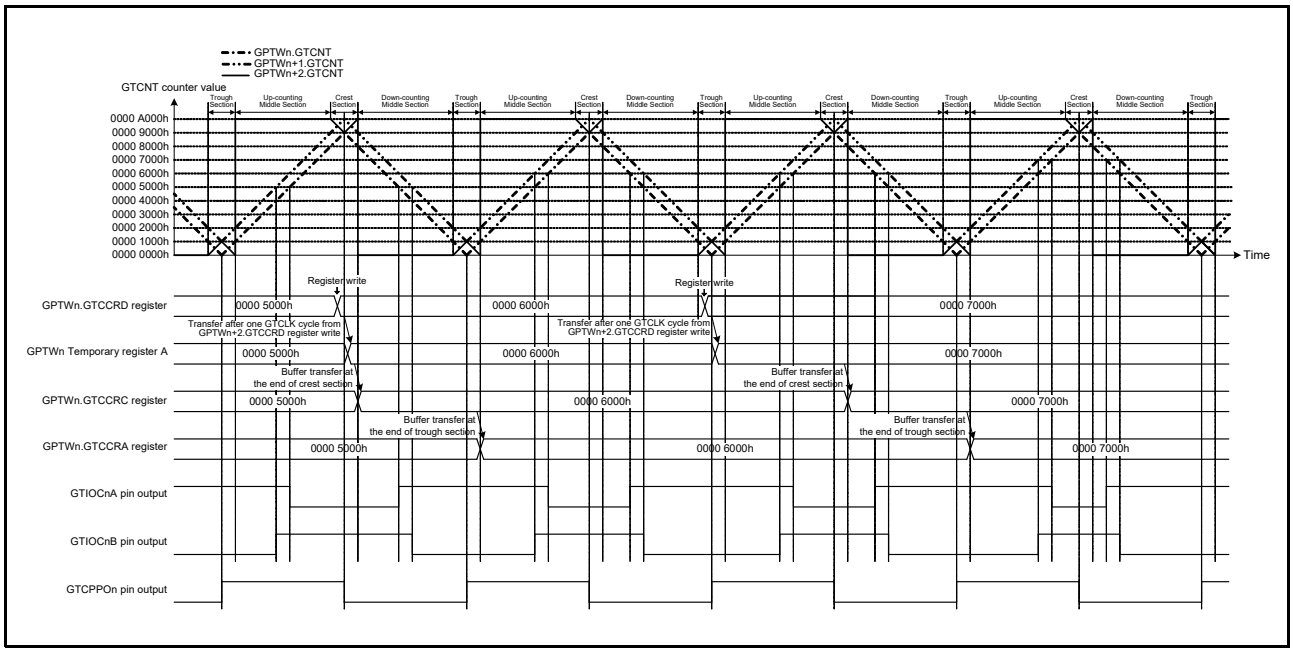


Figure 24.70 Example of Complementary PWM Mode 2 Operation (2)
(GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 4, 7)

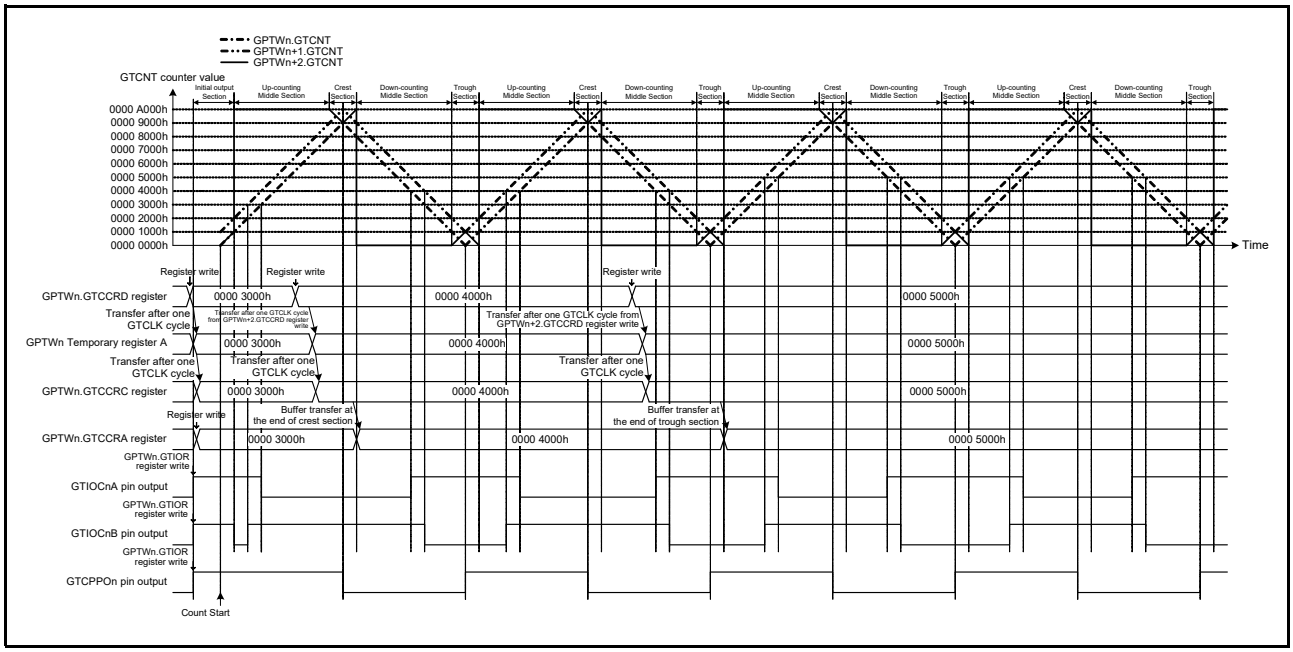


Figure 24.71 Example of Complementary PWM Mode 3 Operation (1)
(Single buffer operation, GTIOCnA pin = High/GTIOCnB pin = High as initial output, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 4, 7)

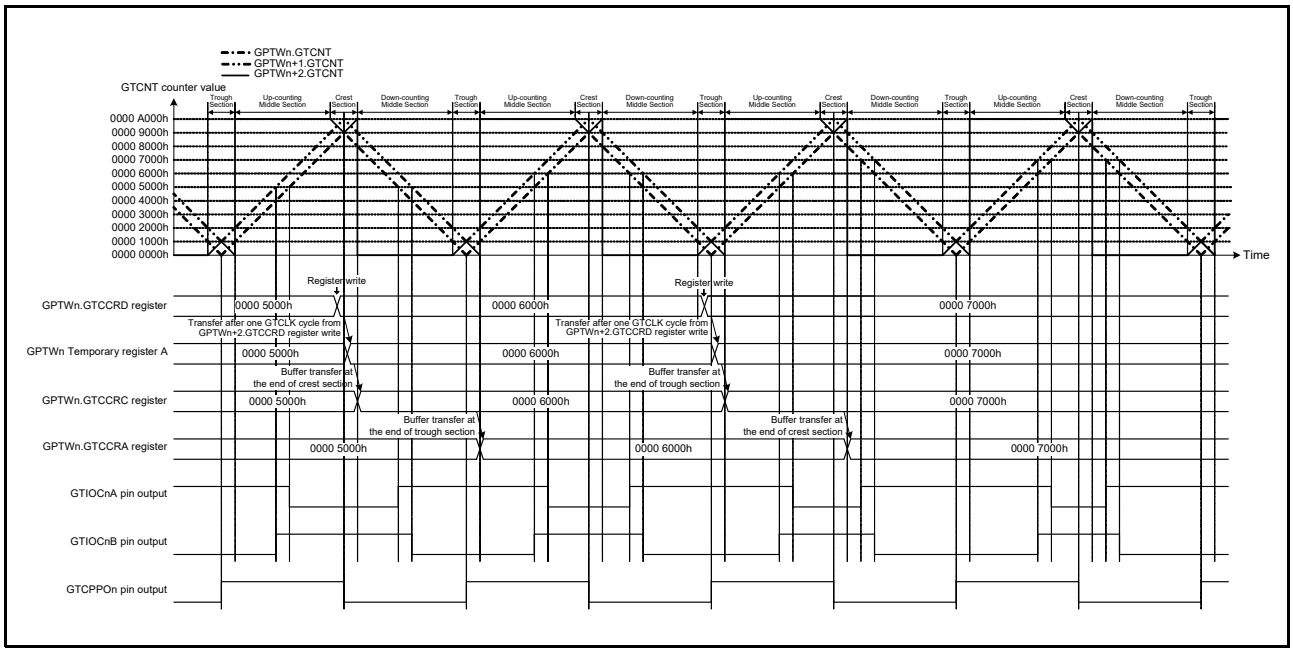


Figure 24.72 Example of Complementary PWM Mode 3 Operation (2)
(Single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 4, 7)

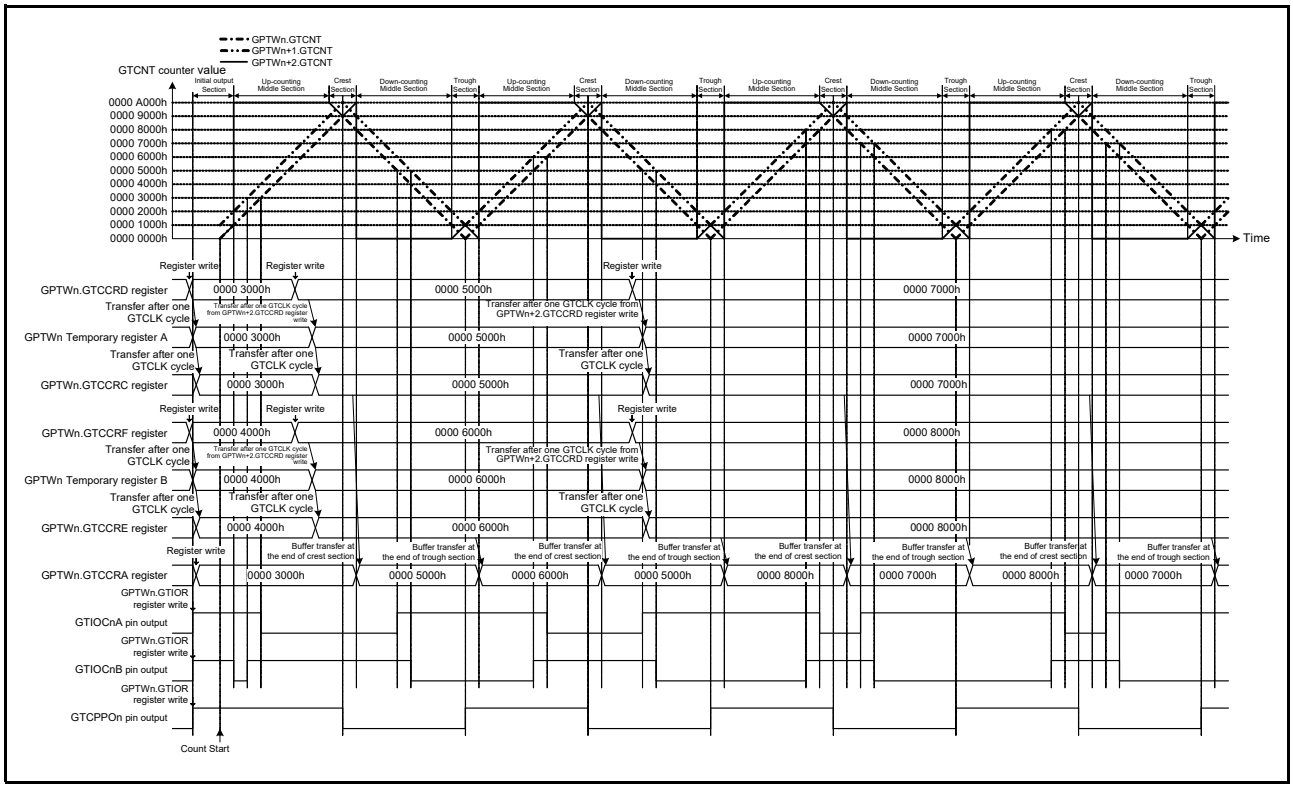


Figure 24.73 Example of Complementary PWM Mode 3 Operation (3)
(Double buffer operation,
GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 4, 7)

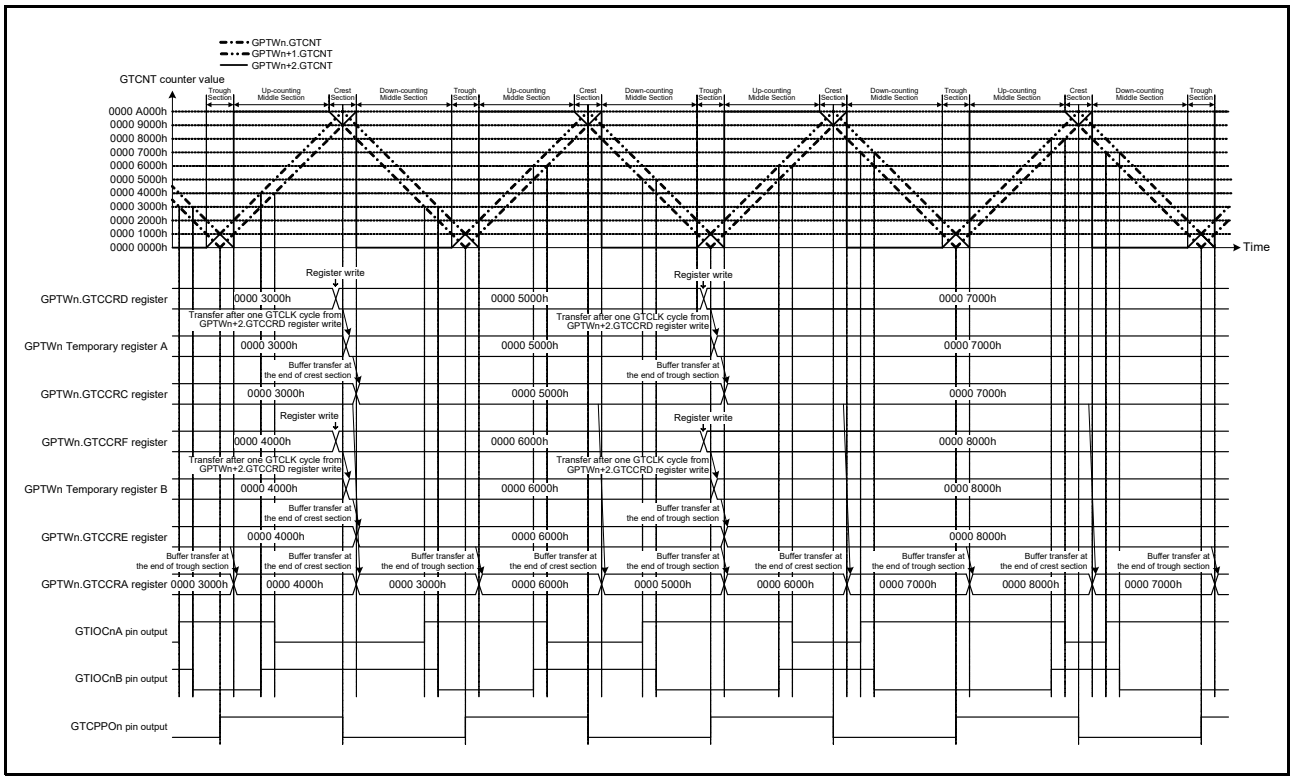


Figure 24.74 Example of Complementary PWM Mode 3 Operation (4)
(Double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections)
(n = 4, 7)

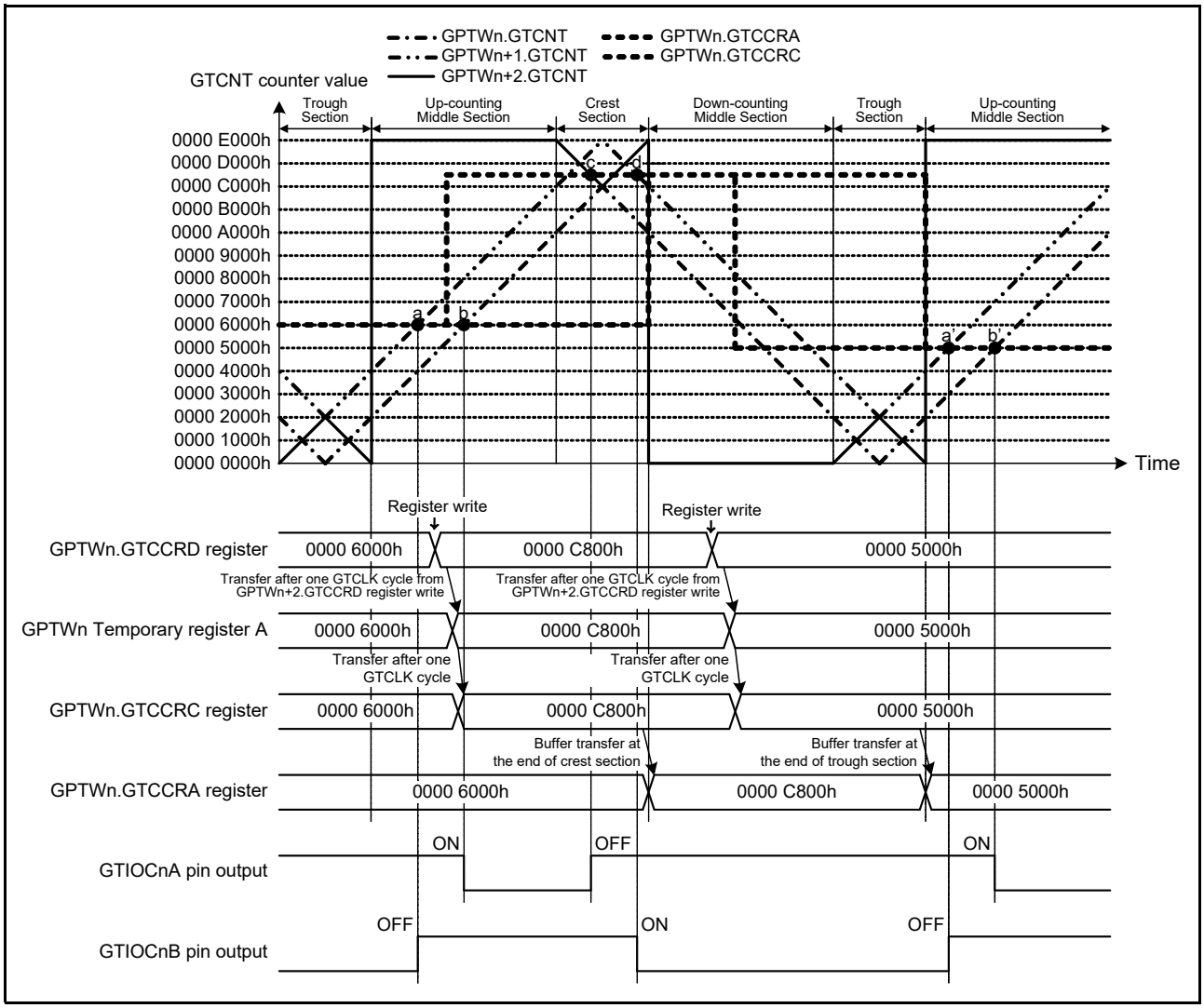


Figure 24.75 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (1) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → b → c → d) (n = 4, 7)

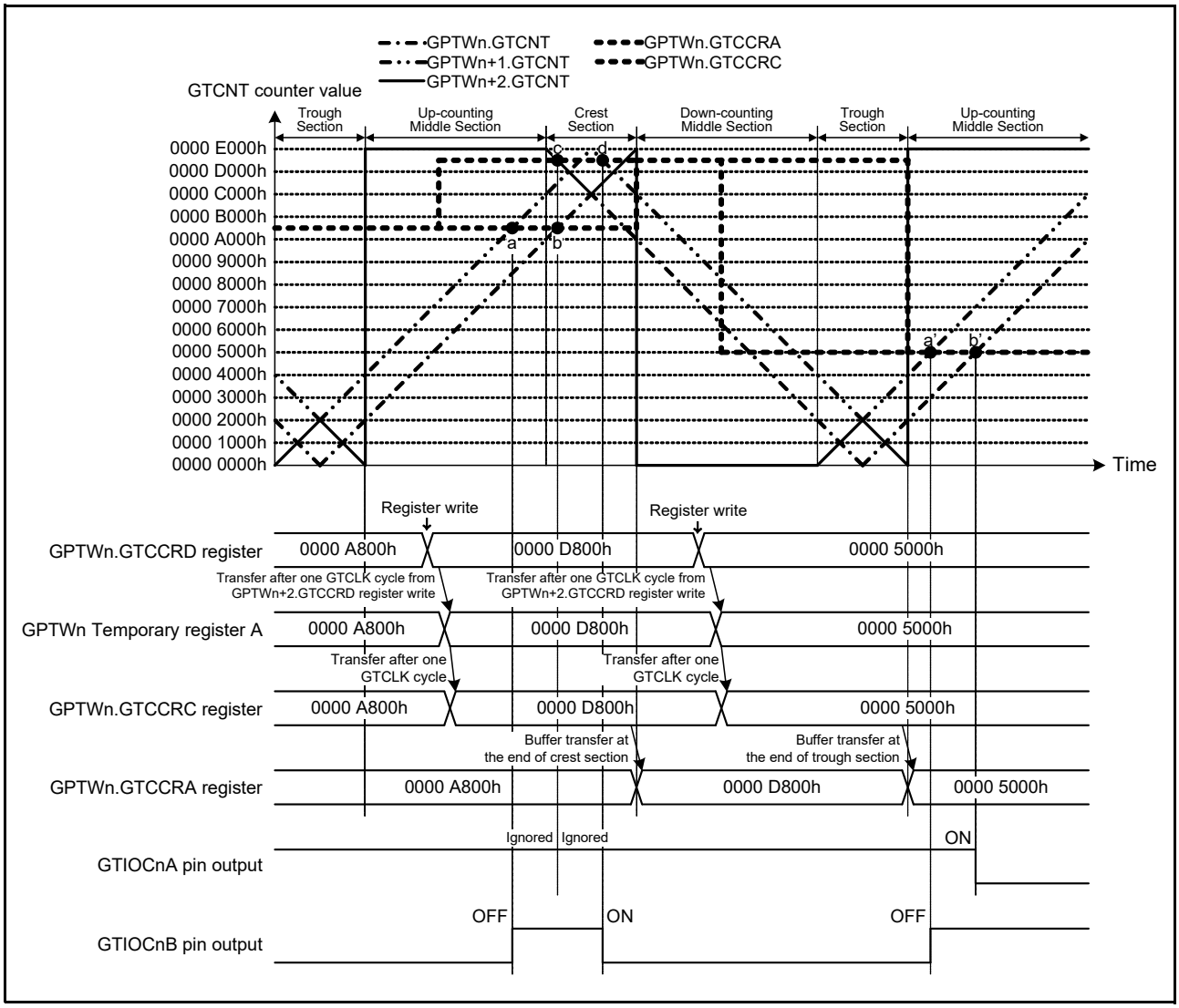


Figure 24.76 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (2) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → (b, c) → d) (n = 4, 7)

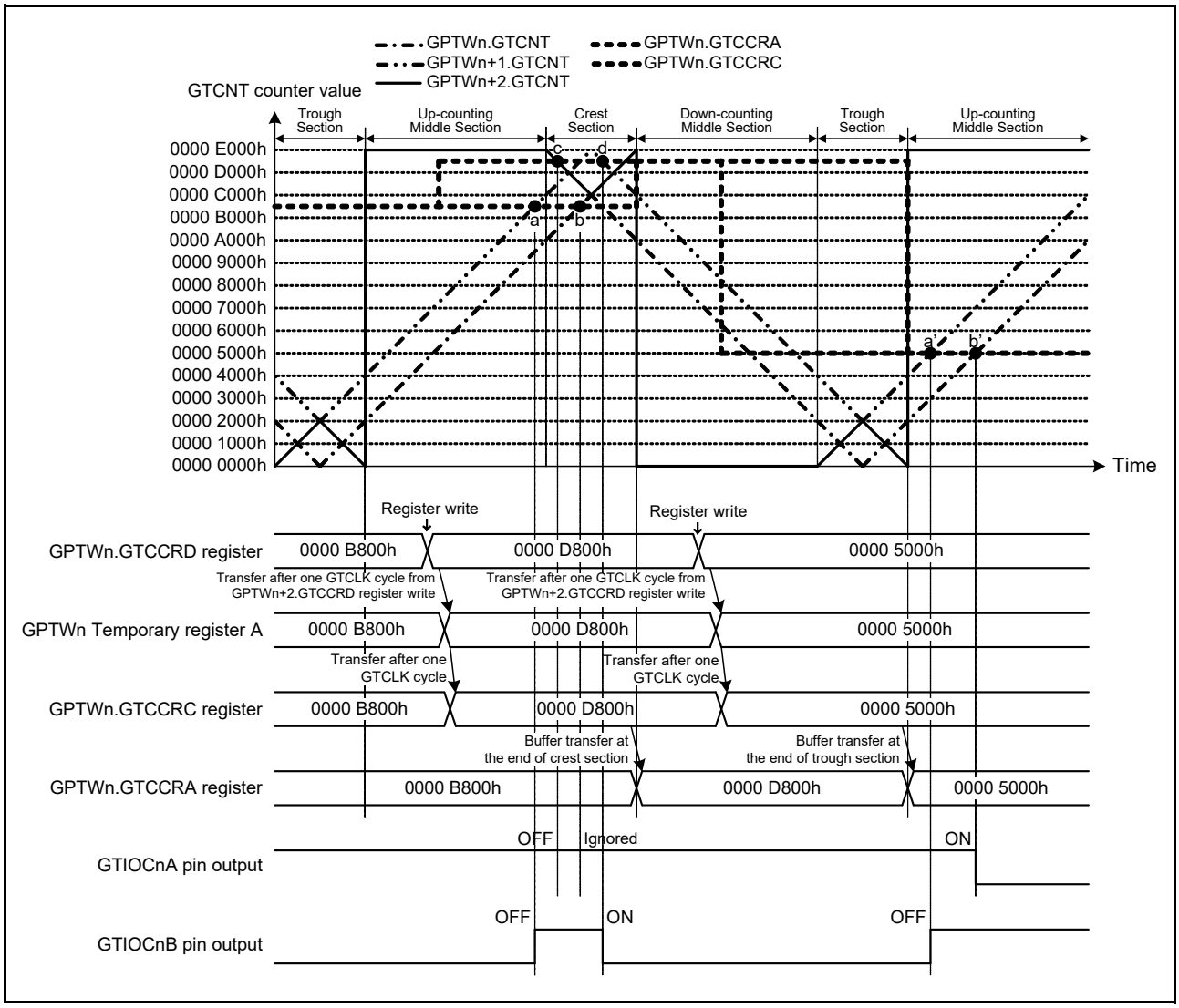


Figure 24.77 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (3) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → c → b → d) (n = 4, 7)

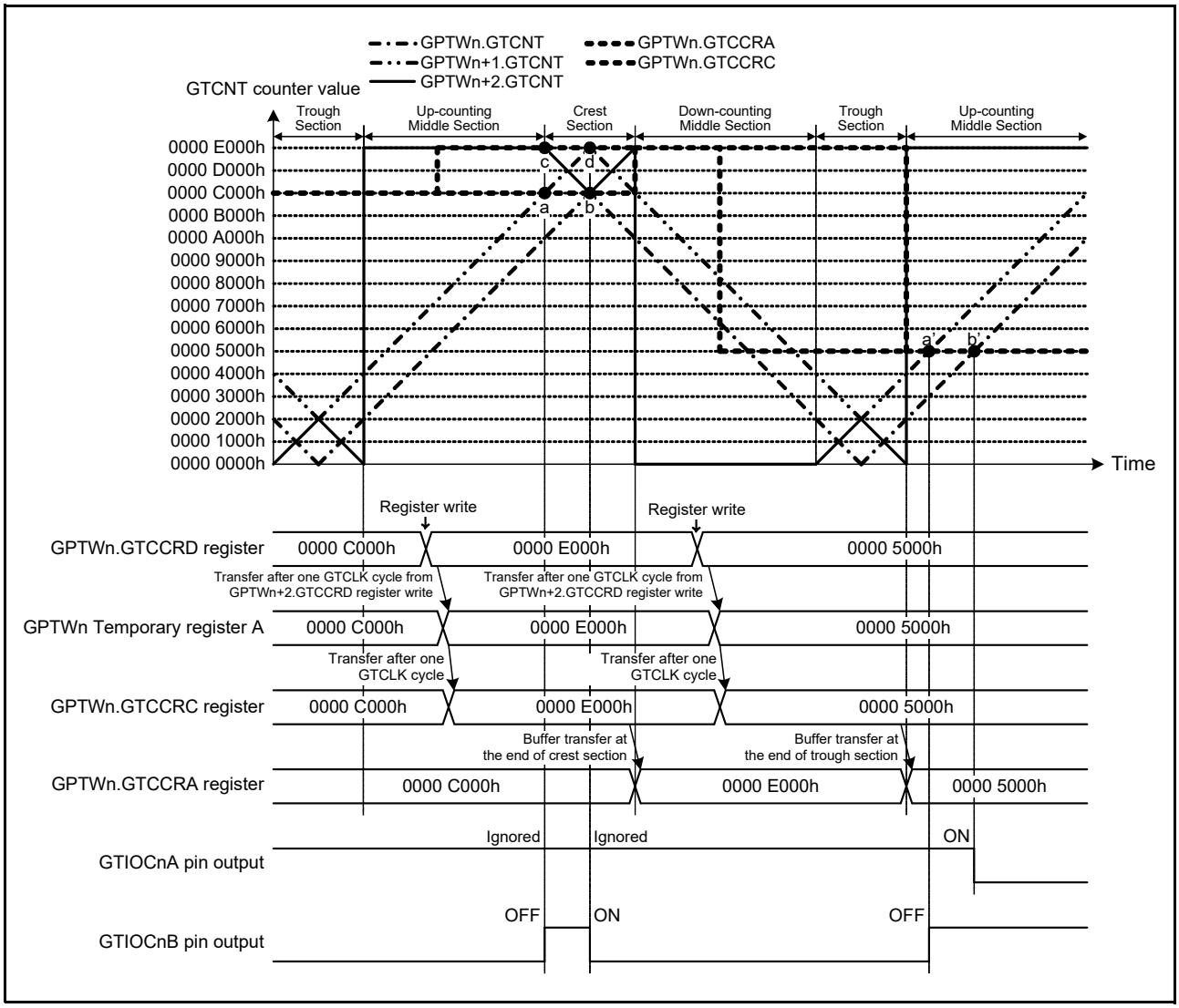


Figure 24.78 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (4) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (a, c) → (b, d)) (n = 4, 7)

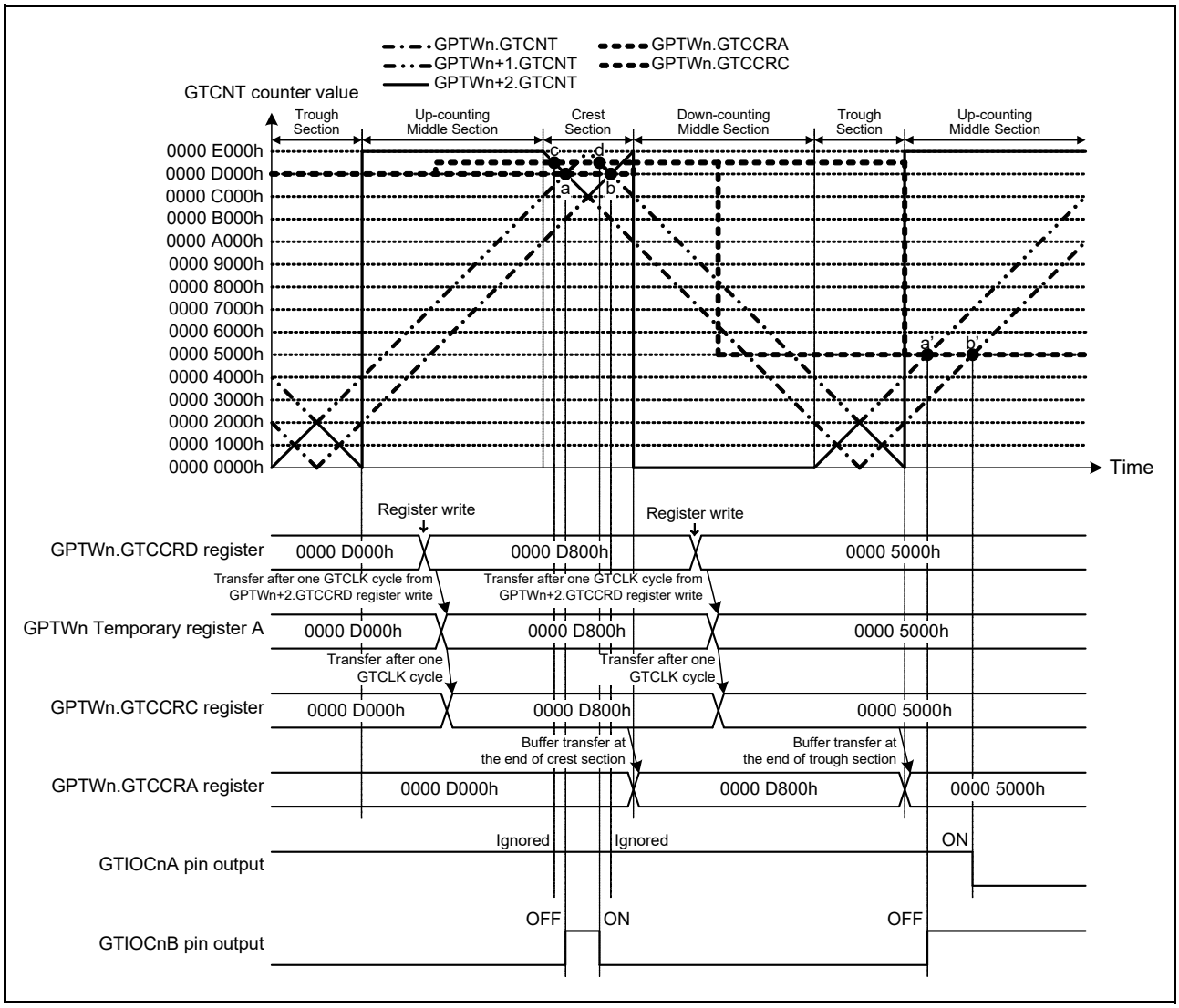


Figure 24.79 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (5) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a → d → b) (n = 4, 7)

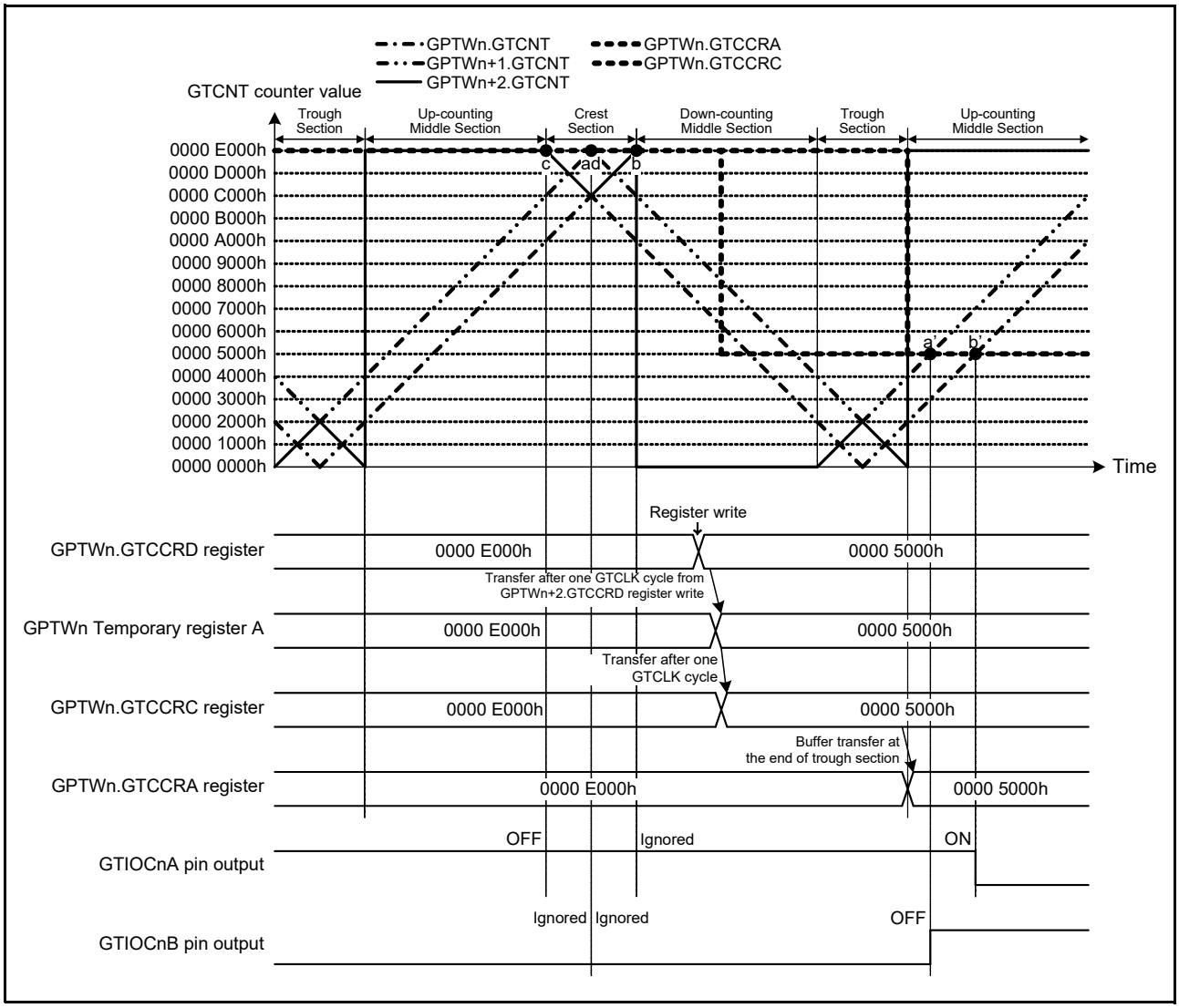


Figure 24.80 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (6) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (a, d) → b) (n = 4, 7)

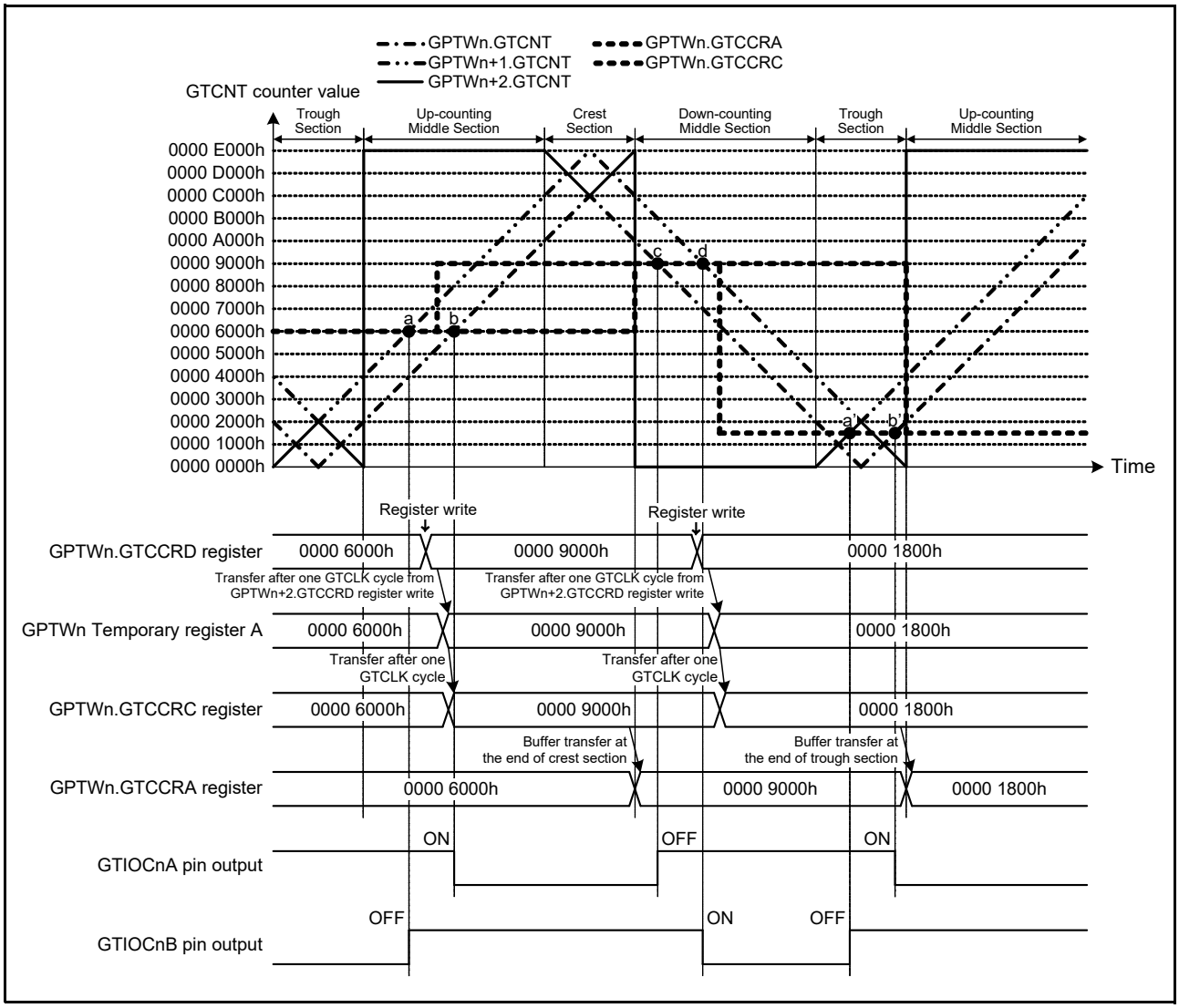


Figure 24.81 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (7) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → d → a' → b') (n = 4, 7)

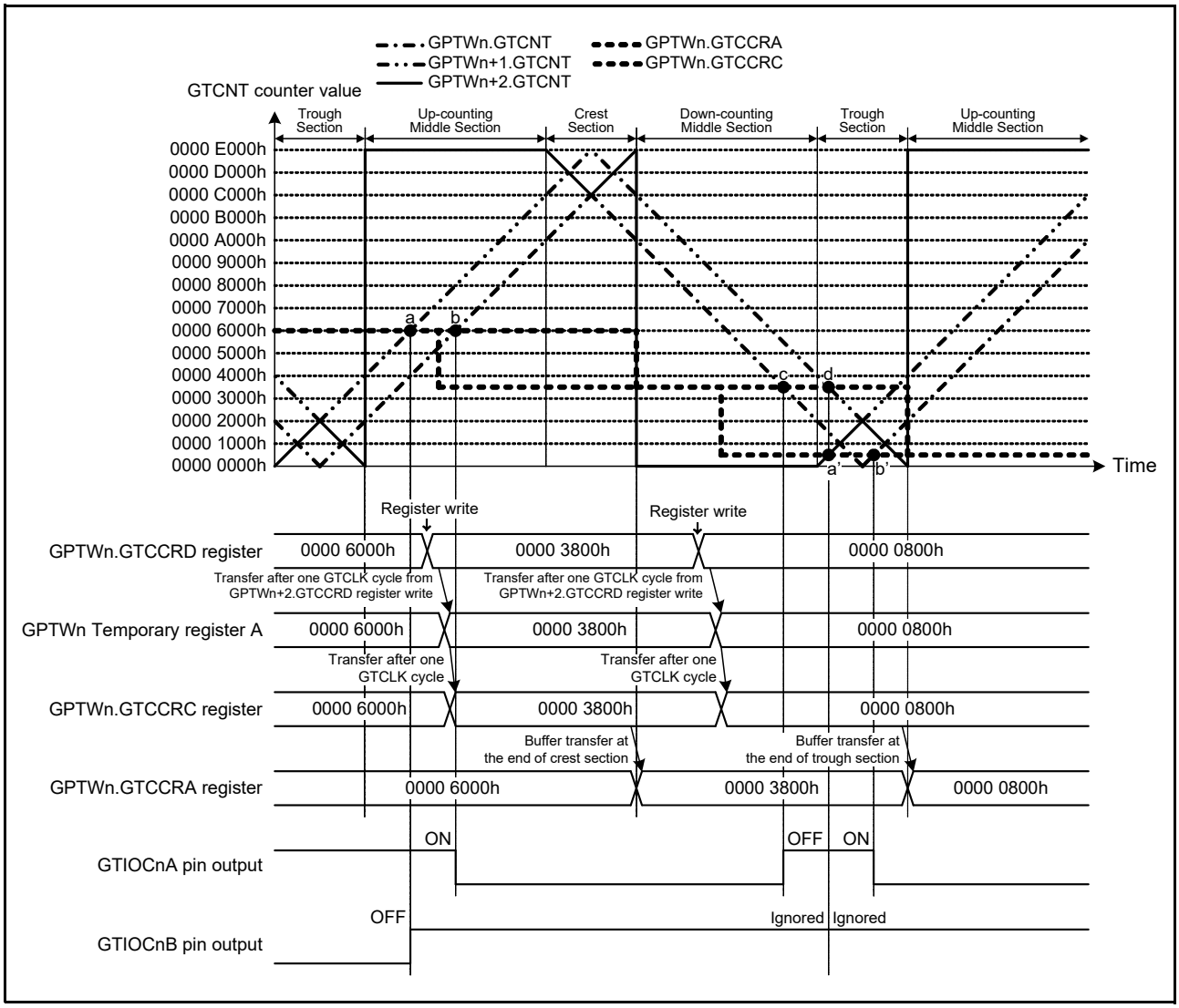


Figure 24.82 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (8) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (d, a') → b') (n = 4, 7)

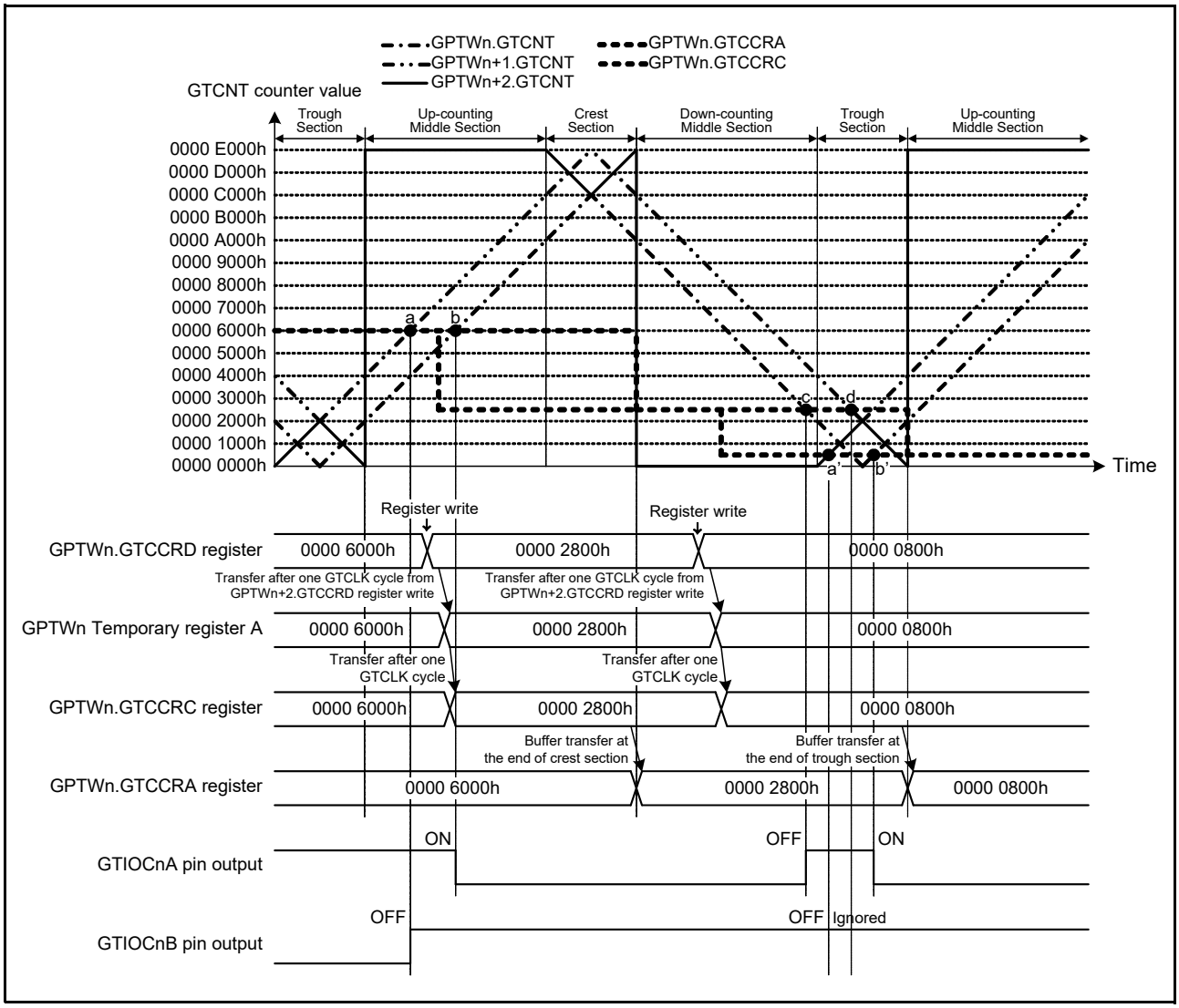


Figure 24.83 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (9) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a' → d → b') (n = 4, 7)

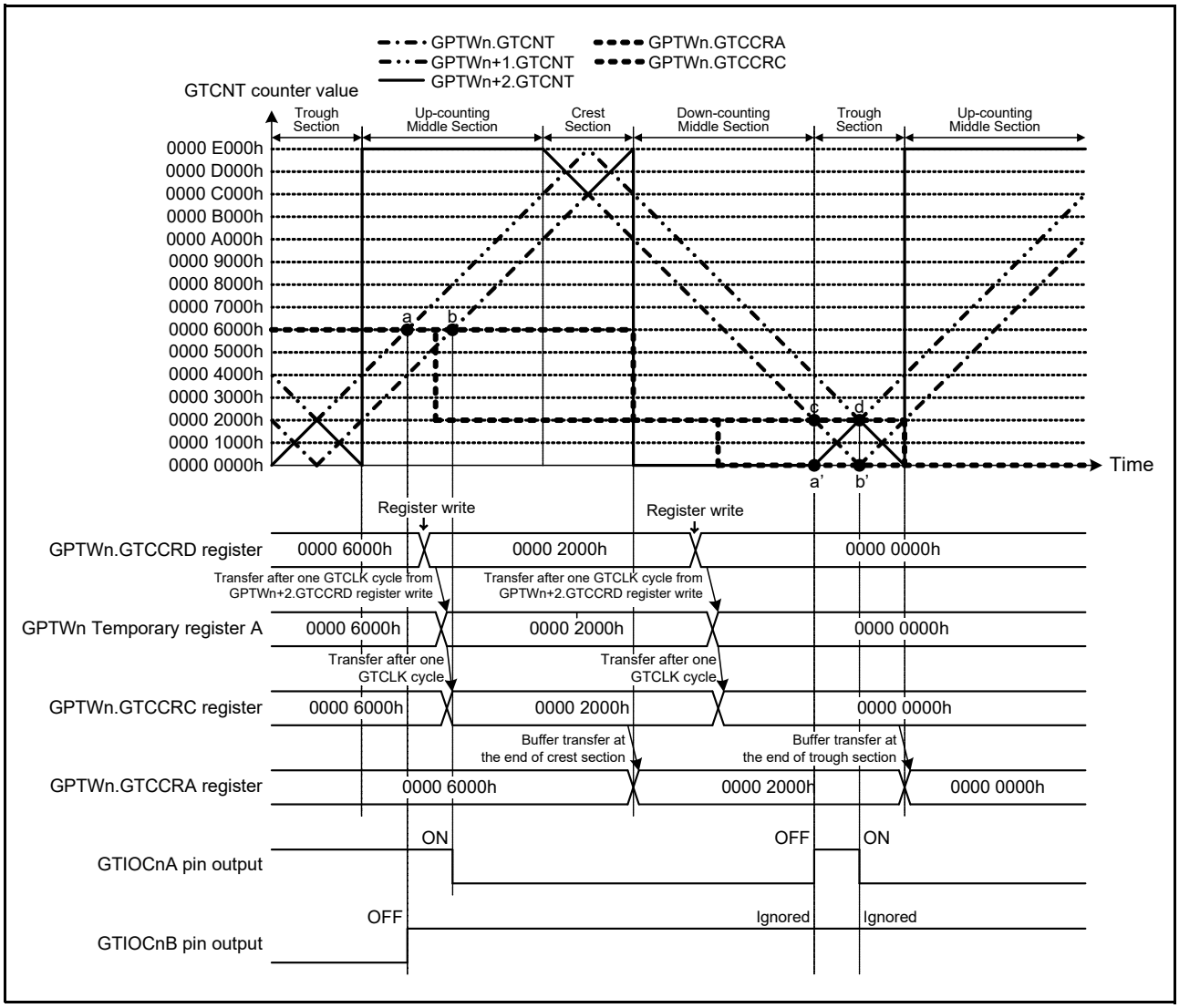


Figure 24.84 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (10) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (c, a') → (d, b')) (n = 4, 7)

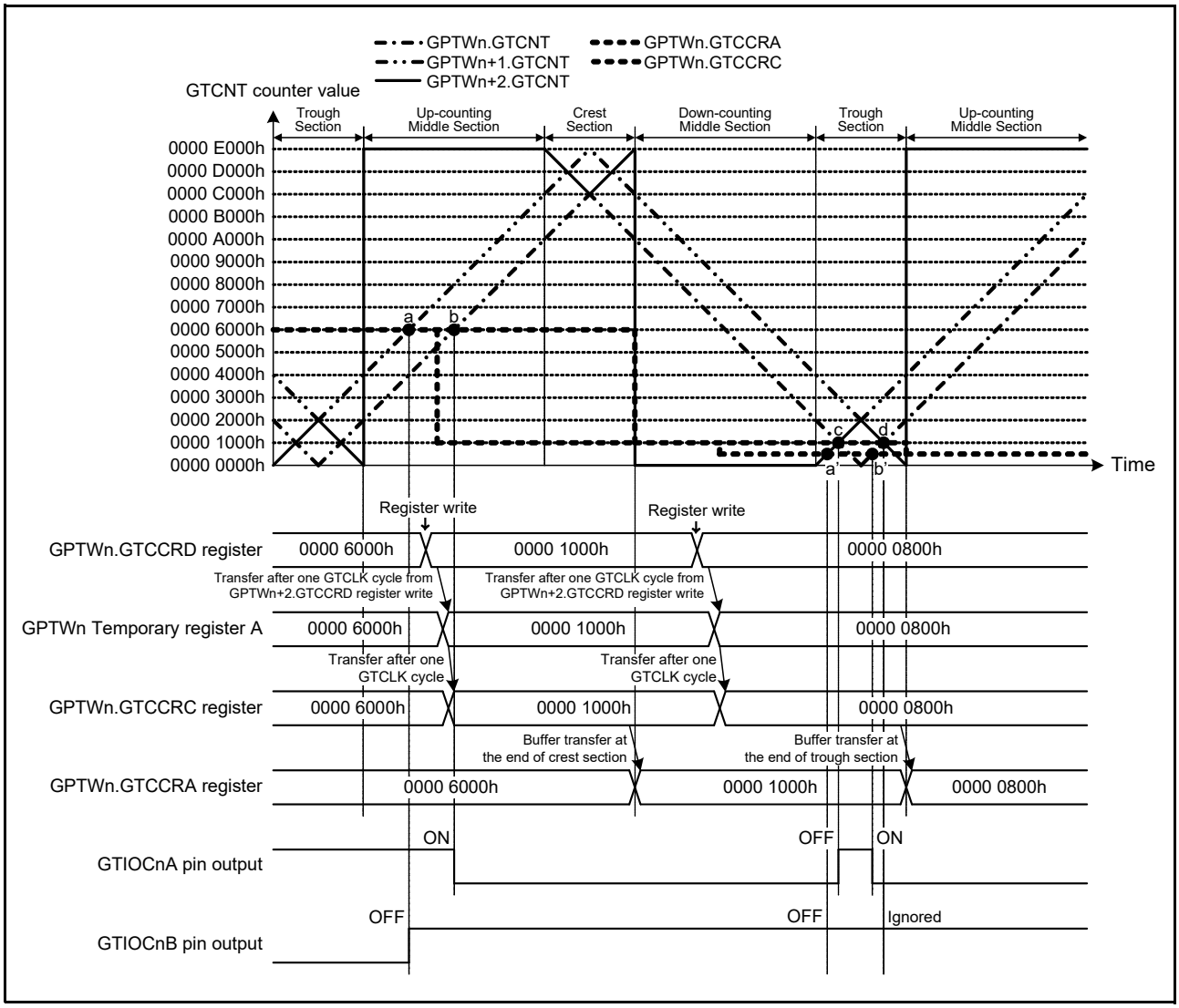


Figure 24.85 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (11) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a' → c → b' → d) (n = 4, 7)

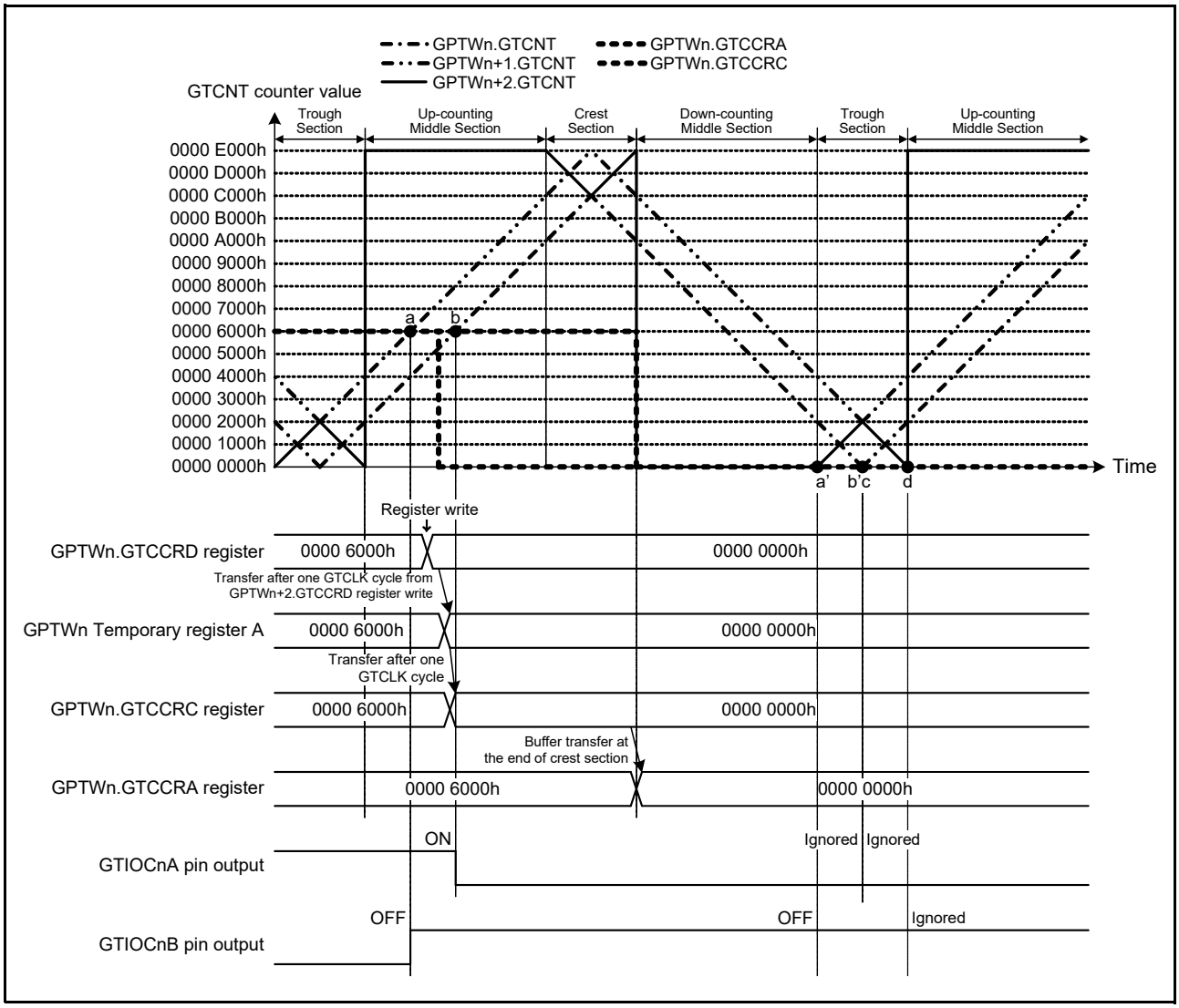


Figure 24.86 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (12) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a' → (b', c) → d) (n = 4, 7)

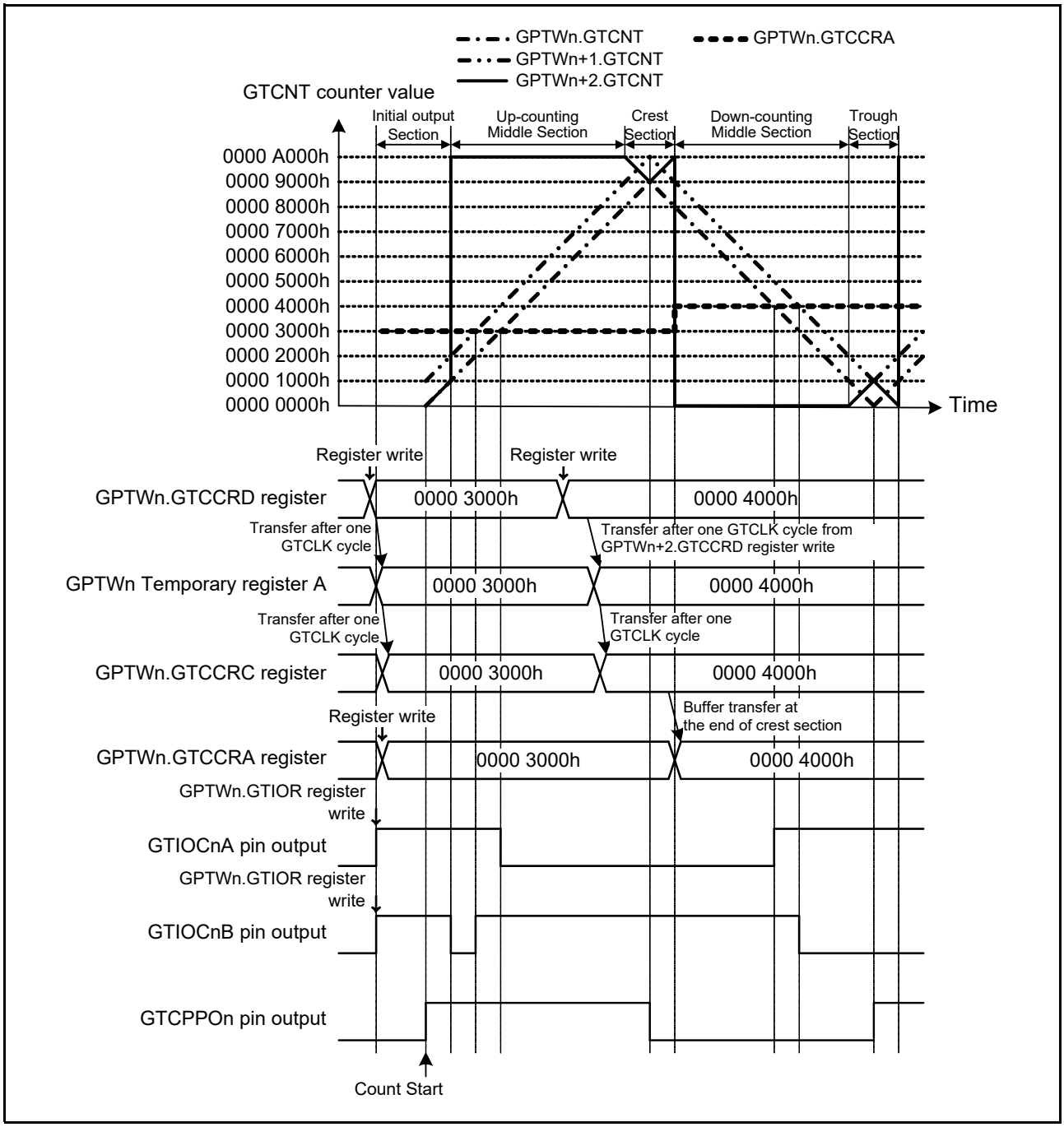


Figure 24.87 Example of Complementary PWM Mode Initial Output Operation (1)
(Complementary PWM mode 1 operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is larger than the dead time value) (n = 4, 7)

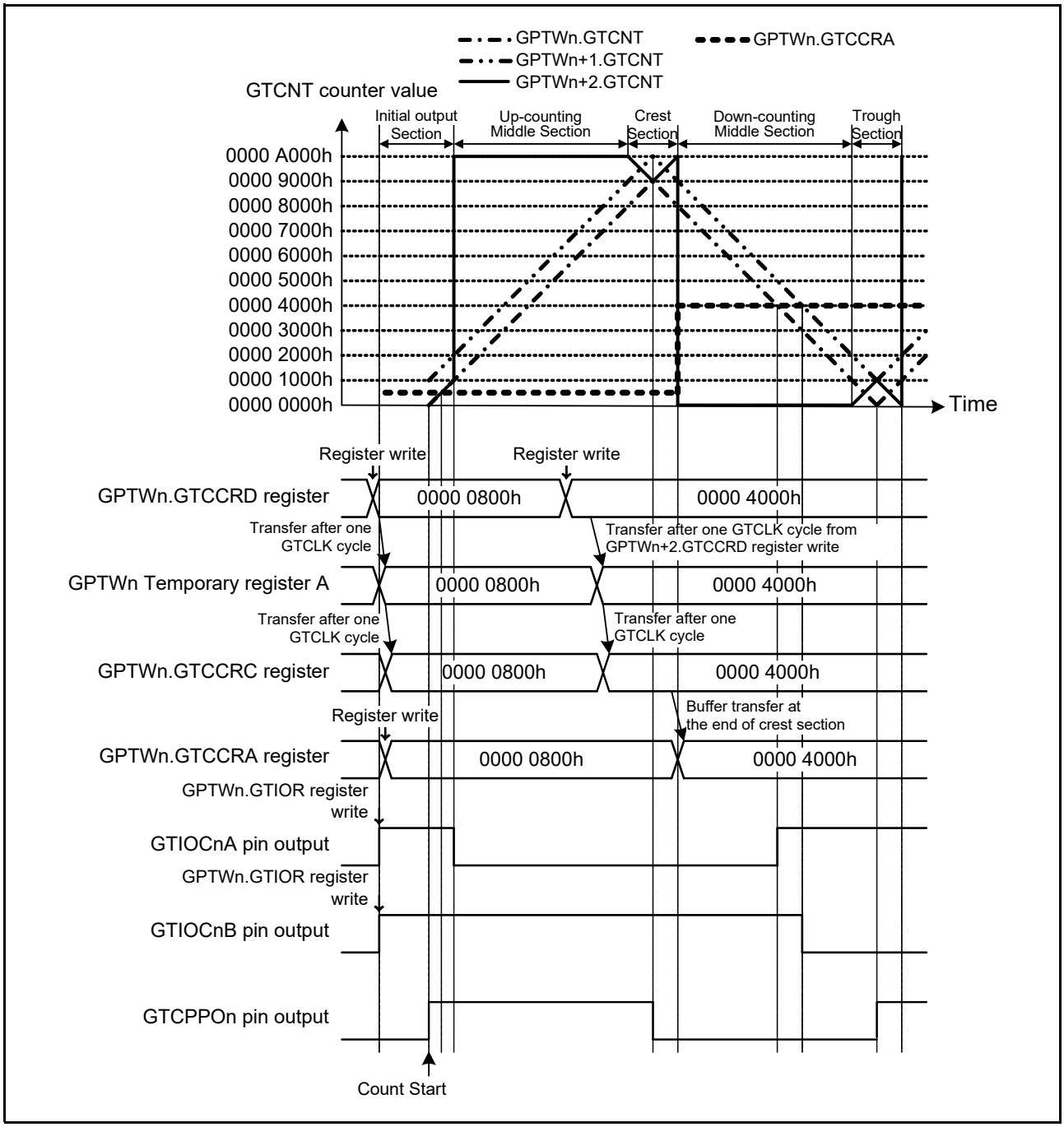


Figure 24.88 Example of Complementary PWM Mode Initial Output Operation (2)
(Complementary PWM mode 1 operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is equal to or less than the dead time value) (n = 4, 7)

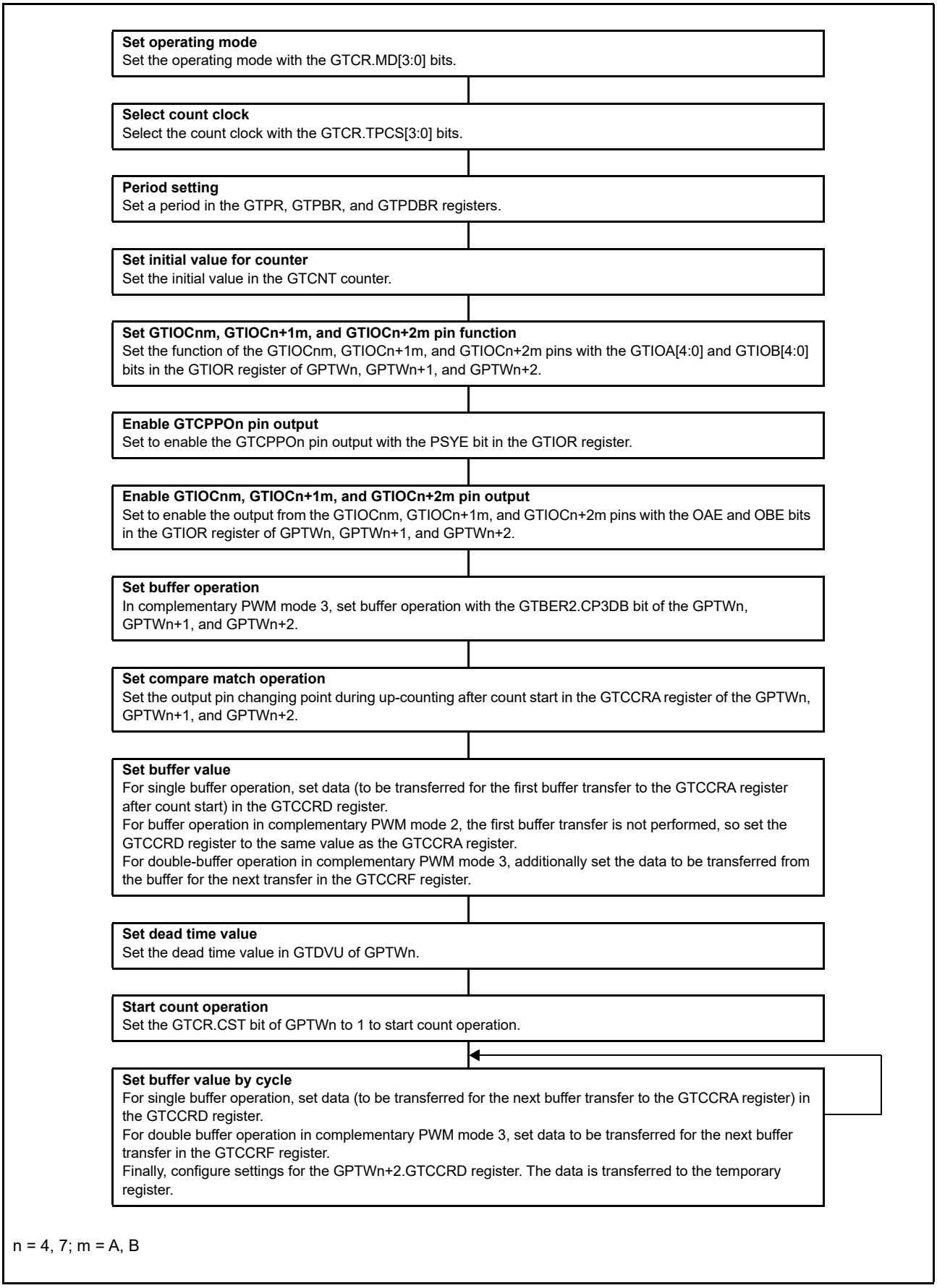


Figure 24.89 Example for Setting Complementary PWM Mode 1, 2, 3

24.3.3.8 Complementary PWM Mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 24.90 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Figure 24.66.

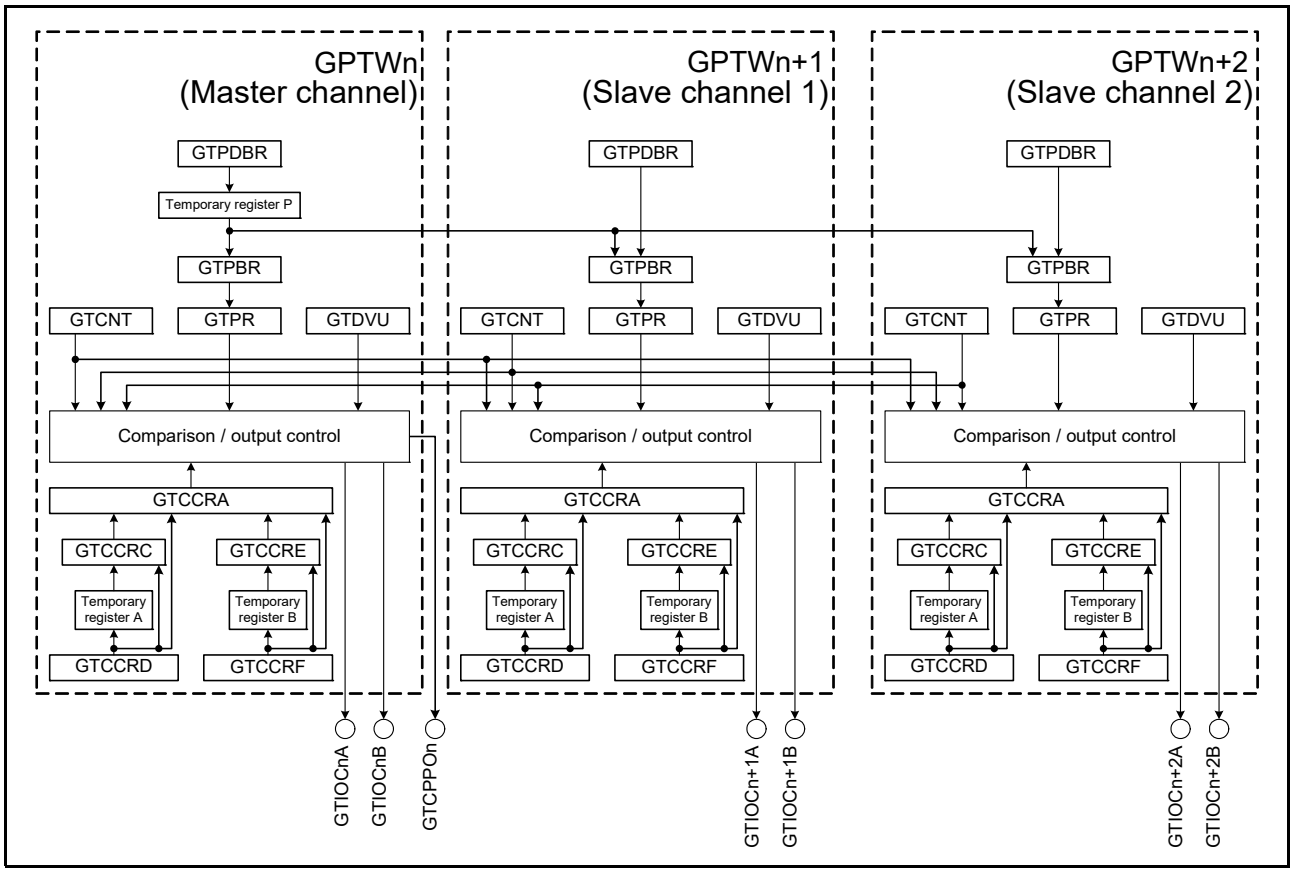


Figure 24.90 Block Diagram in Complementary PWM Mode 4 (n = 4, 7)

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 24.12 and Table 24.13.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 24.90) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation.

Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

The double buffer function by writing to the GTCCRF register can be enabled or disabled with the GTBER2.CP3DB bit.

In double buffer operation, the value written to the GTCCRD register is used as a compare match value for positive-phase OFF (negative-phase ON) during down-counting, and the value written to the GTCCRF register is used as a compare match value for negative-phase OFF (positive-phase ON) during up-counting. Transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Double buffer operation is guaranteed only in the up-counting middle section and down-counting middle section. Setting a value that is less than or equal to the dead time value, or greater than or equal to the count cycle, is prohibited.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.
Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPTWn+2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.
Table 24.17 and Table 24.18 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPTWn+2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from the GTCCRC register to the GTCCRA register) other than those shown in Table 24.17 and Table 24.18 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 24.14.
Table 24.19 and Table 24.20 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPTWn+2.GTCCRD register) to each register by writing a value to the GTCCRD and GTCCRF registers during double buffer operation in complementary PWM mode 4 for each compare match state in each operation section.
Transfers (from the GTCCRD register to the temporary register A, from the GTCCRF register to the temporary register B, from the temporary register A to the GTCCRC register, from the temporary register B to the GTCCRE register, and from the GTCCRC and GTCCRE registers to the GTCCRA register) other than those shown in Table 24.19 and Table 24.20 are the same as double buffer transfer in complementary PWM mode 3 shown in Table 24.15.

Table 24.17 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (1) (n = 4, 7)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	• In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match • In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match • In the case of GTCCRD < GPTWn+2.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn+2.GTCNT GPTWn+2.GTCNT Negative-phase OFF After down-counting dead time start No transfer	No transfer
Down-counting crest section	Before down-counting compare match	Up-counting dead time period • In the case of GTCCRD < GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF After up-counting compare match • In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 24.18 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (2) (n = 4, 7)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	• In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match • In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match • In the case of GTCCRD > GPTWn+2.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+2.GTCNT GPTWn+2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up-counting trough section	Before up-counting compare match	Down-counting dead time period • In the case of GTCCRD > GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn.GTCNT GPTWn.GTCNT Positive-phase ON After down-counting compare match • In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Table 24.19 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4 (1) (n = 4, 7)

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	GTCCRF	• In the case of GTCCRF > GPTWn+1.GTCNT GTCCRF • In the case of GTCCRF ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	GTCCRF	No transfer
	After up-counting compare match	GTCCRD	GTCCRF	No transfer

Table 24.20 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4 (2) (n = 4, 7)

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Down-counting middle section	Down-counting compare match	GTCCRD	GTCCRF	• In the case of GTCCRD < GPTWn.GTCNT GTCCRD • in the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	GTCCRF	No transfer
	After down-counting compare period	GTCCRD	GTCCRF	No transfer

Figure 24.91 to Figure 24.95 show examples of single buffer operation for each operation section in complementary PWM mode 4.

- Up-Counting Middle Section: Figure 24.91
- Up-Counting Crest Section: Figure 24.92
- Down-Counting Crest Section: Figure 24.93
- Down-Counting Middle Section: Figure 24.94
- Down-Counting Trough Section: Figure 24.95

Figure 24.96 to Figure 24.99 show examples of double buffer operation for each operation section in complementary PWM mode 4.

- Up-Counting Middle Section: Figure 24.96 and Figure 24.97
- Down-Counting Middle Section: Figure 24.98 and Figure 24.99

Figure 24.100 shows an example for setting complementary PWM mode 4.

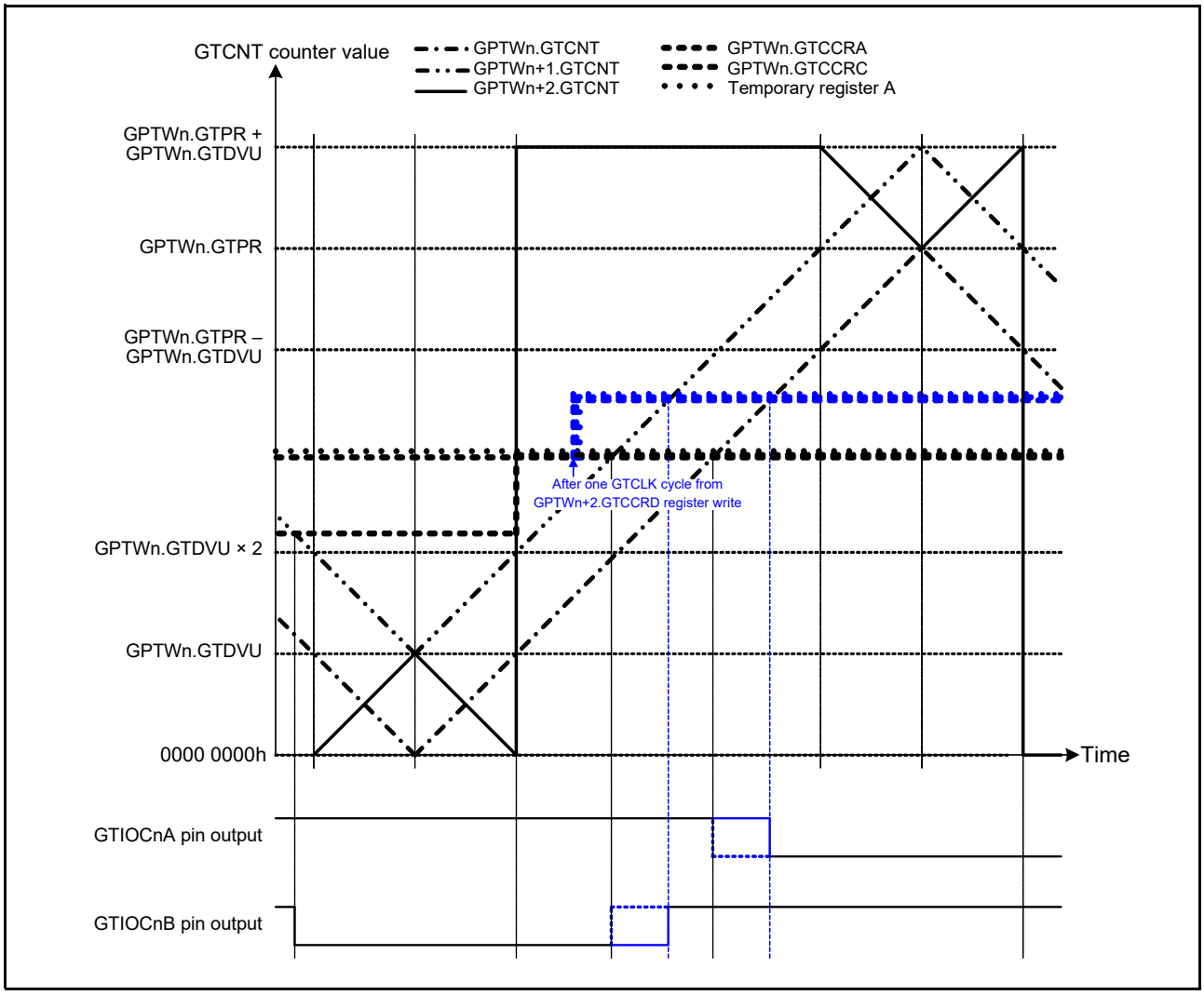


Figure 24.91 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPTWn+1.GTCNT value is written to the GTCCRD register before up-counting compare match) (n = 4, 7)

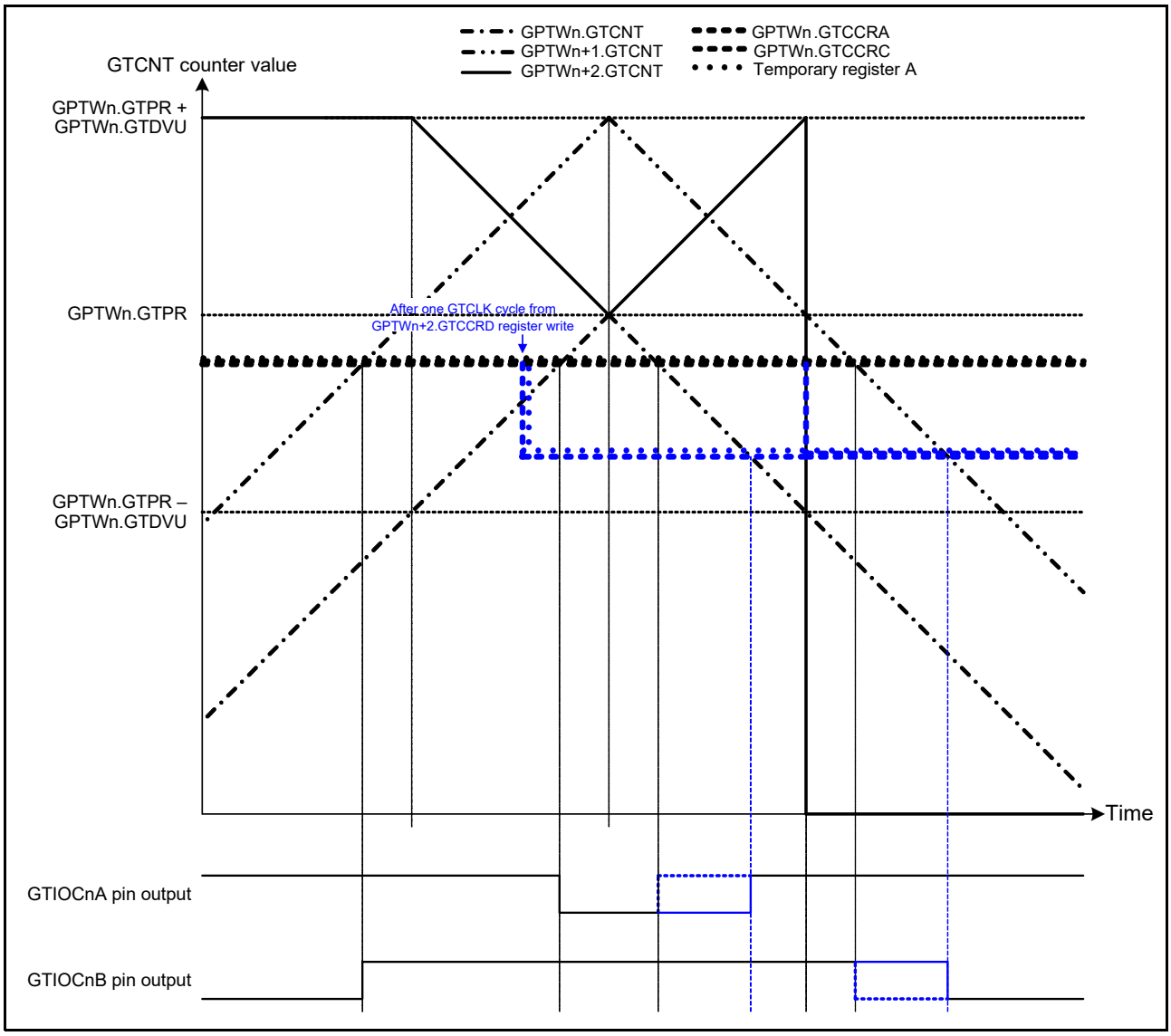


Figure 24.92 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to the GTCCRD register during up-counting dead time) (n = 4, 7)

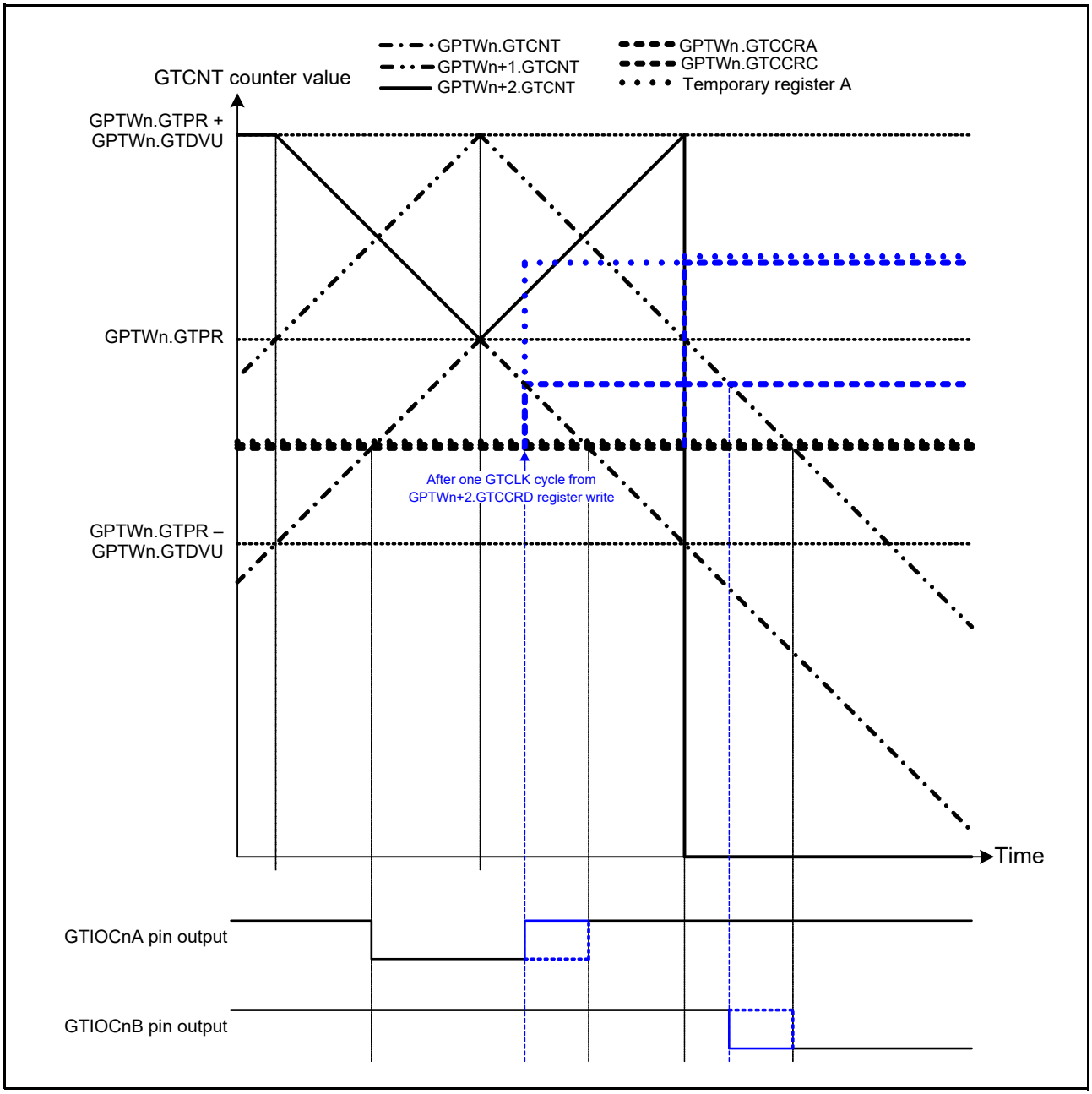


Figure 24.93 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value large than and equal to GPTWn.GTCNT value is written to the GTCCRD register before down-counting compare match and after up-counting compare match) (n = 4, 7)

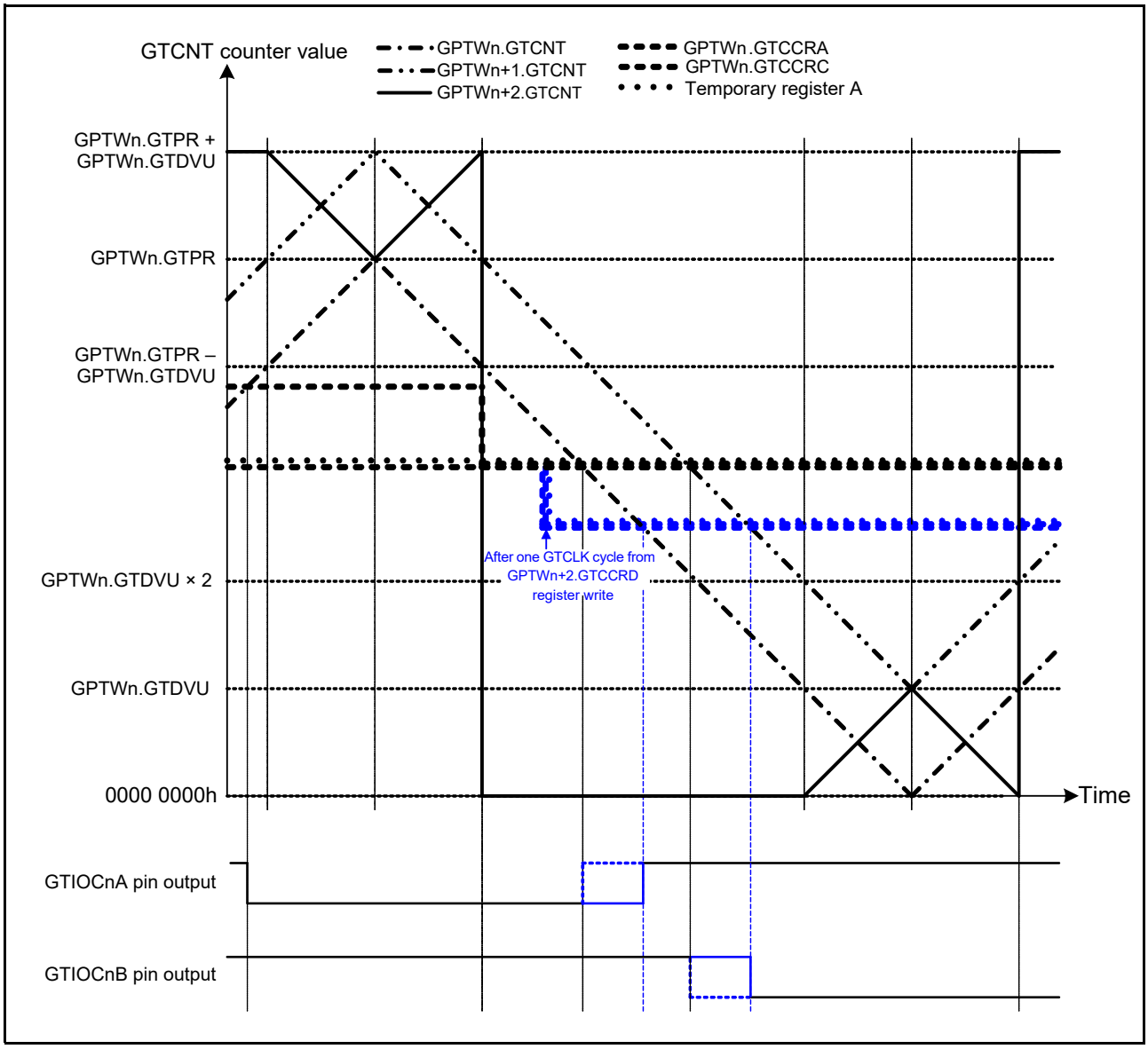


Figure 24.94 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
when a value less than GPTWn.GTCNT value is written to the GTCCRD register before down-counting compare match) (n = 4, 7)

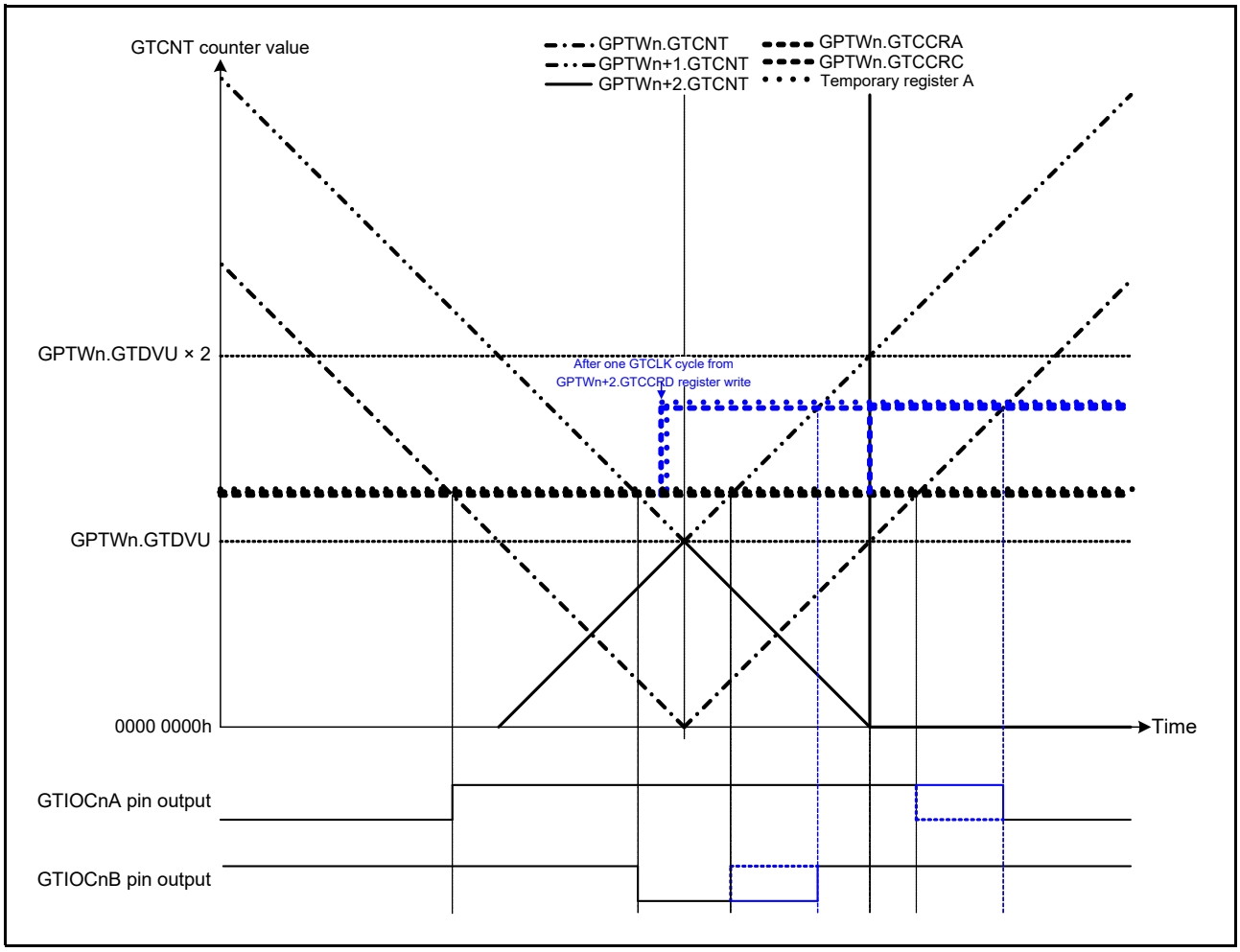


Figure 24.95 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Trough Section)
(Complementary PWM mode 4 single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
when a value large than GPTWn+2.GTCNT value is written to the GTCCRD register after down-counting compare match) (n = 4, 7)

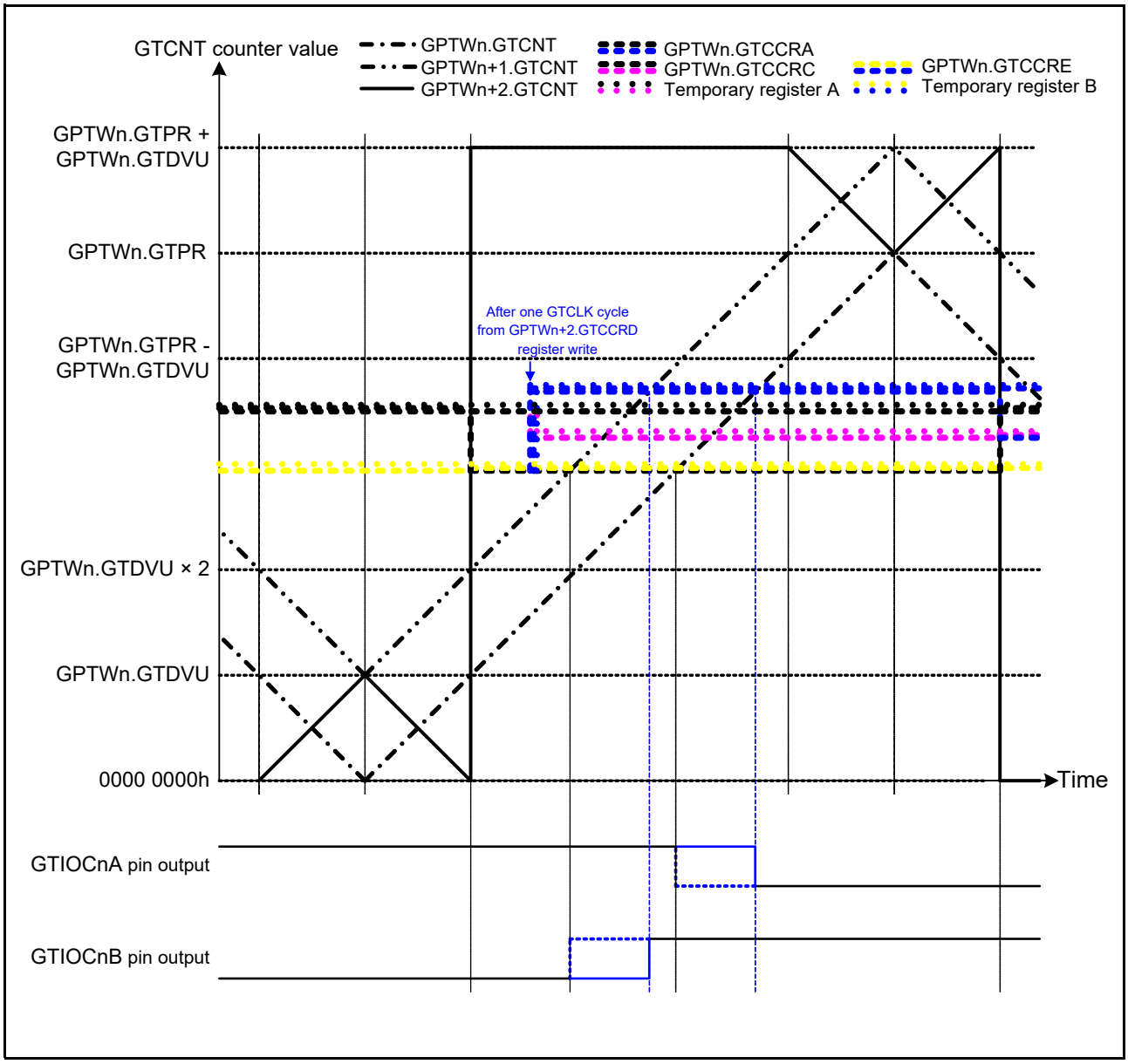


Figure 24.96 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than GPTWn+1.GTCNT is written to the GTCCRF register before the up-counting compare match) (n = 4, 7)

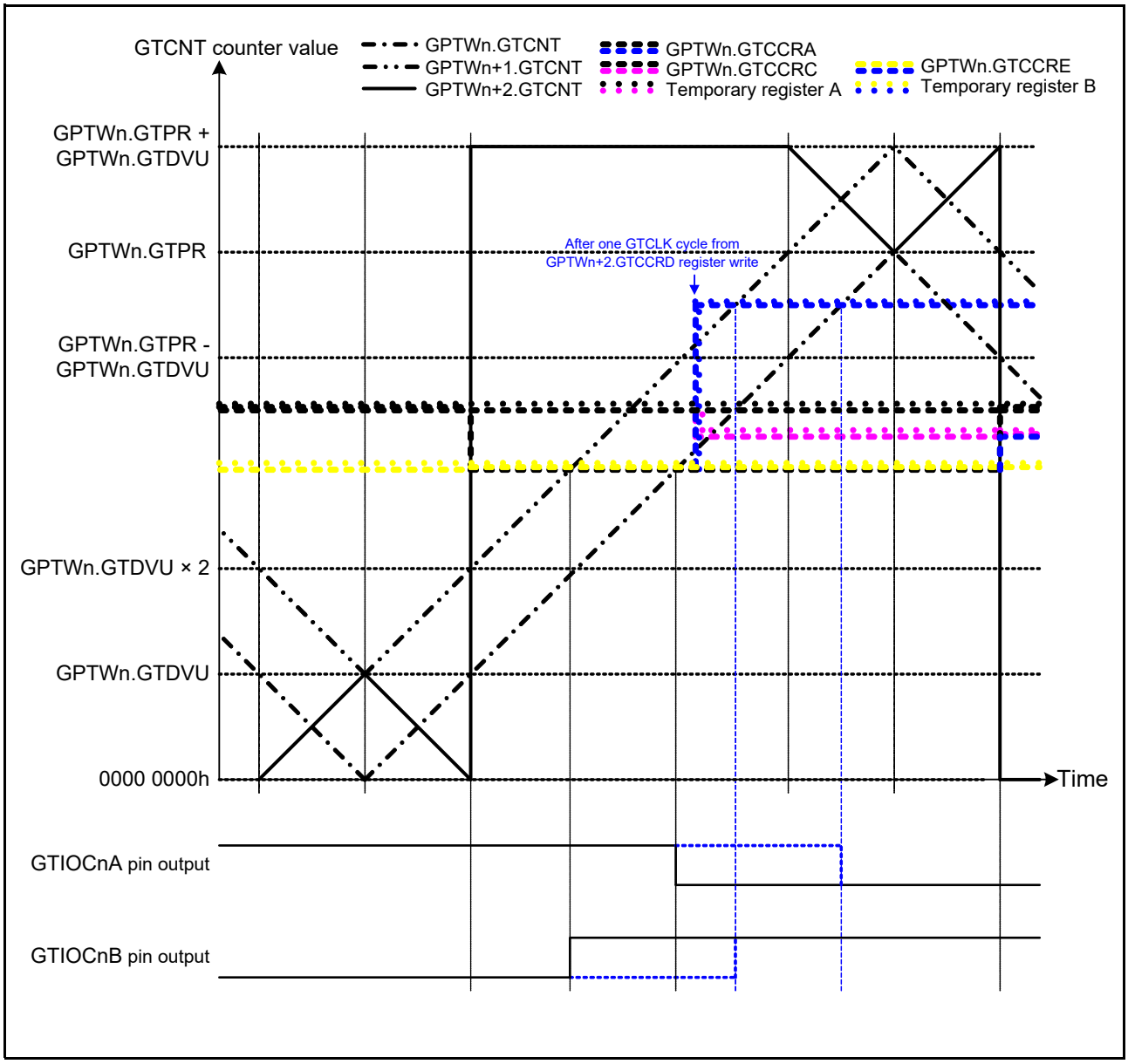


Figure 24.97 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when writing to the GTCCRF register after an up-counting compare match) (n = 4, 7)

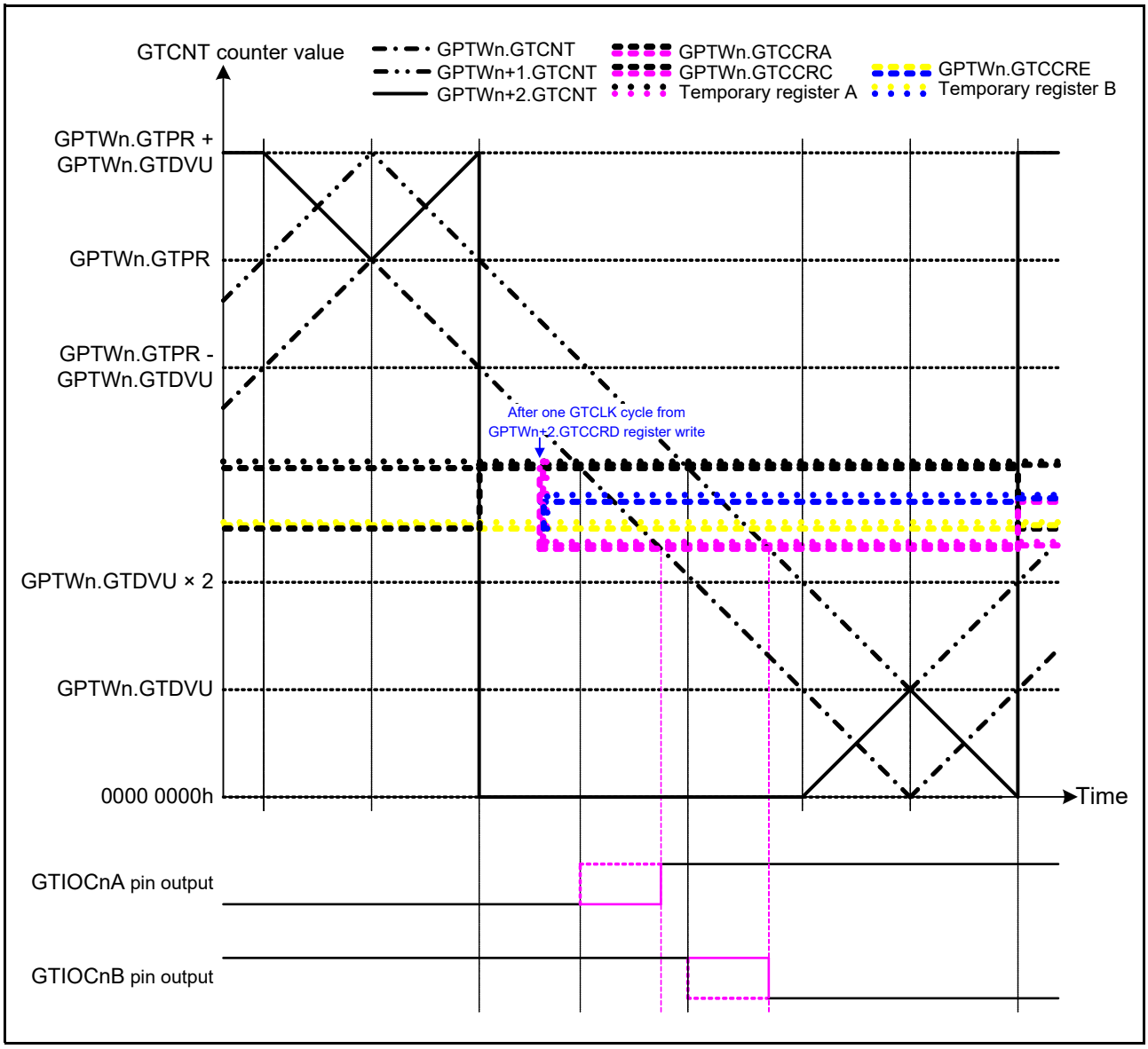


Figure 24.98 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
when a value less than GPTWn.GTCNT is written to the GTCCRD register before the down-counting compare match) (n = 4, 7)

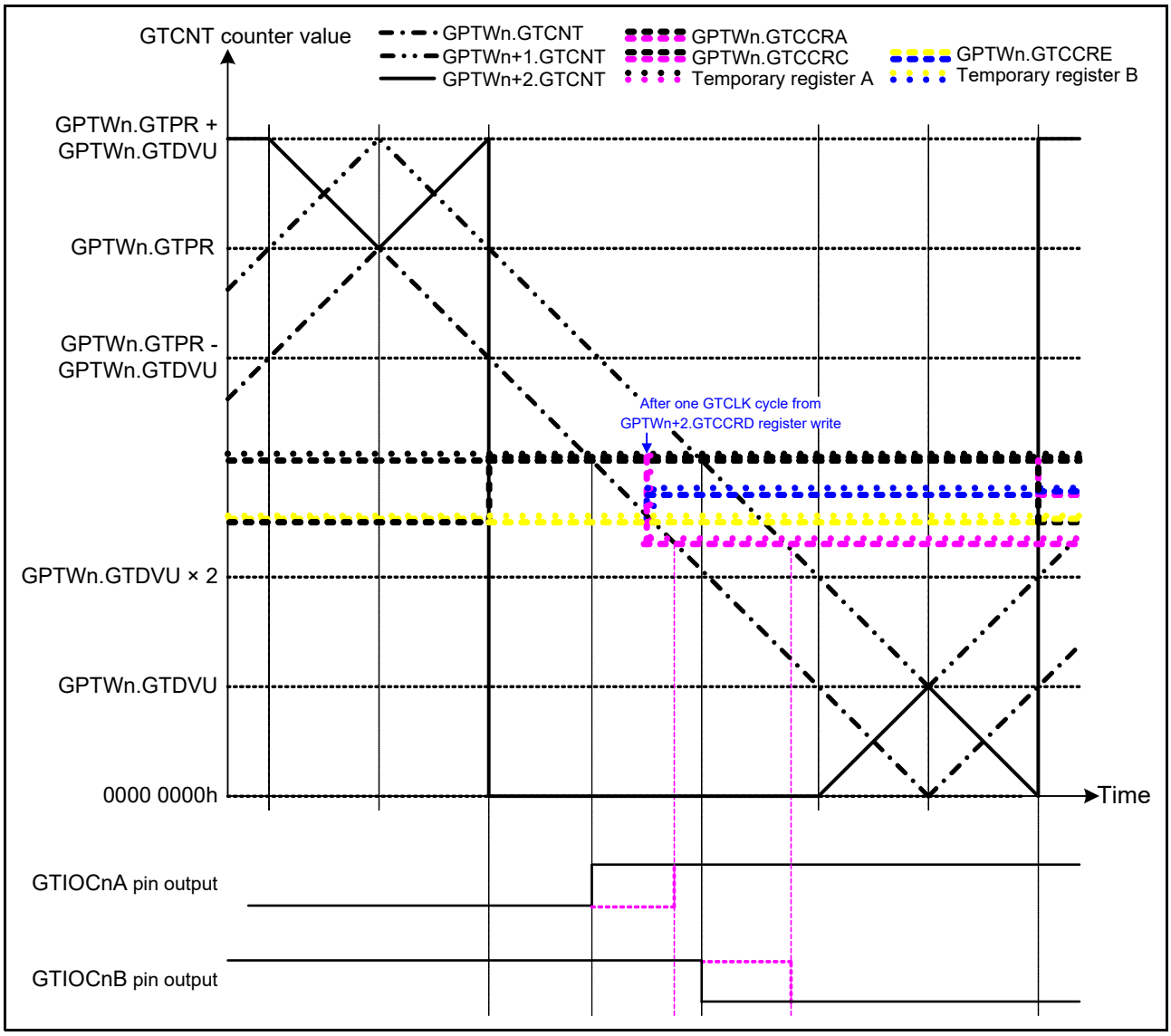


Figure 24.99 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
when writing to the GTCCRD register during the down-counting dead time period) (n = 4, 7)

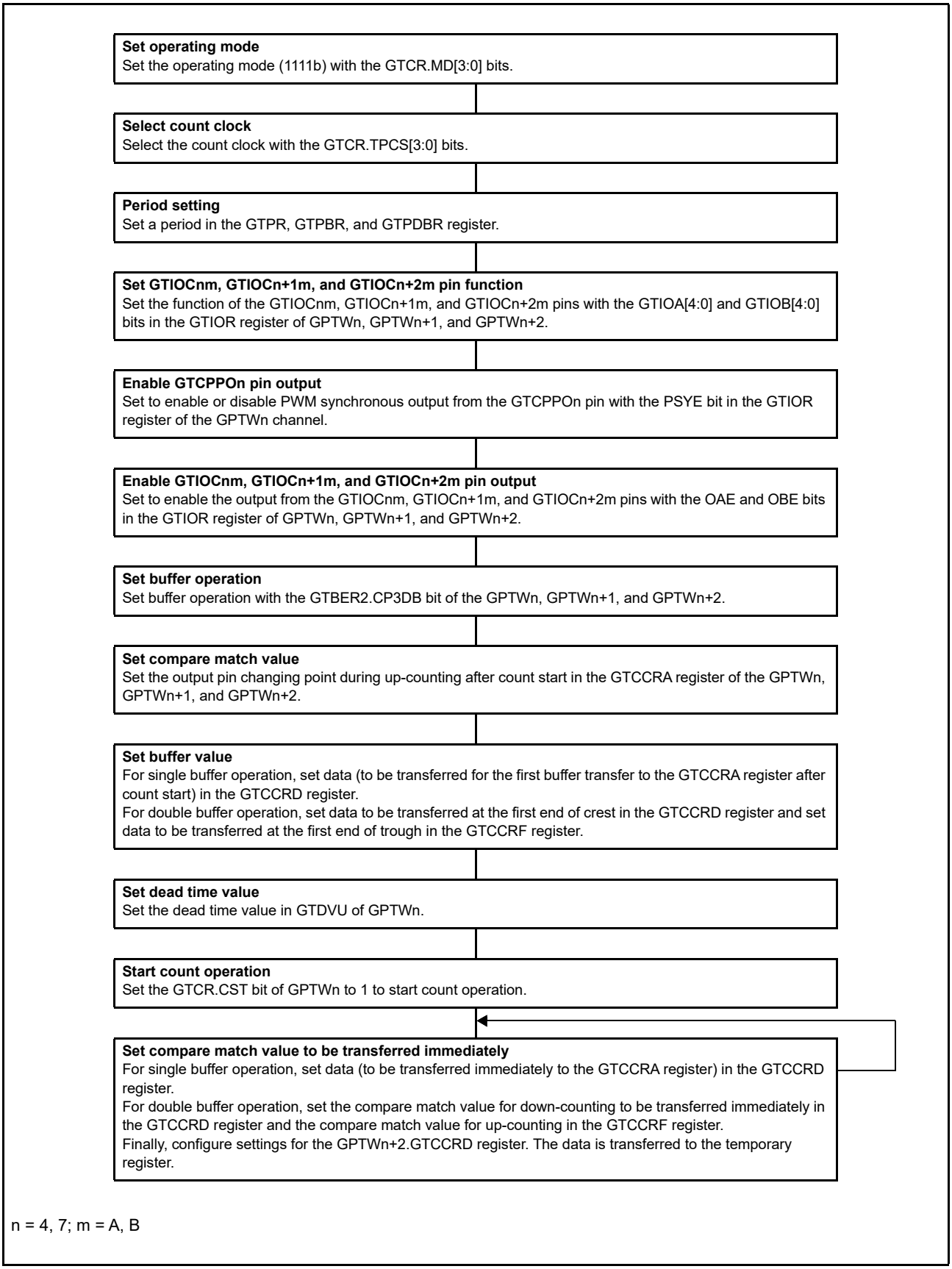


Figure 24.100 Example for Setting Complementary PWM Mode 4

24.3.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA register value) and specified dead time values (the GTDVU and GTDVD registers values) can automatically be set to the GTCCRB register.
The automatic dead time setting function can be used in sawtooth-wave one-shot pulse mode and all the triangle PWM modes.
Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in the GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.
The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD register. Buffer transfer is performed at the end of the cycle (in sawtooth-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing; in triangle-wave mode: a trough).
The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used.

Do not set the dead-time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 24.21. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.
If the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.
If dead-time exceeds the count period by setting 0000 0000h or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register in triangle-wave PWM mode, output change is controlled by the output protection function (refer to section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 15; m = A, B)). When GTCCRA register is greater than or equal to [GTPR register + GTDVm register], [GTPR register – 1] is set in the GTCCRB register as the upper limit. (m = U, D)
Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 24.21 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

PWM Output Operating Mode	Count Direction	Period	Condition for Dead-Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA – GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR (GTCCRA + GTDVU > GTPR)*1	GTPR – GTDVD (GTPR – GTDVU)*1	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR – GTDVU	GTPR
		Second half	GTCCRA – GTDVD < 0 (GTCCRA – GTDVU < 0)*1	GTDVD (GTDVU)*1	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA – GTDVU≤0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA – GTDVD < 0 (GTCCRA – GTDVU < 0)*1	GTDVD (GTDVU)*1	0

Note 1. In the case of GTDTCR.TDFER = 1.

Figure 24.101 to Figure 24.104 show examples of automatic dead time setting function operation. Figure 24.105 and Figure 24.106 show the setting examples.

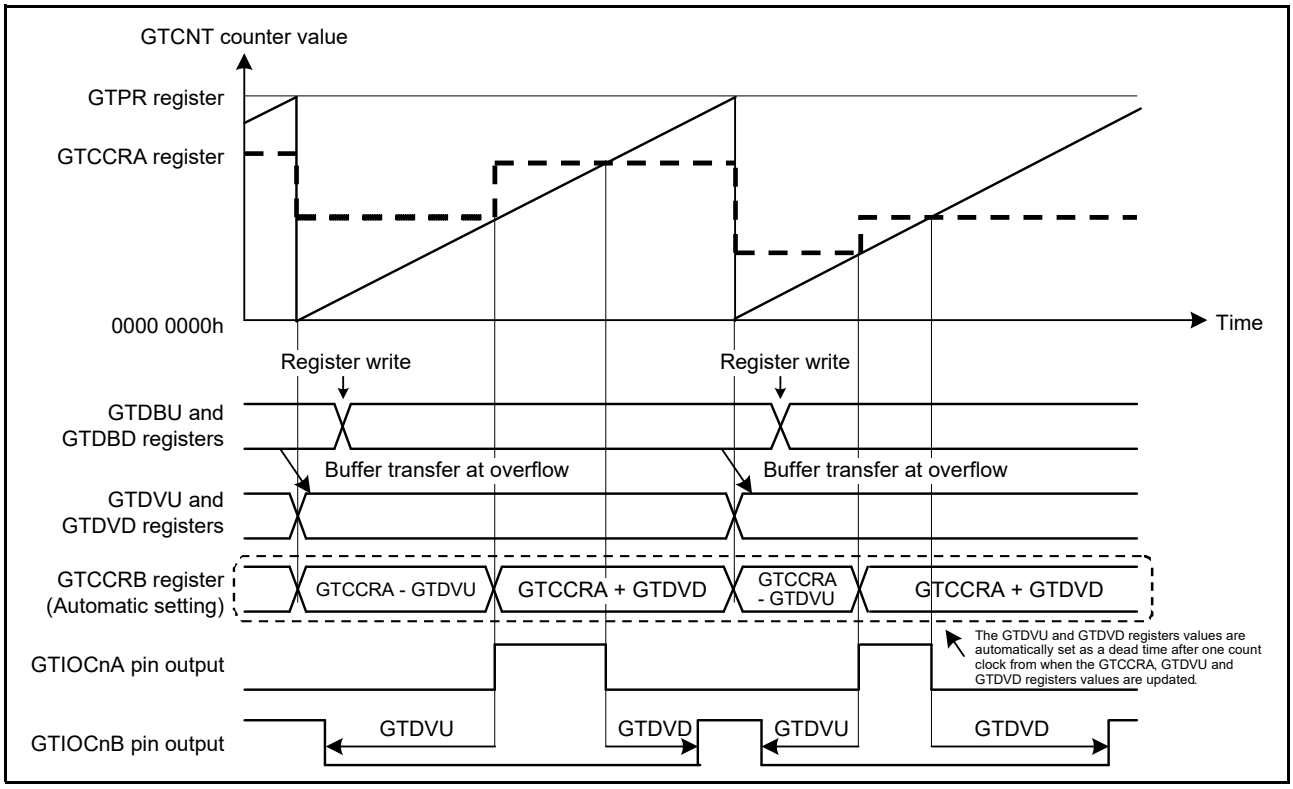


Figure 24.101 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Up-Counting, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 15)

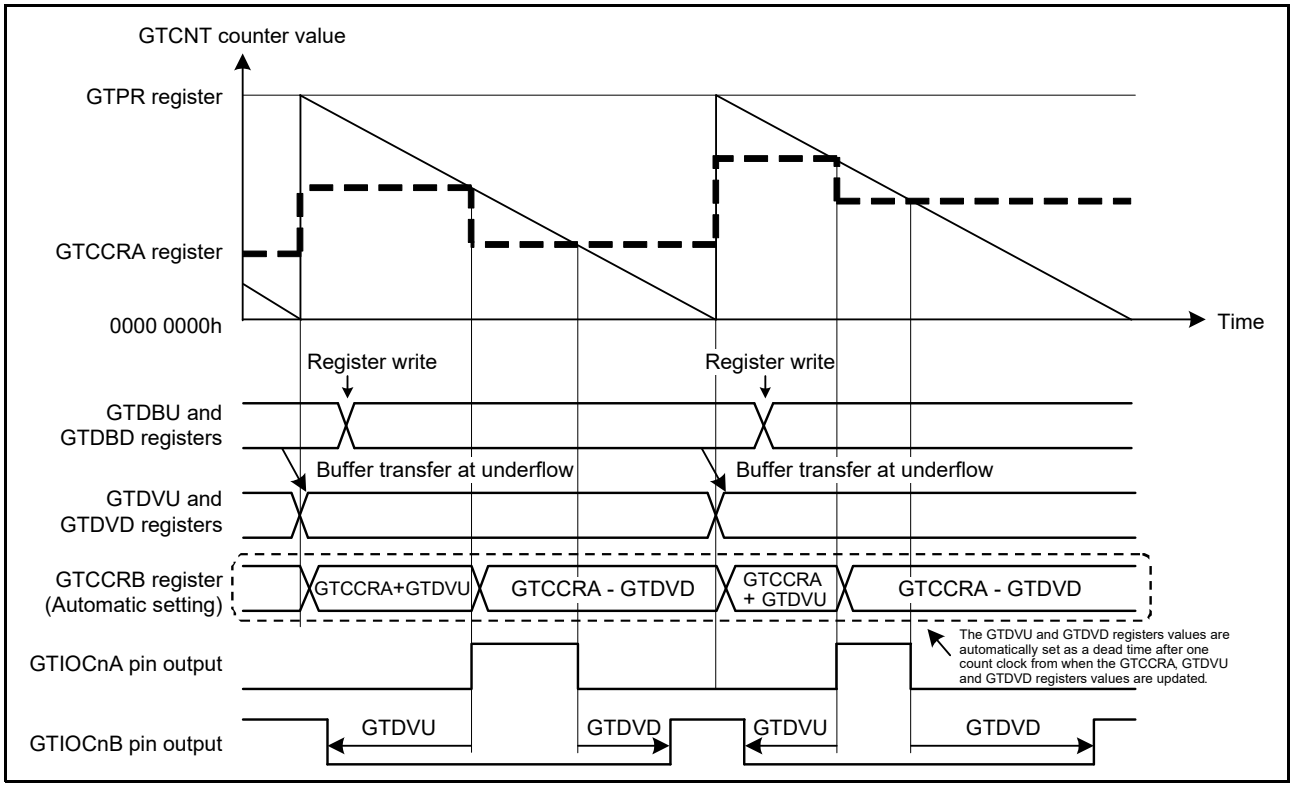


Figure 24.102 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Down-Counting, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 15)

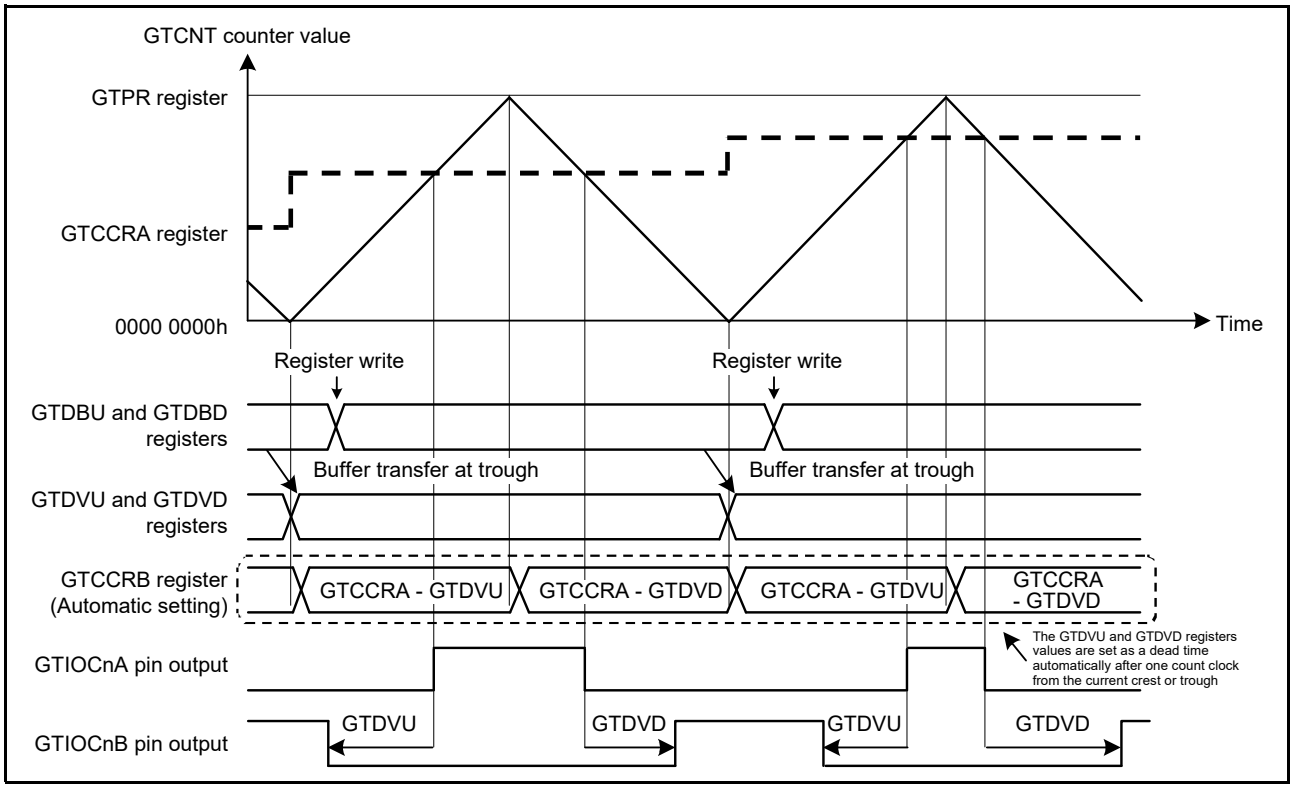


Figure 24.103 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 1, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 15)

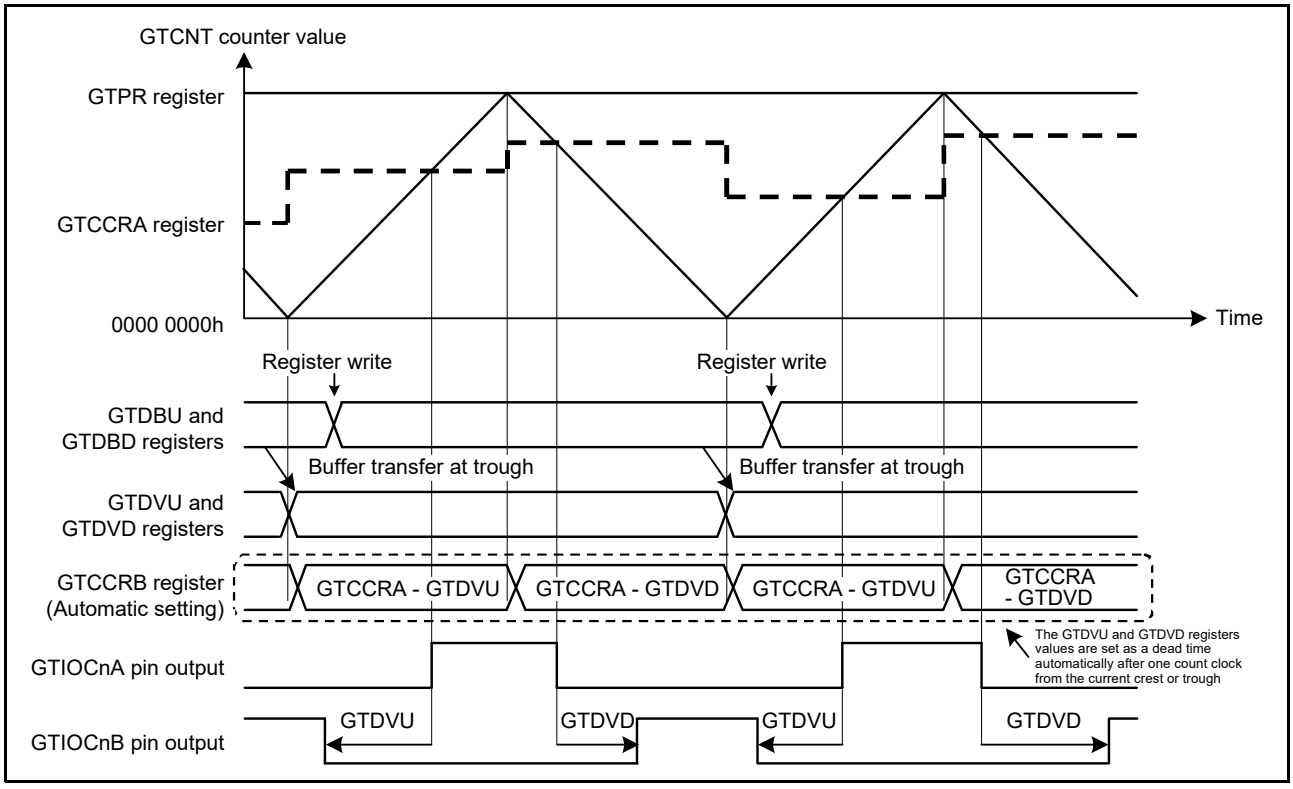
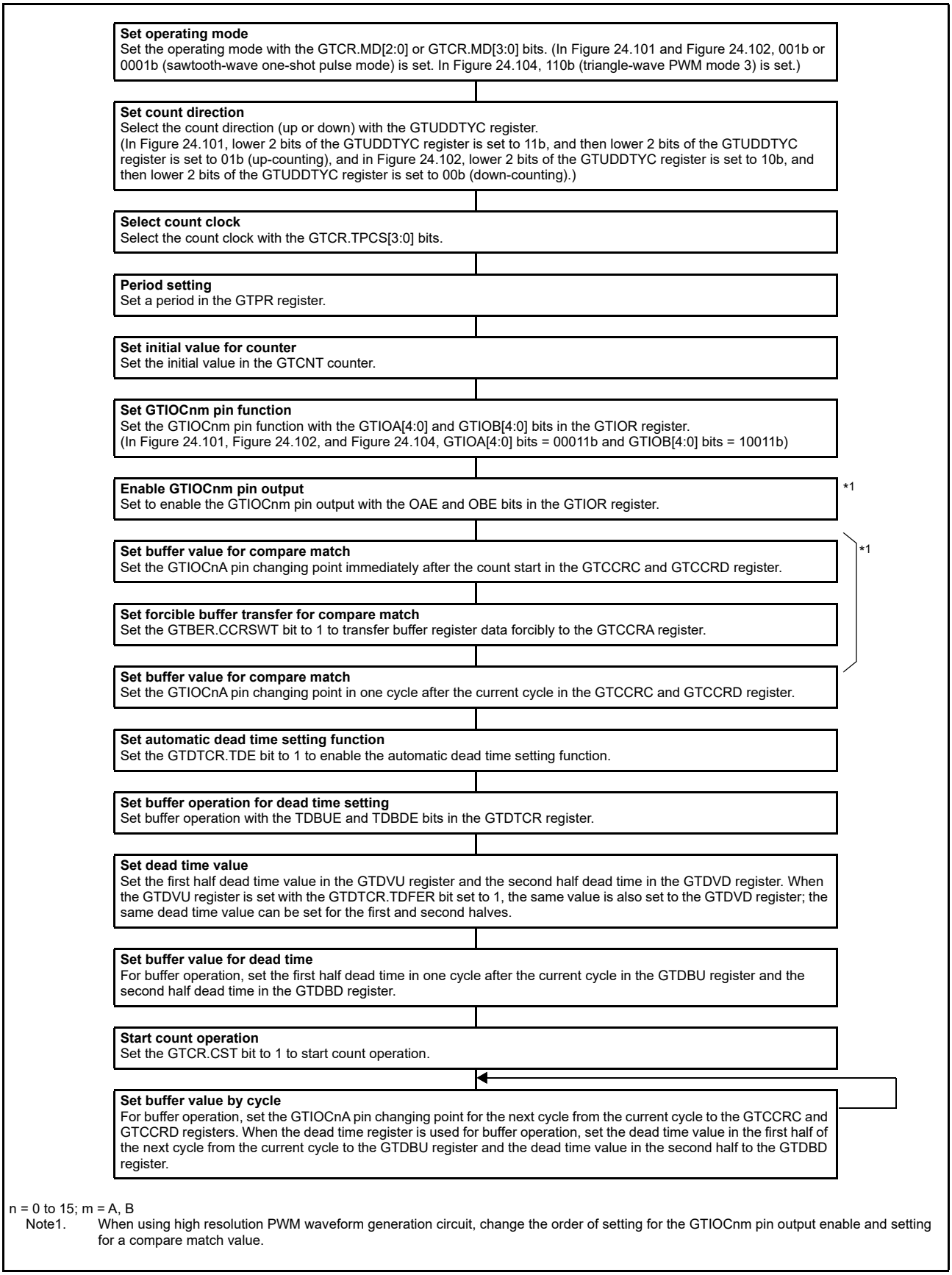


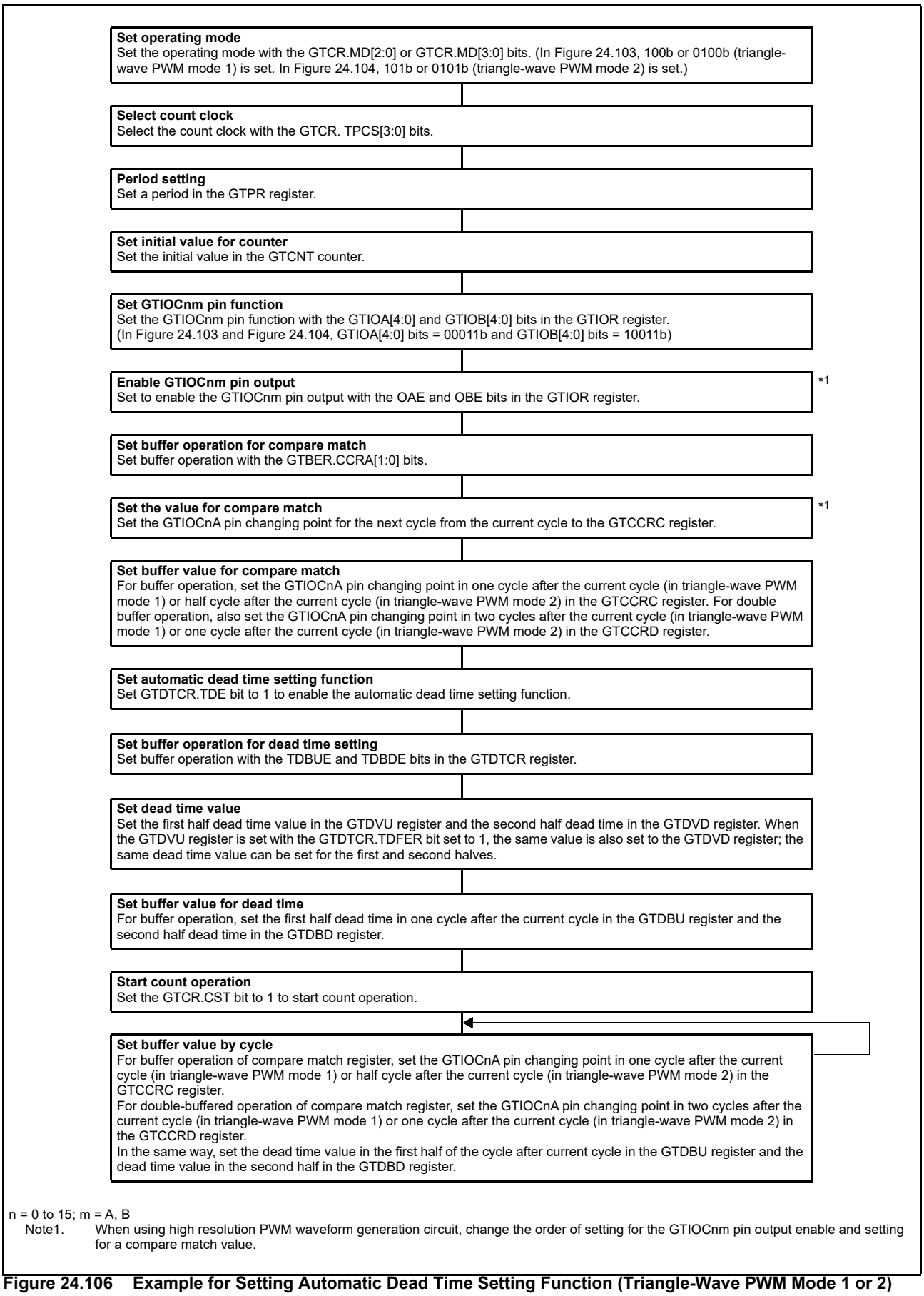
Figure 24.104 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 2 or 3, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 15)



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.105 Example for Setting Automatic Dead Time Setting Function (Sawtooth-Wave One-Shot Pulse Mode, Triangle-Wave PWM Mode 3)



n = 0 to 15; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.106 Example for Setting Automatic Dead Time Setting Function (Triangle-Wave PWM Mode 1 or 2)

24.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDDTYC.UD bit.
In sawtooth-wave mode, if the UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDDTYC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.
In triangle-wave mode, changing the value of the GTUDDTYC.UD bit during counting does not switch the direction of counting. Likewise, changing the value of the GTUDDTYC.UD bit while the GTUDDTYC.UDF bit is 0 and counting is stopped does not actually update the value of the bit. If 1 is written to the GTUDDTYC.UDF bit while counting is stopped, the value of the GTUDDTYC.UD bit at that time is reflected from the time counting is started.
When counting direction is switched during count operation in sawtooth-wave mode, the value of the GTPR register after start of up-counting is reflected to the count period in up-counting operation, and the value of the GTPR register after the start of down-counting is reflected to the count period in down-counting operation.
Figure 24.107 shows an example of count direction changing function operation.

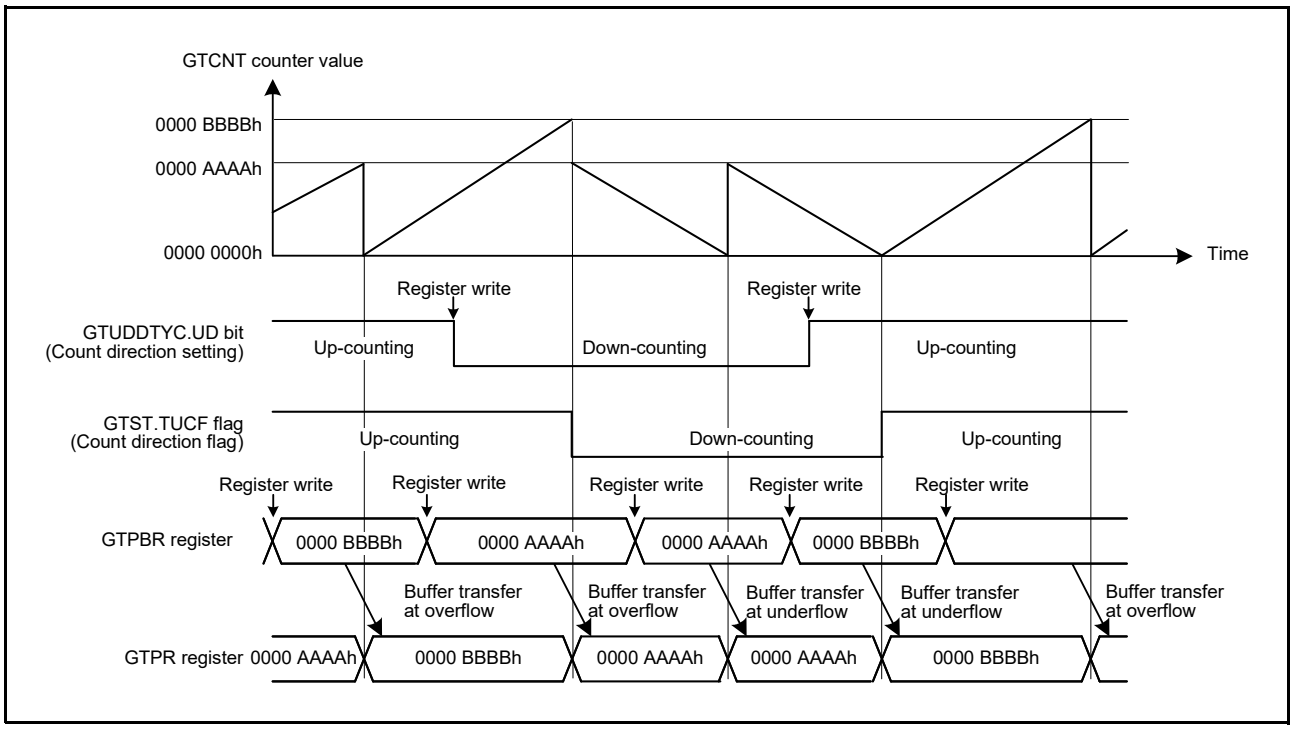


Figure 24.107 Example of Count Direction Changing Function Operation (during Buffer Operation)

24.3.6 Duty Cycle 0%/100% Output Function

Changing the value of the GTUDDTYC.OADTY[1:0] bits and GTUDDTYC.OBDTY[1:0] bits specifies the output duty setting on the GTIOCnA and GTIOCnB pins to 0% or 100%. (n = 0 to 15)

This function is invalid in sawtooth-wave PWM mode 2 or complementary PWM mode. For complementary PWM mode, 0%/100% duty output is possible regardless of the setting of the OADTY[1:0] bits or OBDTY[1:0] bits.

In sawtooth-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at an overflow (when changed during up-counting) or an underflow (when changed during down-counting). When the GTUDDTYC.OADTYF bit or GTUDDTYC.OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the updated output duty-cycle setting is not applied at the count start, but it is applied at an overflow or underflow. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

In triangle-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during count operation, the updated output duty-cycle setting is applied at an underflow when the GTUDDTYC.OABDTYT bit is set to 0. When the GTUDDTYC.OABDTYT bit is set to 1, the updated output duty-cycle setting is applied at an underflow or at the top of the triangle-wave.

When the OADTYF bit or OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the updated output duty-cycle setting is not applied at the start of counting, but is applied at an underflow. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the updated output duty-cycle setting is applied from the start of counting.

During the operation under the settings of duty cycle for 0%/100%, compare match operation inside the GPTW continues to perform interrupt outputs and buffer operations.

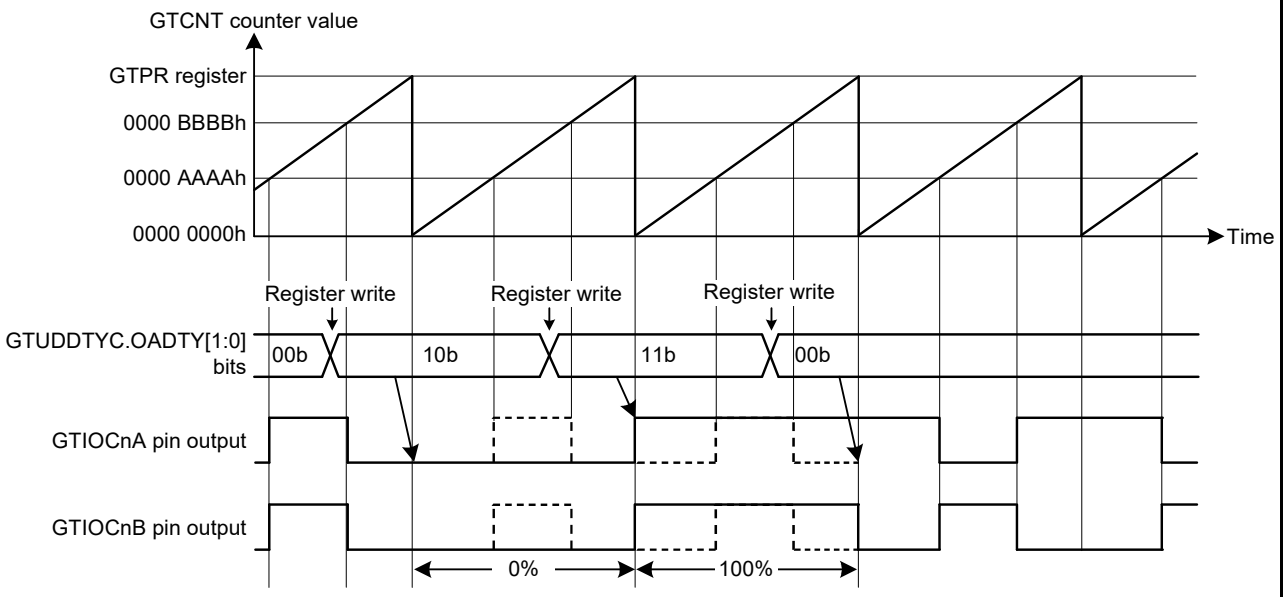
When the output duty setting is changed from 0% or 100% to a compare match, the GTIOR.GTIOA[3:2] bits and the OADTYR bits determine the level output on the GTIOCnA pin and GTIOR.GTIOB[3:2] bits and OBDTYR bits determine the level output on the GTIOCnB pin at the end of the cycle. When the GTIOA[3:2], GTIOB[3:2] bits are set as 01b, output is driven Low at the end of the cycle, and when set as 10b, output is driven High at the end of the cycle. The output is retained at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 00b, and toggled at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 11b. A value to be output is selectable in the OADTYR or OBDTYR bit when the corresponding GTIOA[3:2] or GTIOB[3:2] bits are either 00b or 11b. Table 24.22 lists the output values at the end of the cycle when the output setting is changed from duty 0% or 100% to compare match.

Table 24.22 Output Value after Release of Duty 0%/100%

GTIOR.GTIOm[3:2]	Value at Compare Match Output at the End of the Cycle in the Case of Masking by a Duty Cycle 0% or 100%	The GTUDDTYC.OmDTYR Bit at Duty 0% Setting		The GTUDDTYC.OmDTYR Bit at Duty 100% Setting	
		0	1	0	1
00b (Output retained at the end of the cycle)	Low	0	0	1	0
	High	0	1	1	1
01b (Low output at the end of the cycle)	—	0	0	0	0
10b (High output at the end of the cycle)	—	1	1	1	1
11b (Toggle output at the end of the cycle)	Low	1	1	0	1
	High	1	0	0	0

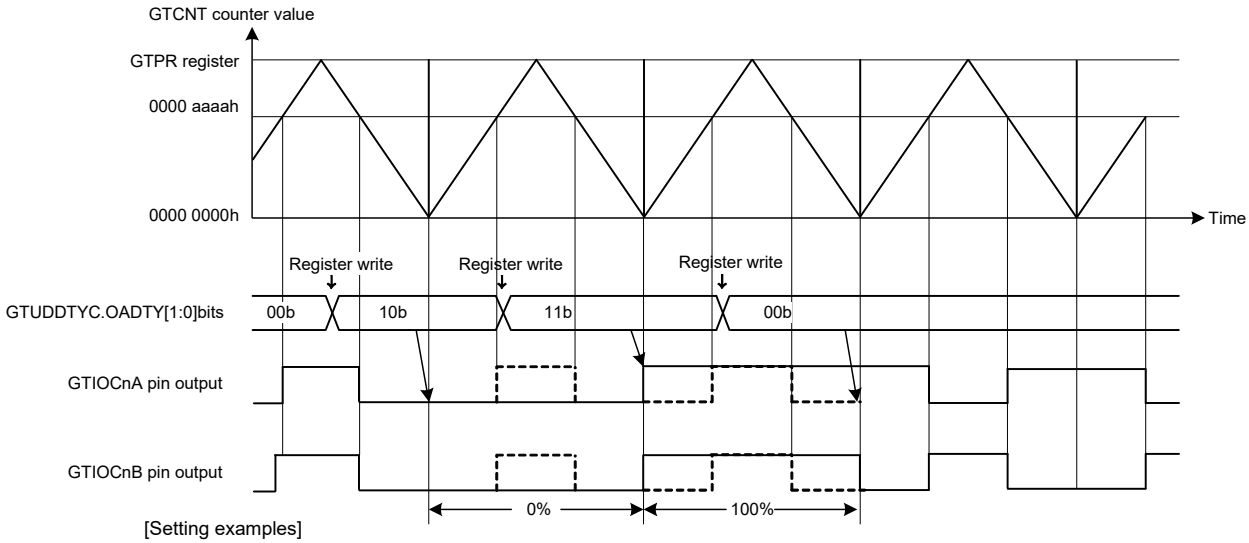
m = A, B

Figure 24.108 to Figure 24.110 shows an example of operation of the output of the duty cycle 0% or 100% (n = 0 to 15).



[Setting examples]
GTIOR.GTIOA[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.
GTUDDTYC.OADTYR bit: 0
The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.
GTIOR.GTIOB[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.
GTUDDTYC.OBDTYR bit: 1
The function of the GTIOB[3:2] bits is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 24.108 Example of Operation of Output of Duty Cycle 0% or 100% (n = 0 to 15)



[Setting examples]

GTIOR.GTIOA[4:0] bits: 00011b
The initial output is low, the output is toggled on a compare match, and the output is maintained at the end of the period.

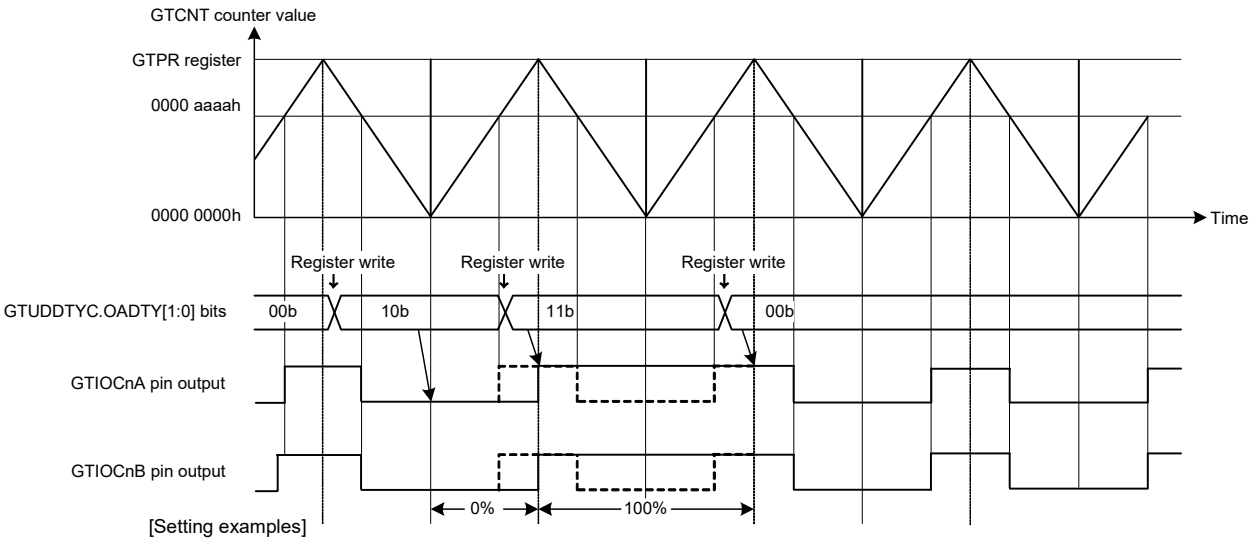
GTUDDTYC.OADTYR bit: 0
The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.

GTIOR.GTIOB[4:0] bits: 00011b
The initial output is low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OBDTYR bit: 1
The function of the GTIOB[3:2] bits is applied to the compare match output value which is masked after 0 or 100% duty setting release.

GTUDDTYC.OABDTYT bit: 0
The activation timing of 0 or 100% duty setting is set to the trough.

Figure 24.109 Example of Operation of Output of Duty Cycle 0% or 100% (Activate at Trough) in Triangle-Wave PWM Mode 1 (n = 0 to 15)



[Setting examples]

GTIOR.GTIOA[4:0] bits: 00011b
The initial output is low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OADTYR bit: 0
The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.

GTIOR.GTIOB[4:0] bits: 00011b
The initial output is low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OBDTYR bit: 1
The function of the GTIOB[3:2] bits is applied to the compare match output value which is masked after 0 or 100% duty setting release.

GTUDDTYC.OABDTYT bit: 1
The activation timing of 0 or 100% duty setting is set to the trough and crest.

Figure 24.110 Example of Operation of Output of Duty Cycle 0% or 100% (Activate at Trough, Crest) in Triangle-Wave PWM Mode 1 (n = 0 to 15)

24.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources in this MCU. There are 3 types of hardware sources, including external trigger input, ELC event input, and GTIOCnm pin input (n = 0 to 15; m = A, B)

24.3.7.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTSSR register, and enable to start counting.
Figure 24.111 shows an example of count start operation by a hardware source. Figure 24.112 shows the setting example.

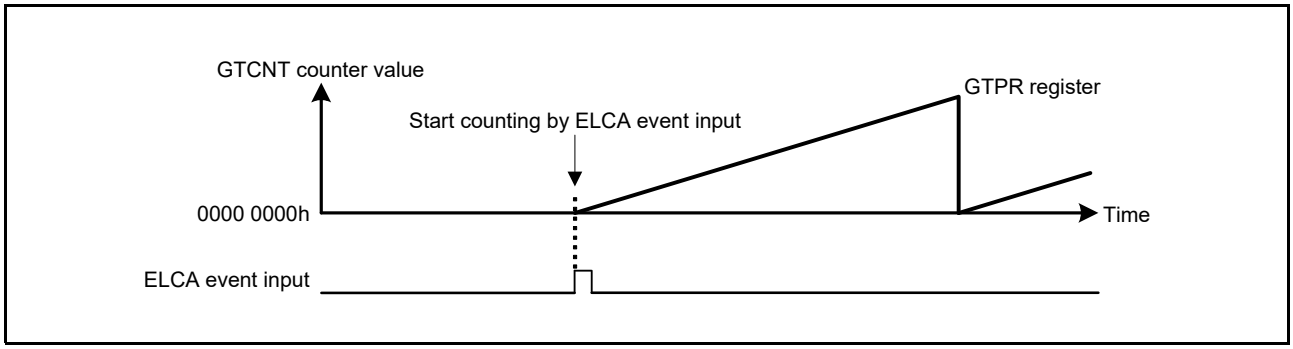


Figure 24.111 Example of Count Start Operation by Hardware Source (At starting by ELCA event)

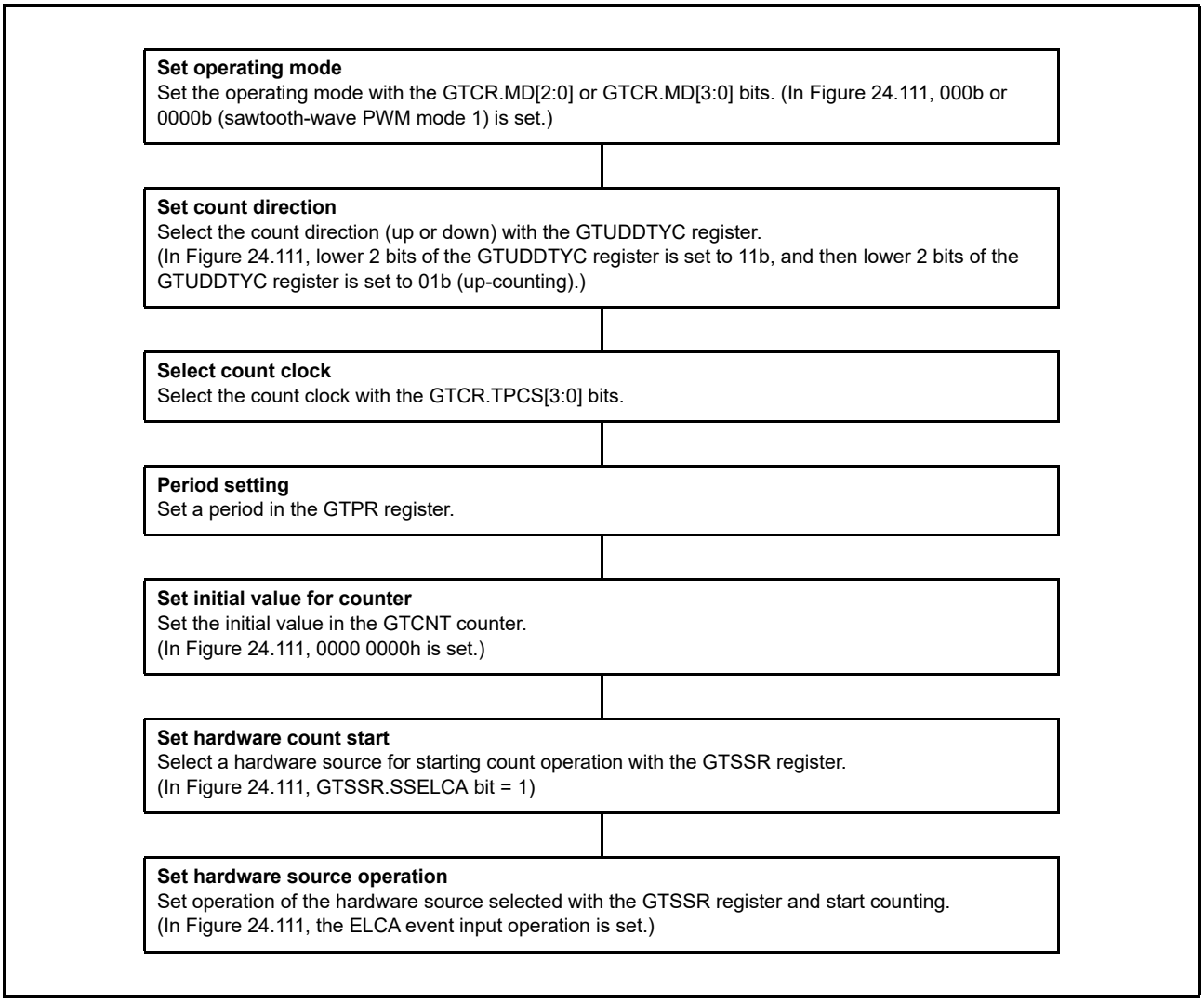


Figure 24.112 Example for Setting Count Start Operation by Hardware Source

Figure 24.113 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin.

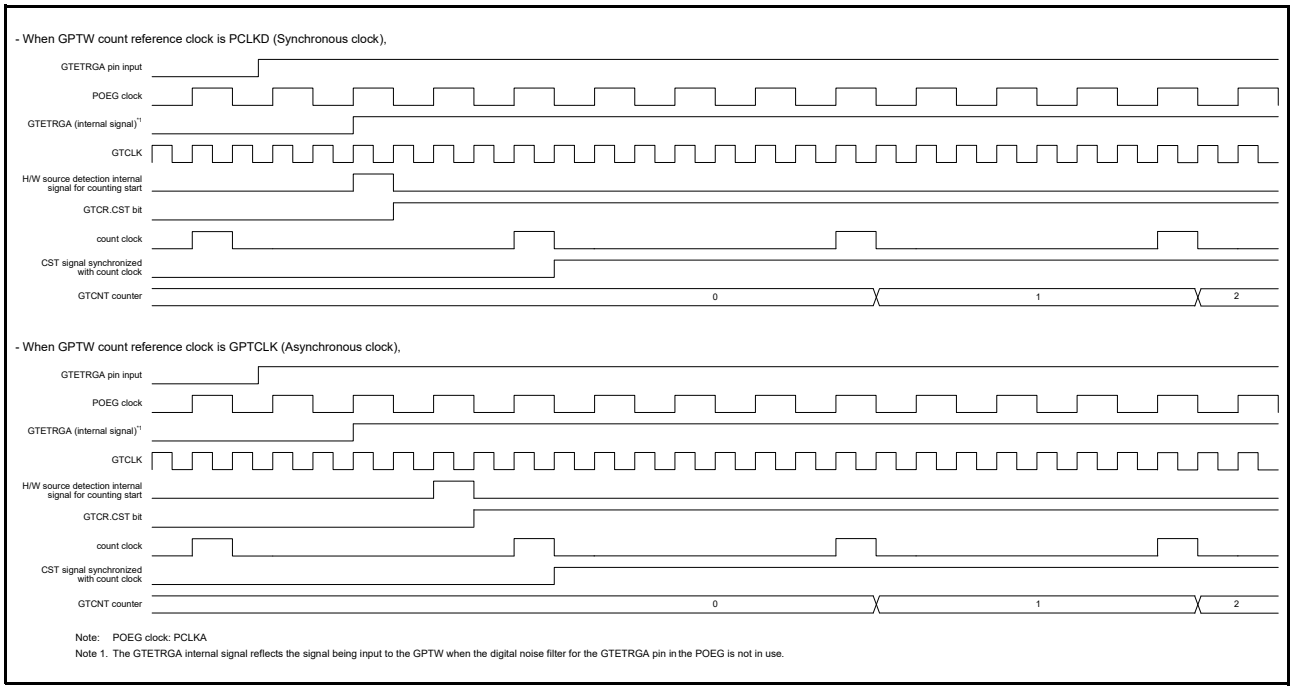


Figure 24.113 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.114 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin.

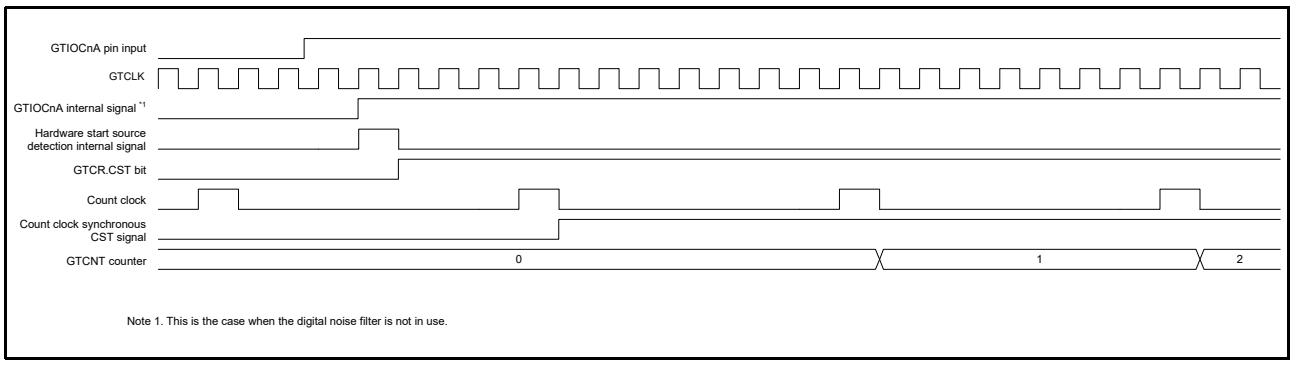


Figure 24.114 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.115 shows an example of timing of operations to start counting in response to event input from the ELCA. This is an example of operations to start counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

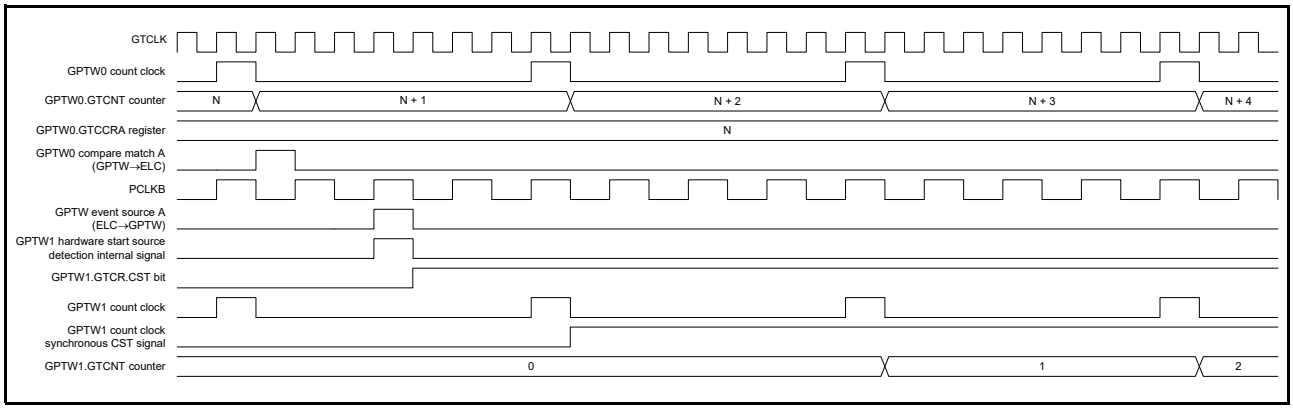


Figure 24.115 Example of Timing of Operations to Start Counting in Response to Event Input from the ELCA

24.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting with the GTPSR register, and enable to stop counting.

Figure 24.116 shows an example of count stop operation by a hardware source. Figure 24.117 shows the setting example. In this example, the count operation is stopped by the ELCA event input, and is restarted by the ELCB event input.

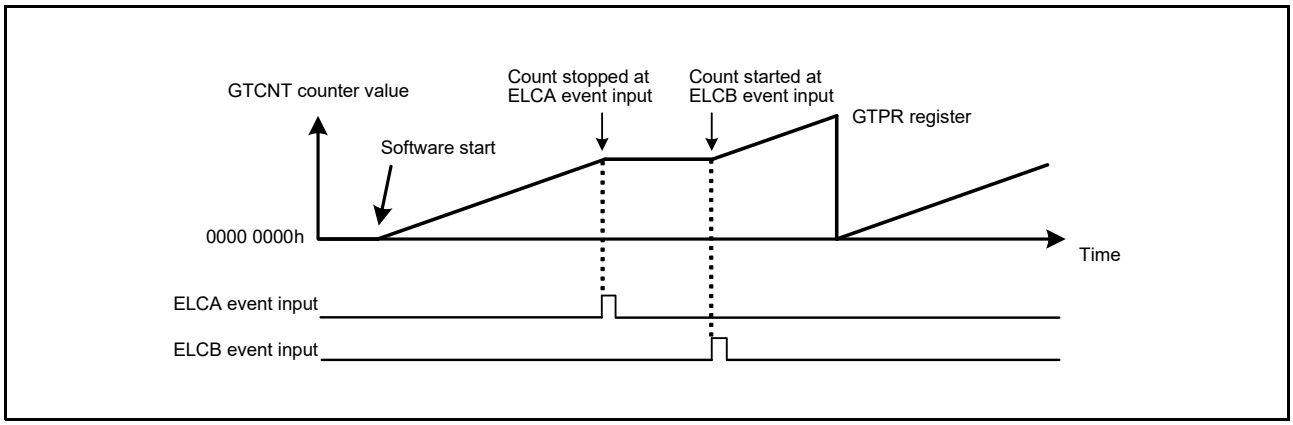


Figure 24.116 Example of Count Stop Operation by Hardware Source (Started by Software, Stopped at ELCA event input, and Restarted at ELCB event input)

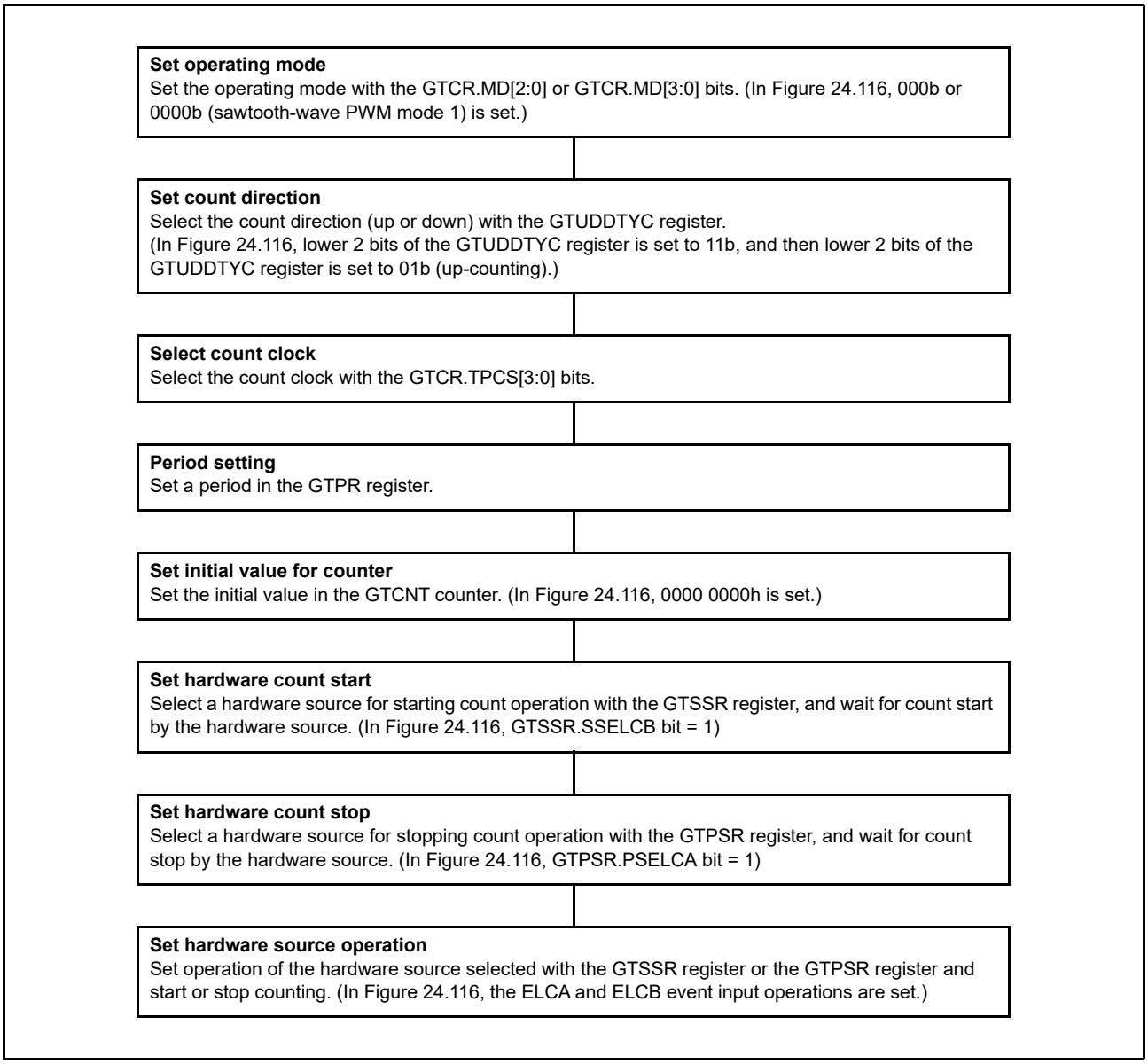


Figure 24.117 Example for Setting Count Stop Operation by Hardware Source

Figure 24.118 shows an example of count start/stop operation by a hardware source. Figure 24.119 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA pin.

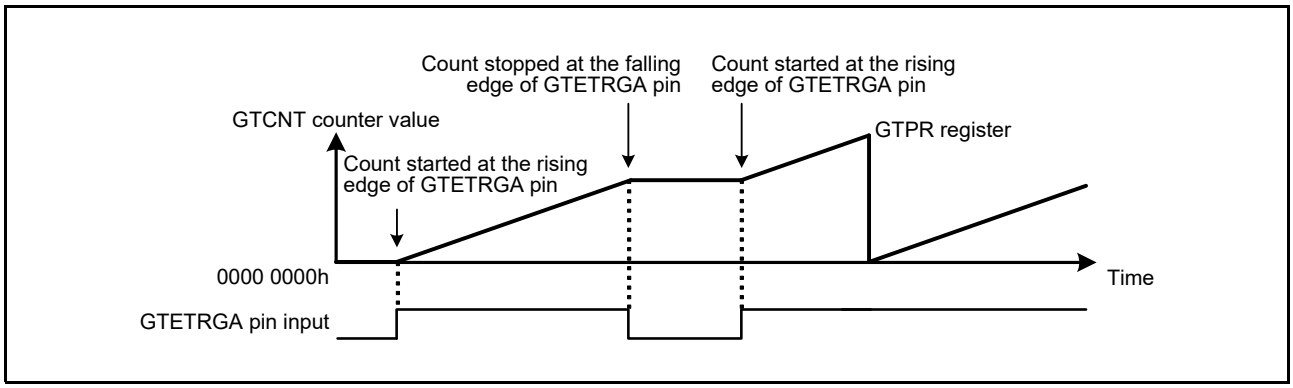


Figure 24.118 Example of Count Start/Stop Operation by Hardware Source (Started at Rising Edge of GTETRGA Pin Input, Stopped at Falling Edge of GTETRGA Pin Input)

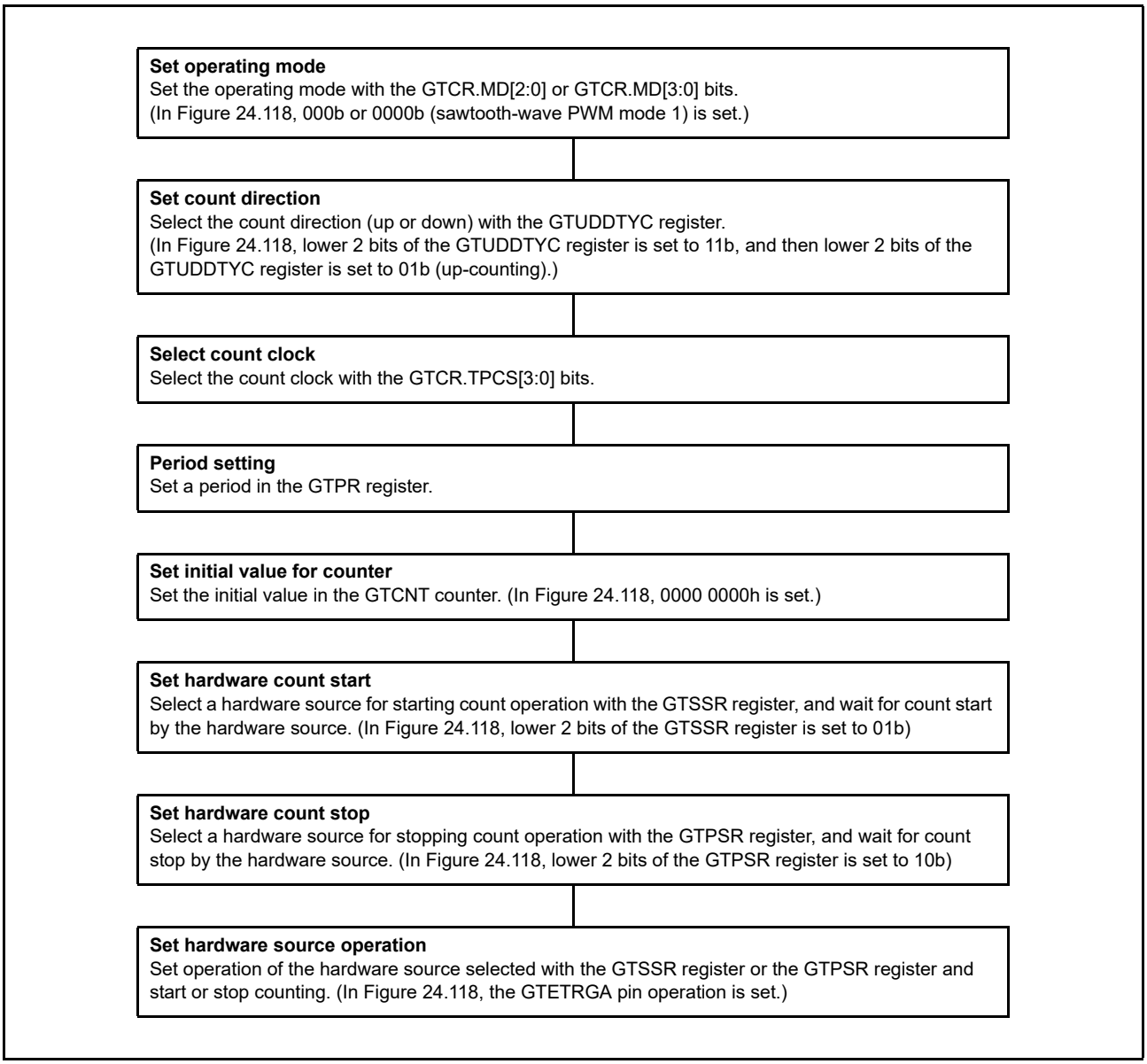


Figure 24.119 Example for Setting Count Start/Stop Operation by Hardware Source

Figure 24.120 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin.

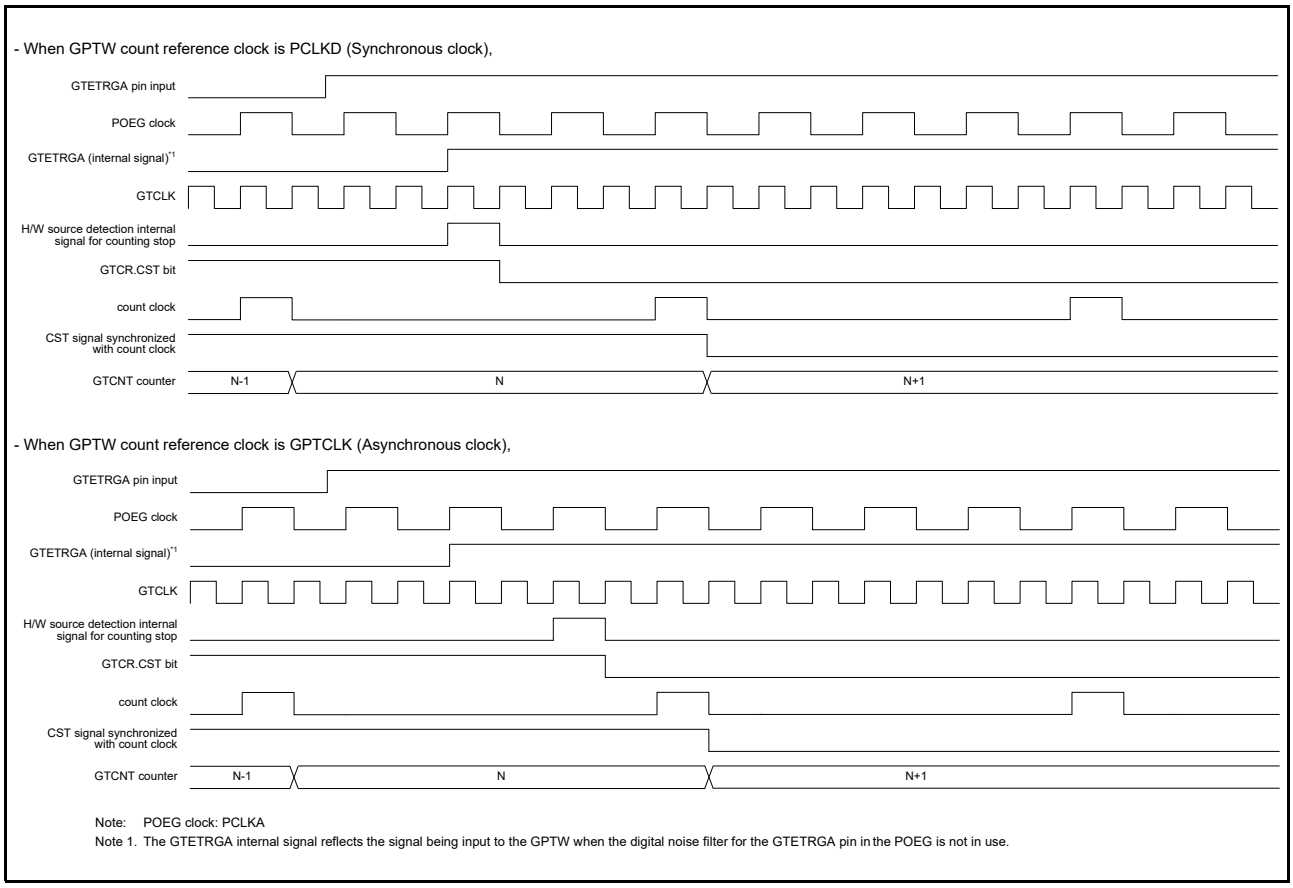


Figure 24.120 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.121 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin.

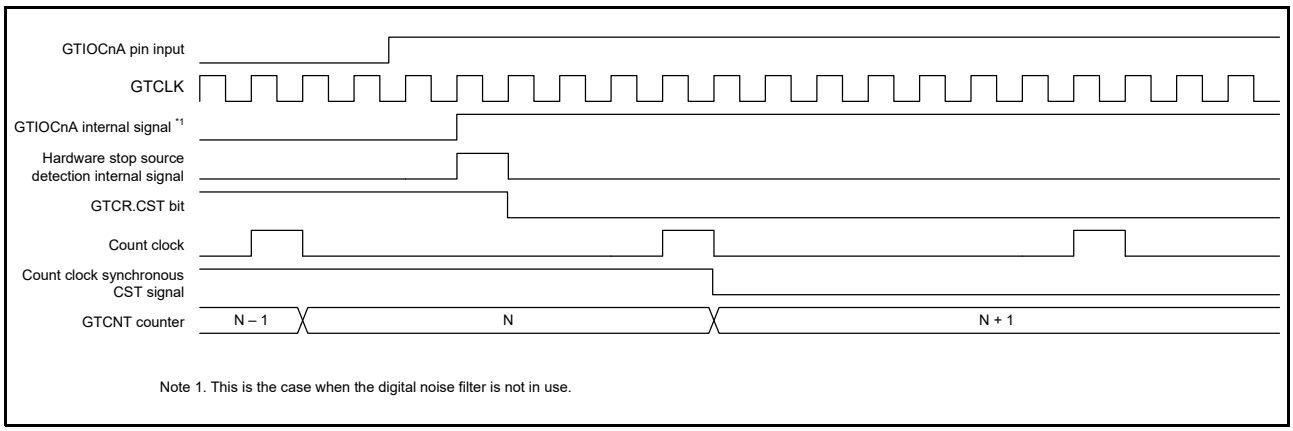


Figure 24.121 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.122 shows an example of timing of operations to stop counting in response to event input from the ELCA. This is an example of operations to stop counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

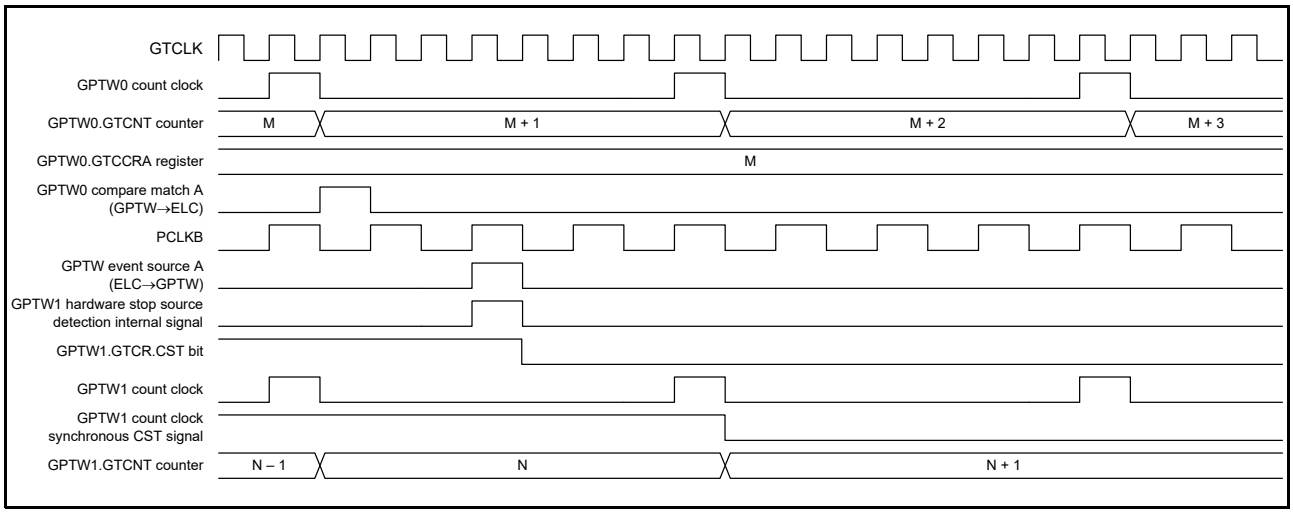


Figure 24.122 Example of Timing of Operations to Stop Counting in Response to Event Input from the ELCA

24.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter with the GTCSR register, and enable clearing the counter.

Note that the GTCIV/GTCIU interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 24.123 and Figure 24.124 show examples of the GTCNT counter clearing operation by a hardware source. Figure 24.125 shows the setting example. In this example, the GTCNT counter is started at the ELCA event input, and is stopped/cleared at the ELCB event input.

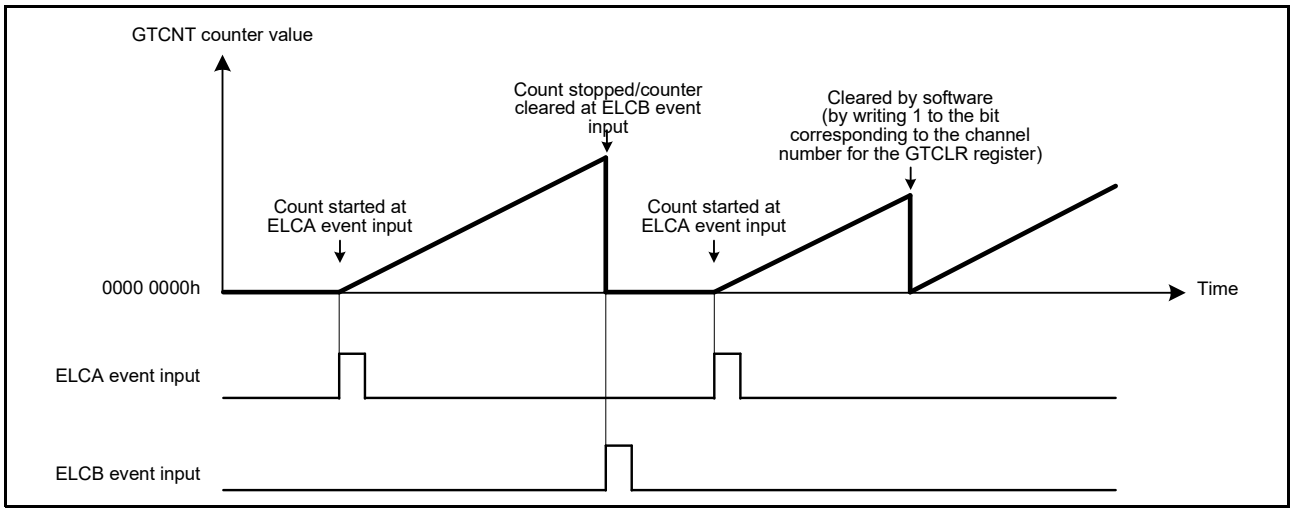


Figure 24.123 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Up-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

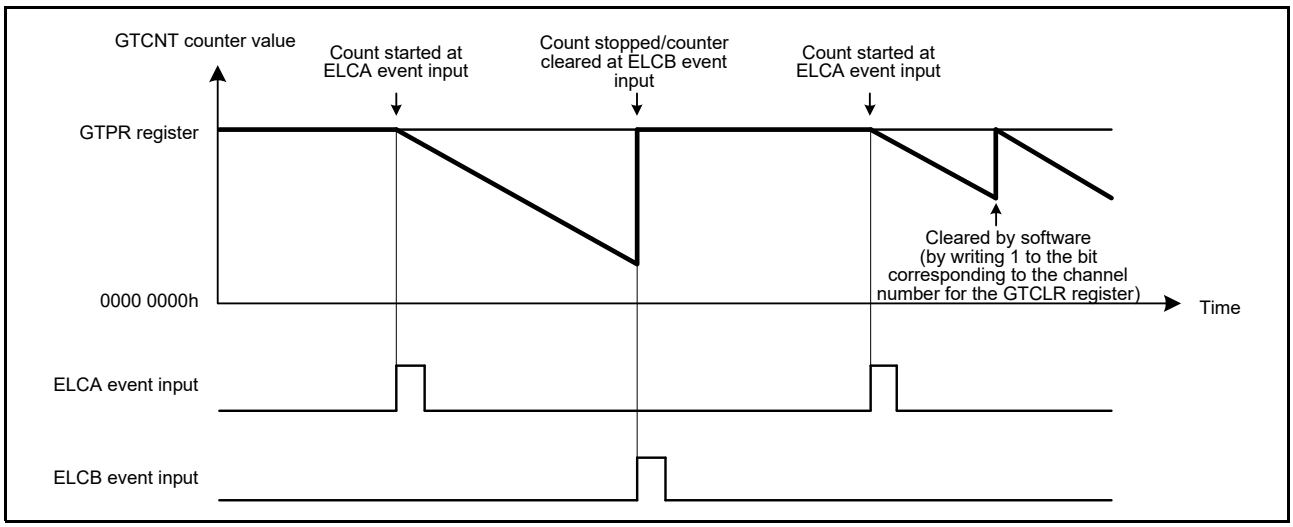


Figure 24.124 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Down-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

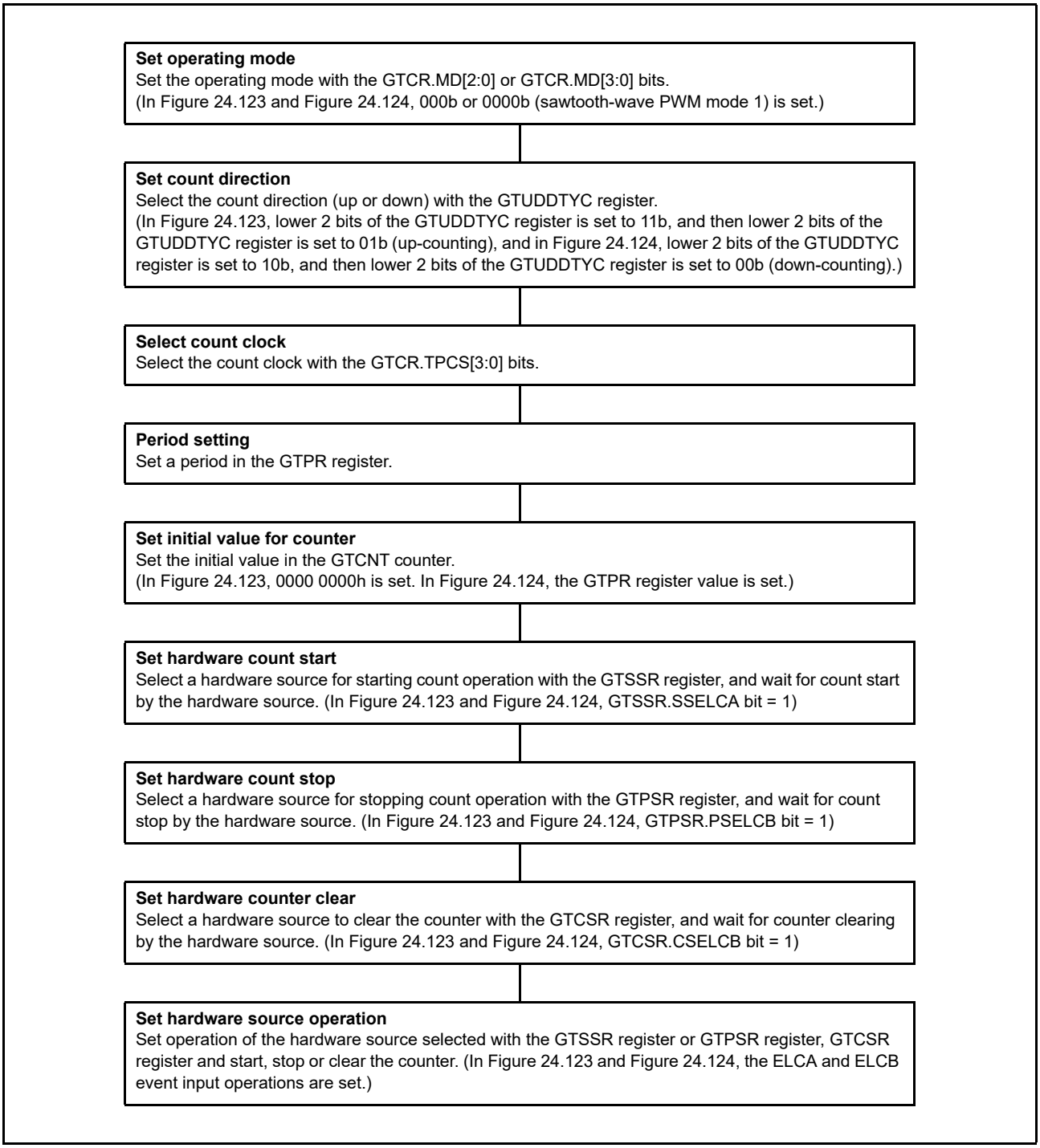


Figure 24.125 Example for Setting Counter Clearing Operation by Hardware Source

The GTCIV/GTCIU interrupt (overflow/underflow interrupt) does not occur when the counter is cleared by a hardware source. In the same way, the GTCIV/GTCIU interrupt does not occur when the counter is cleared by software.
Figure 24.126 shows the relationship between counter clearing by a hardware source and the GTCIV interrupt.

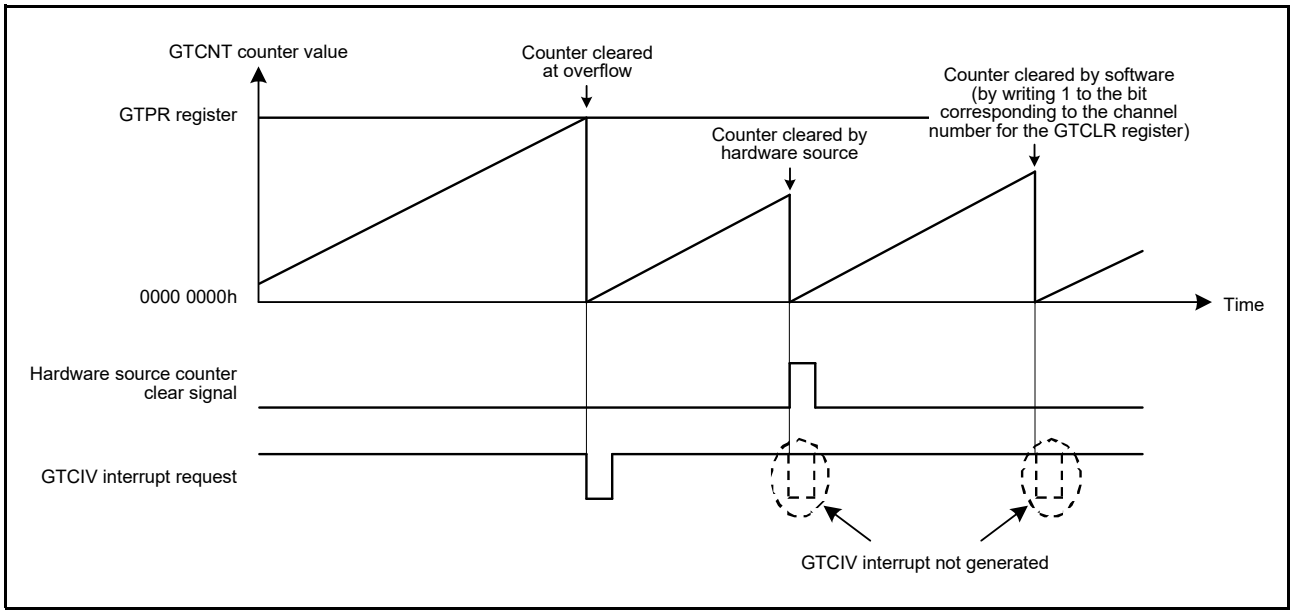


Figure 24.126 Relationship between Counter Clearing by Hardware Source and GTCIV Interrupt

Figure 24.127 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared at the next counting operation after the GPTW detects the internal clearing signal.

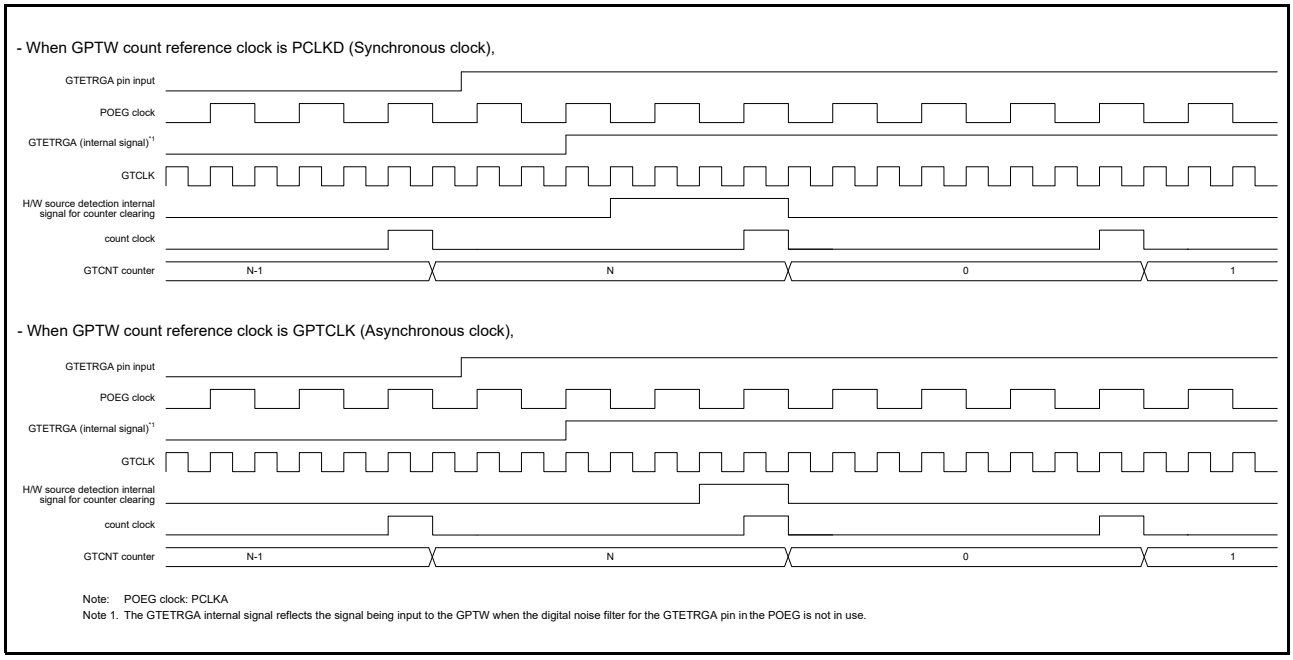


Figure 24.127 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 24.128 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared at the next counting operation after the GPTW detects the internal clearing signal.

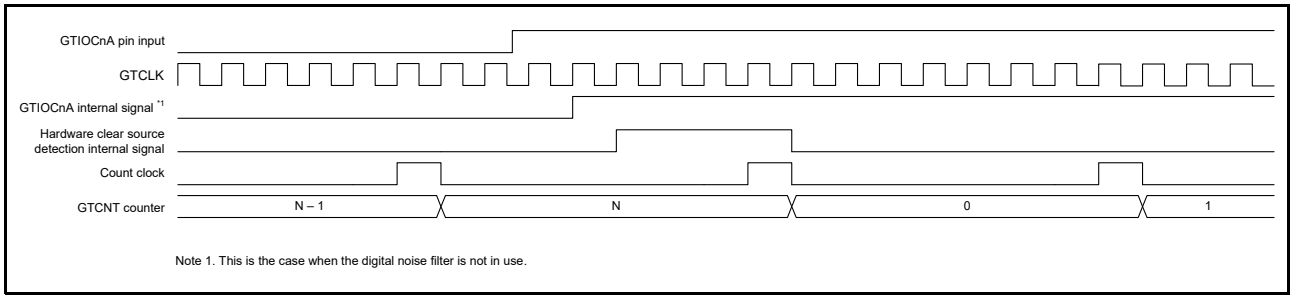


Figure 24.128 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 24.129 shows an example of the timing of operations to clear the counter in response to event input from the ELCA when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared at the next counting operation after the GPTW detects the internal clearing signal.

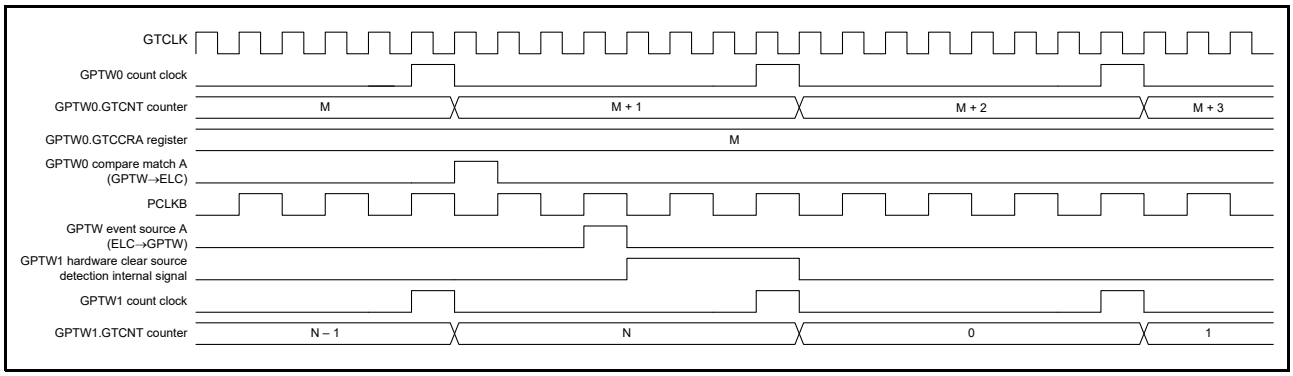


Figure 24.129 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 24.130 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with GTCLK after the GPTW has detected the internal clearing signal.

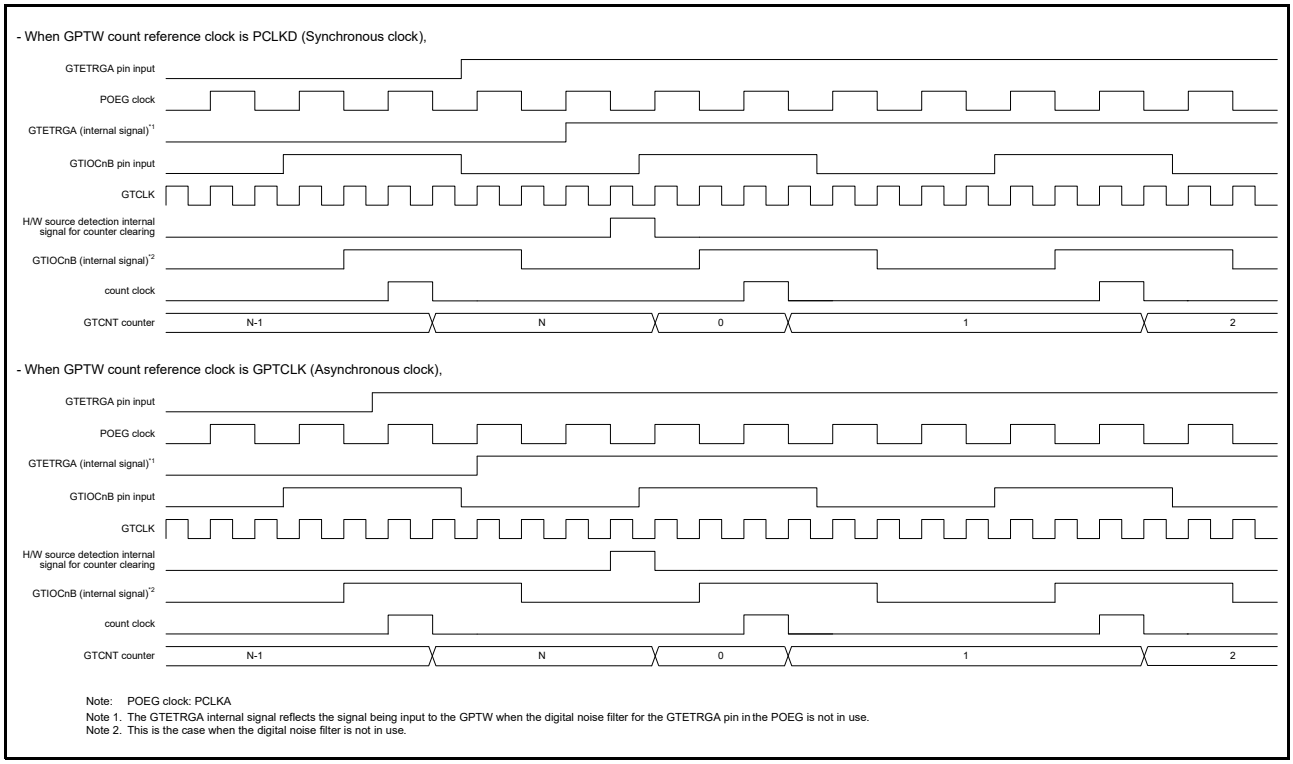


Figure 24.130 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During Counting Triggered by Hardware Source)

Figure 24.131 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with GTCLK after the GPTW has detected the internal clearing signal.

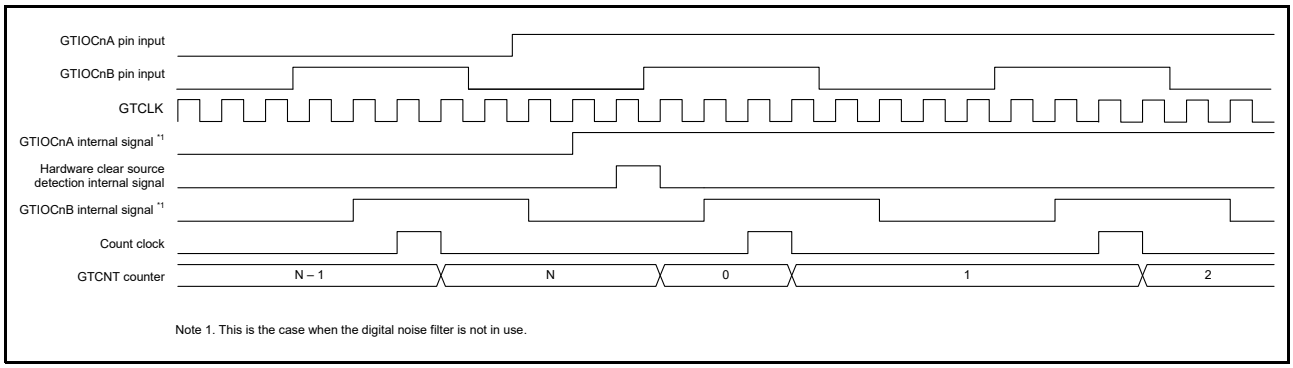


Figure 24.131 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During Counting Triggered by Hardware Source)

Figure 24.132 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELCA when counting is triggered by a hardware source.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared in synchronization with GTCLK after the GPTW has detected the internal clearing signal.

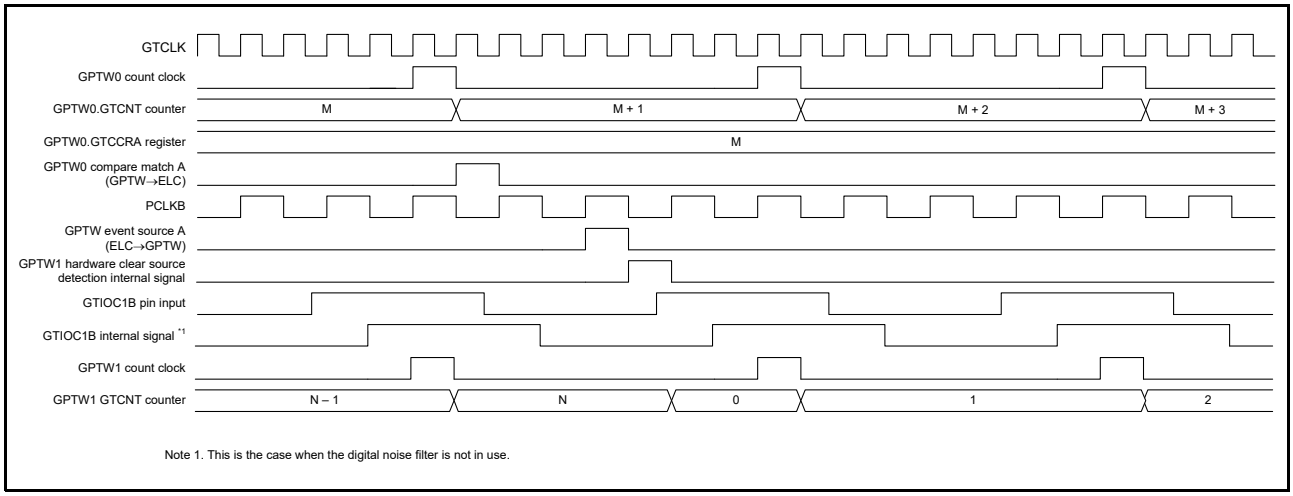


Figure 24.132 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During Counting Triggered by Hardware Source)

24.3.8 Hardware Count Switch Operation

In the triangle-wave PWM mode, the GTCNT counter can be switched by a hardware source. Select the count switch operation with the GTCR.SWMD[2:0] bits and select a hardware source to switch the direction of counting with the GTSWSR register and enable to switch the direction of counting.

When the offset reload is enabled by the GTCR.SWMD[2:0] bits, the offset value of the GTCNT counter reloaded by count switch is set by the GTSWOS register in the range of 0 to GTPR.

If the count switch and counter clear occur at the same time, the counter clear has priority.

Figure 24.133 to Figure 24.136 show examples of the GTCNT count switch operation by a hardware source. Figure 24.137 shows the setting example. In this example, the GTCNT counter is started at the ELCA event input, is stopped/cleared at the ELCB event input, and is switched at the ELCC event input.

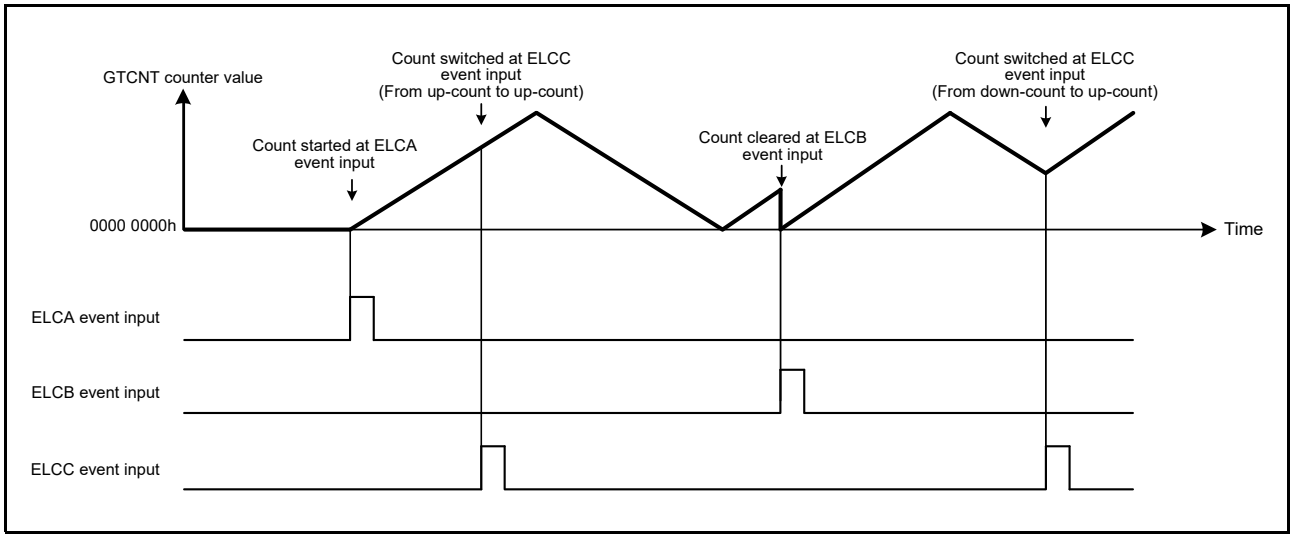


Figure 24.133 Examples of Count Switch Operation by Hardware Source (Switched to Up-count without Reload at ELCC Event Input)

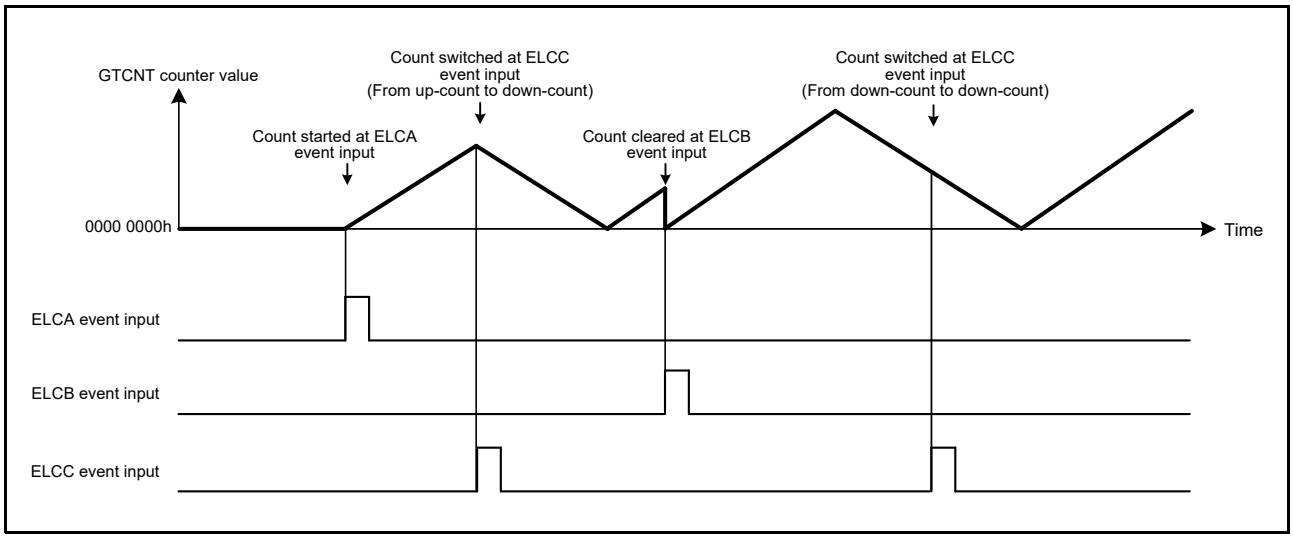


Figure 24.134 Examples of Count Switch Operation by Hardware Source (Switched to Down-count without Reload at ELCC Event Input)

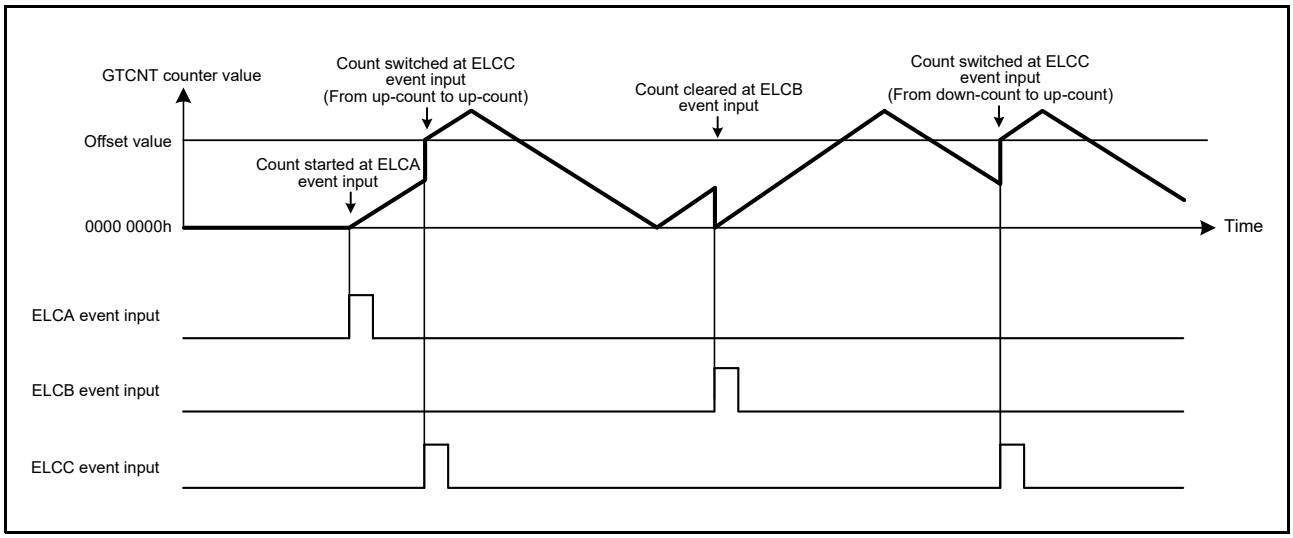


Figure 24.135 Examples of Count Switch Operation by Hardware Source (Switched to Up-count with Reload at ELCC Event Input)

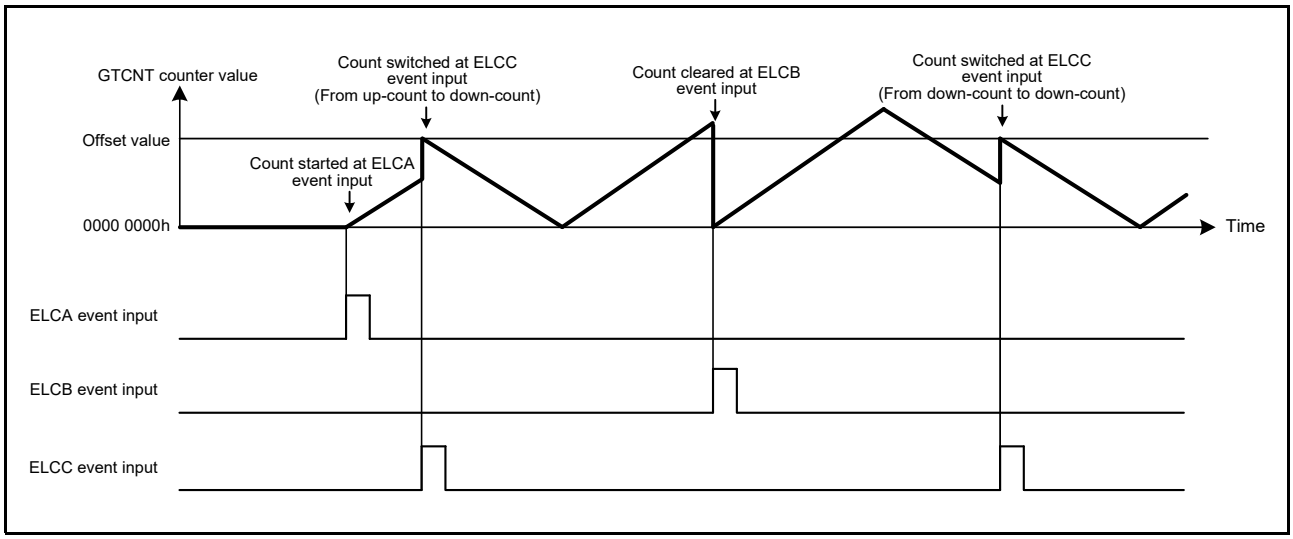


Figure 24.136 Examples of Count Switch Operation by Hardware Source (Switched to Down-count with Reload at ELCC Event Input)

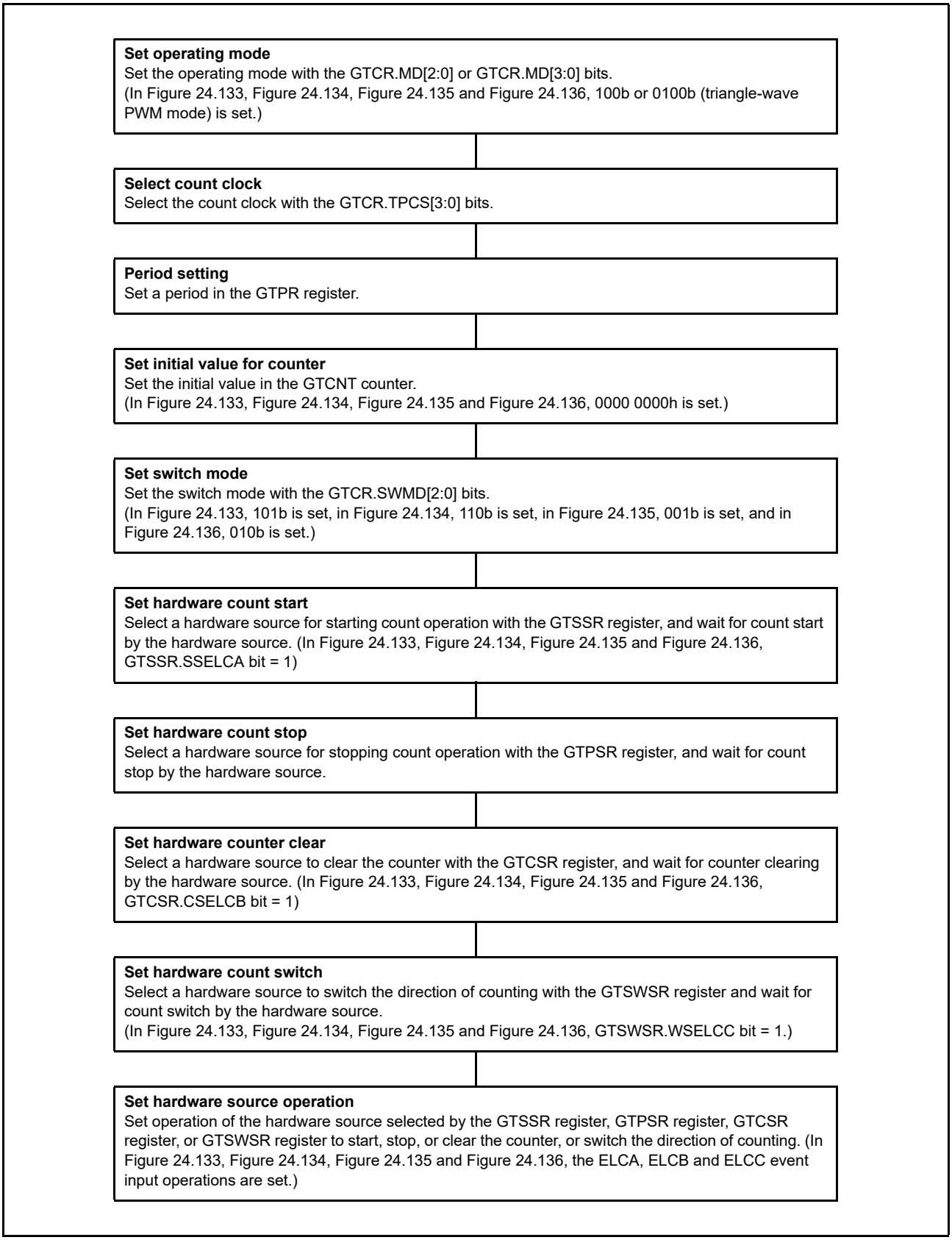


Figure 24.137 Example for Setting Count Switch Operation by Hardware Source

The compare match operation is effective even when the count switch and the compare match occur at the same time. Similarly, even when the count switch occurs at the crest (GTCNT counter matches GTPR register value during up-counting) or trough (GTCNT counter is 0 during down-counting) of triangle-waves, overflow or underflow interrupt is output and the buffer transfer of crest or trough works normally.
If the count switch is required at the crest or trough without the offset reload, the count switch is ignored, and the direction of counting changes to down-count or up-count after the crest or trough. Also, if the count switch is required at the count clear state (GTCNT counter value is 0 during up count) without the offset reload, the count switch is ignored, and the direction of counting continues up-count.
If the switch mode is set to “Switch to up-count with offset reload” with the GTCR.SWMD[2:0] bits and the offset is set to 0 with the GTSWOS register, the count switch works same as the counter clear and the GTCNT counter does not operate as the trough. Also, if the switch mode is set to “Switch to down-count with offset reload” and the offset is set to the GTPR value, the GTCNT counter does not operate as the crest. On the other hand, if the switch mode is set to “Switch to up-count with offset reload” and the offset is set to the GTPR value, the GTCNT counter operate as the crest, if the switch mode is set to “Switch to down-count with offset reload” and the offset is set to 0, the GTCNT counter operate as the trough, and in these cases, the interrupt outputs and the buffer transfers of the crest and the trough works normally. If the crest or the trough due to the count switch occurs at the timing when the GTCNT counter value is the crest or the trough, the buffer transfer and the interrupt output at the crest or trough will occur twice in a row. Table 24.23 shows the triangle-waves condition after the count switch with offset reload.

Table 24.23 The Triangle-Waves Condition After the Count Switch with Offset Reload

Count Direction of Switch Mode (GTCR.SWMD[2:0])	Offset (GTSWOS)	The Triangle-Waves Condition After the Count Switch			
		GTCNT	GTST.TUCF	Triangle-Waves Condition	Buffer Transfer/ Interrupt Output
Up-count	0	GTSWOS register value	1	Counter Clear (Start of Up-count)	Nothing
	1 to GTPR−1			Up-count	Nothing
	GTPR			Crest (End of Up-count)	Buffer Transfer available (Only Triangle-Wave PWM Mode 2)/ GTCIVn Output
Down-count	0		0	Trough (End of Down-count)	Buffer Transfer available/ GTCIUn Output
	1 to GTPR−1			Down-count	Nothing
	GTPR			Start of Down-count	Nothing

Figure 24.138 and Figure 24.139 show examples of the buffer operation due to the count switch when the offset is set to 0 and GTPR. Figure 24.140 shows an example of the buffer operation due to the count switch when the offset is set to values other than the crest and trough.

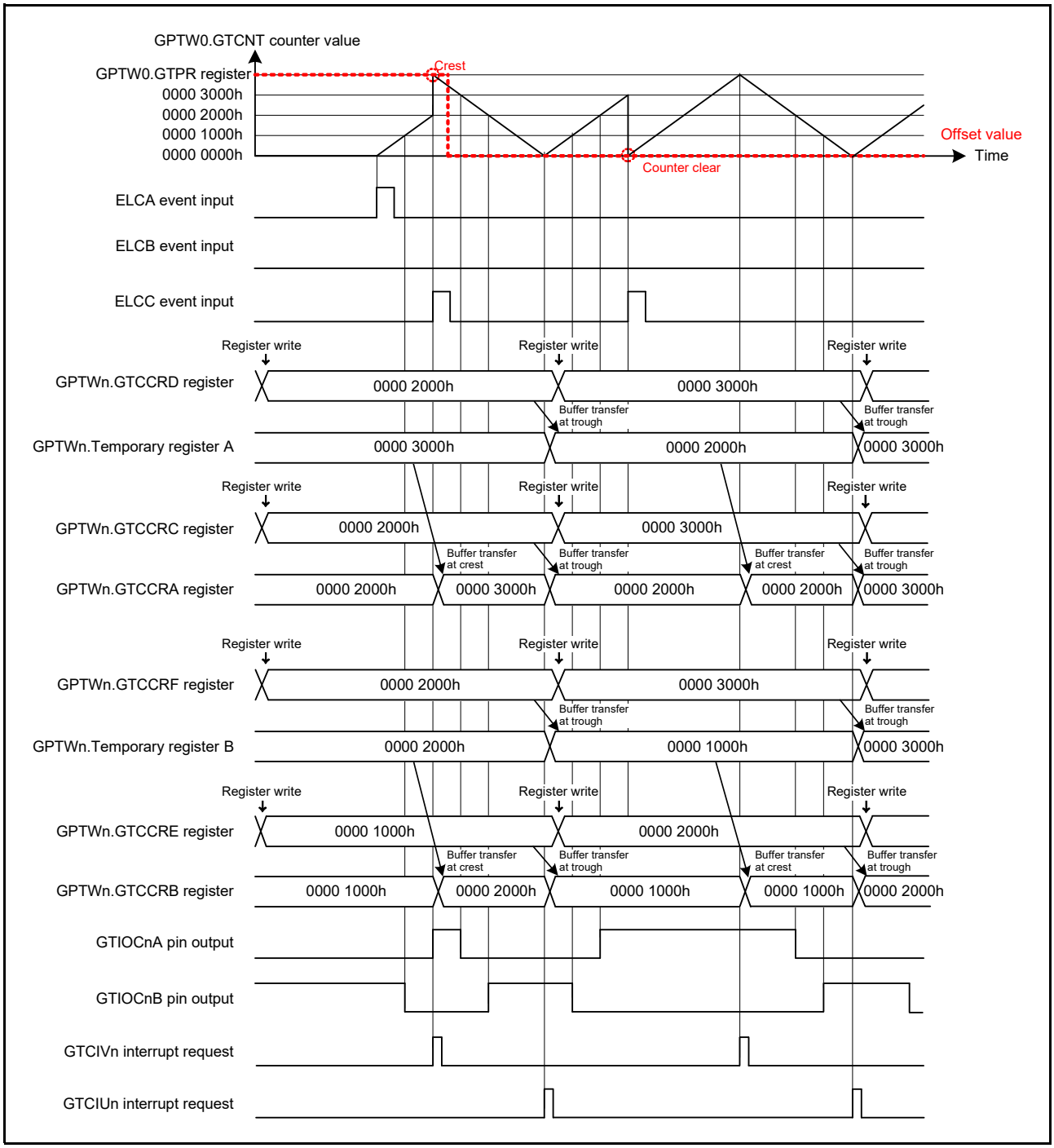


Figure 24.138 Example of the Buffer Operation Due to the Count Switch (Triangle-Wave PWM Mode 3, Offset is Set to 0 and GTPR and Switch to Up-count) (n = 0 to 15)

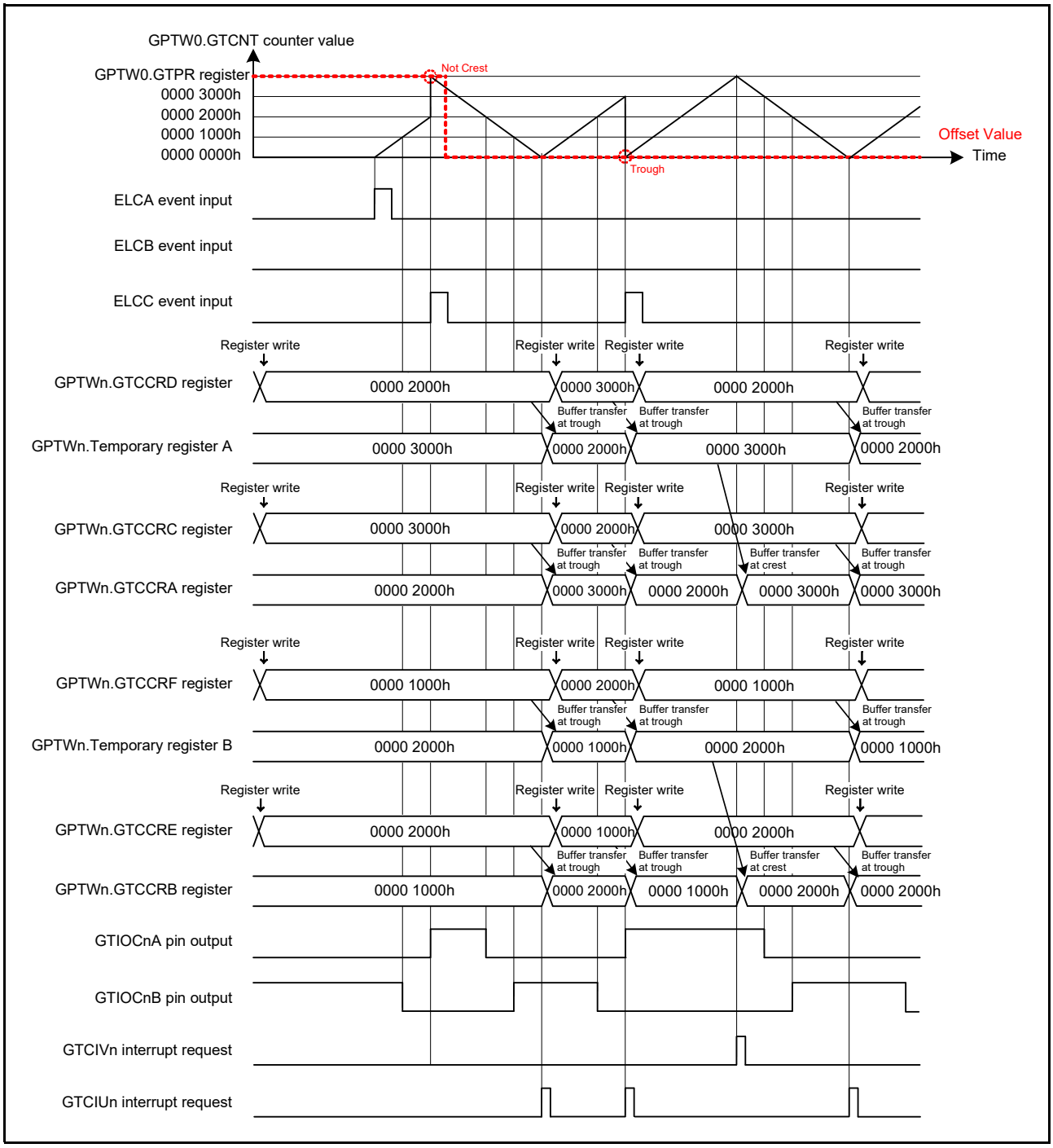


Figure 24.139 Example of the Buffer Operation Due to the Count Switch (Triangle-Wave PWM Mode 3, Offset is Set to 0 and GTPR and Switch to Down-count) (n = 0 to 15)

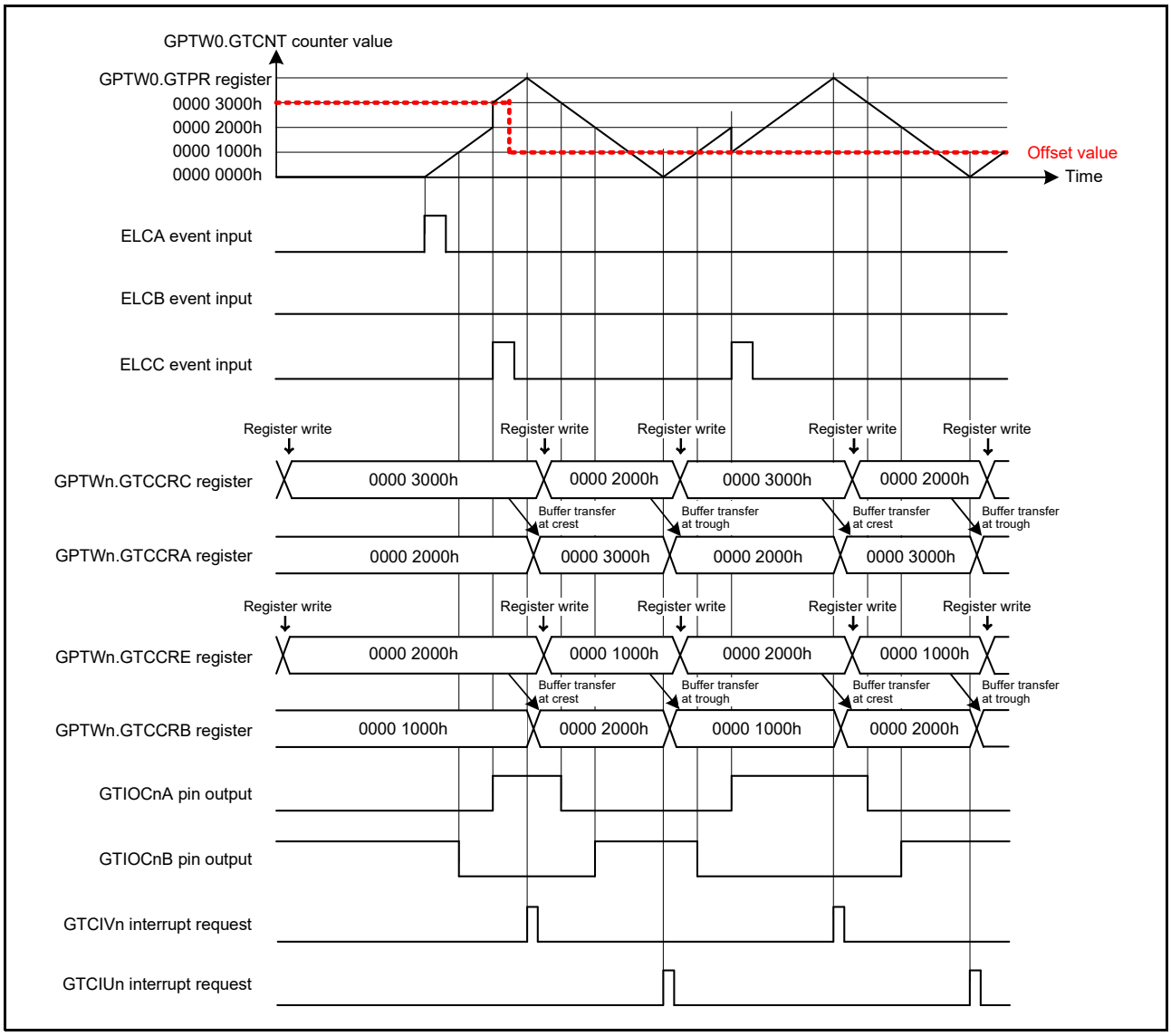


Figure 24.140 Example of the Buffer Operation Due to the Count Switch (Triangle-Wave PWM Mode 2, No Crest and Trough Occurrence Due to the Count Switch) (n = 0 to 15)

When the counter value skips the compare match value by the count switch, no compare match occurs and the GTIOCnm pin output does not change. Figure 24.141 shows an example of the operation when the compare match is skipped by switch operation.

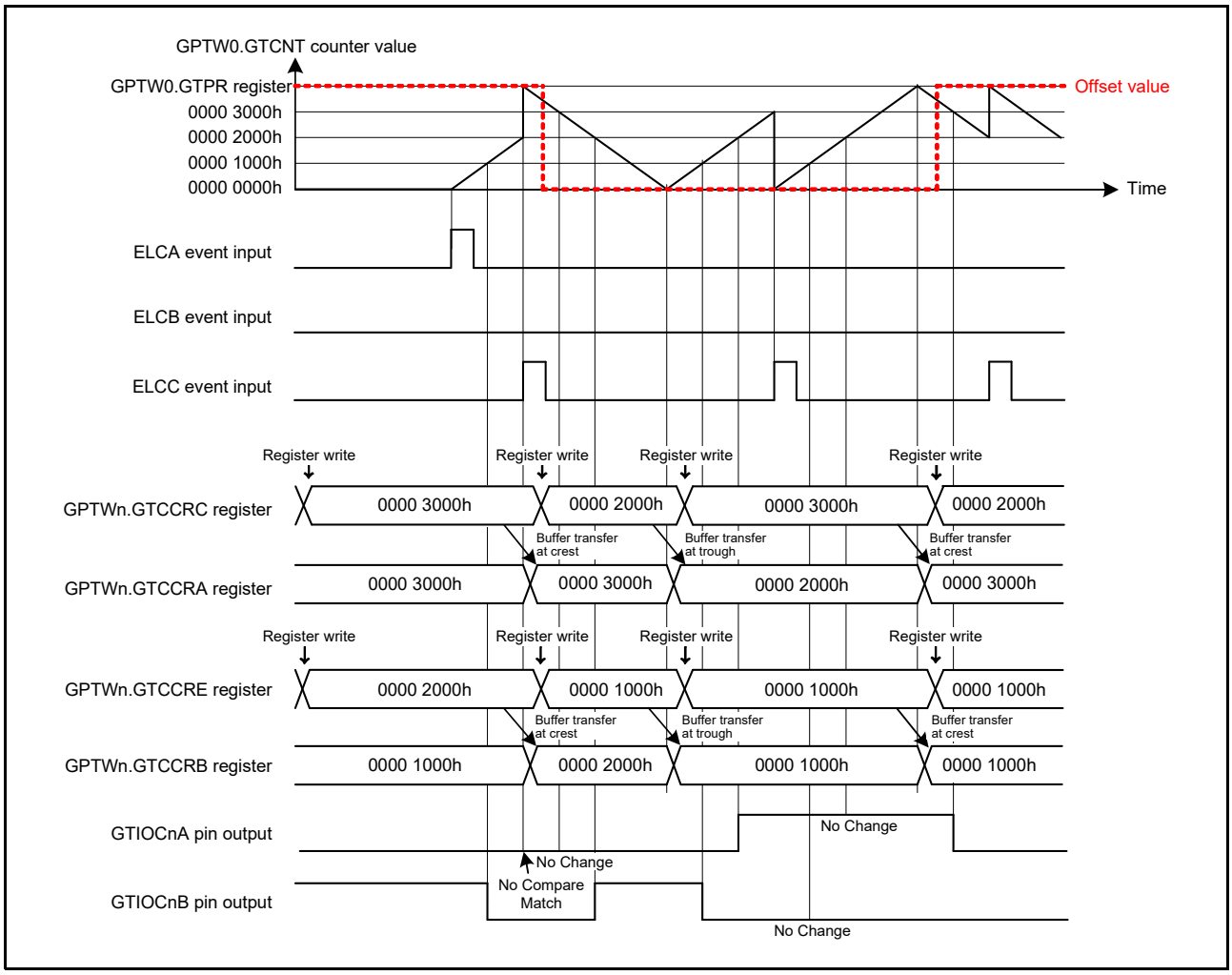


Figure 24.141 Example of the Exception Operation Due to the Count Switch (Switch to Up-count with Offset Reload by ELCC event) (n = 0 to 15)

Figure 24.142 and Figure 24.143 show examples of the timing to switch the direction of counting in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the GTCLK signal is in use as the counter clock for the GTCNT counter.
The GTCNT counter is switched at the next counting operation after the GPTW detects the internal switching signal.

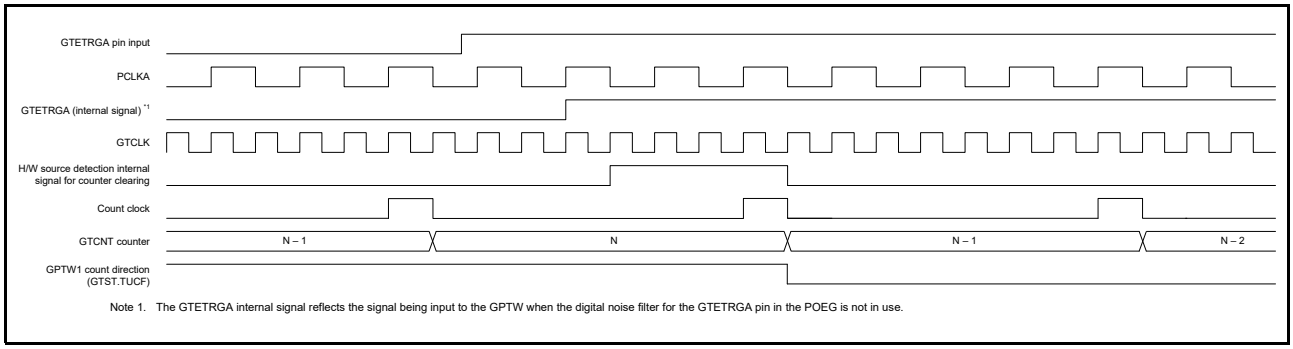


Figure 24.142 Example of the Timing to Switch the Direction of Counting in Response to a Rising Edge of the Input on the GTETRGA Pin: Switch to Counting Down without Reloading During the Counting of Cycles of the Clock Signal Produced by Dividing the GTCLK Frequency

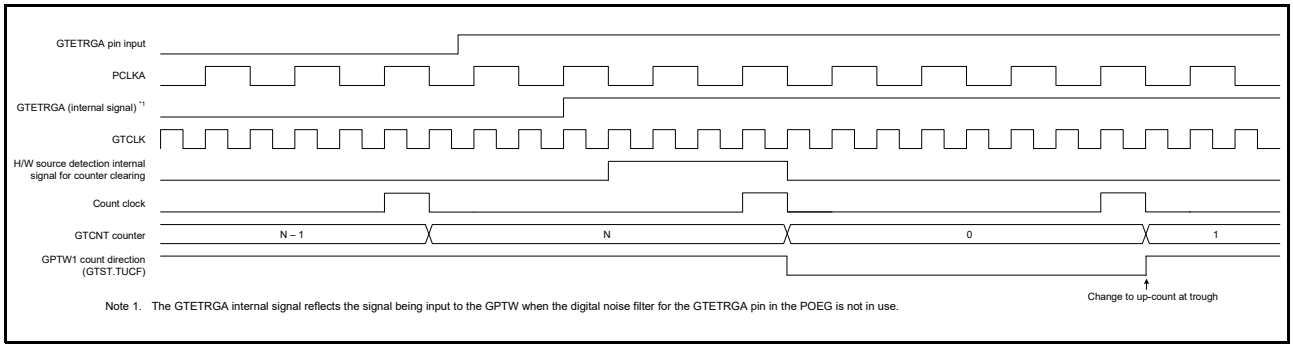


Figure 24.143 Example of the Timing to Switch the Direction of Counting in Response to a Rising Edge of the Input on the GTETRGA Pin: Switch to Counting Down with Reloading During the Counting of Cycles of the Clock Signal Produced by Dividing the GTCLK Frequency and Offset Value is 0

Figure 24.144 and Figure 24.145 show examples of the timing to switch the direction of counting in response to ELCA event input when a clock signal produced by frequency-dividing the GTCLK signal is in use as the counter clock for the GTCNT counter, and the frequency of GTCLK is double that of PCLKB.
This is an example of operations to switch the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is switched in synchronization with GTCLK after the GPTW has detected the internal switching signal.

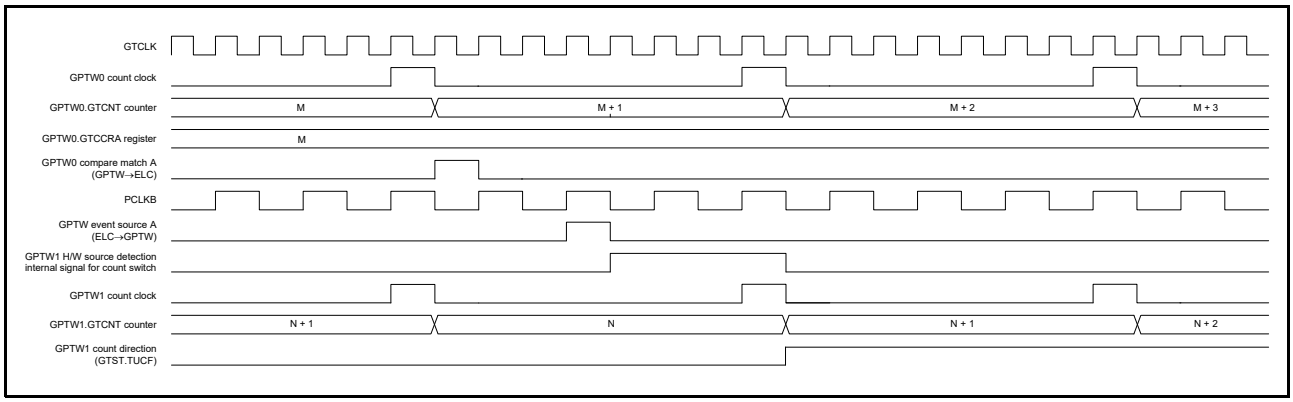


Figure 24.144 Example of the Timing of Count Switch Operations in Response to Event Input from the ELCA: Switch to Counting Up without Reloading During the Counting of Cycles of the Clock Signal Produced by Dividing the GTCLK Frequency

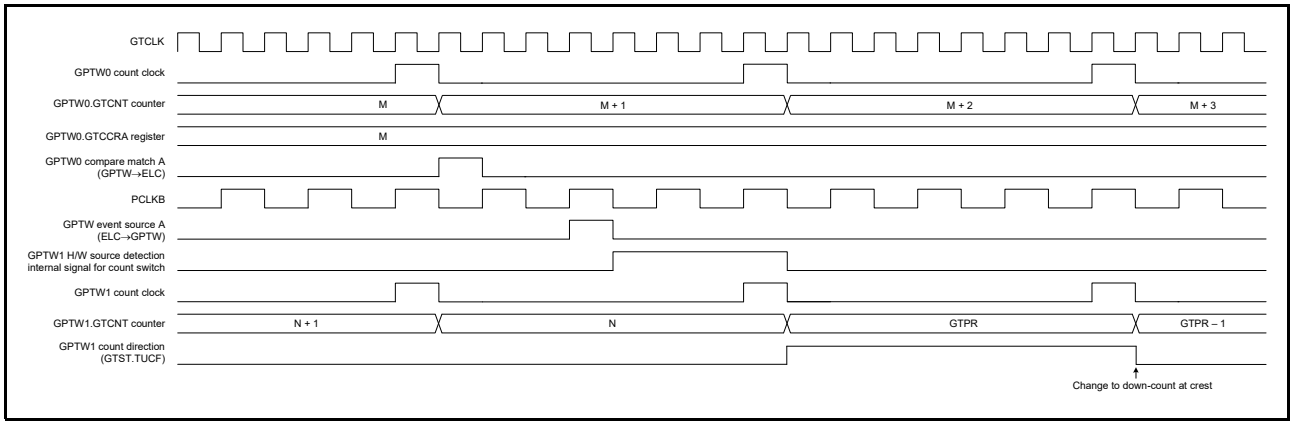


Figure 24.145 Example of the Timing of Count Switch Operations in Response to Event Input from the ELCA: Switch to Counting Up with Reloading During the Counting of Cycles of the Clock Signal Produced by Dividing the GTCLK Frequency and Offset value is GTPR

24.3.9 Inter-Channel Synchronous Operation

Synchronous operation on channels (synchronous start/stop/clear/switch operation) can be performed.

24.3.9.1 Software Control of Inter-Channel Synchronous Operation

The count operations for respective channels can be started, stopped and cleared simultaneously by setting multiple bits of the GTSTR, GTSTP, and GTCLR registers to 1 at the same time.
Count start with phase differences among channels is possible firstly by setting the GTCNT counter value before counting starts, and then, by setting simultaneously multiple bits in the GTSTR register to 1.
The GTCNT counter value settings are synchronized by setting the GTCR.SSCEN bit to 1, and can be written simultaneously to multiple channels set in the same group by the GTCR.SSCGRP[1:0] bits. Synchronous sets are not valid for channels that are set to complementary PWM mode.
When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels for which 1 is set in the GTSECSR register are set to 0 or 1 accordingly, and the GTCNT synchronous set/clear is enabled or disabled on multiple channels at the same time.
As the count operation clock is selected by the GTCR.TPCS[3:0] bits in each channel, if the clock period of channels performing synchronous operation (count start/stop/clear) differ, the synchronous operation timings will not be exactly the same.
The Figure 24.146 shows an example of simultaneous start/stop/clearing of four channels by software, and Figure 24.147 shows an example of phase shift start among four channels by software. Figure 24.148 to Figure 24.150 show examples of simultaneous start/stop/clearing with different count period.

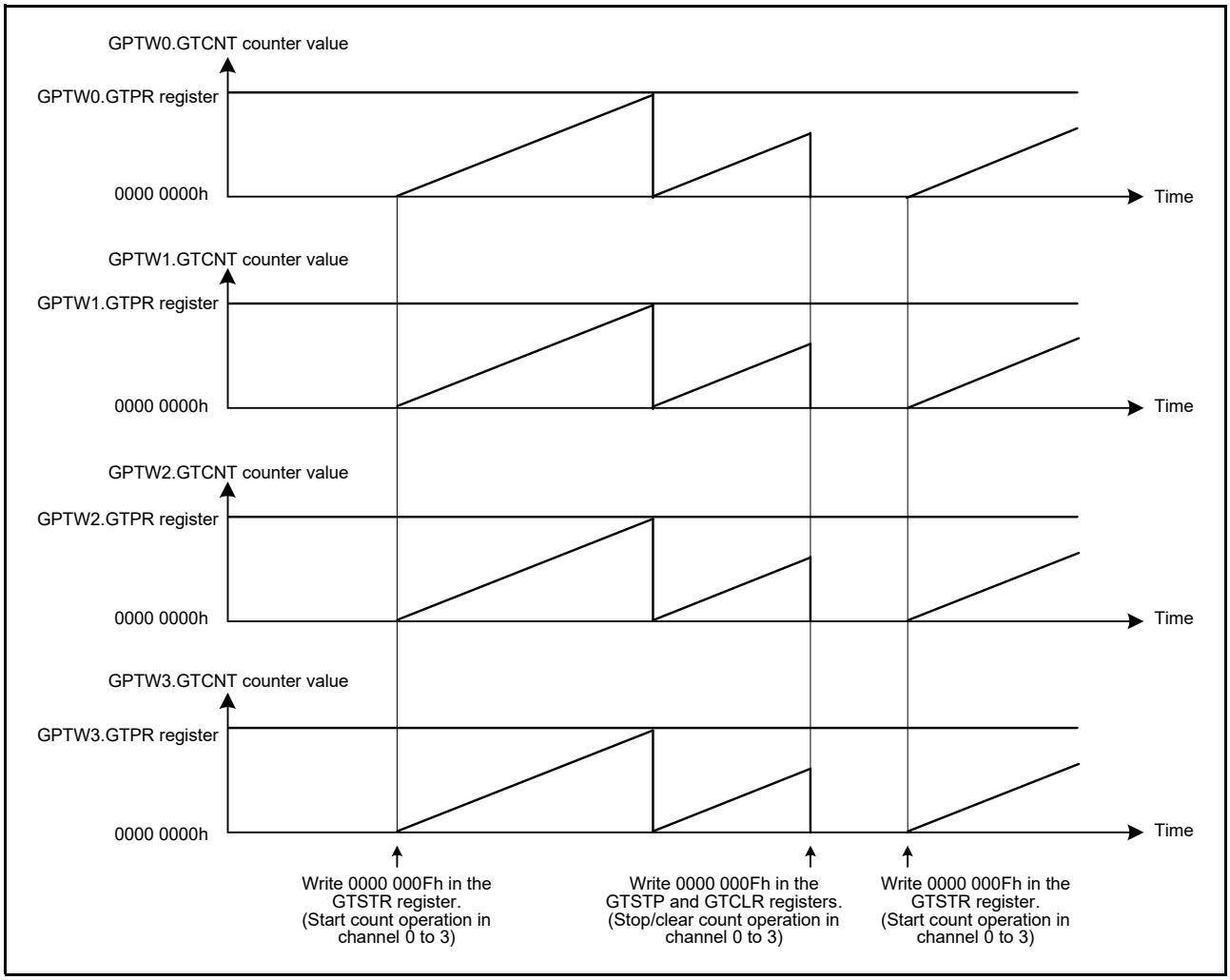


Figure 24.146 Example of Simultaneous Start/Stop/Clearing Operation by Software (with the Same Count Period (GTPR Register Value))

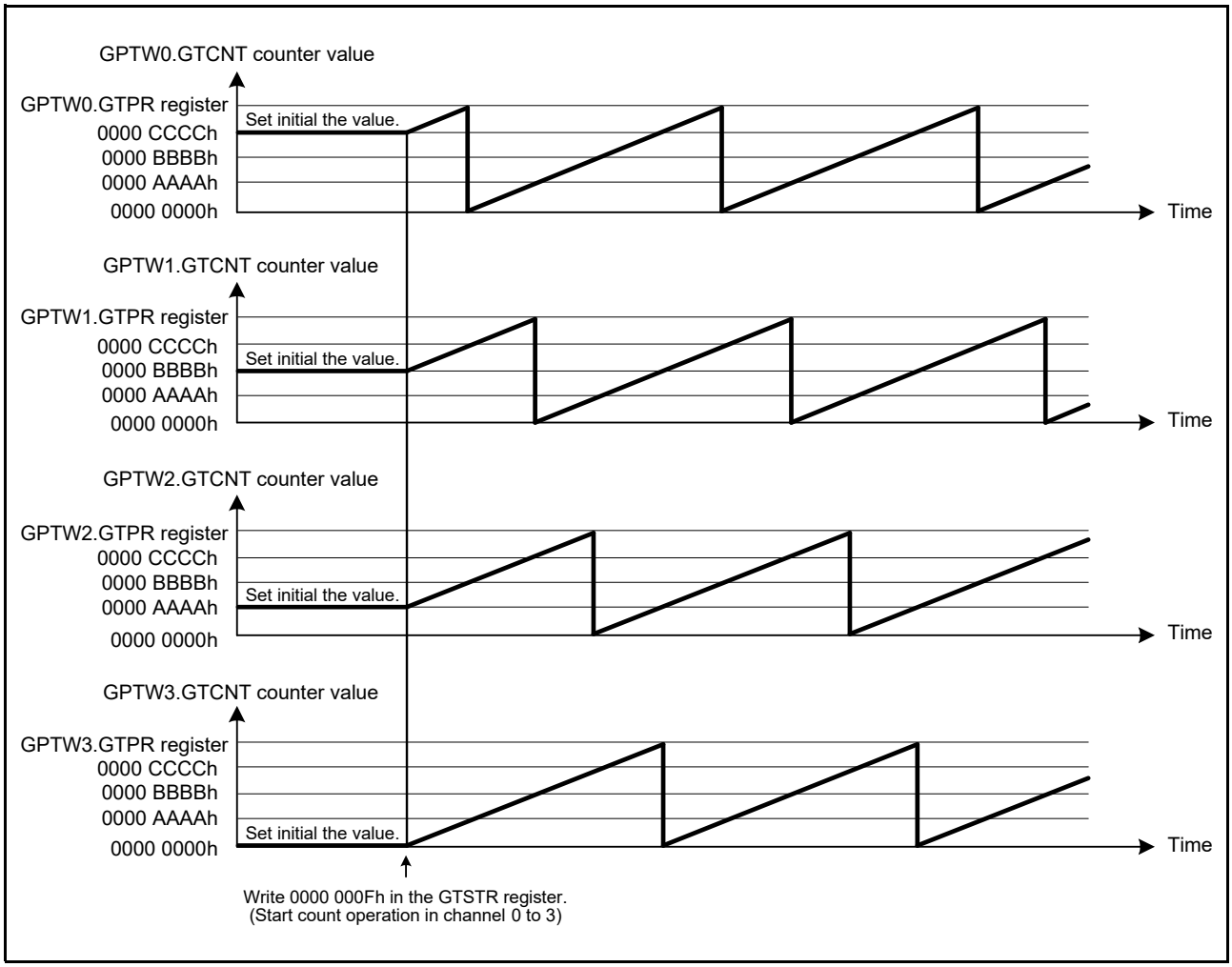


Figure 24.147 Example of Software Phase Shift Start (with Same Count Period (GTPR Register Value))

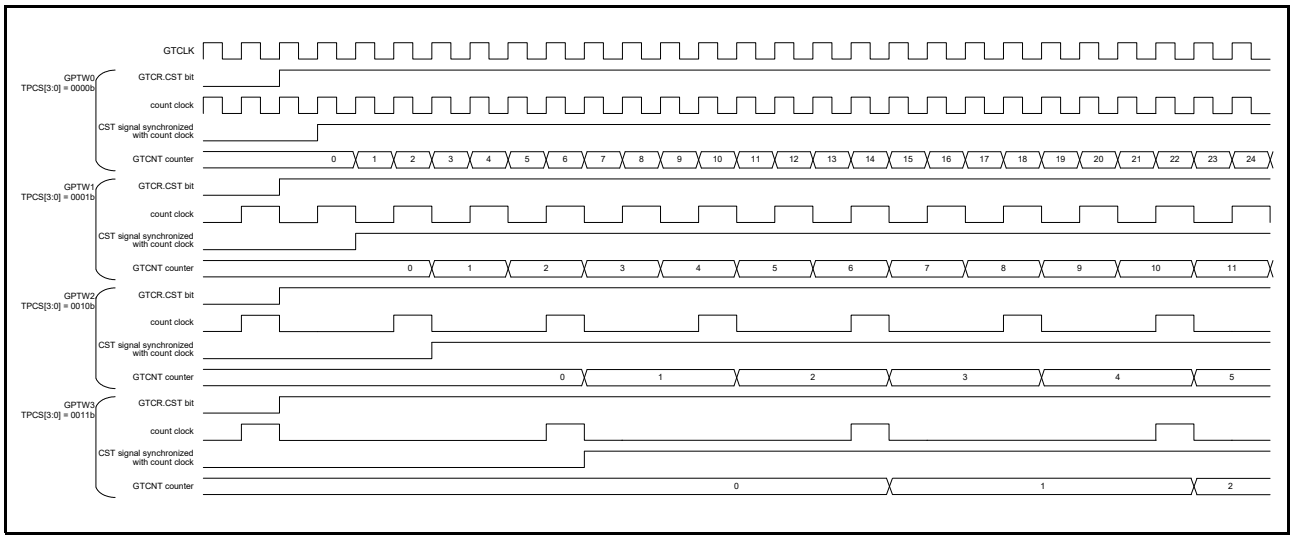


Figure 24.148 Example of Simultaneous Start Operation by Software (with Different Count Period)

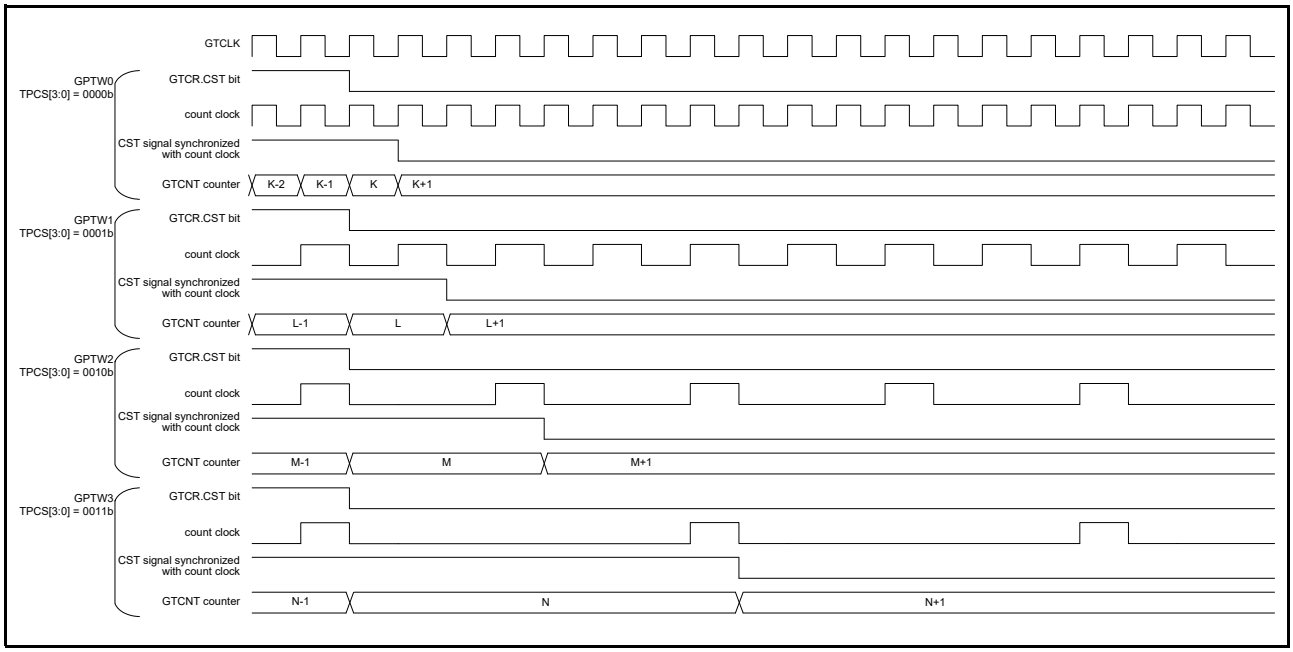


Figure 24.149 Example of Simultaneous Stop Operation by Software (with Different Count Period)

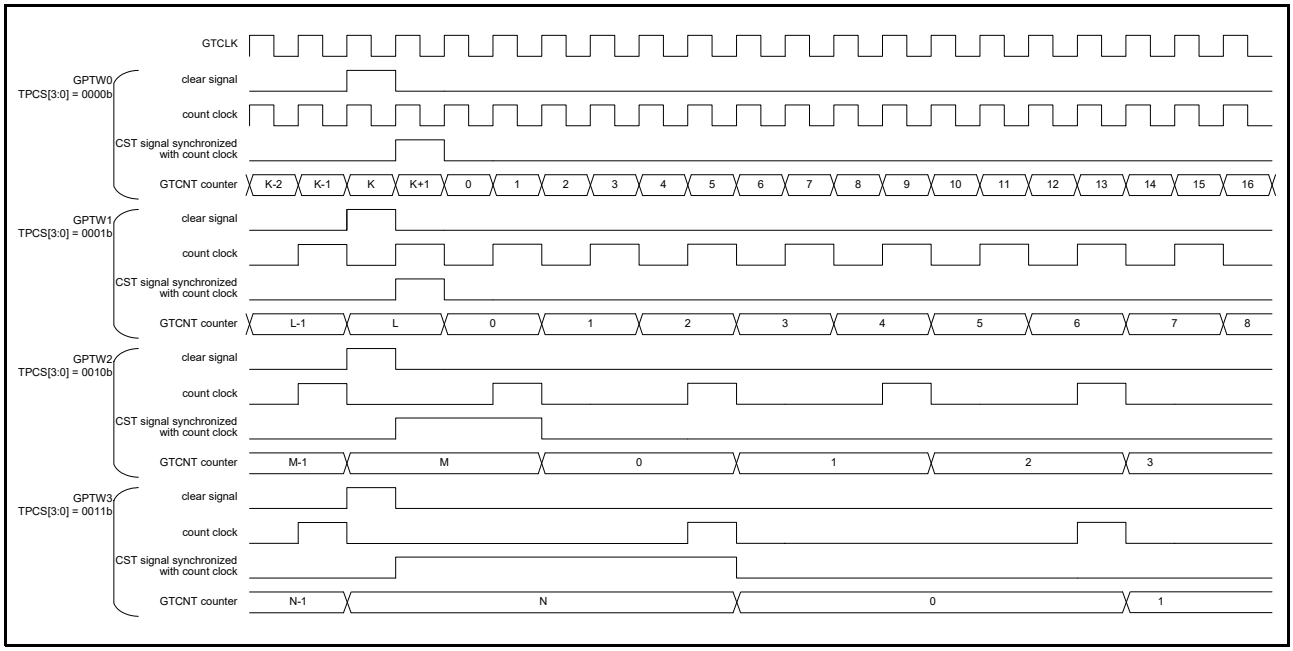


Figure 24.150 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

24.3.9.2 Hardware Source Control of Inter-Channel Synchronous Operation

The count operations for respective channels can be started/stopped/cleared/switched simultaneously by following hardware sources: external trigger input, and ELC event input.

Synchronous operation by the GTIOCnA and GTIOCnB pin inputs (n = 0 to 15) can be performed by setting the ELC event by input capture as a hardware source.

Setting the GTCNT counter values for the respective channels before counting starts makes it possible to start counting with phase differences among the channels.

Figure 24.151 shows an example of simultaneous start/stop/clearing operation of four channels by hardware source, and Figure 24.152 shows an example for setting simultaneous start by hardware source.

Figure 24.153 shows an example of simultaneous start/stop/clearing/switching operation of five channels by hardware source, and Figure 24.154 shows an example for setting simultaneous start by hardware source.

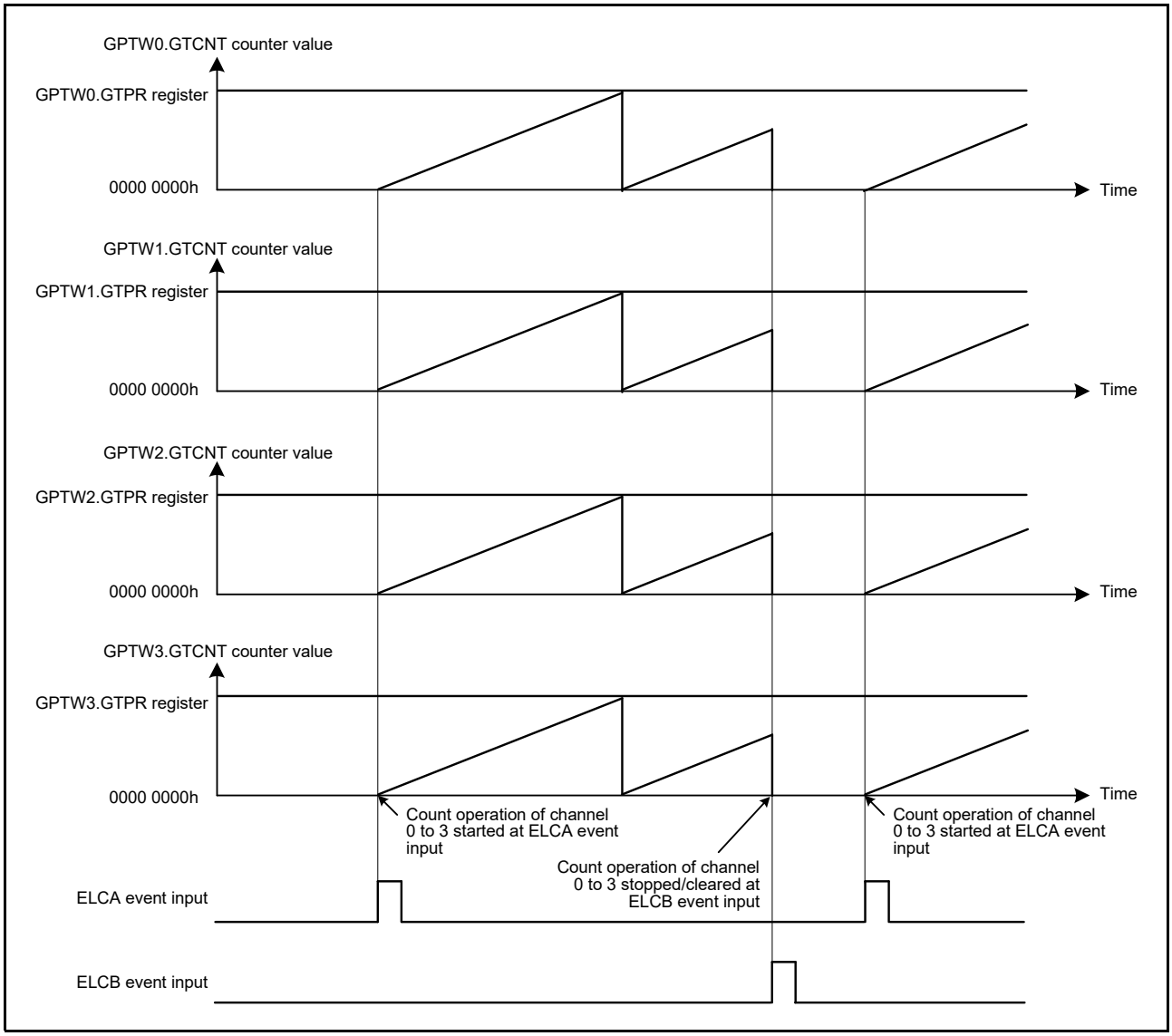


Figure 24.151 Example of Simultaneous Start/Stop/Clearing Operation by Hardware Source (with Same Count Period (GTPR Register Value))

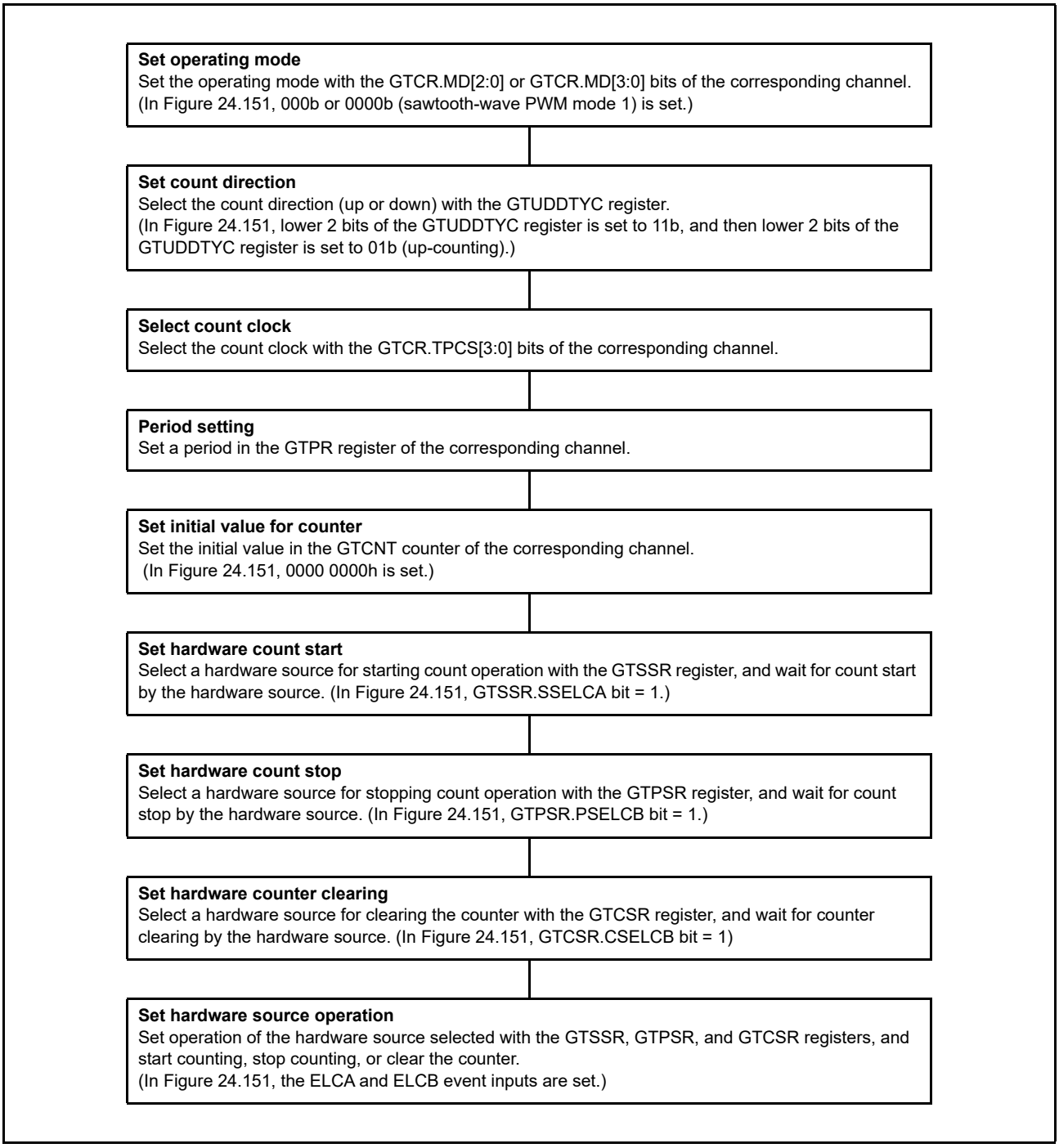


Figure 24.152 Example for Setting Simultaneous Start by Hardware Source

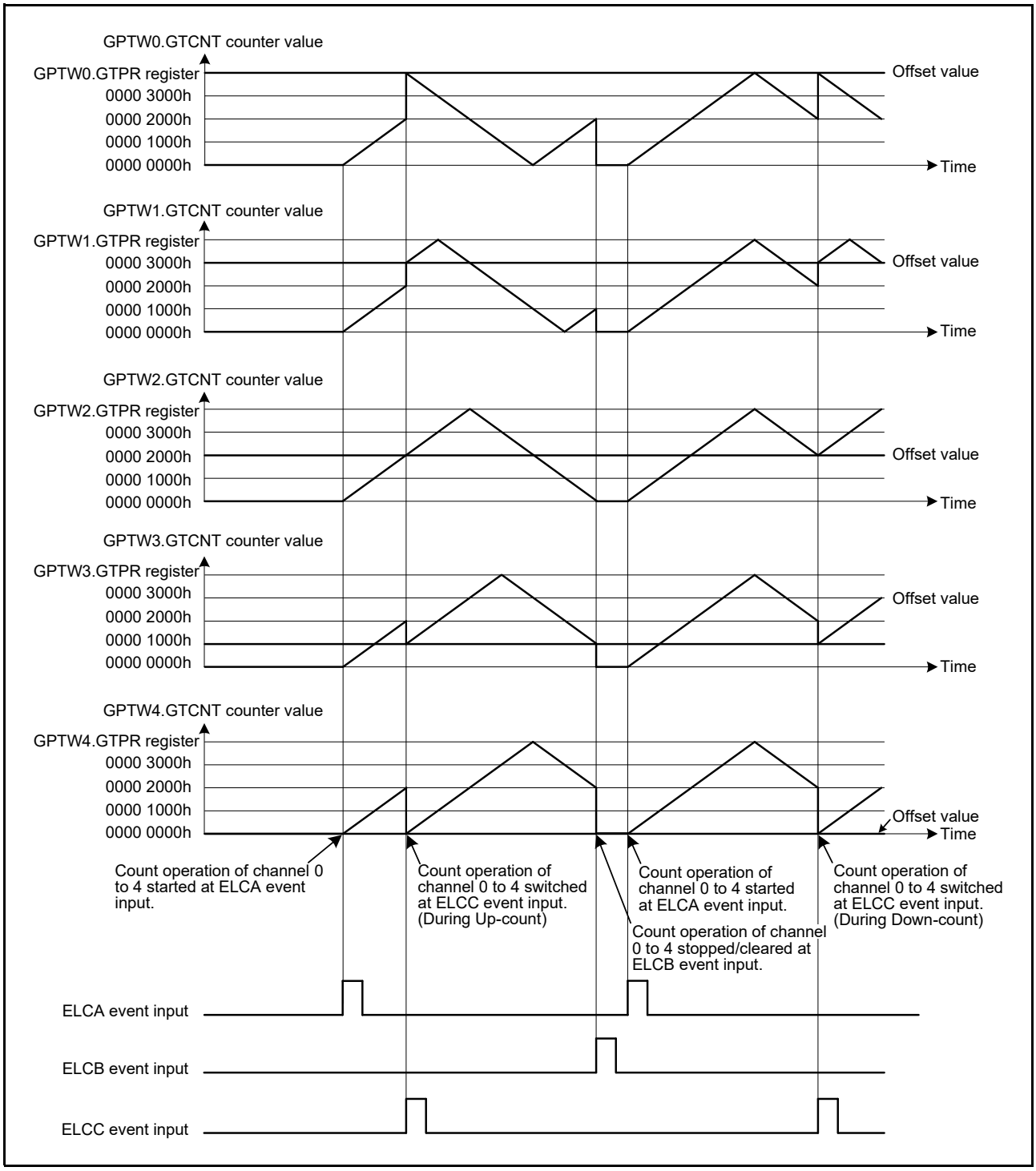


Figure 24.153 Example of Simultaneous Start/Stop/Clearing/Switch Operation by Hardware Source (Switch to Up-count with Offset Reload by ELCC Event, with Same Count Period (GTPR Register Value))

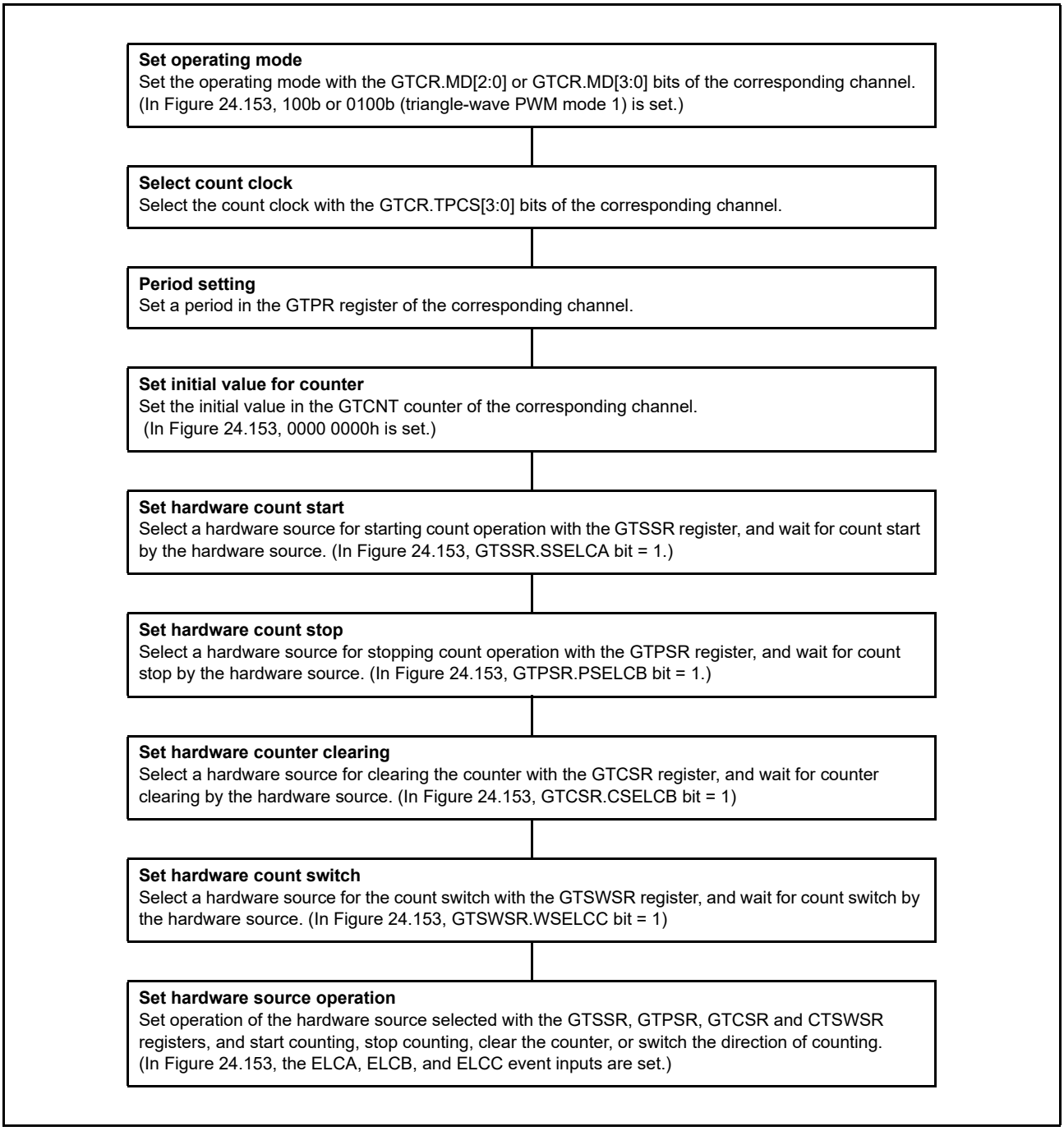


Figure 24.154 Example for Setting Simultaneous Start by Hardware Source (with Count Switch)

24.3.9.3 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, and the GTIOCnA/GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronous clear source and the channel that is synchronously cleared in the same synchronous set/clear group using the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 24.3.9.1, Software Control of Inter-Channel Synchronous Operation, if the count clock selected by the GTCR.TPCS[3:0] bits differs between channels, the synchronous clear operation cannot be performed at exactly the same timing. Similar to the example of simultaneous clearing operation by software in Figure 24.150, if the count clock differs between channels, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels for which 1 is set in the GTSECSR register are set to 0 or 1 accordingly, and the GTCNT synchronous set/clear is enabled or disabled on multiple channels at the same time.

Table 24.24 lists the settings of the registers used to set sources for synchronous clearing. Figure 24.155 shows an example of synchronous clear operation by inter channel cooperation and Figure 24.156 shows an example for setting synchronous clear operation by inter channel cooperation.

Table 24.24 Settings of the Registers Used to Set Sources for Synchronous Clearing

Sources for Synchronous Clearing by Inter Channel Cooperation	Register.Bit(s)	Value
GTCCRA register compare match/input capture	GTINTAD.SCFA	1b
	GTCSR.CSCMSC[2:0]	001b
GTCCRB register compare match/input capture	GTINTAD.SCFB	1b
	GTCSR.CSCMSC[2:0]	010b
GTCCRC register compare match	GTINTAD.SCFC	1b
	GTCSR.CSCMSC[2:0]	011b
GTCCRD register compare match	GTINTAD.SCFD	1b
	GTCSR.CSCMSC[2:0]	100b
GTCCRE register compare match	GTINTAD.SCFE	1b
	GTCSR.CSCMSC[2:0]	101b
GTCCRF register compare match	GTINTAD.SCFF	1b
	GTCSR.CSCMSC[2:0]	110b
Sawtooth wave up-count overflow	GTINTAD.SCFPO	1b
Sawtooth wave down-count underflow	GTINTAD.SCFPU	1b
Clearing due to input on the GTIOCnA or GTIOCnB pin	GTCR.SCGTIOC	1b

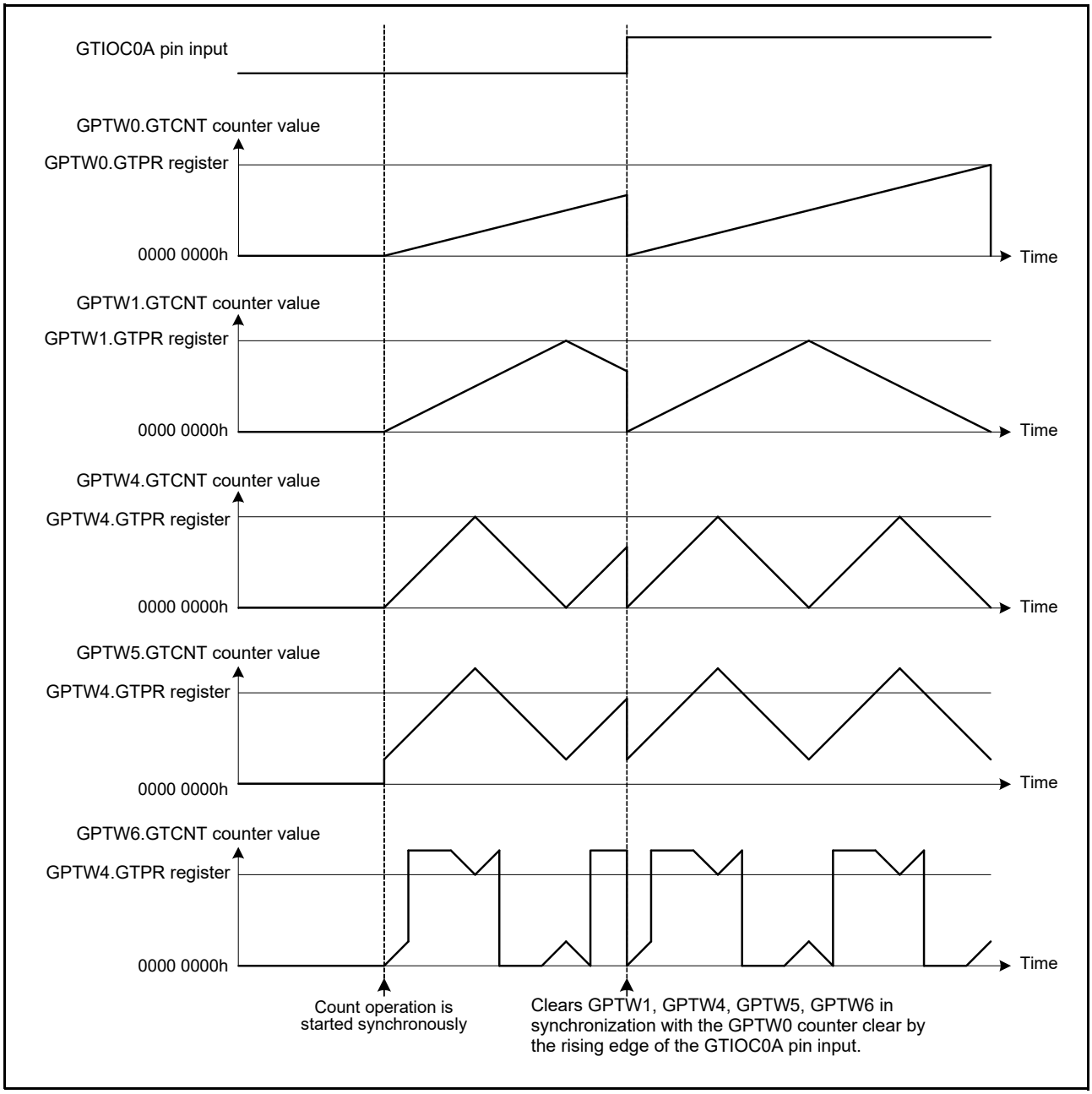


Figure 24.155 Example of Synchronous Clear Operation by Inter Channel Cooperation (Clear Caused by GTIOCnA/GTIOCnB Pin)
(GPTW0 is sawtooth wave and counter is cleared by the rising edge of the GTIOC0A,
GPTW1 is triangle wave,
GPTW4, GPTW5, GPTW6 are complementary PWM mode.
GPTW0, GPTW1, GPTW4, GPTW5, GPTW6 are the same synchronous set/clear group)

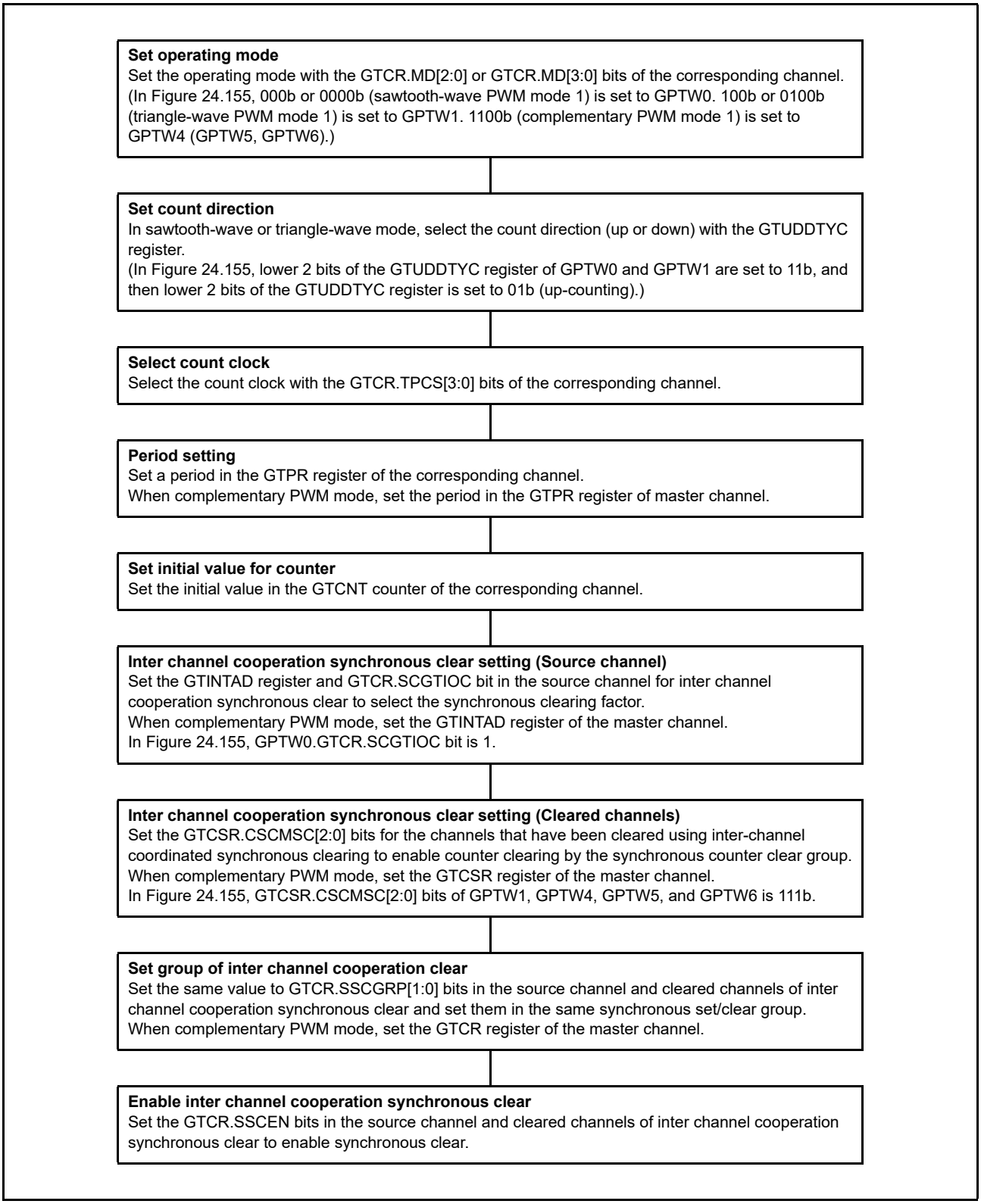


Figure 24.156 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (Clear Caused by GTIOCnA/GTIOCnB Pin)

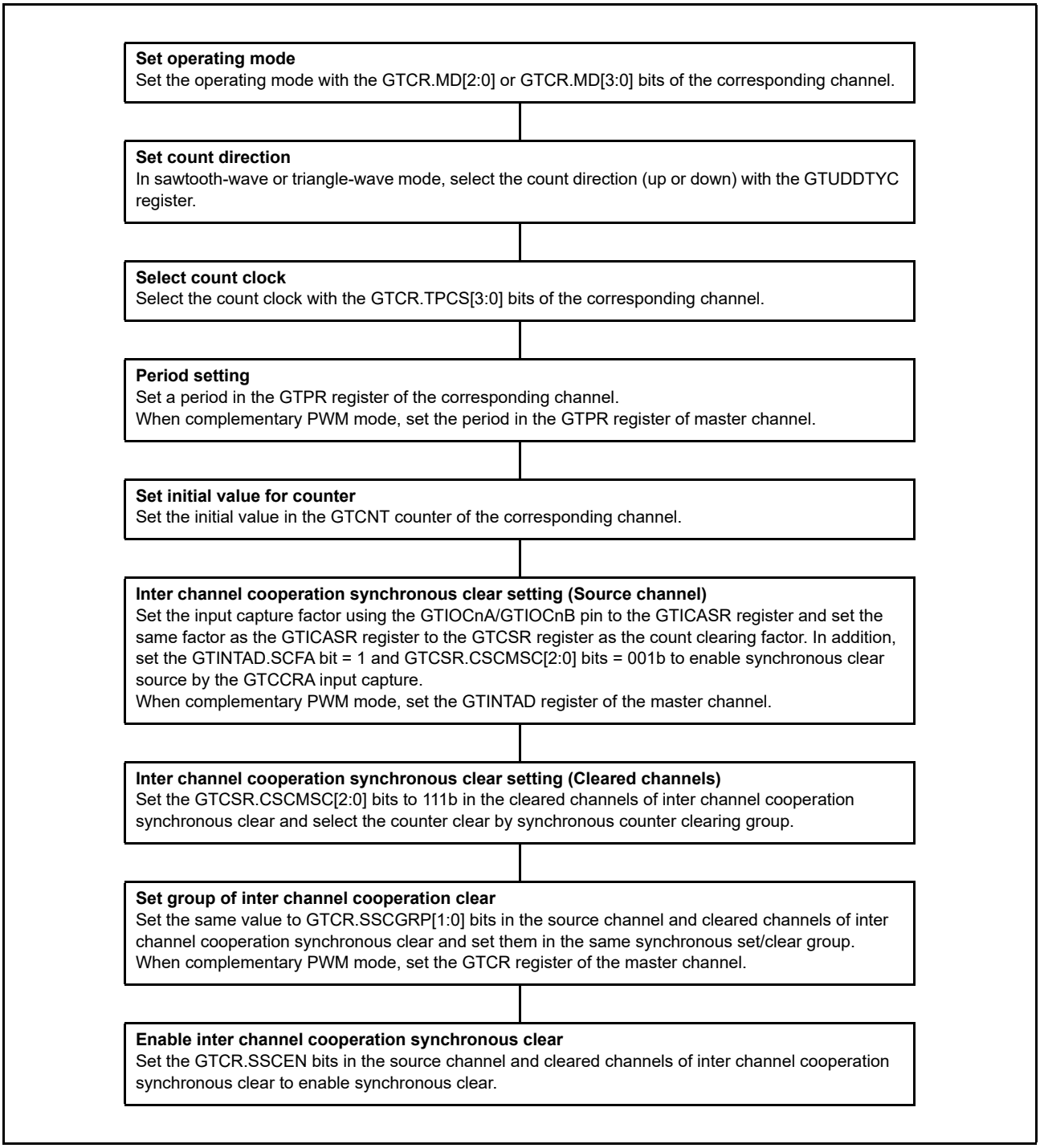


Figure 24.157 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (Clear Caused by the GTCCRA input capture using the GTIOCnA/GTIOCnB Pin)

24.3.9.4 Input Capture Operation by Inter Channel Cooperation

The events of compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, the crest and trough of triangle wave and complementary PWM mode and the count clock can be used as the input capture factor for the GTCCRm register (m = A or B) of other channels.

The input capture factor by inter channel cooperation can be set by the GTICCR register of the channel that generates the input capture factor. The input capture by the factors of other channels is enabled by the GTICmSR.mSOC bit (m = A or B) in the channels to be captured. Set the channels that generate input capture factors and the channels to be captured to the same input capture group using the GTICCR.ICmGRP[1:0] bits (m = A or B).

Figure 24.158 shows an example of input capture operation by inter channel cooperation and Figure 24.159 shows an example for setting input capture operation by inter channel cooperation.

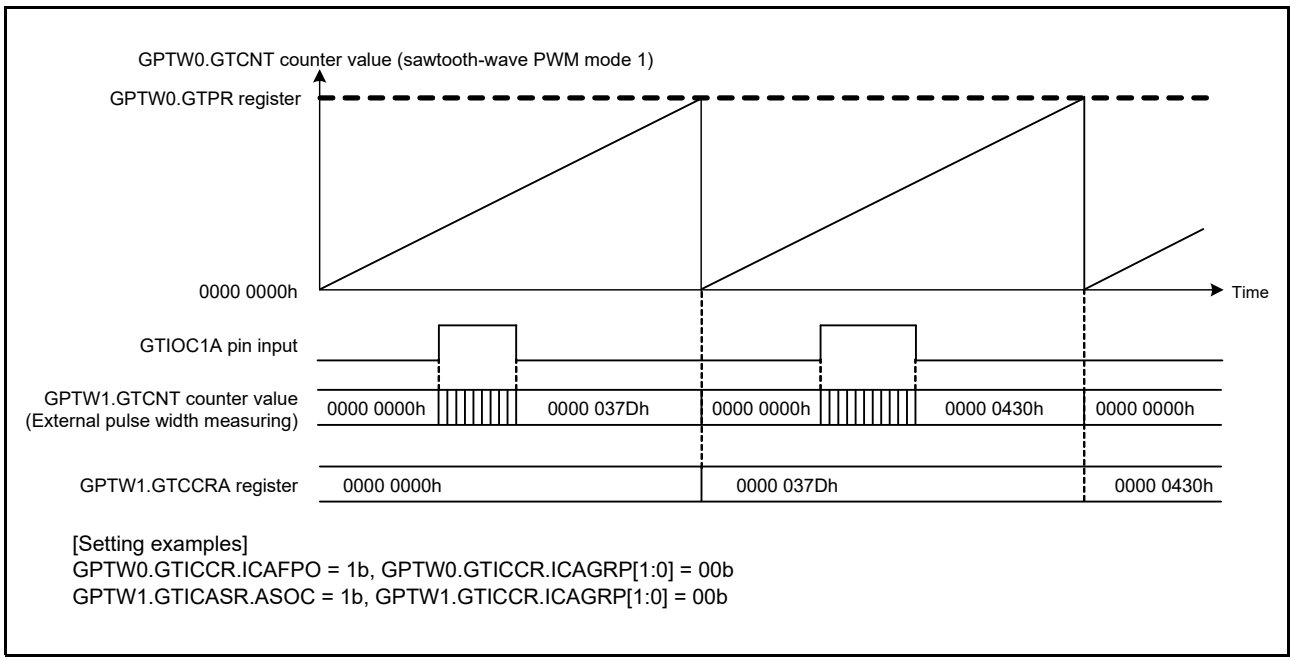


Figure 24.158 Example of Input Capture Operation by Inter Channel Cooperation (Channel 1 is captured by the overflow of Channel 0)

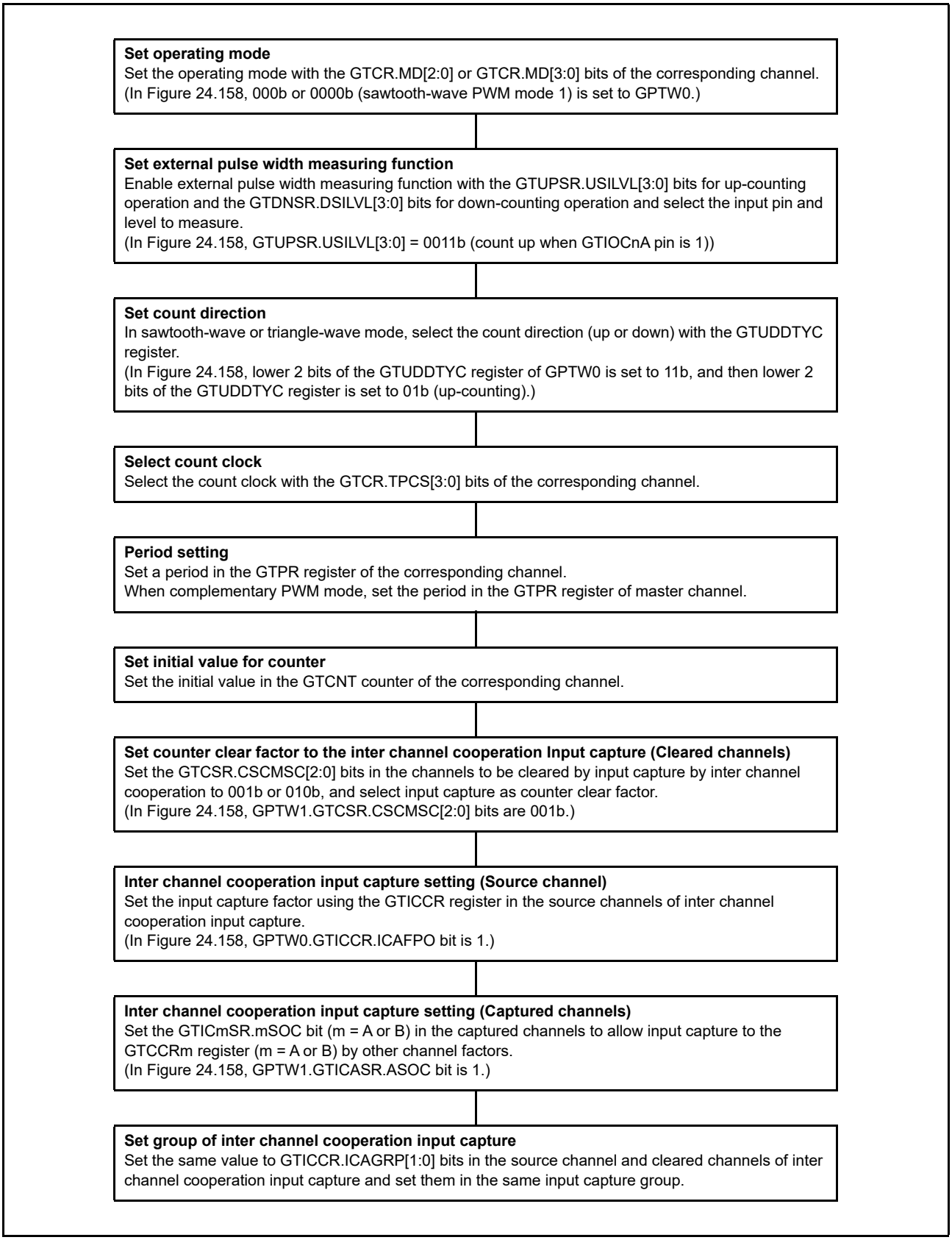


Figure 24.159 Example for Setting Input Capture Operation by Inter Channel Cooperation

24.3.10 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPTW can output two phases of linked PWM waveforms for one channel, or 32 phases of linked PWM waveforms for a maximum of 16 channels by synchronizing operation of the channels.

Figure 24.160 shows an example in which four channels perform synchronous operation in sawtooth-wave PWM mode 1 and eight phases of PWM waveforms are output. The GTIOCnA pin (n = 0 to 3) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output low as the initial output, high at a GTCCRB register compare match, and low at the end of the cycle.

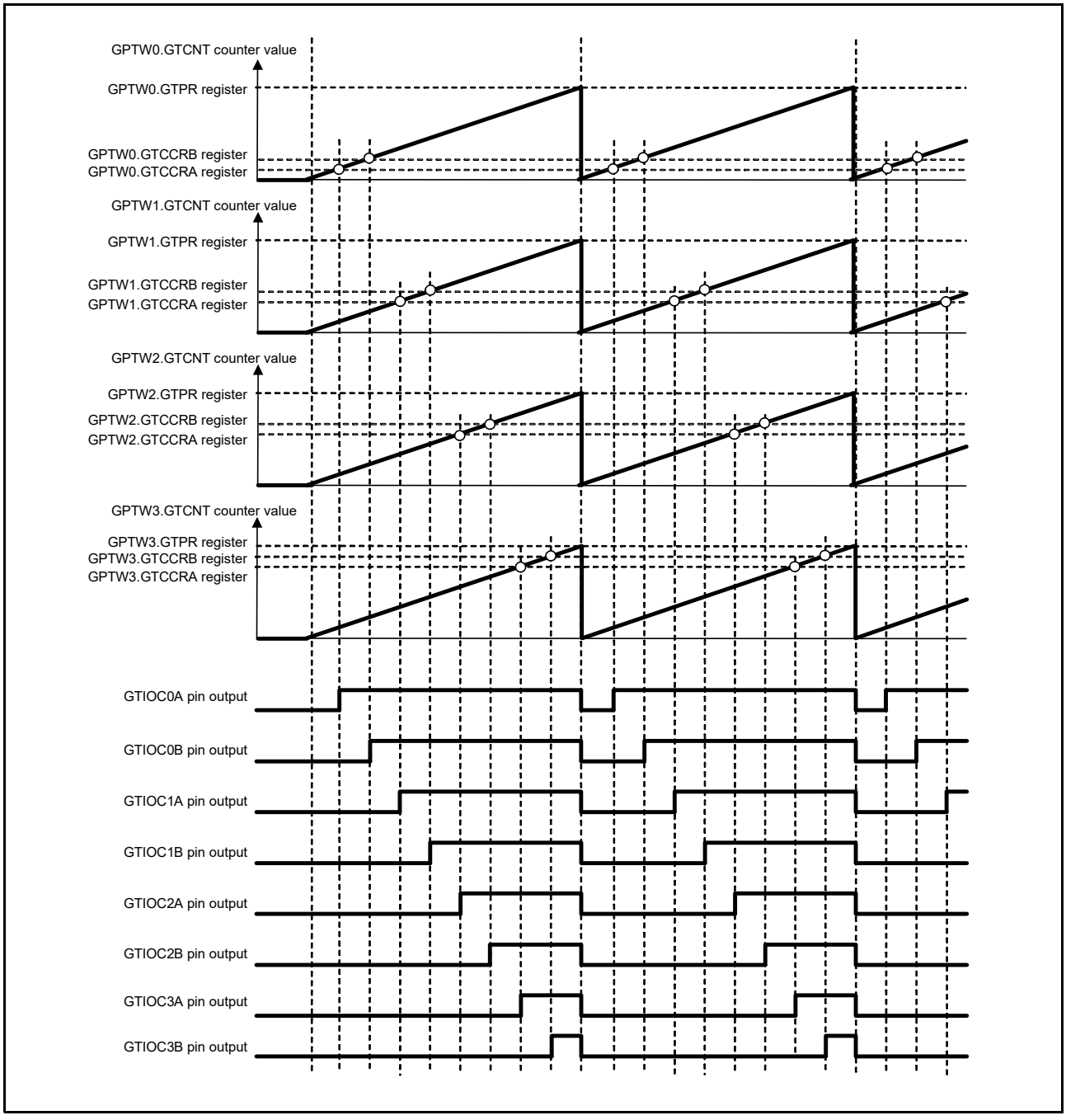


Figure 24.160 Example of Synchronous PWM Output

(2) Three-Phase Sawtooth-Wave Complementary PWM Output

Figure 24.161 shows an example in which three channels perform synchronous operation in sawtooth-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, low at a GTCCRB register compare match, and high at the end of the cycle.

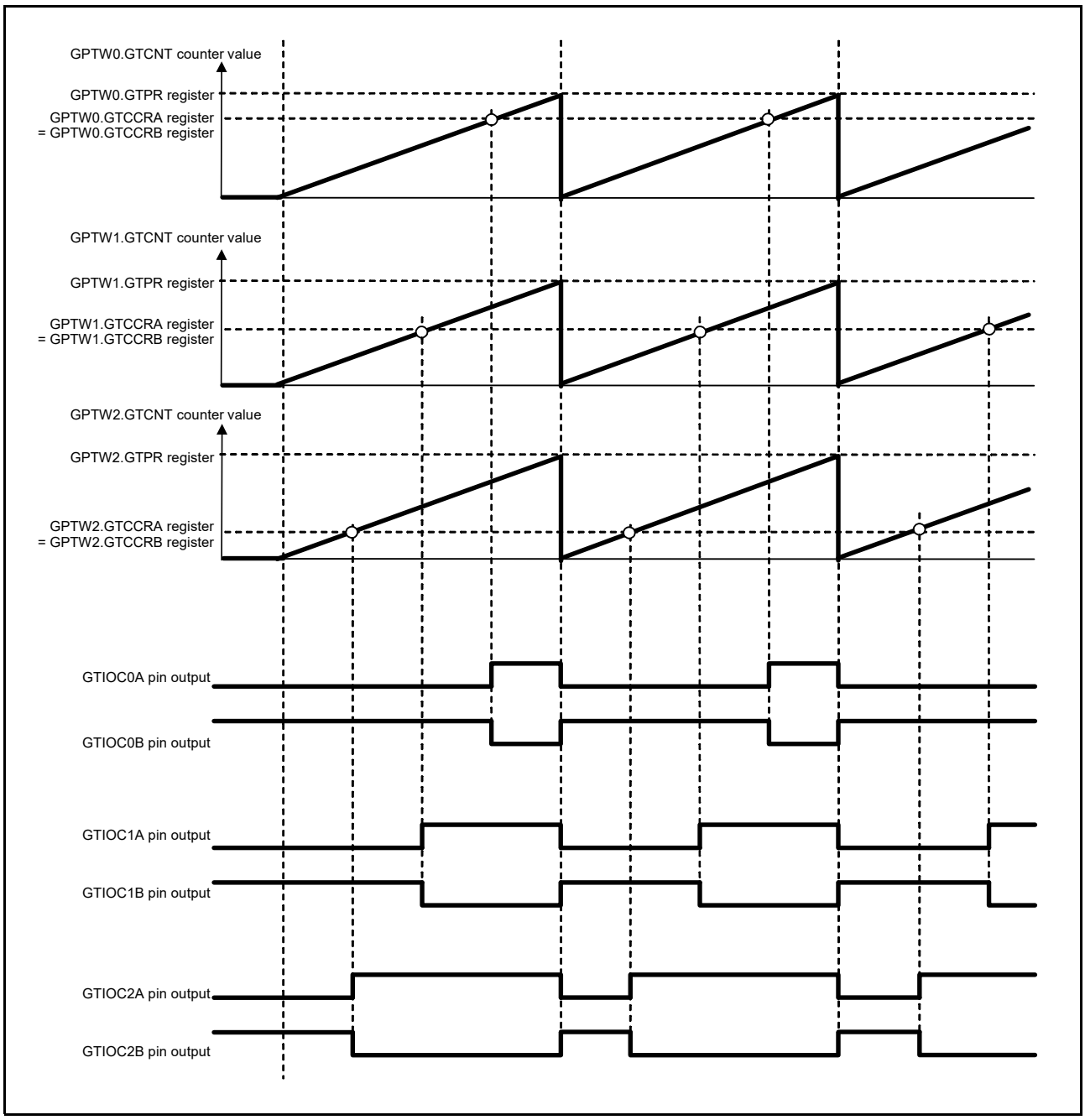


Figure 24.161 Example of Three-Phase Sawtooth-Wave Complementary PWM Output

(3) Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.162 shows an example in which three channels perform synchronous operation in sawtooth-wave one-shot pulse mode with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

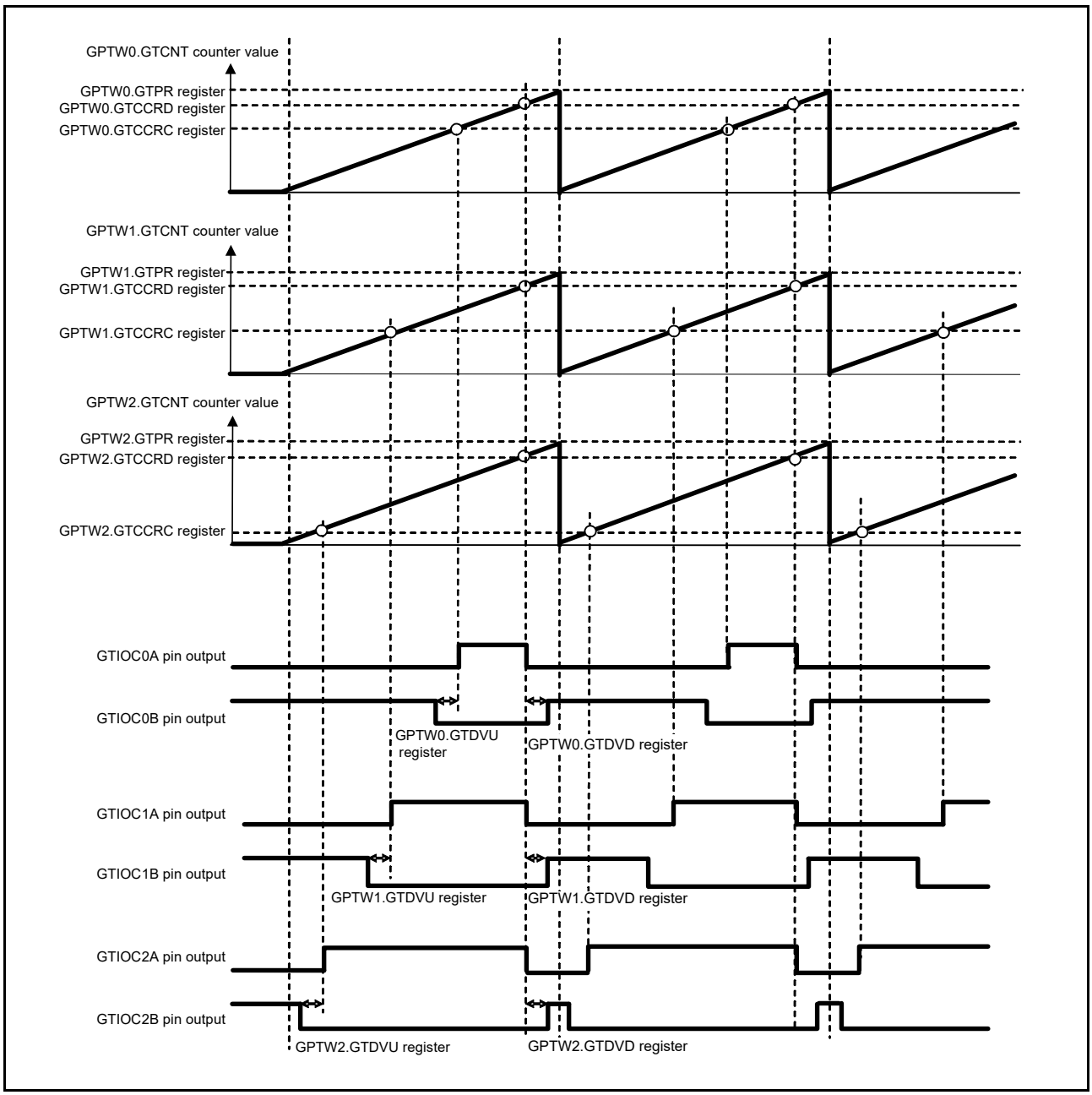


Figure 24.162 Example of Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

(4) Three-Phase Triangle-Wave Complementary PWM Output

Figure 24.163 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

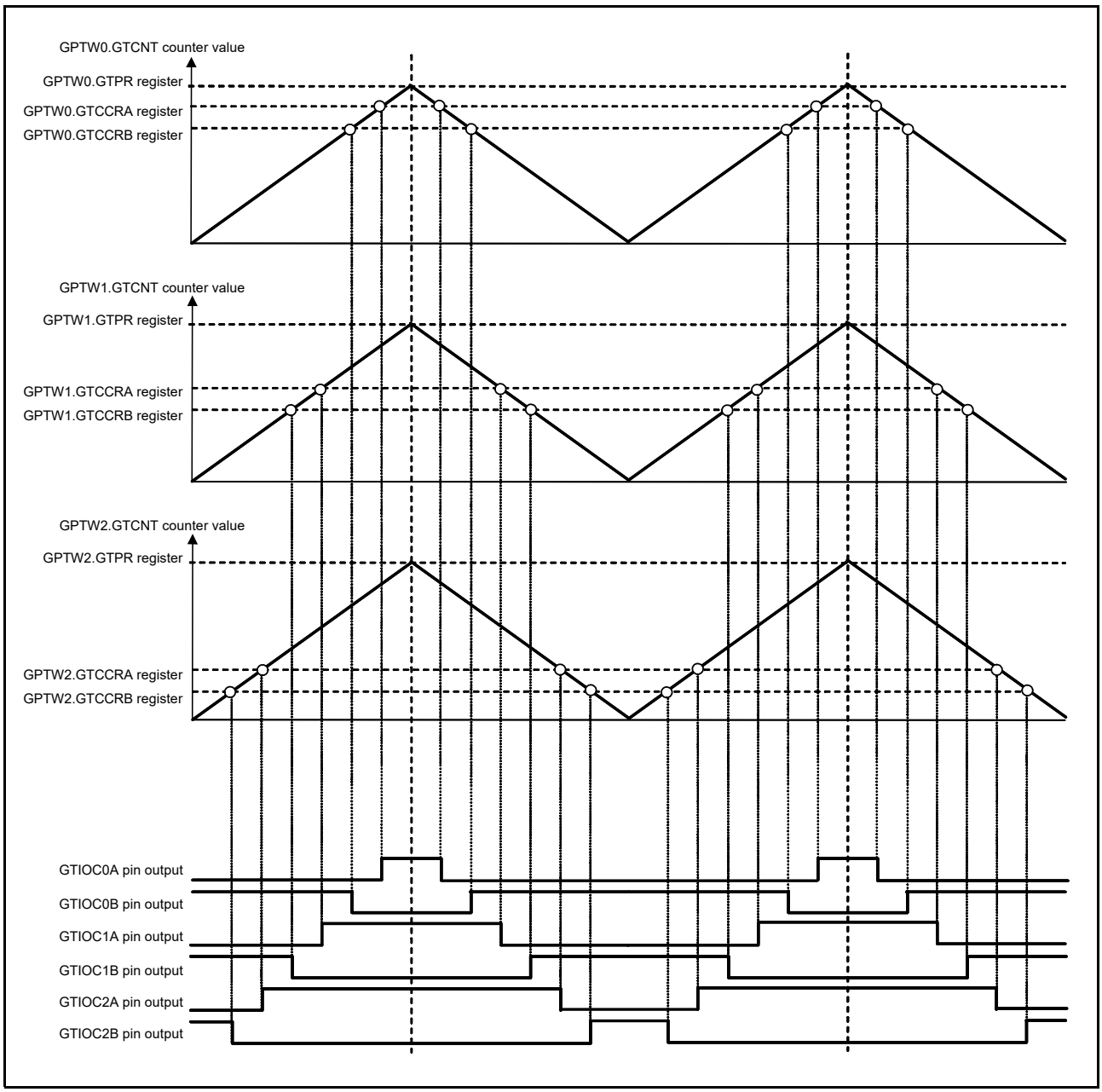


Figure 24.163 Example of Three-Phase Triangle-Wave Complementary PWM Output

(5) Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.164 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

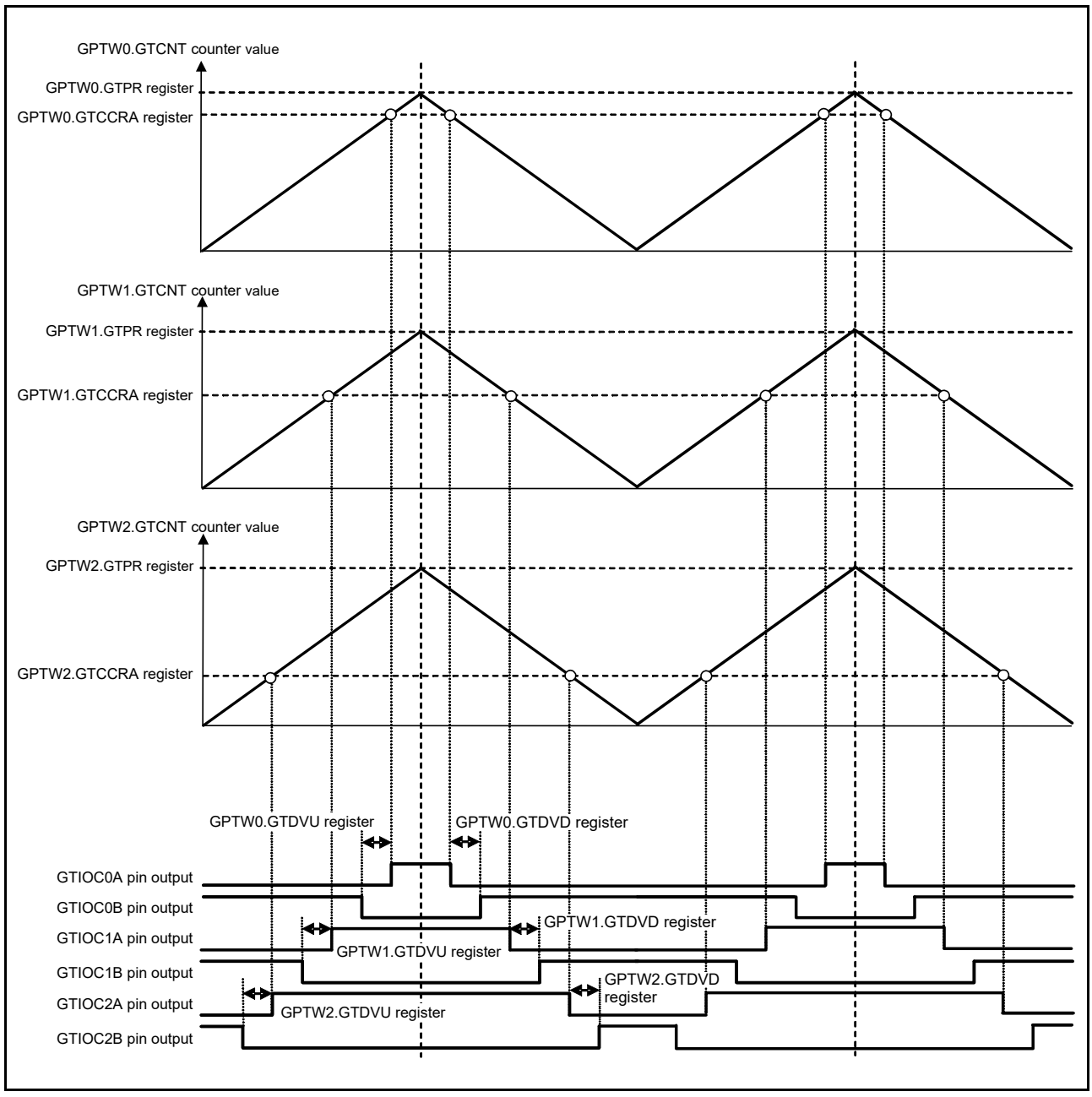


Figure 24.164 Example of Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

(6) Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.165 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 3 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

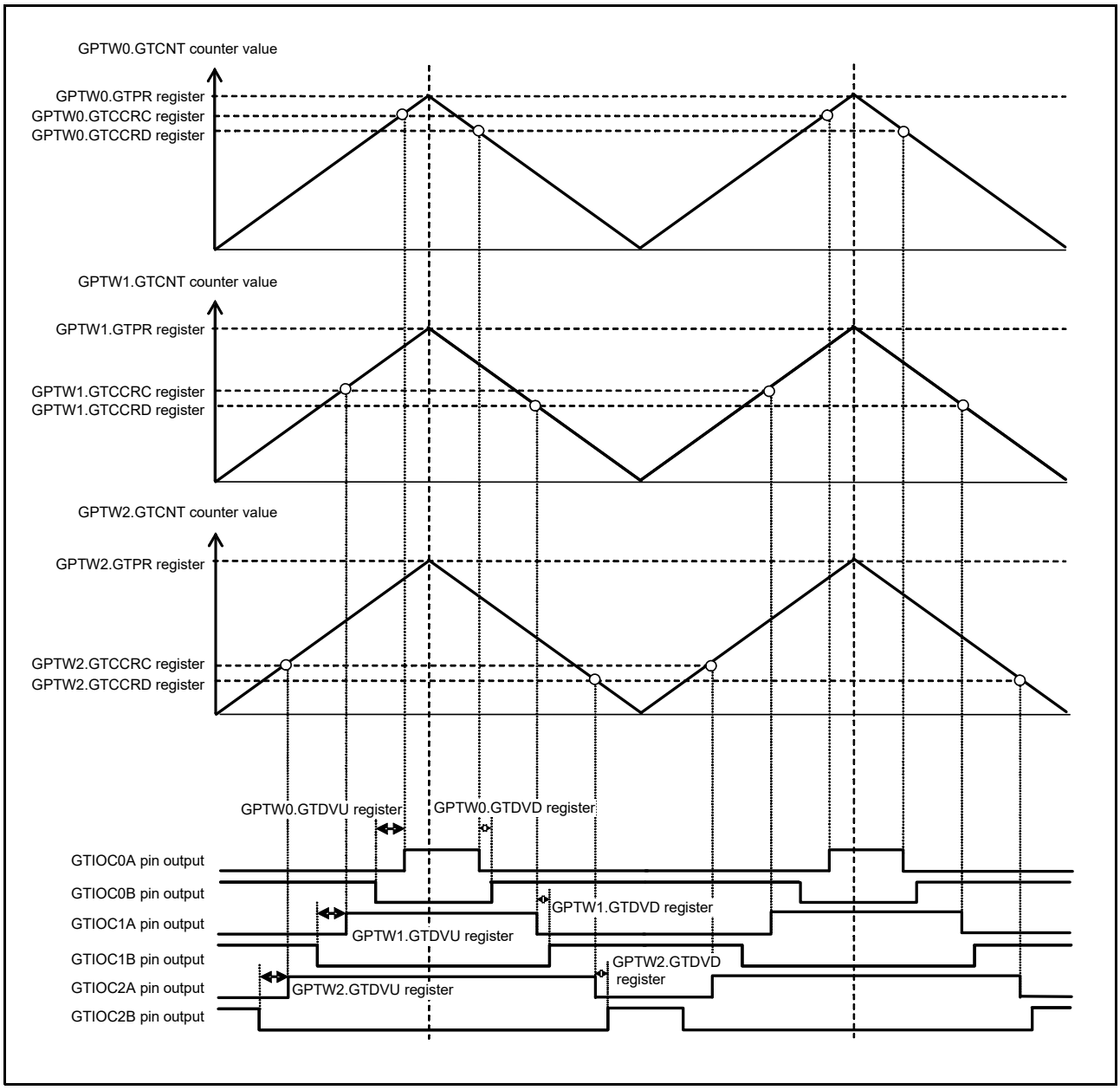


Figure 24.165 Example of Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

24.3.11 Cycle Count Function

By setting the GTPC register, the end of cycle can be counted.

The number of cycle to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of cycle. When the PCNT counter is 1 at the end of cycle, it becomes 0 and counting is stopped to finish the cycle count operation. At that time, the GTST.PCF flag is set, and a cycle count end interrupt request (GTCEIn) is generated if the setting of the GTINTAD.GTINTPC bit is 1. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the cycle count operation is finished.

When the GTCNT counter is stopped while cycle count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value. When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the cycle count function for multiple channels.

Figure 24.166 and Figure 24.167 show examples of PWM cycle count function.

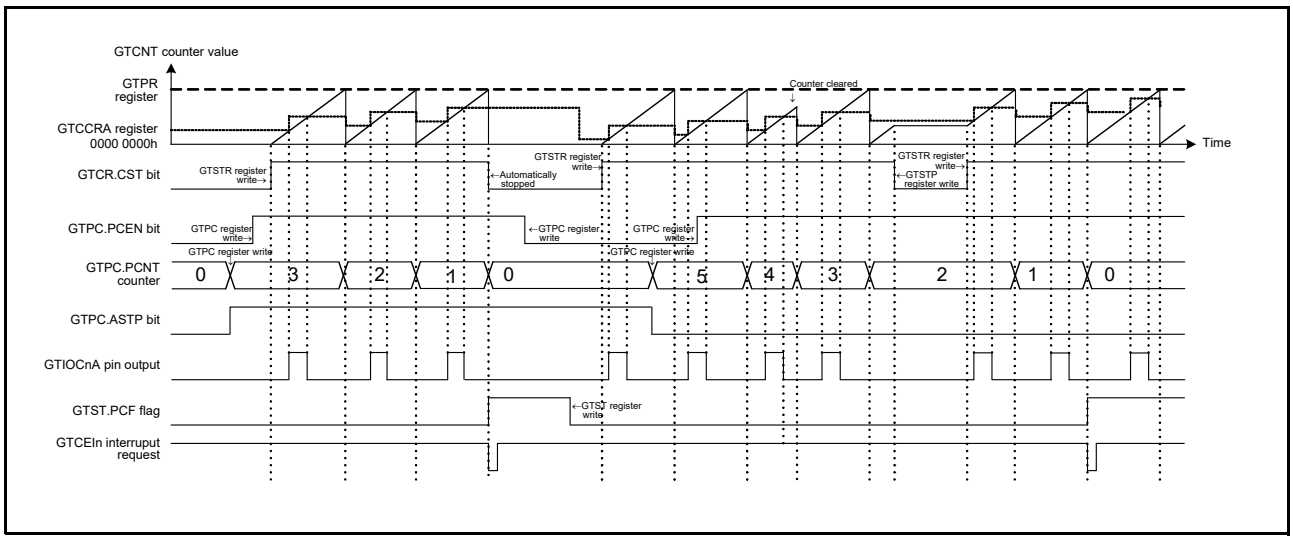


Figure 24.166 Example of PWM Cycle Count Function (Sawtooth-Wave One-Shot Pulse Mode) (n = 0 to 15)

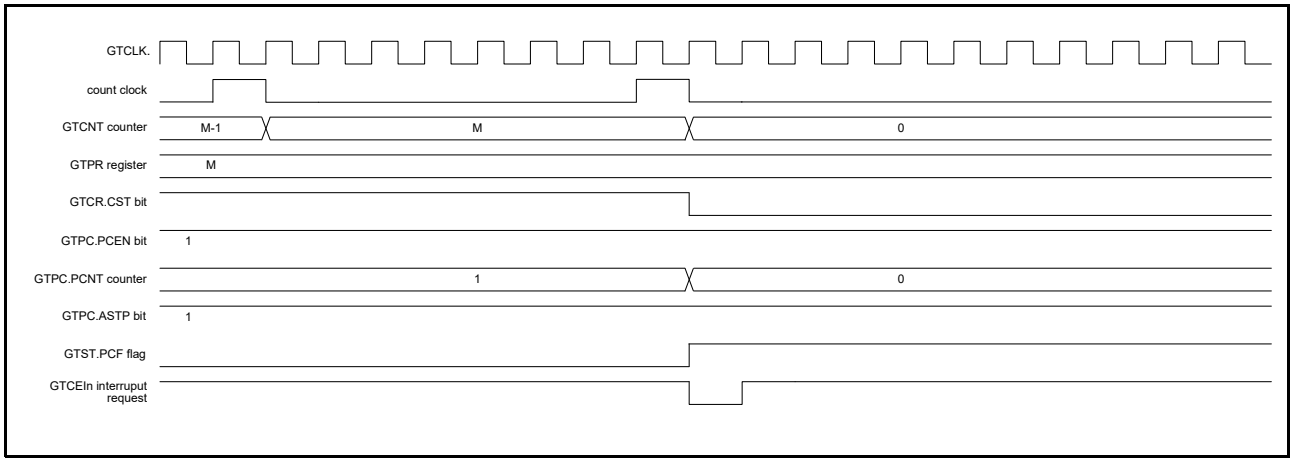


Figure 24.167 Example of the Timing of Operations for PWM Cycle Count Function (Sawtooth-Wave, Up-Counting) (n = 0 to 15)

24.3.12 Phase Counting Mode

The GTCNT counter can be up-counting or down-counting by detecting the phase difference between the GTIOCnA pin input (n = 0 to 15) and the GTIOCnB pin input.

Detection of the phase difference can be performed by setting the relationship between the edge and the level of the GTIOCnA pin input and the GTIOCnB pin input using the GTUPSR and GTDNSR registers for input combination. Refer to section 24.3.1.1, Counter Operation for count operation.

Figure 24.168 to Figure 24.177 show the examples of setting procedure for the phase counting mode 1 to 5, and the Table 24.25 to Table 24.34 show the setting of the GTUPSR and GTDNSR registers and up-counting and down-counting conditions.

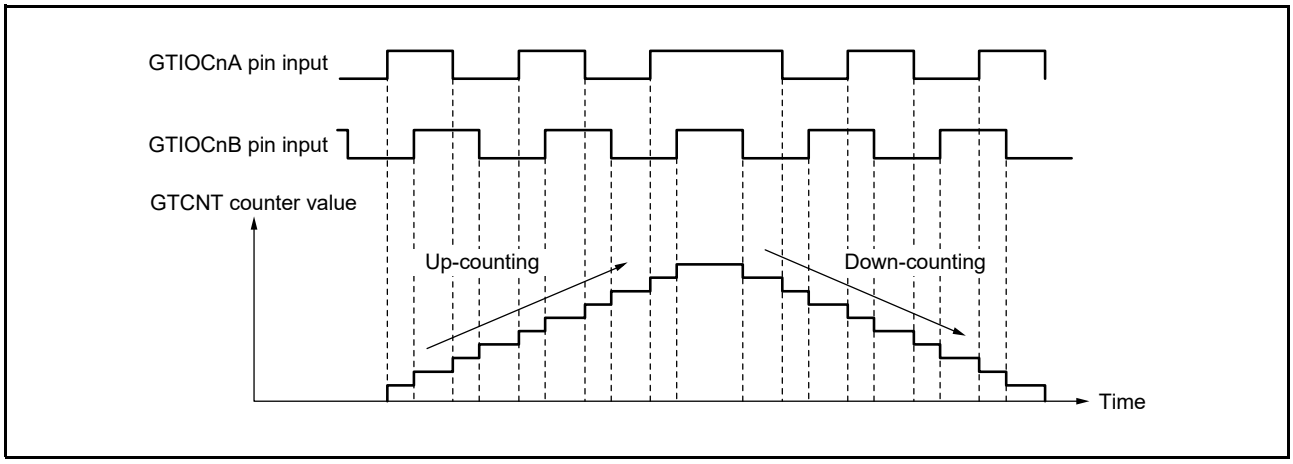


Figure 24.168 Example of setting procedure for the phase counting mode 1 (n = 0 to 15)

Table 24.25 Up-counting and down-counting conditions for the phase counting mode 1 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6900h GTDNSR = 0000 9600h
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

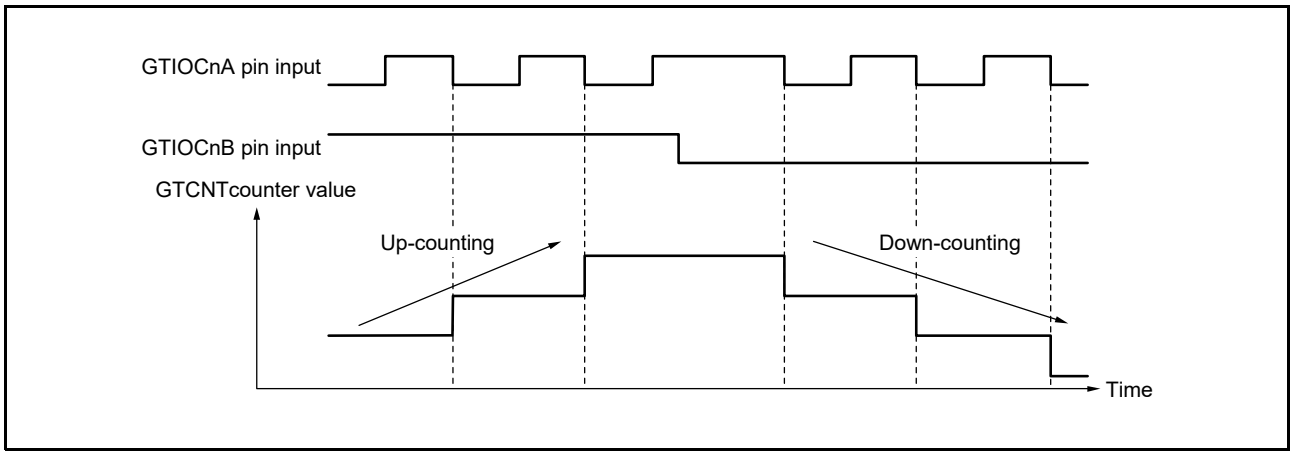


Figure 24.169 Example of setting procedure for the phase counting mode 2 (n = 0 to 15)

Table 24.26 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 0400h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

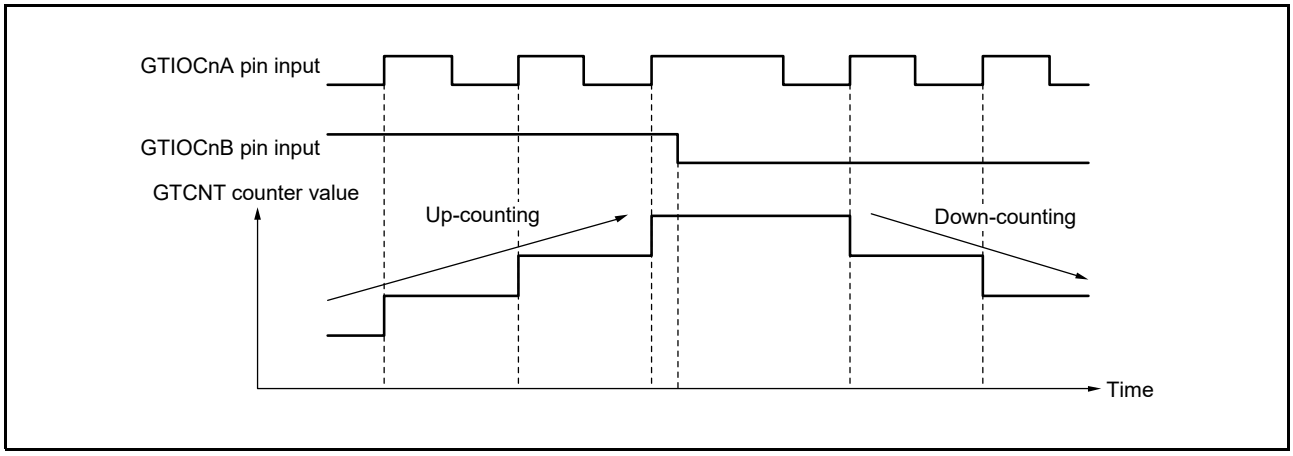


Figure 24.170 Example of setting procedure for the phase counting mode 2 (n = 0 to 15)

Table 24.27 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0200h GTDNSR = 0000 0100h
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

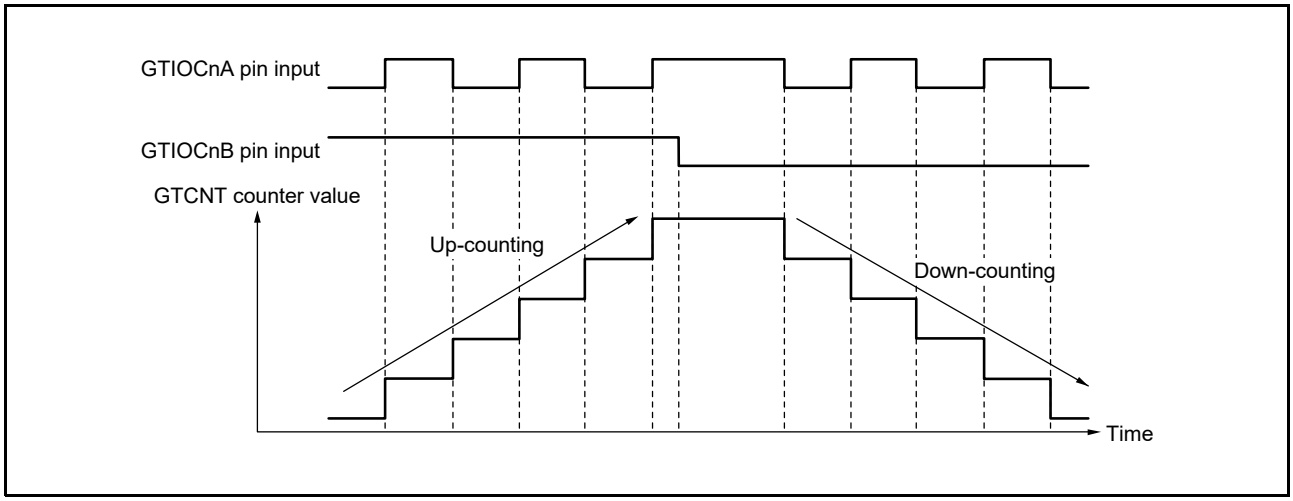


Figure 24.171 Example of setting procedure for the phase counting mode 2 (n = 0 to 15)

Table 24.28 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0A00h GTDNSR = 0000 0500h
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

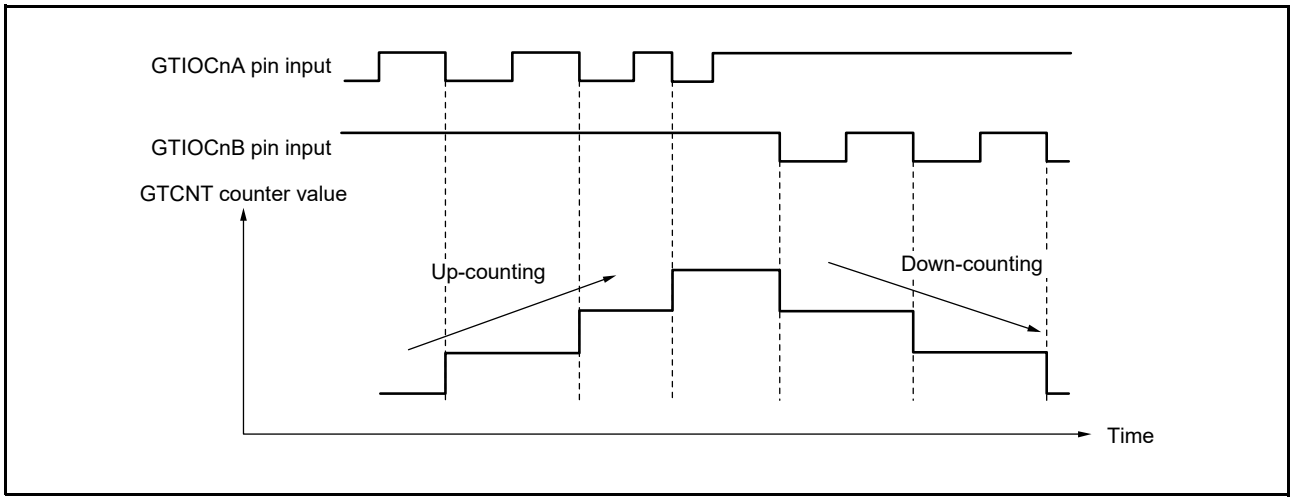


Figure 24.172 Example of setting procedure for the phase counting mode 3 (n = 0 to 15)

Table 24.29 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 8000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

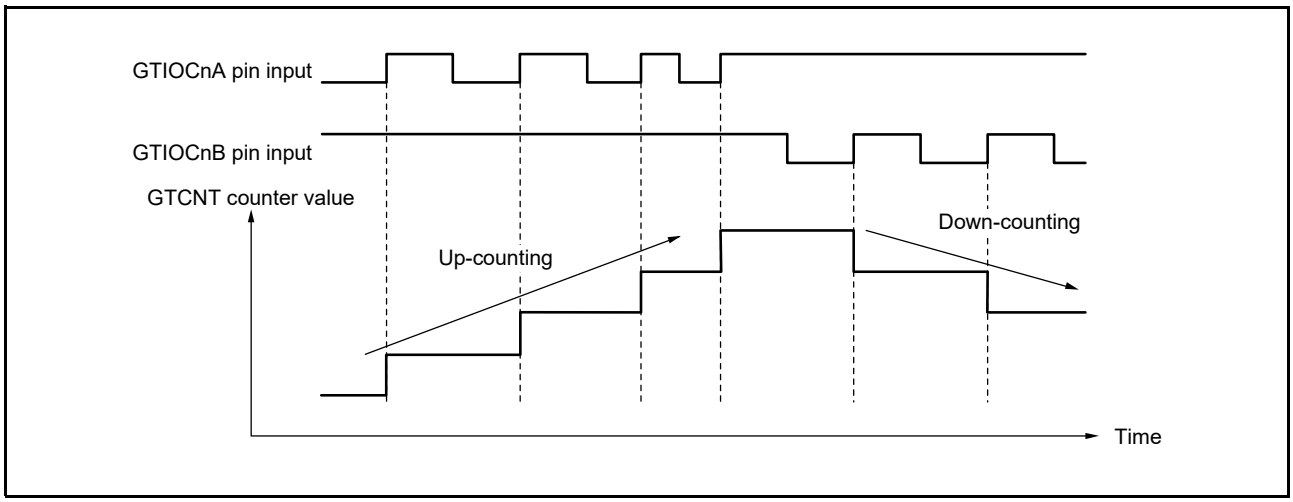


Figure 24.173 Example of setting procedure for the phase counting mode 3 (n = 0 to 15)

Table 24.30 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0200h GTDNSR = 0000 2000h
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

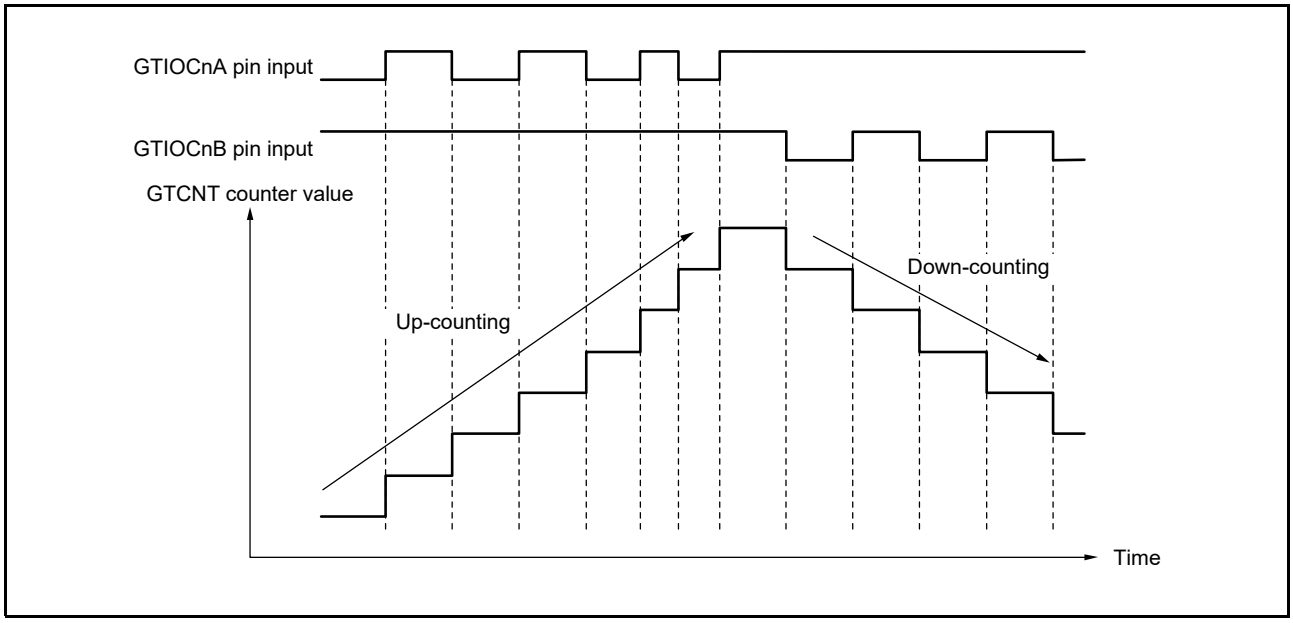


Figure 24.174 Example of setting procedure for the phase counting mode 3 (n = 0 to 15)

Table 24.31 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0A00h GTDNSR = 0000 A000h
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

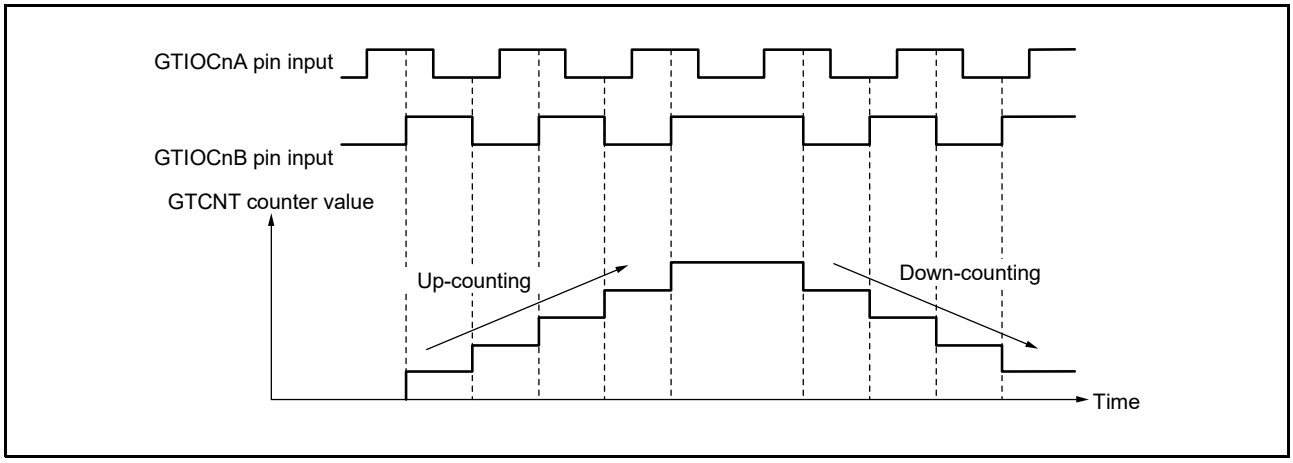


Figure 24.175 Example of setting procedure for the phase counting mode 4 (n = 0 to 15)

Table 24.32 Up-counting and down-counting conditions for the phase counting mode 4 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6000h GTDNSR = 0000 9000h
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

↑: Rising edge

↓: Falling edge

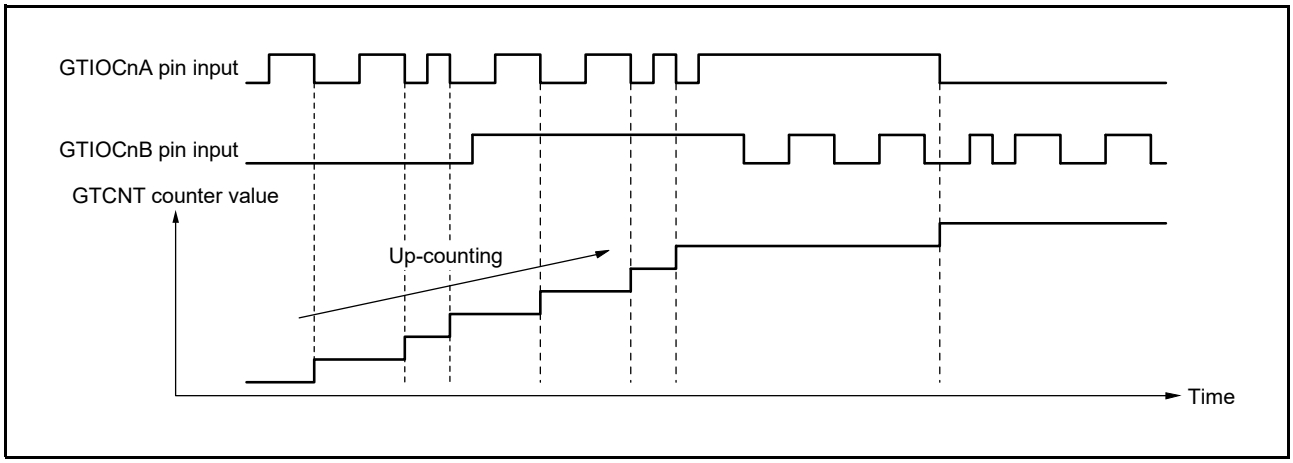


Figure 24.176 Example of setting procedure for the phase counting mode 5 (n = 0 to 15)

Table 24.33 Up-counting and down-counting conditions for the phase counting mode 5 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0C00h GTDNSR = 0000 0000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

↑: Rising edge

↓: Falling edge

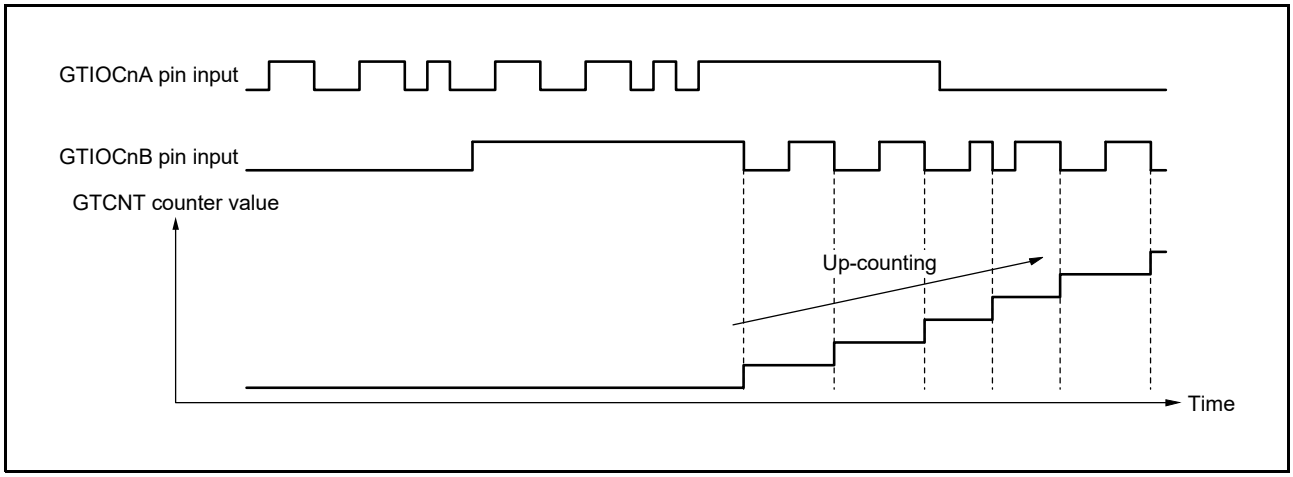


Figure 24.177 Example of setting procedure for the phase counting mode 5 (n = 0 to 15)

Table 24.34 Up-counting and down-counting conditions for the phase counting mode 5 (n = 0 to 15)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 C000h GTDNSR = 0000 0000h
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

24.3.13 External Pulse Width Measuring Function

The pulse width of GTIOCnA pin input (n = 0 to 15), GTIOCnB pin input, and GTETRGA/GTETRGB/GTETRGC/GTETRGD pin inputs can be measured.

The setting to enable or disable count-up of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

The setting to enable or disable count-down of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both count-up and count-down of the GTCNT counter at the same time is prohibited.

The counting operation performs periodic counting with the period of the GTPR register.

If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 24.178 and Figure 24.179 show examples of external pulse width measuring function and Figure 24.180 shows an example for setting external pulse width measuring function.

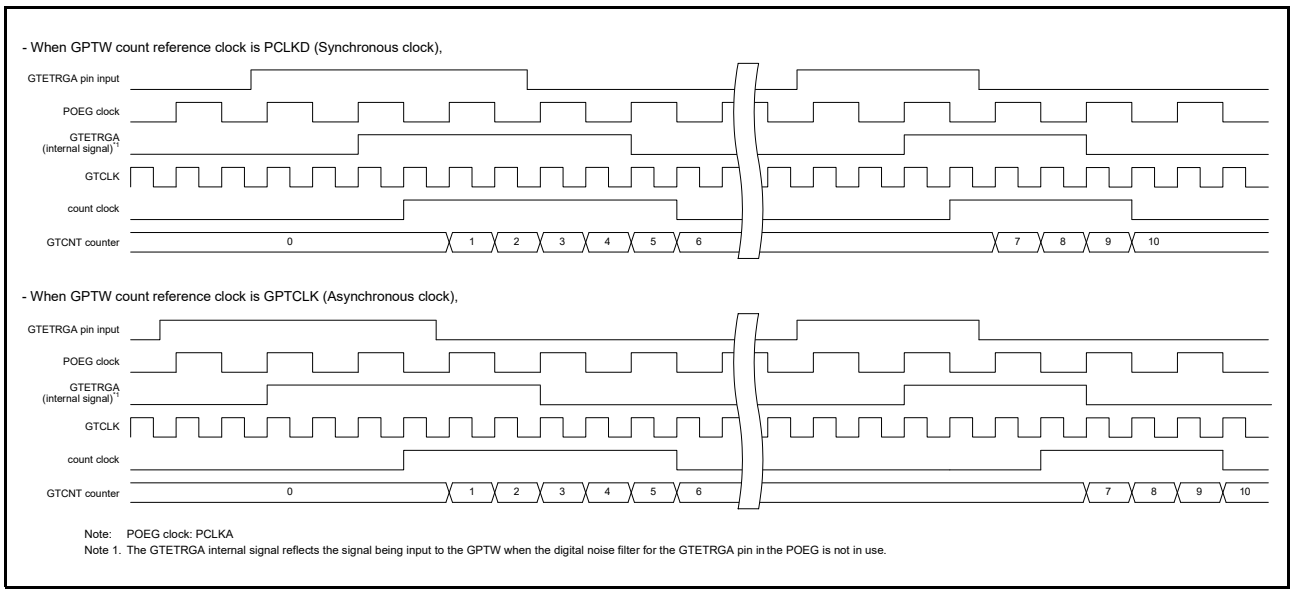


Figure 24.178 Example of External Pulse Width Measuring Function (Up-Counting)

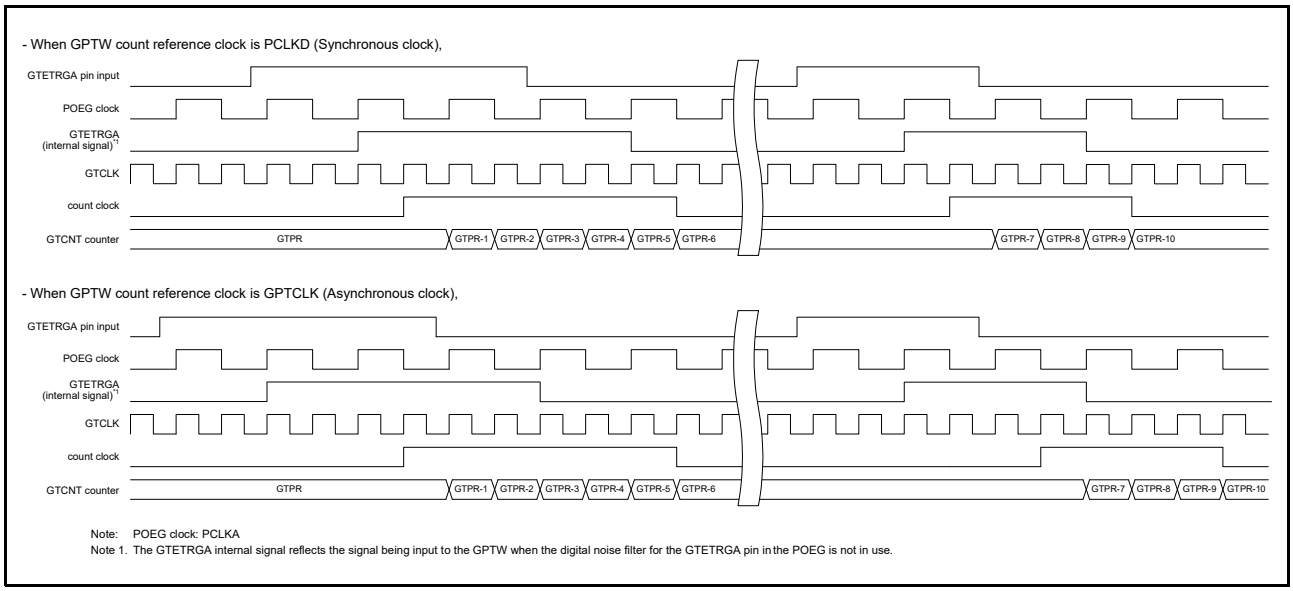


Figure 24.179 Example of External Pulse Width Measuring Function (Down-Counting)

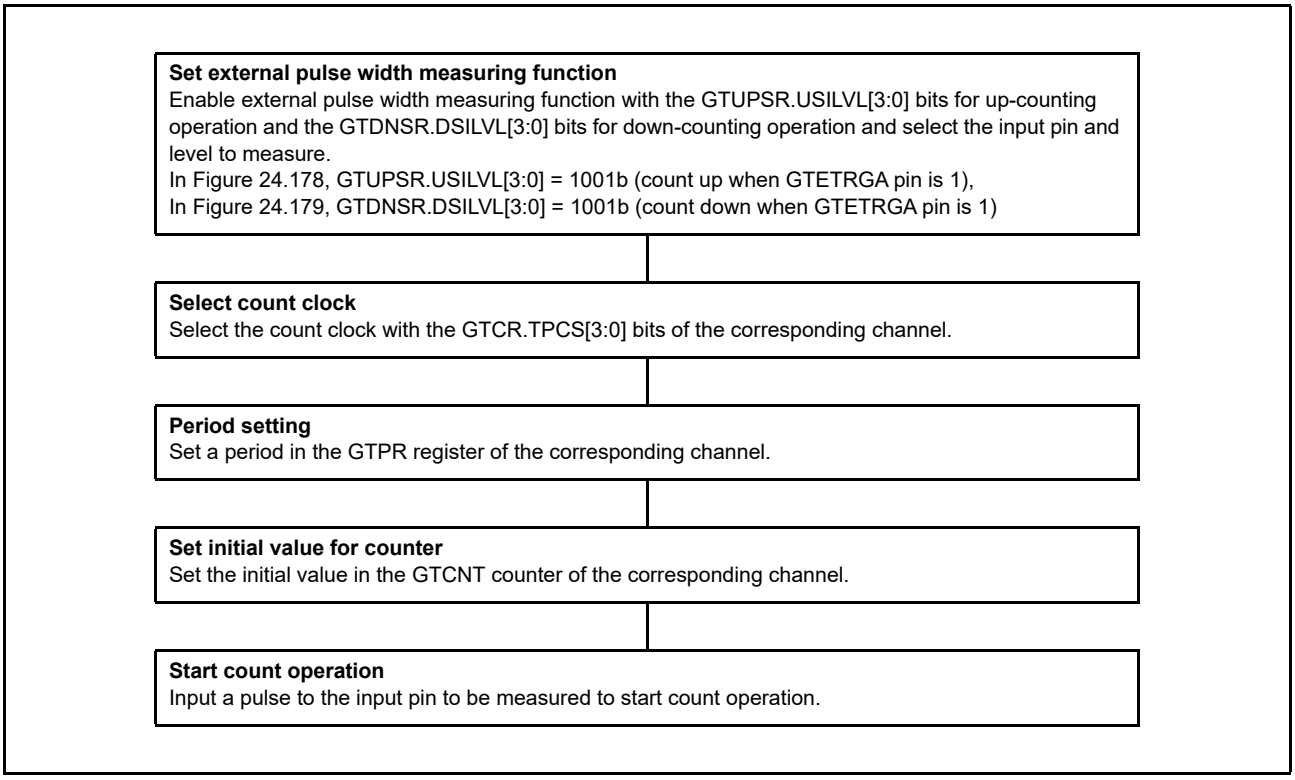


Figure 24.180 Example for Setting External Pulse Width Measuring Function

24.3.14 Output Phase Switching Control Function (OPS)

OPS can easily control Brushless DC motor using Output Phase Switching Control Register (OPSCR).
OPS uses S/W setting value (OPSCR.UF, VF, WF bits) or external signals detected by the Hall element as input signals.
OPS outputs either level signals or chopped signals by GPTW0's PWM as the 6-phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) signals to control motor.
Figure 24.181 shows the block diagram of OPS.
The GPT_UVWEDGE signal is output signal to ELC generated by detecting the edge of input signal.

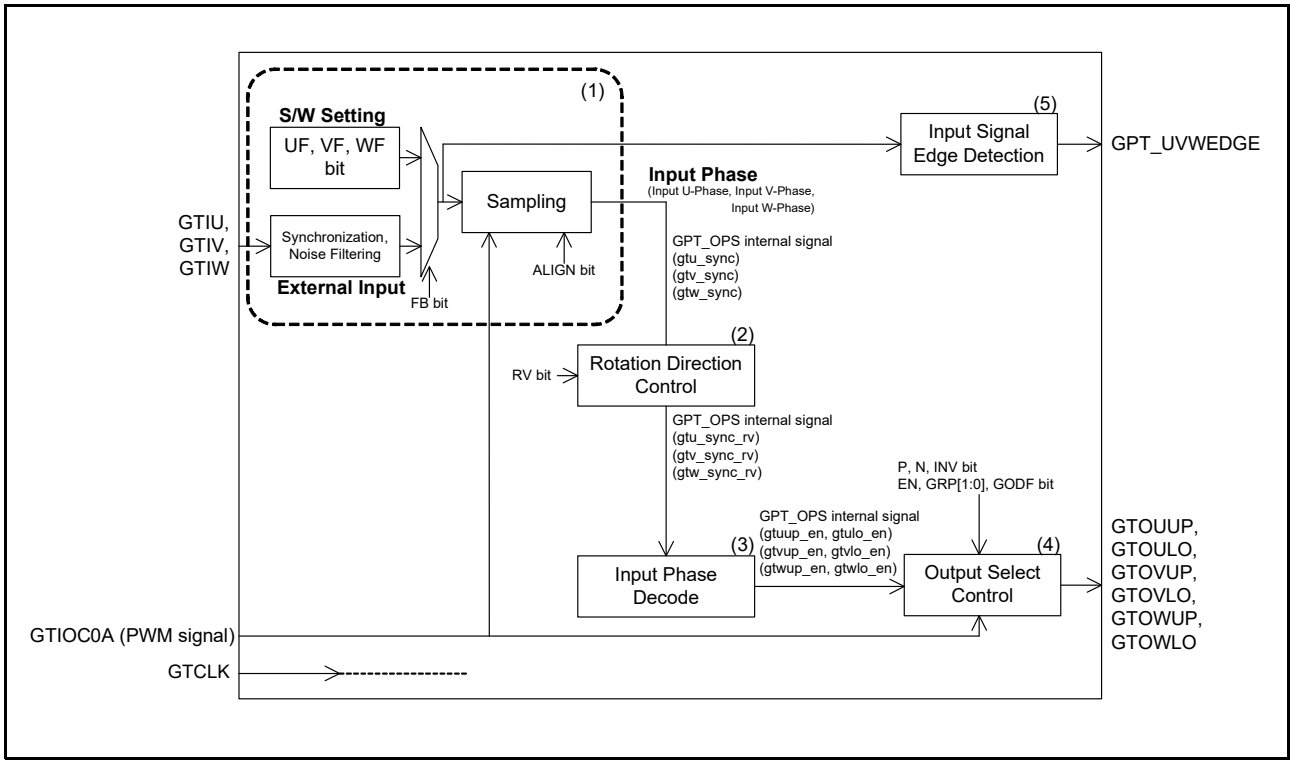


Figure 24.181 OPS Block Diagram

Figure 24.182 and Figure 24.183 show examples of OPS level output operation.

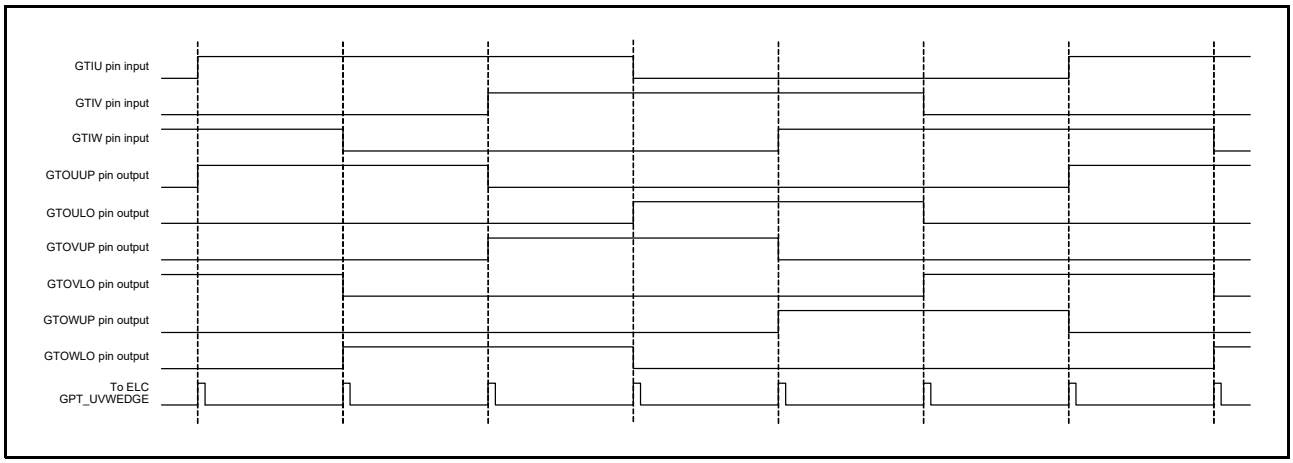


Figure 24.182 Example of OPS Level Output Operation (Forward Rotation) (FB = 0, RV = 0, P = 0, N = 0, INV = 0)

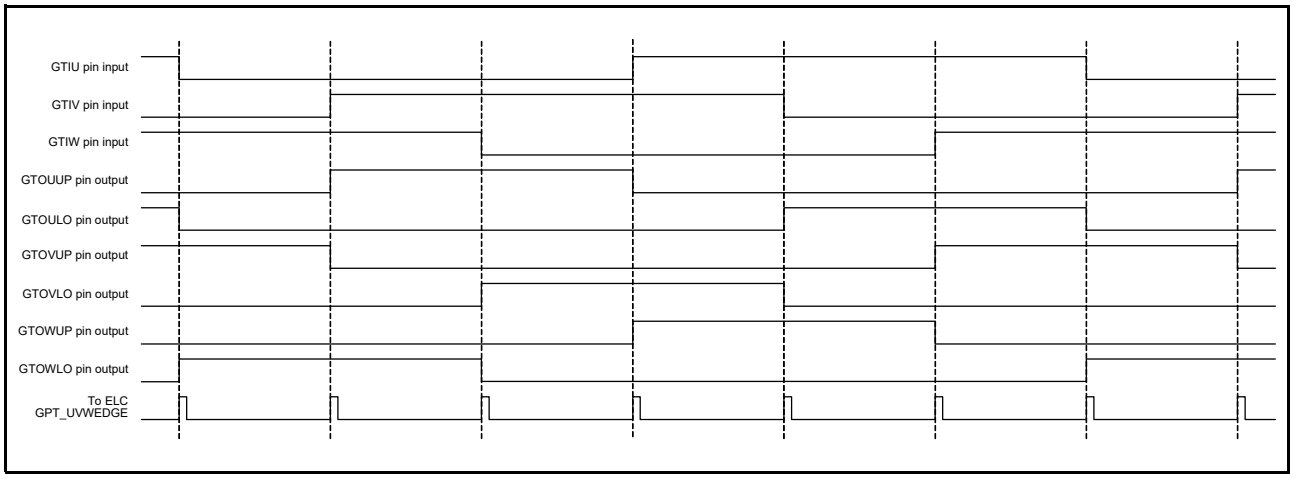


Figure 24.183 Example of OPS Level Output Operation (Reverse Rotation) (FB = 0, RV = 1, P = 0, N = 0, INV = 0)

Figure 24.184 and Figure 24.185 show examples of OPS chopped output operation.

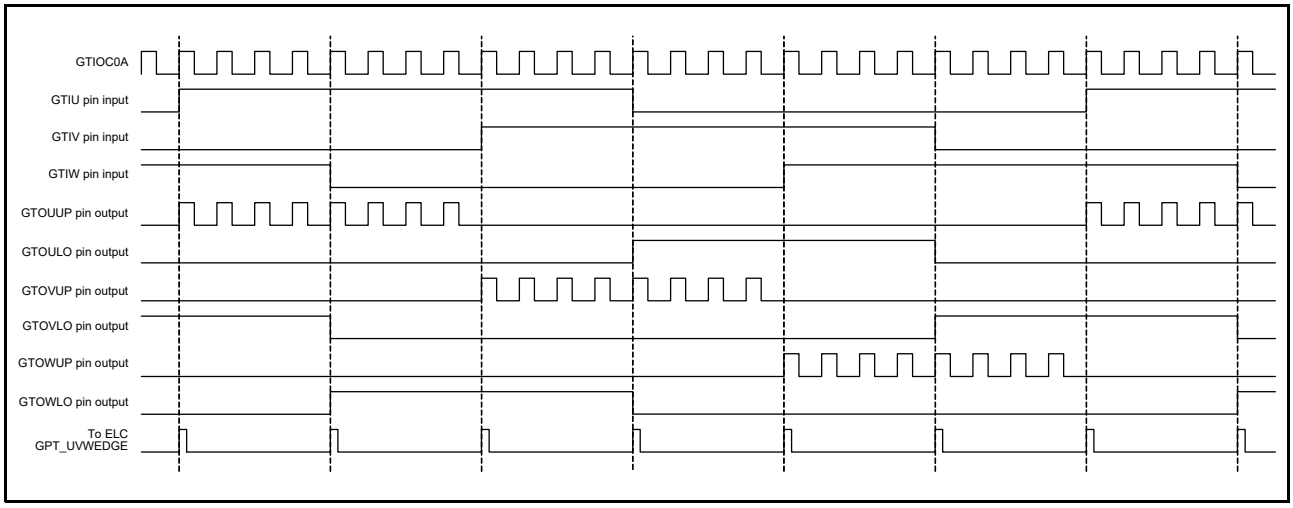


Figure 24.184 Example of OPS Chopped Output Operation (Positive Phase 120-degree) (FB = 0, RV = 0, P = 1, N = 0, INV = 0)

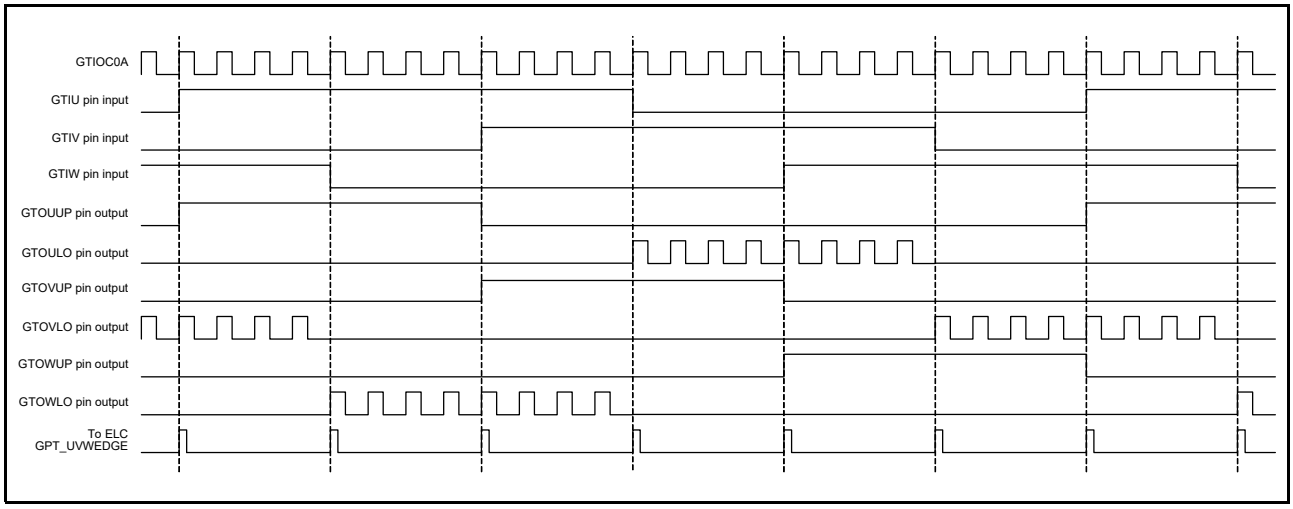


Figure 24.185 Example of OPS Chopped Output Operation (Negative Phase 120-degree) (FB = 0, RV = 0, P = 0, N = 1, INV = 0)

Figure 24.186 shows an example of OPS output disable control operation.

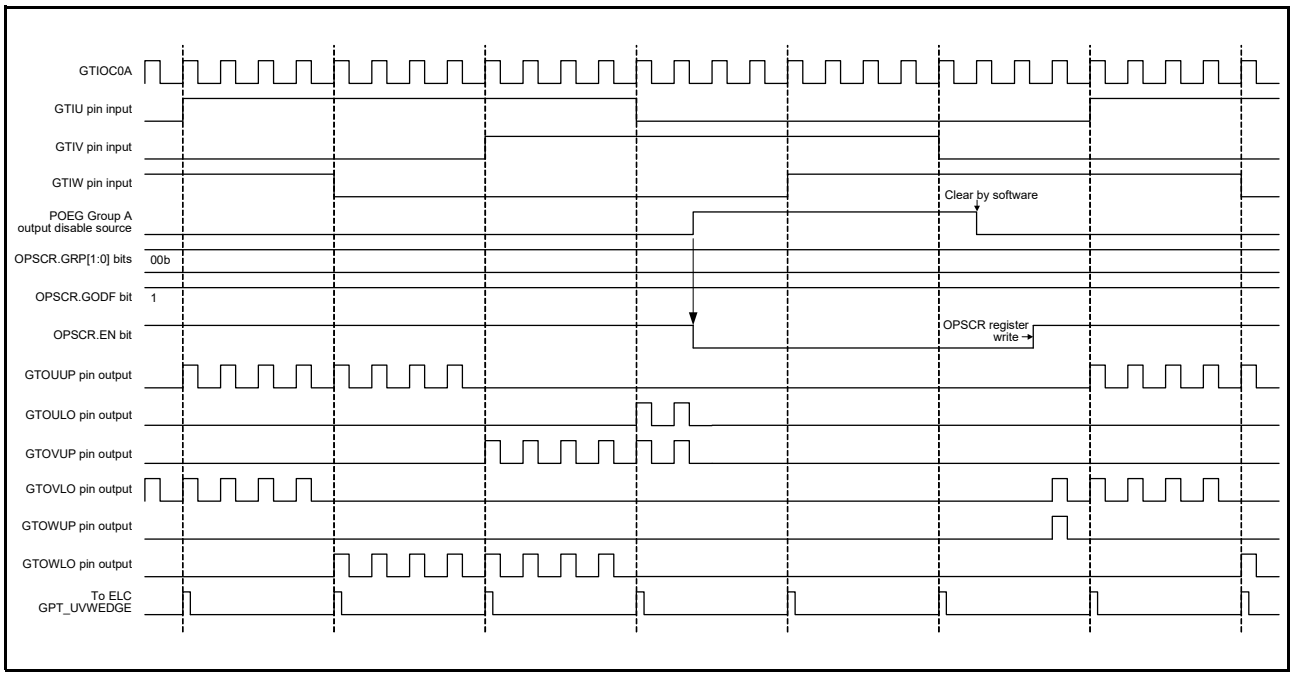


Figure 24.186 Example of OPS Output Disable Control Operation (FB = 0, RV = 0, P = 1, N = 1, INV = 0)

24.3.14.1 Input Selection and Sampling

The FB bit selects either the software setting value or external input for the input signal.
When the FB bit is 0, the GTIU, GTIV, GTIW external input are selected for the input signal to OPS after synchronization with the GPTW count reference clock (GTCLK) and noise filtering.
When the FB bit is 1, the software setting value (UF, VF, WF bits) are selected for the input signal to OPS.
The selected input signals are sampled by the method selected by the ALIGN bit, and they are treated as input phase of OPS.
When the ALIGN bit is 0, the input signals are sampled by GTCLK.
When the ALIGN bit is 1, the input signals are sampled by the falling edge of GTIOC0A pin output.
The signals after sampling can be read by the U, V, W bits.
Table 24.35 shows the input selection by the FB bit and sampling method by the ALIGN bit.

Table 24.35 Input Selection and Sampling Method

OPSCR Register		Input Selection Sampling Method	Interrupt Phase (OPS internal signal)
FB bit	ALIGN bit		
0	0	GTIU, GTIV, GTIW external input GTCLK sampling	Input U-phase (gtu_sync) Input V-phase (gtv_sync) Input W-phase (gtw_sync)
	1	GTIU, GTIV, GTIW external input GTIOC0A falling edge sampling	
1	0	Software setting value UF, VF, WF bit GTCLK sampling	
	1	Software setting value UF, VF, WF bit GTIOC0A falling edge sampling	

24.3.14.2 Rotation Direction Control

When the rotation direction is reverse (RV = 1), the input phase is inverted.

24.3.14.3 Input Phase Decode

The 6-phase signals by decoding input phase after rotation direction control are generated.
Table 24.36 and Table 24.37 show the decode tables of input phase to rotate motor in forward (RV = 0) and reverse (RV = 1).

Table 24.36 The Decode Table of Input Phase (Forward Rotation)

Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0	0	1
1	1	0	1	1	0	0	0	1	0	0	1
0	1	0	0	1	0	0	1	1	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0
0	0	1	0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

Table 24.37 The Decode Table of Input Phase (Reverse Rotation)

Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	0	1	0	0	1	1	0	0	0
1	0	0	0	1	1	0	1	0	0	1	0
1	1	0	0	0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

24.3.14.4 Output Selection Control

The EN, P, N, INV bits select the output wave.
The EN bit enables output of the 6-phase output. When the EN bit is 1, output of the 6-phase output is enabled. When the EN bit is 0, the external pin output is Hi-Z.
The P, N bits select whether chopping positive and negative phase are performed or not. When P, N bits are 1, chopping is performed by GTIOC0A pin output.
When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.
The INV bit selects the polarity (either positive logic or negative logic) of phase output.
Table 24.38 and Table 24.39 show the output selection control method for positive and negative phase output.

Table 24.38 Output Selection Control Method (Positive Phase)

EN bit	P bit	INV bit	GTOnUP
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmup_en)
1	0	1	Negative logic level output (~gtmup_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtmup_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtmup_en))

n = U, V, W
m = u, v, w

Table 24.39 Output Selection Control Method (Negative Phase)

EN bit	P bit	INV bit	GTOnLO
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmlo_en)
1	0	1	Negative logic level output (~gtmlo_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtmlo_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtmlo_en))

n = U, V, W
m = u, v, w

24.3.14.5 Output Selection Control (Group Output Disable Function)

When the GODF bit is 1 and signal value selected by the GRP[1:0] bits is High (Output Disable Request), OPS's output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is cleared by the output disable request signal synchronized with GTCLK.
For the return, after clearing the Output Disable Request by software, set the EN bit to 1.
The timing of EN bit cleared is 3 GTCLK cycles after generating the output disable request. In order to perform the output disable control surely, the output disable request flag in POEG should be cleared in the timing that terminating the output disable request is at least 4 GTCLK cycles after generating the output disable request.
The example of the operation of the group output disable control, see the above-mentioned Figure 24.186.

24.3.14.6 Event Link Controller (ELC) Output

The logical sum of the pulse detected by rising and falling edge of U, V, W phase input is output to the event link controller (ELC). When the high level period of input phase is short, there are cases that the detected edge is not transmitted to the ELC correctly because of the logical sum.

24.3.14.7 OPS Start Operation Setting Flow

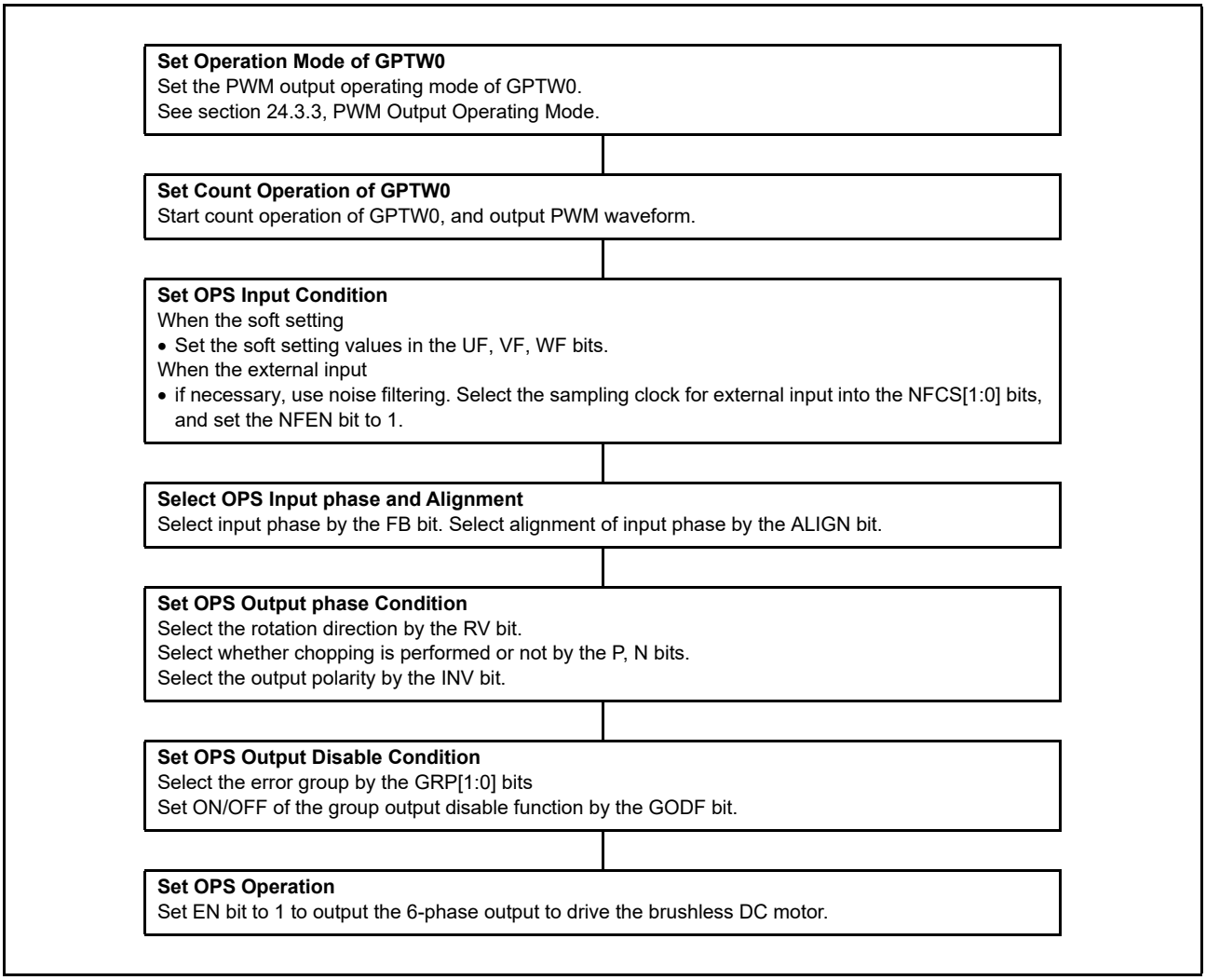


Figure 24.187 Example for Setting OPS Start Operation

24.3.15 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.

Figure 24.188 shows the block diagram of inter channel logical operation.

To prevent hazard to the this function output, the signal after logical operation is latched with GTCLK. After latching, the output disable control is performed.

When the logical operation function which causes the delay of 1 GTCLK is selected, the output enable signal is also delayed with 1 GTCLK and input to the output disable control. The execution of a logical operation does not delay the interrupts, A/D conversion start requests, and event signal output to the ELC.

When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

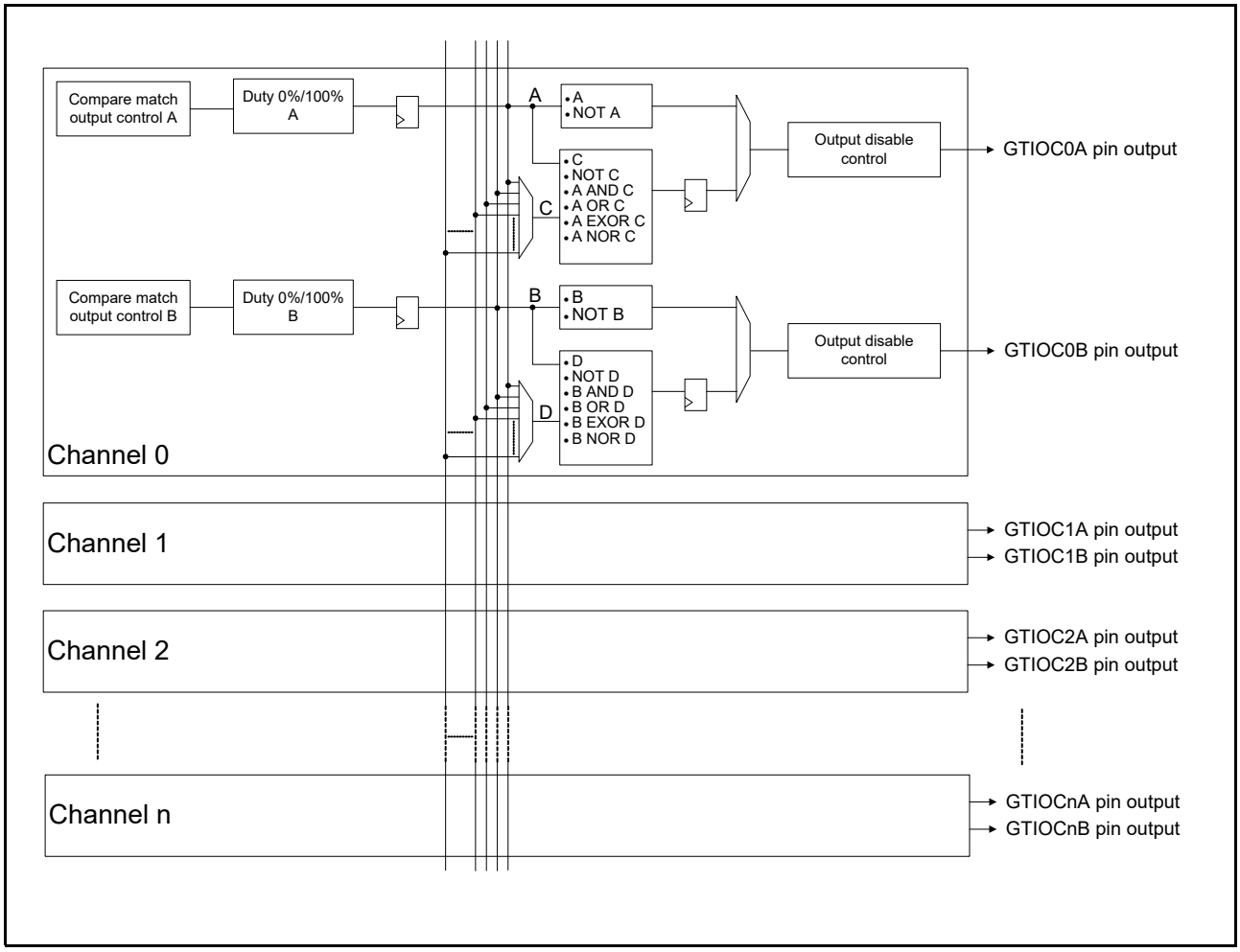


Figure 24.188 Block Diagram of Inter Channel Logical Operation

Figure 24.189 shows an example of inter channel logical operation.

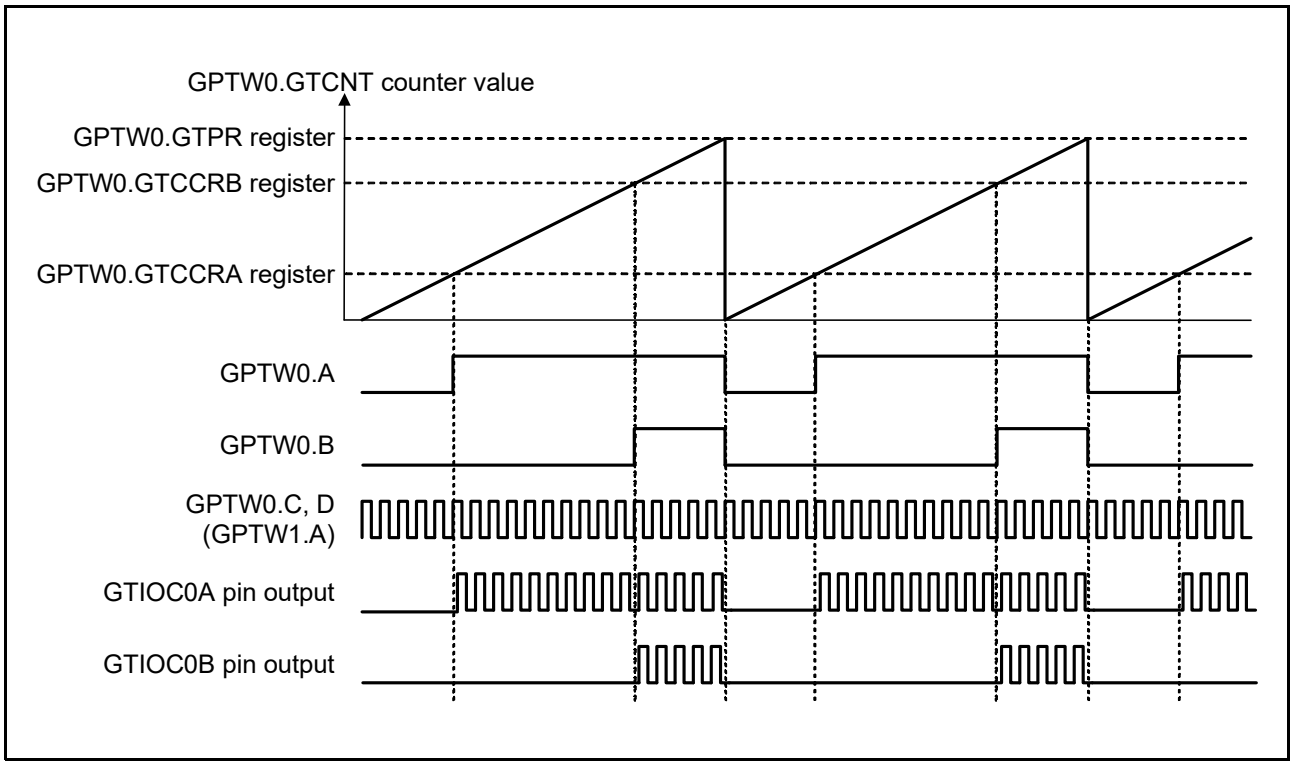


Figure 24.189 Example of Inter Channel Logical Operation (When GPTW0.GTICLF.ICLFA[2:0] = 100b, ICLFB[2:0] = 100b, GPTW0.GTICLF.ICLFSELC[5:0] = 000010b, ICLFSELD[5:0] = 000010b)

24.3.16 GTIOCnA/GTIOCnB Input/Output Pin Polarity Reversal Function

The polarity of the GTIOCnA, GTIOCnB input pins and the GTIOCnA, GTIOCnB output pins can be inverted according to the polarity of the external device.

When the GTCR.AINV bit is 1, the input and output of GTIOCnA are inverted. When the GTCR.BINV bit is 1, the input and output of GTIOCnB are inverted.

The input and output of GTIOCnA and GTIOCnB are inverted immediately after input and at the final output stage, respectively. Configure the settings considering that the external input and output signals are inverted, especially when GTIOCnA and GTIOCnB are used as factors for count start/stop/clear, count-down/count-up, or input capture, or when low/high/toggle output is used as PWM output.

Set GTCR.AINV and BINV after setting the GTIOCnm pin functions during initial settings.

24.3.17 Timer-Linked Synchronization Signal Output Function

The pulse signal (timer-linked synchronization signal) GTSYNC0A can be output to synchronize the A/D converter (ADC16H) and the timer operation of the GPTW upon compare match of the GTCNT counter value and the GTCCRA register value.
It is also possible to output a pulse signal (timer-internal synchronization signal) GTSYNCIN0 for synchronous operation between channels in the GPTW when an overflow (during up-count operation) occurs in sawtooth-wave PWM mode 1.
The SYSG[1:0] bits of the GTSYNCSCR register can be used to select whether to enable or disable output of the timer-linked synchronization signal and the timer-internal synchronization signal.
When GTSYNCSCR.SYSG[1:0] is 01b, the timer-linked synchronization signal pulse is output in response to a compare match with GTCCRA register, and the timer-internal synchronization signal pulse is output when an overflow (during counting up) occurs in sawtooth-wave PWM mode 1.
The SYST bit of the GTSYNCFCR register can be used to select the channel to be synchronized with the timer-internal synchronization signal for the count start/stop/clear operations.
Refer to section 60, 16-Bit A/D Converter (ADC16H) for the control method for synchronously operating ADC16H using the timer-linked synchronization signal.
Set the sawtooth wave PWM mode 1 for the channel that outputs the timer-linked synchronization signal and the timer-internal synchronization signal.
Set the GTSYNCFCR and GTSYNCSCR register before starting the count during initial setting.
Set the GTPR values of the synchronization control channel and synchronization target channel according to the synchronization period cycle of ADC16H to be synchronized with the timer of GPTW.
Set the GTPR value of the synchronization control channel to the same count cycle as the synchronization period of ADC16H. Setting the GTPR value of the synchronization control channel to 0 is prohibited.
Set the GTPR value of the synchronization target channel to a period that is an integral multiple of the GTPR value of the synchronization control channel. Figure 24.194 shows an example of the relationship between ADC16H, synchronization control channel, and synchronization target channel count cycles.
The specifications for the synchronization control channel are as follows.

- Only sawtooth PWM mode 1 is available.
- Only up-counting is possible for the counting direction.
- Synchronization control channel is prohibited from using a frequency-divided clock. Set the GTCR.TPCS[3:0] bits to 0000b.
- The GPTW's own frequency-divided clock is also synchronized with the timer-internal synchronization signal. If the synchronization control channel is stopped, the divided clock is resynchronized when the synchronization control channel is restarted.
- Start counting while all GPTW channels and ADC16H are stopped.
- Always operate the synchronization control channel while GPTW and ADC16H are operating.
- Always operate the synchronization control channel except when resetting.
- Stop the counter when changing the GTCCRA register value and the GTPR value of the synchronization control channel.
- Buffer setting is prohibited for synchronization control channel.
- Clear the counter when stop the synchronization control channel counter and start counting again.

The specifications for the synchronization target channel are as follows.

- Supports all PWM modes. (When using complementary PWM mode as a synchronization target, not only the master channel but also the slave channel must be set as a synchronization target.)
- Synchronization target channel can use divided clock.
- The count start/stop/clear is synchronized with the timer-internal synchronization signal.

- When performing synchronous clearing by inter channel cooperation, the GTSYNCFCR.SYST bit setting must match the opposite channel setting.

Figure 24.190 and Figure 24.191 show examples of synchronization control channel settings.
Figure 24.192 shows a configuration example of the timer-linked synchronization signal and the timer-internal synchronization signal.
Figure 24.193 shows an example of the timer-linked synchronization signal and the timer-internal synchronization signal operation.
Figure 24.194 shows the relationship of synchronization between the synchronization control channel, synchronization target channel, and ADC16H.

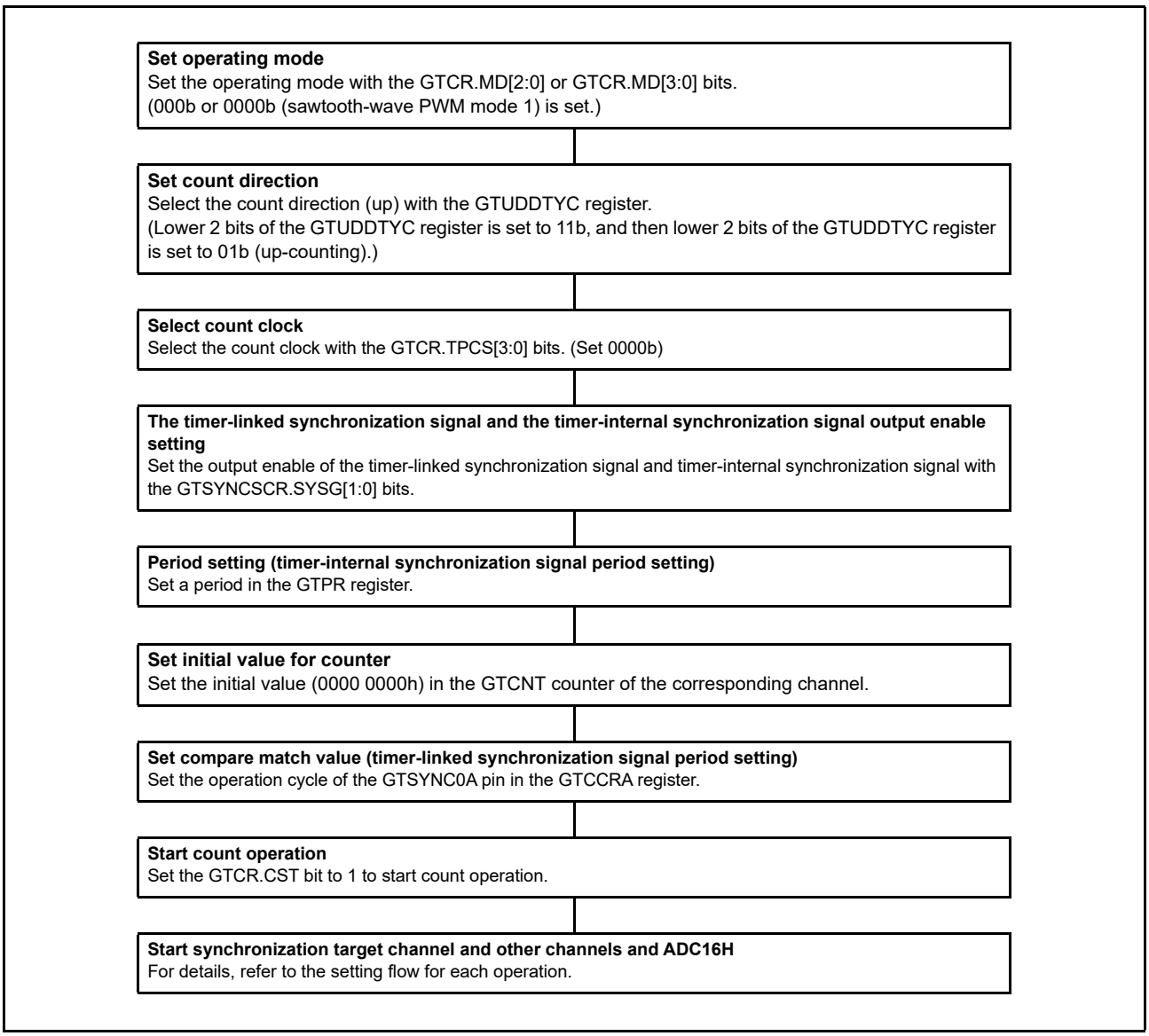


Figure 24.190 Example of Procedure for Starting Operation of the Synchronization Control Channel

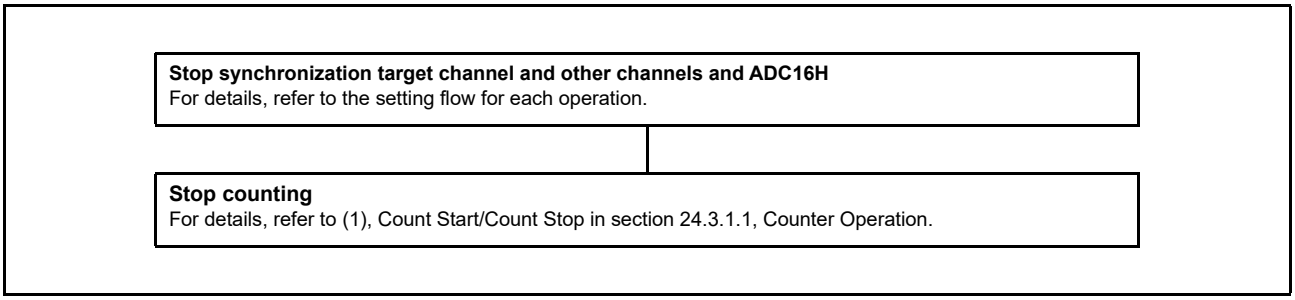


Figure 24.191 Example of Procedure for Stopping Operation of the Synchronization Control Channel

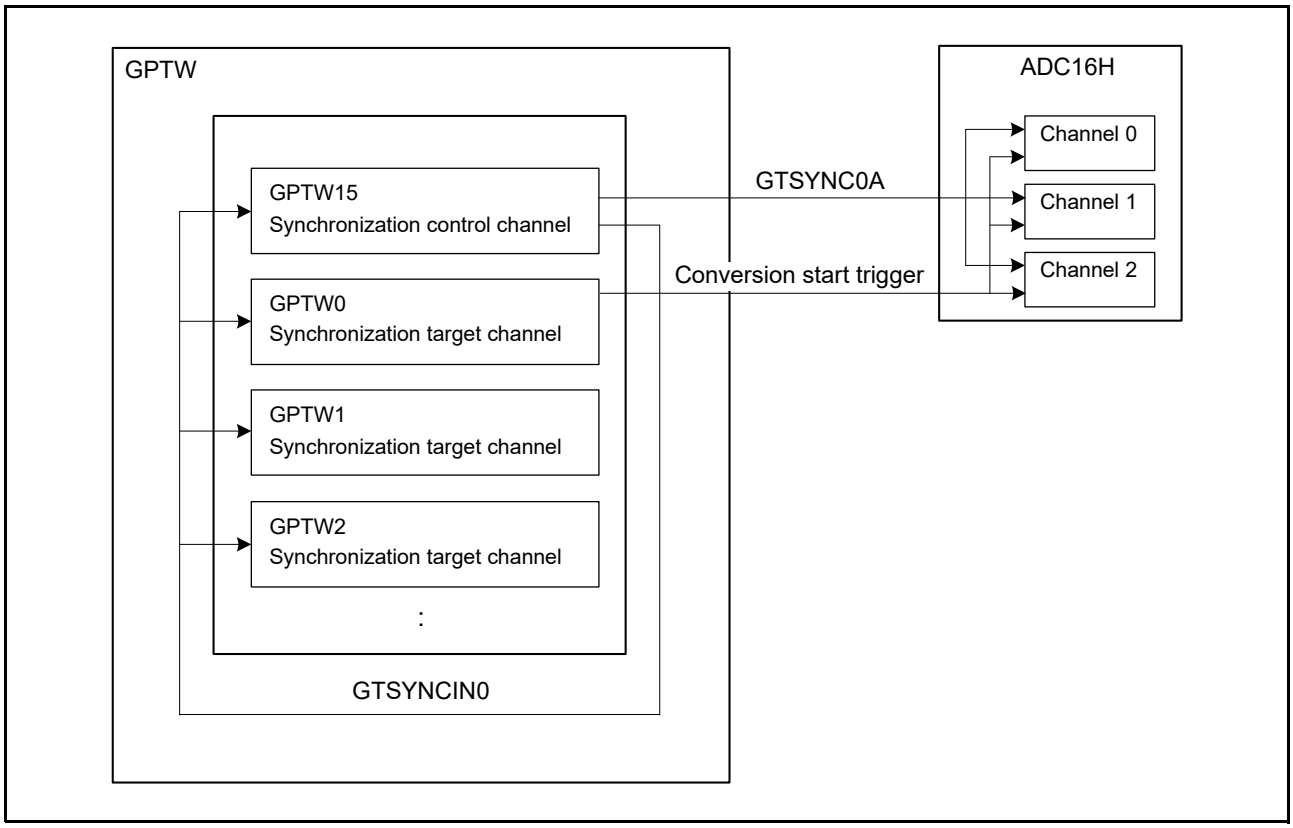


Figure 24.192 Configuration Examples of the Timer-Linked Synchronization Signals and the Timer-Internal Synchronization Signals

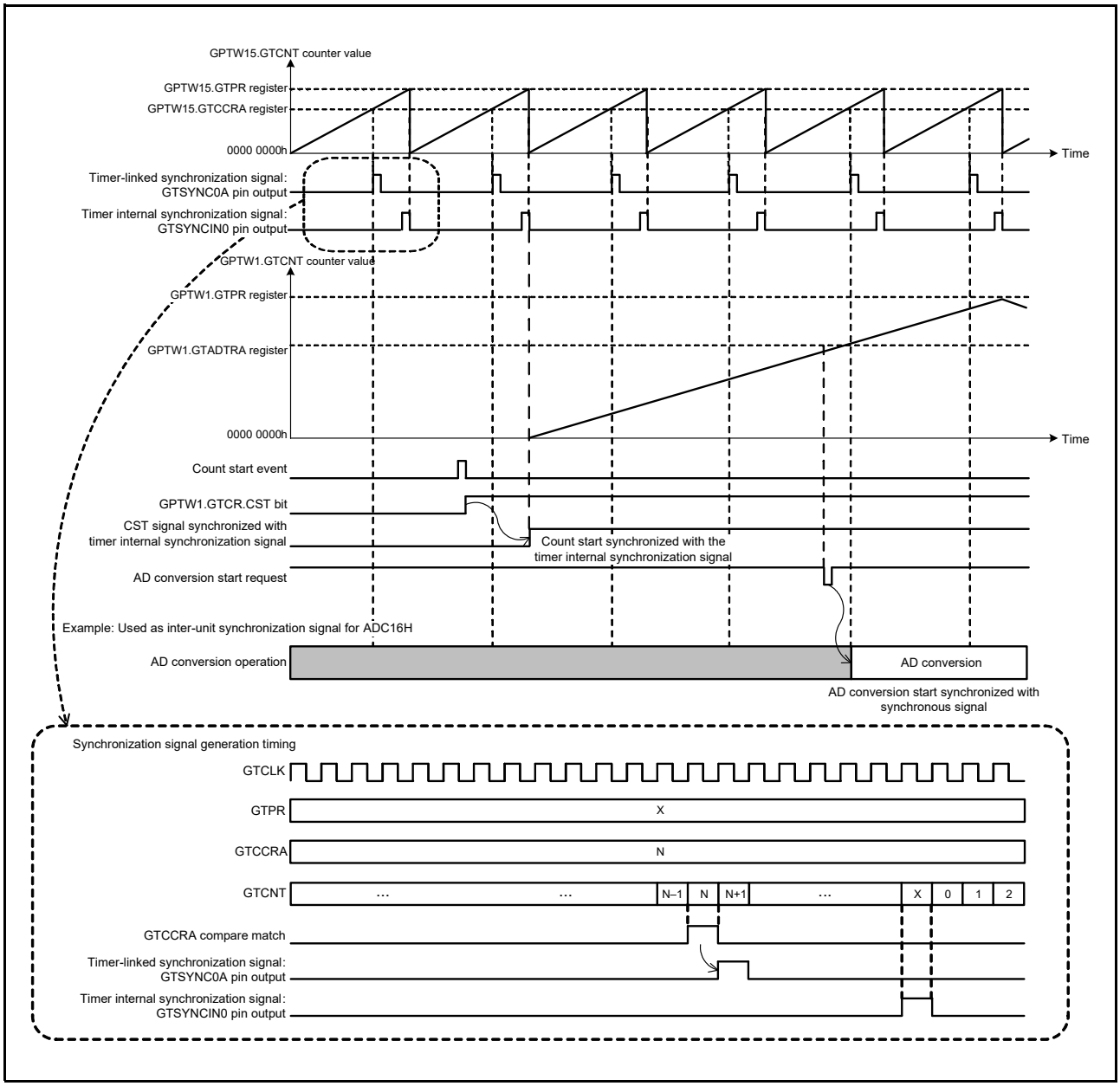


Figure 24.193 Configuration Examples of the Timer-Linked Synchronization Signals and the Timer-Internal Synchronization Signals

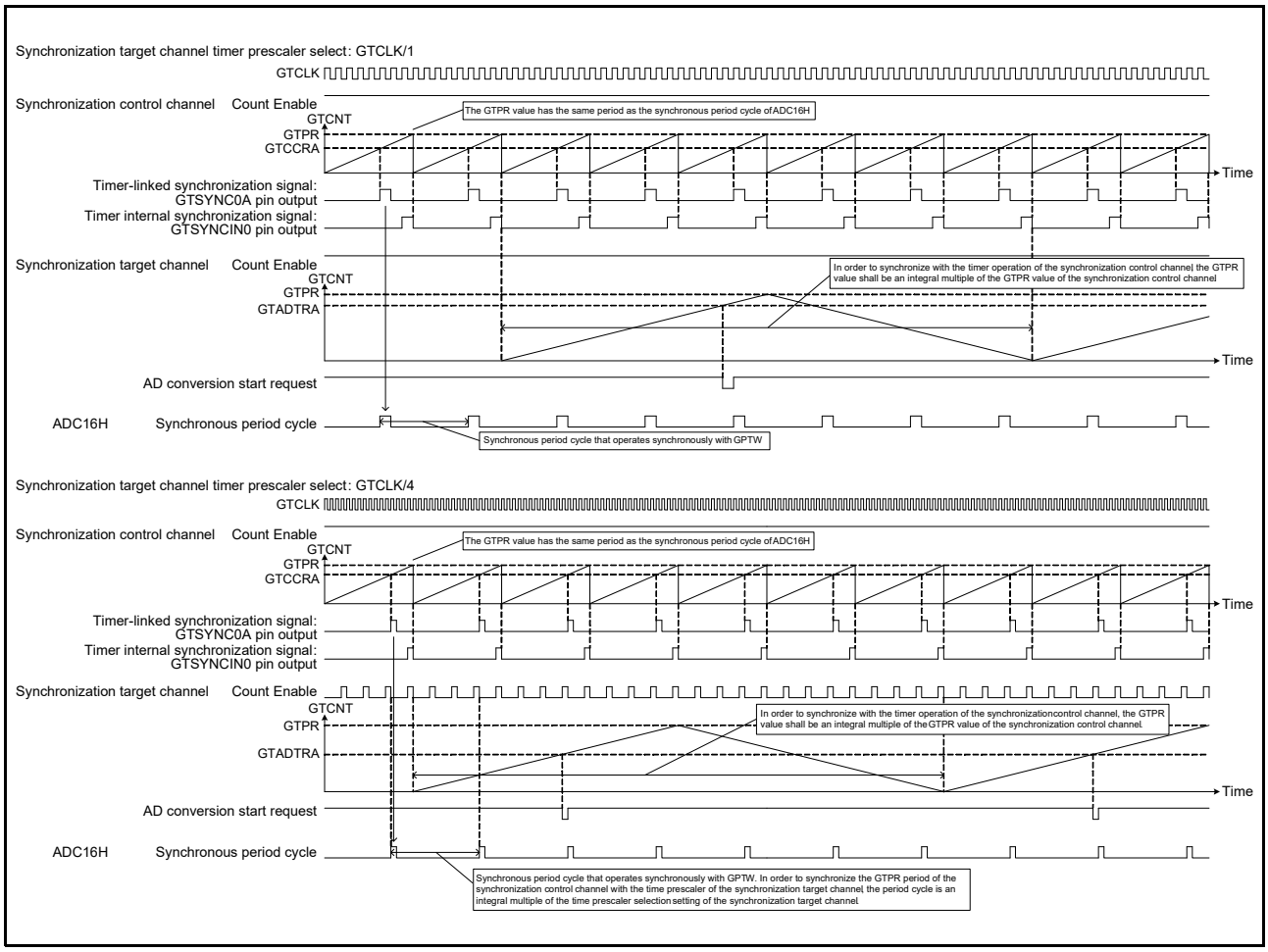


Figure 24.194 Example of the Relationship Between ADC16H, Synchronization Control Channel, and Synchronization Target Channel Count Cycles

24.4 Interrupt Sources

24.4.1 Interrupt Sources and Priorities

There are four kinds of interrupt sources; GTCCRm register (m = A to F) input capture/compare match GTCNT counter overflow (GTPR register compare match)/underflow, cycle count finish, and dead time error.
Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually.
When an interrupt source condition is satisfied, the corresponding status flag in the GTST register is set to 1, and an interrupt request is generated if the corresponding interrupt request enable or disable bit in the GTINTAD register is 1.
The corresponding status flag in the GTST register can be cleared by writing to it. If a flag set and a flag clear occur simultaneously, clearing has priority over setting.
For details, refer to section 15, Interrupt Controller (ICUM). Table 24.40 shows a list of GPTW interrupt sources.

Table 24.40 GPTW Interrupt Sources (n = 0 to 15)

Channel	Name	Interrupt Source	Interrupt Flag
GPTWn	GTCIAn	GTCCRA register input capture/compare match	GPTWn.GTST.TCFA
	GTCIBn	GTCCRB register input capture/compare match	GPTWn.GTST.TCFB
	GTCICn	GTCCRC register compare match	GPTWn.GTST.TCFC
	GTCIDn	GTCCRD register compare match	GPTWn.GTST.TCFD
	GDTEn	Dead time error	GPTWn.GTST.DTEF
	GTCEIn	Cycle count end	GPTWn.GTST.PCF
	GTCIEn	GTCCRE register compare match	GPTWn.GTST.TCFE
	GTCIFn	GTCCRF register compare match	GPTWn.GTST.TCFF
	GTCIVn	GTCNT counter overflow (GTPR register compare match)	GPTWn.GTST.TCFPO
	GTCIUn	GTCNT counter underflow	GPTWn.GTST.TCFPU

(1) GTCIAn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches the GTCCRA register.
- When the GTCCRA register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GTCIBn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches the GTCCRB register.In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GTCICn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches the GTCCRC register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GTCIDn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches the GTCCRD register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 10b, 11b (buffer operation with the GTCCRD register)

(5) GTCIEn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GTCIFn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 10b, 11b (buffer operation with the GTCCRF register)

(7) GTCIVn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at overflows (the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 2, interrupt requests are enabled at overflows (the GTCNT counter value changes

from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0000 0000h) or the time when GTCNT counter value matches GTPR register value.

- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT counter value changes from the GTPR register value to the GTPR register value minus 1).
- In complementary PWM mode, interrupt requests are enabled at crests (the GTCNT counter value of master channel changes from the GTPR register value to the GTPR register value minus 1).
- In counting driven by a hardware source, including counting for the external pulse width measuring function, interrupt requests are enabled on overflows (that is, when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).

(8) GTCIUn Interrupt (n = 0 to 15)

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register to 0 during down-counting).
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT counter value changes from 0 to 1).
- In complementary PWM mode, interrupt requests are enabled at troughs (the GTCNT counter value of master channel changes from 0 to 1).
- In count operation by a hardware source (include external pulse width measuring function), interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).

(9) GDTEn Interrupt (n = 0 to 15)

When automatic dead time setting has been made, the GTST.DTEF flag becomes 1 when the timer output toggle point with dead time added exceeds the count period. If the GTINTAD.GRPDTE bit is 1 at this time, a dead time error interrupt request (GDTE) is generated.
In addition, when the timer output toggle point with dead time added is back within the count period, the GTST.DTEF flag changes from 1 to 0.

(10) GTCEIn Interrupt (n = 0 to 15)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, and the end of the cycle is reached, if the GTINTAD.GTINTPC bit is 1, a cycle count end interrupt (GTCEI) request is generated.

Table 24.41 shows relationships between interrupt signals and interrupt enable bits.

Table 24.41 Relationship between Interrupt Signals and Interrupt Enable Bits (n = 0 to 15)

Interrupt Signal	Interrupt Enable Bit	Status Flag
GTCIAn	GTINTAD.GTINTA bit	GTST.TCFA Flag
GTCIBn	GTINTAD.GTINTB bit	GTST.TCFB Flag
GTCICn	GTINTAD.GTINTC bit	GTST.TCFC Flag
GTCIDn	GTINTAD.GTINTD bit	GTST.TCFD Flag
GDTEn	GTINTAD.GRPDTE	GTST.DTEF Flag
GTCEIn	GTINTAD.GTINTPC	GTST.PCF Flag
GTCIEn	GTINTAD.GTINTE bit	GTST.TCFE Flag
GTCIFn	GTINTAD.GTINTF bit	GTST.TCFF Flag
GTCIVn	GTINTAD.GTINTPR[1:0] bits	GTST.TCFPO Flag
GTCIUn		GTST.TCFPU Flag

24.4.2 DMAC/DTC Activation

The DMAC and DTC can be triggered by the interrupt request in each channel. For details, see section 15, Interrupt Controller (ICUM), section 18, DMA Controller (DMACAa), section 20, Data Transfer Controller (DTCA).

24.4.3 Interrupt and A/D Conversion Start Request Skipping Function

24.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GTCIV) and underflow interrupt (GTCIU) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GTCIV/GTCIU skipping function. However, the dead time error interrupts cannot be linked with the GTCIV/GTCIU skipping function.

When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt enable bit is used only to control the output of the interrupt signal after thinning out.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GTCIV/GTCIU interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GTCIV/GTCIU interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in sawtooth-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GTCIV/GTCIU interrupt requests cannot be generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GTCIV/GTCIU interrupts at overflows only or underflows only in sawtooth-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 24.195 to Figure 24.200 show examples of skipping function operation.

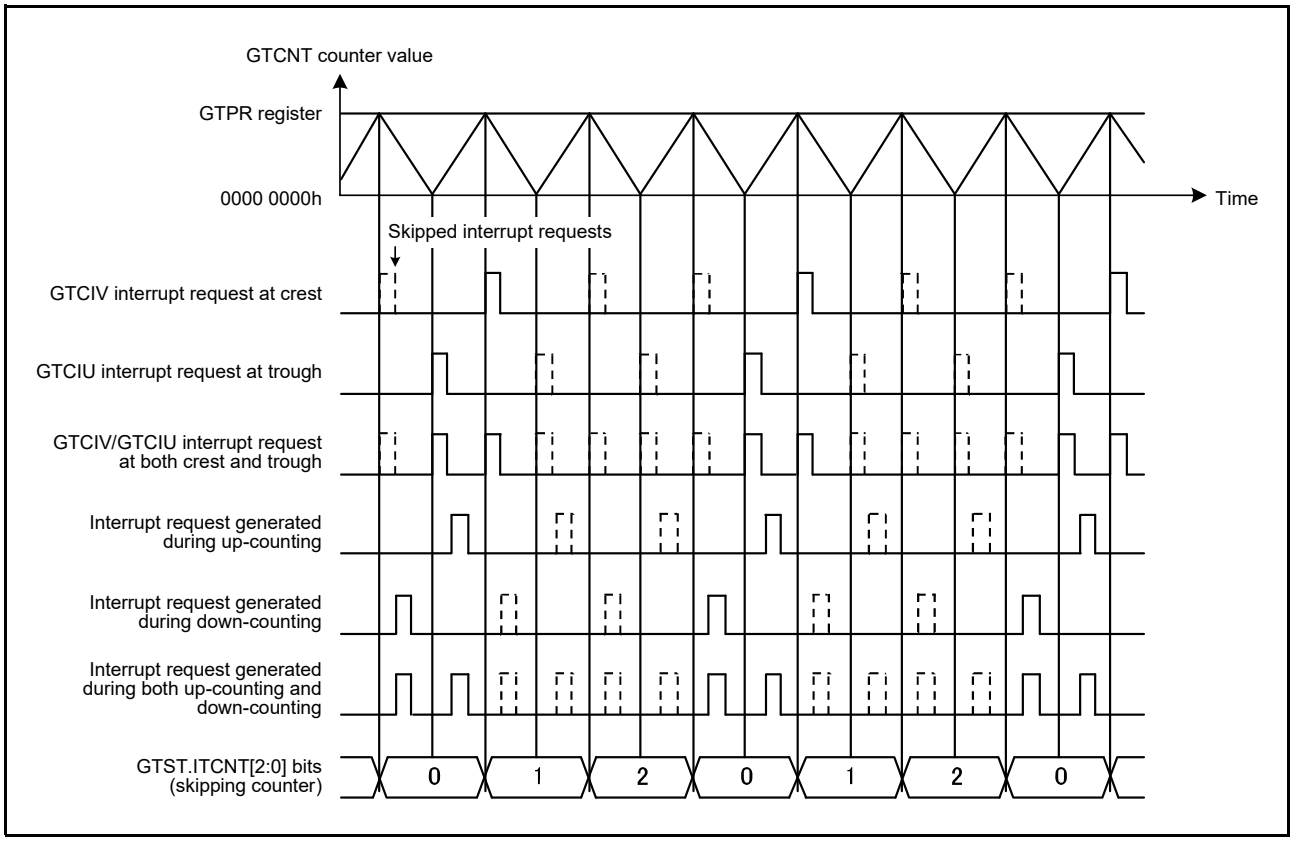


Figure 24.195 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

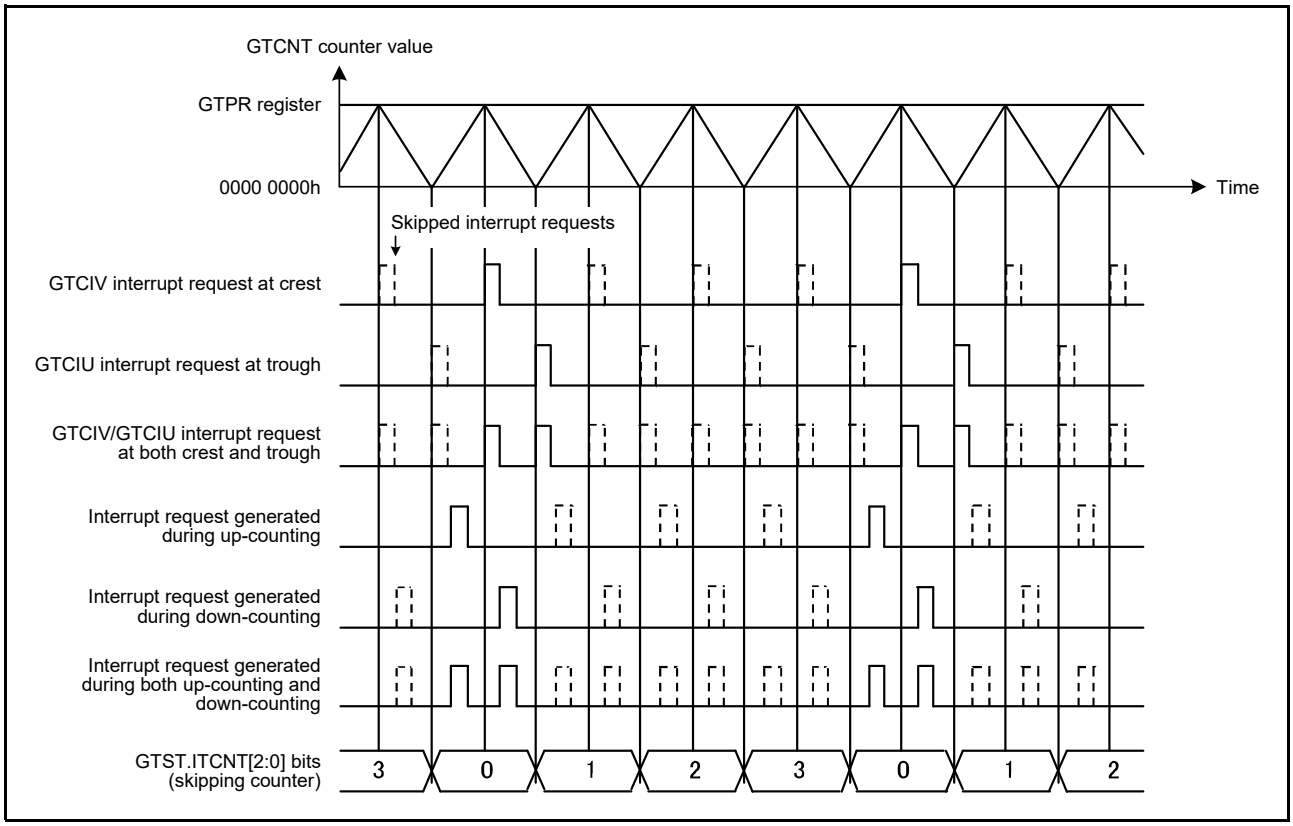


Figure 24.196 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

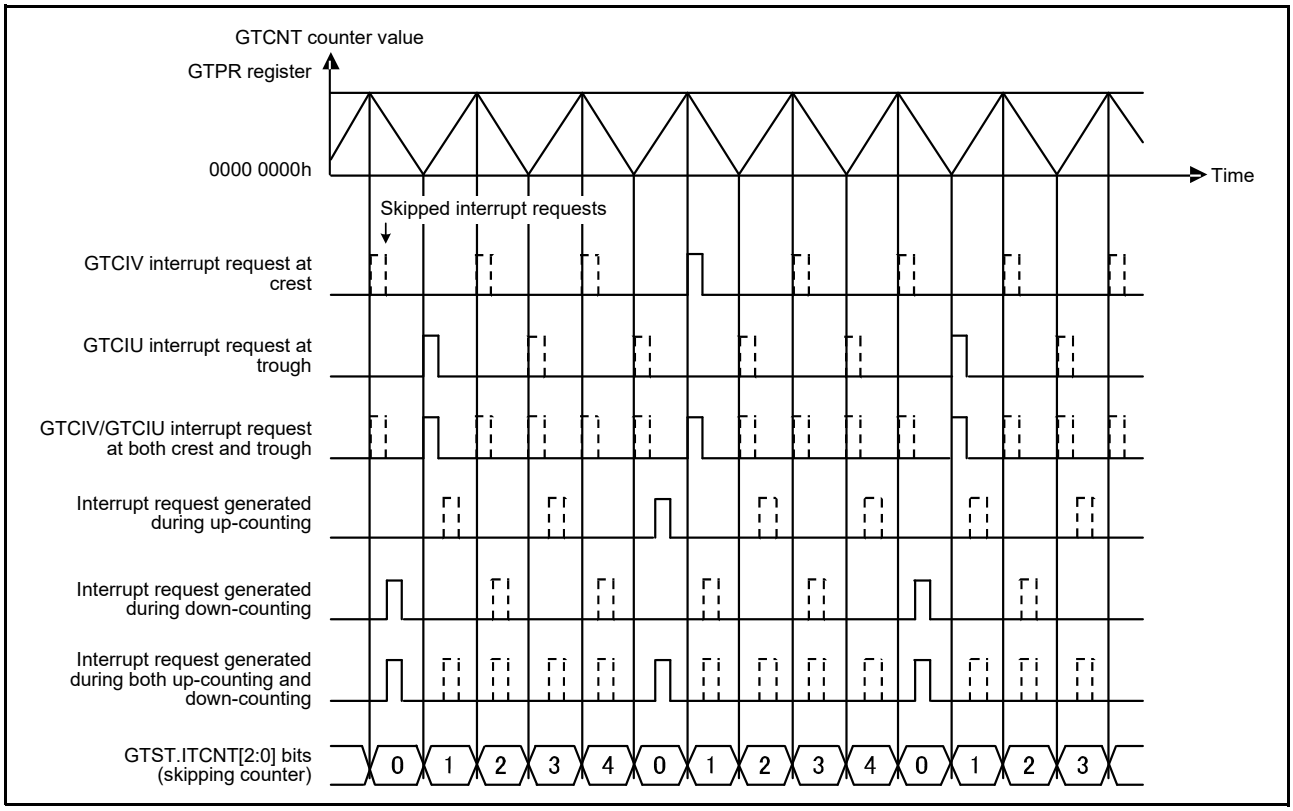


Figure 24.197 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

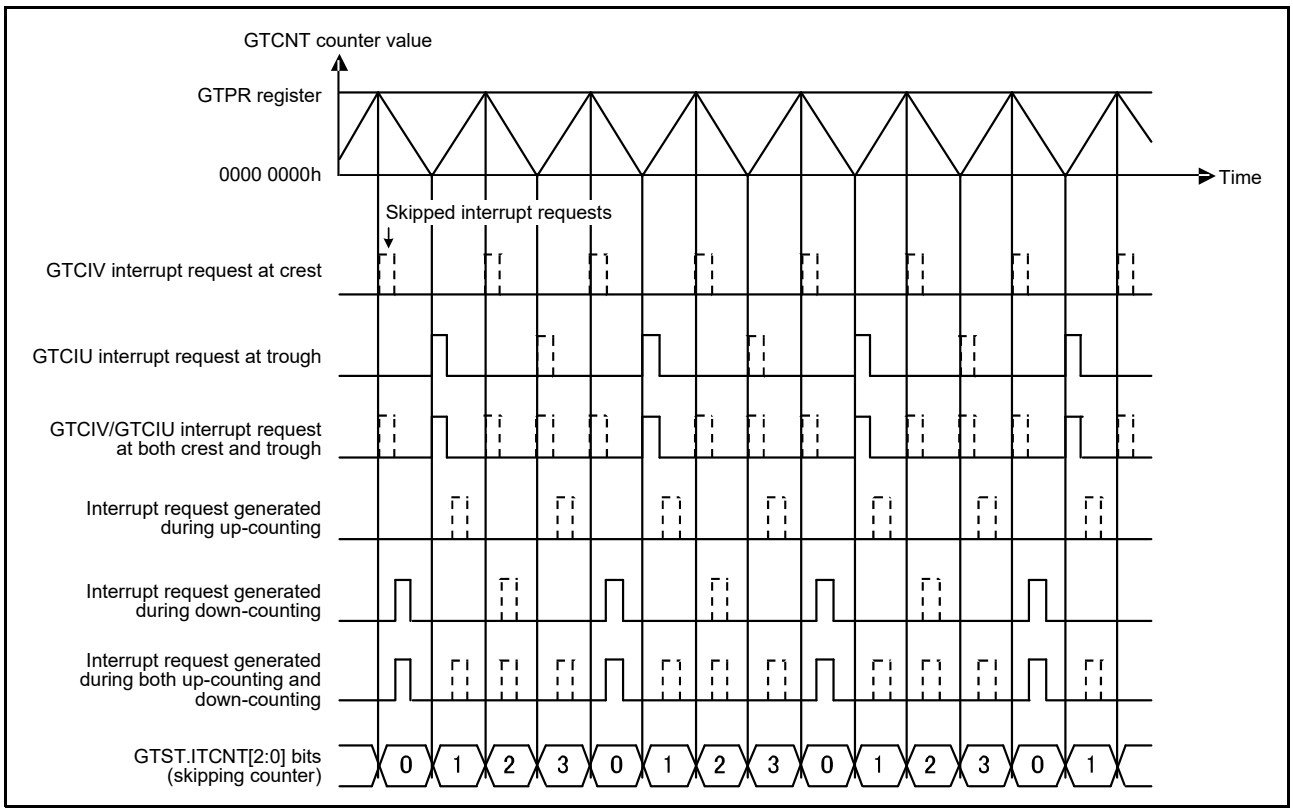


Figure 24.198 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

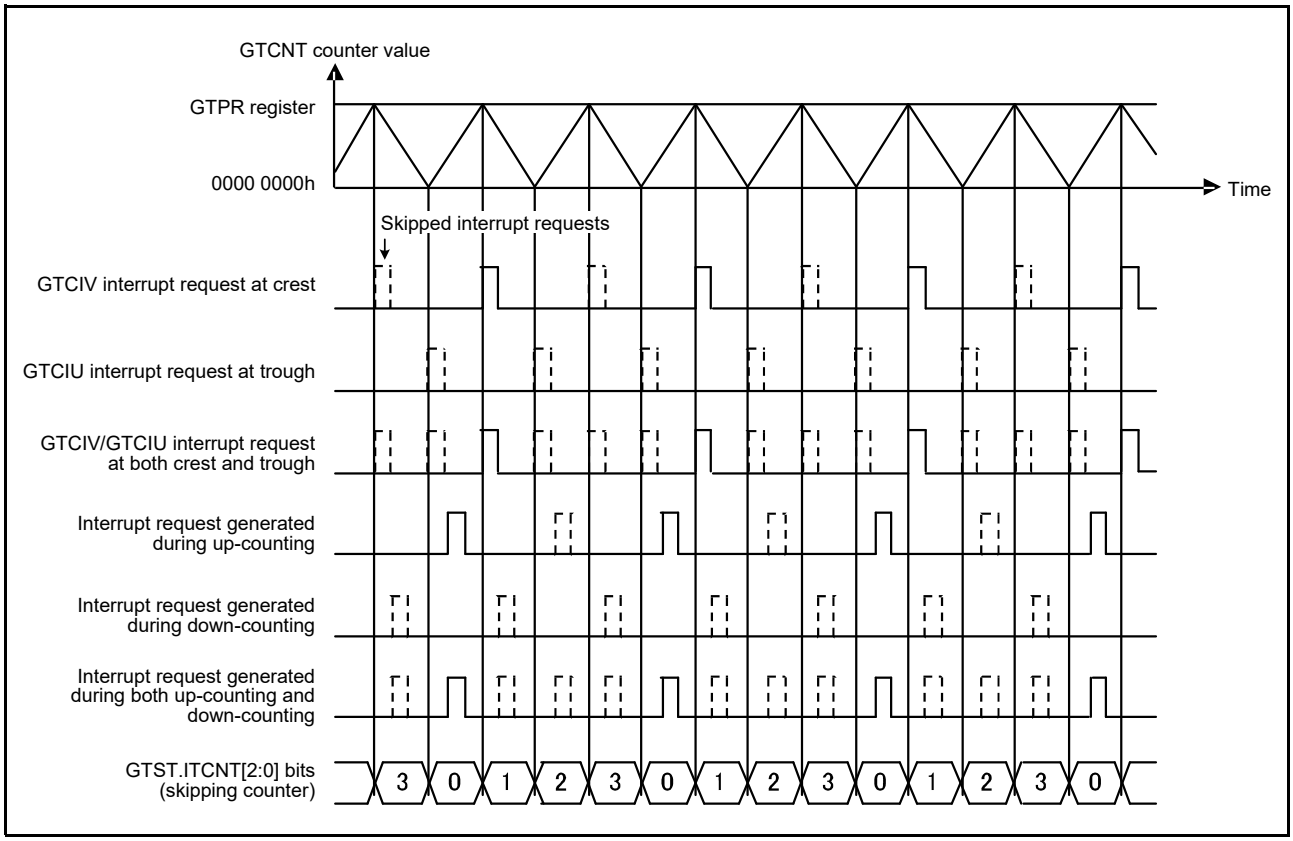


Figure 24.199 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

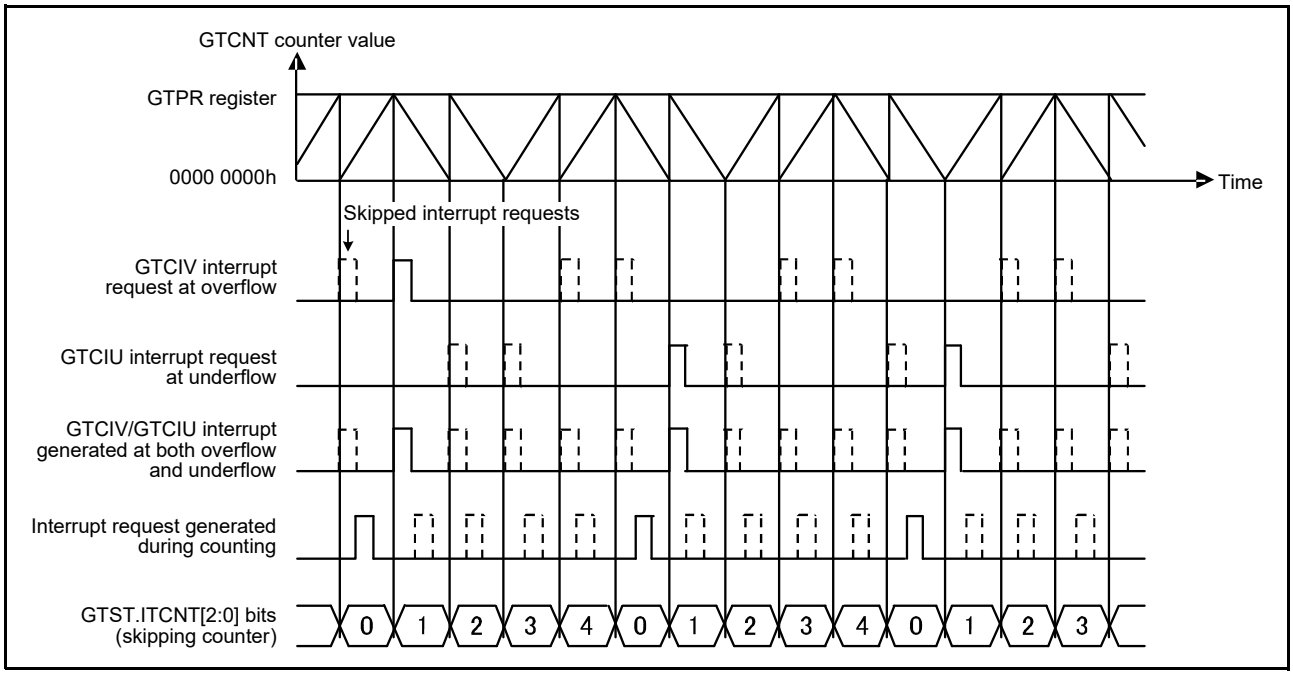


Figure 24.200 Example of Interrupt Skipping Function Operation (Sawtooth Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

24.4.3.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2 and GTEITLB registers. The dead time error interrupt cannot be skipped.

The skipping enable setting and the skipping period can be individually configured using the GTEITLI1, GTEITLI2, and GTEITLB registers for overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer.

The skipping period is configured based on the values of the two independent extended interrupt skipping counters (the EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers). It can be specified during the period in which either skipping counter value is other than 0 or other than the skipping count. It can also be specified during the period in which both skipping counters are other than 0 or other than the skipping count.

Figure 24.201 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping.

The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits count the count source (in Figure 24.201, a crest is selected) selected by the extended interrupt counter 1 count source select bit (the EIVTC1[1:0] bits) from the initial value of 0. It returns to 0 when the counter reaches the skipping count (in Figure 24.201, count is 2) set by the extended interrupt skipping 1 skipping count setting bit (the EIVTT1[3:0] bits), and then repeats the counting operation.

The EITCNT2[3:0] bits can set the initial value, count the count source (in Figure 24.201, a trough is selected) selected by the extended interrupt counter 2 count source select bit (the EIVTC2[1:0] bits), and repeat the count operation which return to 0 when the skipping count (in Figure 24.201, count is 2) set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits and when the written value to the EIVTC2[1:0] bits is other than 00b, while the setting for the extended interrupt skipping counter 2 is not in counting (the EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the extended interrupt skipping counter 2 initial value bit (the EITCNT2IV[3:0] bits) is set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (the GTST.ITCNT[2:0] bits) by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits which are an extended interrupt skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bits or the EIVTC2[1:0] bits to 00b).

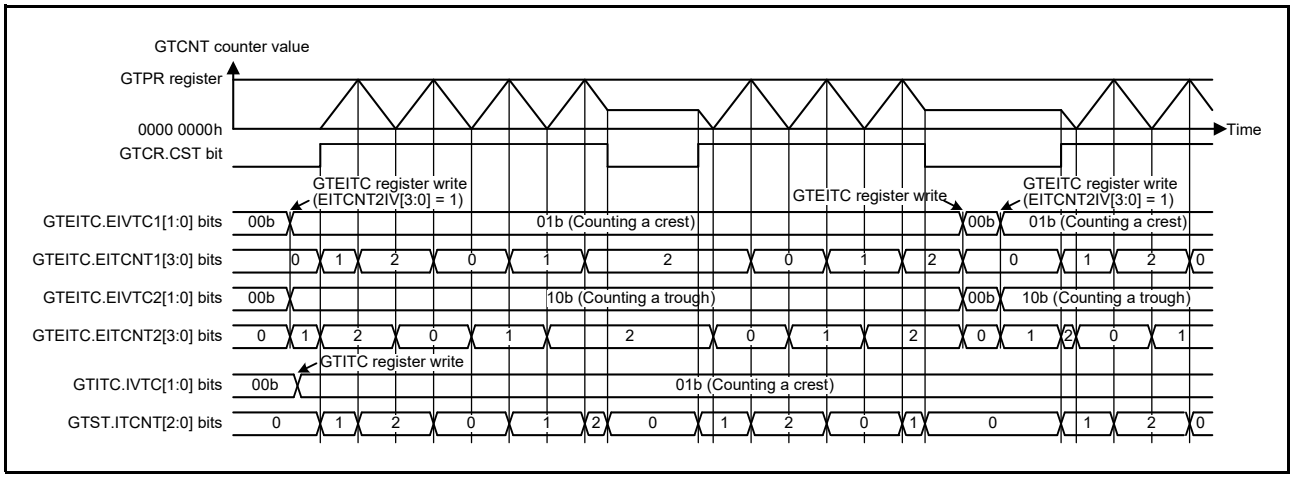


Figure 24.201 Counter Operation for Interrupt Skipping

Interrupt skipping by the GTEITLI1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with interrupt skipping by the GTITC register or GTADCMSC register. A skipping period at this time is represented by ORed skipping periods of respective registers.
Figure 24.202 shows an example of simultaneous interrupt skipping operations by different registers.

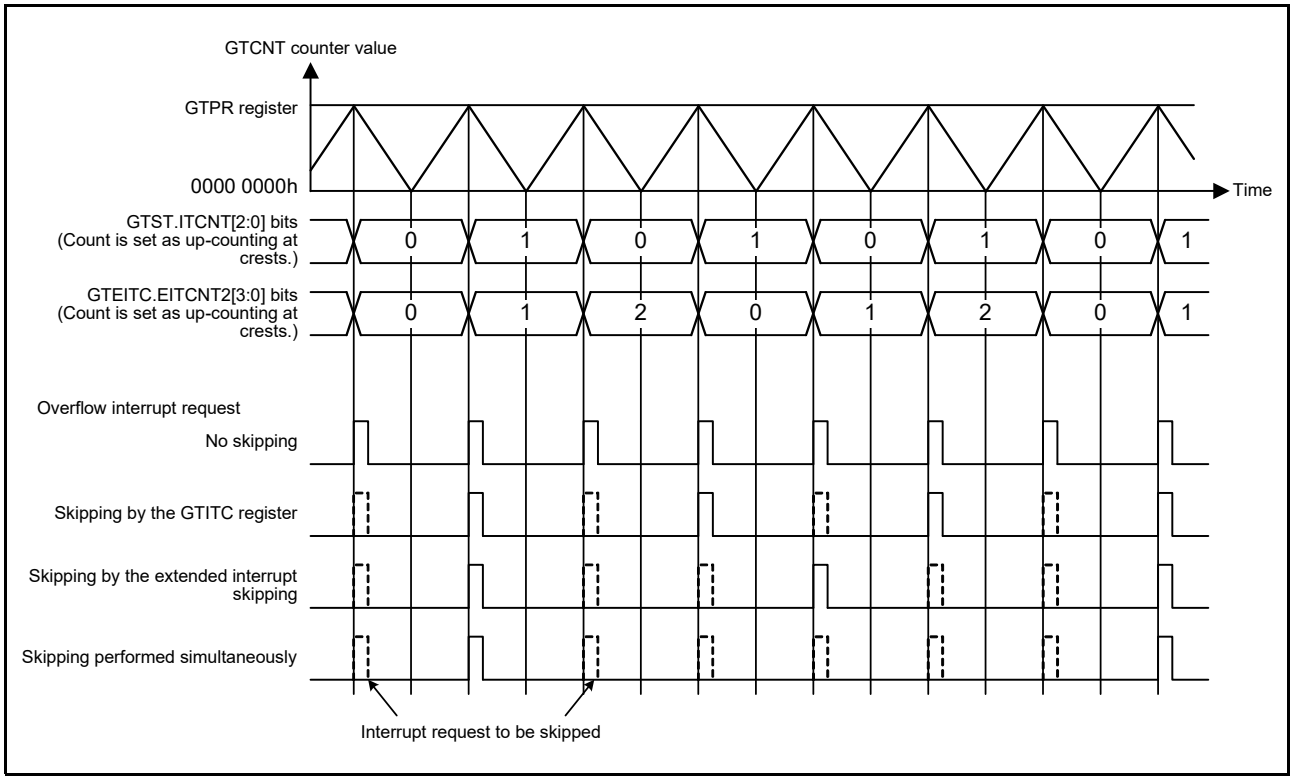


Figure 24.202 Example of Interrupt Skipping Operation (Skipping by the GTITC Register: Counting Crests, Extended Interrupt Skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

When the extended interrupt skipping selected by the GTEITLI1 register is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.
When an A/D conversion start request is skipped by the GTEITLI2 register, skipping of the corresponding status flag update and ELC event source outputs is controlled by the GTITC register and the extended interrupt skipping register, and the setting of A/D conversion start request enable bit in the GTINTAD register has no effect. The A/D conversion start request enable bit is used only for A/D conversion start request output (ELC event source output) after skipping.
Skipping of buffer transfer by the GTEITLB register applies to buffer operations enabled by the GTBER, GTDTCR, and GTBER2 registers, as well as all buffer operations performed by sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or complementary PWM mode (except for buffer transfer from GTCCRC, GTCCRE to GTCCRA).
Interrupt skipping and buffer transfer skipping operate independently. For a buffer transfer accompanied by an interrupt output, it is possible to output an interrupt without performing a buffer transfer, or to perform a buffer transfer without outputting an interrupt.
Figure 24.203 to Figure 24.210 show examples of skipping function operation.

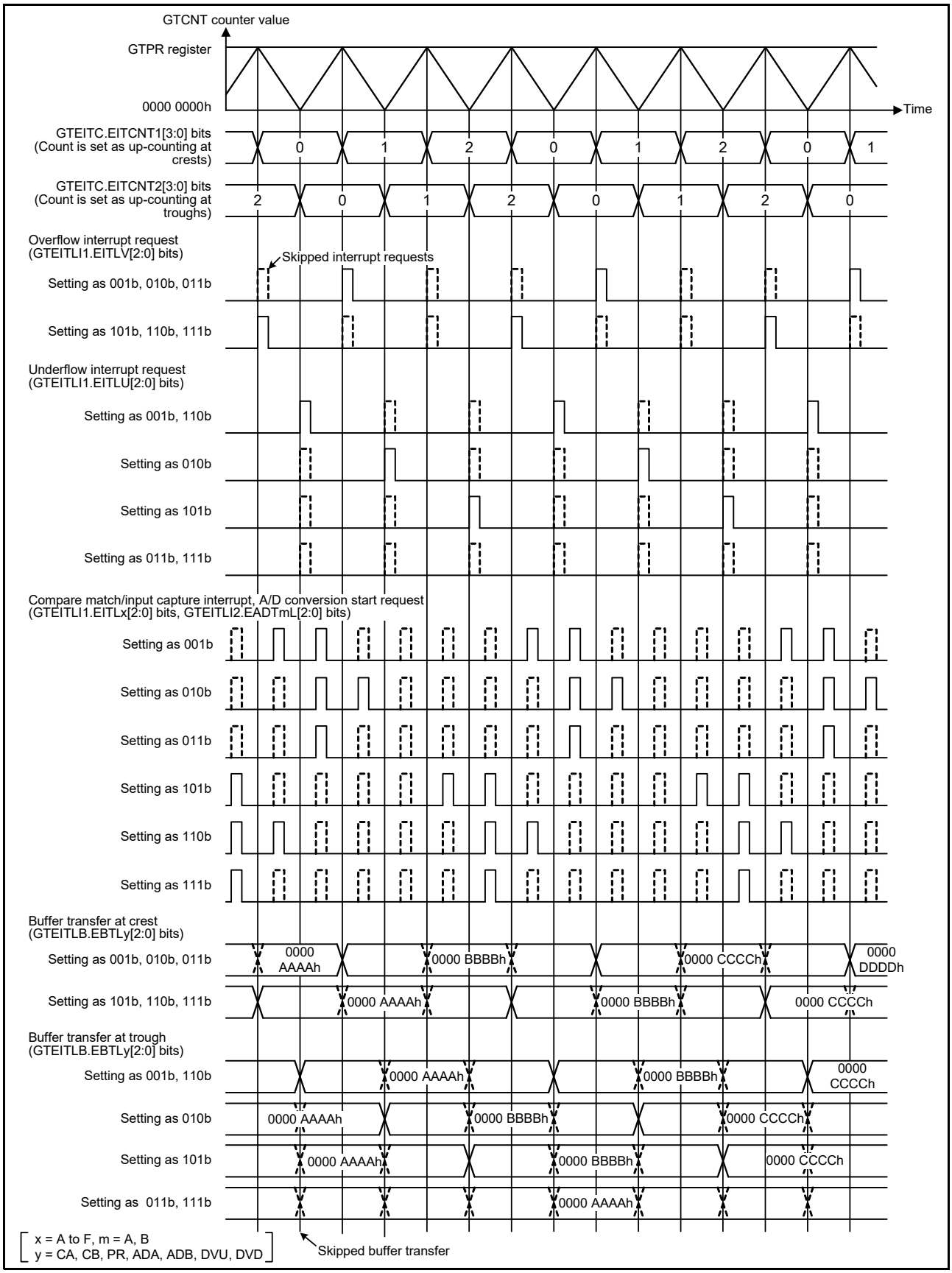


Figure 24.203 Example of Extended Interrupt Skipping Function (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Troughs, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0)

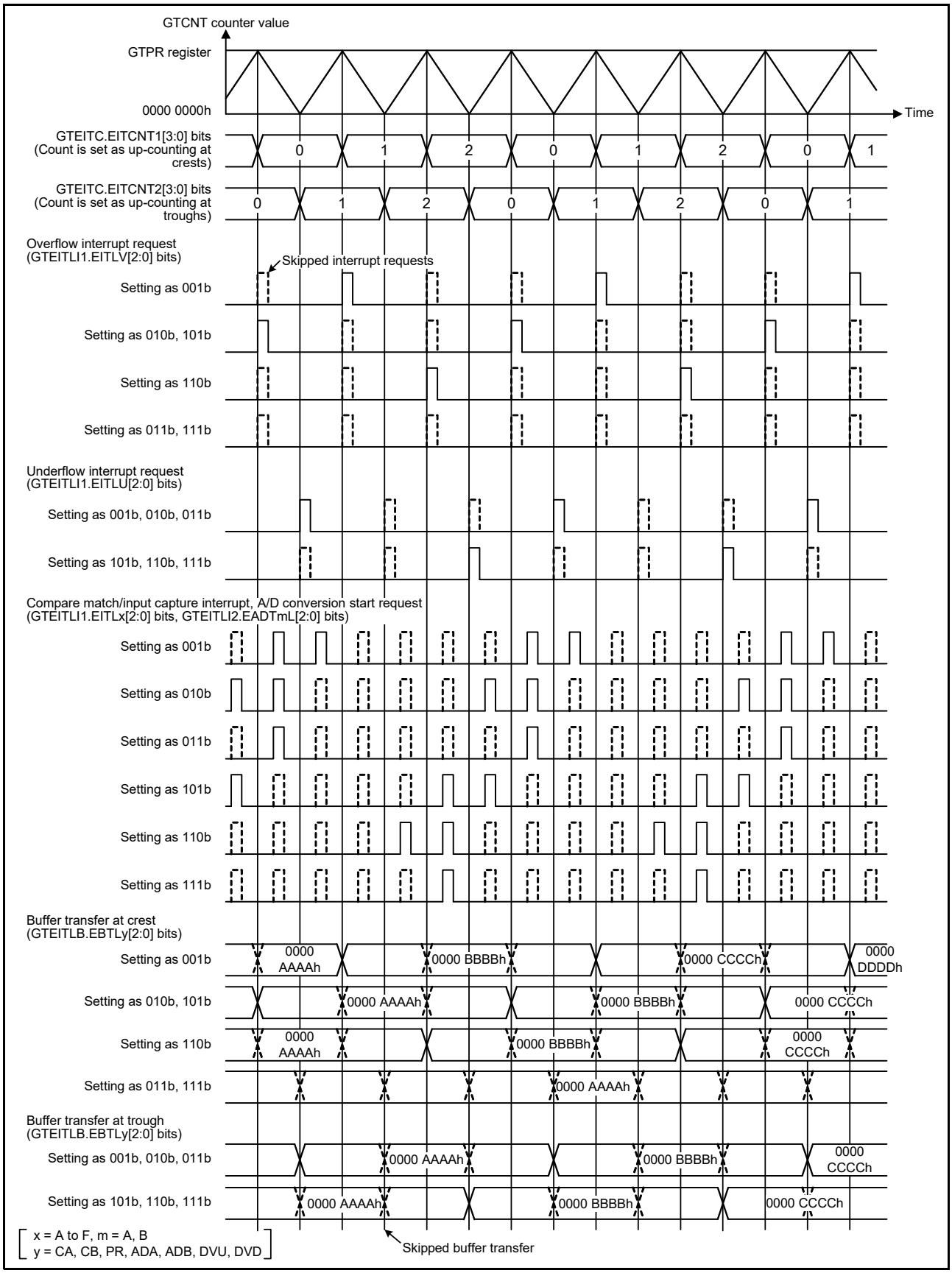


Figure 24.204 Example of Extended Interrupt Skipping Function (Triangle Waves, Count Crests by the Extended Interrupt Skipping 1 with Skipping Count: 2, Count Troughs by the Extended Interrupt Skipping 2 with Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 1)

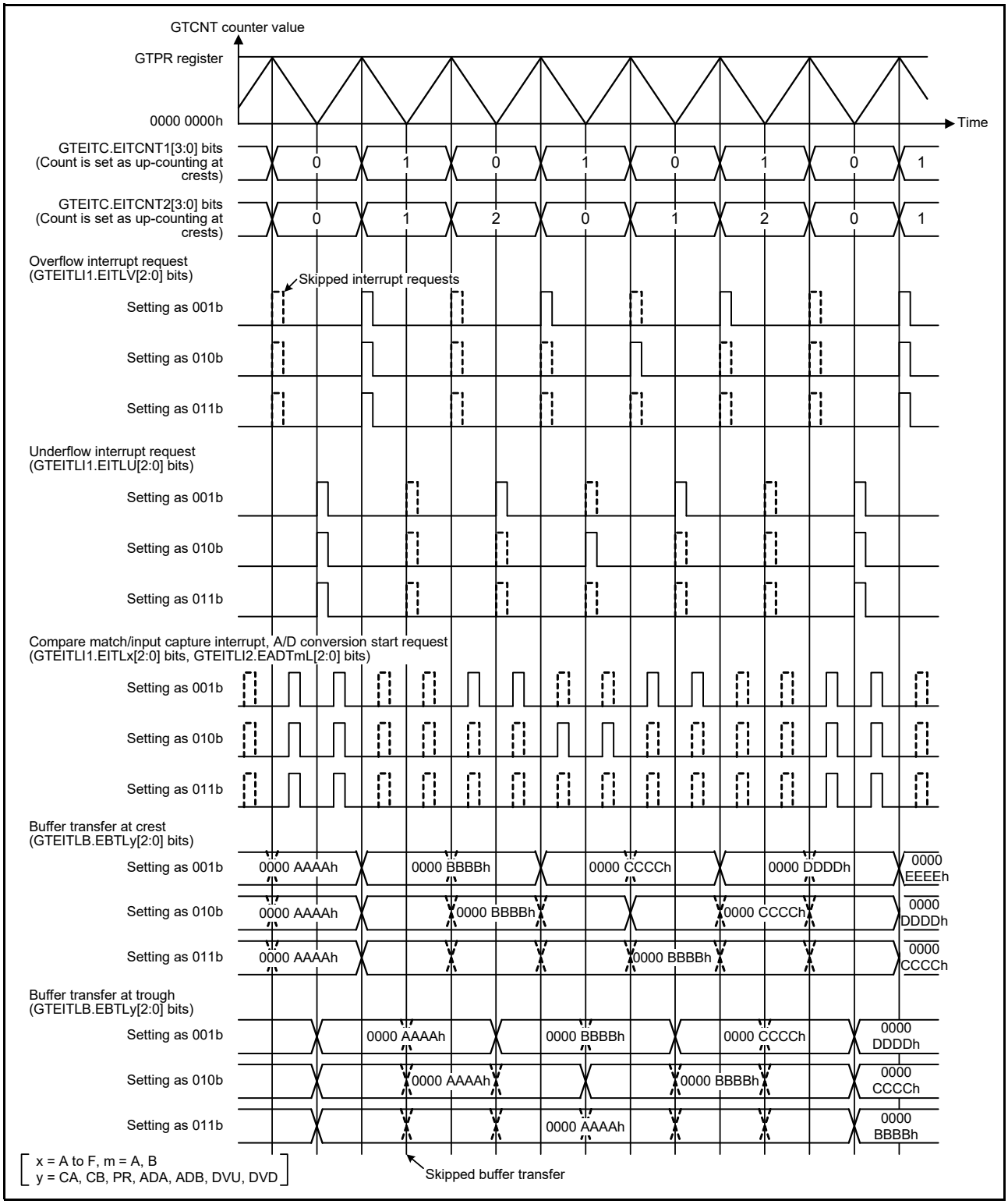


Figure 24.205 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

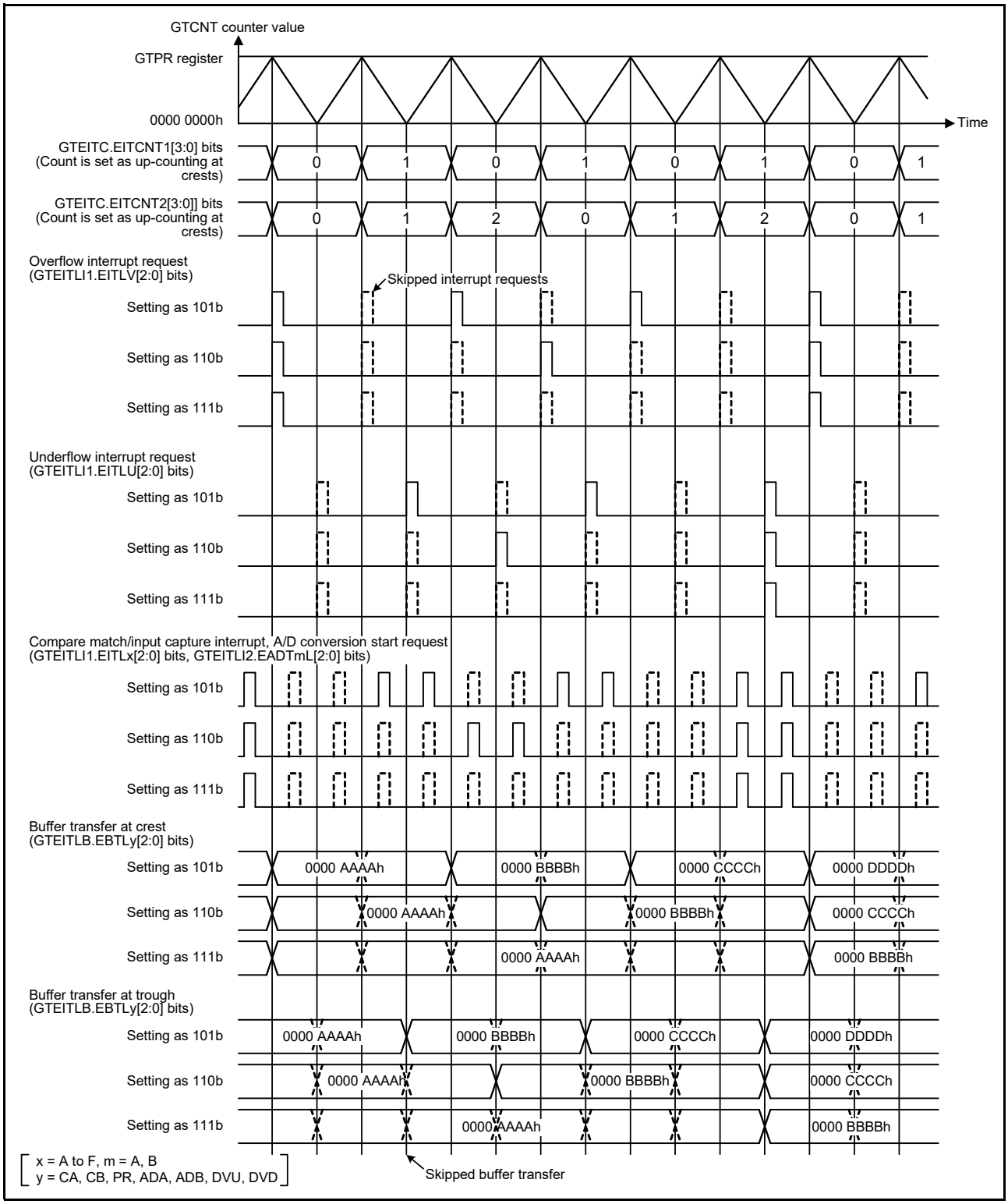


Figure 24.206 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

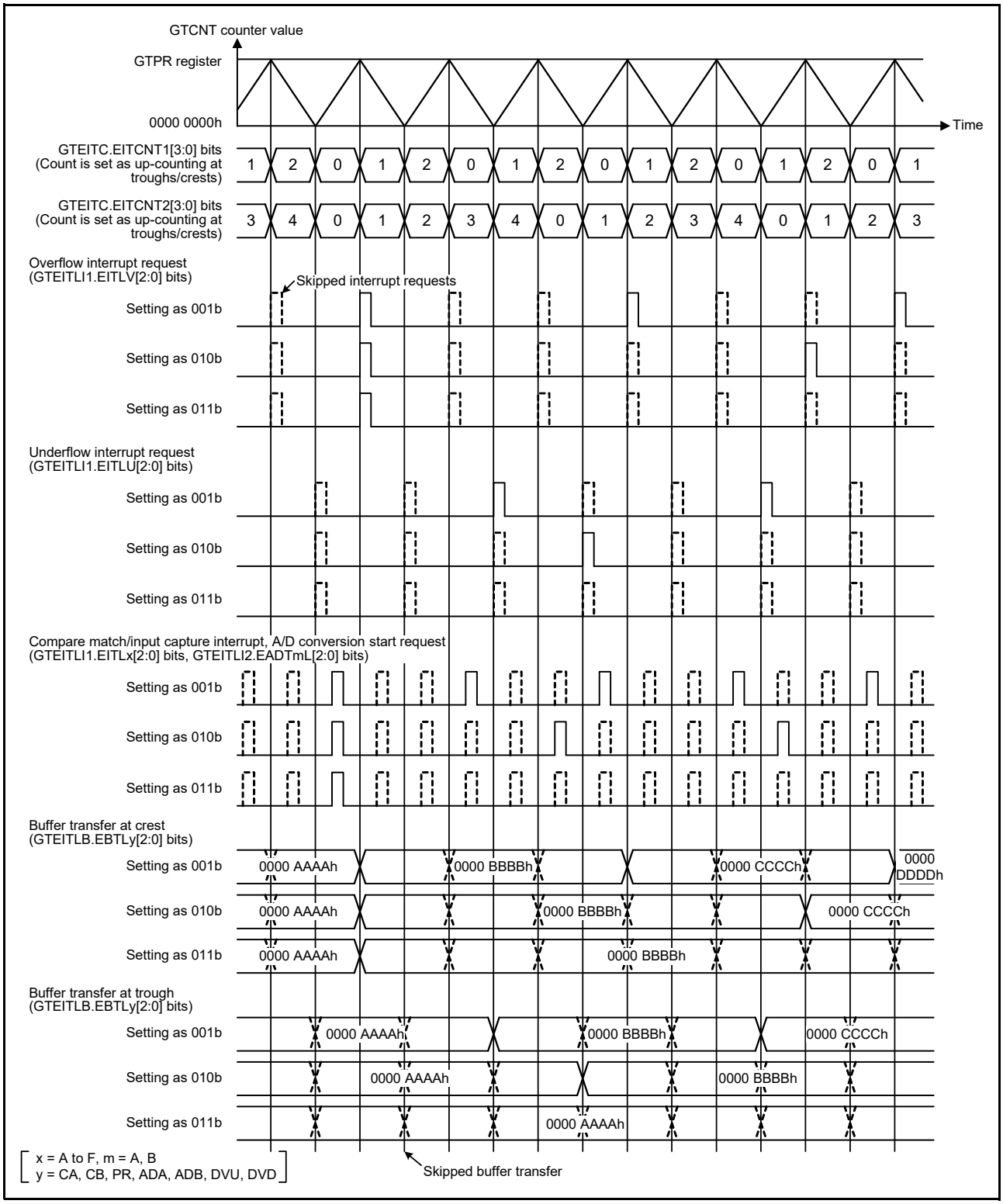


Figure 24.207 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

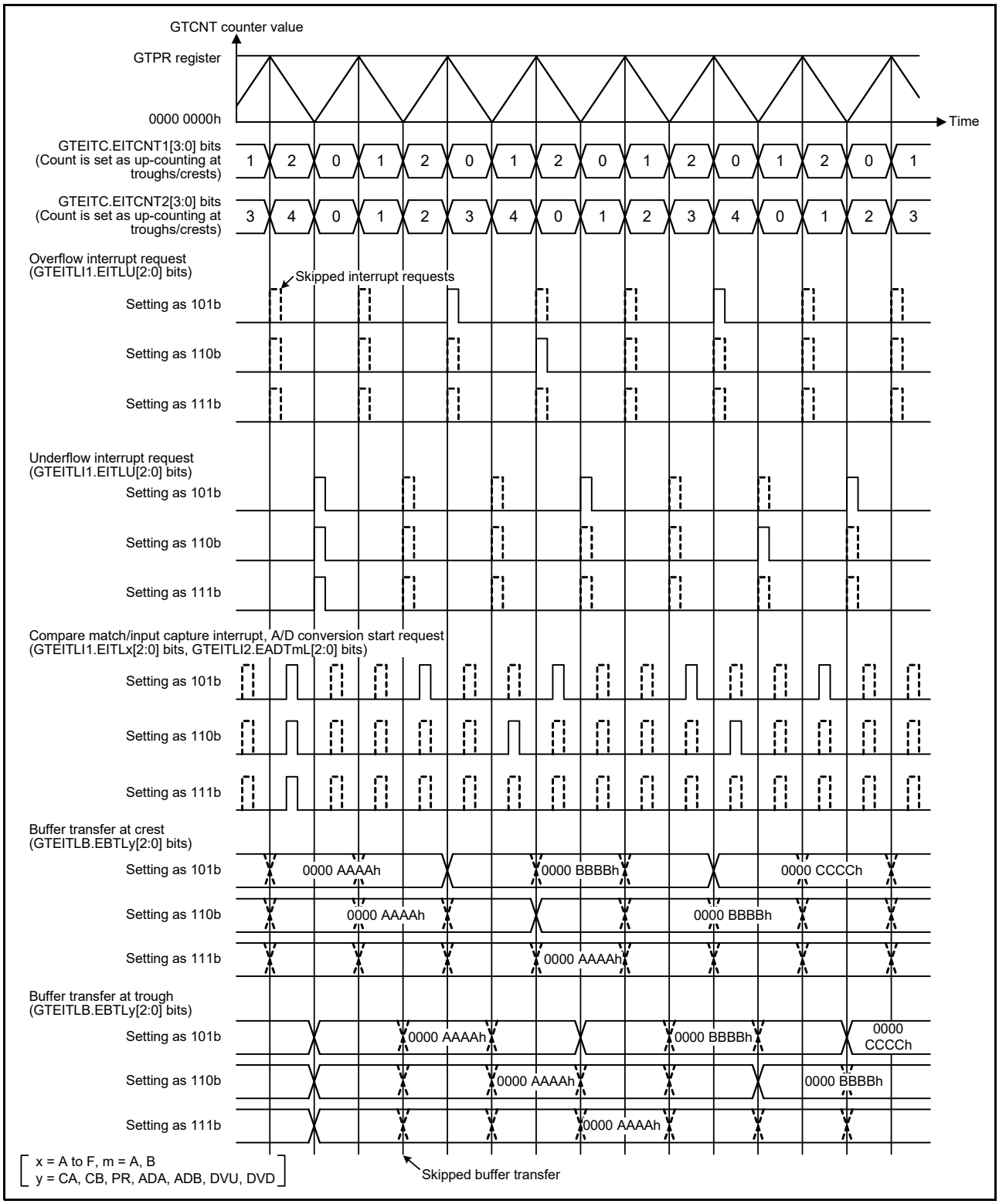


Figure 24.208 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

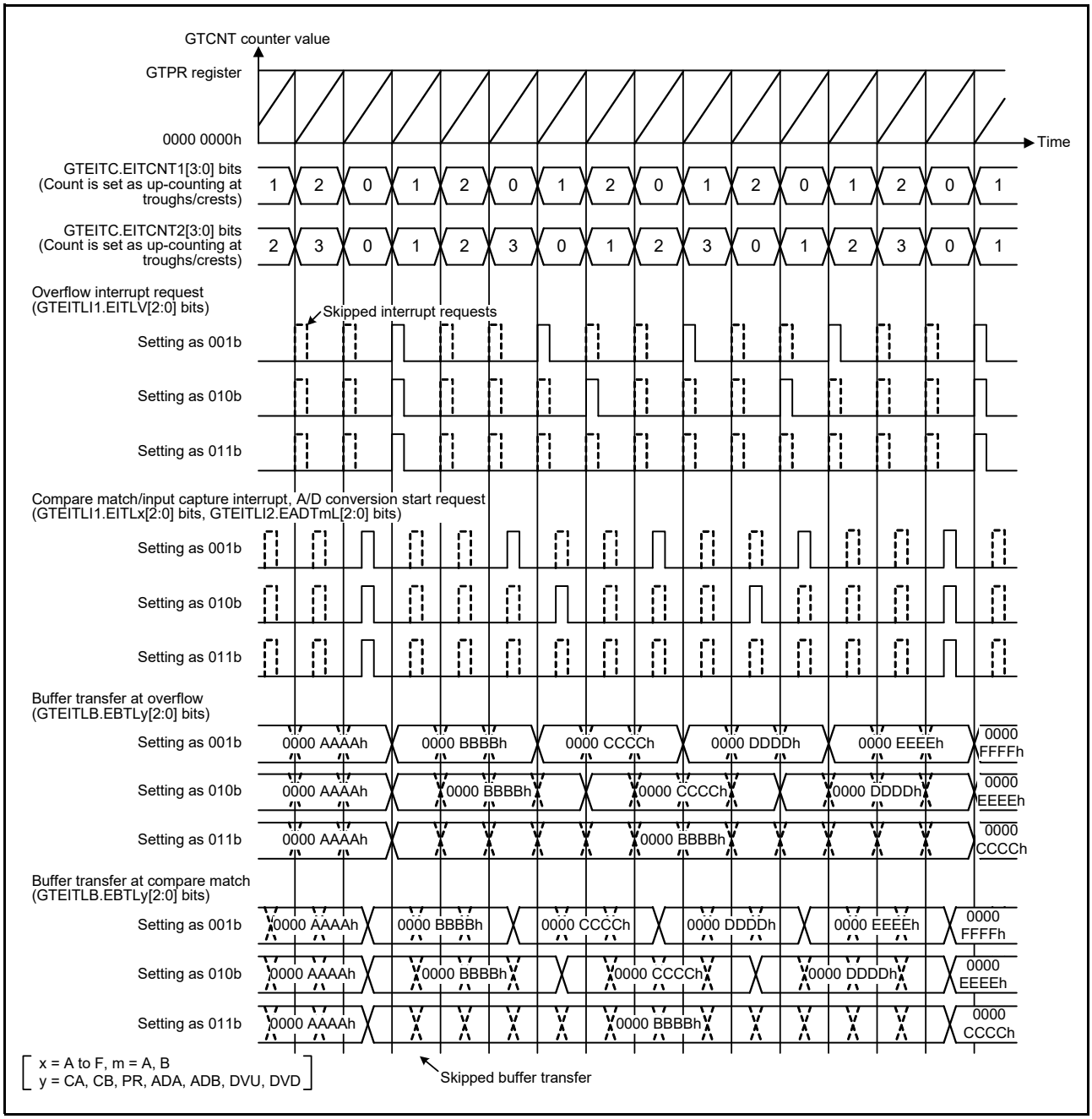


Figure 24.209 Example of Extended Interrupt Skipping Function Operation (Up-Counting in Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

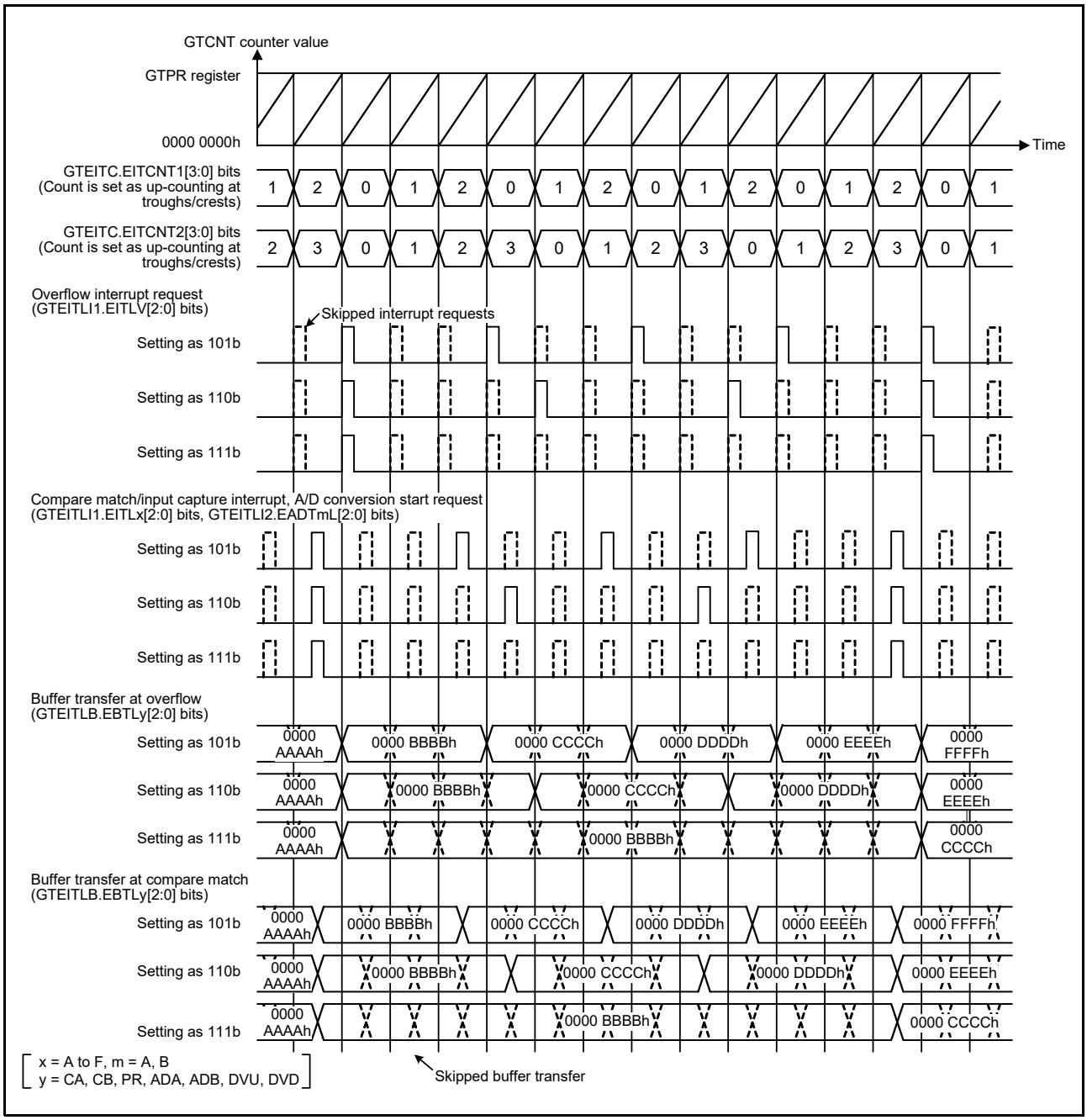


Figure 24.210 Example of Extended Interrupt Skipping Function Operation (Up-Counting Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

Figure 24.211 shows an example of the extended interrupt skipping operation on the input capture. When the setting is for the input capture operation setting (GTCR.ICDS bit = 0) at count stop of the GTCNT counter, the interrupt by the input capture and the extended skipping on buffer transfer is enabled even at count stop of the GTCNT counter.
When the input capture is generated at count stop of the GTCNT counter by setting the ICDS bit as 0, an interrupt and a buffer transfer are skipped if the skipping counter value is the same with the skipping period set in a corresponding interrupt function select bit.

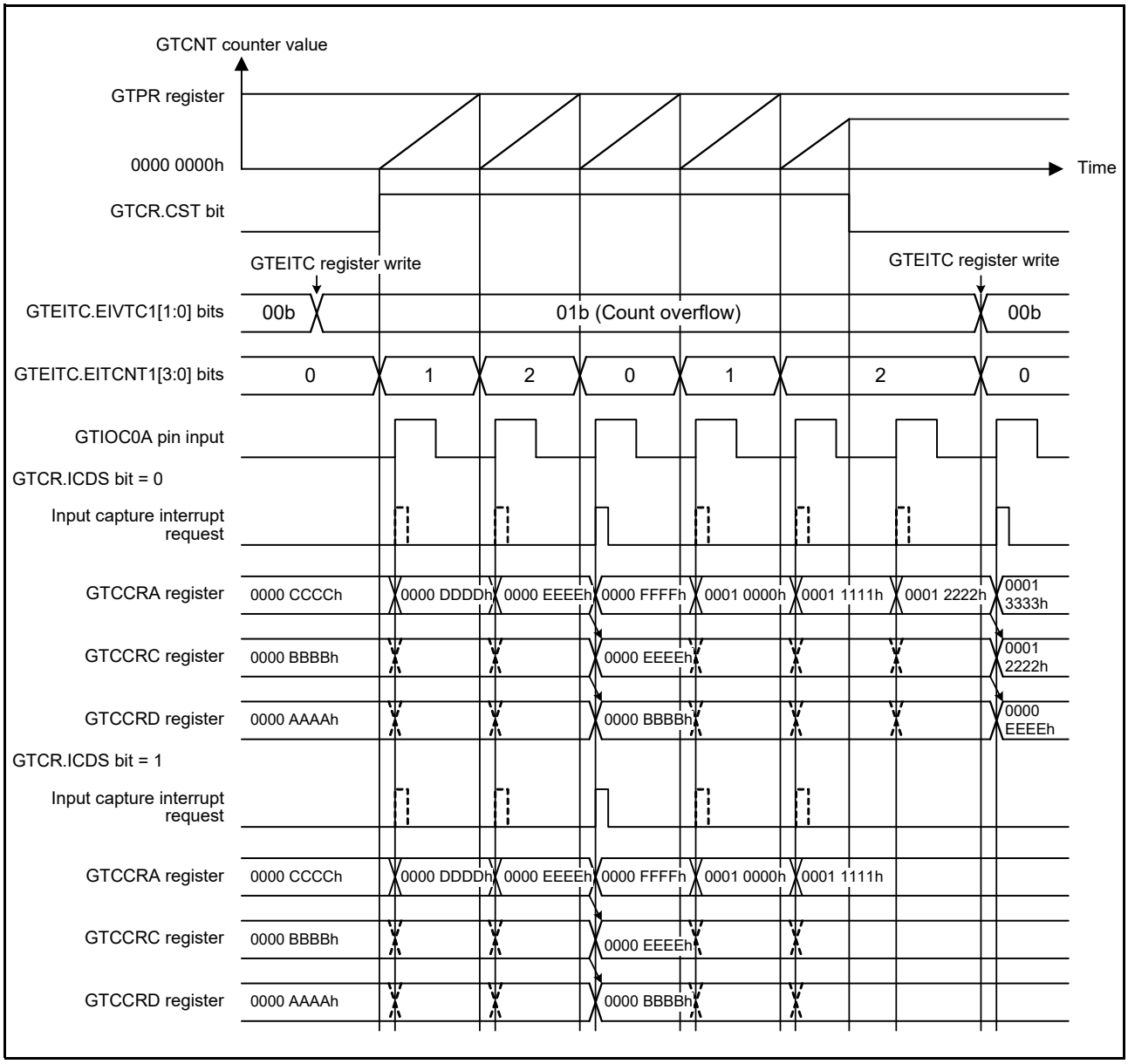


Figure 24.211 Example of Input Capture Operation in the Extended Interrupt Skipping Function Operation (Up-Counting Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Skipped in the Period other than the EITCNT1 as 0, Input Capture at the Input Rising)

Figure 24.212 shows an example for setting the extended interrupt skipping.
The initial value of the extended interrupt skipping counter 2 is set by the value written to the GTEITC.ETICNT2IV[3:0] bits. The initial value setting is applied when the setting of the extended interrupt skipping counter source is changed from “not counting” (GTEITC.EIVTC2[1:0] bits = 00b) to “counting” (GTEITC.EIVTC2[1:0] bits = other than 00b). Values written to the ETICNT2IV[3:0] bits are applied only in this condition.

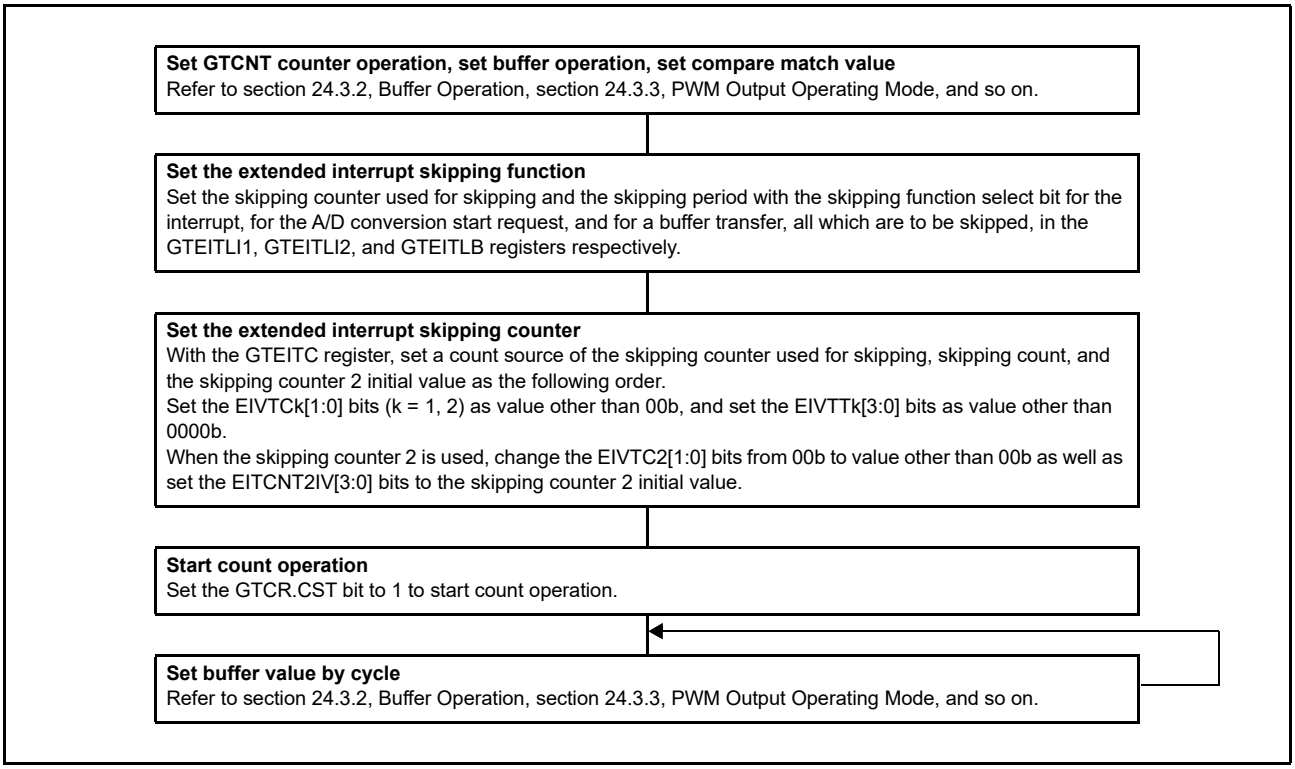


Figure 24.212 Example for Setting the Extended Interrupt Skipping

24.4.3.3 A/D conversion Start Request Compare Match Skipping Function

The A/D conversion start request and GTADTR register buffer transfer can be skipped by counting the compare matches of GTADTRA register and GTADTRB register based on settings of the GTADCMSC register and GTADCMSS register. The skipping period is set, related to the operation of the two independent A/D conversion start request compare match skipping counters (the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits in the GTADCMSC register), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 24.213 shows the counter operation for A/D conversion start request compare match skipping.

The counter operation for A/D conversion start request compare match skipping is set by the GTADCMSC register.

The ADCMSCNT1[3:0] bits count the count source (in Figure 24.213, a crest is selected) selected by the A/D conversion start request compare match skipping counter 1 count source select bit (the GTADCMSC.ADCMSC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 24.213, count is 2) set by the A/D conversion start request compare match skipping 1 skipping count setting bit (the ADCMST1[3:0] bits) is achieved.

The ADCMSCNT2[3:0] bits count the count source (in Figure 24.213, a trough is selected) selected by the A/D conversion start request compare match skipping counter 2 count source select bit (the GTADCMSC.ADCMSC2[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 24.213, count is 2) set by the A/D conversion start request compare match skipping 2 skipping count setting bit (the ADCMST2[3:0] bits) is achieved.

The ADCMSCNTk[3:0] bits (k = 1, 2) can be set for the initial value. The initial value is set when the GTADCMSC register is written by the access of 16 bits or 32 bits while the setting for the A/D conversion start request compare match skipping counter k is not counted (the ADCMSCk[1:0] bits are 00b) and the write value to the ADCMSCk[1:0] bits is other than 00b. When setting the initial value, the write value to the A/D conversion start request compare match skipping counter k initial value bits (the ADCMSCNTkIV[3:0] bits) are set as the initial value for the ADCMSCNTk[3:0] bits.

The A/D conversion start request compare match skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits for the A/D conversion start request compare match skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped.

When values for the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits are to be reset (0000b), set the ADCMSC1[1:0] and ADCMSC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the ADCMSC1[1:0] bits or the ADCMSC2[1:0] bits to 00b).

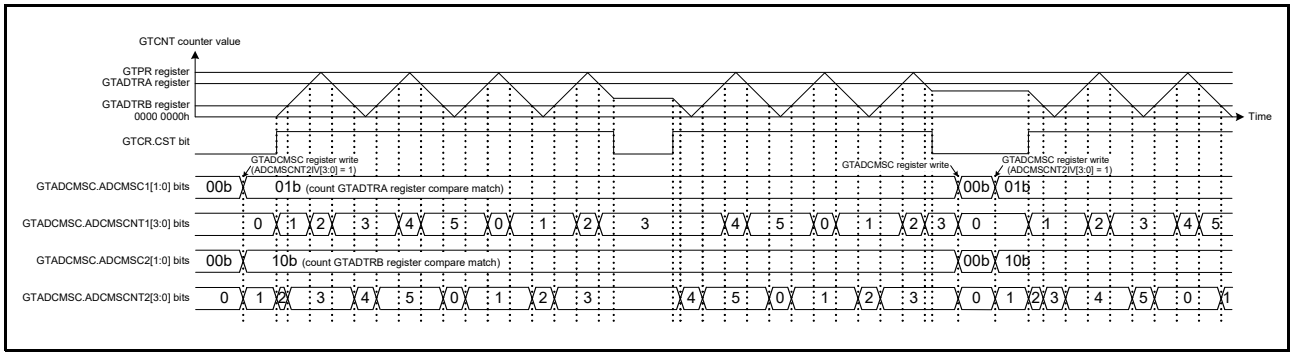


Figure 24.213 Example of A/D Conversion Start Request Compare Match Skipping Function Operation

A/D conversion start request skipping by the ADCMSC register can be performed simultaneously with skipping by the GTITC register or GTEITC register. A skipping period in this case is represented by ORed skipping periods of respective registers.

Figure 24.214 shows an example of simultaneous interrupt skipping operations by different registers.

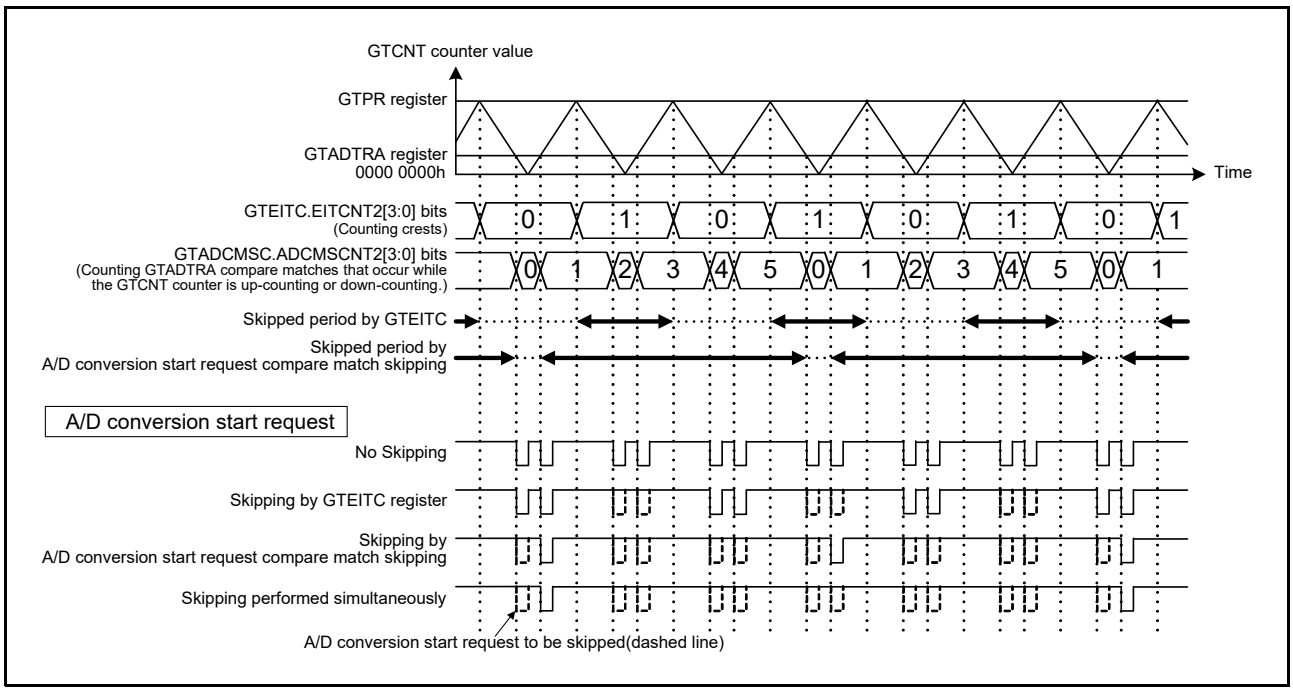


Figure 24.214 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (GTADTRA register compare match: GTINTAD.ADTRAUEN = 1, ADTRADEN = 1, Extended Interrupt Skipping: GTEITC.EIVTT2[3:0] = 001b, EIVTC2[1:0] = 01b, GTEITLI2.EADTAL[2:0] = 010b, A/D Conversion Start Request Compare Match Skipping: GTADCMSC.ADCMST2[3:0] = 101b, ADCMSC2[1:0] = 01b, ADCMSAL[2:0] = 010b)

When an A/D conversion start request is skipped by the GTADCMSS register, updating of the corresponding status flag and ELC event output are controlled by the A/D conversion start request enable bit in the GTINTAD register. All operations triggered by A/D conversion start request that are disabled by the GTINTAD register are not performed. Skipping of buffer transfer by the GTADCMSS register applies to all buffer operations enabled by the GTBER and GTBER2 registers, as well as all buffer operations performed in sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or complementary PWM mode (except for buffer transfer from GTCCRC, GTCCRE to GTCCRA). The skipping of an A/D conversion start request and the skipping of buffer transfer operate independently. Figure 24.215 and Figure 24.216 show examples of extended skipping function operation.

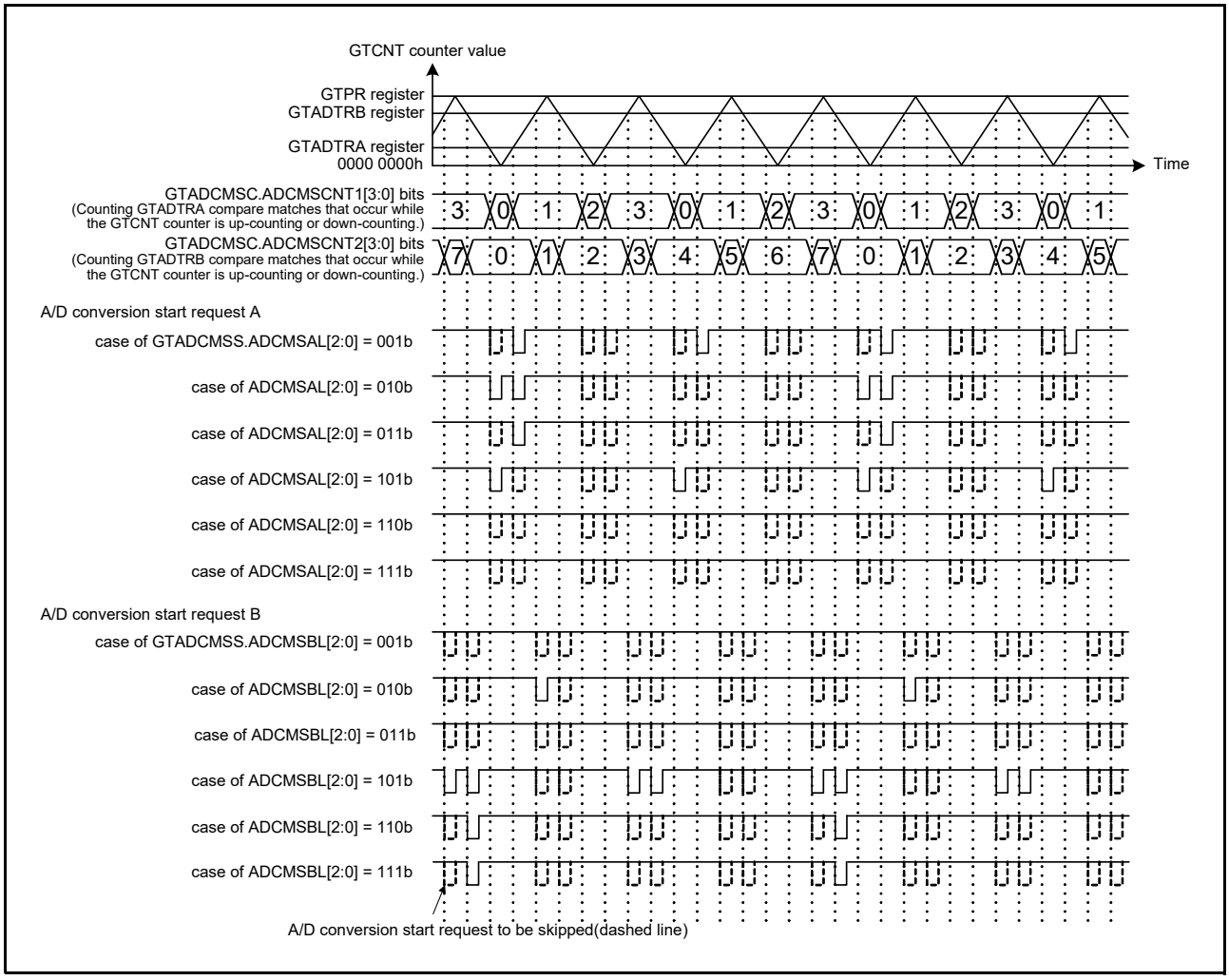


Figure 24.215 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match: GTINTAD.ADTRAUEN = 1, ADTRADEN = 1, GTADCMSS.ADCMSC1[1:0] = 01b, ADCMST1[3:0] = 011b, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match: GTINTAD.ADTRBUEN = 1, ADTRBDEN = 1, GTADCMSS.ADCMSC2[1:0] = 10b, ADCMST2[3:0] = 111b)

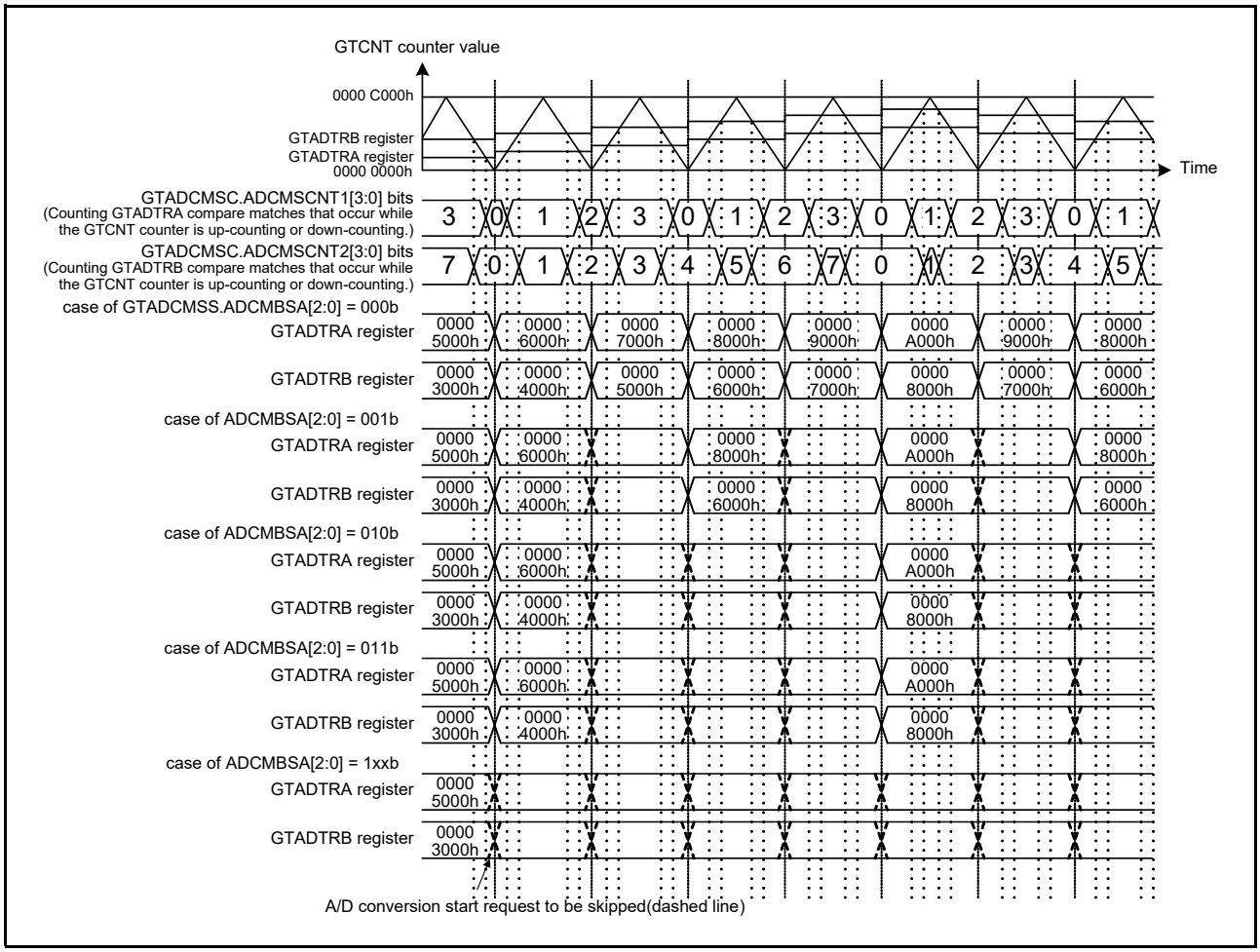


Figure 24.216 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match: GTINTAD.ADTRAUEN = 1, ADTRADEN = 1, GTADCMSS.ADCMSC1[1:0] = 01b, ADCMST1[3:0] = 011b, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match: GTINTAD.ADTRBUEN = 1, ADTRBDEN = 1, GTADCMSS.ADCMSC2[1:0] = 10b, ADCMST2[3:0] = 111b, Buffer transfer of GTADTRA and GTADTRB at trough: GTBER.BD[2] = 1, ADTTA[1:0] = 10b)

Figure 24.217 shows an example of setting the A/D conversion start request compare match skipping.
The initial value of the A/D conversion start request compare match skipping counter 2 is set by the value written to the GTADCMSC.ADCMSCNT2IV[3:0] bits. The initial value setting is applied when the setting of the extended interrupt skipping counter 2 count source is changed from “not counting” (the GTADCMSC.ADCMSC2[1:0] bits = 00b) to “counting” (the GTADCMSC.ADCMSC2[1:0] bits = other than 00b). Value written to the ADCMSCNT2IV[3:0] bits are applied only in this condition.

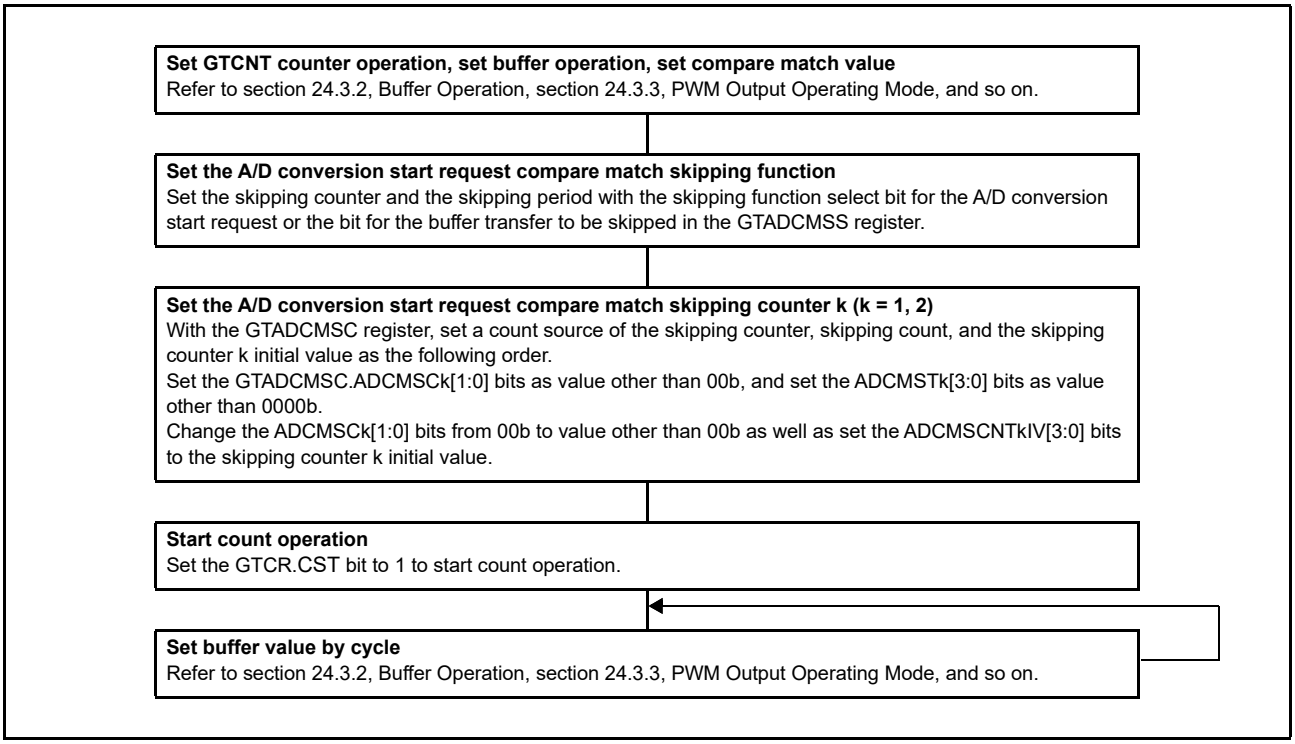


Figure 24.217 Example for Setting the A/D Conversion Start Request Compare Match Skipping

24.5 A/D Conversion Start Request

An A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The request can be directly output to the A/D converter or output as an event signal to the ELC.

Each of the GTADTRA and GTADTRB registers has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The generation timing of the A/D conversion start request can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected using the GTADSMR.ADSMSk bit (k = 0, 1) and the output is enabled by the ADSMENk bit, a signal is output on the GTADSMk pin synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal. The output is driven high at the generation of the A/D conversion start request signal, and is driven low at the end of the cycle. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPTW.

Figure 24.218 shows an example of A/D conversion start request operation, Figure 24.219 shows an example for setting A/D conversion start request operation, and Figure 24.220 shows an example for A/D conversion start request timing operation.

Figure 24.220 shows an example of outputting A/D conversion start request A generated by a compare match with the GTADTRA register directly to the A/D converter. When the GPTW operates with the PCLKD clock and the A/D converter operates with the PCLKA clock, A/D conversion start request A is output to the A/D converter at the next rising edge of PCLKA. When both the GPTW and A/D converter operate with the GPTCLK clock, the request is output to the A/D converter without delay. With the other combinations of clocks, the request is output to the A/D converter in synchronization with the operating clock for the A/D converter.

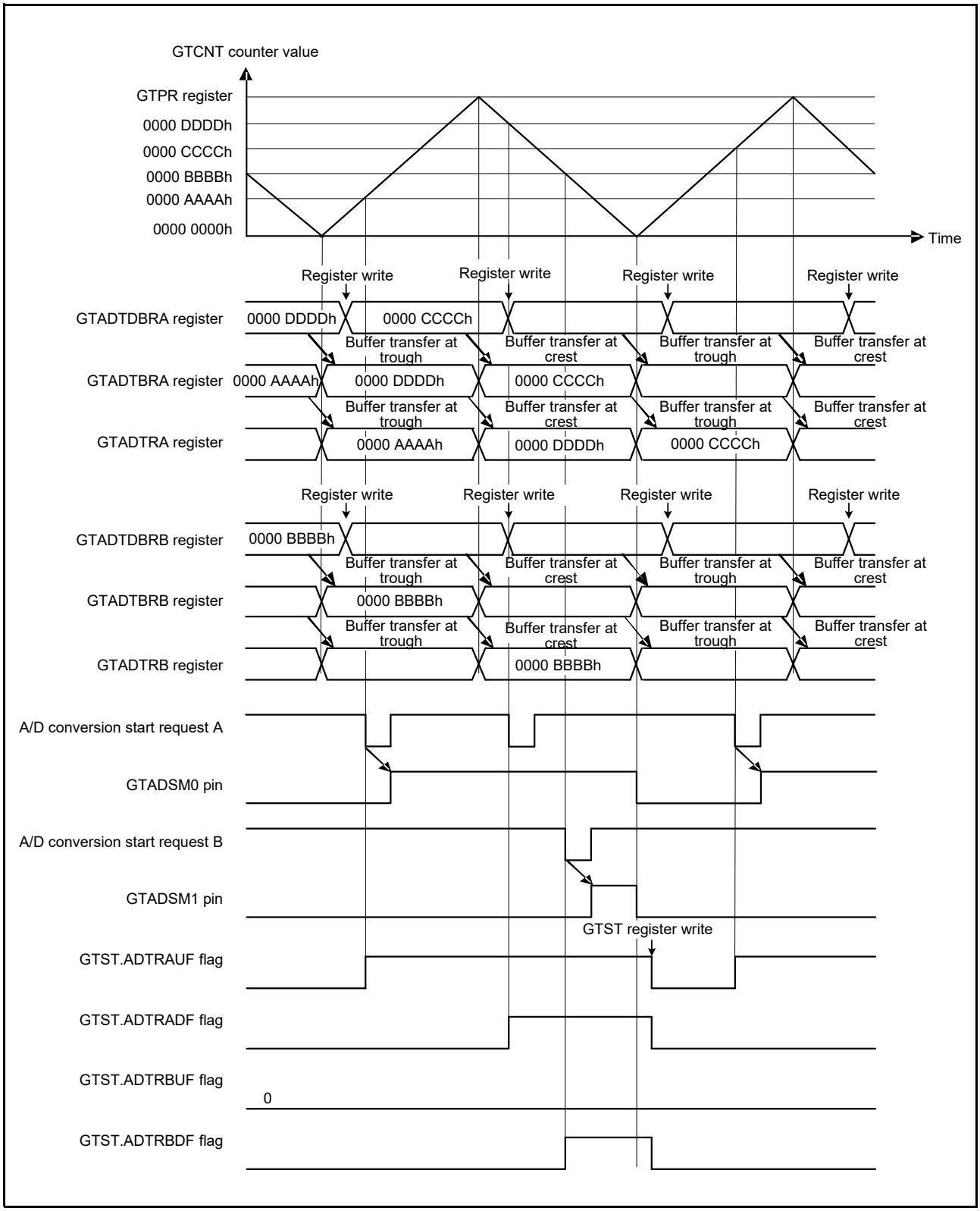


Figure 24.218 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Requested by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Requested by GTADTRB Register at Down-Counting, Monitoring of the GTADTRA Register Up-Counting by the GTADSM0 Pin, Monitoring of the GTADTRB Register Down-Counting by the GTADSM1 Pin)

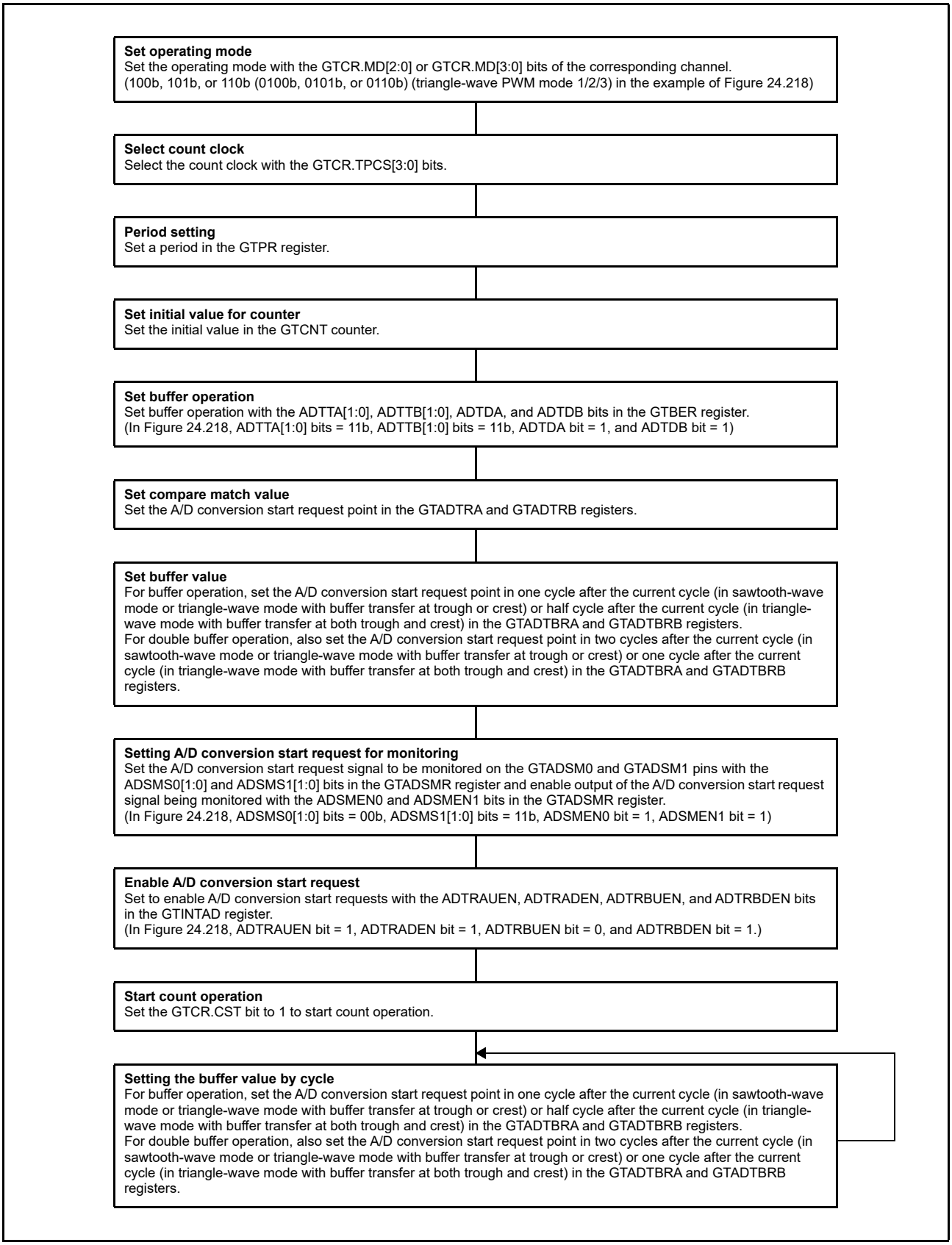


Figure 24.219 Example for Setting A/D Conversion Start Request Operation

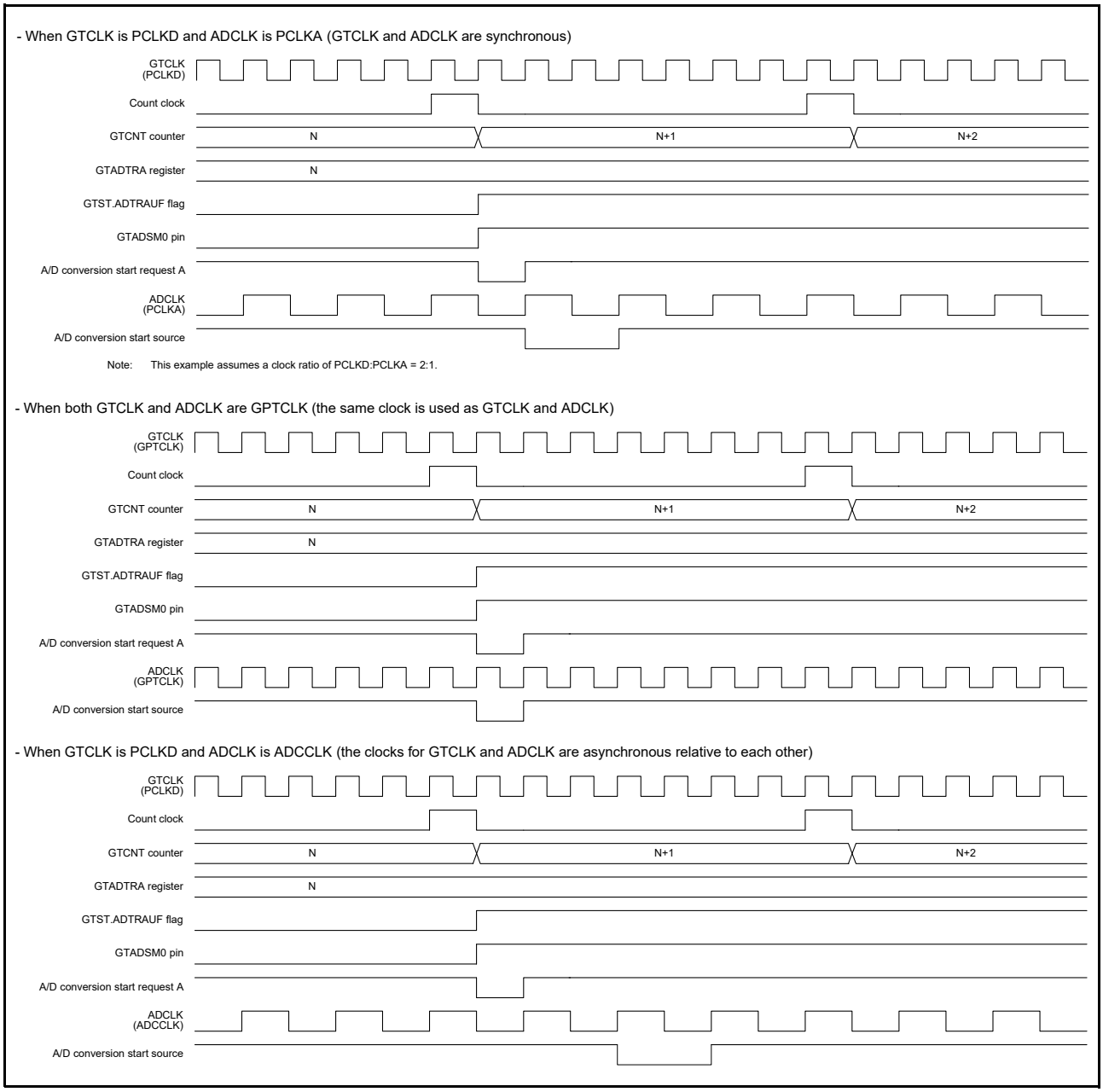


Figure 24.220 Examples of Timing of Operations with an A/D Conversion Start Request

24.6 Operations Linked by the ELC

24.6.1 Event Signal Output to the ELC

GPTW is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D conversion start request.
The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output to the ELC.
The following is a list of the event signals output by the GPTW to the ELC. Each GPTW channel has the first 11 event signals. The last of the listed event signals is common to all channels. A dead time error interrupt does not have a corresponding event signal.

- Generation of compare match A interrupt
- Generation of compare match B interrupt
- Generation of compare match C interrupt
- Generation of compare match D interrupt
- Generation of compare match E interrupt
- Generation of compare match F interrupt
- Generation of overflow interrupt
- Generation of underflow interrupt
- Generation of A/D conversion start request A
- Generation of A/D conversion start request B
- End of the cycle count operation
- GPTW (OPS) U-/V-/W-phase input edge detected

24.6.2 GPTW Operations in Response to Receiving Event Signals from the ELC

The GPTW can perform the following operations by the signals for sources A to H which are output from the ELC. Each signal is provided to all channels. Select an event source for a desired operation by a relevant source select register in a channel.

- Operation of count start/count stop/counter clearing/count switch
- Operation of up-counting/down-counting
- Operation of input capture A and B

Refer to operation by hardware sources in section 24.3.1.1, Counter Operation for respective operations.

24.7 Noise Filter Function

Each pin for use in input capture to the GPTW is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling periods.
The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel.
Figure 24.221 shows the timing of noise filtering.

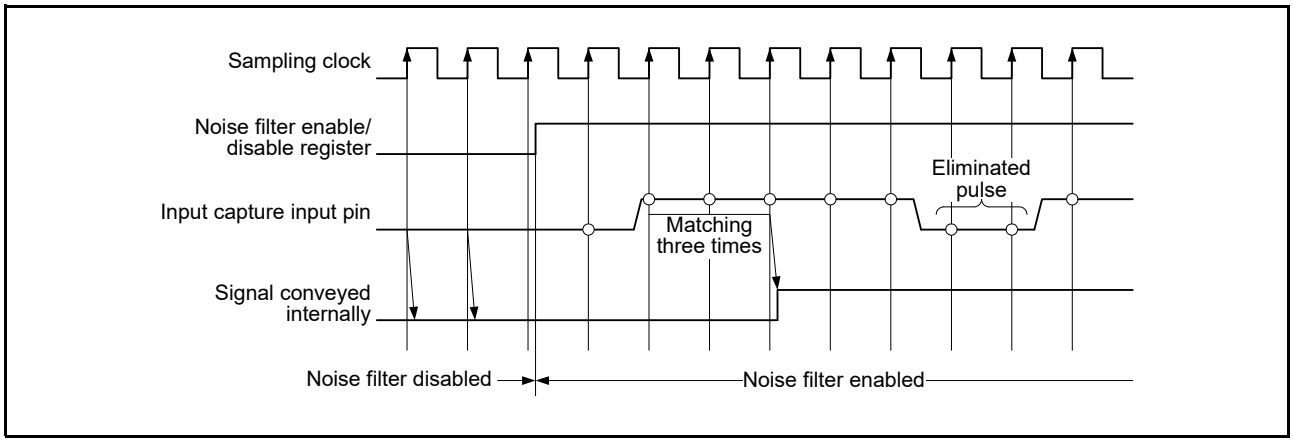


Figure 24.221 Timing of Noise Filtering

If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a minimum delay of (sampling interval × 2 + GTCLK) due to noise filtering for the input capture input.

24.8 Protection Function

24.8.1 Write-Protection for Registers

In order to avoid incorrect writing to registers, write access to registers can be enabled or disabled per channel by setting the GTWP.WP bit for the given channel.
The WP bits enable or disable writing to the registers listed below.
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTSWSR, GTSWOS, GTBER2, GTOLBR, GTICCR, GTSYNCFCR, GTSYNCSCR
The GTSTR, GTSTP, and GTCLR registers, which can control all channels by updating the registers of any channel, can be protected from accidental writes by enabling or disabling writes to the bit corresponding to each channel using the GTWP.STRWP, STPWP, and CLRWP bits on a per-channel basis.
Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels, can be enabled or disabled for each channel by the setting of the GTWP.CMNWP bit.
Protection by the GTWP register only targets writing operation by the CPU. Register updates that occur with CPU writes are not protected.

24.8.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended with setting the BD[0], BD[1], BD[2], and BD[3] bits in the GTBER register.
Specifically, buffer transfer can be temporarily disabled, even when a buffer transfer condition is satisfied during buffer register write, by setting the corresponding BD[0], BD[1], BD[2], and BD[3] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.
The BD[0], BD[1], BD[2], and BD[3] bits can be set on a per-channel basis by writing directly to the GTBER register or they can be set simultaneously for multiple channels specified by the GTSECSR register using the GTSECR register.
Figure 24.222 shows an example of operation for disabling buffer operation.

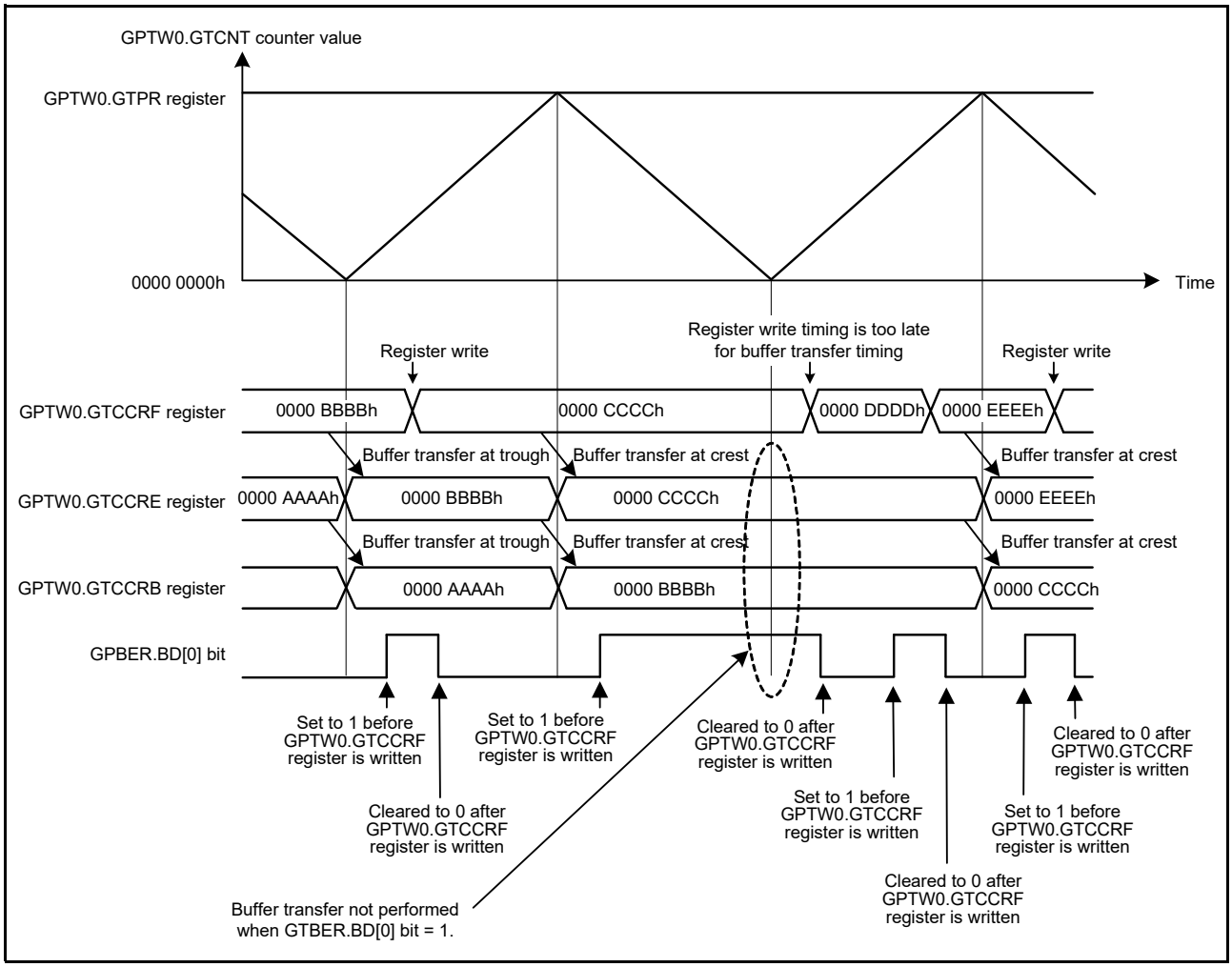


Figure 24.222 Example of Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests)

24.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.
Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

(1) Select the channels for simultaneously setting of the GTSECSR register

Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.

(2) Simultaneously set the GTBER.BD bits by updating the GTSECR register

In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register. Figure 24.223 and Figure 24.224 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

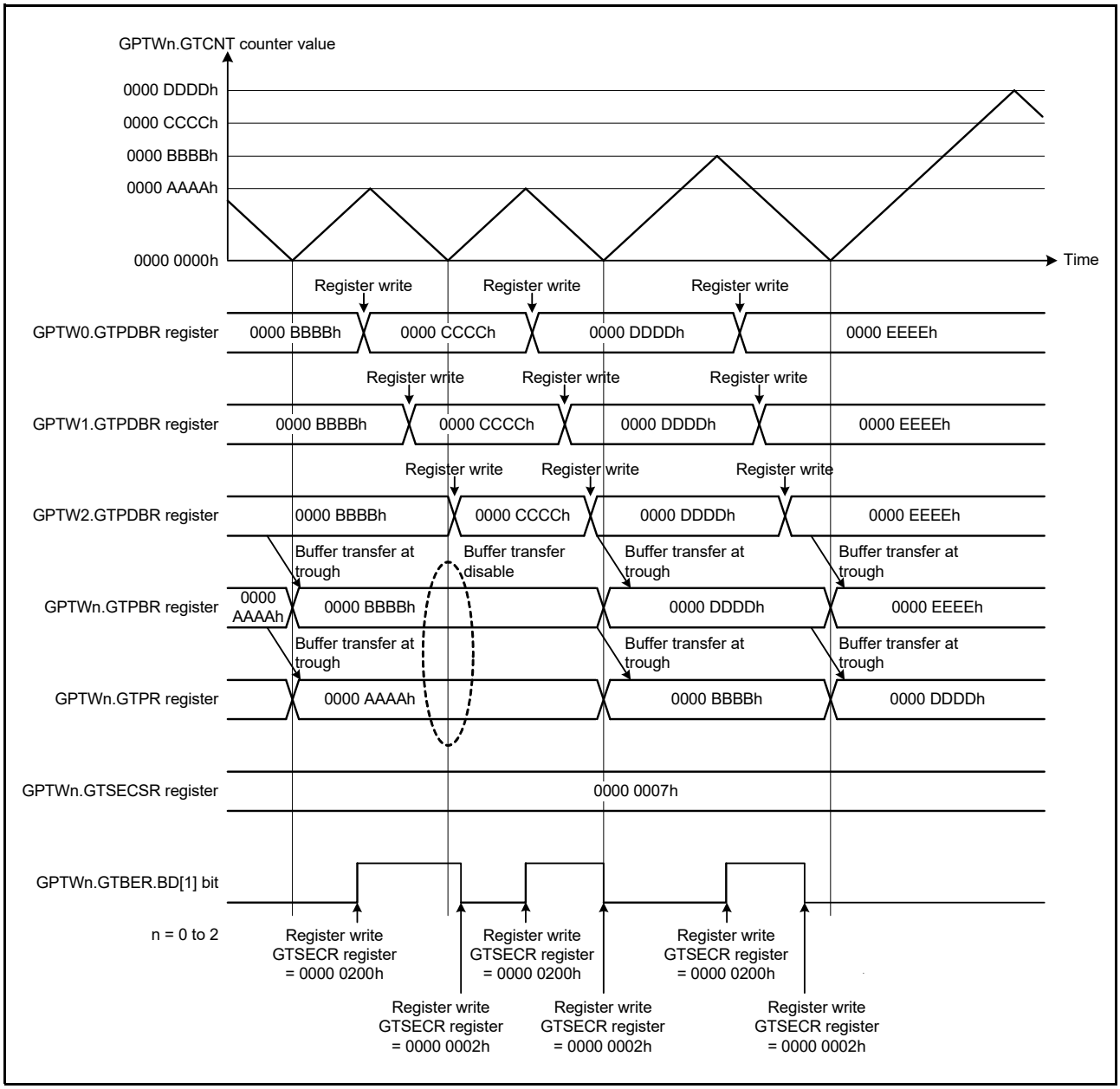


Figure 24.223 Example of Multiple Channel Operation for Simultaneous Control of Buffer Operation (Triangle Waves, Double Buffer Operation)

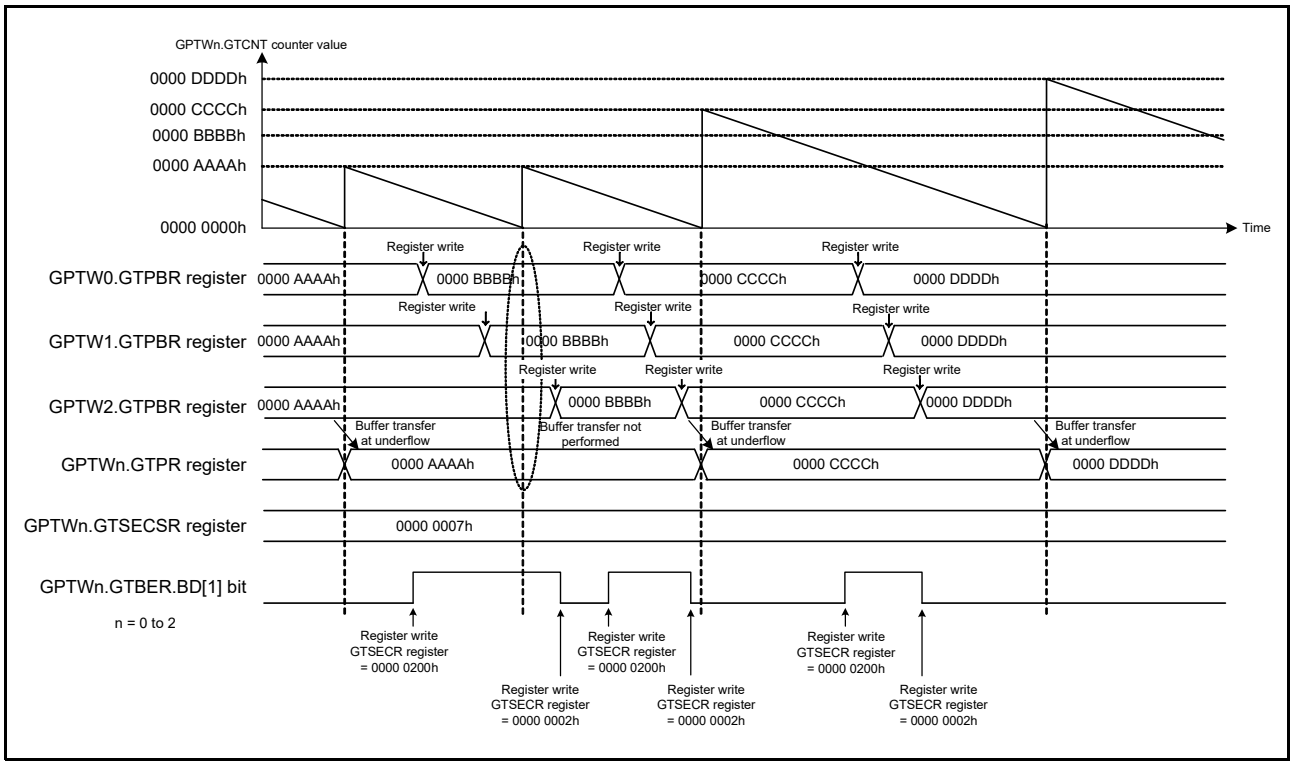


Figure 24.224 Example of Multiple Channel Operation for Simultaneous Control of Buffer Operation (Sawtooth Waves, Single Buffer Operation)

24.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or by the buffer transfer extended skipping function.

(1) In sawtooth-wave one-shot pulse mode

In sawtooth-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 24.42 lists the types of buffer transfer of the GTCCR register during counting in sawtooth-wave one-shot pulse mode. While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer.

In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 24.42 GTCCR Buffer Operation in Sawtooth-Wave One-Shot Pulse Mode

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer-enabled period	Overflow or underflow	No transfer	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer-disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer-enabled period	Overflow or underflow	Overflow or underflow	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer-disabled period	No transfer	No transfer	Overflow or underflow	No transfer	GTCCRm compare match

Figure 24.225 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in sawtooth-wave one-shot pulse mode.

Figure 24.226 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

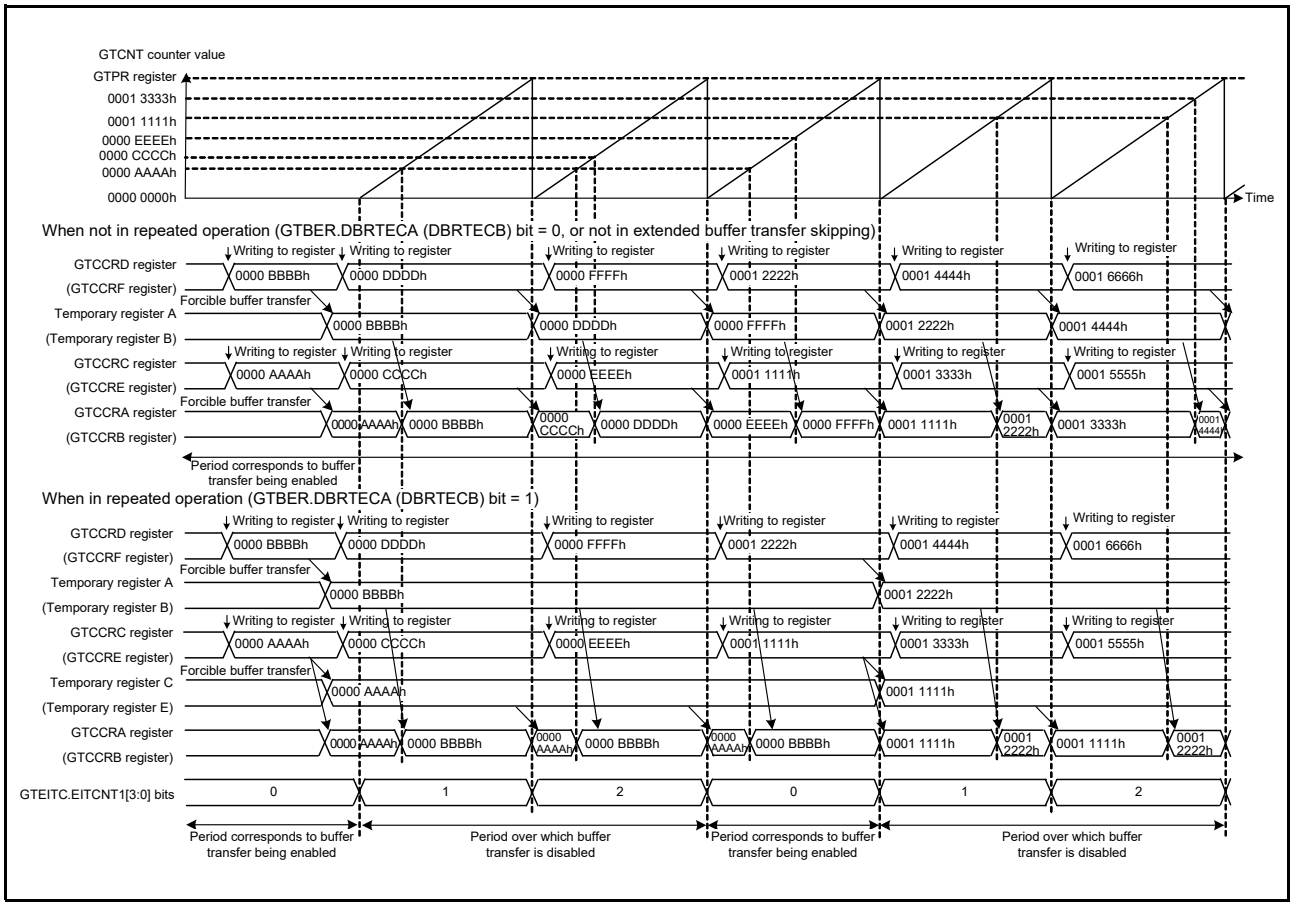


Figure 24.225 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Sawtooth-Wave One-Shot Pulse Mode, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

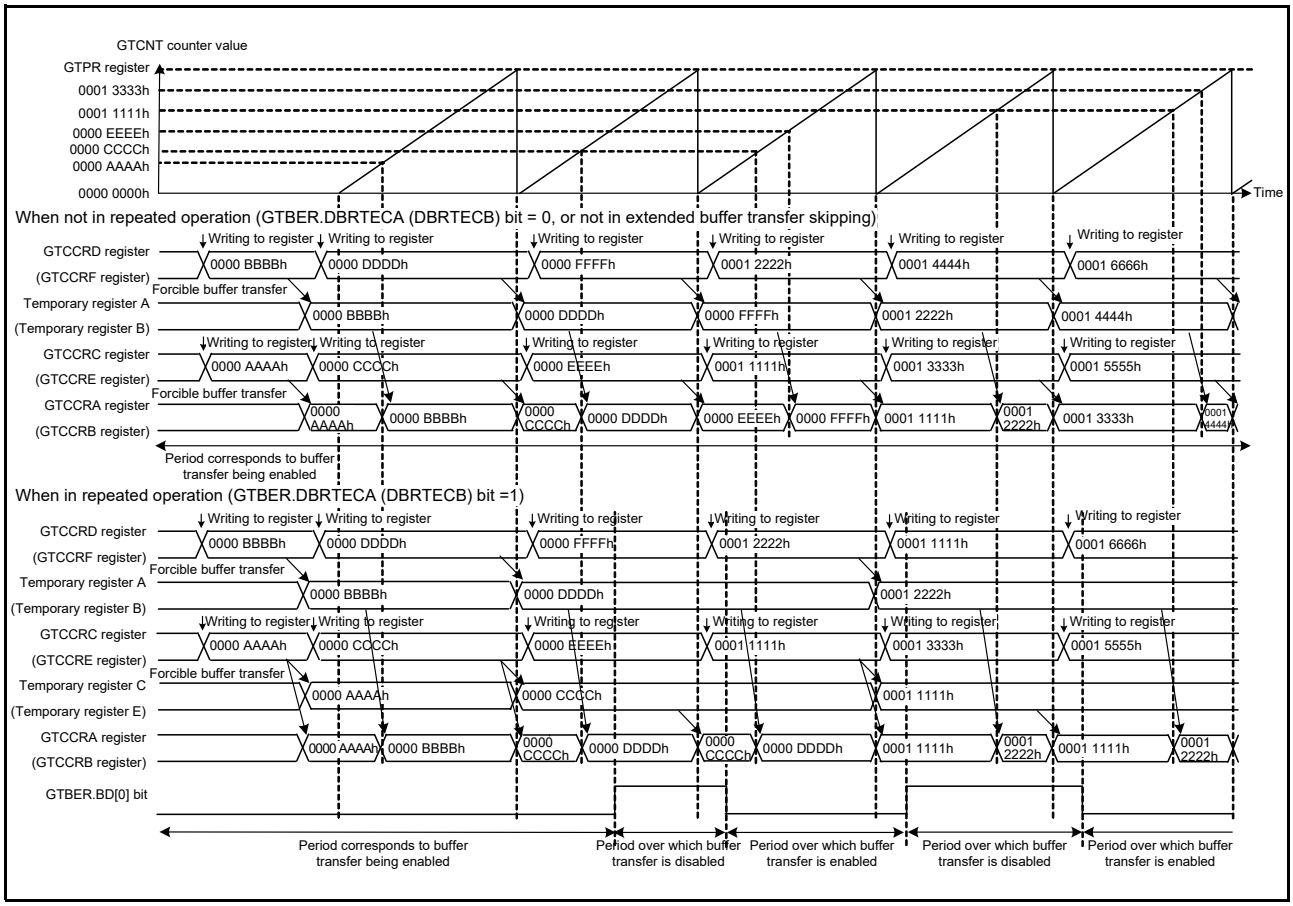


Figure 24.226 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Sawtooth-Wave One-Shot Pulse Mode, Updating the GTBER.BD[0] Bit)

(2) In triangle-wave PWM mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 24.43 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 24.43 GTCCR Buffer Operation in Triangle-Wave PWM Mode 3

		Timing of transfer				
GTBER.DBRTECm	**Buffer transfer**	**GTCCRx ↓ GTCCRm**	**GTCCRx ↓ Temporary register x**	**Temporary register x ↓ GTCCRm**	**GTCCRy ↓ Temporary register m**	**Temporary register m ↓ GTCCRm**
0	Transfer-enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer-disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer-enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer-disabled period	No transfer	No transfer	Trough	No transfer	Crest

Figure 24.227 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in triangle-wave PWM mode 3.

Figure 24.228 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

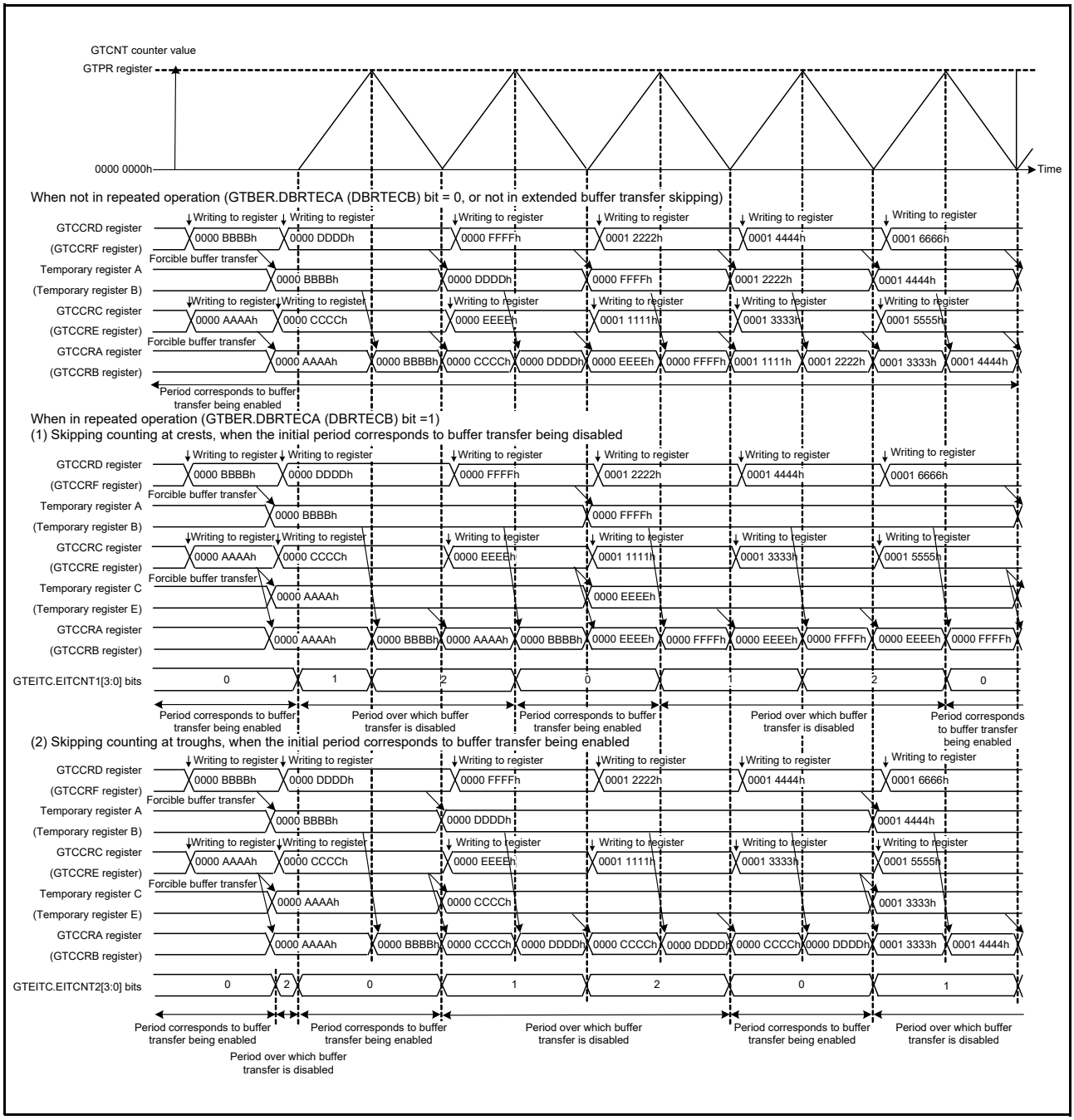


Figure 24.227 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

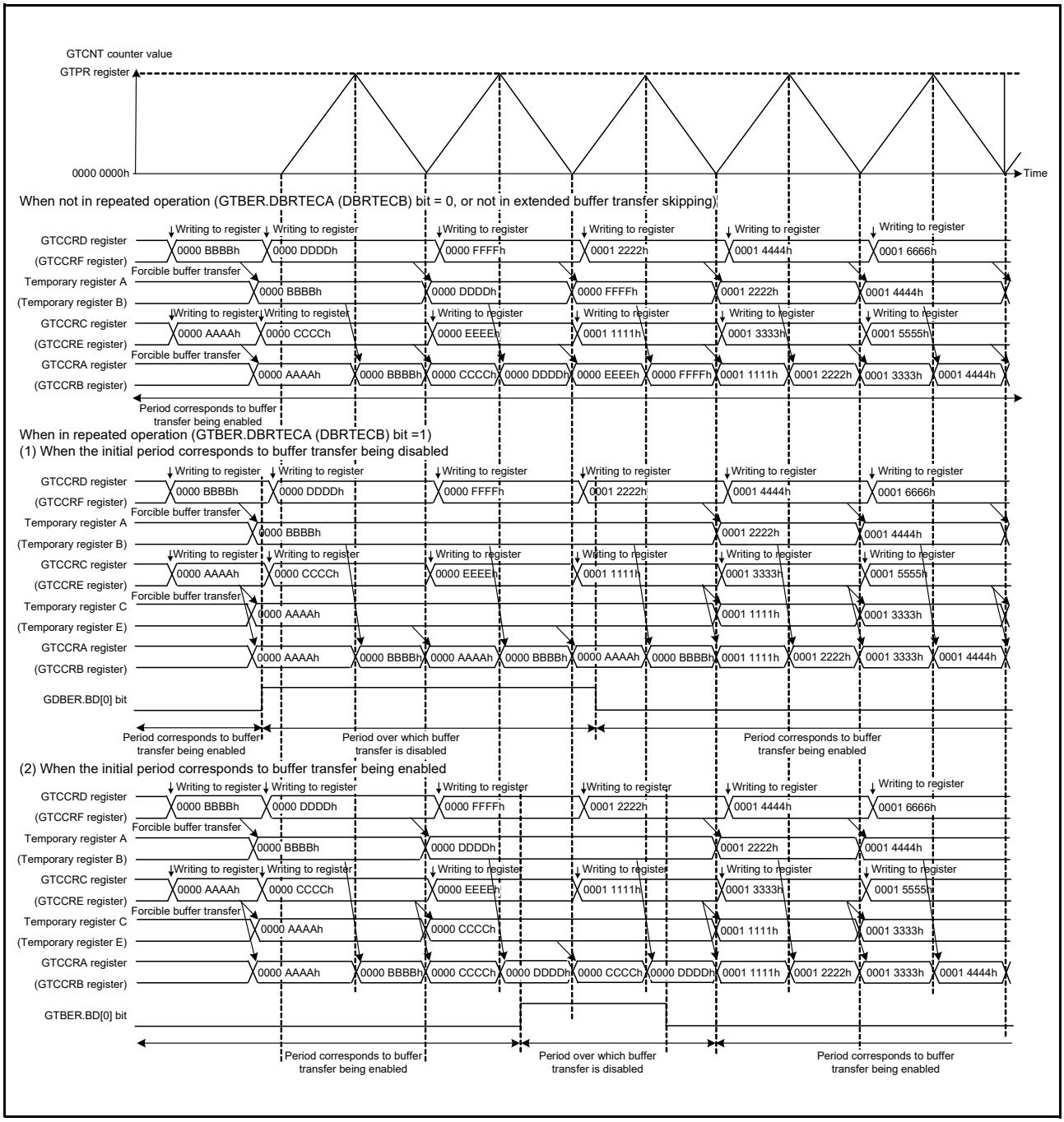


Figure 24.228 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Updating the GTBER.BD[0] Bit)

24.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 15; m = A, B)

For protection from system failure, the output negate control which changes the GTIOCnm pin output forcibly by the output stop request from the POEG is provided.

Output protection is required when a dead-time error or the same output level being on the GTIOCnA and GTIOCnB pins is detected. At that time, the detection of disabling of output is output to the group of the POEG set in the GTINTAD.GRP[1:0] bits based on the setting of the output stop detection enable bits, i.e. the GRPDTE, GRPABH, and GRPABL bits in the GTINTAD register. The POEG takes the logical OR of detection of the types described above and other forms of detection of the disabling of output, and outputs requests for the disabling of output to the GPTW.

By setting the GTINTAD.GRP[1:0] bits, one output stop request can be selected out of four output stop requests which are input from the POEG as output stop request signal common in the GTIOCnA and GTIOCnB pins. Selected output stop request can be checked by reading the GTST.ODF flag.

The states of outputs at the time of control for their negation can be set with the GTIOR.OADF[1:0] bits for the GTIOCnA pin and with the GTIOR.OBDF[1:0] bits for the GTIOCnB pin.

Transition to the output negate condition by generating the output stop request from the POEG is performed asynchronously, while a release from the output negate condition by making the output stop request no longer satisfied is performed at the end of the cycle. The output negate condition is released no sooner than 3 GTCLK cycles after the output stop request is cleared. To reliably control output negation, clear the flag of POEG so that the output disable request is terminated at least 4 GTCLK cycles after it is generated.

To release the output stop condition during event count operation, in sawtooth-wave PWM mode 2, or without waiting the end of the cycle, set the OADF[1:0] bits to 00b for the GTIOCnA pin, and set the OBDF[1:0] bits to 00b for the GTIOCnB pin.

Figure 24.229 shows the output negate control operation for the GTIOCnm pin output.

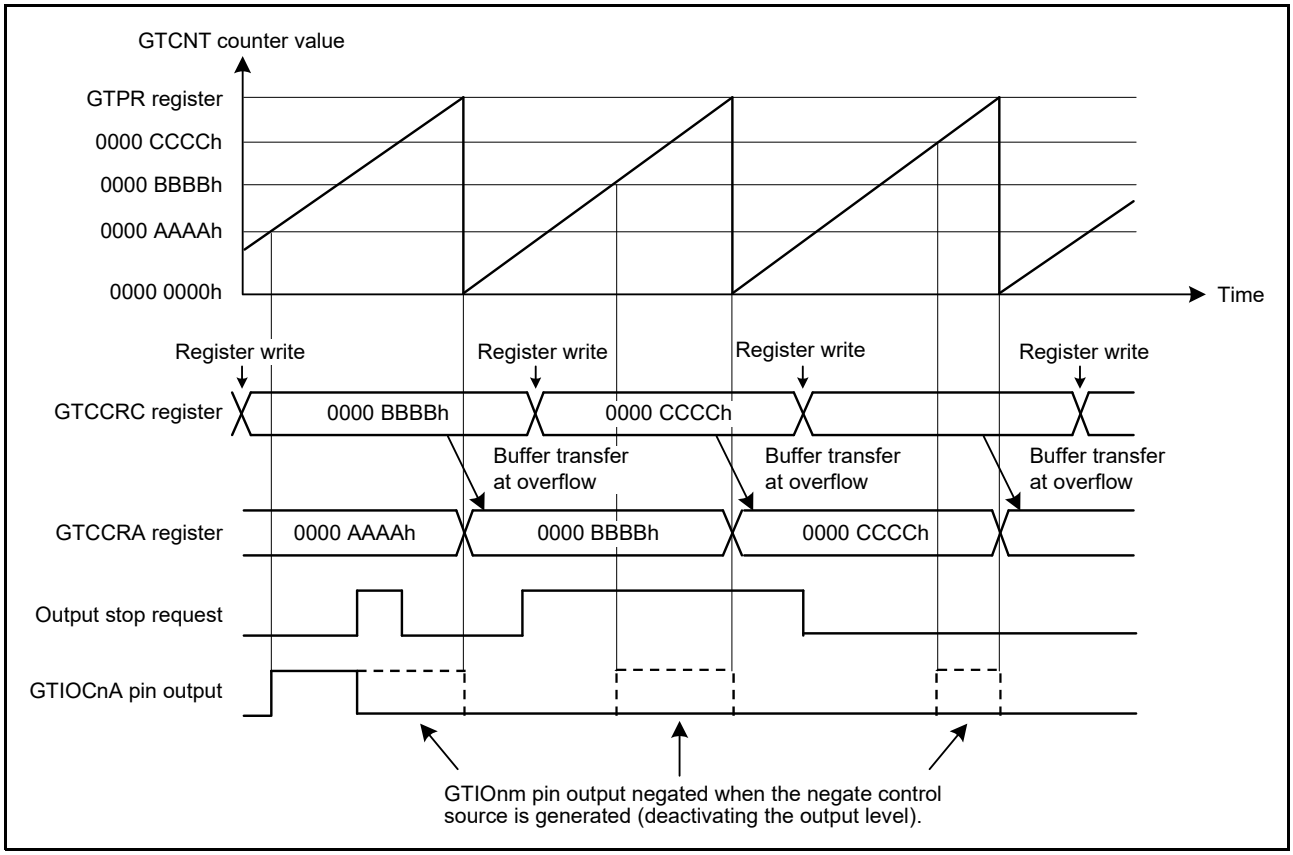


Figure 24.229 Example of Operation for GTIOCnm Pin Output Negate Control (Sawtooth-Wave Up-Counting, Buffer Operation, Active Level: High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) and Low Output at Output Negate) (n = 0 to 15; m = A, B)

24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 15; m = A, B)

To prepare for a case when an incorrect value (0000 0000h or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.
The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.
Figure 24.230 shows the output protection function state transition.

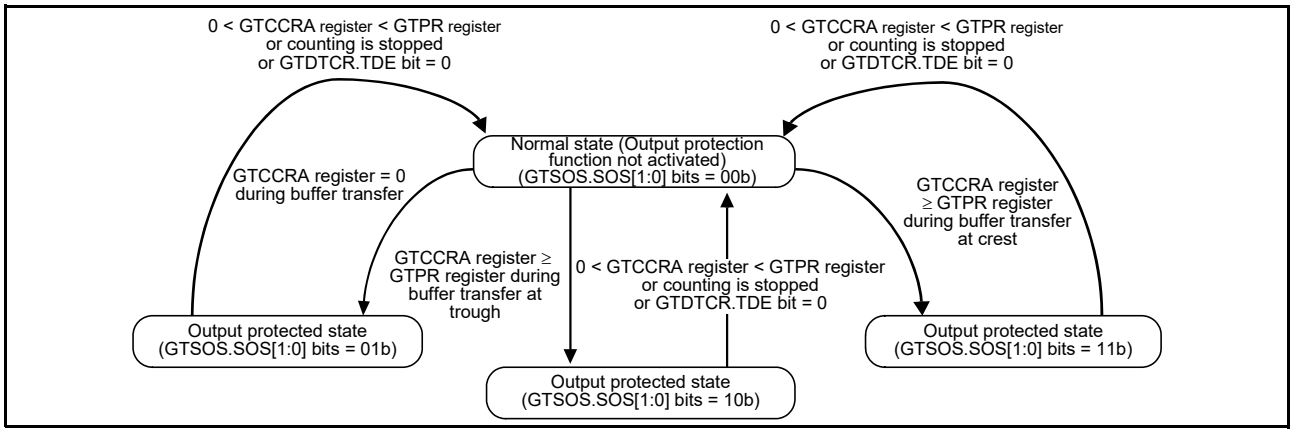
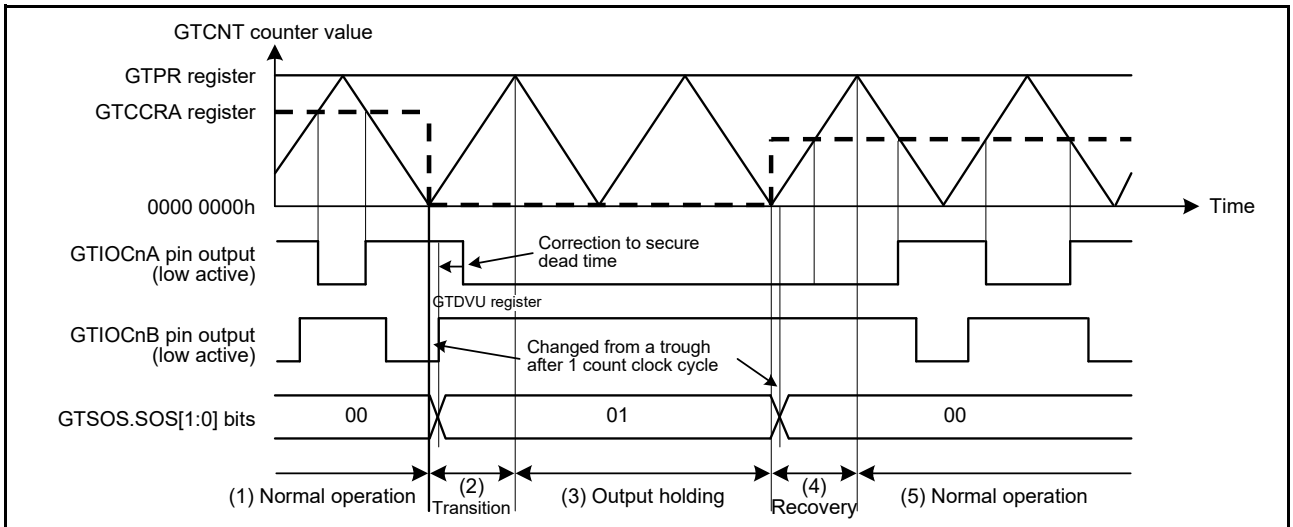


Figure 24.230 Output Protection Function

(1) Output Protection Function when the GTCCRA Register is Set to 0000 0000h during Buffer Transfer

Figure 24.231 and Figure 24.232 show examples of output protection function operation when the GTCCRA register is set to 0000 0000h during buffer transfer at troughs, and Figure 24.233 and Figure 24.234 show examples when the GTCCRA register is set to 0000 0000h during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 24.231 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 15)

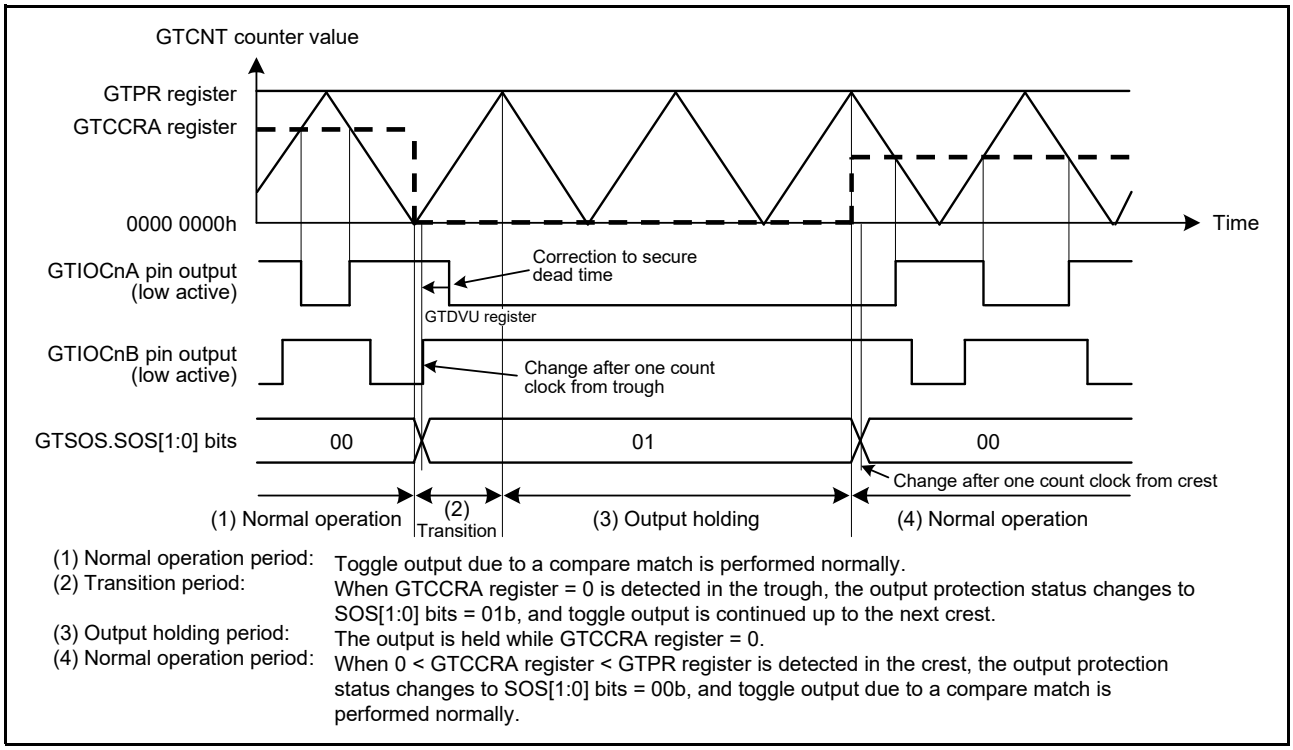


Figure 24.232 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 15)

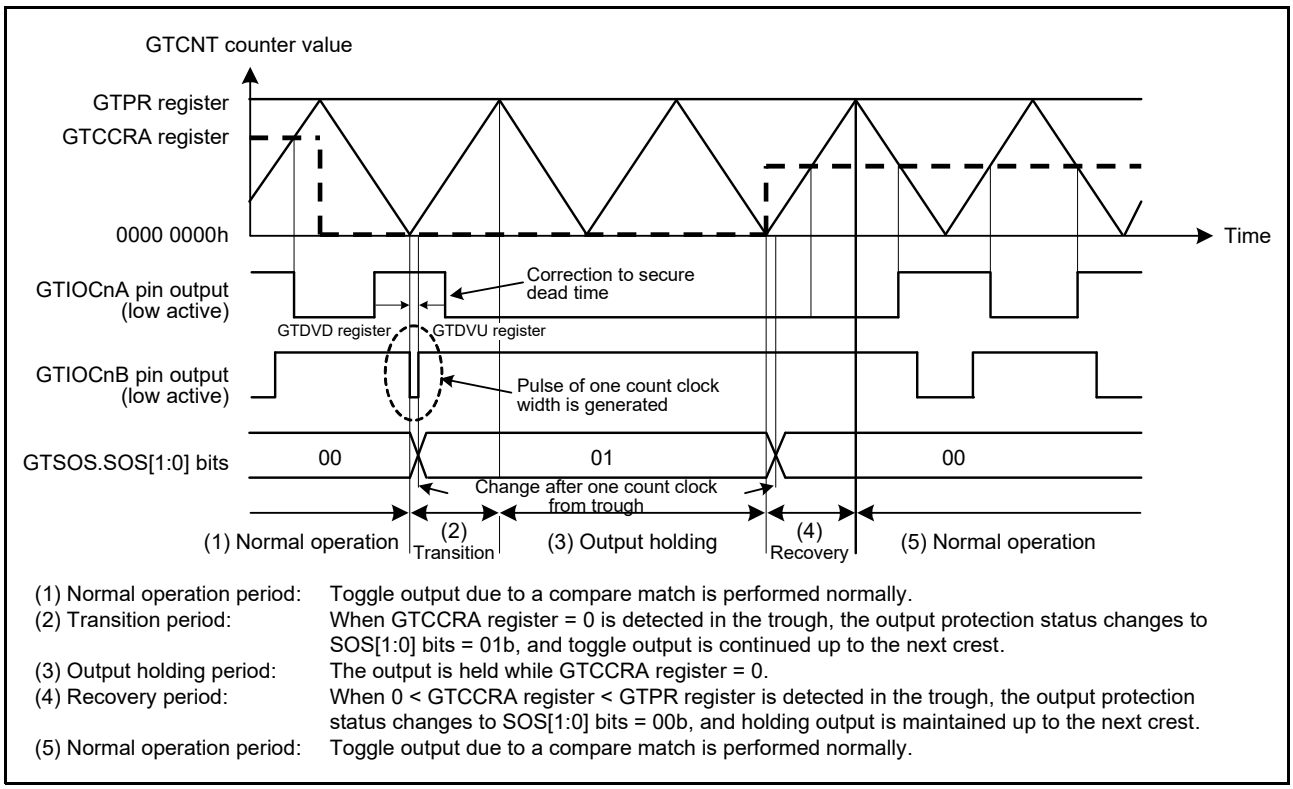


Figure 24.233 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 15)

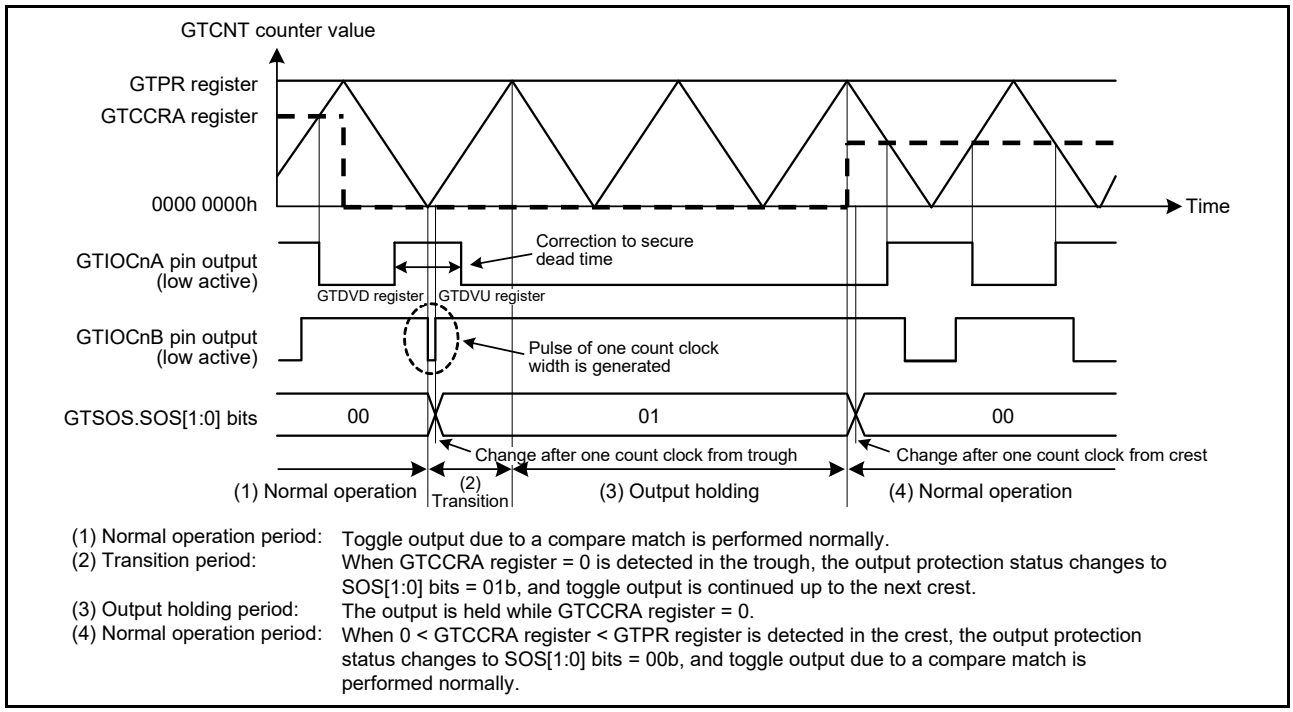


Figure 24.234 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 15)

(2) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 24.235 and Figure 24.236 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.

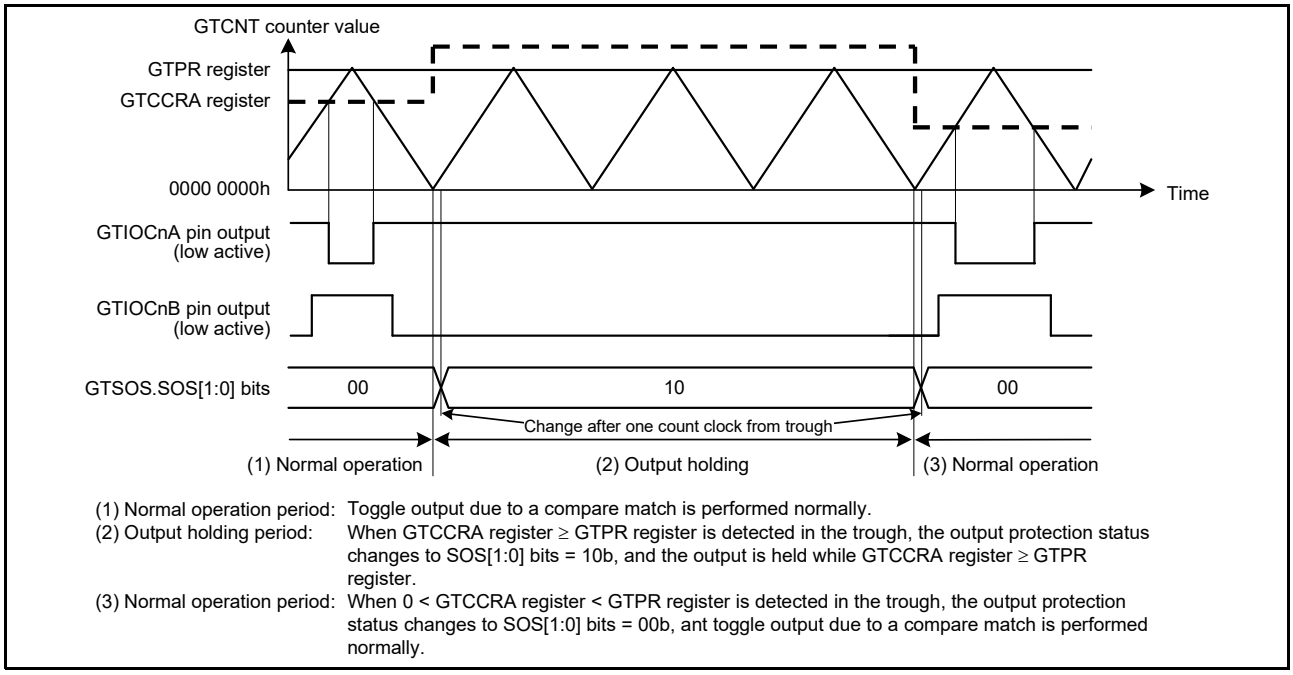
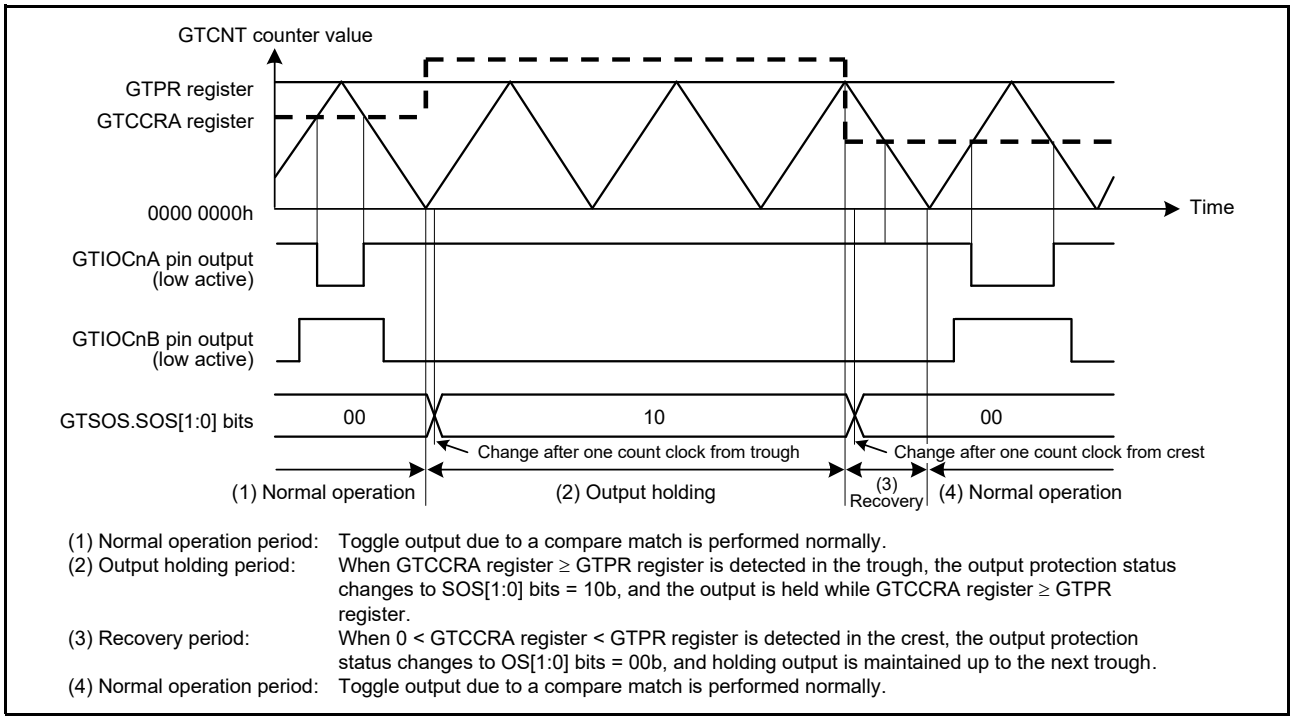


Figure 24.235 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 15)

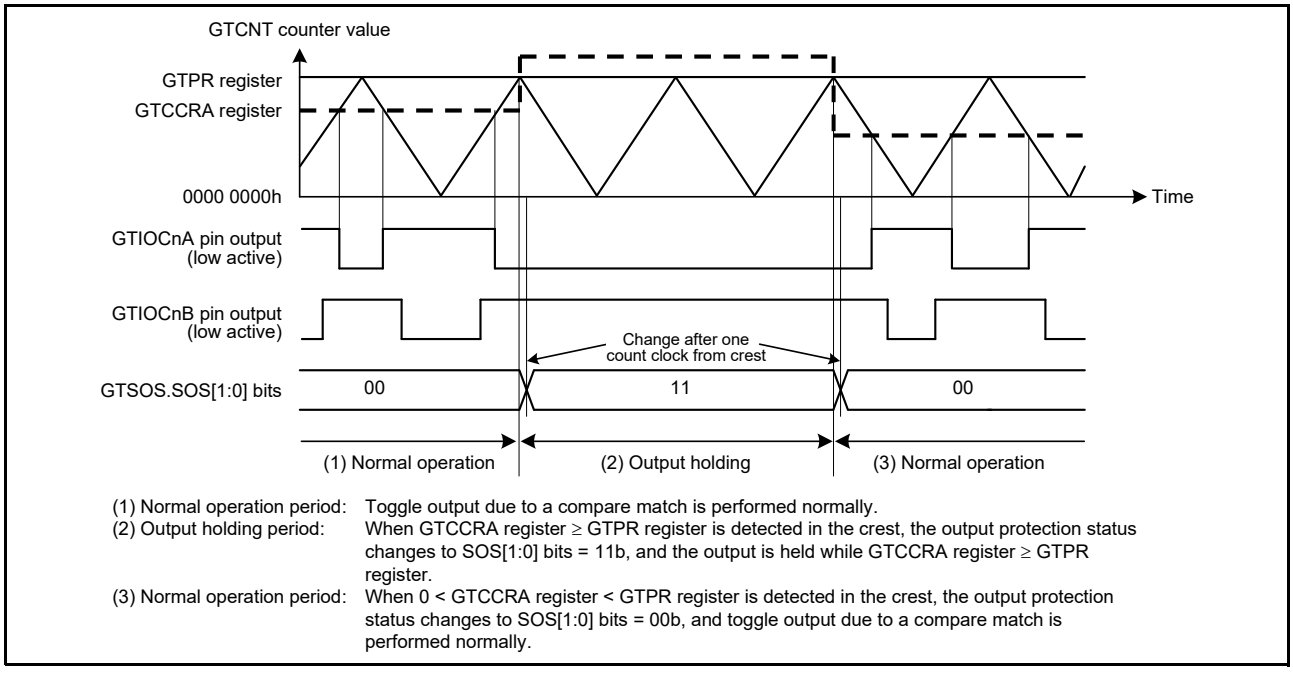


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to OS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 24.236 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 15)

(3) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests

Figure 24.237 and Figure 24.238 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to SOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 24.237 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 15)

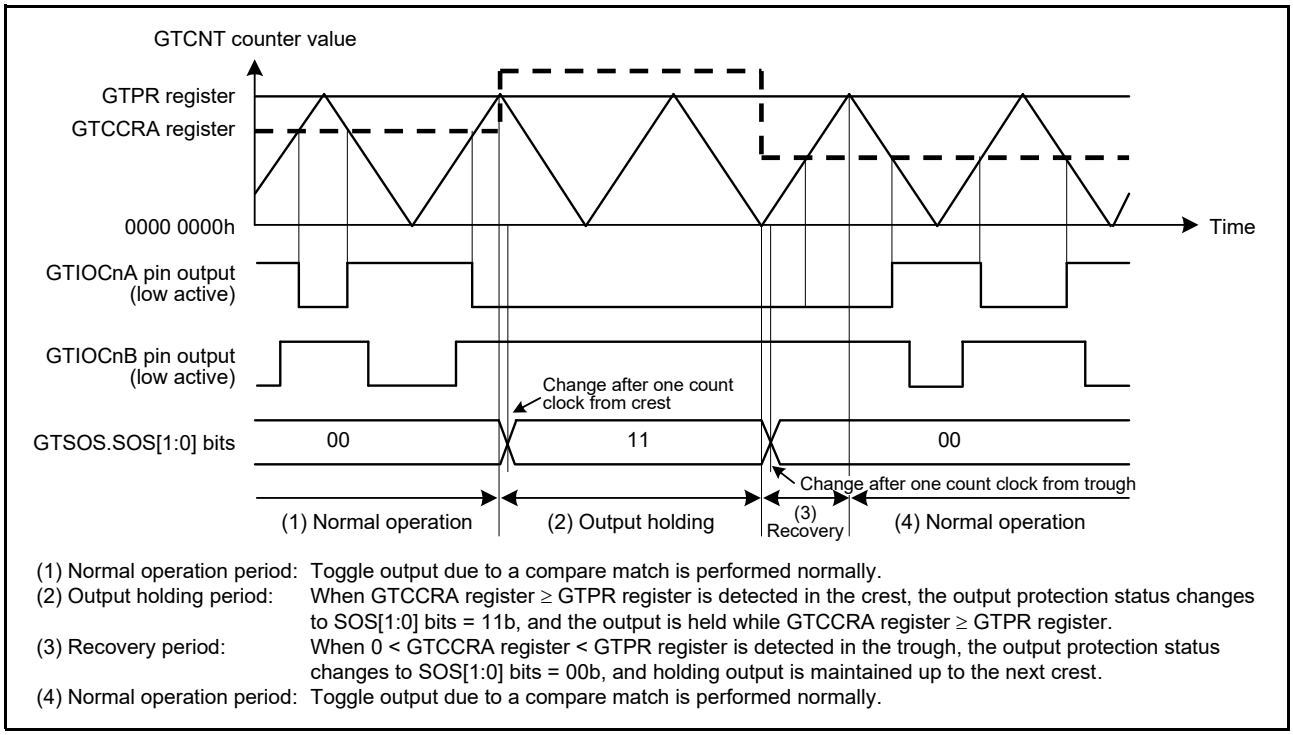


Figure 24.238 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 15)

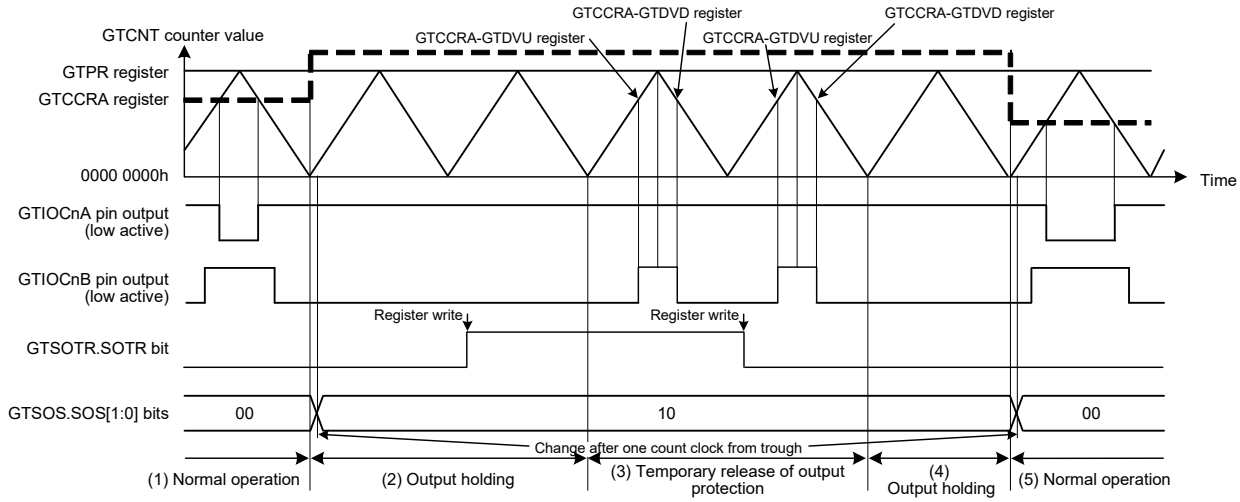
(4) Restricted Specification of Output Protection Function

Even if an incorrect value (0000 0000h or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, if the following condition is not satisfied, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is greater than 0000 0000h, and less than the setting value of the GTPR register

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.
When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.
Figure 24.239 shows an example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register. Even when the SOTR bit is set to 1, the GTIOCnB pin output is held until the first trough.
(3) Output protection temporary release period:	After the SOTR bit is set to 1, the GTIOCnB pin output is performed normally from the first trough. Even after the SOTR bit is set to 0, temporary release of output protection for the GTIOCnB pin continues until the next trough.
(4) Output holding period:	The output is held until 0 < GTCCRA register < GTPR register is detected in the tough.
(5) Normal operation period:	When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 24.239 Example of the Operation of Temporary Release of Output Protection When the Setting of the GTCCRA Register ≥ GTPR Register during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 15)

24.9 Initialization Method of Output Pins

24.9.1 Pin Settings after Reset

The GPTW registers are initialized at a reset. Initialize the GPTW for output to the external pins by making the port mode settings and setting the OAE and OBE bits in the GTIOR register, and then start the counter counting.

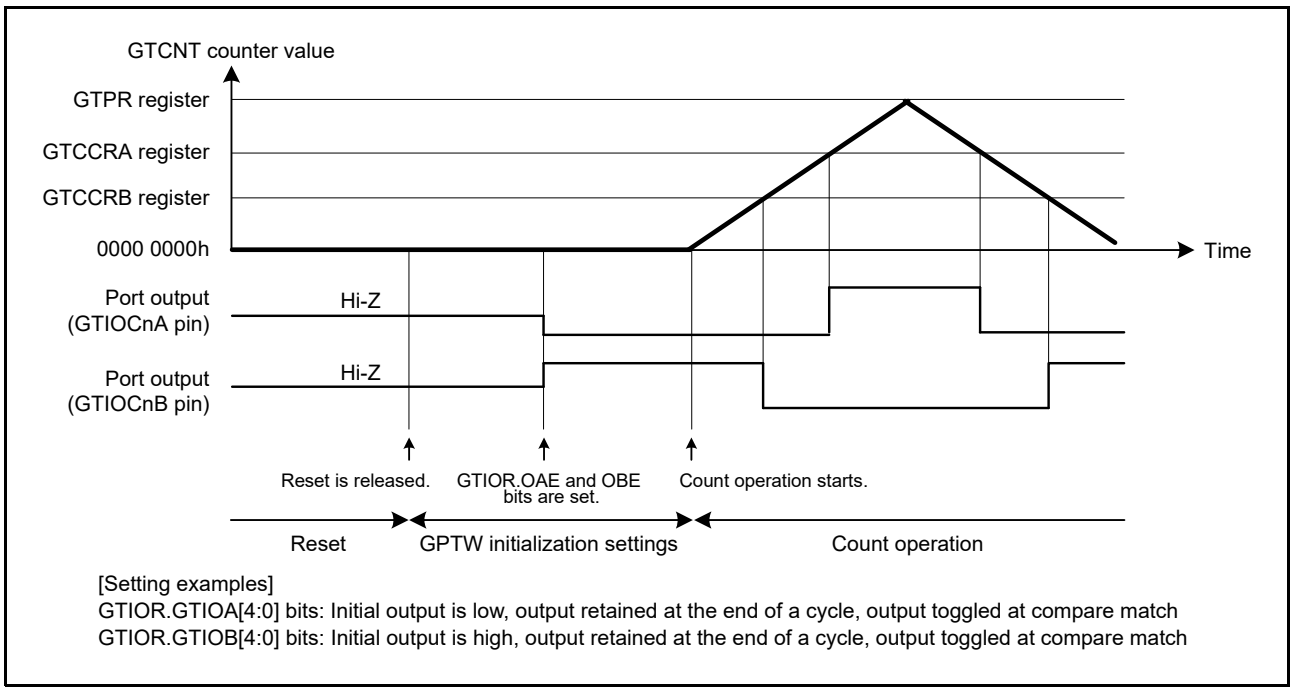


Figure 24.240 Example of Pin Settings after Reset (n = 0 to 15)

24.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPTW operation, the following four types of pin processing can be performed before pin initialization.

(1) Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.
(2) Set OAHLD and OBHLD bits to 0, specify arbitrary output values at OADFLT and OBDFLT bits in the GTIOR register, and output the arbitrary values at count stop.
(3) Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, and PMR registers of the I/O port in advance. Set the OAE and OBE bits in the GTIOR register to 0 and the control bit in PMR register that corresponds to the pin to 0 to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.
(4) Drive the output to a high impedance state using the POEG function.

When automatic dead time setting is enabled, set the GTDTCR.TDE bit to 0 once after counting is stopped.
When counting is stopped, only the values of registers that are changed by a GPTW external source will change. If counting is resumed, operation will carry on from where it was stopped.
If counting is stopped, registers should be initialized before restarting the count operation.

24.10 Usage Notes

24.10.1 Module Stop Function Setting

Operation of the GPTW can be disabled or enabled by the module stop control register. The initial setting is for operation of the GPTW to be halted. Register access is enabled by clearing module stop state. For details, see section 11, Low Power Consumption.

24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

(1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode

The GTCCRA register should satisfy the following conditions:

GTCCRA register > GTDVU register,
GTCCRA register > GTDVD register, and
GTCCRA register < GTPR register.

When the GTCCRA register is set to 0000 0000h or a value greater than or equal to the GTPR register value during count operation, the output protection function is activated.
However, if the following conditions are not satisfied, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is 0000 0001h or greater, and less than the setting value of the GTPR register

For details, refer to section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 15; m = A, B).

(2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode

Set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRA register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA register = 0000 0000h] or [GTCCRA register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur. Similarly, set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRB register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB register = 0000 0000h] or [GTCCRB register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC register < GTCCRD register
  GTCCRC register > GTDVU register
  GTCCRD register < GTPR register – GTDVD register
- In down-counting:
  GTCCRC register > GTCCRD register
  GTCCRC register < GTPR register – GTDVU register
  GTCCRD register > GTDVD register

(4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC register < GTCCRD register < GTPR register
- In down-counting: GTPR register > GTCCRC register > GTCCRD register > 0

Similarly, GTCCRE and GTCCRF registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE register < GTCCRF register < GTPR register
- In down-counting: GTPR register > GTCCRE register > GTCCRF register > 0

(5) In Sawtooth-Wave PWM Mode 1

The GTCCRA register should be set with the range of 0000 0000h < GTCCRA register < GTPR register. If GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is set, a compare match occurs within the cycle only when GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is satisfied. If GTCCRA register > GTPR register is set, no compare match occurs.
Similarly, GTCCRB register should be set with the range of 0000 0000h < GTCCRB register < GTPR register. If GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is set, a compare match occurs within the cycle only when GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is satisfied. If GTCCRB register > GTPR register is set, no compare match occurs.

(6) In Complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In Complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.
In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

24.10.3 Range of Settings for the GTPBR and GTPDBR Registers in the Complementary PWM Mode

When buffer transfer with the use of the GTPR register is to proceed at the end of a crest section in complementary PWM mode 1, 3, or 4, set the GTPBR and GTPDBR registers so that the GTPR register is within the following range after the transfer.
The range consists of values no less than the GTCNT counter value of the master channel at the end of a crest section, that is, the GTPR register before the transfer – the GTDVU register (GTPBR register ≥ GTPR register – GTDVU register, GTPDBR register ≥ GTPR register – GTDVU register)
When buffer transfer with the use of the GTPR register is to proceed at the end of a trough section or on counter clearing, no restriction applies to the ranges of settings for the GTPBR and GTPDBR registers.

24.10.4 Setting Range of the GTCNT Counter

Other than the sawtooth-wave PWM mode 2 and complementary PWM mode, set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

24.10.5 The GTCNT Counter Start/Stop

The GTCNT counter start/stop control by the GTCR.CST bit is synchronized with the count clock selected by the GTCR.TPCS[3:0] bits. Since the GTCNT counter starts or stops after one count clock cycle selected by the TPCS[3:0] bits following the CST bit updating, events until the GTCNT counter actually starts are ignored, and events may be generated after the CST bit becomes 0.
Similarly, when synchronizing with the timer-internal synchronization signal of the timer-linked synchronization signal output function, since the GTCNT counter starts or stops after timer-internal synchronization signal selected by the GTSYNCFCR.SYST bits, events until the GTCNT counter actually starts are ignored, and events may be accepted and the interrupt may be generated after the GTCR.CST bit becomes 0.

24.10.6 Order of Priority in Events

(1) GTCNT Counter

Order of priority in events to update the GTCNT counter is shown below.

Table 24.44 Order of Priority in Updating GTCNT Counter

The GTCNT counter updating source	Order of priority
CPU writing (GTCNT counter writing/GTCLR register writing)	High
Clearing by a hardware source set by the GTCSR register	↑
Up-counting/down-counting by a hardware source set by the GTUPSR and GTDNSR registers	↑
Count operation	Low

In case of contention between up-counting by the GTUPSR register and down-counting by the GTDNSR register, the counter value is not updated.
When updating of the GTCNT counter conflicts with reading by the CPU, the value before updating is reflected.

(2) The GTCR.CST Bit

In case that start/stop by a hardware source set by the GTSSR and GTPSR registers conflicts with the CPU writing (GTCR register writing/GTSTR register writing /GTSTP register writing), the CPU writing takes priority.
In case that stop by the cycle count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the cycle count operation is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.
In case of contention between start by a hardware source set by the GTSSR register and stop by a hardware source set by the GTPSR register, the state of the CST bit does not change.
In case of contention between the CST bit updating and the CPU reading (GTCR register reading/GTSTR register reading /GTSTP register reading), data before updating is read.

(3) GTCCRm Register (m = A to F)

In case of contention between the GTCCRm register writing and an input capture/buffer transfer, the GTCCRm register writing takes priority over an input capture/buffer transfer.
In case that an input capture conflicts with the CPU writing or the counter updating by a hardware source, the counter value before updating is captured.
In case of contention between the GTCCRm register updating and the CPU reading, data before updating is read.

(4) GTPR Register

In case of contention between a buffer transfer and the GTPR register writing, the GTPR register writing takes priority over a buffer transfer.
In case of contention between the GTPR register updating and the CPU reading, data before updating is read.

(5) GTADTRm Register (m = A, B)

In case of contention between a buffer transfer and the GTADTRm register writing, the GTADTRm register writing takes priority over a buffer transfer.
In case of contention between the GTADTRm register updating and the CPU reading, data before updating is read.

(6) GTDVm Register (m = U, D)

In case of contention between a buffer transfer and the GTDVm register writing, the GTDVm register writing takes priority over a buffer transfer.
In case of contention between the GTDVm register updating and the CPU reading, data before updating is read.

(7) GTIOR.GTIOm bit (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm bit, writing to GTIOR.GTIOm bit has priority over buffer transfer operation.
When there is a conflict between updating the GTIOR.GTIOm bit and reading by the CPU, pre-update data is read.

24.10.7 Note on Counter Clearing in the Complementary PWM Mode

Counter clearing at the end of a trough section or the end of the initial output section must be avoided in the complementary PWM mode. This can be achieved by using synchronous clearing in response to a match in comparison in another channel set as the source of the trigger, as described in section 24.3.9.3, Synchronous Clear Operation by Inter Channel Cooperation. When using counter clearing other than as stated above, adjust the timing of the operations to avoid counter clearing at the end of a trough section or the end of the initial output section.

24.10.8 Note on Disabling PWM Initial Output After Synchronous Clearing in the Complementary PWM Mode

When initial output on the GTIOCnA and GTIOCnB pins after synchronous clearing in a trough section in the complementary PWM mode has been disabled by setting the GTIOR.CPSCIR bit to 1, the respective values set in the compare match registers (the GTCCRA, GTCCRC, GTCCRD,GTCCRE, and GTCCRF registers) must be more than twice that of the GTDVU register.

24.10.9 Correspondence to Count Switch Operation of Each Function

Some functions cannot be used at the same time as the count switch operation. Table 24.45 shows the availability of each function at the same time as the count switch operation.

Table 24.45 Availability of Each Function at the Same Time as the Count Switch Operation (1/2)

Function	Availability at the same time as the count switch operation
Counter Start/Count Stop	Available
Periodic Count Operation	Available
Event Count Operation	Not available
Counter Clear Operation	Available
Compare Match Function	Available
Input Capture Function	Available
GTIOCnA/B input/output pin polarity reversal function	Available
Buffer Operation	Available
Sawtooth-Wave PWM Mode 1	Not available
Sawtooth-Wave PWM Mode 2	Not available

Table 24.45 Availability of Each Function at the Same Time as the Count Switch Operation (2/2)

Function	Availability at the same time as the count switch operation
Sawtooth-Wave One-Shot Pulse Mode	Not available
Triangle-Wave PWM Mode 1	Available
Triangle-Wave PWM Mode 2	Available
Triangle-Wave PWM Mode 3	Available
Complementary PWM Mode 1	Not available
Complementary PWM Mode 2	Not available
Complementary PWM Mode 3	Not available
Complementary PWM Mode 4	Not available
Automatic Dead Time Setting Function	Not available
Count Direction Changing Function	Not available
Function of Output Duty 0% and 100%	Not available
Hardware Count Start/Count Stop and Clear Operation	Available
Synchronous Operation	Available
PWM Output Operation	Available
Cycle Count Function	Not available
Phase Counting Mode	Not available
External pulse width measuring function	Not available
Output Phase Switching (OPS)	Not available
Inter Channel Logical Operation Function	Available
Timer-linked synchronization signal output function	Not available
Interrupt Sources	Available
DMAC/DTC Activation	Available
Interrupt Skipping Function	Not available
A/D Conversion Start Request	Available
Operations Linked by the ELC	Available
Noise Filter Function	Available
Write-Protection for Registers	Available
Simultaneous Control of Buffer Operations of Multiple Channels	Available
Repeated Double-Buffered Operation	Not available
GTIOCnm Pin Output Negate Control	Available
Output Protection Function for GTIOCnm Pin Output	Not available
Initialization Method of Output Pins	Available

24.10.10 Note on Transitions to Software Standby Mode

Only place the MCU in software standby mode after having ensured that no counter start source events will be generated and then stopped counting by setting the GTCR.CST bit to 0.

24.10.11 Interval between Interrupt Requests

When GPTCLK is selected as the GPTW counter reference clock (GTCLK), interrupts may be lost if the interval between the interrupt signals from the same source is shorter than the value obtained from the equation below. Note that this restriction does not apply to interrupt signals from different sources.

Interrupt interval (ns) = GPTCLK period (ns) × 6 + PCLKA period (ns) × 4

In addition, when the combination of the selected GPTW and ADC clocks is any of those indicated in Table 24.46, A/D conversion start requests may be lost if the interval between the A/D conversion start requests due to the same event is shorter than the value obtained from the equation below. Note that this restriction does not apply to requests due to different events.

Event interval (ns) = GPTCLK period (ns) × 6 + ADCLK period (ns) × 4

Table 24.46 Clock Combinations that Impose a Restriction on the Event Interval

GTCLK	ADCLK
GPTCLK	PCLKA or ADCCLK
PCLKD	ADCCLK or GPTCLK

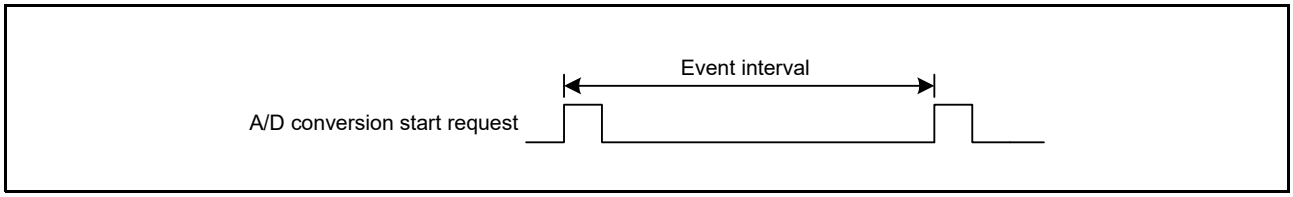


Figure 24.241 Example of an Interval between A/D Conversion Start Request Events

25. High Resolution PWM Waveform Generation Circuit (HRPWMb)

25.1 Overview

This MCU has a high resolution PWM waveform generation circuit (HRPWM0) which finely adjusts a PWM waveform generated by GPTW0 to GPTW3 with a resolution of at least around 49 ps. Using the HRPWM, a digital power supply control with high conversion efficiency can be realized. Table 25.1 lists the specifications of the HRPWM, Figure 25.1 shows a block diagram, and Table 25.2 lists the I/O pins.

Table 25.1 Specifications of the HRPWM

Item	Description
Function	• A maximum of four channels of complementary PWM waveforms are finely adjusted at high resolution. • Using a delay locked loop (DLL) circuit, a high resolution of 1/64 or 1/128 times of a GTCLK period (minimum of around 49 ps) is achieved. • Timing for rising and falling of the PWM waveform can be adjusted individually.*1 • A waveform generated by the GPTW can be output directly by bypassing the HRPWM.
Operating clock (GTCLK)	Either PCLKD or GPTCLK can be selected by setting the GPTW register.
Operating frequency (f(GTCLK))	80 to 200 MHz
Protect Zone filter	Security attribute can be set.*2

Note 1. Adjustment of timing for rising and falling of PWM signals which are output from the GPTW0 to GPTW3 is realized by a delay of a resolution of 1/64 or 1/128 times of a GTCLK period.

Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

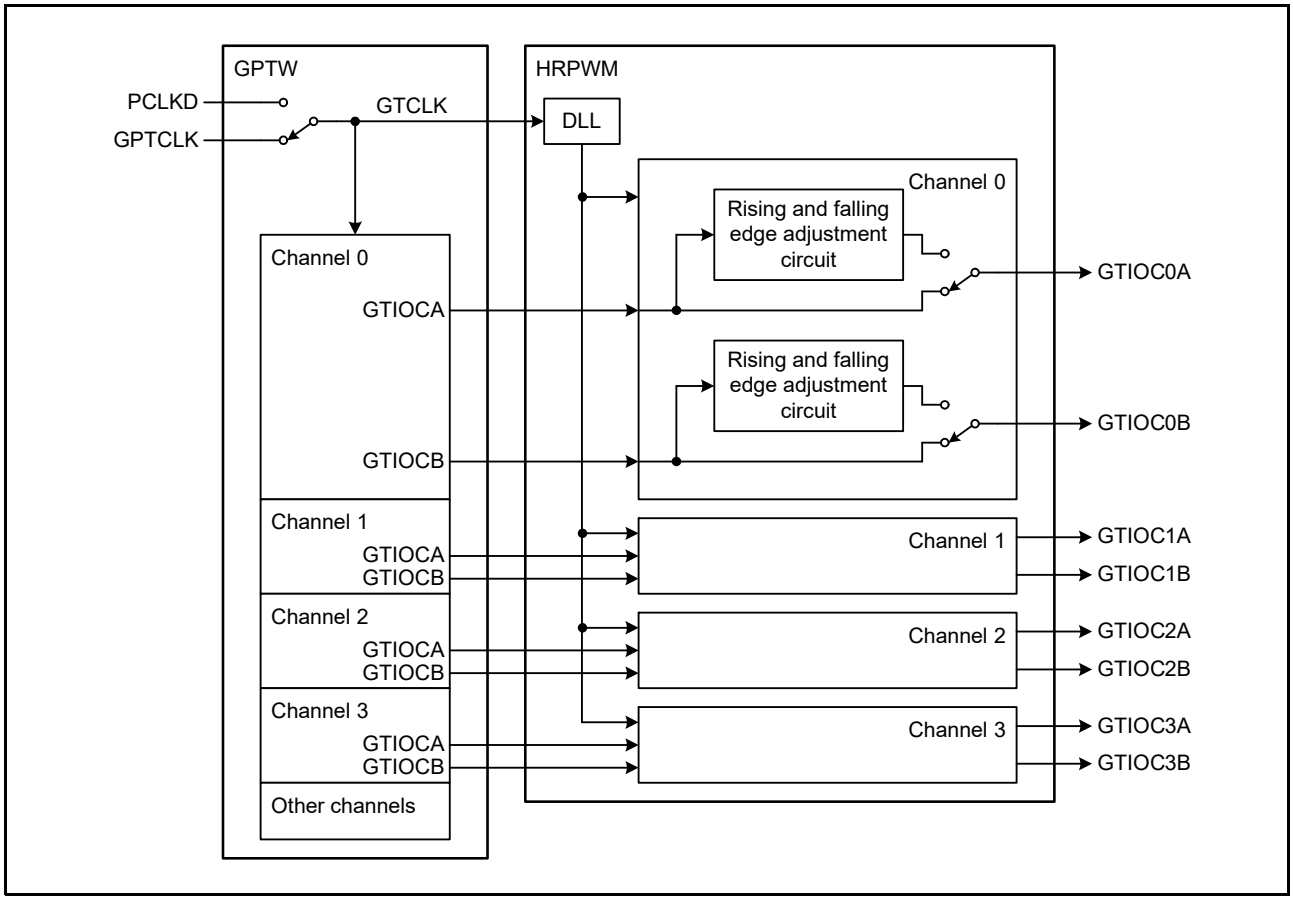


Figure 25.1 Block Diagram of the HRPWM

Table 25.2 I/O Pins of the HRPWM

Pin Name	I/O	Function
GTIOC0A	Output	High resolution output of a PWM waveform generated by the GPTW0
GTIOC0B	Output	High resolution output of a PWM waveform generated by the GPTW0
GTIOC1A	Output	High resolution output of a PWM waveform generated by the GPTW1
GTIOC1B	Output	High resolution output of a PWM waveform generated by the GPTW1
GTIOC2A	Output	High resolution output of a PWM waveform generated by the GPTW2
GTIOC2B	Output	High resolution output of a PWM waveform generated by the GPTW2
GTIOC3A	Output	High resolution output of a PWM waveform generated by the GPTW3
GTIOC3B	Output	High resolution output of a PWM waveform generated by the GPTW3

25.2 Register Descriptions

25.2.1 HRPWM Operation Control Register (HROCR)

Address(es): HRPWM0.HROCR 8742 4000h, HRPWM0_NS.HROCR 0742 4000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FRANGE	—	—	—	—	—	—	HRRST	DLLEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLLEN	DLL Operation Enable	0: DLL operation is disabled 1: DLL operation is enabled	R/W
b1	HRRST	High Resolution PWM Waveform Generation Circuit Reset	0: Release from reset 1: Reset	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FRANGE	Operating Clock Frequency Range Setting	0: 80 to 160 MHz (resolution: 1/128 times of the period) 1: 155 to 200 MHz (resolution: 1/64 times of the period)	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the operation of the entire HRPWM module. The register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTW0.GTWP.WP = 0).

DLLEN Bit (DLL Operation Enable)

This bit selects whether the internal DLL in the HRPWM is activated or not. After activation, wait for 20 µs until the output of the DLL becomes stable.

HRRST Bit (High Resolution PWM Waveform Generation Circuit Reset)

This bit resets the internal state of the HRPWM. After releasing from reset, wait for five cycles of GTCLK. When resetting during operation, the PWM output is fixed to low after five cycles of GTCLK.

FRANGE Bit (Operating Clock Frequency Range Setting)

This bit is used to set the frequency range of the HRPWM operating clock. Set this bit when the DLLEN bit is 0. The resolution of the delay time is determined by the setting of this bit.

25.2.2 HRPWM Operation Control Register 2 (HROCR2)

Address(es): HRPWM0.HROCR2 8742 4002h, HRPWM0_NS.HROCR2 0742 4002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HRDIS3	HRDIS2	HRDIS1	HRDIS0	—	—	—	—	HRSEL3	HRSEL2	HRSEL1	HRSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HRSEL0	Channel 0 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW0 from the GTIOC0A and GTIOC0B pins 1: Output a high resolution PWM waveform from the GTIOC0A and GTIOC0B pins	R/W
b1	HRSEL1	Channel 1 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW1 from the GTIOC1A and GTIOC1B pins 1: Output a high resolution PWM waveform from the GTIOC1A and GTIOC1B pins	R/W
b2	HRSEL2	Channel 2 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW2 from the GTIOC2A and GTIOC2B pins 1: Output a high resolution PWM waveform from the GTIOC2A and GTIOC2B pins	R/W
b3	HRSEL3	Channel 3 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW3 from the GTIOC3A and GTIOC3B pins 1: Output a high resolution PWM waveform from the GTIOC3A and GTIOC3B pins	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	HRDIS0	Channel 0 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 0 rising and falling edge adjustment circuit 1: Disable operation of Channel 0 rising and falling edge adjustment circuit	R/W
b9	HRDIS1	Channel 1 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 1 rising and falling edge adjustment circuit 1: Disable operation of Channel 1 rising and falling edge adjustment circuit	R/W
b10	HRDIS2	Channel 2 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 2 rising and falling edge adjustment circuit 1: Disable operation of Channel 2 rising and falling edge adjustment circuit	R/W
b11	HRDIS3	Channel 3 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 3 rising and falling edge adjustment circuit 1: Disable operation of Channel 3 rising and falling edge adjustment circuit	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the HRPWM module by channel. The register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTW0.GTWP.WP = 0).

HRSELn Bit (Channel n High Resolution PWM Waveform Output Select) (n = 0 to 3)

This bit allows to select whether to output a waveform generated by the HRPWM from the GTIOCnA and GTIOCnB pins or to output an original waveform of the GPTW by bypassing the circuit.
The PWM waveform generated by the HRPWM is output with a delay of three full cycles of GTCLK from the output when the circuit is bypassed, even if the setting for the amount of delay is 0.

HRDISn Bit (Channel n Rising and Falling Edge Adjustment Circuit Disable) (n = 0 to 3)

This bit enables/disables the operation of the rising and falling edge adjustment circuit by relevant channel in the GPTW.

Set this bit to 1 for a channel which will not be used.

25.2.3 GTIOCnA Pin Rising Edge Adjustment Register (HRREARnA) (n = 0 to 3)

Address(es): HRPWM0.HRREAR0A 8742 4018h, HRPWM0.HRREAR1A 8742 401Ch, HRPWM0.HRREAR2A 8742 4020h, HRPWM0.HRREAR3A 8742 4024h,
HRPWM0_NS.HRREAR0A 0742 4018h, HRPWM0_NS.HRREAR1A 0742 401Ch, HRPWM0_NS.HRREAR2A 0742 4020h, HRPWM0_NS.HRREAR3A 0742 4024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DLY[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	DLY[6:0]	Amount of Delay Select	• When the HROCR.FRANGE bit is 0 b6 b0 0000000: Apply delay of 0/128 times GTCLK period 0000001: Apply delay of 1/128 times GTCLK period 0000010: Apply delay of 2/128 times GTCLK period : : 1111110: Apply delay of 126/128 times GTCLK period 1111111: Apply delay of 127/128 times GTCLK period • When the HROCR.FRANGE bit is 1 b6 b0 0000000: Apply delay of 0/64 times GTCLK period 0000010: Apply delay of 1/64 times GTCLK period 0000100: Apply delay of 2/64 times GTCLK period : : 1111100: Apply delay of 62/64 times GTCLK period 1111110: Apply delay of 63/64 times GTCLK period xxxxxx1: Setting prohibited	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to rising edge of a PWM waveform output from the GTIOCnA pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).
This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0). After writing this register, confirm that the value has actually changed before proceeding with further processing.

25.2.4 GTIOCnA Pin Falling Edge Adjustment Register (HRFEARnA) (n = 0 to 3)

Address(es): HRPWM0.HRFEAR0A 8742 4028h, HRPWM0.HRFEAR1A 8742 402Ch, HRPWM0.HRFEAR2A 8742 4030h, HRPWM0.HRFEAR3A 8742 4034h,
HRPWM0_NS.HRFEAR0A 0742 4028h, HRPWM0_NS.HRFEAR1A 0742 402Ch, HRPWM0_NS.HRFEAR2A 0742 4030h, HRPWM0_NS.HRFEAR3A 0742 4034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DLY[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	DLY[6:0]	Amount of Delay Select	• When the HROCR.FRANGE bit is 0 b6 b0 0000000: Apply delay of 0/128 times GTCLK period 0000001: Apply delay of 1/128 times GTCLK period 0000010: Apply delay of 2/128 times GTCLK period : : 1111110: Apply delay of 126/128 times GTCLK period 1111111: Apply delay of 127/128 times GTCLK period • When the HROCR.FRANGE bit is 1 b6 b0 0000000: Apply delay of 0/64 times GTCLK period 0000010: Apply delay of 1/64 times GTCLK period 0000100: Apply delay of 2/64 times GTCLK period : : 1111100: Apply delay of 62/64 times GTCLK period 1111110: Apply delay of 63/64 times GTCLK period xxxxxx1: Setting prohibited	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to falling edge of a PWM waveform output from the GTIOCnA pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0). After writing this register, confirm that the value has actually changed before proceeding with further processing.

25.2.5 GTIOCnB Pin Rising Edge Adjustment Register (HRREARnB) (n = 0 to 3)

Address(es): HRPWM0.HRREAR0B 8742 401Ah, HRPWM0.HRREAR1B 8742 401Eh, HRPWM0.HRREAR2B 8742 4022h, HRPWM0.HRREAR3B 8742 4026h,
HRPWM0_NS.HRREAR0B 0742 401Ah, HRPWM0_NS.HRREAR1B 0742 401Eh, HRPWM0_NS.HRREAR2B 0742 4022h, HRPWM0_NS.HRREAR3B 0742 4026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DLY[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	DLY[6:0]	Amount of Delay Select	• When the HROCR.FRANGE bit is 0 b6 b0 0000000: Apply delay of 0/128 times GTCLK period 0000001: Apply delay of 1/128 times GTCLK period 0000010: Apply delay of 2/128 times GTCLK period : : 1111110: Apply delay of 126/128 times GTCLK period 1111111: Apply delay of 127/128 times GTCLK period • When the HROCR.FRANGE bit is 1 b6 b0 0000000: Apply delay of 0/64 times GTCLK period 0000010: Apply delay of 1/64 times GTCLK period 0000100: Apply delay of 2/64 times GTCLK period : : 1111100: Apply delay of 62/64 times GTCLK period 1111110: Apply delay of 63/64 times GTCLK period xxxxxx1: Setting prohibited	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to rising edge of a PWM waveform output from the GTIOCnB pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0). After writing this register, confirm that the value has actually changed before proceeding with further processing.

25.2.6 GTIOCnB Pin Falling Edge Adjustment Register (HRFEARnB) (n = 0 to 3)

Address(es): HRPWM0.HRFEAR0B 8742 402Ah, HRPWM0.HRFEAR1B 8742 402Eh, HRPWM0.HRFEAR2B 8742 4032h, HRPWM0.HRFEAR3B 8742 4036h, HRPWM0_NS.HRFEAR0B 0742 402Ah, HRPWM0_NS.HRFEAR1B 0742 402Eh, HRPWM0_NS.HRFEAR2B 0742 4032h, HRPWM0_NS.HRFEAR3B 0742 4036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DLY[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	DLY[6:0]	Amount of Delay Select	• When the HROCR.FRANGE bit is 0 b6 b0 0000000: Apply delay of 0/128 times GTCLK period 0000001: Apply delay of 1/128 times GTCLK period 0000010: Apply delay of 2/128 times GTCLK period : : 1111110: Apply delay of 126/128 times GTCLK period 1111111: Apply delay of 127/128 times GTCLK period • When the HROCR.FRANGE bit is 1 b6 b0 0000000: Apply delay of 0/64 times GTCLK period 0000010: Apply delay of 1/64 times GTCLK period 0000100: Apply delay of 2/64 times GTCLK period : : 1111100: Apply delay of 62/64 times GTCLK period 1111110: Apply delay of 63/64 times GTCLK period xxxxxx1: Setting prohibited	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to falling edge of a PWM waveform output from the GTIOCnB pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0). After writing this register, confirm that the value has actually changed before proceeding with further processing.

25.3 Operation

25.3.1 Adjustment to the Timing of Rising and Falling Edges in PWM Waveforms

The HRPWM delays timing for rising and falling of the PWM signals which are output from the GPTW with a resolution of 1/64 or 1/128 times (at least around 49 ps) of a GTCLK period (tC(GTCLK)) based on the delay locked loop (DLL) operating with GTCLK as a reference clock. The PWM signal which was delayed is output from the GTIOCnA and GTIOCnB pins (n = 0 to 3).

The amount of delay can be adjusted within a range of 0/64 to 63/64 or 0/128 to 127/128 of tC(GTCLK). The amount of delay out of this range cannot be specified. When setting the amount of delay with the value out of such range, change the setting in the general PWM timer compare capture register x (GTCCRx) in the GPTW.

The HRPWM is required to set the HROCR.DLLEN bit to 1 before operation and wait for stabilization time as it has the DLL. In addition, wait for five cycles of GTCLK after setting the HROCR.HRRST bit to 0 in order to output the reliable PWM waveform. Refer to Figure 25.2 for the detailed procedures.

Specify the amount of delay by the HRREARnA and HRFEARnA registers for the GTIOCnA pin and by the HRREARnB and HRFEARnB registers for the GTIOCnB pin. The PWM waveform has specific setting registers for delay at rising and at falling respectively, which allows fine adjustments for the width of high and low. Refer to Table 25.3 for the correspondence between pins and registers.

The amount of delay specified by a relevant register is reflected to the pin via a temporary register. Timing for reflecting the setting differs by PWM output operation mode in the general PWM timer. Refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

In the HRPWM, setting of DLL enabling/disabling and reset setting/releasing is common for all channels; however, the rest of settings can be specified individually by channel.

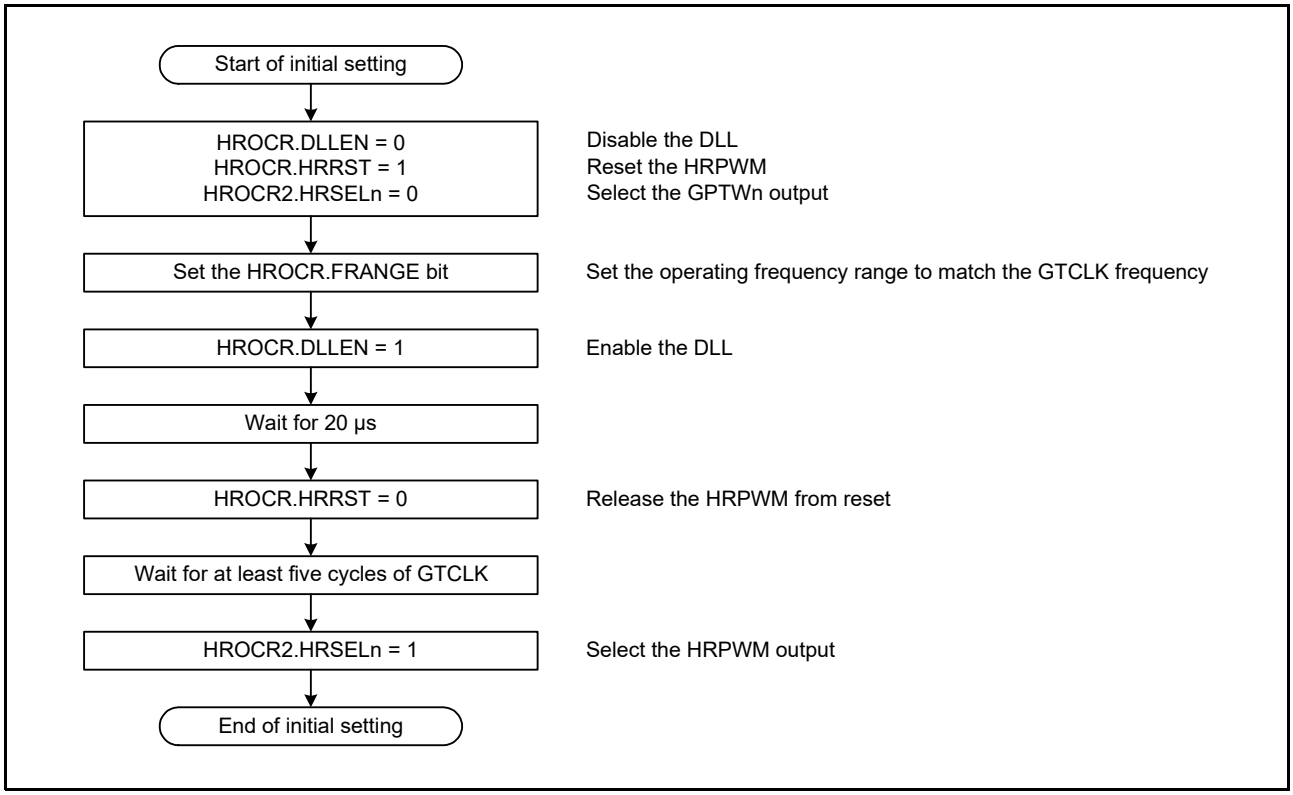


Figure 25.2 Initialization flow for the HRPWM (n = 0 to 3)

Table 25.3 Correspondence between PWM Waveform Output Pins and Delay Setting Registers

PWM Waveform Output Pin	Rising Edge Adjustment Register	Falling Edge Adjustment Register
GTIOC0A	HRREAR0A	HRFEAR0A
GTIOC0B	HRREAR0B	HRFEAR0B
GTIOC1A	HRREAR1A	HRFEAR1A
GTIOC1B	HRREAR1B	HRFEAR1B
GTIOC2A	HRREAR2A	HRFEAR2A
GTIOC2B	HRREAR2B	HRFEAR2B
GTIOC3A	HRREAR3A	HRFEAR3A
GTIOC3B	HRREAR3B	HRFEAR3B

25.3.2 Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3)

Settings of the HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB registers are transferred to a temporary register first, and reflected to the amount of delay for the PWM waveform in the GTIOCnA and GTIOCnB pins then. Such settings are transferred to a temporary register at the end of PWM period (at an overflow (during up-count operation) or an underflow (during down-count operation) in sawtooth wave mode, and at a trough in triangle wave mode).

Figure 25.3 and Figure 25.4 show examples of the operation of the HRREAR0A and HRFEAR0A registers.

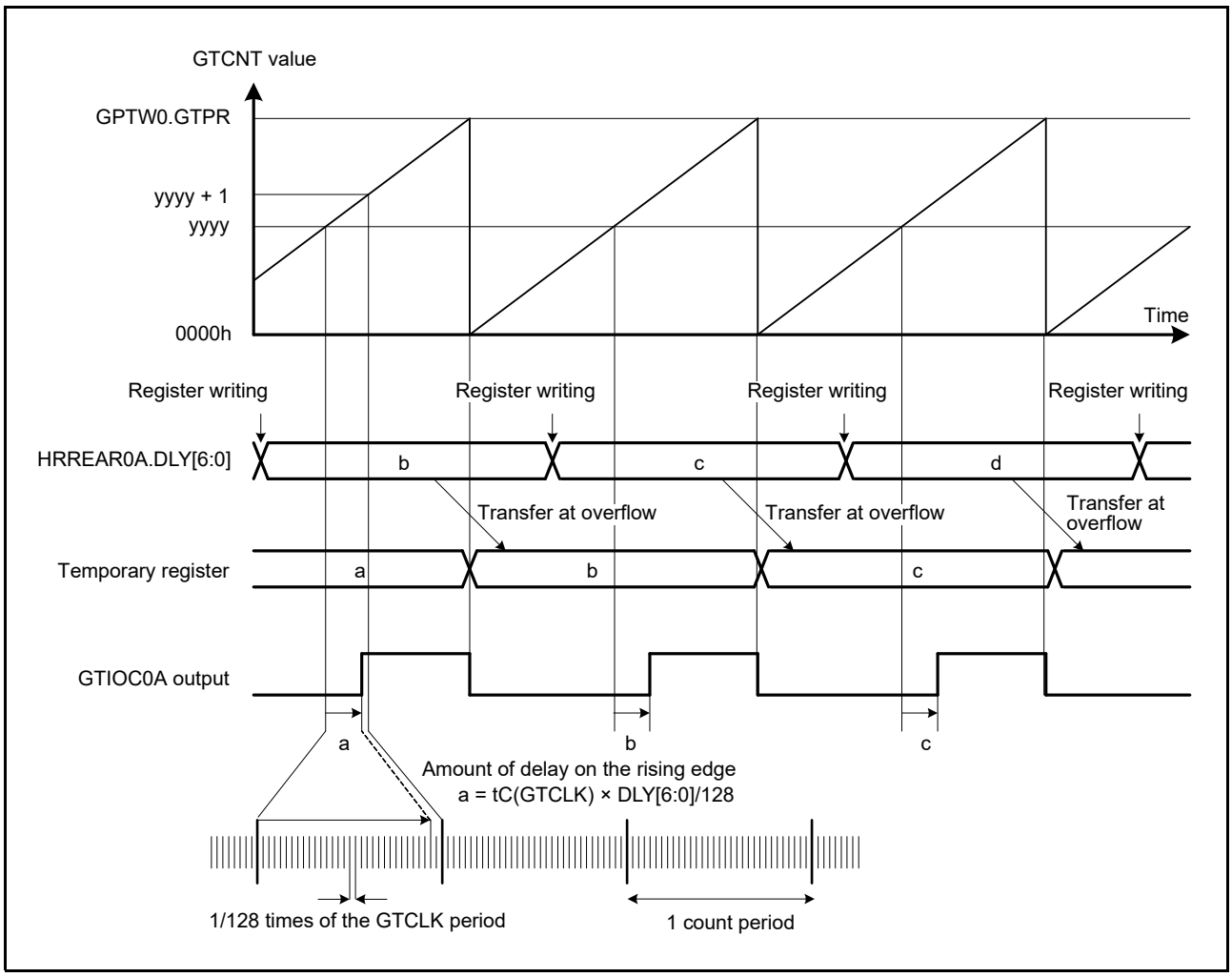


Figure 25.3 Operation Example When Changing the HRREAR0A Register Setting (in Sawtooth Wave Mode)

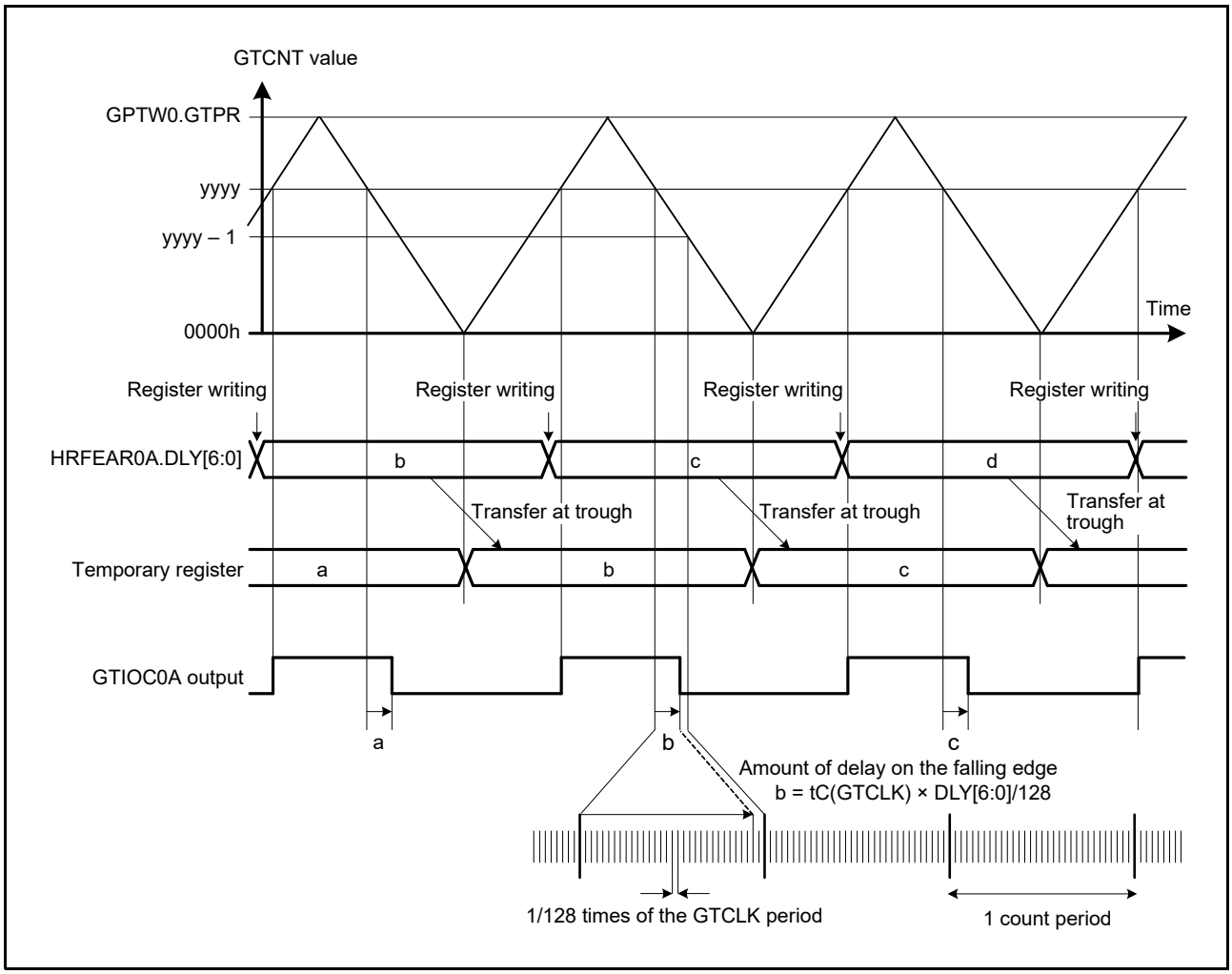


Figure 25.4 Operation Example When Changing the HRFEAR0A Register Setting (in Triangle Wave Mode)

25.4 Usage Notes

25.4.1 Settings for the Module-Stop Function

Operation of the HRPWM can be disabled or enabled using Module Stop Control Register (MSTPCRD). The HRPWM is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, refer to section 11, Low Power Consumption.

25.4.2 Note on Selecting Timer Prescaler in the GTCNT Counter While Using the HRPWM

The PWM signals output from the GPTW are generated based on the GTCNT counter operating at resolution equivalent to one clock period which is set by the GPTWn.GTCR.TPCS[3:0] bits (n = 0 to 3). Note that a delay at resolution of 0/128 × tC(GTCLK) to 127/128 × tC(GTCLK) is applied to edges of the PWM signals at low resolution generated based on a selected clock when a setting for the GPTWn.GTCCRx register (x = A to F) is changed while the TPCS[3:0] bits are specified to the setting other than 0000b (GTCLK/1).

25.4.3 Notes on Delay Settings for HRPWM

Within three cycles of GTCLK from the end of the PWM period, do not change the settings for the HRFEARnA, HRREARnA, HRFEARnB, and HRREARnB registers. Any changes made during the above period will be reflected with a delay of one cycle of the PWM waveform.

26. GPTW Port Output Enable (POEGd)

26.1 Overview

The POEG issues requests to disable output from output pins of the general PWM timer (GPTW). The combination of output pins of the POEG to be disabled can be specified from any channel of the GPTW.
Select the method of detection for disabling the output from the list below.

- Input level detection on the GTETRGx pin (x = A to D)
- Detection of input level or input edge on the POEGx pin (x = A to D)
- Output disabling condition detection by the GPTW
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGx pins can also be used as the external trigger input pins for the GPTW.
Table 26.1 shows specifications, Figure 26.1 shows a system block diagram around the POEG, Figure 26.2 shows a block diagram of the POEG, and Table 26.2 shows input pins.

Table 26.1 POEG Specifications (x = A to D)

Item	Description
Output disabling request by input level detection on the GTETRGx pin	• The output disabling request is issued to the GPTW when a POEGGx.PIDF flag is set to 1 by the detection of input of the selected level on the GTETRGx pin.
Output disabling request by detection of input level or edge on the POEGx pin	• The output disabling request is issued to the GPTW when a POEGPICRx.PIDF flag is set to 1 by the detection of either input of the selected level or the selected edge on the POEGx pin.
Output disabling request by output disabling condition detection signal from the GPTW	• The output disabling request is issued to the GPTW when the GPTW detects a simultaneous active level (high or low) on the GTIOCA and GTIOCB pins and the POEGGx.IOCF flag is set to 1. • The output disabling request is issued to the GPTW when the GPTW detects a dead-time error and the POEGGx.IOCF flag is set to 1.
Output disabling request in response to detecting the stopping of oscillation	The output disabling request is issued to the GPTW when the oscillation stop detection circuit for the main clock detects the oscillation stop and the POEGGx.OSTPF flag is set to 1.
Output disabling request by software	The output disabling request is issued to the GPTW when the software sets the POEGGx.SSF bit to 1.
Interrupt	• An interrupt is generated by an output disabling request via the POEGGx.PIDF flag. • An interrupt is generated by an output disabling request via the POEGGx.IOCF flag. • An interrupt is generated by an output disabling request via the POEGPICRx.PIDF flag.
External trigger output to the GPTW	The input signal from the GTETRGx pin is output to the GPTW as an external trigger.
Noise removal	Each GTETRGx pin has a digital noise filter. • Four types of sampling clock are selectable for the filter. Each POEGx pin has a digital noise filter. • Six types of sampling clock are selectable for the filter. • Match detection conditions can be selected from two to 64 consecutive samples with the same level.
Protect Zone filter	Security attribute can be set for each group of the POEG.*1

Note 1. When each group is set to the secure attribute, registers in the group are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each group is set to the non-secure attribute, registers in the group are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

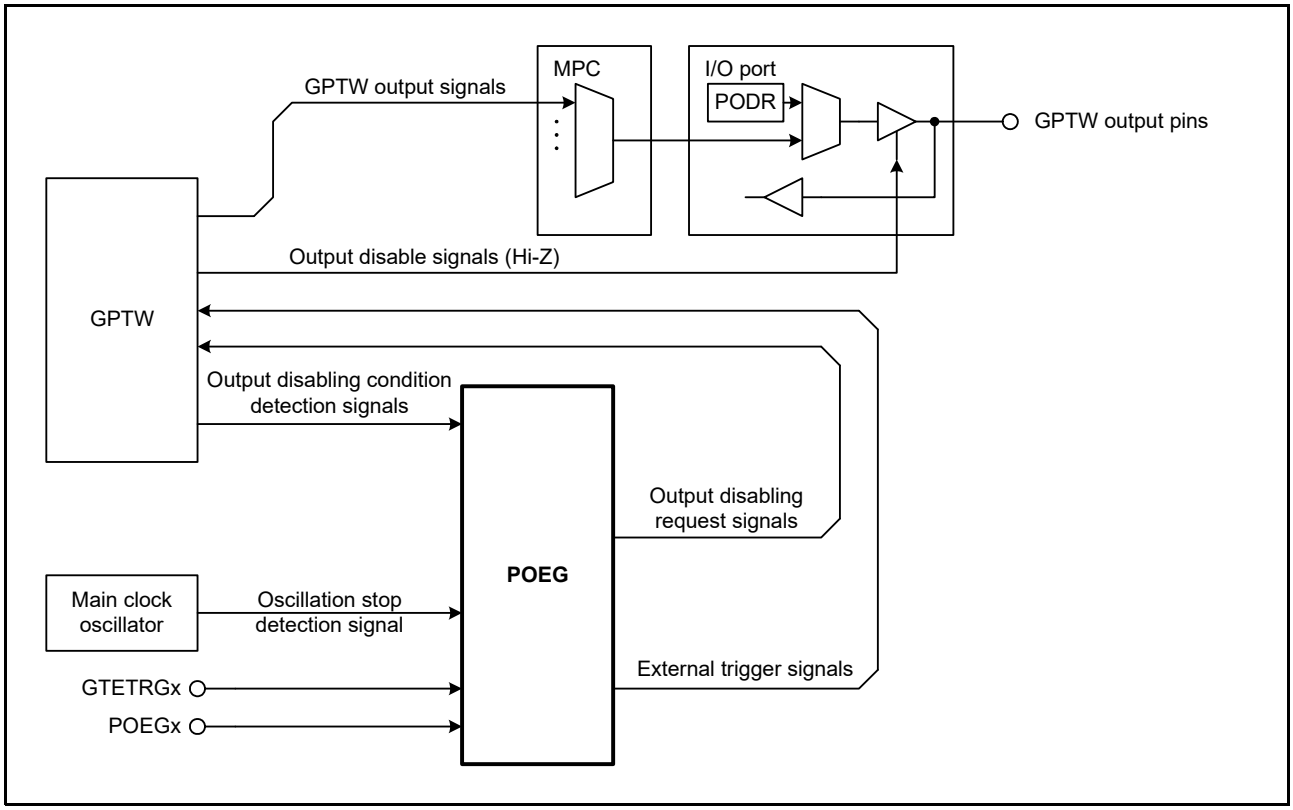


Figure 26.1 POEG System Block Diagram

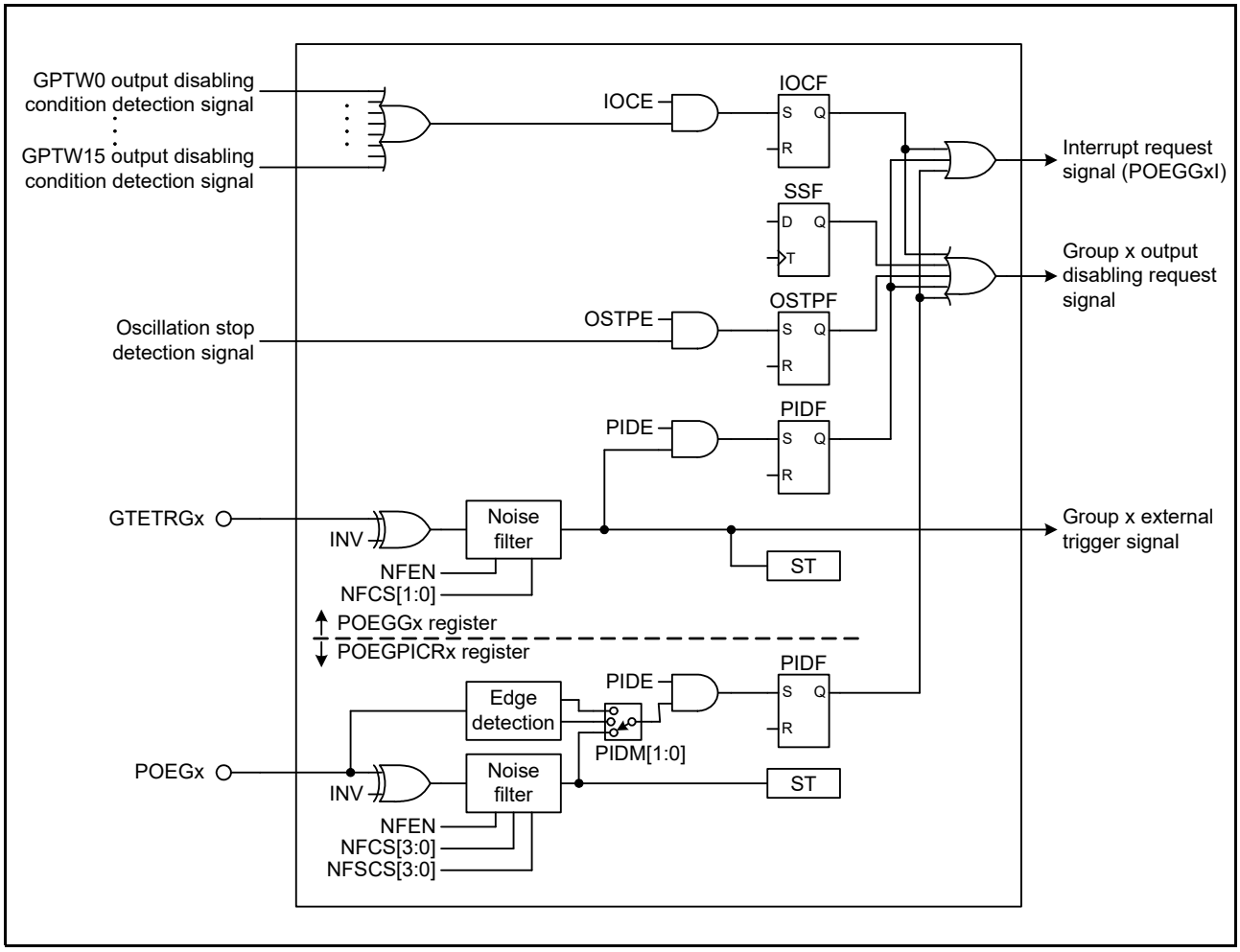


Figure 26.2 POEG Block Diagram (x = A to D)

Table 26.2 POEG I/O Pins

Pin Name	I/O	Function
GTETRGA	Input	Input pin A for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGB	Input	Input pin B for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGC	Input	Input pin C for output disable request signal for GPTW output pin and GPTW external trigger
GTETRGD	Input	Input pin D for output disable request signal for GPTW output pin and GPTW external trigger
POEGA	Input	Input pin A for output disable request signal for GPTW output pin
POEGB	Input	Input pin B for output disable request signal for GPTW output pin
POEGC	Input	Input pin C for output disable request signal for GPTW output pin
POEGD	Input	Input pin D for output disable request signal for GPTW output pin

26.2 Register Descriptions

26.2.1 POEG Group x Setting Register (POEGGx) (x = A to D)

Address(es): POEG.POEGGA 8741 2000h, POEG.POEGGB 8741 2100h, POEG.POEGGC 8741 2200h, POEG.POEGGD 8741 2300h, POEG_NS.POEGGA 0741 2000h, POEG_NS.POEGGB 0741 2100h, POEG_NS.POEGGC 0741 2200h, POEG_NS.POEGGD 0741 2300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: Indicates that an output disabling request by the GTETRGx signal has not been issued. 1: Indicates that an output disabling request by the GTETRGx signal has been issued.	R/(W) *1
b1	IOCF	GPTW Output Disabling Condition Detection Flag	0: Indicates that an output disabling request by the GPTW output disabling condition detection signal has not been issued. 1: Indicates that an output disabling request by the GPTW output disabling condition detection signal has been issued.	R/(W) *1
b2	OSTPF	Oscillation Stop Detection Flag	0: Indicates that an output disabling request by oscillation stop has not been issued. 1: Indicates that an output disabling request by oscillation stop has been issued.	R/(W) *1
b3	SSF	Software Output Disabling Request	0: Does not issue output disabling request by software. 1: Issue an output disabling request by software.	R/W
b4	PIDE	Port Input Signal Enable	0: Output disabling request by the GTETRGx signal is disabled. 1: Output disabling request by the GTETRGx signal is enabled.	R/W*2
b5	IOCE	GPTW Output Disabling Condition Detection Signal Enable	0: Output disabling request by the GPTW output disabling condition detection signal is disabled. 1: Output disabling request by the GPTW output disabling condition detection signal is enabled.	R/W*2
b6	OSTPE	Oscillation Stop Detection Signal Enable	0: Output disabling request by the oscillation stop detection signal is disabled. 1: Output disabling request by the oscillation stop detection signal is enabled.	R/W*2
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGx Signal Status Flag	0: The GTETRGx signal output to the GPTW is low. 1: The GTETRGx signal output to the GPTW is high.	R
b27 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	INV	GTETRGx Input Inverting	0: Does not invert the input signal from the GTETRGx pin. 1: Invert the input signal from the GTETRGx pin.	R/W
b29	NFEN	Noise Filter Enable	0: Digital noise filter on the GTETRGx pin is disabled. 1: Digital noise filter on the GTETRGx pin is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	b31 b30 0 0: The input signal from the GTETRGx pin is sampled every PCLKA/1 clock 0 1: The input signal from the GTETRGx pin is sampled every PCLKA/8 clock 1 0: The input signal from the GTETRGx pin is sampled every PCLKA/32 clock 1 1: The input signal from the GTETRGx pin is sampled every PCLKA/128 clock	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. This bit can be written only once after reset.

The POEGGx register controls output disabling requests and an external trigger for the GPTW in response to the various detection signals.

SSF Bit (Software Output Disabling Request)

Writing 1 to this bit leads to an output disabling request being issued to the GPTW, and writing 0 to the bit releases the GPTW from the output disabling request.

26.2.2 POEGx Pin Input Control Register (POEGPICRx) (x = A to D)

Address(es): POEG.POEGPICRA 8741 2030h, POEG.POEGPICRB 8741 2130h, POEG.POEGPICRC 8741 2230h, POEG.POEGPICRD 8741 2330h, POEG_NS.POEGPICRA 0741 2030h, POEG_NS.POEGPICRB 0741 2130h, POEG_NS.POEGPICRC 0741 2230h, POEG_NS.POEGPICRD 0741 2330h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	NFEN	NFCS[3:0]				PIDM[1:0]		—	—	NFSCS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ST	—	—	—	INV	—	—	—	PIDE	—	—	—	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: Indicates that an output disabling request by the POEGx signal has not been issued. 1: Indicates that an output disabling request by the POEGx signal has been issued.	R/(W) *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PIDE	Port Input Signal Enable	0: Output disabling request by the POEGx signal is disabled. 1: Output disabling request by the POEGx signal is enabled.	R/W*2
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	INV	POEGx Input Inverting	0: Does not invert the input signal from the POEGx pin. 1: Invert the input signal from the POEGx pin.	R/W*2
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ST	POEGx Signal Status Flag	0: The POEGx signal after the noise filter is low. 1: The POEGx signal after the noise filter is high.	R
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	NFSCS[3:0]	Noise Filter Subsequent Match Count Select	b19 b16 0 0 0 0: 2 times 0 0 0 1: 3 times 0 0 1 0: 4 times 0 0 1 1: 5 times 0 1 0 0: 6 times 0 1 0 1: 7 times 0 1 1 0: 8 times 0 1 1 1: 16 times 1 0 0 0: 32 times 1 0 0 1: 64 times Settings other than above are prohibited.	R/W*2
b21, b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23, b22	PIDM[1:0]	POEGx Input Detection Mode Select	b23 b22 0 0: Level 0 1: Rising edge 1 0: Falling edge 1 1: Setting prohibited.	R/W*2

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	NFCS[3:0]	Noise Filter Clock Select	b27 b24 0 0 0 0: The input signal from the POEGx pin is sampled every PCLKA/1 clock 0 0 0 1: The input signal from the POEGx pin is sampled every PCLKA/2 clock 0 0 1 0: The input signal from the POEGx pin is sampled every PCLKA/4 clock 0 0 1 1: The input signal from the POEGx pin is sampled every PCLKA/8 clock 0 1 0 0: The input signal from the POEGx pin is sampled every PCLKA/16 clock 0 1 0 1: The input signal from the POEGx pin is sampled every PCLKA/32 clock Settings other than above are prohibited.	R/W*2
b28	NFEN	Noise Filter Enable	0: Digital noise filter on the POEGx pin is disabled. 1: Digital noise filter on the POEGx pin is enabled.	R/W*2
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. This bit can be written only once after reset.

26.3 Operation

The POEG has four groups (A to D), all of which operate the same and are independent of each other.
The output signal of each channel of the GPTW is connected to all groups of the POEG, and the output signal of each group of the POEG is connected to all channels of the GPTW. Use the GPTW register to configure which group the GPTW channel is controlled by.

26.3.1 Output Disabling Request by Detection of Input Level on the GTETRGx Pin (x = A to D)

The output disabling requests using the GTETRGx pin are issued via the POEGGx.PIDF flag.
When the input level set by the POEGGx.NFCS[1:0], NFEN, and INV bits is detected while the POEGGx.PIDE bit is 1, the PIDF flag is set to 1 and an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, clear the PIDF flag. For details, refer to section 26.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register.

26.3.1.1 Digital Noise Filter

Each GTETRGx pin has a digital noise filter (x = A to D). Figure 26.3 shows an example of the noise filter operation. This example assumes that the POEGGx.INV bit is 0 (no inversion, i.e., high level detection) and NFEN bit is 1 (noise filter enabled).
The signal level on the GTETRGx pin is sampled with the sampling clock selected by the POEGGx.NFCS[1:0] bits, and when last three consecutive samples are high, the POEG determines that high is input and issues an output disabling request to the GPTW.
At this time, if low is detected even once, the POEG does not determine that high is input. Note that changes in the GTETRGx pin level that occur between sampling timings are ignored.
The output of the digital noise filter is used for two purposes: output disabling request via the POEGGx.PIDF flags and an external trigger output to the GPTW.

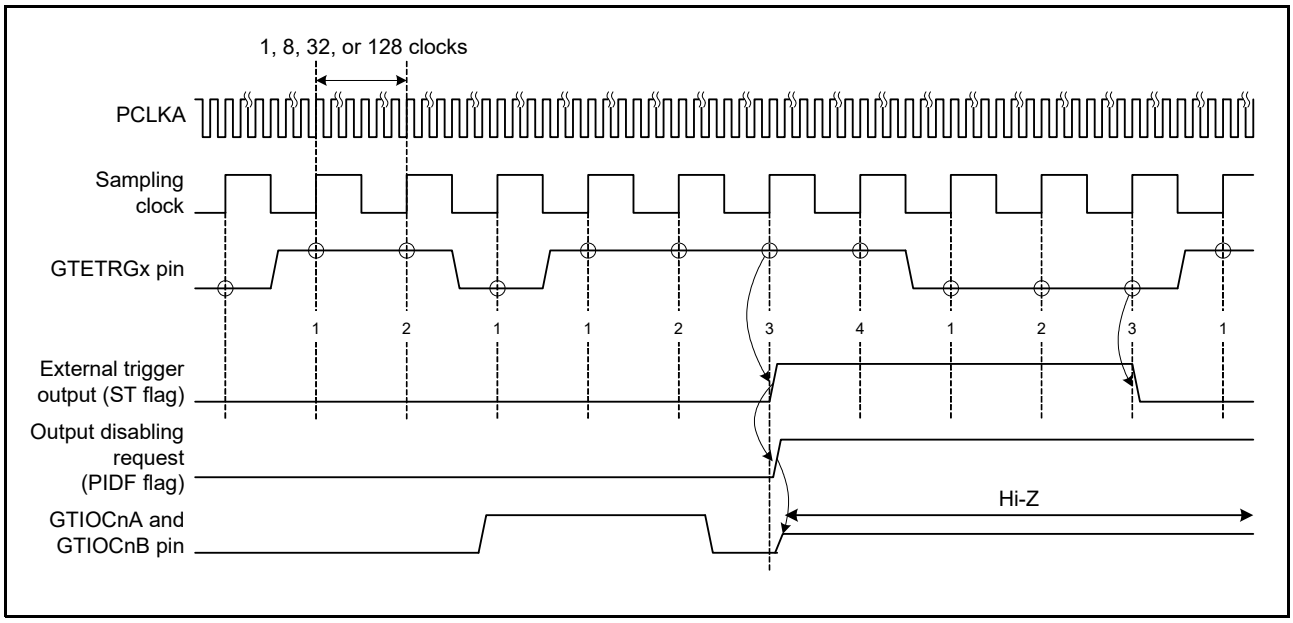


Figure 26.3 Example for Operation of Digital Noise Filter

26.3.2 Output Disabling Request by Detection of Input Level or Input Edge on the POEGx Pin (x = A to D)

The output disabling requests using the POEGx pin are issued via the POEGPICRx.PIDF flag.
When the input level or edge set by the POEGPICRx.NFEN, NFCS[3:0], PIDM[1:0], NFSCS[3:0], and INV bits is detected while the POEGPICRx.PIDE bit is 1, the PIDF flag is set to 1 and an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, clear the PIDF flag. For details, refer to section 26.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register.

26.3.2.1 Digital Noise Filter

Each POEGx pin has a digital noise filter (x = A to D). Figure 26.4 shows an example of the noise filter operation. This example assumes that the POEGPICRx.INV bit is 0 (no inversion, i.e., high level detection), NFSCS[3:0] bits are 0001b (three times), and NFEN bit is 1 (noise filter enabled).
The signal level on the POEGx pin is sampled with the sampling clock selected by the POEGPICRx.NFCS[3:0] bits, and when last consecutive samples for the number of times set by the NFSCS[3:0] bits (three times in this example) are high, the POEG determines that high is input and issues an output disabling request to the GPTW.
At this time, if low is detected even once, the POEG does not determine that high is input. Note that changes in the POEGx pin level that occur between sampling timings are ignored.

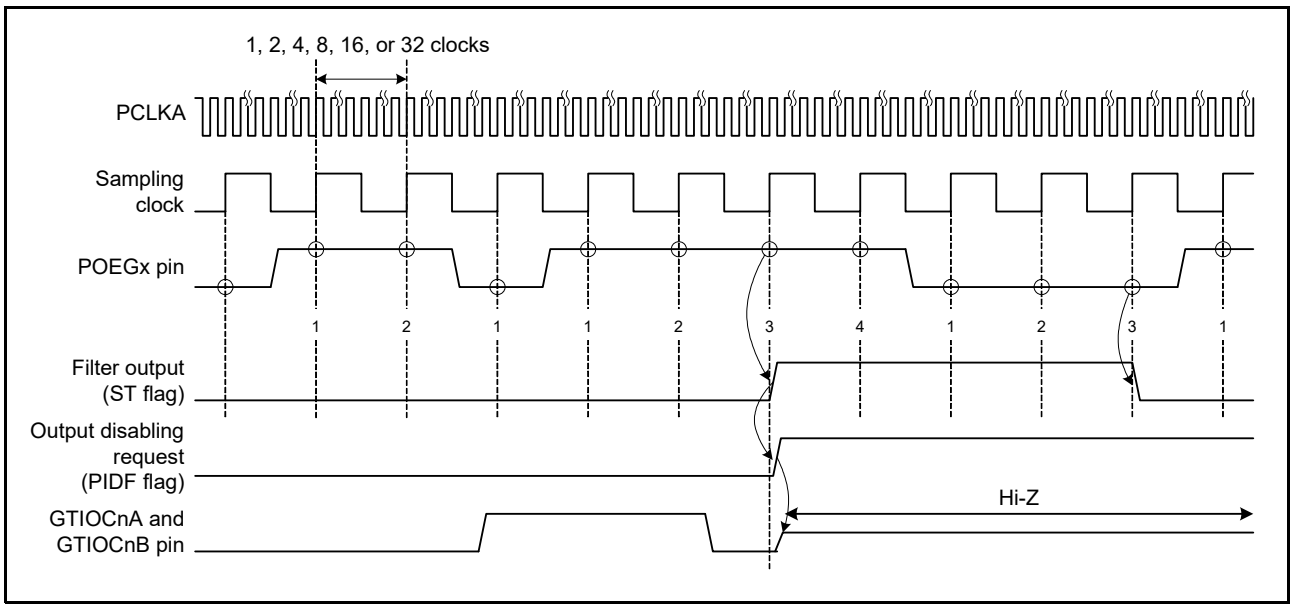


Figure 26.4 Example for Operation of Digital Noise Filter

26.3.3 Output Disabling Request by Output Disabling Condition Detection Signal from GPTW

When the GPTW detects a dead-time error or simultaneous high or low output while the POEGGx.IOCE bit is 1, the IOCF flag is set to 1 and an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, clear the IOCF flag. For details, refer to section 26.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register.
To detect a dead-time error, simultaneous high output, or simultaneous low output in the GPTW, set the GPTWn.GTINTAD.GRPDTE, GRPABH, or GRPABL bit to 1, respectively. The output disabling condition detection signal generated by each GPTW channel is output only to the POEG group selected by the GPTWn.GTINTAD.GRP[1:0] bits. For details, refer to section 24.2.17, General PWM Timer Interrupt Output Setting Register (GTINTAD).

26.3.4 Output Disabling Request by Oscillation Stop Detection Signal

When the oscillation stop detection circuit in the clock generation circuit detects the oscillation stop while the POEGGx.OSTPE bit is 1, the OSTPF flag is set to 1 and an output disabling request is issued to all GPTW channels. To cancel the output disabling request, clear the OSTPF flag. For details, refer to section 26.3.6.2, Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register.

26.3.5 Output Disabling Request by a Register

When the POEGGx.SSF bit is set to 1, an output disabling request is issued to all GPTW channels.
To cancel the output disabling request, set the SSF bit to 0. For details, refer to section 26.3.6, Canceling Output Disabling Request.

26.3.6 Canceling Output Disabling Request

There are two ways to cancel an output disabling request as follows.

- Reset (return to the initial state)
- Clearing the flags in the POEGGx or POEGPICRx register

26.3.6.1 Cancellation of Output Disabling Request by Reset

Any type of reset can cancel an output disabling request. For details on resets, refer to section 6, Resets.

26.3.6.2 Cancellation of Output Disabling Request by Clearing Flags in the POEGGx or POEGPICRx Register

An output disabling request is canceled by setting all of the following flags and bit that are set to 1 to 0.

- POEGGx.PIDF flag
- POEGGx.IOCF flag
- POEGGx.OSTPF flag
- POEGGx.SSF bit
- POEGPICRx.PIDF flag

When the output disabling request is canceled, the GPTW captures this signal at the end of the next PWM cycle and releases the PWM outputs from the disabled state after three GTCLK cycles at the earliest. Figure 26.5 shows the timing for releasing the output disabled state.
To clear the flag, make sure that the detection signals that set the flag to 1 is 0 before writing 0 to the flag. Writing 0 to the flag when the detection signal is 1 does not clear the flag.
The status of the level detection signals can be monitored using the following status flags.

- GTETRGx pin input signal: POEGGx.ST flag (GTETRGx signal status flag)
- Output disabling condition detection signals: GPTWn.GTST.DTEF flag (dead time error flag)
  GPTWn.GTST.OABLF flag (simultaneous low output flag)
  GPTWn.GTST.OABHF flag (simultaneous high output flag)
- Oscillation stop detection signal: OSTDSR.OSTDF flag (oscillation stop detection flag)
- POEGx pin input signal: POEGPICRx.ST flag (POEGx signal status flag)

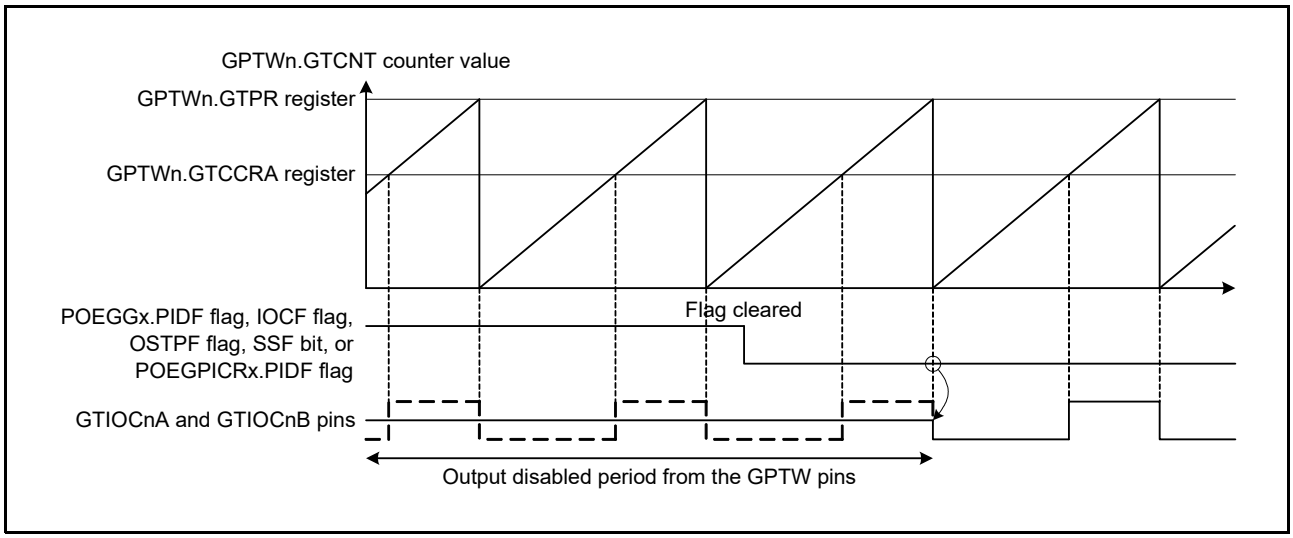


Figure 26.5 Timing for Releasing the Output Disabled State of the GPTW Pins by Canceling the Output Disabling Request

26.4 Interrupt Sources

The POEG has the following interrupt sources for each group.

- Detection of input level on the GTETRGx pin (x = A to D)
- Detection of output disabling condition detection signal from the GPTW
- Detection of input level or input edge on the POEGx pin (x = A to D)

Table 26.3 lists the interrupt sources.

Table 26.3 Interrupt sources

Interrupt Name	Symbol	Corresponding Flag	Interrupt Source
Group A Interrupt	POEGGAI	POEGGA.PIDF	Input signal from the GTETRGA pin
		POEGGA.IOCF	Output disabling condition detection signal from the GPTW
		POEGPICRA.PIDF	Input signal from the POEGA pin
Group B Interrupt	POEGGBI	POEGGB.PIDF	Input signal from the GTETRGB pin
		POEGGB.IOCF	Output disabling condition detection signal from the GPTW
		POEGPICRB.PIDF	Input signal from the POEGB pin
Group C Interrupt	POEGGCI	POEGGC.PIDF	Input signal from the GTETRGC pin
		POEGGC.IOCF	Output disabling condition detection signal from the GPTW
		POEGPICRC.PIDF	Input signal from the POEGC pin
Group D Interrupt	POEGGDI	POEGGD.PIDF	Input signal from the GTETRGD pin
		POEGGD.IOCF	Output disabling condition detection signal from the GPTW
		POEGPICRD.PIDF	Input signal from the POEGD pin

26.5 External Trigger Output to GPTW

The POEG outputs the input signals on the GTETRGx pin (x = A to D) to the GPTW as external trigger signals via polarity conversion and a digital noise filter. The external trigger signal can be monitored by the POEGGx.ST flag. The GPTW can perform the following operations with the external trigger signal.

- Count start
- Count stop
- Counter clear
- Count up
- Count down
- Input capture

For details, refer to section 24, General PWM Timer (GPTW).

26.6 Usage Notes

26.6.1 Transitions to Low-Power Consumption Mode

When using the POEG, do not enter software standby mode 1 or 2 or deep software standby mode 1, 2, or 3. In this mode, the POEG is stopped and output disabling request cannot be issued.

26.6.2 Setting the Function for Stopping the Module

Operation of the POEG can be disabled or enabled using the module stop control register D (MSTPCRD). After reset is released, the POEG function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

26.6.3 Duplication of Output Disabling Request

The output disabling request via flags is a logical OR of the POEGGx.PIDF, IOCF, OSTPF flags, the SSF bit, and the POEGPICRx.PIDF flag. Therefore, note that the output disabling request will not be canceled if any one of the flags/bit is set to 1.

27. 8-Bit Timer (TMRa)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a base clock for the RSCI.
In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

Table 27.1 lists the specifications of the TMR. Table 27.2 lists the TMR functions.
Figure 27.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 27.2 shows that of the 8-bit timer module (unit 1).

Table 27.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0 to TMR3)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0 to TMR3) (2) Event counting operation (TMR0 to TMR3) (3) Counting restart operation (TMR0 to TMR3)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Capable of generating base clock for RSCI	Generates base clock for RSCI.*1
Low power consumption function	Each unit can be placed in a module stop state
Protect Zone filter	Security attribute can be set for each unit.*2

Note 1. For details, refer to section 42, Serial Communications Interface (RSCI).
Note 2. When each unit is set to the secure attribute, registers in the unit are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each unit is set to the non-secure attribute, registers in the unit are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 27.2 TMR Functions

Item		Unit 0			Unit 1		
Counter mode		**8 Bits**		**16 Bits**	**8 Bits**		**16 Bits**
Channel		**TMR0**	**TMR1**	**TMR0 + TMR1**	**TMR2**	**TMR3**	**TMR2 + TMR3**
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB TMRI0	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
Timer output	Low output	✓	✓	✓	✓	✓	✓
	High output	✓	✓	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓	✓
DTC activation	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
RSCI base clock generation*1		✓		—	✓		—
ELC output event	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	✓	✓	✓	✓	✓	✓
ELC input event	Counting start	✓	✓	—	✓	✓	—
	Event counting	✓	✓	—	✓	✓	—
	Counting restart	✓	✓	—	✓	✓	—
Module stop setting*2		MSTPCRF.MSTPF7 bit (unit 0), MSTPCRF.MSTPF6 bit (unit 1)					

✓: Possible

—: Impossible

Note 1. For details, refer to section 42, Serial Communications Interface (RSCI).

Note 2. For details, refer to section 11, Low Power Consumption.

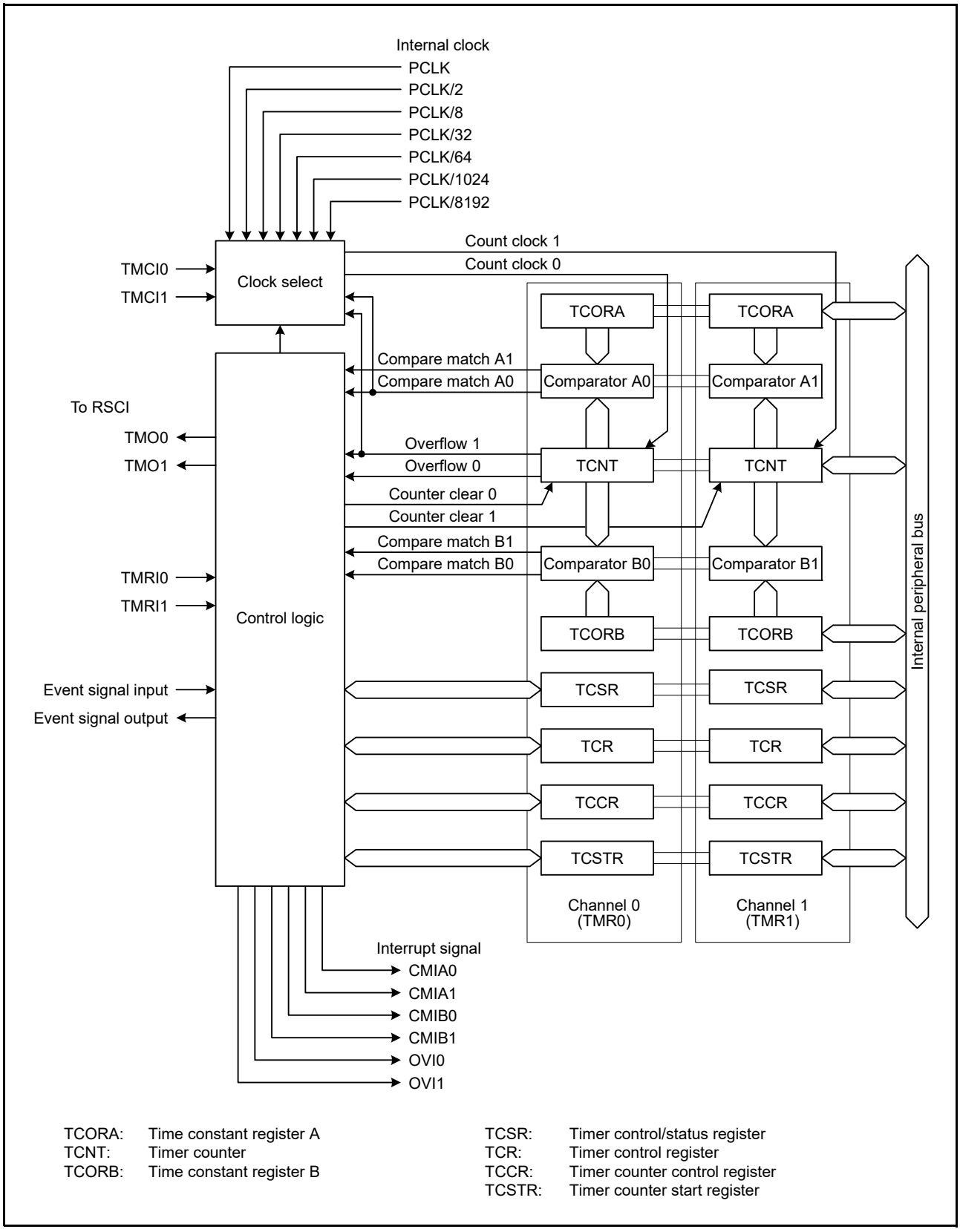


Figure 27.1 Block Diagram of TMR (Unit 0)

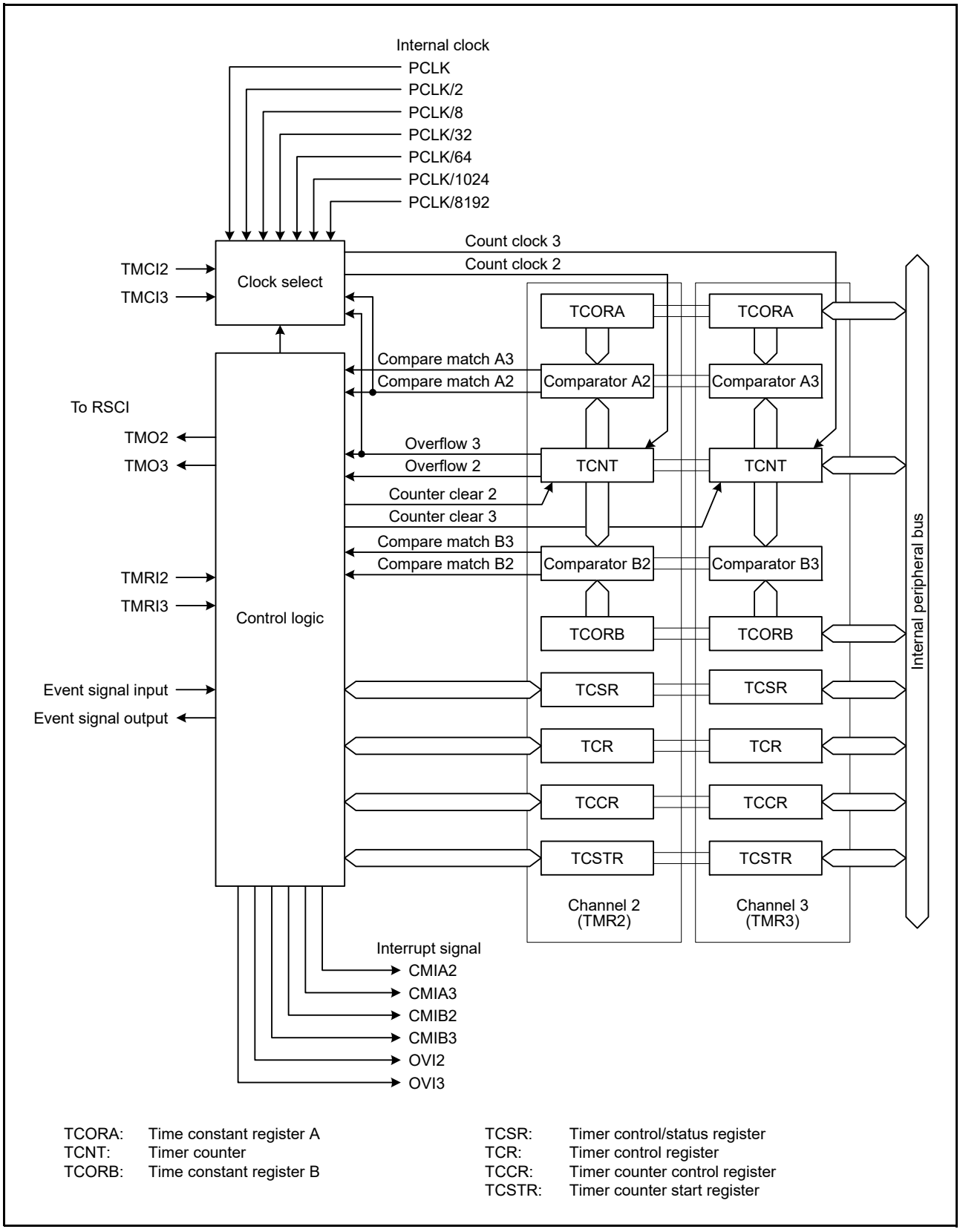


Figure 27.2 Block Diagram of TMR (Unit 1)

Table 27.3 lists the I/O pins of the TMR.

Table 27.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
		TMRI0	Input	Inputs external counter reset
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset

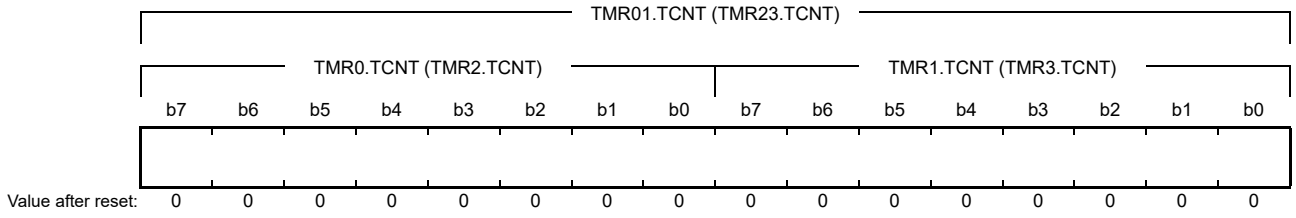
27.2 Register Descriptions

Table 27.4 Register Allocation for 16-Bit Access

Address	Register	Upper 8 Bits	Lower 8 Bits
8782 5008h	TMR01.TCNT	TMR0.TCNT	TMR1.TCNT
8782 5004h	TMR01.TCORA	TMR0.TCORA	TMR1.TCORA
8782 5006h	TMR01.TCORB	TMR0.TCORB	TMR1.TCORB
8782 500Ah	TMR01.TCCR	TMR0.TCCR	TMR1.TCCR
8782 5018h	TMR23.TCNT	TMR2.TCNT	TMR3.TCNT
8782 5014h	TMR23.TCORA	TMR2.TCORA	TMR3.TCORA
8782 5016h	TMR23.TCORB	TMR2.TCORB	TMR3.TCORB
8782 501Ah	TMR23.TCCR	TMR2.TCCR	TMR3.TCCR
0782 5008h	TMR01_NS.TCNT	TMR0_NS.TCNT	TMR1_NS.TCNT
0782 5004h	TMR01_NS.TCORA	TMR0_NS.TCORA	TMR1_NS.TCORA
0782 5006h	TMR01_NS.TCORB	TMR0_NS.TCORB	TMR1_NS.TCORB
0782 500Ah	TMR01_NS.TCCR	TMR0_NS.TCCR	TMR1_NS.TCCR
0782 5018h	TMR23_NS.TCNT	TMR2_NS.TCNT	TMR3_NS.TCNT
0782 5014h	TMR23_NS.TCORA	TMR2_NS.TCORA	TMR3_NS.TCORA
0782 5016h	TMR23_NS.TCORB	TMR2_NS.TCORB	TMR3_NS.TCORB
0782 501Ah	TMR23_NS.TCCR	TMR2_NS.TCCR	TMR3_NS.TCCR

27.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 8782 5008h, TMR1.TCNT 8782 5009h, TMR2.TCNT 8782 5018h, TMR3.TCNT 8782 5019h, TMR01.TCNT 8782 5008h, TMR23.TCNT 8782 5018h, TMR0_NS.TCNT 0782 5008h, TMR1_NS.TCNT 0782 5009h, TMR2_NS.TCNT 0782 5018h, TMR3_NS.TCNT 0782 5019h, TMR01_NS.TCNT 0782 5008h, TMR23_NS.TCNT 0782 5018h



TCNT is an 8-bit readable/writable up-counter.

TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) comprise a single 16-bit counter (TMR01.TCNT, TMR23.TCNT) so they can be accessed together in 16-bit units.

The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a count clock.

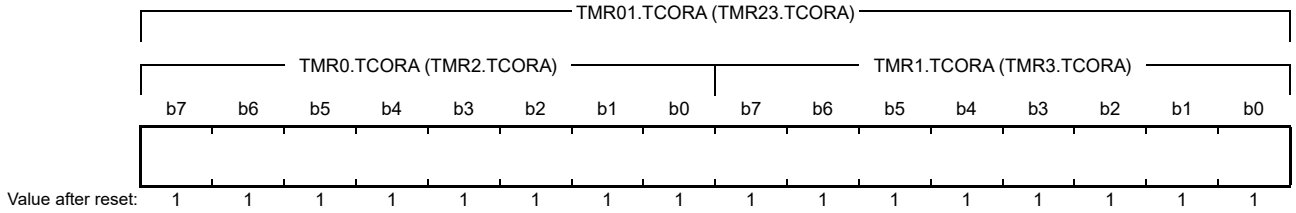
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt is output provided the interrupt request is enabled by the TCR.OVIE bit.

For details on the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUM), and Table 27.6, TMR Interrupt Sources.

27.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 8782 5004h, TMR1.TCORA 8782 5005h, TMR2.TCORA 8782 5014h, TMR3.TCORA 8782 5015h, TMR01.TCORA 8782 5004h, TMR23.TCORA 8782 5014h, TMR0_NS.TCORA 0782 5004h, TMR1_NS.TCORA 0782 5005h, TMR2_NS.TCORA 0782 5014h, TMR3_NS.TCORA 0782 5015h, TMR01_NS.TCORA 0782 5004h, TMR23_NS.TCORA 0782 5014h



TCORA is an 8-bit readable/writable register.

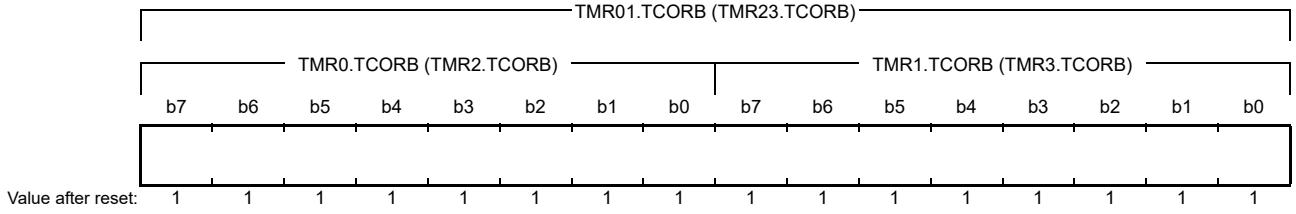
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register (TMR01.TCORA, TMR23.TCORA) so they can be accessed together in 16-bit units.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt is output provided the interrupt request is enabled by the TCR.CMIEA bit.

However, comparison is not performed during writing to TCORA. The timer output from the TMOn pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

27.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 8782 5006h, TMR1.TCORB 8782 5007h, TMR2.TCORB 8782 5016h, TMR3.TCORB 8782 5017h, TMR01.TCORB 8782 5006h, TMR23.TCORB 8782 5016h, TMR0_NS.TCORB 0782 5006h, TMR1_NS.TCORB 0782 5007h, TMR2_NS.TCORB 0782 5016h, TMR3_NS.TCORB 0782 5017h, TMR01_NS.TCORB 0782 5006h, TMR23_NS.TCORB 0782 5016h



TCORB is an 8-bit readable/writable register.

TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register (TMR01.TCORB, TMR23.TCORB) so they can be accessed together in 16-bit units.

The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt is output provided the interrupt request is enabled by the TCR.CMIEB bit.

However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

27.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 8782 5000h, TMR1.TCR 8782 5001h, TMR2.TCR 8782 5010h, TMR3.TCR 8782 5011h, TMR0_NS.TCR 0782 5000h, TMR1_NS.TCR 0782 5001h, TMR2_NS.TCR 0782 5010h, TMR3_NS.TCR 0782 5011h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external counter reset signal*1 (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match A Interrupt Enable	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match B Interrupt Enable	0: Compare match B interrupt requests (CMIBn) are disabled 1: Compare match B interrupt requests (CMIBn) are enabled	R/W

Note 1. To use an external counter reset signal, set the corresponding pin function. For details, refer to section 22, I/O Ports.

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match A Interrupt Enable)

Selects whether compare match A interrupt requests (CMIAn) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

CMIEB Bit (Compare Match B Interrupt Enable)

Selects whether compare match B interrupt requests (CMIBn) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

27.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 8782 500Ah, TMR1.TCCR 8782 500Bh, TMR2.TCCR 8782 501Ah, TMR3.TCCR 8782 501Bh, TMR01.TCCR 8782 500Ah, TMR23.TCCR 8782 501Ah, TMR0_NS.TCCR 0782 500Ah, TMR1_NS.TCCR 0782 500Bh, TMR2_NS.TCCR 0782 501Ah, TMR3_NS.TCCR 0782 501Bh, TMR01_NS.TCCR 0782 500Ah, TMR23_NS.TCCR 0782 501Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*1	See Table 27.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 27.5.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select	0: Cleared at rising edge of the external counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 22, I/O Ports.

TCCR register is a 8-bit register used to configure the basic operation of the counter. Two TCCR registers can be accessed simultaneously by accessing the address of the even channel TCCR register in 16-bit units.

CKS[2:0] Bits (Clock Select)

CSS[1:0] Bits (Clock Source Select)

The CKS[2:0] and CSS[1:0] bits select a clock. For details, see Table 27.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Table 27.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register					Description
	CSS[1:0]		CKS[2:0]			
	b4	b3	b2	b1	b0	
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 22, I/O Ports.
Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no TCNT count clock is generated. Do not use this setting.

27.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 8782 5002h, TMR2.TCSR 8782 5012h, TMR0_NS.TCSR 0782 5002h, TMR2_NS.TCSR 0782 5012h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare-match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

- TMR1.TCSR, TMR3.TCSR

Address(es): TMR1.TCSR 8782 5003h, TMR3.TCSR 8782 5013h, TMR1_NS.TCSR 0782 5003h, TMR3_NS.TCSR 0782 5013h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	1	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

27.2.7 Timer Counter Start Register (TCSTR)

Address(es): TMR0.TCSTR 8782 500Ch, TMR1.TCSTR 8782 500Dh, TMR2.TCSTR 8782 501Ch, TMR3.TCSTR 8782 501Dh, TMR0_NS.TCSTR 0782 500Ch, TMR1_NS.TCSTR 0782 500Dh, TMR2_NS.TCSTR 0782 501Ch, TMR3_NS.TCSTR 0782 501Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCS
Value after reset:	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPC or ELOPD register of the event controller (ELC).

For details, refer to section 27.7, Link Operation by ELC, or section 21, Event Link Controller (ELC).

27.3 Operation

27.3.1 Pulse Output

Figure 27.3 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (cleared by compare match A) so that TCNT is cleared at a compare match of TCORA.
2. Set the TCSR.OSA[1:0] bits to 10b (high is output) and TCSR.OSB[1:0] bits to 01b (low is output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

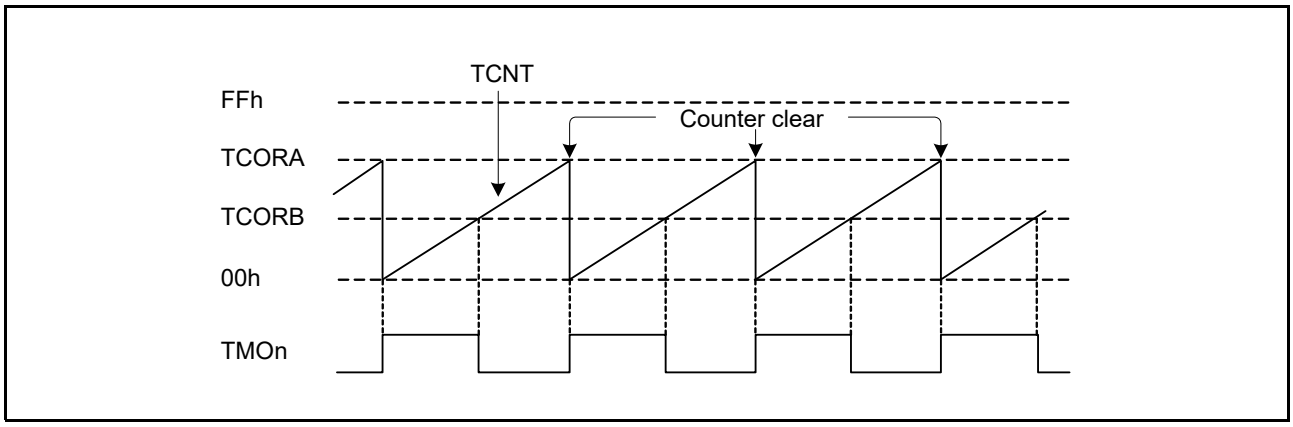


Figure 27.3 Example of Pulse Output (n = 0 to 3)

27.3.2 External Counter Reset Input

Figure 27.4 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

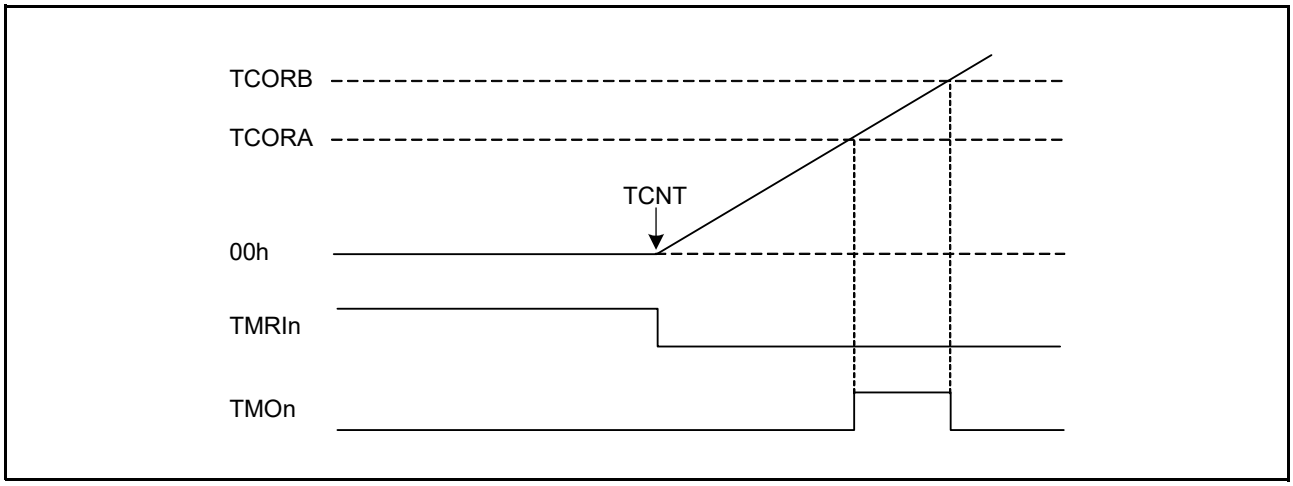


Figure 27.4 Example of External Counter Reset Signal Input (n = 0 to 3)

27.4 Operation Timing

27.4.1 TCNT Count Timing

Figure 27.5 shows the count timing of TCNT for internal clock. Figure 27.6 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

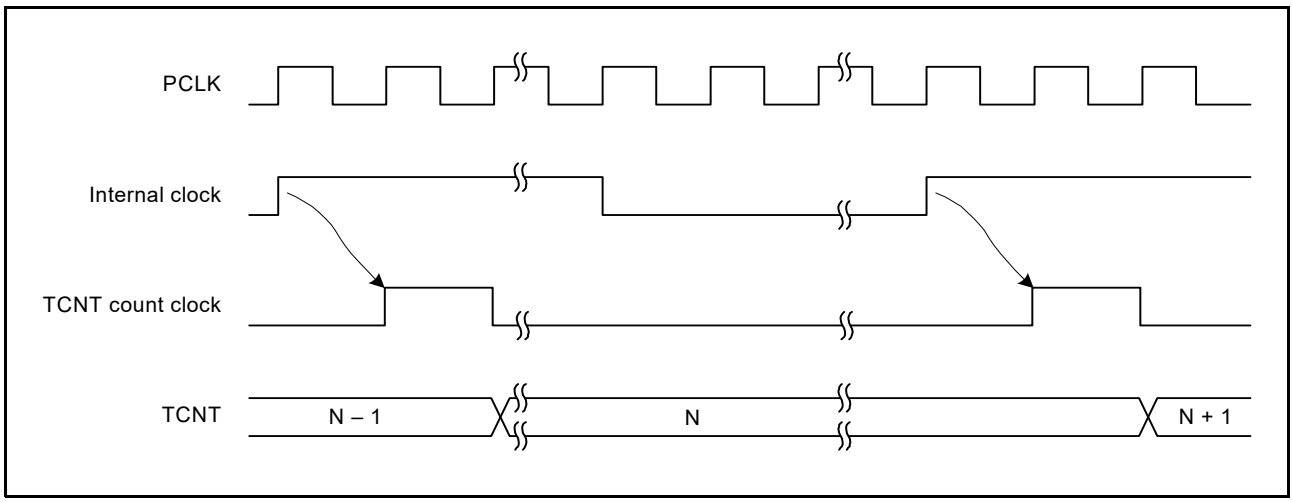


Figure 27.5 Count Timing for Internal Clock

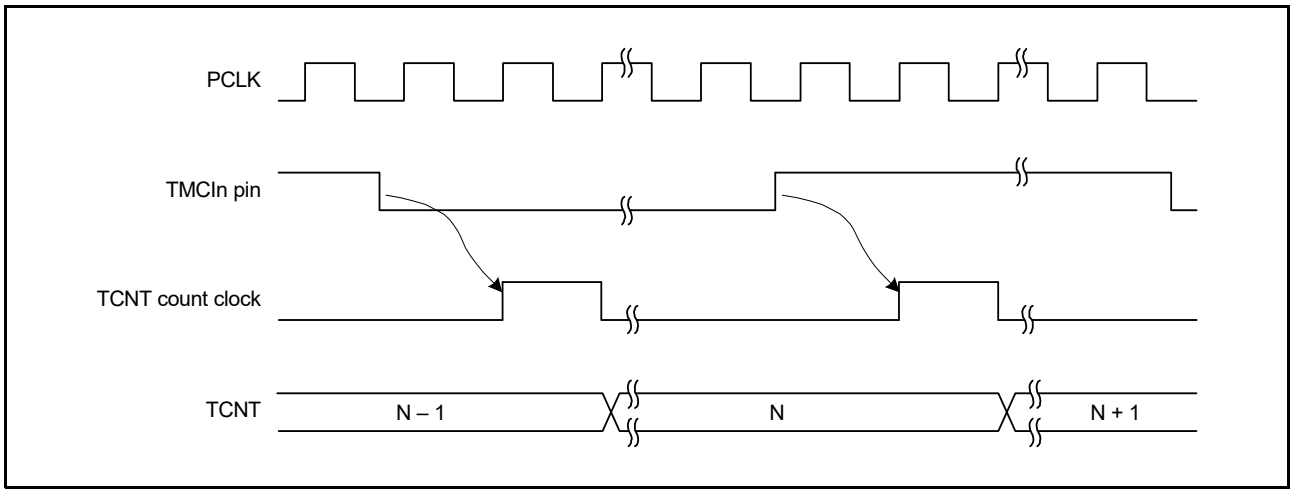


Figure 27.6 Count Timing for External Clock (at Both Edges)

27.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 27.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUM) and Table 27.6.

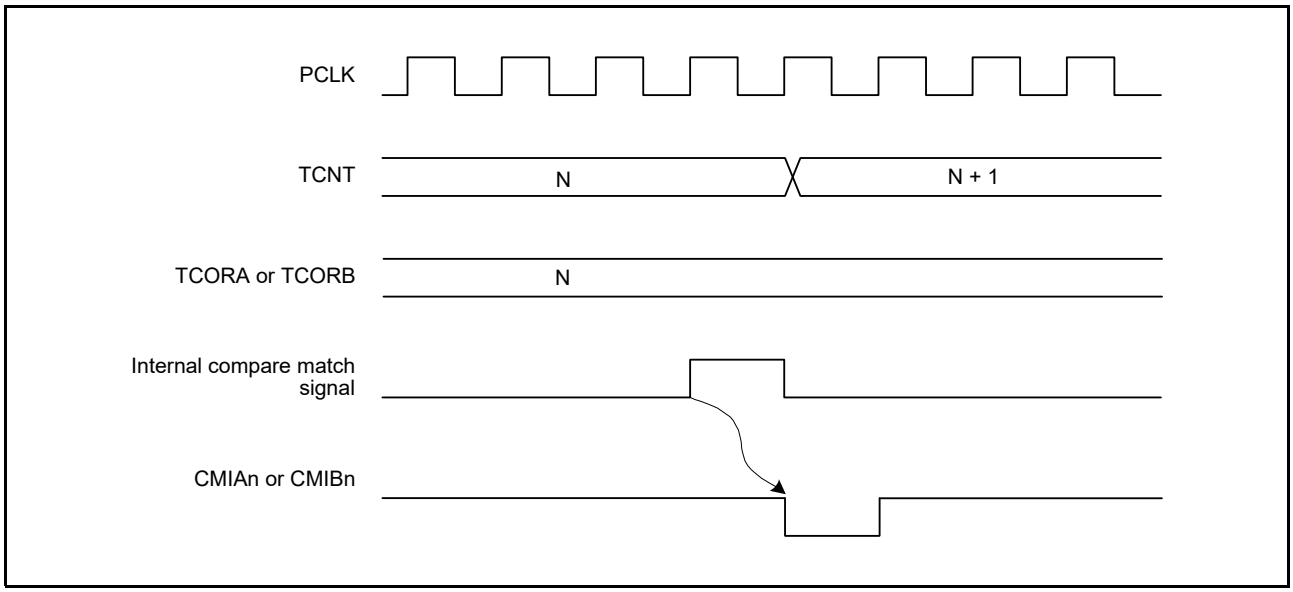


Figure 27.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

27.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).
Figure 27.8 shows the timing when the timer output is toggled by the compare match A signal.

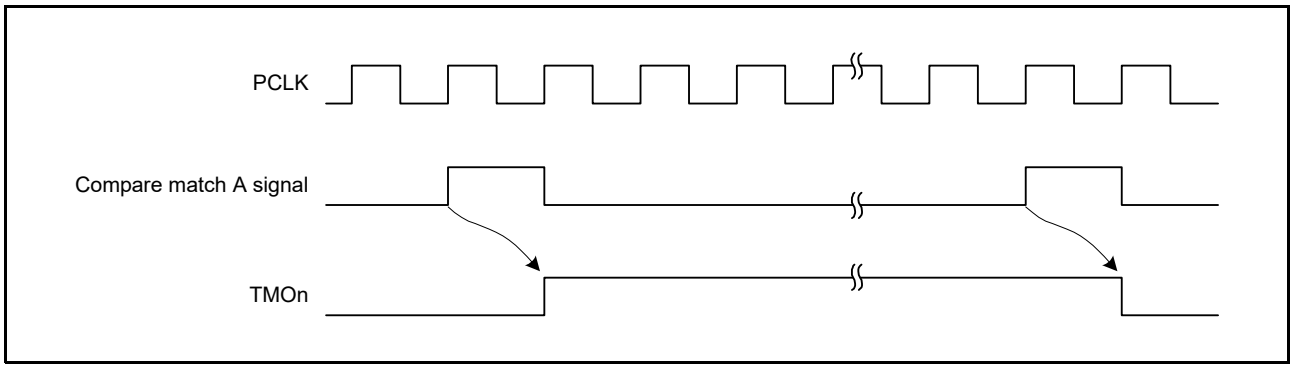


Figure 27.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

27.4.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of the TCR.CCLR[1:0] bits.
Figure 27.9 shows the timing of this operation.

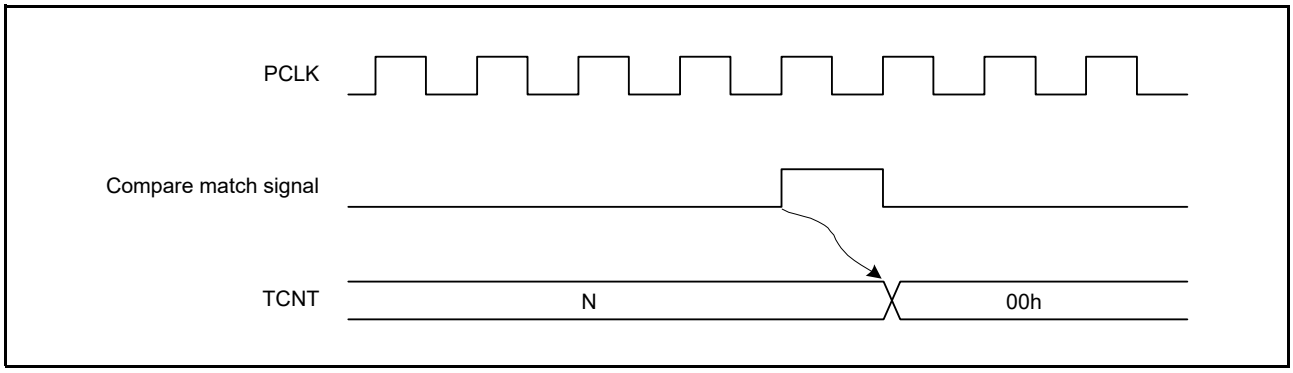


Figure 27.9 Timing of Counter Clear by Compare Match

27.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT.
Figure 27.10 and Figure 27.11 show the timing of this operation.

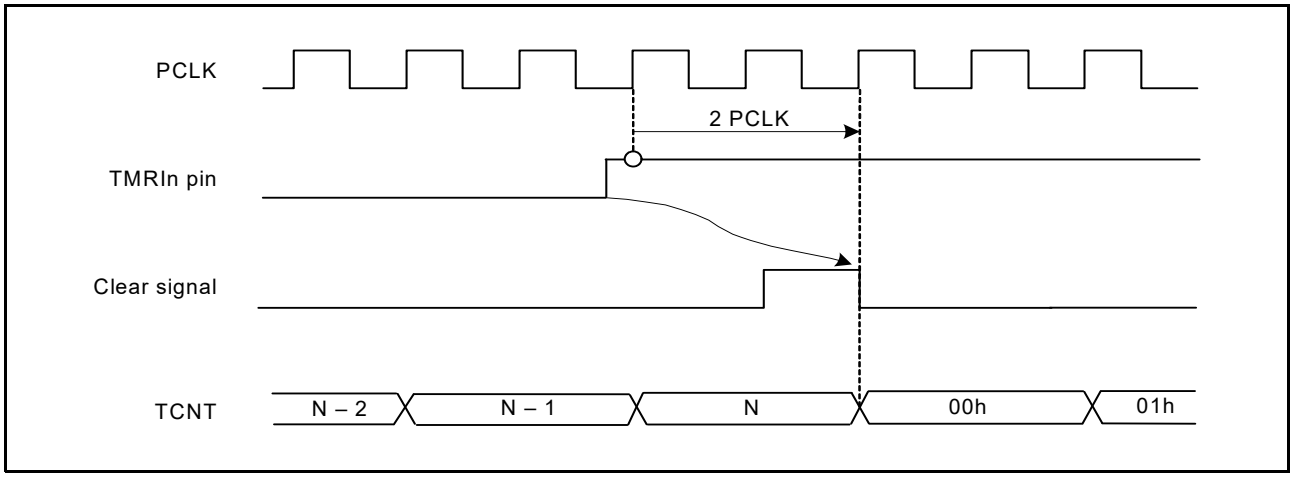


Figure 27.10 Clear Timing by External Counter Reset Signal (Rising Edge)

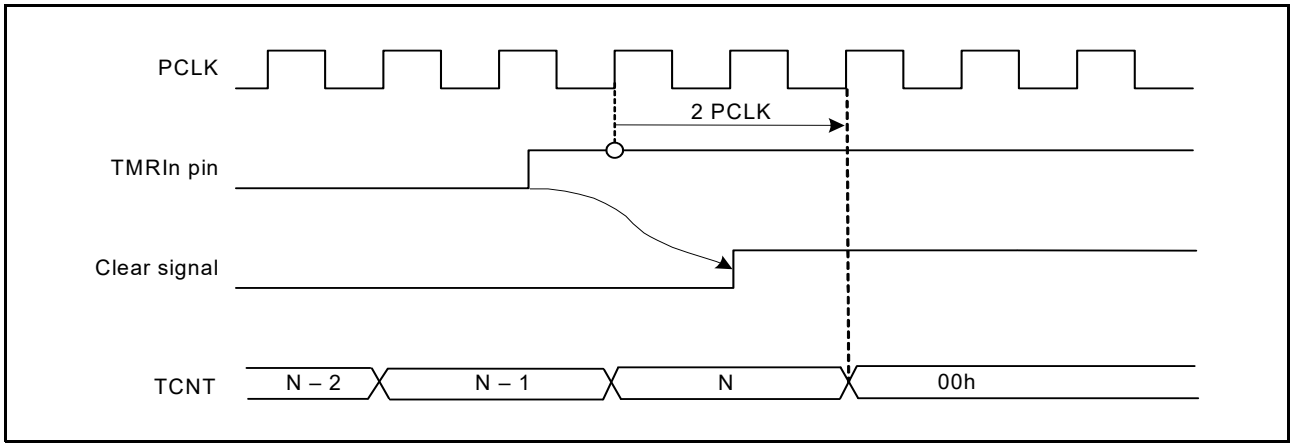


Figure 27.11 Clear Timing by External Counter Reset Signal (High Level)

27.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 27.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUM) and Table 27.6.

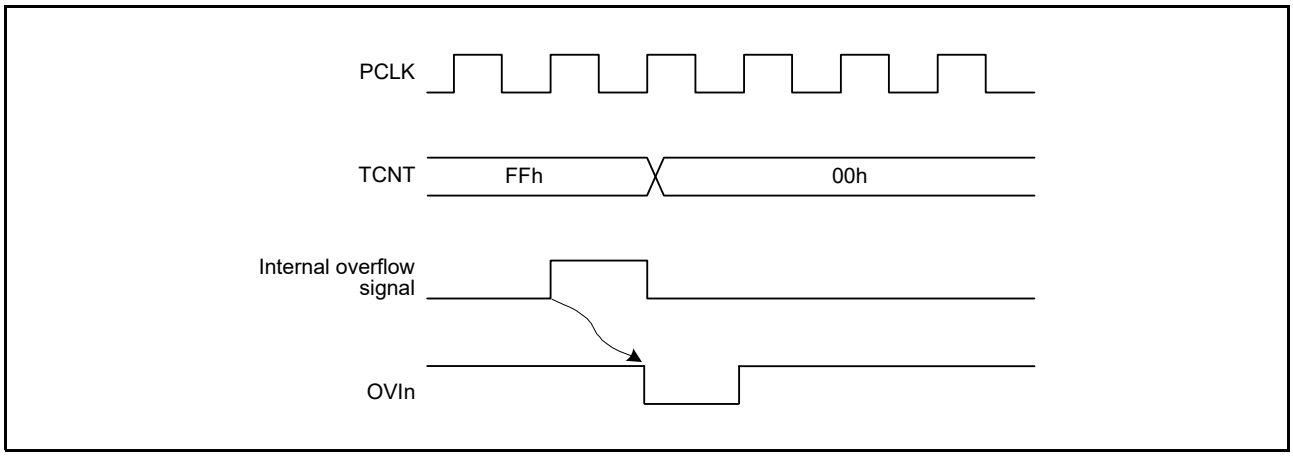


Figure 27.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

27.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either TMR0.TCCR or TMR1.TCCR are set to 11b, the TMR of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of TMR0 could be counted by TMR1 (compare match count mode).
This section describes unit 0. The operation of unit 1 with cascaded connection is the same as unit 0.

27.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the timer functions as a single 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

(1) Counter Clear Specification

- The settings of the TMR0.TCR.CCLR[1:0] bits become effective for the 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits have been set for counter clear at compare match, the 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared even if counter clear by the TMRI0 pin has been set.
- The settings of the TMR1.TCR.CCLR[1:0] bits are ignored.

(2) Pin Output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions.

27.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, TMR1.TCNT counts the number of occurrences of compare match A for TMR0. TMR0 and TMR1 are controlled independently. Conditions such as generation of interrupts, output from the TMOn pin (n = 0, 1), and counter clear are in accordance with the settings for each channel.

27.6 Interrupt Sources

27.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIAn, CMIBn, and OVIn. Their interrupt sources and priorities are listed in Table 27.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 27.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMIA0	TMR0.TCORA compare match	Possible
CMIB0	TMR0.TCORB compare match	Possible
OVI0	TMR0.TCNT overflow	Not possible
CMIA1	TMR1.TCORA compare match	Possible
CMIB1	TMR1.TCORB compare match	Possible
OVI1	TMR1.TCNT overflow	Not possible
CMIA2	TMR2.TCORA compare match	Possible
CMIB2	TMR2.TCORB compare match	Possible
OVI2	TMR2.TCNT overflow	Not possible
CMIA3	TMR3.TCORA compare match	Possible
CMIB3	TMR3.TCORB compare match	Possible
OVI3	TMR3.TCNT overflow	Not possible

27.7 Link Operation by ELC

27.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 to TMR3.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMRn.TCR.OVIE, TMRn.TCR.CMIEA, and TMRn.TCR.CMIEB (n = 0 to 3)). For details, refer to section 21, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

27.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

(1) Count Start

When the TMR count start operation is selected by the ELOPC or ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPC or ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.
If the specified event occurs while the TCS bit is 1, the event is ignored.
Write 0 to the TCSTR.TCS bit to stop counting.
When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.
The TCS bit is valid only when the ELOPC.TMR0MD[1:0], ELOPC.TMR1MD[1:0], ELOPD.TMR2MD[1:0], or ELOPD.TMR3MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPC or ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPC or ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

27.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

(1) Count Start

When the event specified by ELSRn occurs during the write cycle to the TCSTR.TCS bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

27.8 Usage Notes

27.8.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

27.8.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = PCLK / (N + 1)$$

27.8.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 27.13.

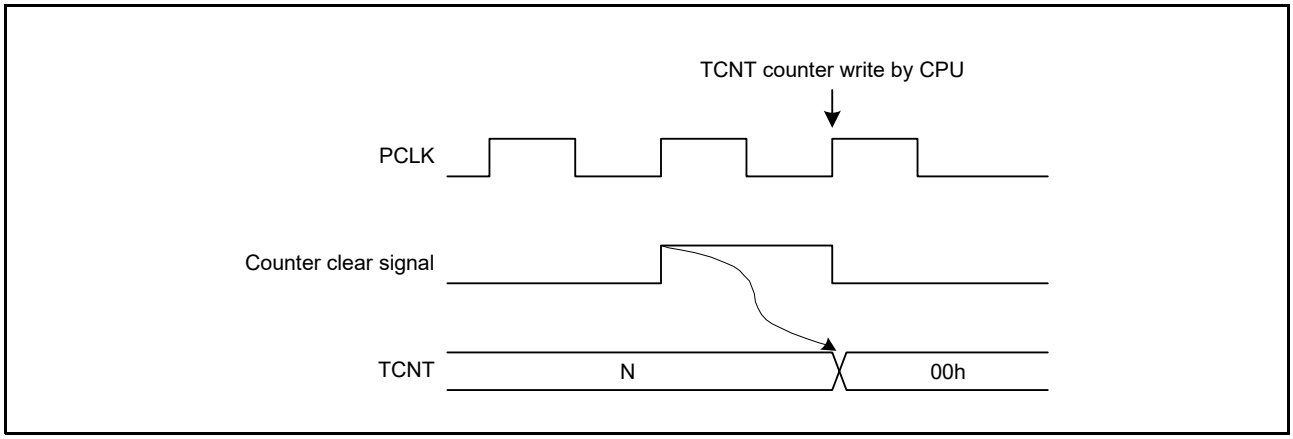


Figure 27.13 Conflict between TCNT Write and Counter Clear

27.8.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 27.14.

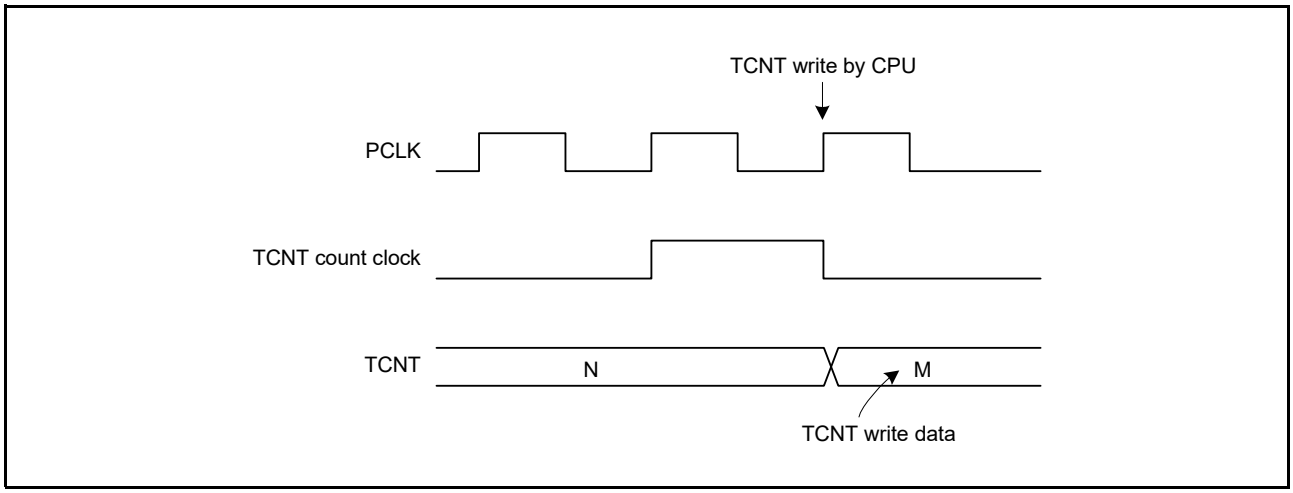


Figure 27.14 Conflict between TCNT Write and Increment

27.8.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 27.15, the write takes priority and the compare match signal does not reach High level.

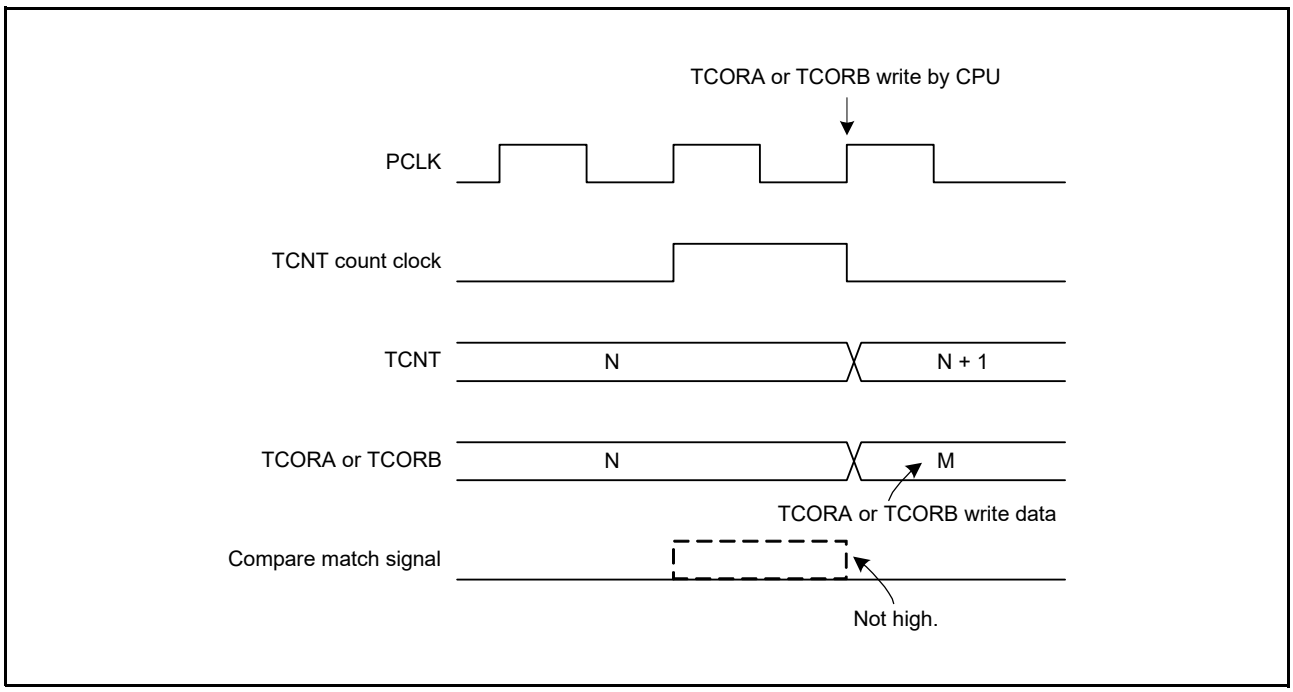


Figure 27.15 Conflict between TCORA or TCORB Write and Compare Match

27.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 27.7.

Table 27.7 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

27.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 27.8 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 27.8, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented. The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 27.8 Switching of Internal Clocks and TCNT Operation (1/2)

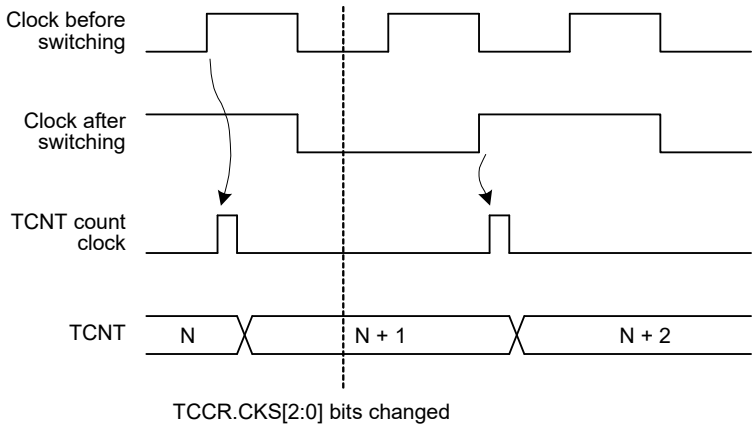
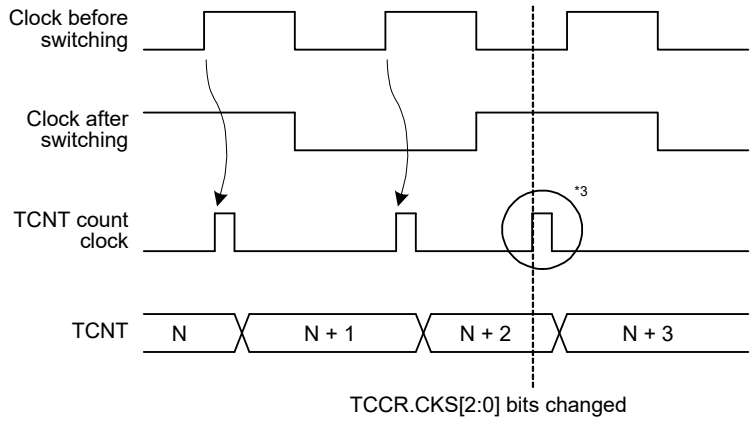
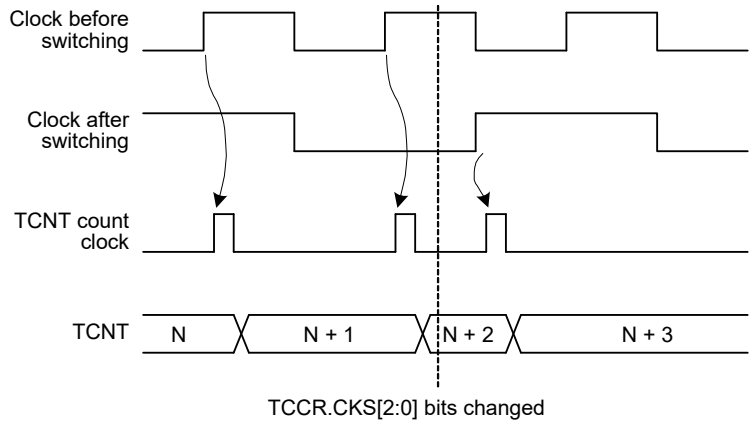
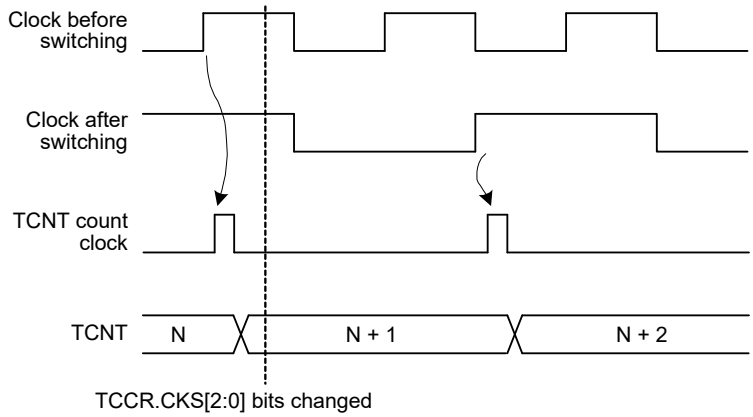
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 27.8 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

27.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

27.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.
At this time, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 27.16 shows operation timing when the compare match interrupt signal is continuously output.

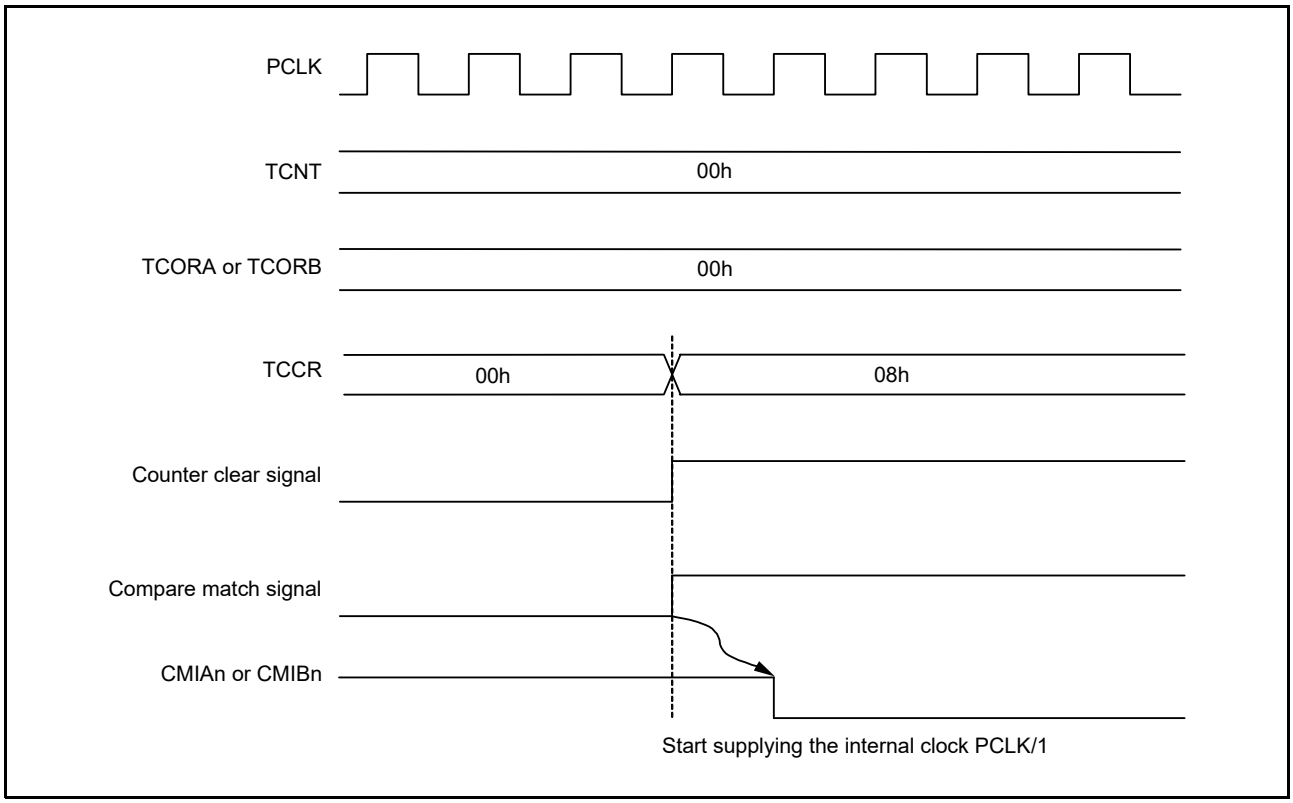


Figure 27.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

27.8.10 Transition to Software Standby Mode

Before transitioning to software standby mode, set the TCCR.CSS[1:0] bits to 00b and the TCCR.CKS[2:0] bits to 000b for all channels to stop the clock supply to the counters. After returning from software standby mode, restoring the TCCR.CSS[1:0] and the TCCR.CKS[2:0] bits to their original values will resume counting from the value held prior to the transition.

28. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

28.1 Overview

Table 28.1 lists the specifications for the CMT.
Figure 28.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 28.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. CMT1 count start, event counter, or count restart operation is possible.
Low power consumption function	Each unit can be placed in a module stop state.
Protect Zone filter	Security attributes can be set for each unit.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

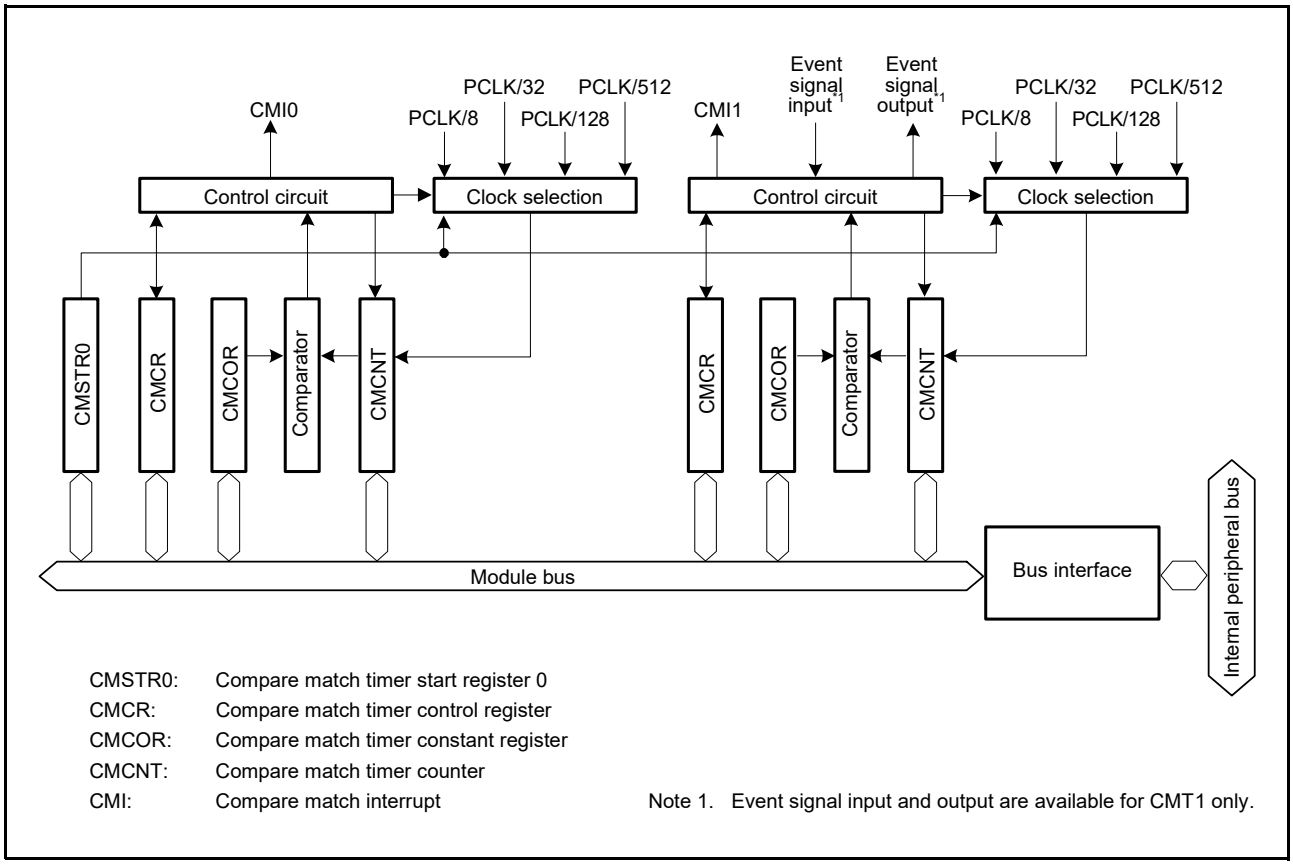


Figure 28.1 CMT (Unit 0) Block Diagram

28.2 Register Descriptions

28.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): CMT.CMSTR0 8782 6000h, CMT_NS.CMSTR0 0782 6000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR0	Count Start 0	0: CMT0.CMCNT count is stopped. 1: CMT0.CMCNT count is started.	R/W
b1	STR1	Count Start 1	0: CMT1.CMCNT count is stopped. 1: CMT1.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

28.2.2 Compare Match Timer Start Register 1 (CMSTR1)

Address(es): CMT.CMSTR1 8782 6010h, CMT_NS.CMSTR1 0782 6010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR3	STR2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR2	Count Start 2	0: CMT2.CMCNT count is stopped. 1: CMT2.CMCNT count is started.	R/W
b1	STR3	Count Start 3	0: CMT3.CMCNT count is stopped. 1: CMT3.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

28.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 8782 6002h, CMT1.CMCR 8782 6008h, CMT2.CMCR 8782 6012h, CMT3.CMCR 8782 6018h, CMT0_NS.CMCR 0782 6002h, CMT1_NS.CMCR 0782 6008h, CMT2_NS.CMCR 0782 6012h, CMT3_NS.CMCR 0782 6018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

28.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 8782 6004h, CMT1.CMCNT 8782 600Ah, CMT2.CMCNT 8782 6014h, CMT3.CMCNT 8782 601Ah, CMT0_NS.CMCNT 0782 6004h, CMT1_NS.CMCNT 0782 600Ah, CMT2_NS.CMCNT 0782 6014h, CMT3_NS.CMCNT 0782 601Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

28.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 8782 6006h, CMT1.CMCOR 8782 600Ch, CMT2.CMCOR 8782 6016h, CMT3.CMCOR 8782 601Ch, CMT0_NS.CMCOR 0782 6006h, CMT1_NS.CMCOR 0782 600Ch, CMT2_NS.CMCOR 0782 6016h, CMT3_NS.CMCOR 0782 601Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

28.3 Operation

28.3.1 Cyclic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 28.2 shows the operation of the CMCNT counter.

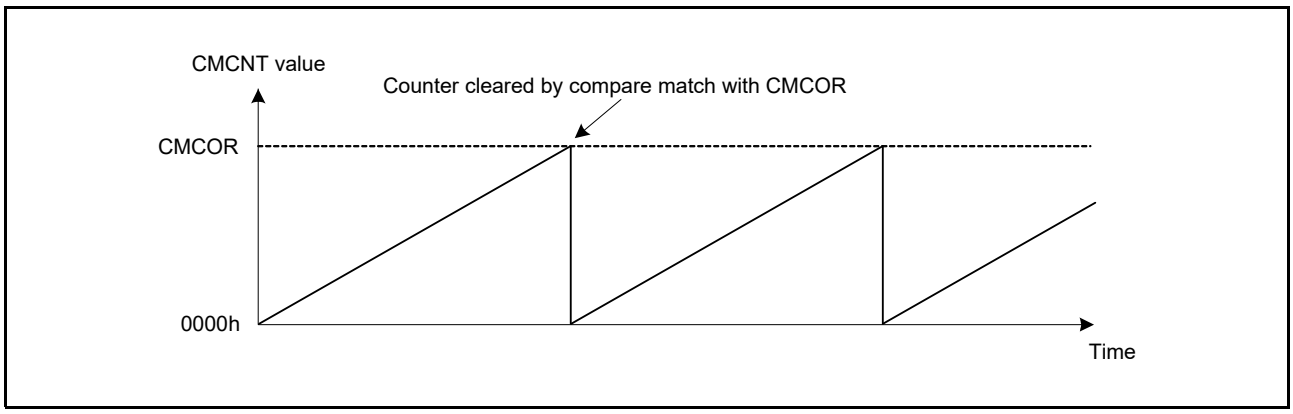


Figure 28.2 CMCNT Counter Operation

28.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 28.3 shows the timing of the CMCNT counter.

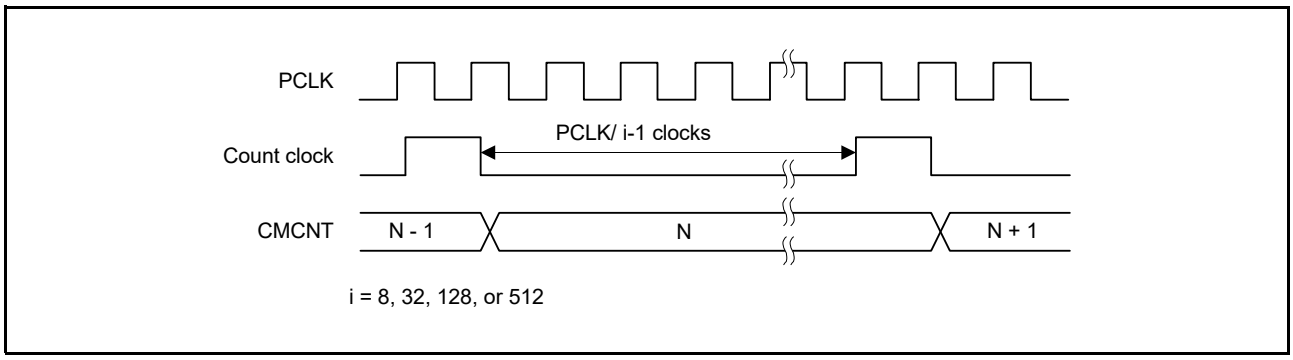


Figure 28.3 CMCNT Count Timing

28.4 Interrupts

28.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUM).

Table 28.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match in CMT0	Possible	Possible
CMI1	Compare match in CMT1	Possible	Possible
CMI2	Compare match in CMT2	Possible	Possible
CMI3	Compare match in CMT3	Possible	Possible

28.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 28.4 shows the timing of a compare match interrupt.

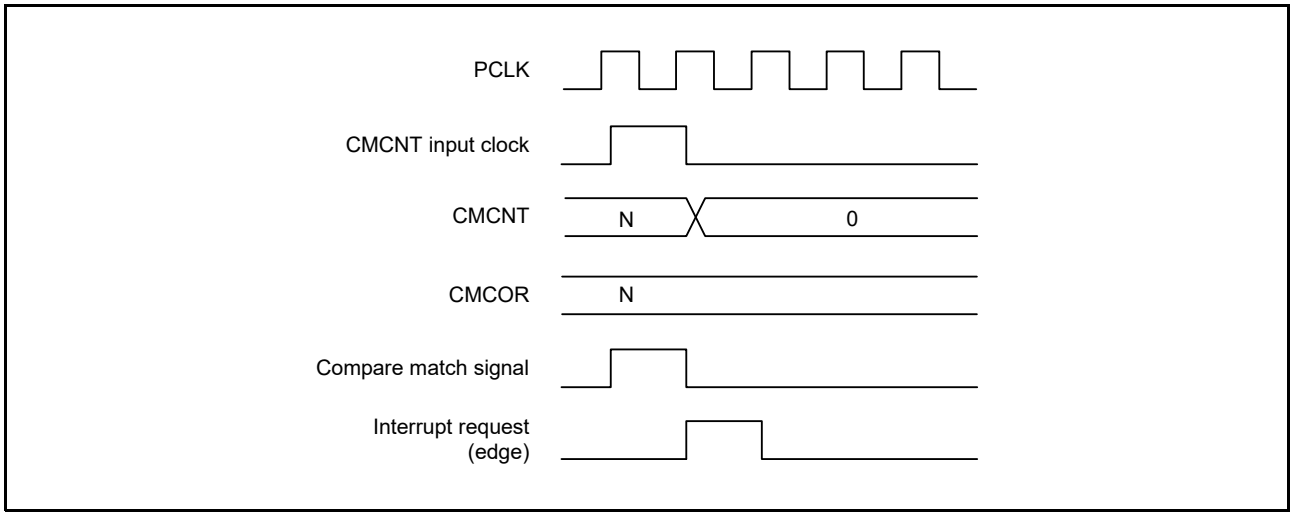


Figure 28.4 Timing of a Compare Match Interrupt

28.5 Link Operations by ELC

28.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

28.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR0 register of the ELC.

(1) Count Start

When the CMT count start operation is selected by the ELOPA register of the ELC and the event specified by ELSR0 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

(2) Event Count

When the CMT event count operation is selected by the ELOPA register of the ELC and the event specified by ELSR0 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the CMT count restart operation is selected by the ELOPA register of the ELC and the event specified by ELSR0 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

28.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

(1) Count Start

When the event specified by ELSR0 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSR0 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSR0 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

28.6 Usage Notes

28.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

28.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 28.5 shows the timing to clear the CMCNT counter.

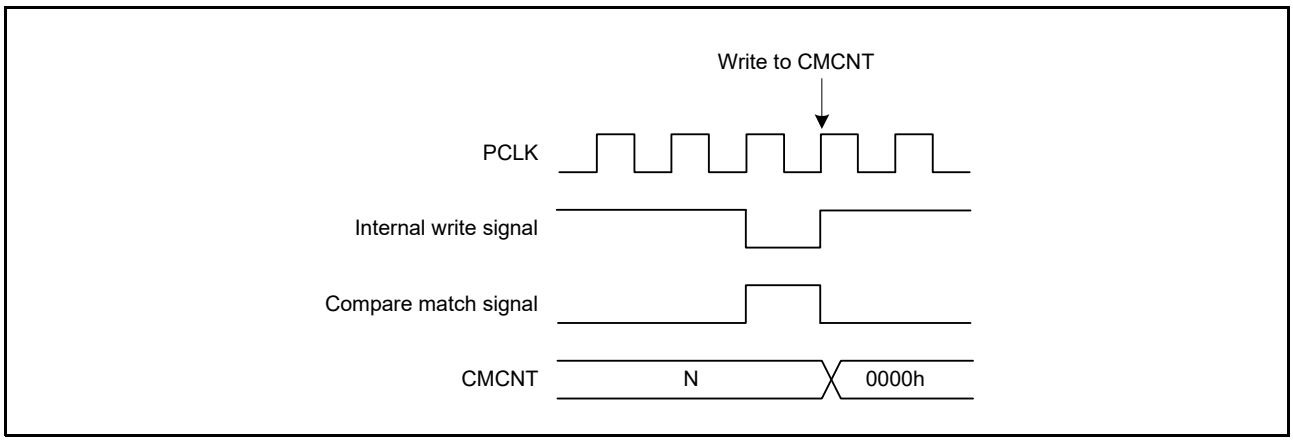


Figure 28.5 Conflict between CMCNT Counter Writing and Compare Match

28.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 28.6 shows the timing to write the CMCNT counter.

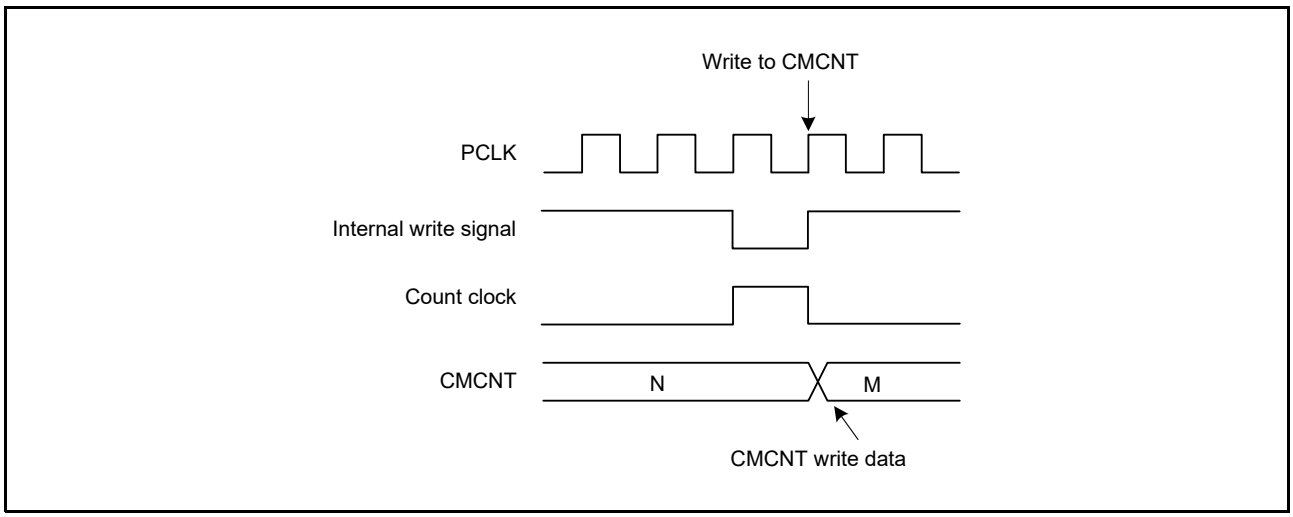


Figure 28.6 Conflict between CMCNT Counter Writing and Incrementing

28.6.4 Transition to Software Standby Mode

Before transitioning to software standby mode, set the CMSTR0 and CMSTR1 registers to 00h to stop all counters. After returning from software standby mode, setting the CMSTRm.STRn bit (m = 0, 1; n = 0 to 3) to 1 will resume counting from the value held prior to the transition.

29. Compare Match Timer W (CMTW)

This MCU includes two units (unit 0 and unit 1) with one channel of 32-bit compare match timer W (CMTW).
CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.
In this section, “PCLK” is used to refer to PCLKB.

29.1 Overview

Table 29.1 shows the specifications of the CMTW.
Figure 29.1 shows a block diagram of the CMTW0 and Figure 29.2 shows a block diagram of the CMTW1.

Table 29.1 CMTW Specifications

Item	Function
Number of channels	Two channels (unit 0, unit 1)
Timer counter	16-bit/32-bit selectable up-counter The counter returns to 0000 0000h after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLK/8, PCLK/32, PCLK/128, and PCLK/512
Input capture	Up to two input capture input signals available.
Output compare	Up to two output compare output signals available.
Compare match	One compare match available (no output compare output pin used).
Interrupts	Compare match interrupt Input capture 0 and 1 interrupts Output compare 0 and 1 interrupts
Event link function (output) (Unit 0)	Compare match
Event link function (input) (Unit 0)	One of the following operations is enabled after an event is accepted: • Count start operation • Event count operation • Count restart operation
Low power consumption function	Each unit can be placed in the module stop state.
Protect Zone filter	Security attributes can be set for each unit.*1

Note 1. When each unit is set to the secure attribute, registers in the unit are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each unit is set to the non-secure attribute, registers in the unit are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

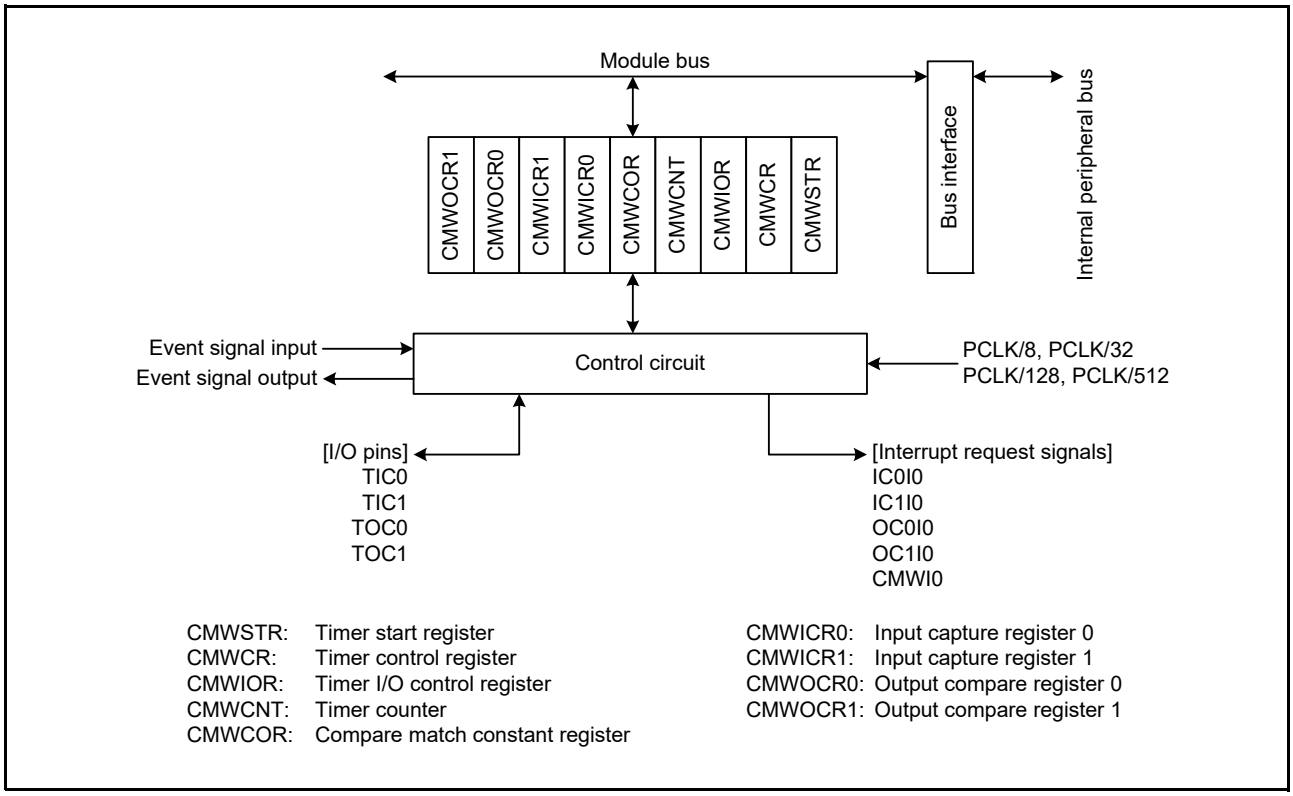


Figure 29.1 CMTW0 Block Diagram

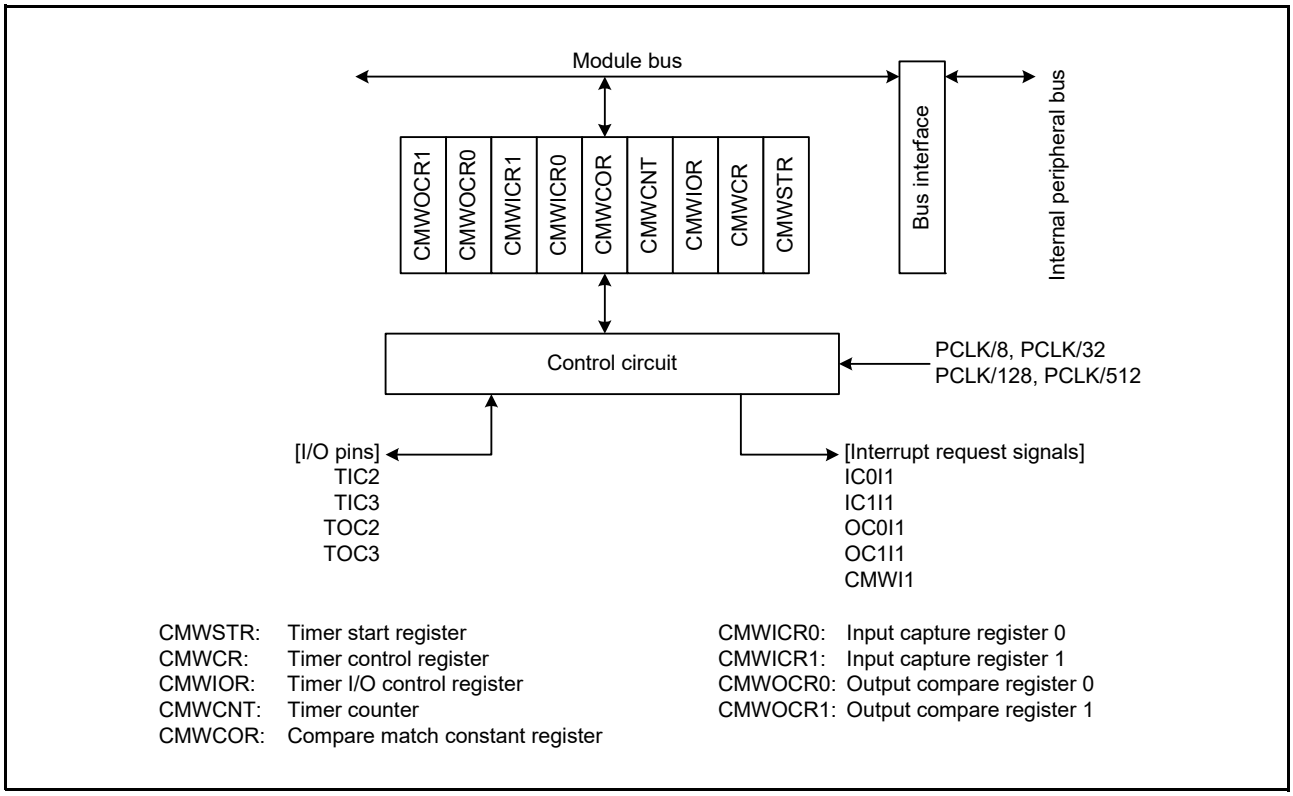


Figure 29.2 CMTW1 Block Diagram

Table 29.2 shows the CMTW pin configuration.

Table 29.2 CMTW Pin Configuration

Unit	Pin Name	I/O	Description
CMTW0	TIC0	Input	Input capture input for the CMTW0.CMWICR0 register
	TIC1	Input	Input capture input for the CMTW0.CMWICR1 register
	TOC0	Output	Output compare output for the CMTW0.CMWOCR0 register
	TOC1	Output	Output compare output for the CMTW0.CMWOCR1 register
CMTW1	TIC2	Input	Input capture input for the CMTW1.CMWICR0 register
	TIC3	Input	Input capture input for the CMTW1.CMWICR1 register
	TOC2	Output	Output compare output for the CMTW1.CMWOCR0 register
	TOC3	Output	Output compare output for the CMTW1.CMWOCR1 register

29.2 Register Descriptions

29.2.1 Timer Start Register (CMWSTR)

Address(es): CMTW0.CMWSTR 8782 6100h, CMTW1.CMWSTR 8782 6140h,
CMTW0_NS.CMWSTR 0782 6100h, CMTW1_NS.CMWSTR 0782 6140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR	Counter Start	0: CMWCNT counter count is stopped. (The value immediately before count operation stops is retained and the count operation is stopped.) 1: CMWCNT counter count is started.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

STR Bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of the STR bit.

29.2.2 Timer Control Register (CMWCR)

Address(es): CMTW0.CMWCR 8782 6104h, CMTW1.CMWCR 8782 6144h,
CMTW0_NS.CMWCR 0782 6104h, CMTW1_NS.CMWCR 0782 6144h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CMWIE	Compare Match Interrupt Request Enable	0: Interrupt request (CMWI) disabled 1: Interrupt request (CMWI) enabled	R/W
b4	IC0IE	Input Capture 0 Interrupt Request Enable	0: Interrupt request (IC0I) disabled 1: Interrupt request (IC0I) enabled	R/W
b5	IC1IE	Input Capture 1 Interrupt Request Enable	0: Interrupt request (IC1I) disabled 1: Interrupt request (IC1I) enabled	R/W
b6	OC0IE	Output Compare 0 Interrupt Request Enable	0: Interrupt request (OC0I) disabled 1: Interrupt request (OC0I) enabled	R/W
b7	OC1IE	Output Compare 1 Interrupt Request Enable	0: Interrupt request (OC1I) disabled 1: Interrupt request (OC1I) enabled	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	CMS	Timer Counter Size	0: 32 bits 1: 16 bits	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b13	CCLR[2:0]	Counter Clear	b15 b13 0 0 0: CMWCNT counter cleared by CMWCOR register compare match 0 0 1: Clearing of CMWCNT counter disabled 0 1 0: Clearing of CMWCNT counter disabled 0 1 1: Clearing of CMWCNT counter disabled 1 0 0: CMWCNT counter cleared by CMWICR0 register input capture 1 0 1: CMWCNT counter cleared by CMWICR1 register input capture 1 1 0: CMWCNT counter cleared by CMWOCR0 register compare match 1 1 1: CMWCNT counter cleared by CMWOCR1 register compare match	R/W

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] Bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral module clock (PCLK). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE Bit (Compare Match Interrupt Request Enable)

Enables or disables compare match interrupt request (CMWI) generation when the CMWCNT counter and the CMWCOR register values match.

IC0IE Bit (Input Capture 0 Interrupt Request Enable)

Enables or disables input capture 0 interrupt request (IC0I) generation when input capture is generated in the CMWICR0 register.

IC1IE Bit (Input Capture 1 Interrupt Request Enable)

Enables or disables input capture 1 interrupt request (IC1I) generation when input capture is generated in the CMWICR1 register.

OC0IE Bit (Output Compare 0 Interrupt Request Enable)

Enables or disables output compare 0 interrupt request (OC0I) generation when the CMWCNT counter and the CMWOCR0 register values match.

OC1IE Bit (Output Compare 1 Interrupt Request Enable)

Enables or disables output compare 1 interrupt request (OC1I) generation when the CMWCNT counter and CMWOCR1 register values match.

CMS Bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CCLR[2:0] Bits (Counter Clear)

Select the CMWCNT counter clearing source.

29.2.3 Timer I/O Control Register (CMWIOR)

Address(es): CMTW0.CMWIOR 8782 6108h, CMTW1.CMWIOR 8782 6148h,
CMTW0_NS.CMWIOR 0782 6108h, CMTW1_NS.CMWIOR 0782 6148h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IC0[1:0]	Input Capture 0 Control	b1 b0 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b3, b2	IC1[1:0]	Input Capture 1 Control	b3 b2 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b4	IC0E	Input Capture 0 Enable	0: Input capture 0 operation disabled 1: Input capture 0 operation enabled	R/W
b5	IC1E	Input Capture 1 Enable	0: Input capture 1 operation disabled 1: Input capture 1 operation enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	OC0[1:0]	Output Compare 0 Control	b9 b8 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b11, b10	OC1[1:0]	Output Compare 1 Control	b11 b10 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b12	OC0E	Output Compare 0 Enable	0: Output compare 0 operation disabled 1: Output compare 0 operation enabled	R/W
b13	OC1E	Output Compare 1 Enable	0: Output compare 1 operation disabled 1: Output compare 1 operation enabled	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	CMWE	Compare Match Enable	0: Compare match operation disabled 1: Compare match operation enabled	R/W

Note 1. After reset, low is output until the CMWIOR register is set.

29.2.4 Timer Counter (CMWCNT)

Address(es): CMTW0.CMWCNT 8782 6110h, CMTW1.CMWCNT 8782 6150h,
CMTW0_NS.CMWCNT 0782 6110h, CMTW1_NS.CMWCNT 0782 6150h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWCNT counter is a readable/writable up-counter.
Before starting count operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCNT counter, write data in 32-bit units with the upper bits set to 0000h. The CMWCNT counter can only be accessed in longword units.
When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting. When the CMWSTR.STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

29.2.5 Compare Match Constant Register (CMWCOR)

Address(es): CMTW0.CMWCOR 8782 6114h, CMTW1.CMWCOR 8782 6154h,
CMTW0_NS.CMWCOR 0782 6114h, CMTW1_NS.CMWCOR 0782 6154h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCOR register, write data in 32-bit units with the upper bits set to 0000h. The CMWCOR register can only be accessed in longword units. To detect an overflow, set the CMWCOR register value to FFFF FFFFh (32-bit count operation) or 0000 FFFFh (16-bit count operation). When the CMWCNT counter is set to 0, a compare match interrupt request (CMWI) can be used as an overflow detection signal.

29.2.6 Input Capture Register n (CMWICRn) (n = 0, 1)

Address(es): CMTW0.CMWICR0 8782 6118h, CMTW0.CMWICR1 8782 611Ch, CMTW1.CMWICR0 8782 6158h, CMTW1.CMWICR1 8782 615Ch, CMTW0_NS.CMWICR0 0782 6118h, CMTW0_NS.CMWICR1 0782 611Ch, CMTW1_NS.CMWICR0 0782 6158h, CMTW1_NS.CMWICR1 0782 615Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWICRn register is a read-only register in which the CMWCNT value is stored when an input capture is generated.

When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 in these registers are valid. Writing to these registers is invalid. The CMWICRn register can only be accessed in longword units.

29.2.7 Output Compare Register n (CMWOCRn) (n = 0, 1)

Address(es): CMTW0.CMWOCR0 8782 6120h, CMTW0.CMWOCR1 8782 6124h, CMTW1.CMWOCR0 8782 6160h, CMTW1.CMWOCR1 8782 6164h, CMTW0_NS.CMWOCR0 0782 6120h, CMTW0_NS.CMWOCR1 0782 6124h, CMTW1_NS.CMWOCR0 0782 6160h, CMTW1_NS.CMWOCR1 0782 6164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWOCRn register is a readable/writable register that set the value to be compared when an output compare is generated.

When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 of these registers become valid. When writing to these registers, write data in 32-bit units with the upper bits set to 0000h.

The CMWOCRn register can only be accessed in longword units. The initial value of CMWOCR0 and CMWOCR1 registers is FFFF FFFFh.

29.3 Operation

When the CMWCR register is set and then the STR bit in CMWSTR is set to 1, the CMTW starts count operation. Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting the CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

29.3.1 Cyclic Count Operation

When the counter clock is selected by using the CMWCR.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting cycles of the selected clock. When clearing of the counter is selected by the CMWCR.CCLR[2:0] bits and the counter clearing source is generated, the CMWCNT counter becomes 0000 0000h and continues counting. When clearing of the counter is not selected, an overflow is generated when FFFF FFFFh changes to 0000 0000h during 32-bit count operation and 0000 FFFFh changes to 0000 0000h during 16-bit count operation, and the CMWCNT counter continues counting.

29.3.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. At this time, a compare match interrupt request (CMWI) is generated. The CMWCNT counter restarts counting from 0000 0000h.

To enable overflow detection, the CMWCOR register value should be set to FFFF FFFFh (when the counter size is 32 bits) or 0000 FFFFh (when the counter size is 16 bits). When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. In this case, the compare match interrupt request (CMWI) is generated. The CMWCNT counter then restarts counting from 0000 0000h.

Figure 29.3 shows an example of procedure for setting compare match operation.

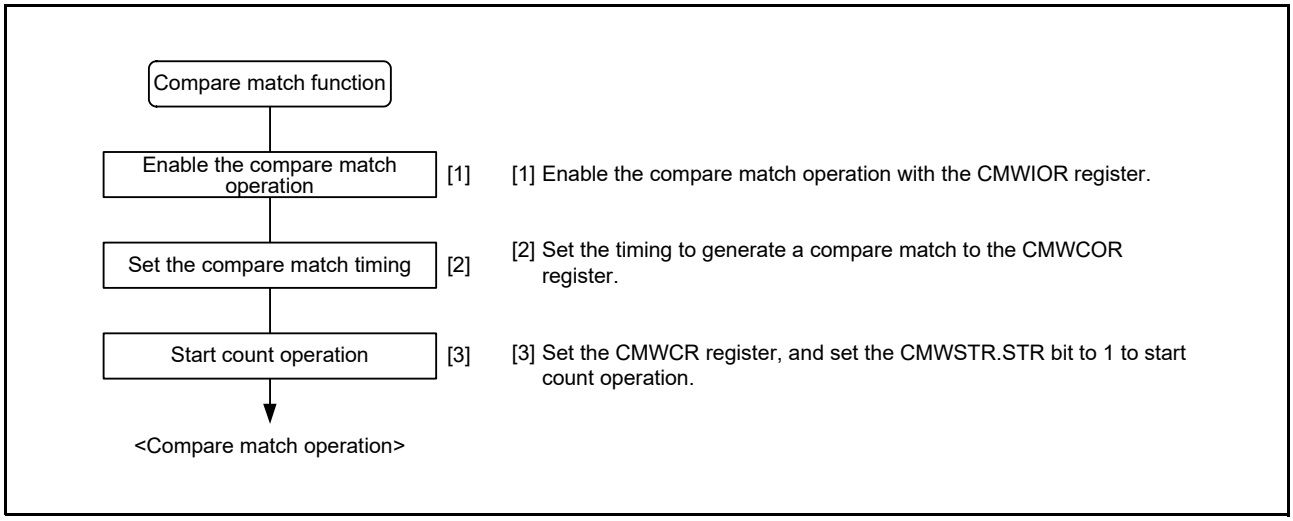


Figure 29.3 Procedure for Setting Compare Match Operation

Figure 29.4 shows an example when compare match with CMWCOR is set as a counter clearing source.

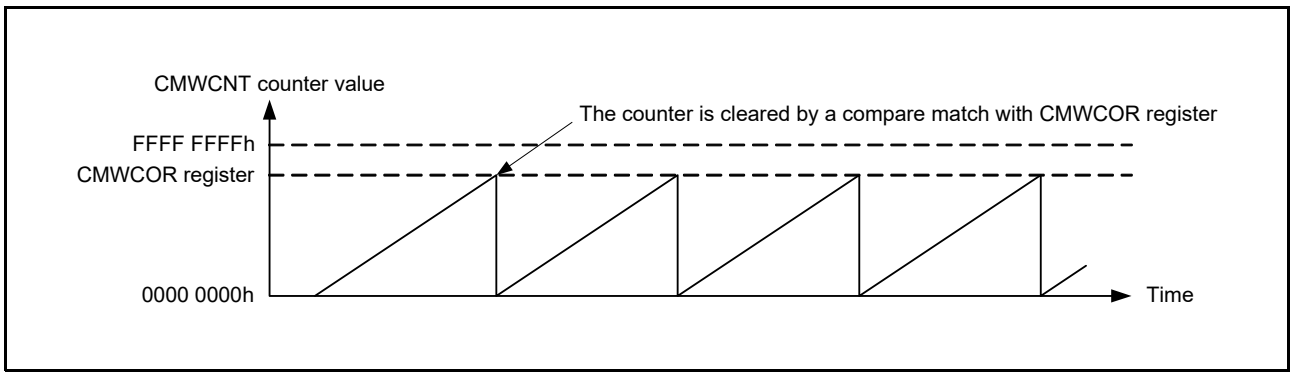


Figure 29.4 Example of Compare Match Operation

Figure 29.5 shows an example when CMWCOR is set to FFFF FFFFh and an overflow is detected.

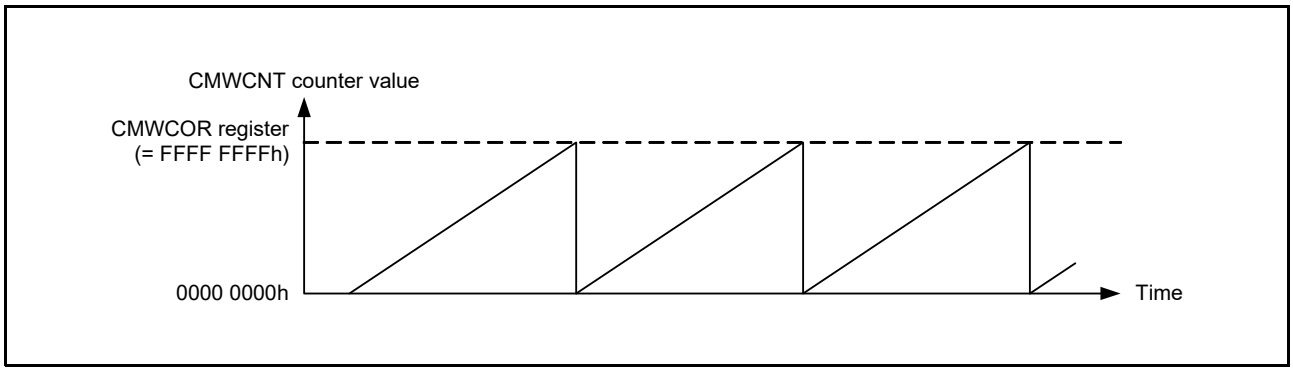


Figure 29.5 Example of Compare Match Operation (Overflow Detected)

29.3.3 Output Compare Function

The output compare function can be used for toggle waveform output. When the CMWCNT counter value matches either of the values of the CMWOCR0 or CMWOCR1 register, the output compare interrupt request (OC0I or OC1I) is generated. Figure 29.6 shows an example of procedure for setting output compare operation.

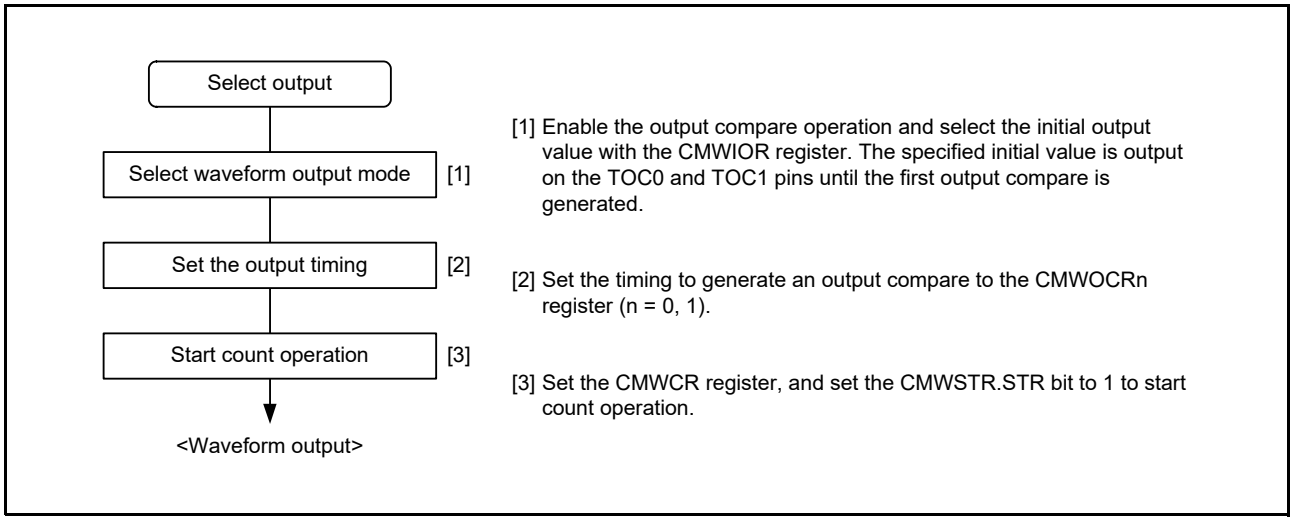


Figure 29.6 Procedure for Setting Output Compare Operation

Figure 29.7 shows an example of toggle waveform output from the TOC0 and TOC1 pins when the counter is set to be cleared by compare match with the CMWOCR1 register.

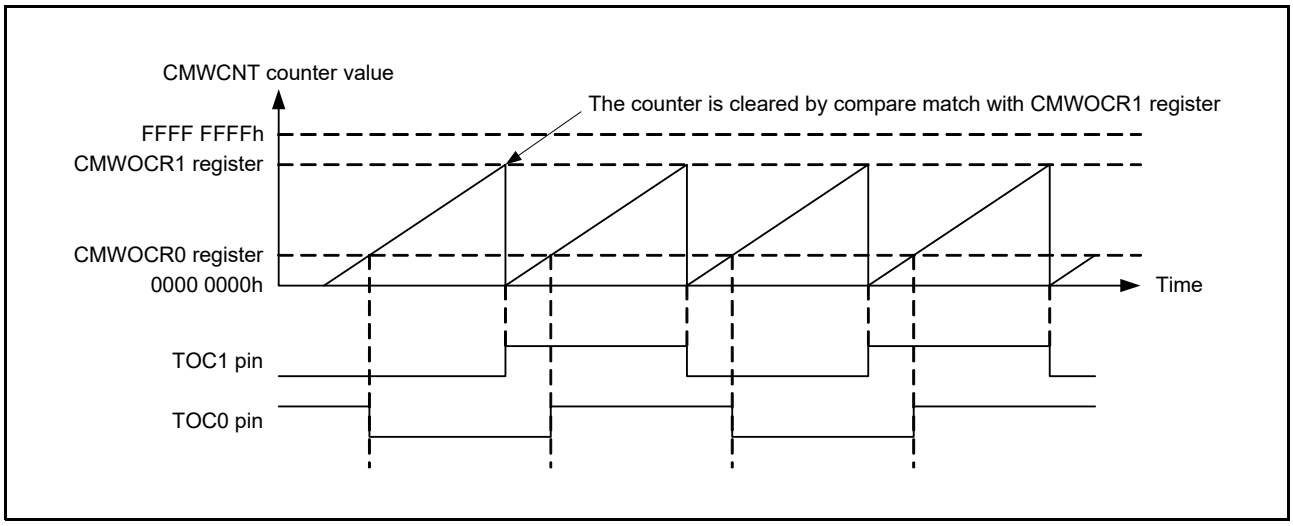


Figure 29.7 Example of Output Compare Operation (Unit 0)

29.3.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to the CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to the CMWICR0 or CMWICR1 register using the input capture operation, an input capture interrupt request (IC0I or IC1I) is generated. Figure 29.8 shows an example of procedure for setting input capture operation.

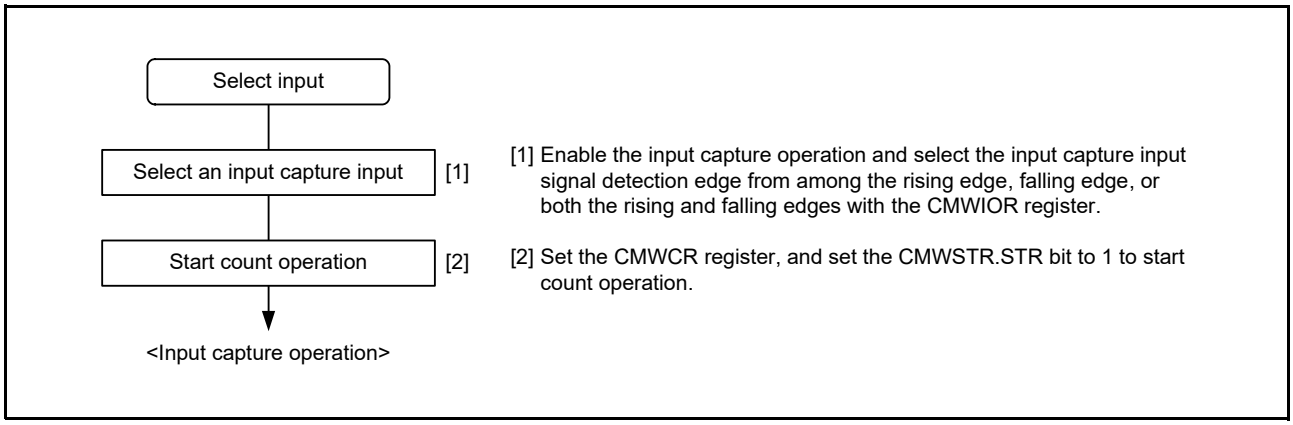


Figure 29.8 Procedure for Setting Input Capture Operation

Figure 29.9 shows an example in which both edges are selected for the TIC0 pin input capture detection edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

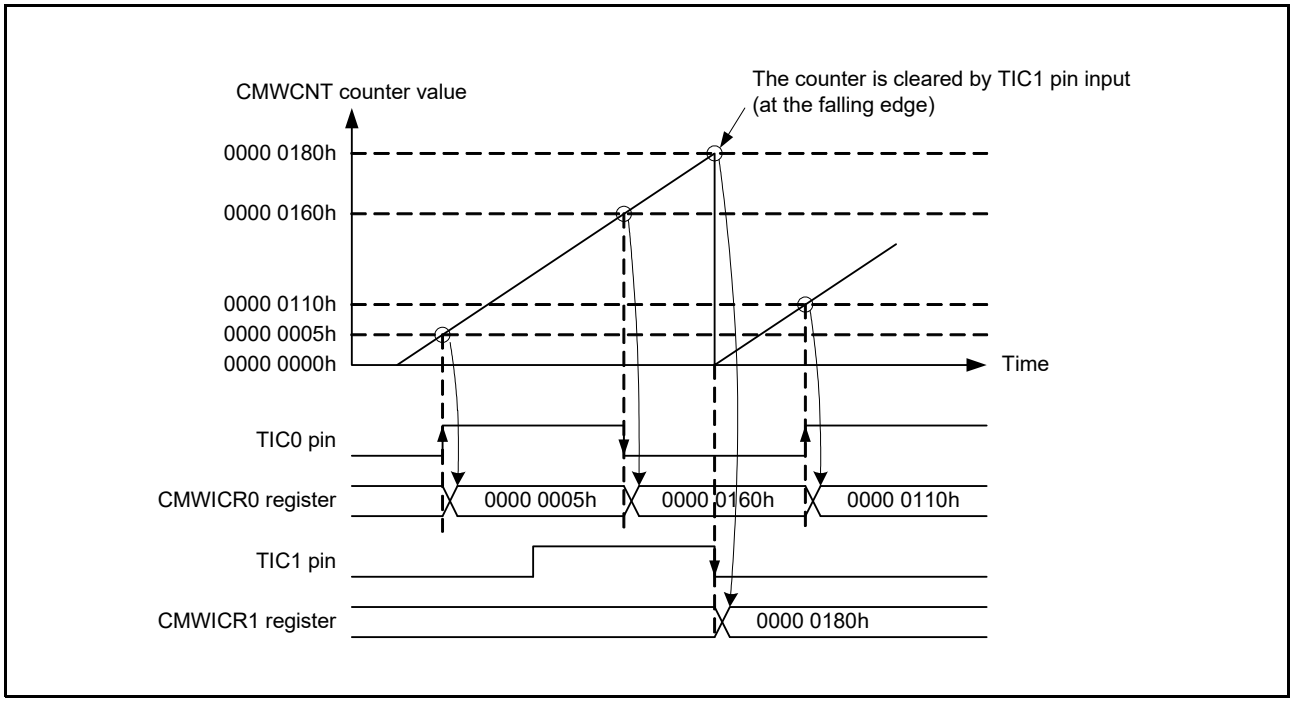


Figure 29.9 Example of Input Capture Operation (Unit 0)

29.3.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit.
When the counter is used as a 16-bit counter, set the value of the CMWCOR register in 32-bit units with the upper 16 bits set to 0000h. 0000 FFFFh should be set to detect an overflow. Similarly, set the values of the CMWOCR0 and CMWOCR1 registers in 32-bit units with the upper 16 bits set to 0000h. Read the CMWOCR0 and CMWOCR1 registers in 32-bit units. The upper 16 bits can be read as 0000h.

29.3.6 Count Timing of CMWCNT Counter

By setting the CMWCR.CKS[1:0] bits, one of four clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected as the counter clock to be input to the CMWCNT counter.
Figure 29.10 shows the timing.

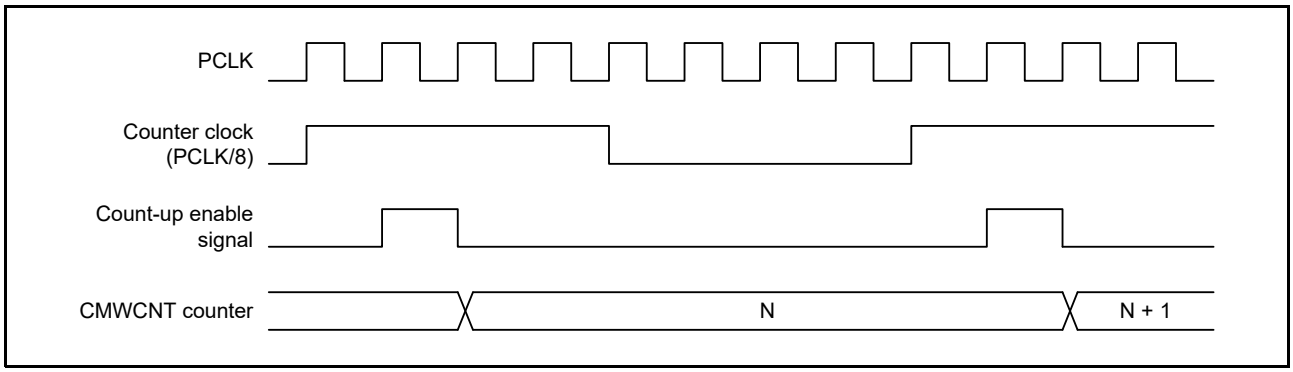


Figure 29.10 Count Timing (PCLK/8)

29.3.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCRn register (n = 0, 1) and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). The compare match signal is generated if the CMWCNT count-up enable signal is input after a match between the CMWOCRn register and CMWCNT counter values. When a compare match signal is generated, output of the output compare pin (TOC pin) is toggled.
Figure 29.11 shows output compare output timing.

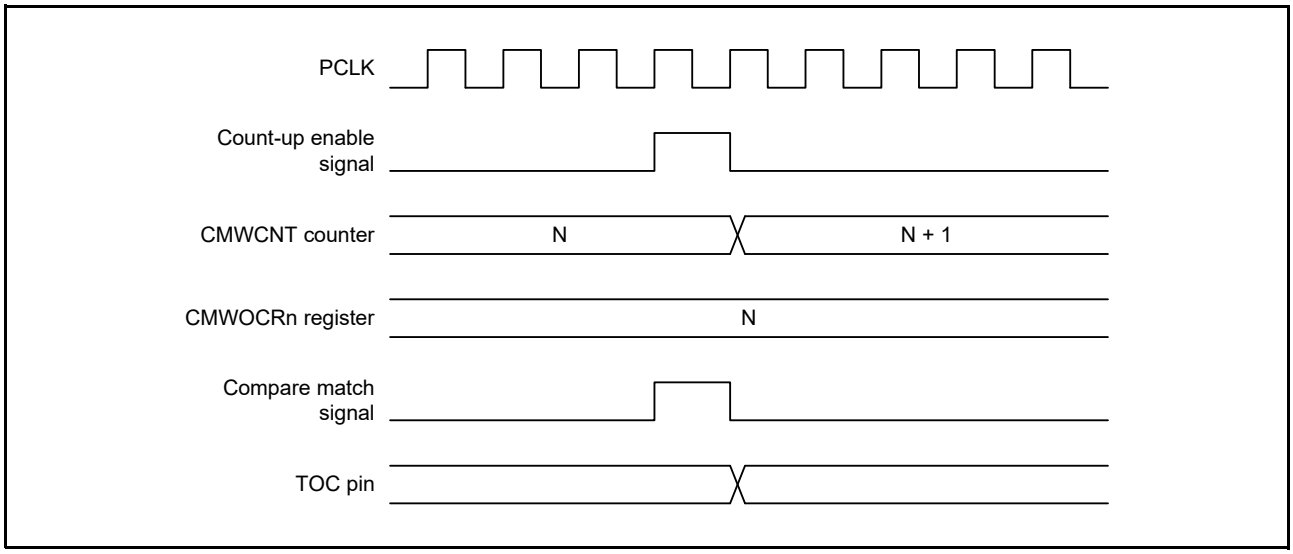


Figure 29.11 Output Compare Output Timing

29.3.8 Input Capture Timing

Figure 29.12 shows the timing of input capture operation at both edges.
When the edge of the TIC0 and TIC1 pins is detected, the CMWCNT counter value is transferred to the CMWICR0 and CMWICR1 registers, respectively.

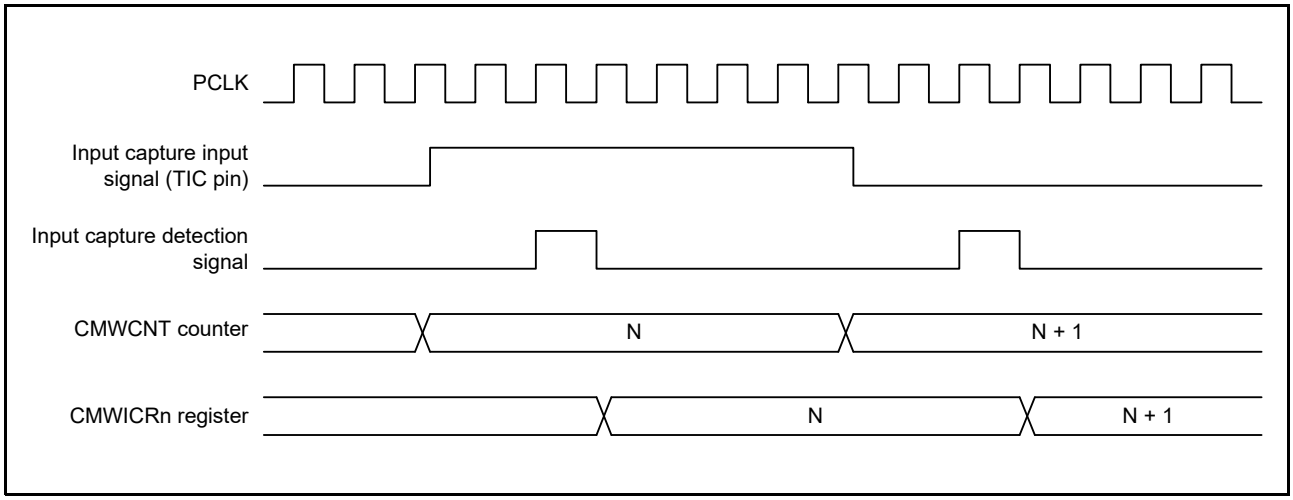


Figure 29.12 Input Capture Timing (Unit 0, Both-Edge Detection) (n = 0, 1)

29.4 Interrupts

29.4.1 CMTW Interrupt Sources and DTC/DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (IC0I and IC1I), two output compare interrupt requests (OC0I and OC1I), and a compare match interrupt request (CMWI).
Table 29.3 shows the interrupt sources. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately generated to the interrupt controller.
Each interrupt request can activate the DMAC or DTC. When the DMAC is used for data transfer, an interrupt request is not generated to the CPU. For generating an interrupt request to the CPU during data transfer using the DTC, refer to section 20, Data Transfer Controller (DTCA).

Table 29.3 CMTW Interrupt Sources

Unit	Name	Interrupt Request	Interrupt Request Enable Bit	DMAC/DTC Activation
CMTW0	CMWI0	Compare match of CMTW0.CMWCOR register	CMTW0.CMWCR.CMWIE	Possible
	IC0I0	Input capture of CMTW0.CMWICR0 register	CMTW0.CMWCR.IC0IE	Possible
	IC1I0	Input capture of CMTW0.CMWICR1 register	CMTW0.CMWCR.IC1IE	Possible
	OC0I0	Output compare of CMTW0.CMWOCR0 register	CMTW0.CMWCR.OC0IE	Possible
	OC1I0	Output compare of CMTW0.CMWOCR1 register	CMTW0.CMWCR.OC1IE	Possible
CMTW1	CMWI1	Compare match of CMTW1.CMWCOR register	CMTW1.CMWCR.CMWIE	Possible
	IC0I1	Input capture of CMTW1.CMWICR0 register	CMTW1.CMWCR.IC0IE	Possible
	IC1I1	Input capture of CMTW1.CMWICR1 register	CMTW1.CMWCR.IC1IE	Possible
	OC0I1	Output compare of CMTW1.CMWOCR0 register	CMTW1.CMWCR.OC0IE	Possible
	OC1I1	Output compare of CMTW1.CMWOCR1 register	CMTW1.CMWCR.OC1IE	Possible

29.4.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt request (CMWI) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value).The compare match signal, therefore, is not generated until a count-up enable signal is generated after the values of the CMWCNT counter and CMWCOR register have matched. Figure 29.13 shows the timing of compare match interrupt generation.

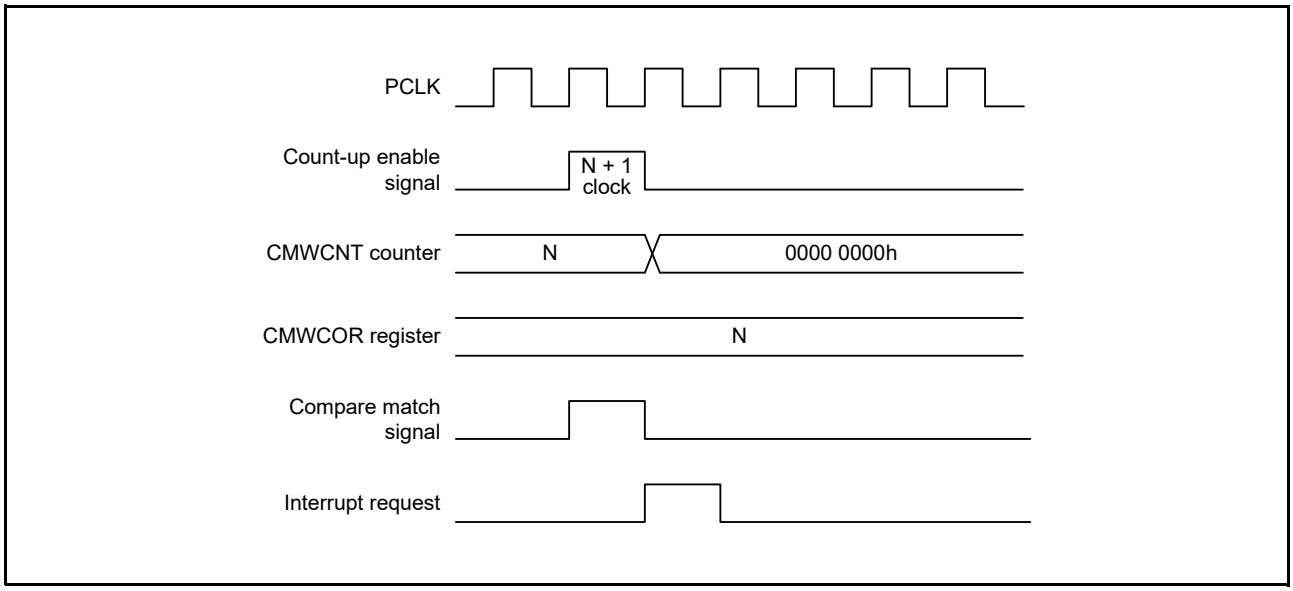


Figure 29.13 Timing of Compare Match Interrupt Generation

(a) Timing of Output Compare Interrupt Generation

Figure 29.14 shows the timing of output compare interrupt generation.

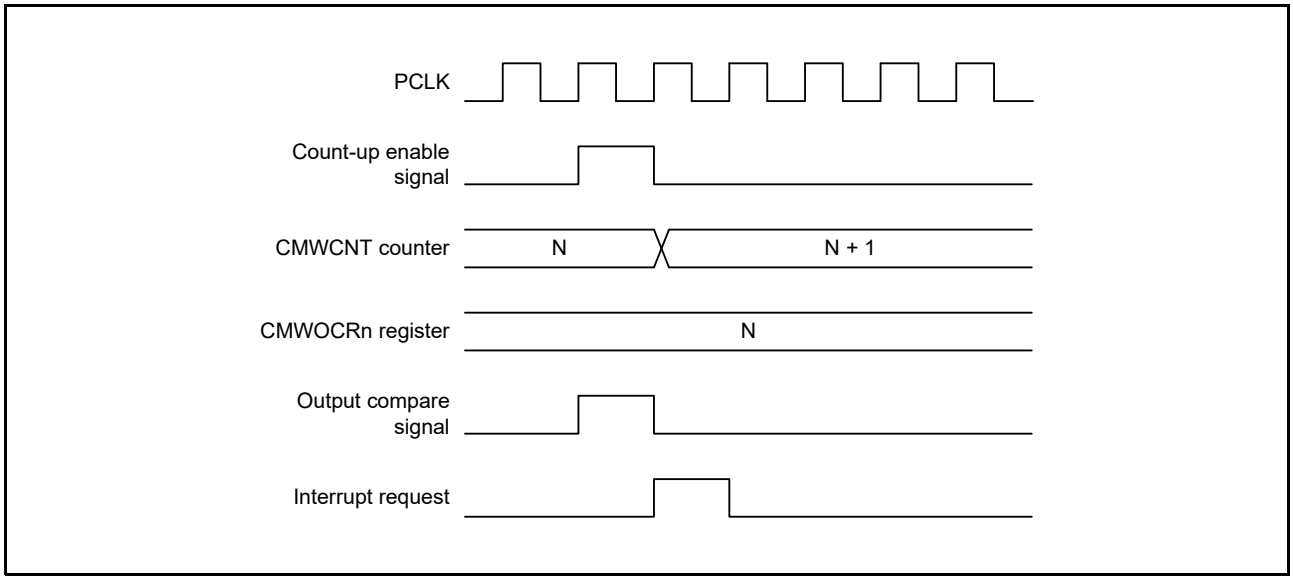


Figure 29.14 Timing of Output Compare Interrupt Generation (n = 0, 1)

(b) Timing of Input Capture Interrupt Generation

Figure 29.15 shows the timing of input capture interrupt generation.

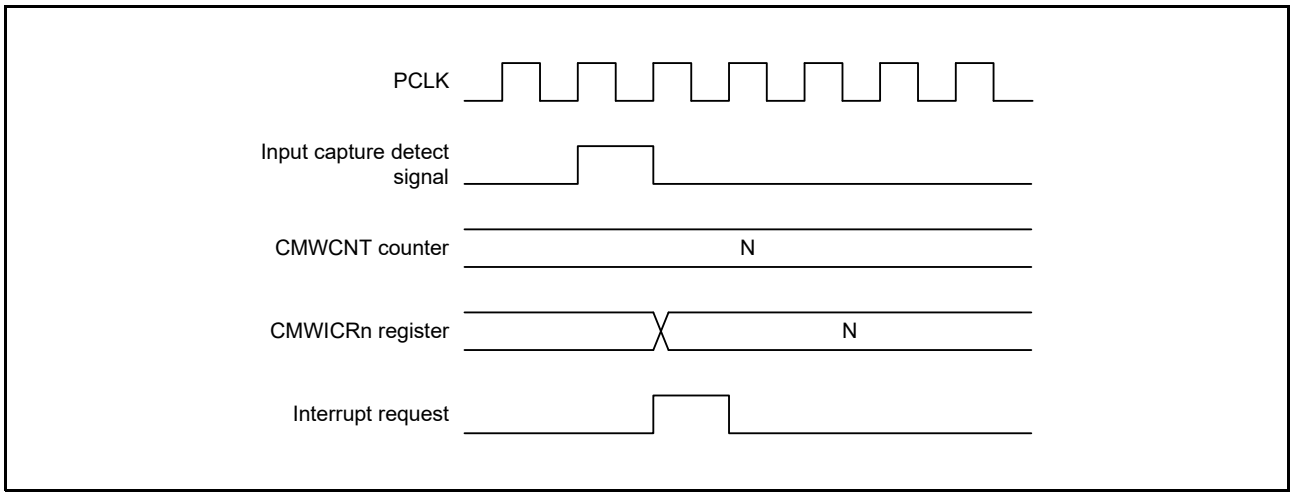


Figure 29.15 Timing of Input Capture Interrupt Generation (n = 0, 1)

29.5 Link Operations by ELC

29.5.1 Event Signal Output to ELC

The CMTW uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal.
The CMTW outputs the event signal upon a compare match. The corresponding channel is channel 0. The event signal can be output regardless of the corresponding interrupt request enable bit (CMWCR.CMWIE).
For details, refer to section 21, Event Link Controller (ELC).

(1) Compare Match Event

In response to a compare match, the CMTW simultaneously issues an interrupt request and a compare match event signal to the ELC. The event signal is issued regardless of the settings of the corresponding interrupt request enable bit (CMWCR.CMWIE bit).

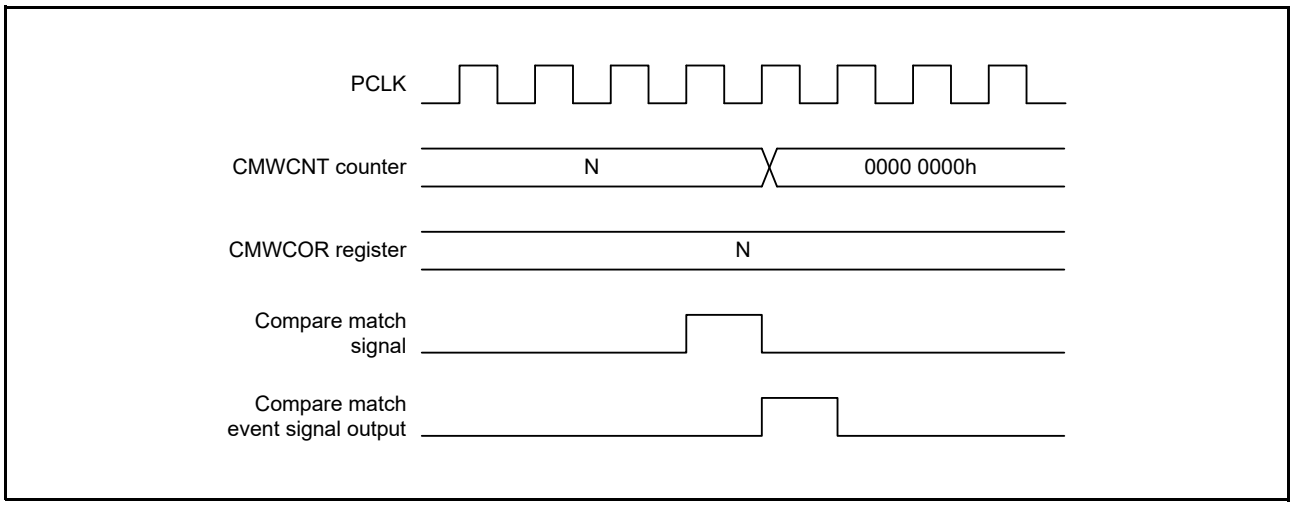


Figure 29.16 Timing of Issuing a Compare Match Event Signal

29.5.2 CMTW Operation When Receiving an Event Signal from ELC

The CMTW can perform any of the following operations according to the event preset by the ELSRn register of the ELC.

(1) Count Start

The CMTW count start operation is selected by the ELOPB register of the ELC. If the event specified by the ELSRn register occurs, the CMWSTR.STR bit is set to 1, starting the CMTW count operation.
However, if the specified event occurs while the CMWSTR.STR bit is 1, the event is ignored.

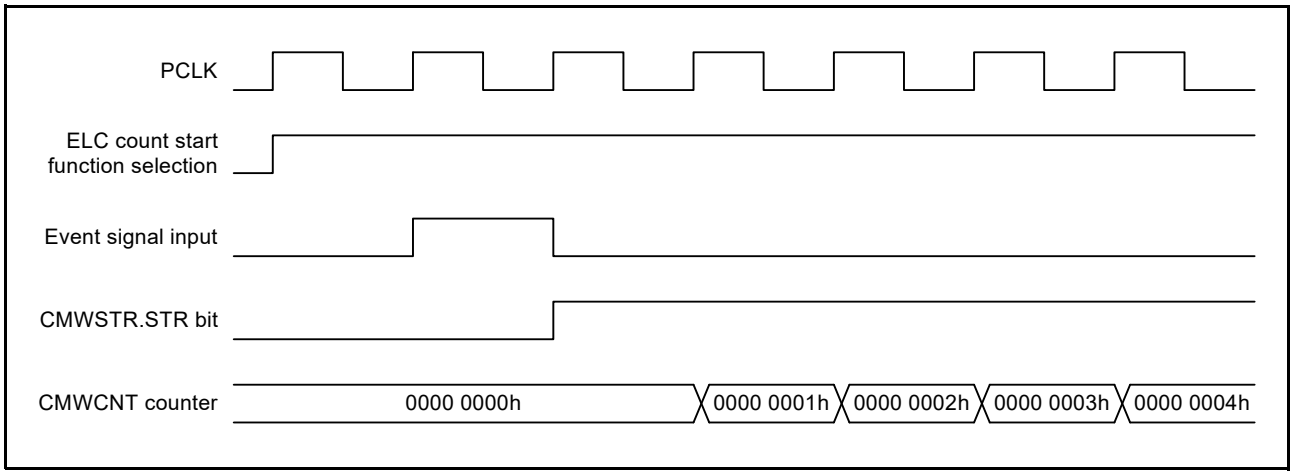


Figure 29.17 Count Start Operation on Acceptance of the Event Signal

(2) Event Count

The CMTW event count operation is selected by the ELOPB register of the ELC. If the event specified by the ELSRn register occurs when the CMWSTR.STR bit is 1, the event is counted as the count source regardless of the CMWCR.CKS[1:0] bit setting.
Reading the counter value returns the number of events that have been actually input.

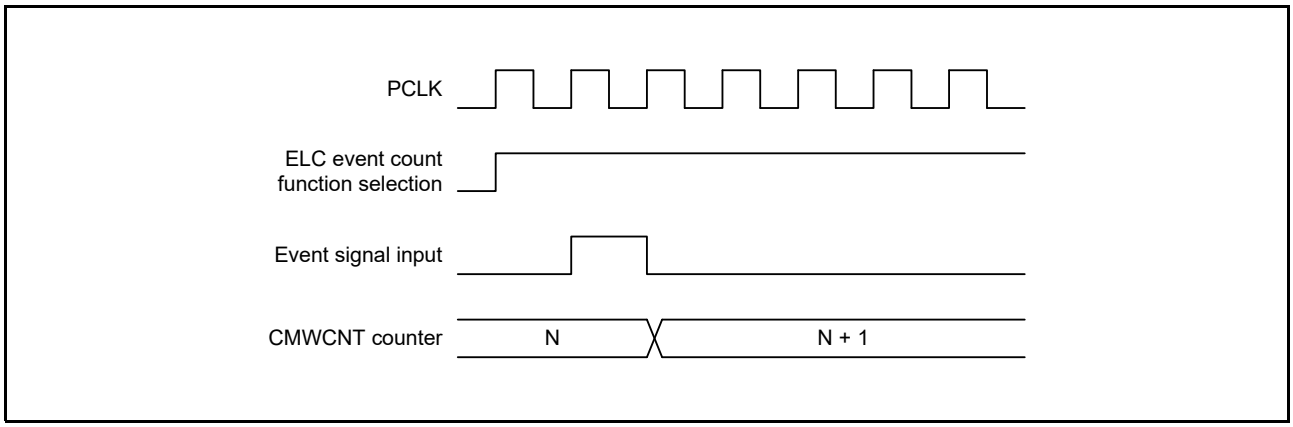


Figure 29.18 Event Count Operation on Acceptance of the Event Signal

(3) Count Restart

The CMTW count restart operation is selected by the ELOPB register of the ELC. If the event specified by the ELSRn register occurs, the CMWCNT counter value becomes 0000 0000h. If the CMWSTR.STR bit is 1, the count operation can be continued.

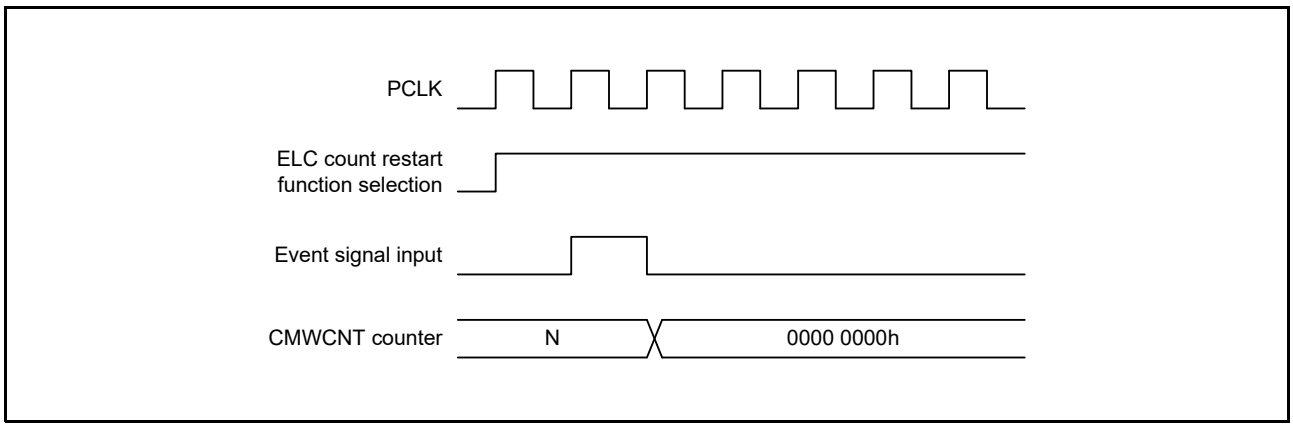


Figure 29.19 Count Restart Operation on Acceptance of the Event Signal

29.5.3 Conflict between Event Link Operation and Register Access

The followings are the notes on using CMTW for event link operations.
Table 29.4 lists count operations when event link operation and register access conflict.

Table 29.4 Count Operations When Event Link Operation and Register Access Conflict (n = 0, 1)

Event Link Operation	Register Access	CMWCNT Counter Status	Operation to be Performed
Count start	Writing to the CMWSTR.STR bit	Stopped state	Count start
		Compare match	Count start
		Counting up	Count start
Event count	Writing to CMWCNT counter	—	Event count
	Writing to CMWCOR register	Compare match	Compare match
Count restart	Writing to CMWCNT counter	Other than compare match	Count restart
	Writing to CMWCNT counter	Compare match	Compare match
	(No access to registers)	Compare match	Compare match
(No events)	Writing to CMWCNT counter	Compare match	Output of compare match interrupt request Writing to CMWCNT counter
		Counting up	Writing to CMWCNT counter
	Writing to CMWCOR register	Compare match	Compare match
	Writing to CMWOCRn register	Compare match	Compare match
	Reading from CMWCNT counter	Counting up	Counting up and reading of the previous value

(1) Count Start

When writing to the STR bit in the CMWSTR register and acceptance of the event signal are in contention, writing to the STR bit does not proceed since setting of the STR bit to 1 in response to the event takes priority.

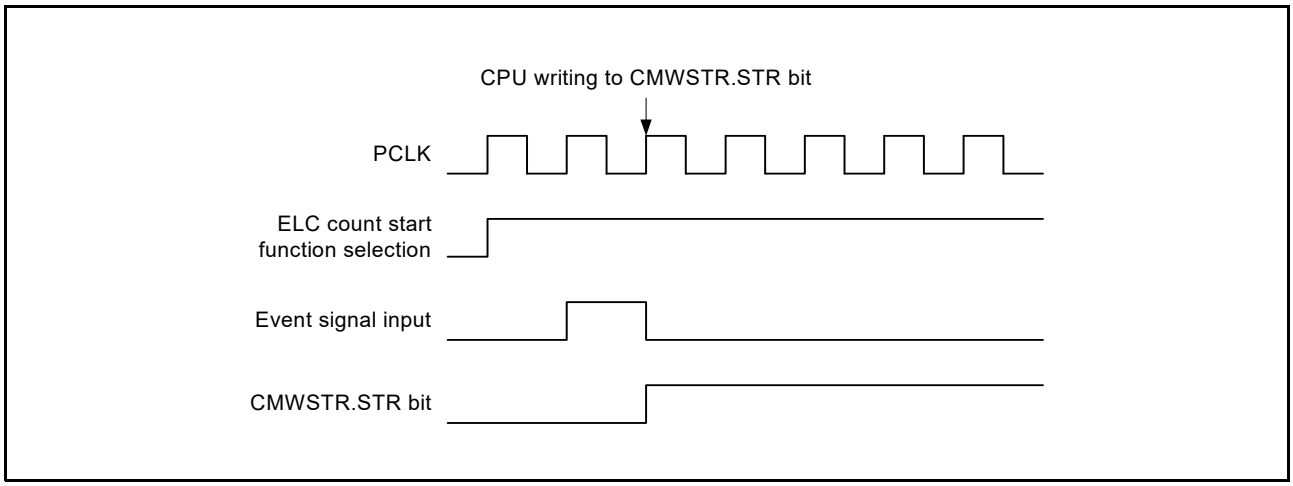


Figure 29.20 Conflict between Event Acceptance and Register Access in Count Start Operation

(2) Event Count

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the count operation in response to the event takes priority.

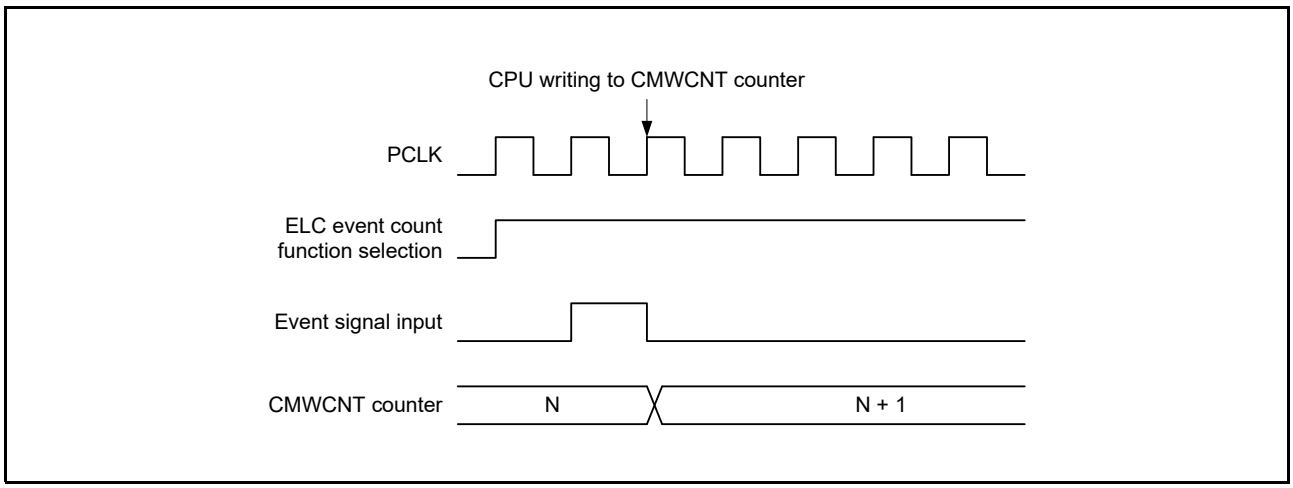


Figure 29.21 Conflict between Event Acceptance and Register Access in Event Counting Operation

(3) Count Restart

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the counter value initialization in response to the event occurrence takes priority.

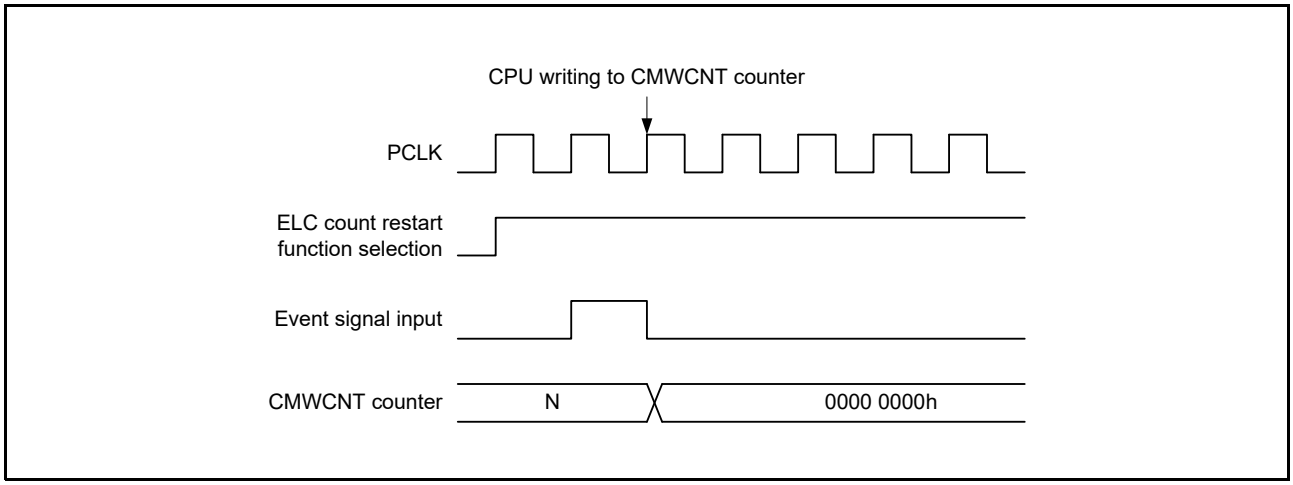


Figure 29.22 Conflict between Event Acceptance and Register Access in Count Restart Operation

29.6 Usage Notes

29.6.1 Setting the Module Stop Function

The CMTW operation can be enabled or disabled using the MSTPCRF register. The CMTW0 and CMTW1 are initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

29.6.2 Conflict between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during writing to the CMWCNT counter, the compare match request is output but the counter is not cleared since writing to the counter takes priority.
Figure 29.23 shows the timing of conflict between CMWCNT counter writing and compare match.

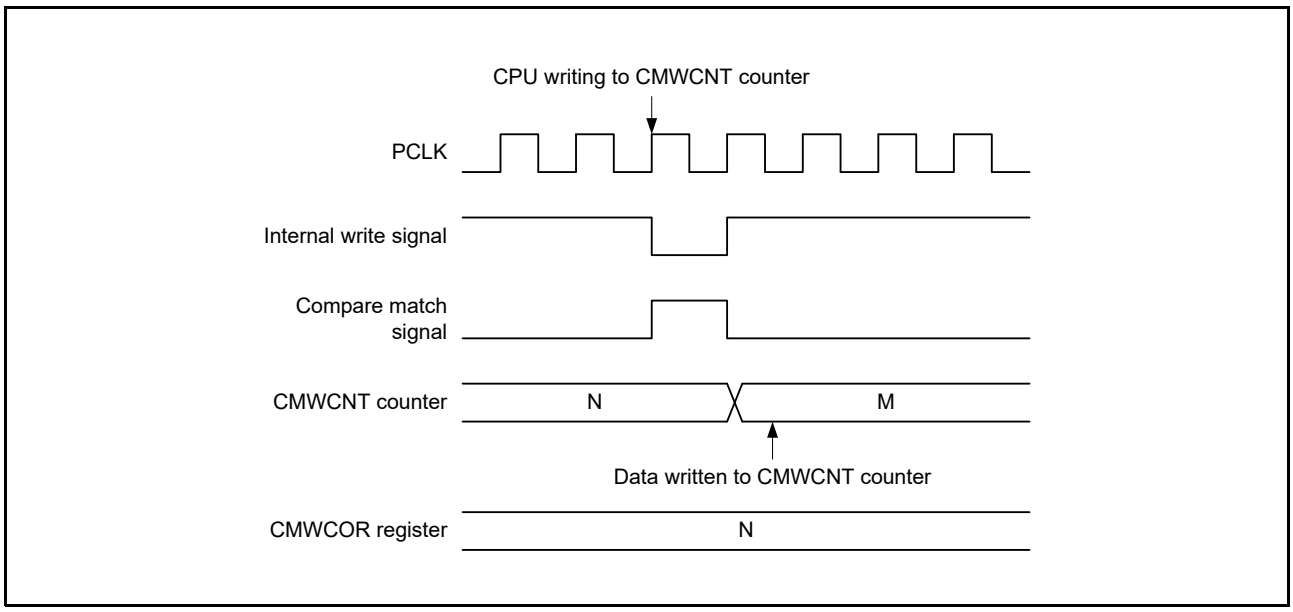


Figure 29.23 Conflict between CMWCNT Counter Writing and Compare Match

29.6.3 Conflict between CMWCNT Counter Writing and Incrementing or Clearing

In case of conflict between incrementation or clearing of the CMWCNT counter and writing to the CMWCNT counter, the counter is not actually incremented or cleared since writing to the CMWCNT counter takes priority.
Figure 29.24 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

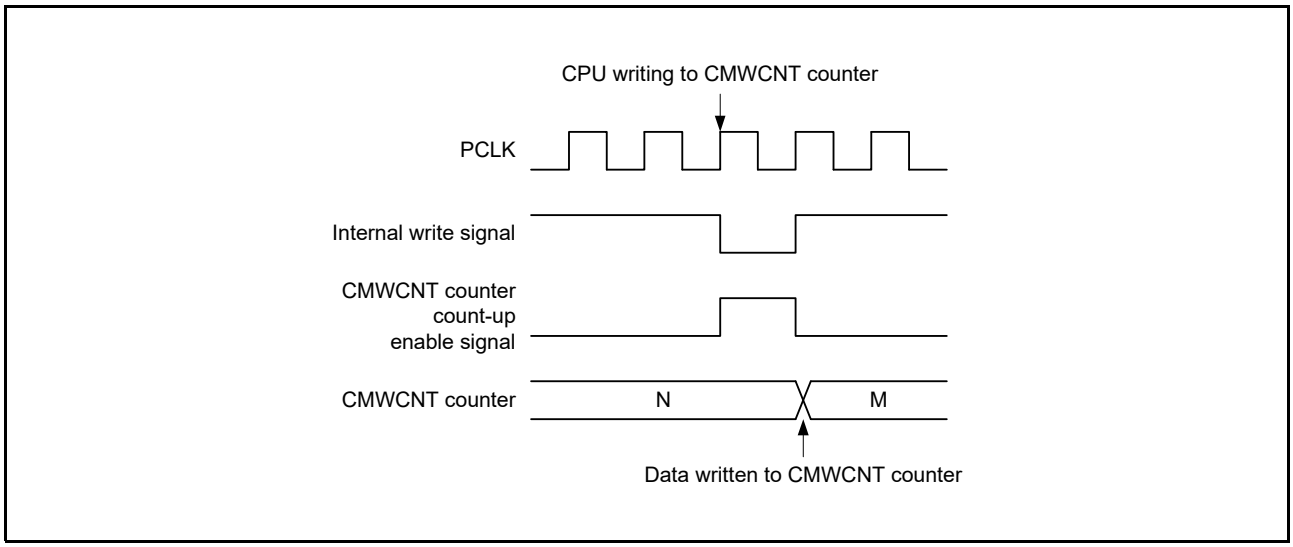


Figure 29.24 Conflict between CMWCNT Counter Writing and Incrementing

29.6.4 Conflict between CMWCOR Register Writing and Compare Match

If the compare match is generated during the CMWCOR register write cycle, the writing to the CMWCOR register takes priority and also the compare match signal is output.
Figure 29.25 shows the timing of conflict between CMWCOR register writing and compare match.

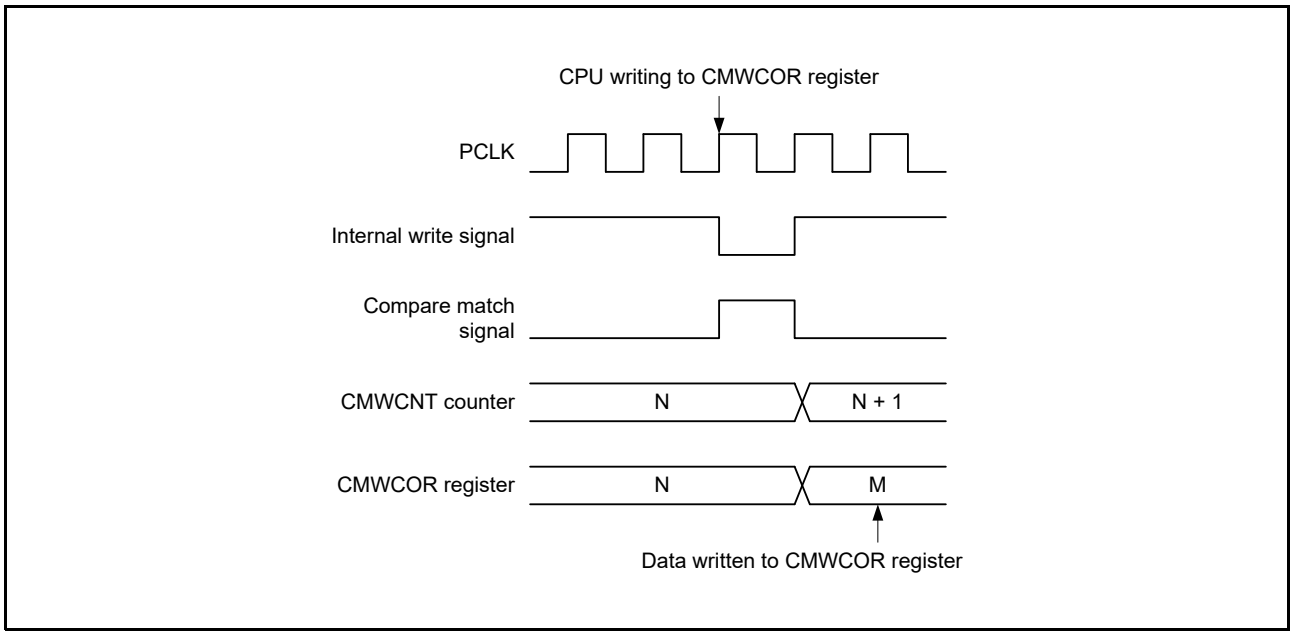


Figure 29.25 Conflict between CMWCOR Register Writing and Compare Match

29.6.5 Conflict between CMWOCRn Register Writing and Compare Match (n = 0, 1)

If the compare match is generated during the CMWOCRn register write cycle, the writing to the CMWOCRn register takes priority and also the compare match signal is output.
Figure 29.26 shows the timing of conflict between CMWOCRn register writing and compare match.

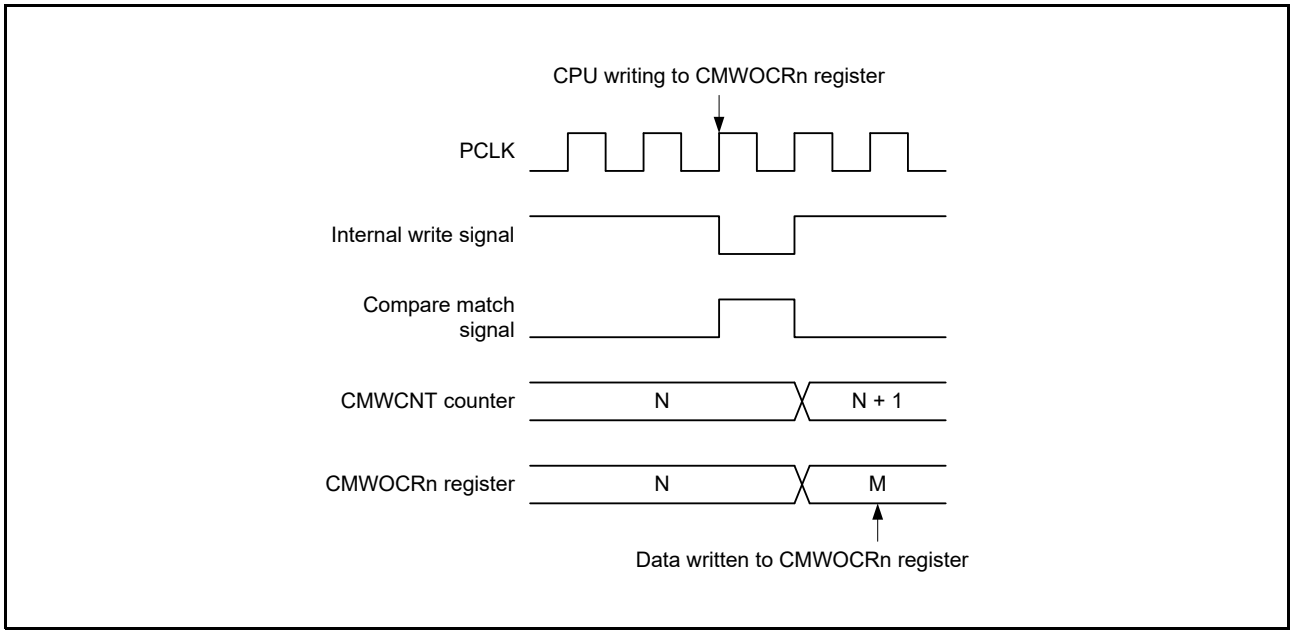


Figure 29.26 Conflict between CMWOCRn Register Writing and Compare Match

29.6.6 Conflict between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.
Figure 29.27 shows the timing of conflict between CMWCNT counter reading and incrementing.

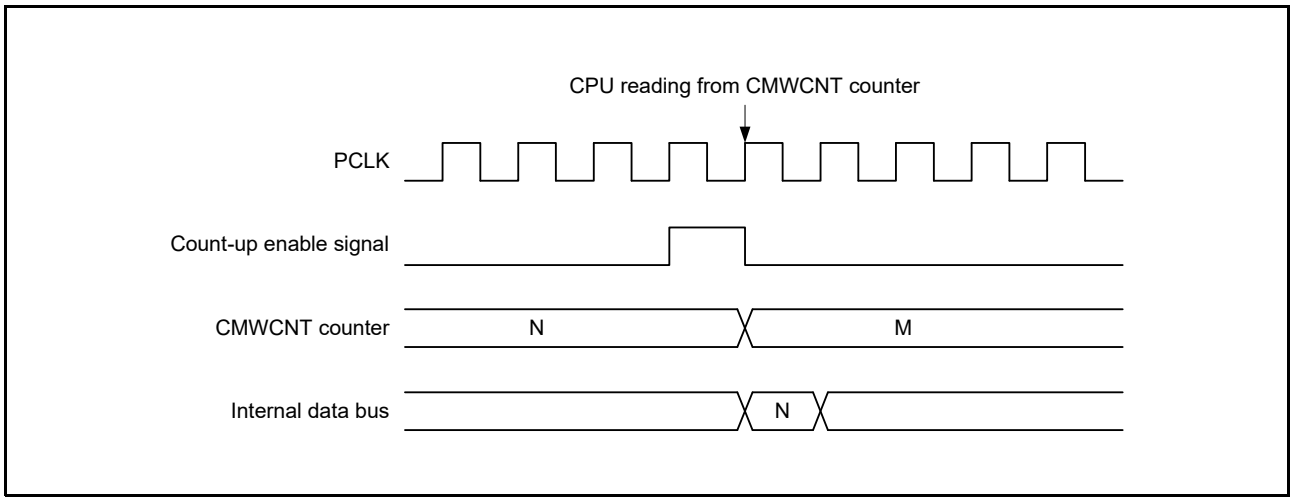


Figure 29.27 Conflict between CMWCNT Counter Reading and Incrementing

29.6.7 Conflict between CMWICRn Register Reading and Input Capture (n = 0, 1)

If the input capture detection signal is generated at the same time that the data of the CMWICRn register is read, the value having been in the CMWICRn register before updated by input capture transfer is read.
Figure 29.28 shows the timing of conflict between CMWICRn register reading and input capture.

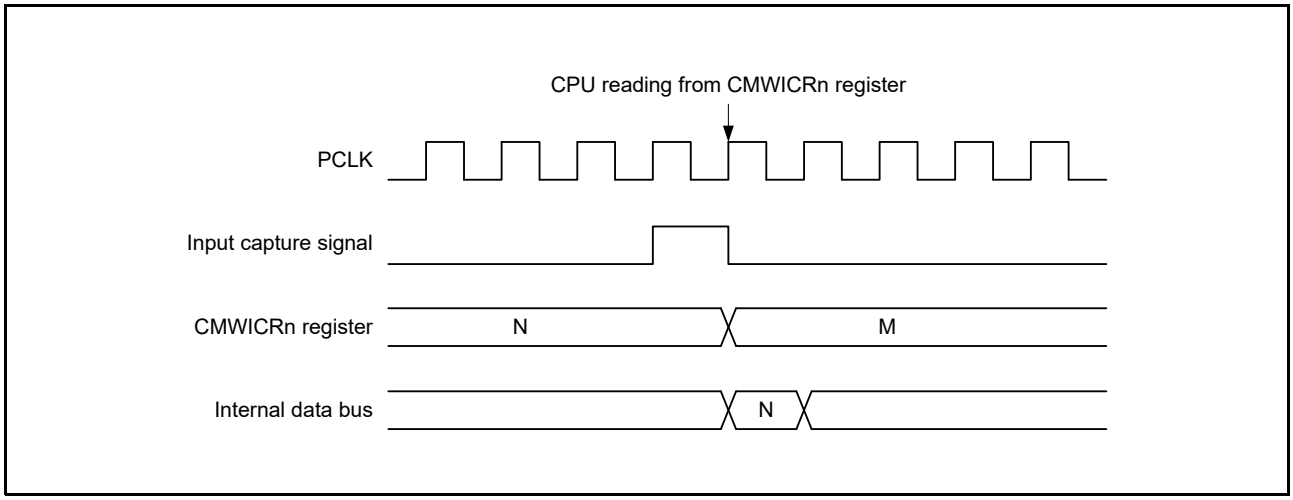


Figure 29.28 Conflict between CMWICRn Register Reading and Input Capture

29.6.8 Transition to Software Standby Mode

Before transitioning to software standby mode, set the CMWSTR.STR bits to 0 for all channels to stop the counters. After returning from software standby mode, setting the CMWSTR.STR bit to 1 will resume counting from the value held prior to the transition.

30. Realtime Clock (RTCBb)

30.1 Overview

The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.

For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.

For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.

The clock source of the time counters is selectable as the sub-clock or LOCO clock.

The RTC uses the 128-Hz clock which is acquired by the clock source divided by the prescaler as the reference clock.

Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

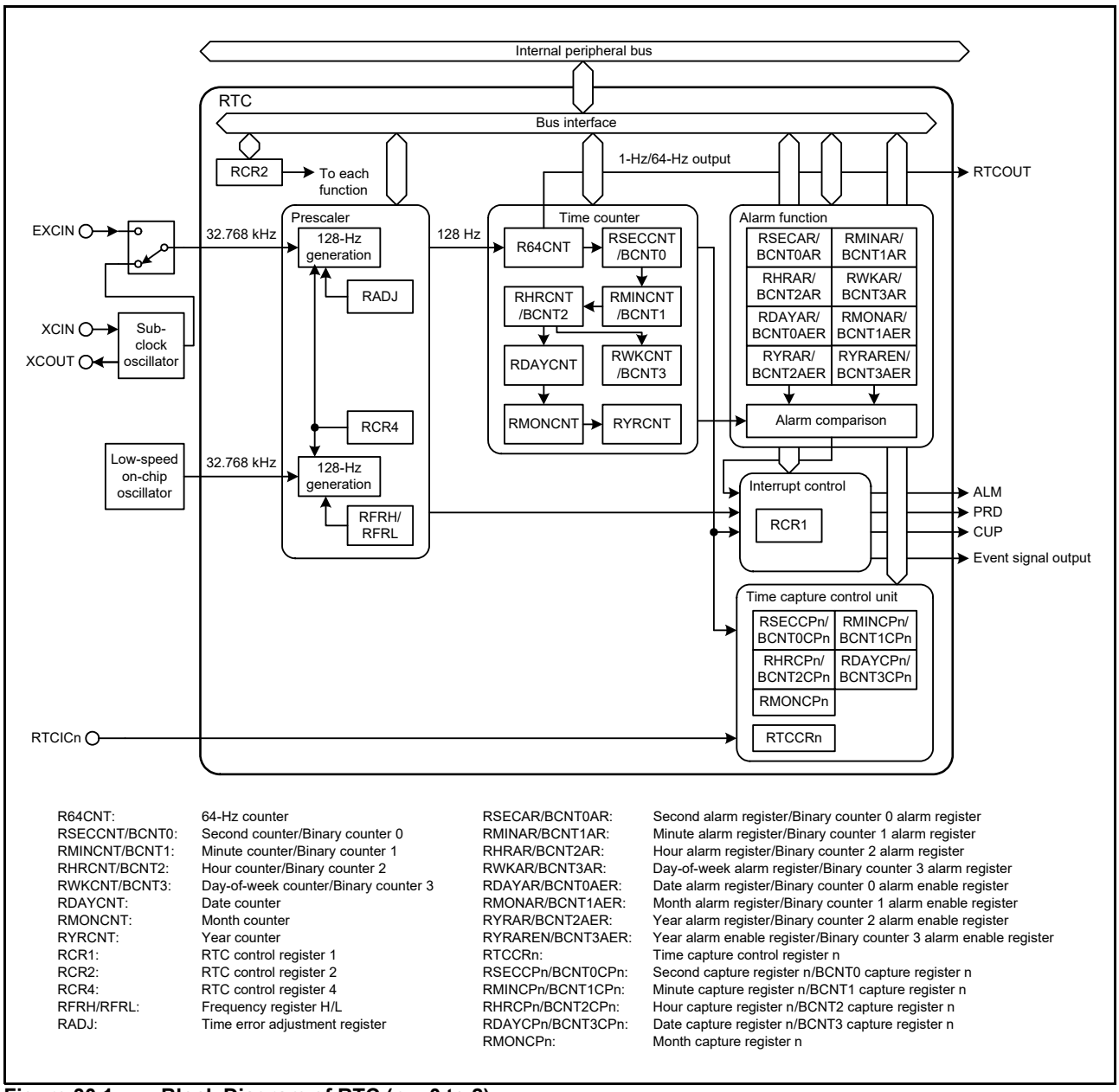
Table 30.1 lists the specifications of the RTC, Figure 30.1 shows a block diagram of the RTC, and Table 30.2 shows the pin configuration of the RTC.

Table 30.1 RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Clock source*1	Sub-clock (XCIN or EXCIN) or LOCO clock
Clock and calendar functions	• Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1 Hz/64 Hz) output
Interrupts	• Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: - Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected - Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) An interrupt is generated at either of the following timings: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time. • Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output
Protect Zone filter	Security attribute can be set.*2

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the clock source.

Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.



R64CNT:	64-Hz counter	RSECAR/BCNT0AR:	Second alarm register/Binary counter 0 alarm register
RSECCNT/BCNT0:	Second counter/Binary counter 0	RMINAR/BCNT1AR:	Minute alarm register/Binary counter 1 alarm register
RMINCNT/BCNT1:	Minute counter/Binary counter 1	RHRAR/BCNT2AR:	Hour alarm register/Binary counter 2 alarm register
RHRCNT/BCNT2:	Hour counter/Binary counter 2	RWKAR/BCNT3AR:	Day-of-week alarm register/Binary counter 3 alarm register
RWKCNT/BCNT3:	Day-of-week counter/Binary counter 3	RDAYAR/BCNT0AER:	Date alarm register/Binary counter 0 alarm enable register
RDAYCNT:	Date counter	RMONAR/BCNT1AER:	Month alarm register/Binary counter 1 alarm enable register
RMONCNT:	Month counter	RYRAR/BCNT2AER:	Year alarm register/Binary counter 2 alarm enable register
RYRCNT:	Year counter	RYRAREN/BCNT3AER:	Year alarm enable register/Binary counter 3 alarm enable register
RCR1:	RTC control register 1	RTCCRn:	Time capture control register n
RCR2:	RTC control register 2	RSECCPn/BCNT0CPn:	Second capture register n/BCNT0 capture register n
RCR4:	RTC control register 4	RMINCPn/BCNT1CPn:	Minute capture register n/BCNT1 capture register n
RFRH/RFRL:	Frequency register H/L	RHRCPn/BCNT2CPn:	Hour capture register n/BCNT2 capture register n
RADJ:	Time error adjustment register	RDAYCPn/BCNT3CPn:	Date capture register n/BCNT3 capture register n
		RMONCPn:	Month capture register n

Figure 30.1 Block Diagram of RTC (n = 0 to 2)

Table 30.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	These pins are used to connect a 32.768-kHz crystal for the sub-clock.
XCOUT	Output	
EXCIN	Input	This pin is used to input a 32.768-kHz external clock for the sub-clock.
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in deep software standby mode.
RTCIC0	Input	These pins are used to input time capture events
RTCIC1	Input	
RTCIC2	Input	

30.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 30.6.5, Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. Note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode or deep software standby mode immediately after setting any of these registers. For details, refer to section 30.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

30.2.1 64-Hz Counter (R64CNT)

Address(es): RTC.R64CNT 8780 2000h, RTC_NS.R64CNT 0780 2000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates a period of one second by counting the 128-Hz reference clock.

The state in the sub-second range can be confirmed by reading this counter.

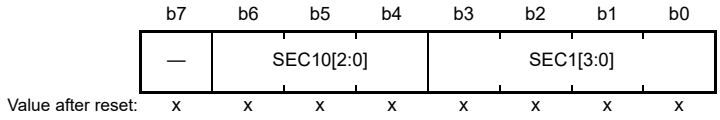
This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.

30.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 8780 2002h, RTC_NS.RSECCNT 0780 2002h



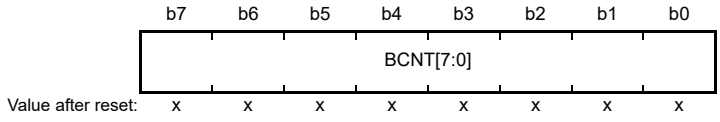
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RSECCNT register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 8780 2002h, RTC_NS.BCNT0 0780 2002h



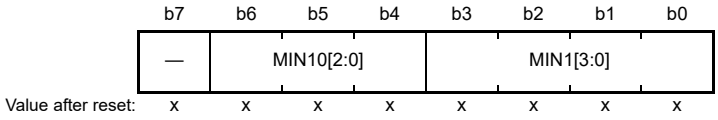
x: Undefined

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.

30.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 8780 2004h, RTC_NS.RMINCNT 0780 2004h



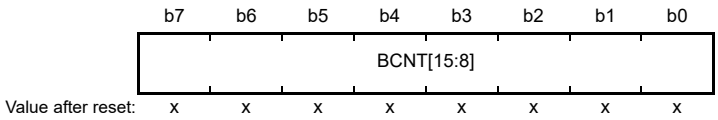
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMINCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 8780 2004h, RTC_NS.BCNT1 0780 2004h



x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.

30.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 8780 2006h, RTC_NS.RHRCNT 0780 2006h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT2 8780 2006h, RTC_NS.BCNT2 0780 2006h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.

30.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 8780 2008h, RTC_NS.RWKCNT 0780 2008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT3 8780 2008h, RTC_NS.BCNT3 0780 2008h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.

30.2.6 Date Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 8780 200Ah, RTC_NS.RDAYCNT 0780 200Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode.
The RDAYCNT counter is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

30.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 8780 200Ch, RTC_NS.RMONCNT 0780 200Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode.
The RMONCNT counter is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMONCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

30.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 8780 200Eh, RTC_NS.RYRCNT 0780 200Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode.
The RYRCNT counter is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.
A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RYRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

30.2.9 Second Alarm Register (RSECAR)/ Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): RTC.RSECAR 8780 2010h, RTC_NS.RSECAR 0780 2010h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	0: The register value is not compared with the RSECCNT counter value. 1: The register value is compared with the RSECCNT counter value.	R/W

The RSECAR register is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.
RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RSECAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 8780 2010h, RTC_NS.BCNT0AR 0780 2010h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AR register is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is set to 00h by an RTC software reset.

30.2.10 Minute Alarm Register (RMINAR)/ Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 8780 2012h, RTC_NS.RMINAR 0780 2012h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value. 1: The register value is compared with the RMINCNT counter value.	R/W

The RMINAR register is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.
RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMINAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 8780 2012h, RTC_NS.BCNT1AR 0780 2012h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AR register is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is set to 00h by an RTC software reset.

30.2.11 Hour Alarm Register (RHRAR)/ Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): RTC.RHRAR 8780 2014h, RTC_NS.RHRAR 0780 2014h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value. 1: The register value is compared with the RHRCNT counter value.	R/W

The RHRAR register is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
After writing to the RHRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 8780 2014h, RTC_NS.BCNT2AR 0780 2014h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AR register is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is set to 00h by an RTC software reset.

30.2.12 Day-of-Week Alarm Register (RWKAR)/ Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): RTC.RWKAR 8780 2016h, RTC_NS.RWKAR 0780 2016h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value. 1: The register value is compared with the RWKCNT counter value.	R/W

The RWKAR register is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.

RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RWKAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AR 8780 2016h, RTC_NS.BCNT3AR 0780 2016h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AR register is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is set to 00h by an RTC software reset.

30.2.13 Date Alarm Register (RDAYAR)/ Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): RTC.RDAYAR 8780 2018h, RTC_NS.RDAYAR 0780 2018h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value. 1: The register value is compared with the RDAYCNT counter value.	R/W

The RDAYAR register is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.

RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RDAYAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AER 8780 2018h, RTC_NS.BCNT0AER 0780 2018h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) becomes 1.

This register is set to 00h by an RTC software reset.

30.2.14 Month Alarm Register (RMONAR)/ Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): RTC.RMONAR 8780 201Ah, RTC_NS.RMONAR 0780 201Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value. 1: The register value is compared with the RMONCNT counter value.	R/W

The RMONAR register is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.
RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMONAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AER 8780 201Ah, RTC_NS.BCNT1AER 0780 201Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) becomes 1.
This register is set to 00h by an RTC software reset.

30.2.15 Year Alarm Register (RYRAR)/ Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 8780 201Ch, RTC_NS.RYRAR 0780 201Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRAR register is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RYRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 8780 201Ch, RTC_NS.BCNT2AER 0780 201Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) becomes 1.
This register is set to 0000h by an RTC software reset.

30.2.16 Year Alarm Enable Register (RYRAREN)/ Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 8780 201Eh, RTC_NS.RYRAREN 0780 201Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value. 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 8780 201Eh, RTC_NS.BCNT3AER 0780 201Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) becomes 1.
This register is set to 00h by an RTC software reset.

30.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 8780 2022h, RTC_NS.RCR1 0780 2022h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled. 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz. 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second.*1 0 1 1 1: A periodic interrupt is generated every 1/128 second. 1 0 0 0: A periodic interrupt is generated every 1/64 second. 1 0 0 1: A periodic interrupt is generated every 1/32 second. 1 0 1 0: A periodic interrupt is generated every 1/16 second. 1 0 1 1: A periodic interrupt is generated every 1/8 second. 1 1 0 0: A periodic interrupt is generated every 1/4 second. 1 1 0 1: A periodic interrupt is generated every 1/2 second. 1 1 1 0: A periodic interrupt is generated every 1 second. 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

Note 1. When the LOCO clock is selected (RCR4.RCKSEL bit = 1) while the PES[3:0] bits are 0110b, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated synchronously with the clock source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.
If the times indicated by the counters and alarm settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the AIE bit.

CIE Bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.
If the periods indicated by the counters and PES[3:0] settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the PIE bit.

RTCOS Bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

30.2.18 RTC Control Register 2 (RCR2)

Address(es): RTC.RCR2 8780 2024h, RTC_NS.RCR2 0780 2024h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start*4	0: Prescaler and counter are stopped. 1: Prescaler and counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and the target registers for RTC software reset*1 are initialized. • In reading 0: In normal time operation, or an RTC software reset has completed. 1: During an RTC software reset	R/W
b2	ADJ30	30-Second Adjustment*2	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal time operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*3, *4	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3, *4	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute (every 32 seconds in binary counter mode). 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds (every 8 seconds in binary counter mode).	R/W
b6	HR24	Hours Mode*2, *4	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select*4	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRn, RSECCPn/BCNT0CPn, RMINCPn/BCNT1CPn, RHRCPn/BCNT2CPn, RDAYCPn/BCNT3CPn, RMONCPn, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP
Note 2. This bit is reserved in binary counter mode. The write value should be 0.
Note 3. When the LOCO clock is selected, the setting of this bit is disabled.
Note 4. After writing to this bit, confirm that its value has actually changed before proceeding with further processing. Refer to section 30.6.5, Notes on Writing to and Reading from Registers regarding changes to the values of the AADJE, AADJP, and HR24 bits.

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.
The START bit is updated in synchronization with the clock source. When the START bit is modified, check that the bit is updated before proceeding to the next processing.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software reset.
When 1 is written to the RESET bit, the initialization starts in synchronization with the clock source. When the initialization is completed, the RESET bit is automatically set to 0.
When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings.

ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.
The 30-second adjustment is performed in synchronization with the clock source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is set to 0 by an RTC software reset.
This bit is reserved in binary counter mode. The write value should be 0.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT signal is to be output from an external pin, set the RTCOE bit to 1 and set up the port control for the pin.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is set to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is set to 0 by an RTC software reset.

HR24 Bit (Hours Mode)

This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.
This bit is reserved in binary counter mode. The write value should be 0.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
After setting the count mode, execute an RTC software reset and start again from the initial settings.
The CNTMD bit is updated in synchronization with the clock source, so when the value of the CNTMD bit has been changed, check that the value of the bit has actually been updated before applying the RTC software reset. The count mode changes to that which was specified beforehand in the CNTMD bit after the RTC software reset is applied.
For details on initial settings, refer to section 30.3.1, Outline of Initial Settings of Registers after Power On.

30.2.19 RTC Control Register 4 (RCR4)

Address(es): RTC.RCR4 8780 2028h, RTC_NS.RCR4 0780 2028h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RCKSEL	Clock Source Select	0: Sub-clock is selected. 1: LOCO clock is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR4 register is used for selecting the clock source. This function is used in both calendar count mode and binary count mode.
When the RCKSEL bit is set to 0, the time is counted with the sub-clock. When the bit is set to 1, the time is counted with the LOCO clock.

RCKSEL Bit (Clock Source Select)

This bit selects the clock source from the sub-clock and the LOCO clock.
The clock source should be selected only once before the initial settings of the RTC registers at power on.

30.2.20 Frequency Register H/L (RFRH/RFRL)

Address(es): RTC.RFRH 8780 202Ah, RTC_NS.RFRH 0780 202Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC[16]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

x: Undefined

Address(es): RTC.RFRL 8780 202Ch, RTC_NS.RFRL 0780 202Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- RFRH register

Bit	Symbol	Bit Name	Description	R/W
b0	RFC[16]	Frequency Divide Ratio Setting	Set the divide ratio of the LOCO clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the LOCO clock by n + 1.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- RFRL register

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Divide Ratio Setting	Set the divide ratio of the LOCO clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the LOCO clock by n + 1.	R/W

RFRH/RFRL is a register for controlling the prescaler when the LOCO clock is selected.
The RTC time counter operates on a 128-Hz clock signal as the reference clock. Therefore, when the main clock is selected, the LOCO clock is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency divide ratio in the RFC[16:0] bits to generate a 128-Hz clock from the LOCO clock frequency. As for the calculation method, refer to the formula below.

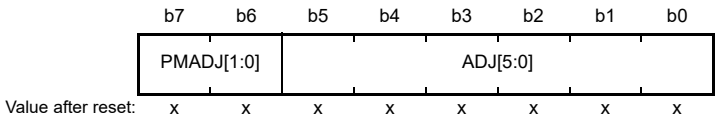
A value from 0 00D9h through 0 0127h can be specified in the RFC[16:0] bits; if a value outside of this range is specified, the RTC does not operate correctly. Rewrite this register while the RCR2.START bit is 0 (counter is stopped). The frequency of the peripheral module clock and the LOCO clock should be in the relationship that the peripheral module clock ≥ the LOCO clock.

- Calculation method of frequency comparison value

RFC[16:0] = (LOCO clock frequency) ÷ 128 – 1

30.2.21 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 8780 202Eh, RTC_NS.RADJ 0780 202Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the clock source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the clock source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the clock source. When RADJ is modified, check that all the bits have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.
The setting of this register is enabled only when the sub-clock is selected.
When the LOCO clock is selected, adjustment is not performed.

ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

30.2.22 Time Capture Control Register n (RTCCRn) (n = 0 to 2)

Address(es): RTC.RTCCR0 8780 2040h, RTC.RTCCR1 8780 2042h, RTC.RTCCR2 8780 2044h, RTC_NS.RTCCR0 0780 2040h, RTC_NS.RTCCR1 0780 2042h, RTC_NS.RTCCR2 0780 2044h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	0	x	x	0	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status Flag	0: No event is detected. 1: An event is detected.	R/(W) *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (clock source). 1 1: The noise filter is on (clock source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input. 1: The RTCICn pin is enabled as the time capture event input.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0.

The RTCCRn register is used both in calendar count mode and in binary count mode.
RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.
RTCCRn is updated in synchronization with the clock source. When RTCCRn is modified, check that all the bits except for the TCST flag have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.

TCCT[1:0] Bits (Time Capture Control)

These bits control the edge detection of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). The detection edge is selectable. The TCCT[1:0] bits should be set while the RCR2.START and TCEN bits are 1.

TCST Flag (Time Capture Status Flag)

This flag indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected.
When the TCST flag is 0, no event is detected.
When the TCST flag is 1, this flag indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.
Writing 0 sets the TCST flag to 0. In addition, writing any other value except 0 has no effect.
Set the TCST flag while the TCCT[1:0] bits are 00b (no event is detected).
The TCST flag is set to 0 in synchronization with the clock source. When the TCST flag is set to 0, check that the flag has been updated before continuing with further processing.

TCNF[1:0] Bits (Time Capture Noise Filter Control)

These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the noise filter is on, the clock source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.
Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCEN bit to 1, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits.

TCEN Bit (Time Capture Event Input Pin Enable)

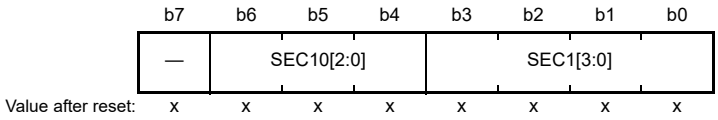
This bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the time capture event input pin functions (RTCIC0, RTCIC1, and RTCIC2) are allocated to the same pins as other multiplexed functions, make the general input port pin settings for the corresponding pins, set the corresponding TAMPICR1.CHnEN bit*1 (n = 0 to 2) to 1, and then set this bit to 1. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

Note 1. For details, refer to section 12, Battery Backup Function (VBATTBa).

30.2.23 Second Capture Register n (RSECCPn) (n = 0 to 2)/ BCNT0 Capture Register n (BCNT0CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 8780 2052h, RTC.RSECCP1 8780 2062h, RTC.RSECCP2 8780 2072h, RTC_NS.RSECCP0 0780 2052h, RTC_NS.RSECCP1 0780 2062h, RTC_NS.RSECCP2 0780 2072h



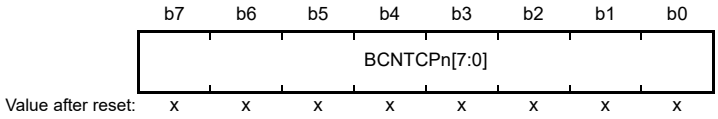
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

The RSECCPn register is a read-only register that captures the RSECCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 8780 2052h, RTC.BCNT0CP1 8780 2062h, RTC.BCNT0CP2 8780 2072h, RTC_NS.BCNT0CP0 0780 2052h, RTC_NS.BCNT0CP1 0780 2062h, RTC_NS.BCNT0CP2 0780 2072h



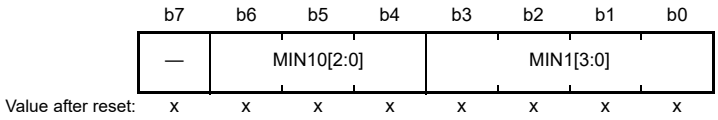
x: Undefined

The BCNT0CPn register is a read-only register that captures the BCNT0 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

30.2.24 Minute Capture Register n (RMINCPn) (n = 0 to 2)/ BCNT1 Capture Register n (BCNT1CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 8780 2054h, RTC.RMINCP1 8780 2064h, RTC.RMINCP2 8780 2074h, RTC_NS.RMINCP0 0780 2054h, RTC_NS.RMINCP1 0780 2064h, RTC_NS.RMINCP2 0780 2074h



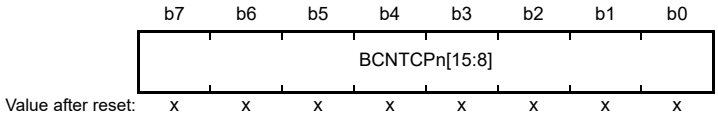
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

The RMINCPn register is a read-only register that captures the RMINCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 8780 2054h, RTC.BCNT1CP1 8780 2064h, RTC.BCNT1CP2 8780 2074h, RTC_NS.BCNT1CP0 0780 2054h, RTC_NS.BCNT1CP1 0780 2064h, RTC_NS.BCNT1CP2 0780 2074h



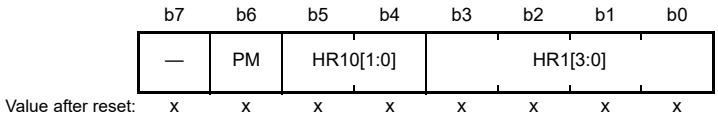
x: Undefined

The BCNT1CPn register is a read-only register that captures the BCNT1 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

30.2.25 Hour Capture Register n (RHRCPn) (n = 0 to 2)/ BCNT2 Capture Register n (BCNT2CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 8780 2056h, RTC.RHRCP1 8780 2066h, RTC.RHRCP2 8780 2076h, RTC_NS.RHRCP0 0780 2056h, RTC_NS.RHRCP1 0780 2066h, RTC_NS.RHRCP2 0780 2076h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

The RHRCPn register is a read-only register that captures the RHRCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

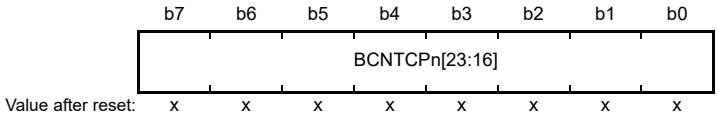
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 8780 2056h, RTC.BCNT2CP1 8780 2066h, RTC.BCNT2CP2 8780 2076h, RTC_NS.BCNT2CP0 0780 2056h, RTC_NS.BCNT2CP1 0780 2066h, RTC_NS.BCNT2CP2 0780 2076h



x: Undefined

The BCNT2CPn register is a read-only register that captures the BCNT2 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.

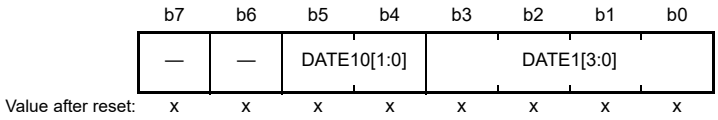
This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

30.2.26 Date Capture Register n (RDAYCPn) (n = 0 to 2)/ BCNT3 Capture Register n (BCNT3CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 8780 205Ah, RTC.RDAYCP1 8780 206Ah, RTC.RDAYCP2 8780 207Ah, RTC_NS.RDAYCP0 0780 205Ah, RTC_NS.RDAYCP1 0780 206Ah, RTC_NS.RDAYCP2 0780 207Ah



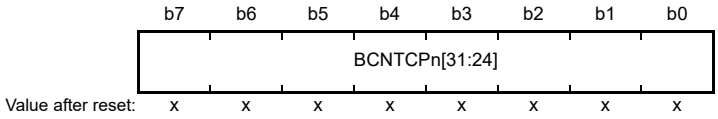
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset.	R

The RDAYCPn register is a read-only register that captures the RDAYCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 8780 205Ah, RTC.BCNT3CP1 8780 206Ah, RTC.BCNT3CP2 8780 207Ah, RTC_NS.BCNT3CP0 0780 205Ah, RTC_NS.BCNT3CP1 0780 206Ah, RTC_NS.BCNT3CP2 0780 207Ah

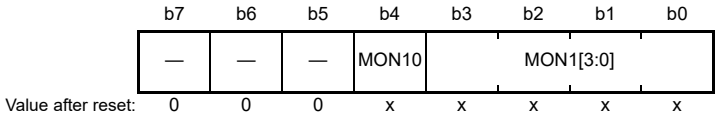


x: Undefined

The BCNT3CPn register is a read-only register that captures the BCNT3 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

30.2.27 Month Capture Register n (RMONCPn) (n = 0 to 2)

Address(es): RTC.RMONCP0 8780 205Ch, RTC.RMONCP1 8780 206Ch, RTC.RMONCP2 8780 207Ch, RTC_NS.RMONCP0 0780 205Ch, RTC_NS.RMONCP1 0780 206Ch, RTC_NS.RMONCP2 0780 207Ch



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	MON10	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

The RMONCPn register is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

30.3 Operation

30.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

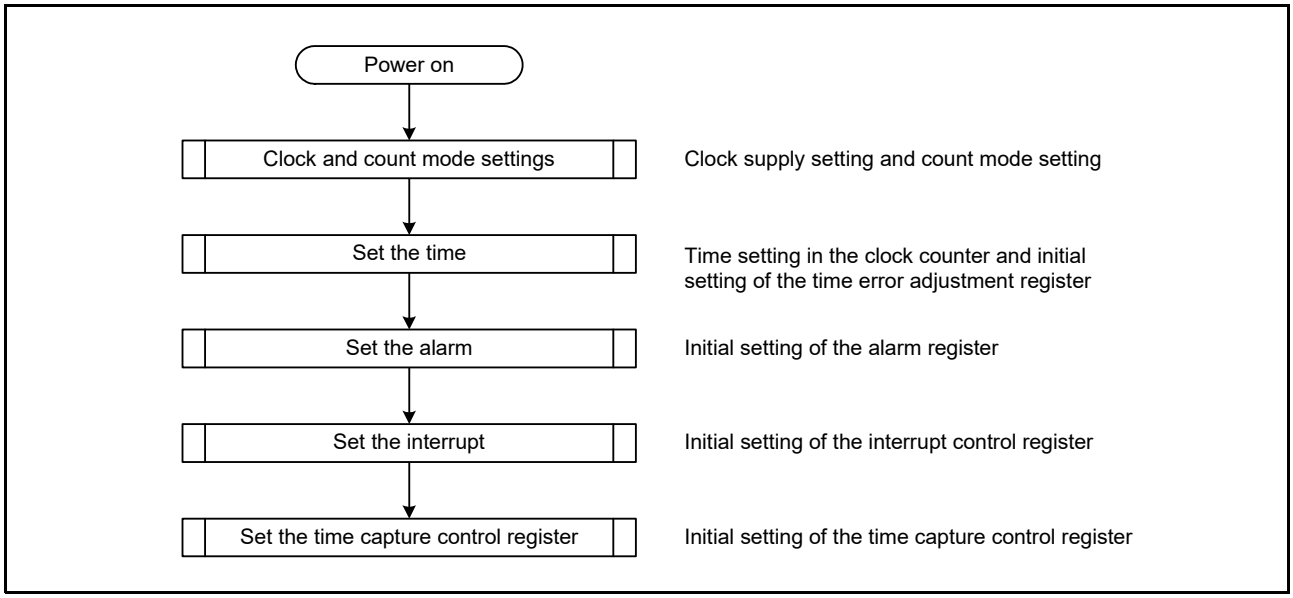
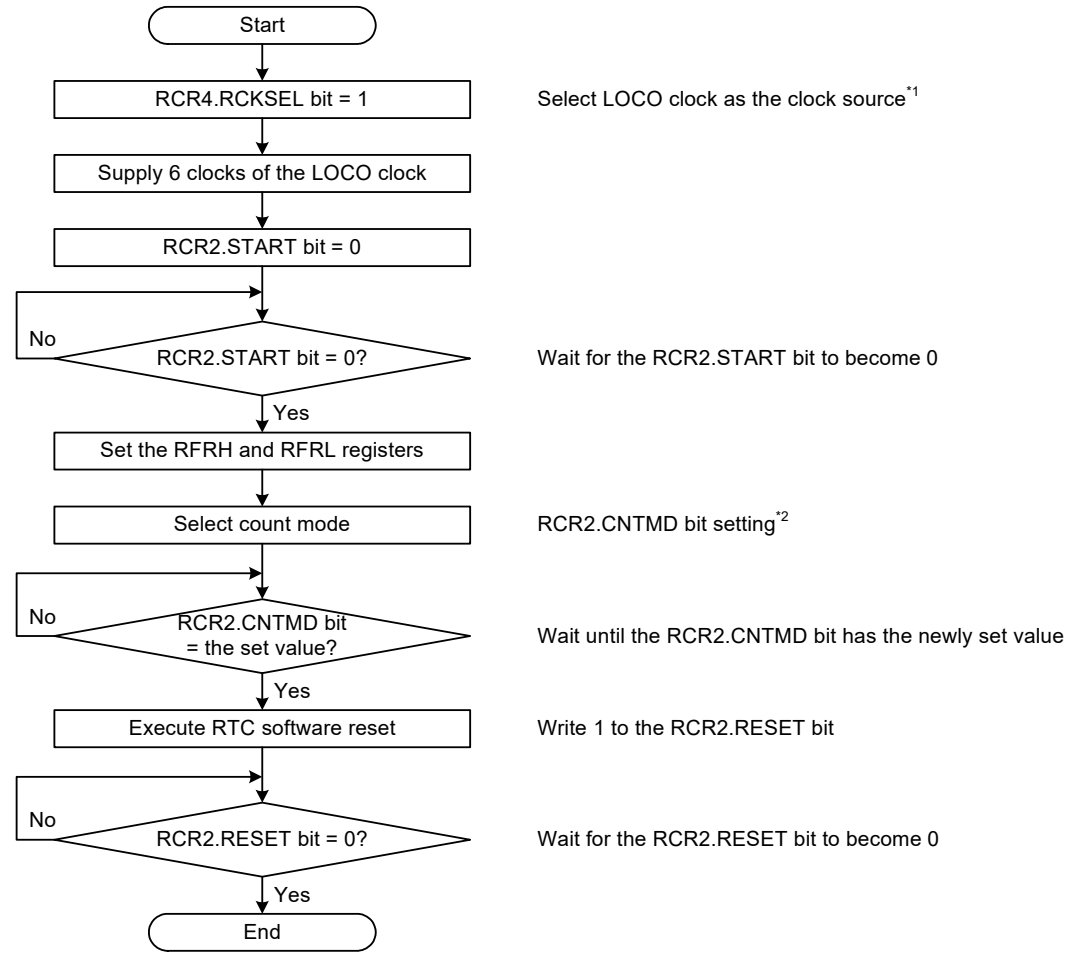


Figure 30.2 Outline of Initial Settings after Power On

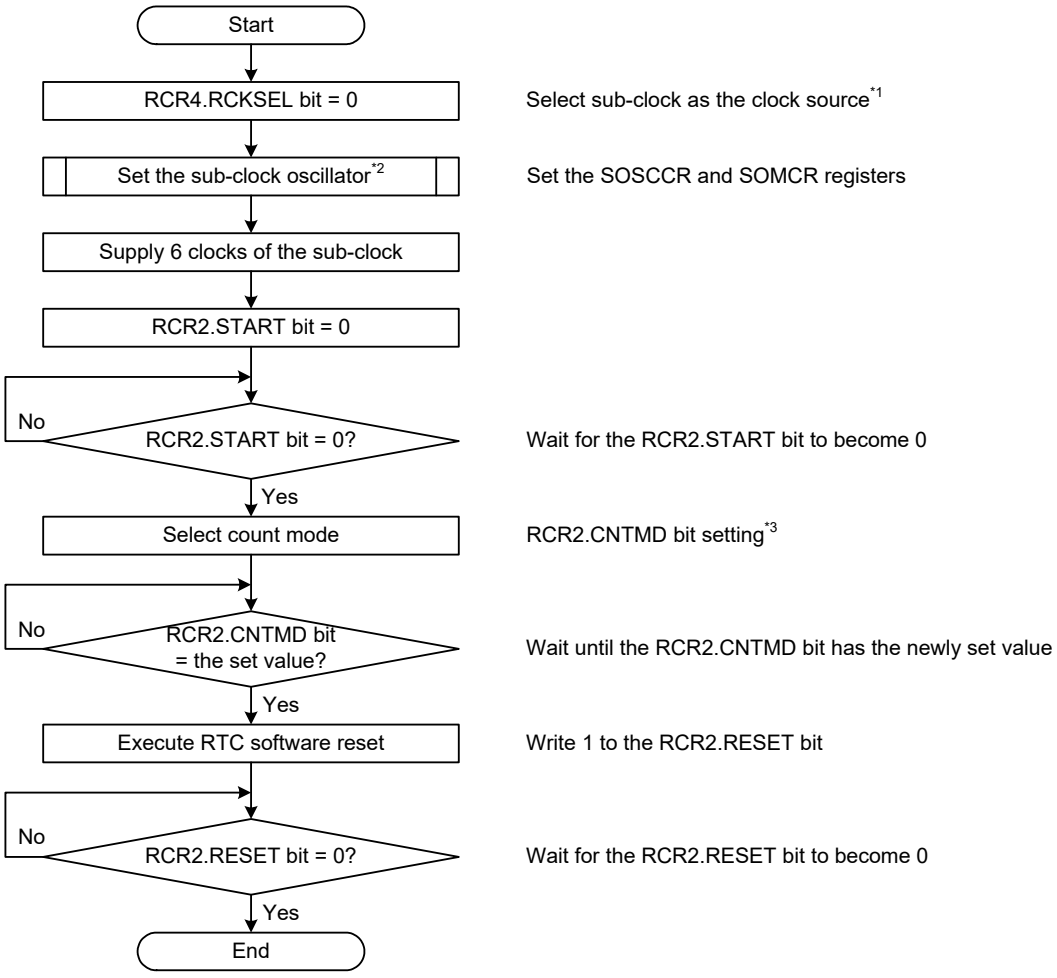
30.3.2 Clock and Count Mode Setting Procedure

Figure 30.3, Figure 30.4, and Figure 30.5 show how to set the clock and the count mode.



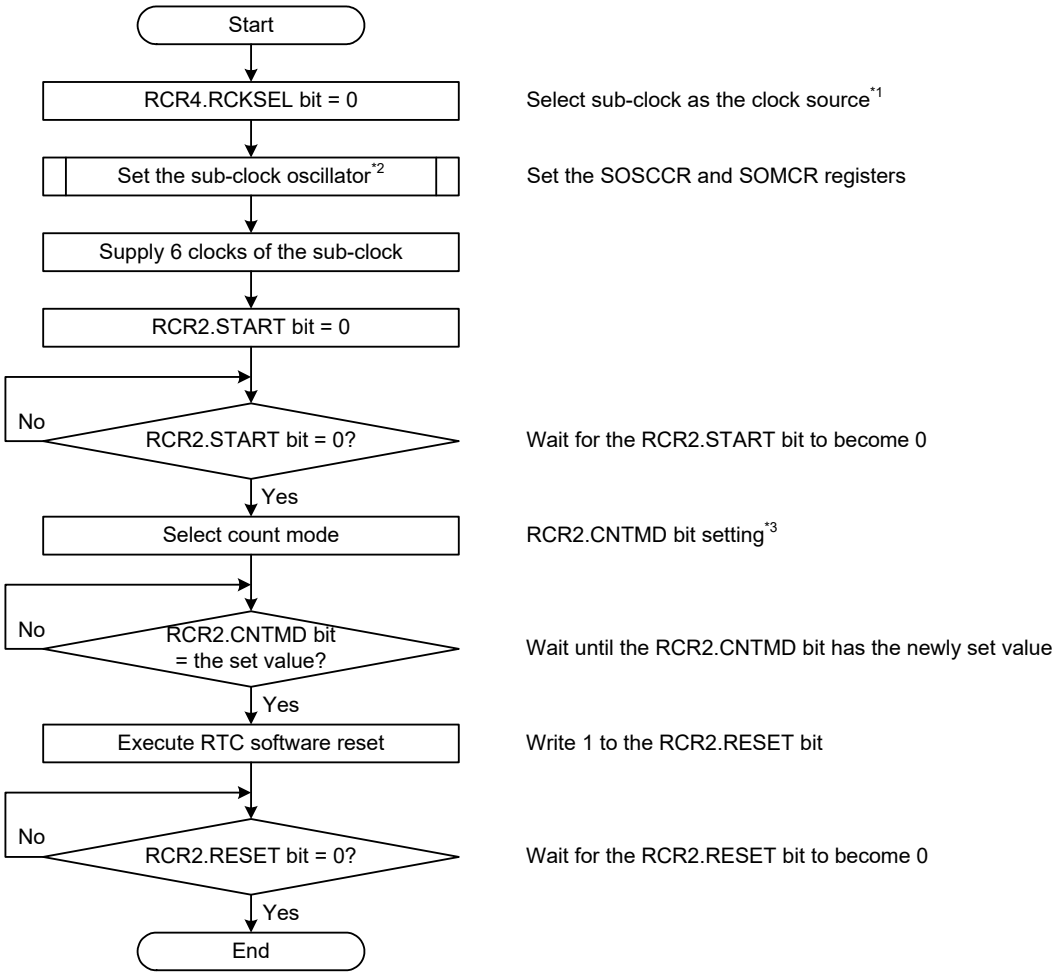
Note 1. Only write to the RCR4.RCKSEL bit at this point in the flow.
Note 2. This step is not necessary if the count mode has been set concurrently with setting the RCR2.START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 30.3 Clock and Count Mode Setting Procedure (When Using LOCO Clock)



Note 1. Only write to the RCR4.RCKSEL bit at this point in the flow.
Note 2. For the procedure for setting up a sub-clock oscillator, refer to section 9, Clock Generation Circuit.
Note 3. This step is not necessary if the count mode has been set concurrently with setting the RCR2.START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 30.4 Clock and Count Mode Setting Procedure (When Using Sub-Clock Oscillator)

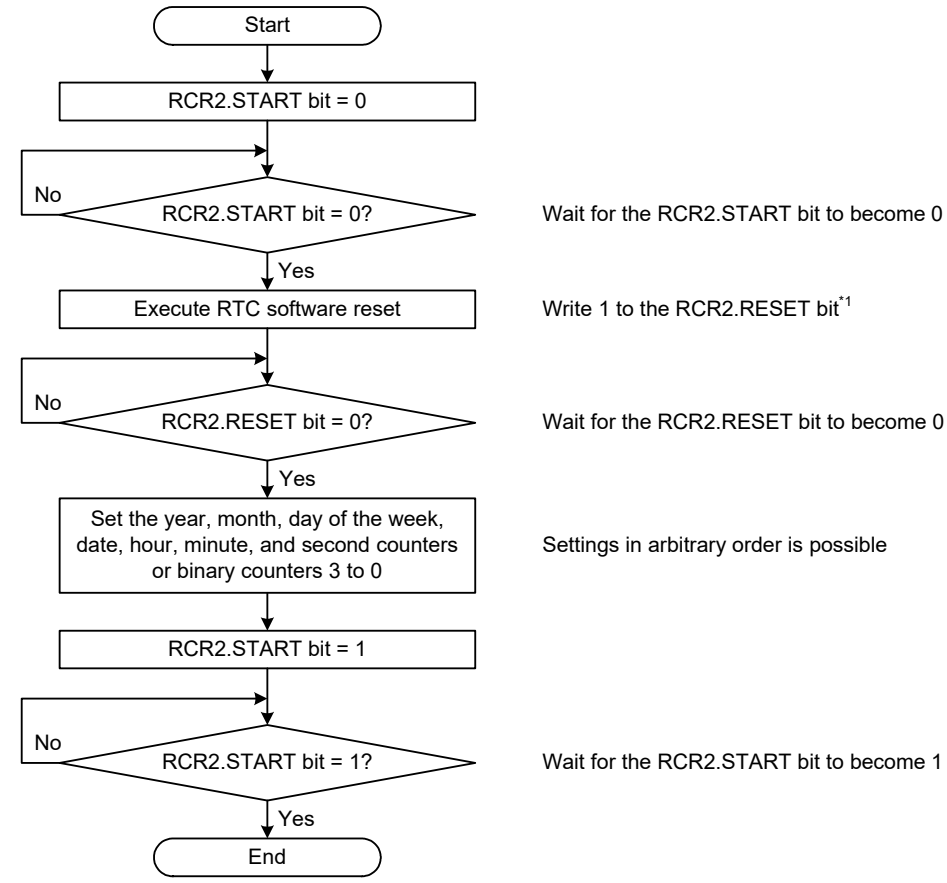


Note 1. Only write to the RCR4.RCKSEL bit at this point in the flow.
Note 2. For the procedure for setting up a sub-clock oscillator, refer to section 9, Clock Generation Circuit.
Note 3. This step is not necessary if the count mode has been set concurrently with setting the RCR2.START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 30.5 Clock and Count Mode Setting Procedure (When Using External Sub-Clock Input)

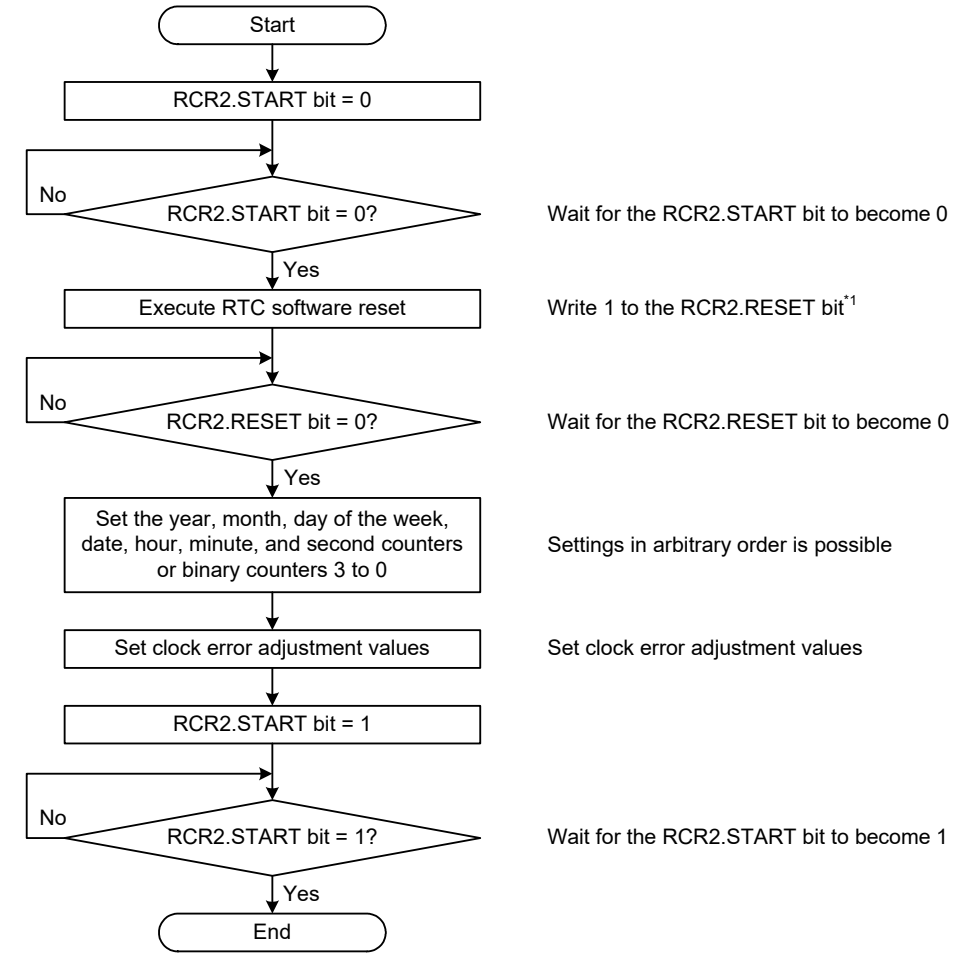
30.3.3 Setting the Time

Figure 30.6 and Figure 30.7 show how to set the time.



Note 1. This step is not required in the initial settings after power is supplied because an RTC software reset is applied in the clock setting procedure within the initial settings.

Figure 30.6 Setting the Time (When Using LOCO Clock)



Note 1. This step is not required in the initial settings after power is supplied because an RTC software reset is applied in the clock setting procedure within the initial settings.

Figure 30.7 Setting the Time (When Using Sub-Clock)

30.3.4 30-Second Adjustment

Figure 30.8 shows how to execute 30-second adjustment.

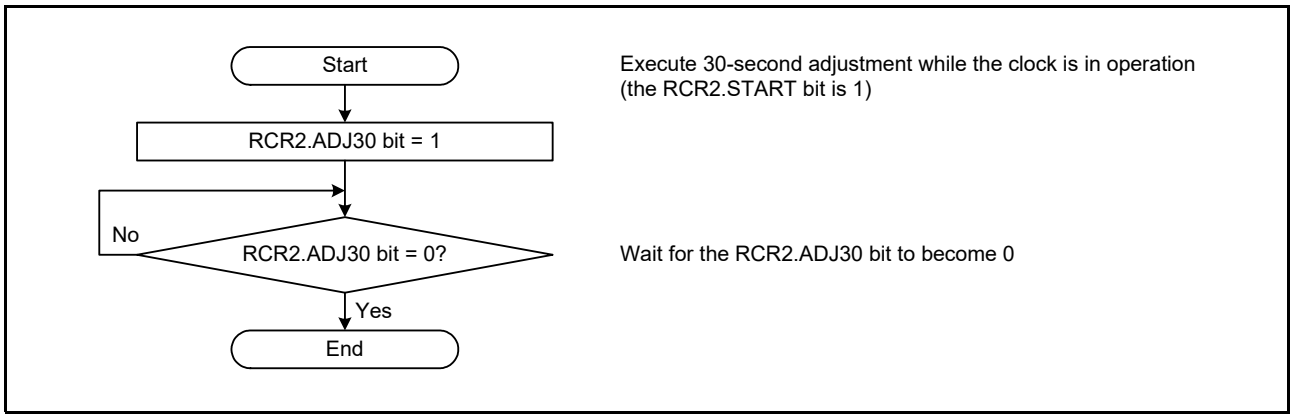


Figure 30.8 30-Second Adjustment

30.3.5 Reading 64-Hz Counter and Time

Figure 30.9 shows how to read the 64-Hz counter and time.
If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 30.9, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

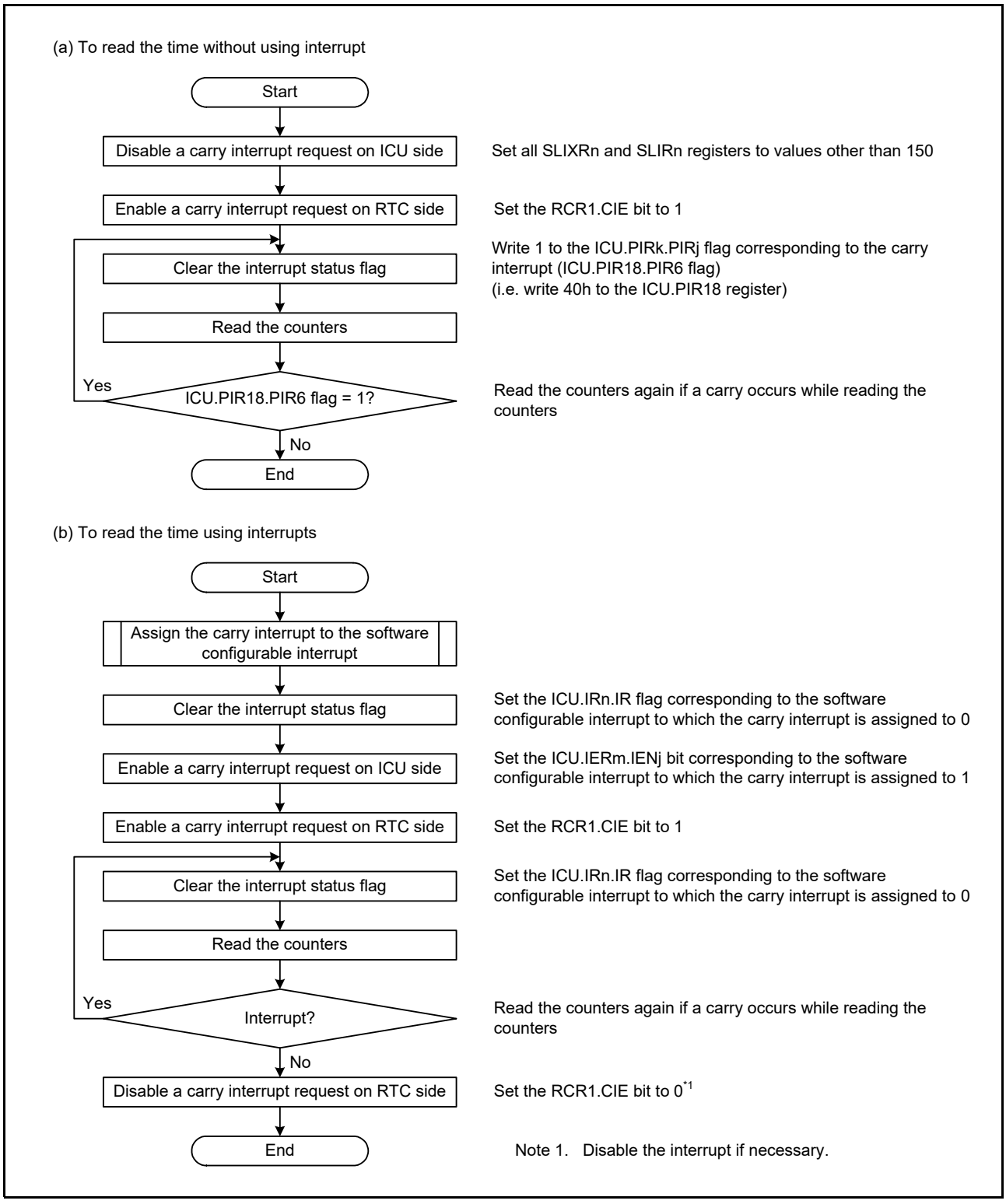


Figure 30.9 Reading Time

30.3.6 Alarm Function

Figure 30.10 shows how to use the alarm function.

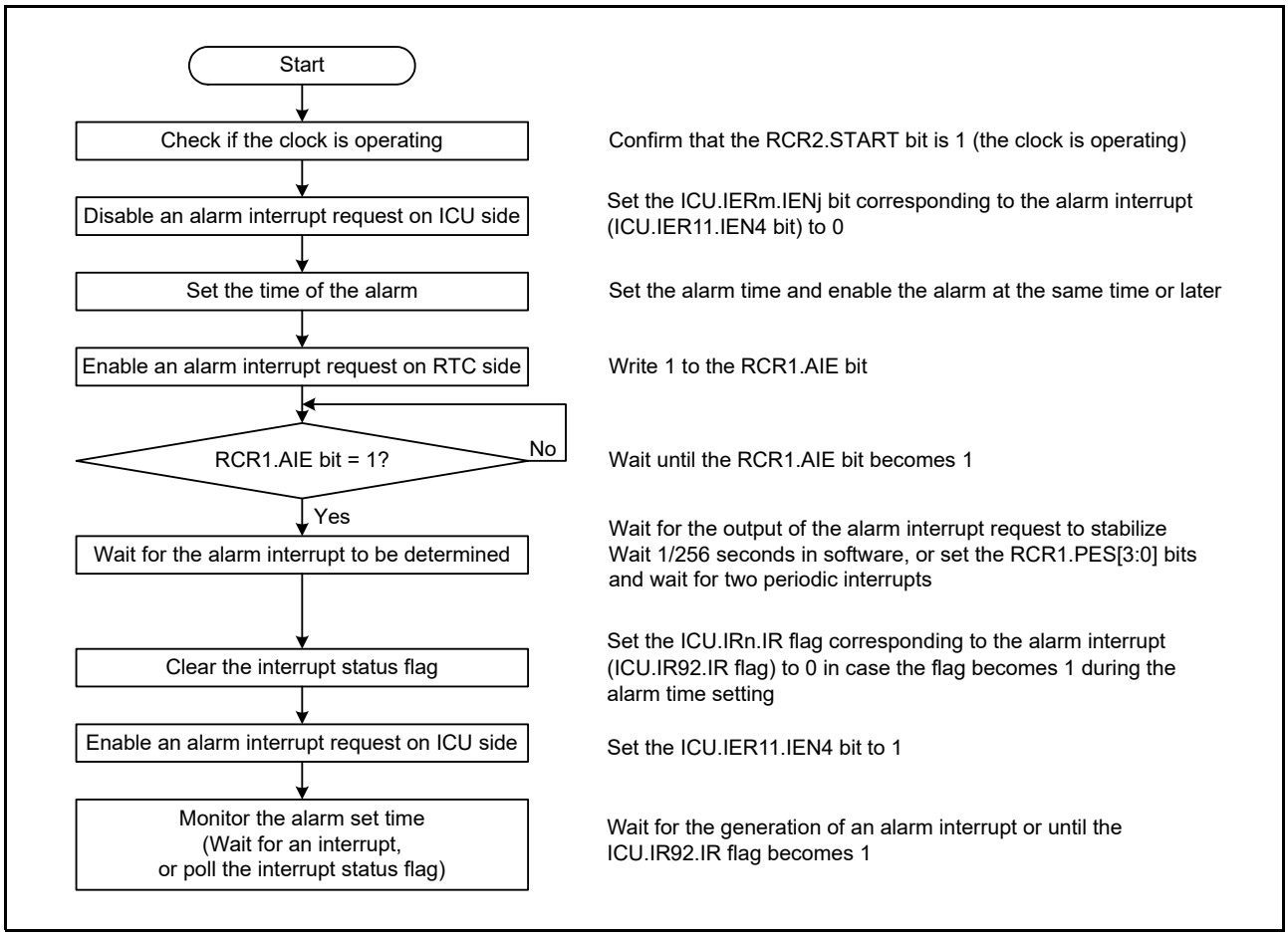


Figure 30.10 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the alarm interrupt (ICU.IR92.IR flag) is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the alarm interrupt (ICU.IER11.IEN4 bit), an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the ICU.IR92.IR flag to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state. In deep software standby mode, the MCU returns from the deep software standby mode even when the alarm interrupt request is disabled.

30.3.7 Procedure for Disabling Alarm Interrupt

Figure 30.11 shows the procedure for disabling the enabled alarm interrupt request.

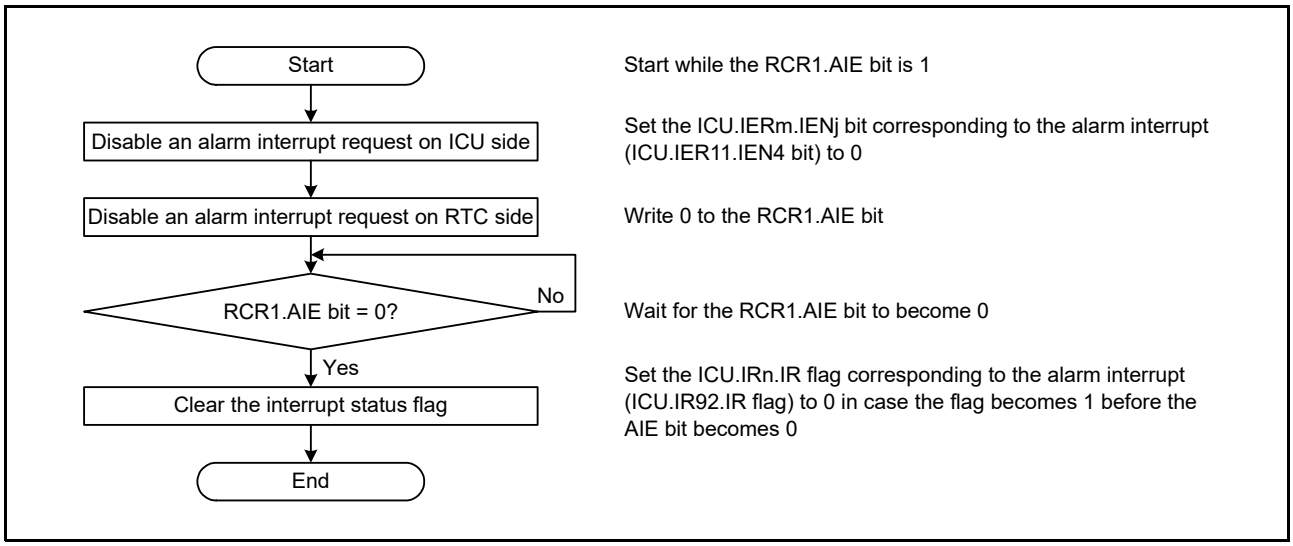


Figure 30.11 Procedure for Disabling Alarm Interrupt Request

30.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

30.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

30.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

30.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

30.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

30.3.9 Time Capture Function

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST flag is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in Figure 30.12 and operation when the noise filter is on is shown in Figure 30.13.

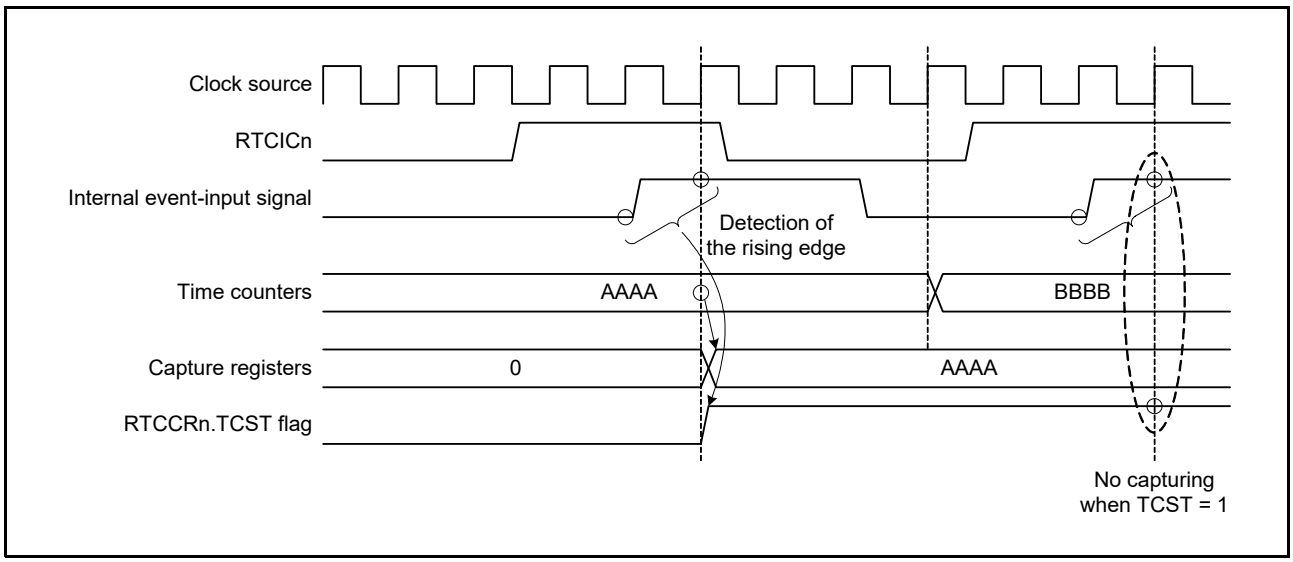


Figure 30.12 Timing of a Time Capture Operation (with the Filter Off) (n = 0 to 2)

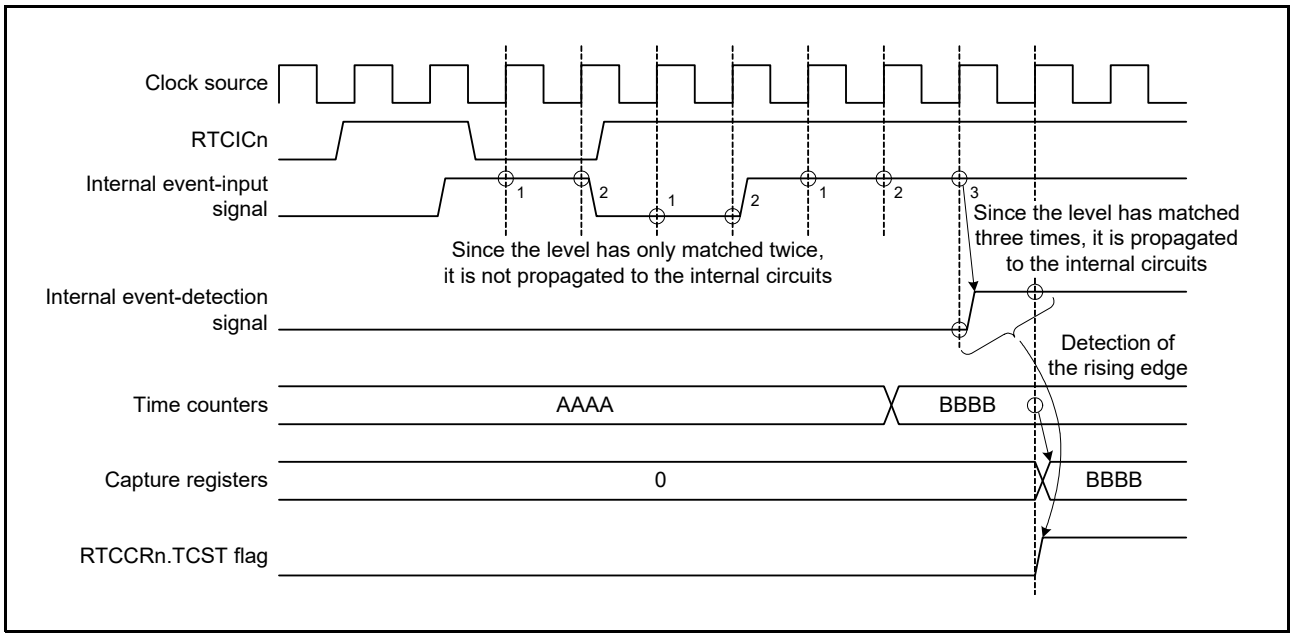


Figure 30.13 Timing of a Time Capture Operation (with the Filter On) (n = 0 to 2)

30.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 30.3 lists interrupt sources for the RTC.

Table 30.3 RTC Interrupt Sources

Name	Interrupt Sources	Interrupt Status Flag	Interrupt Request Enable Bit
ALM	Alarm interrupt	IR92.IR	IER11.IEN4
PRD	Periodic interrupt	IR93.IR	IER11.IEN5
CUP	Carry interrupt	PIR18.PIR6	— (software configurable interrupt)

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 30.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the ICU.IR92.IR flag to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

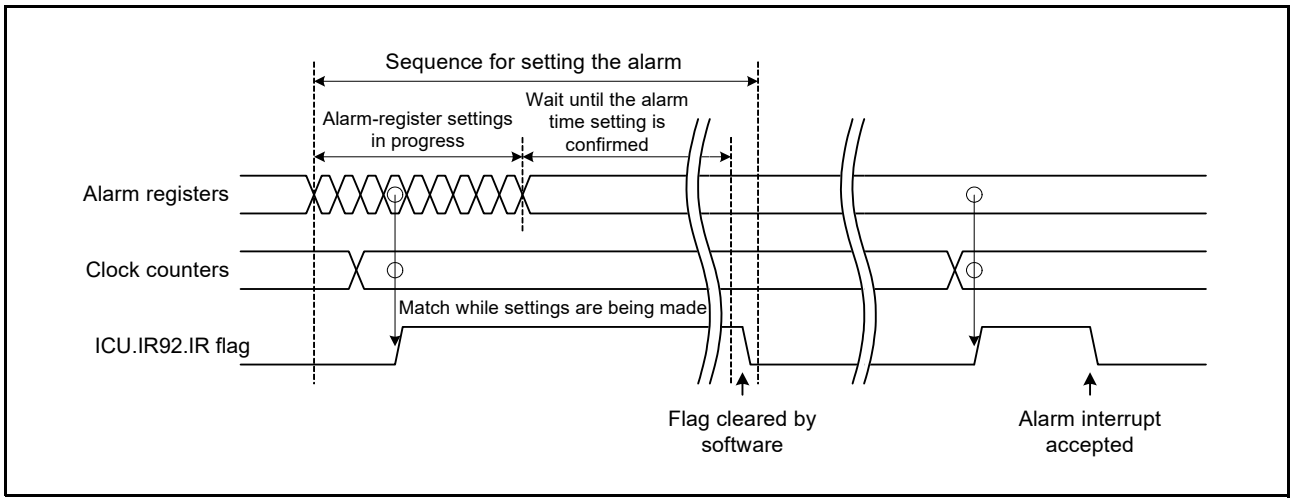


Figure 30.14 Timing Chart for the Alarm Interrupt

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

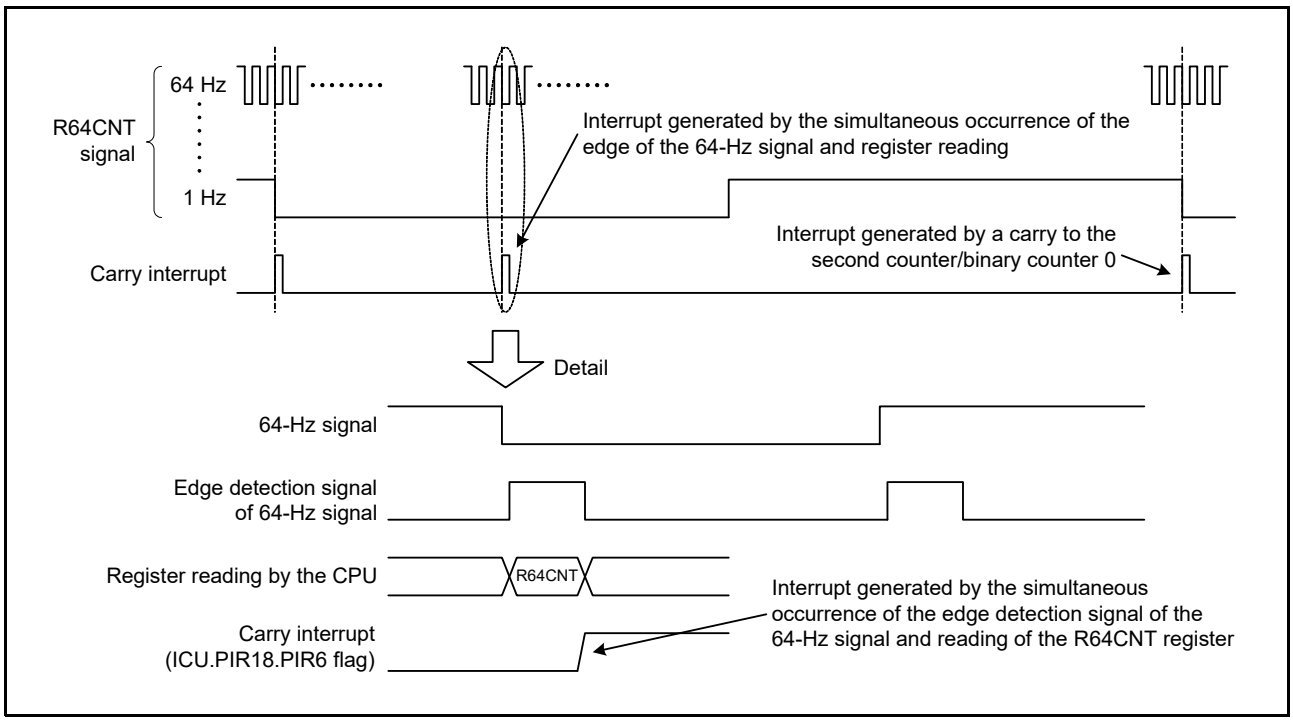


Figure 30.15 Carry Interrupt Timing Chart

30.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

30.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during software standby mode or deep software standby mode, the periodic event signals for the ELC are not output.

30.6 Usage Notes

30.6.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24, RFRH, RFRL

The counter must be stopped before writing to any of the above registers.

30.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 30.16.
The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

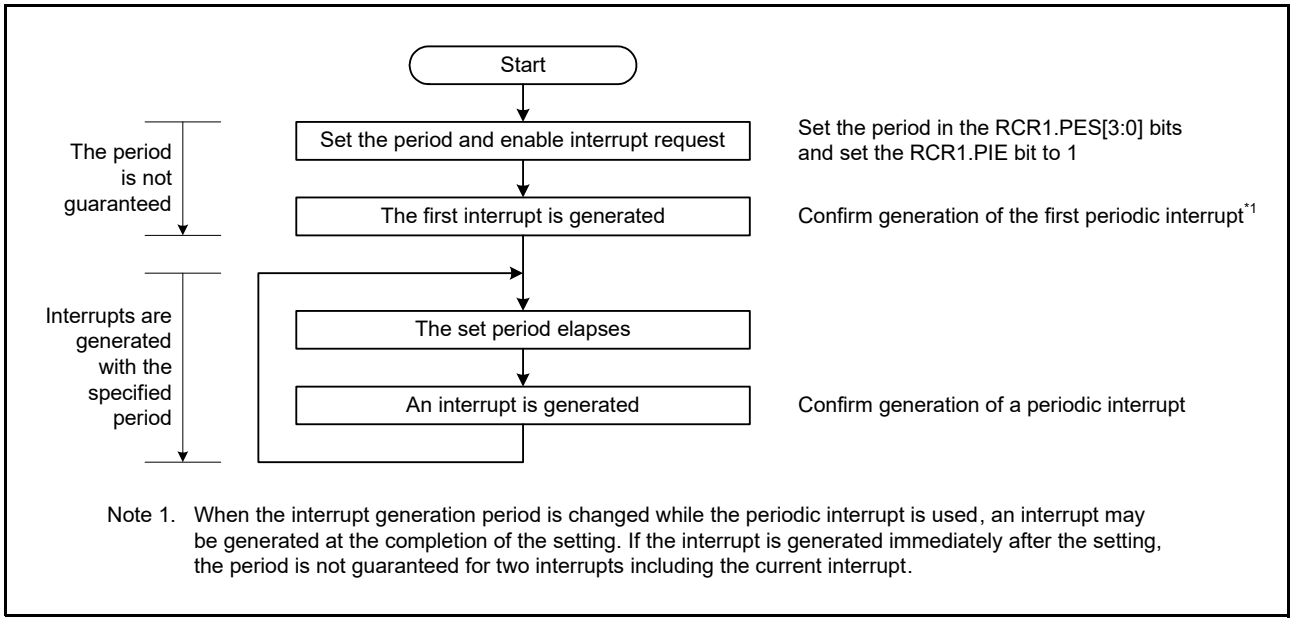


Figure 30.16 Using Periodic Interrupt Function

30.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

30.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, deep software standby mode, or battery backup) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

30.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter/binary counter 0 after having written to the counter register, follow the procedure in section 30.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4, RFRH, or RFRL register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, deep software standby mode, software standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the clock source have elapsed.

30.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 30.3.1, Outline of Initial Settings of Registers after Power On.

30.6.7 Notes on Using the Time Capture Function

When the time capture function is to be used, set both the RTCCRn.TCEN and TAMPICR1.CHnEN bits (n = 0 to 2) to 1. When tampering is not to be detected, leave the TAMPCR, TCECR, and TAMPICR2 registers in the tamper detector at their initial values.
When the time capture function is to be used to obtain the time stamps in response to the detection of tampering, use the noise filter in the tamper detector and set the RTCCRn.TCNF[1:0] bits to 00b (switch the noise filter of the RTC off).

30.6.8 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.
For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 30.17 or Figure 30.18.

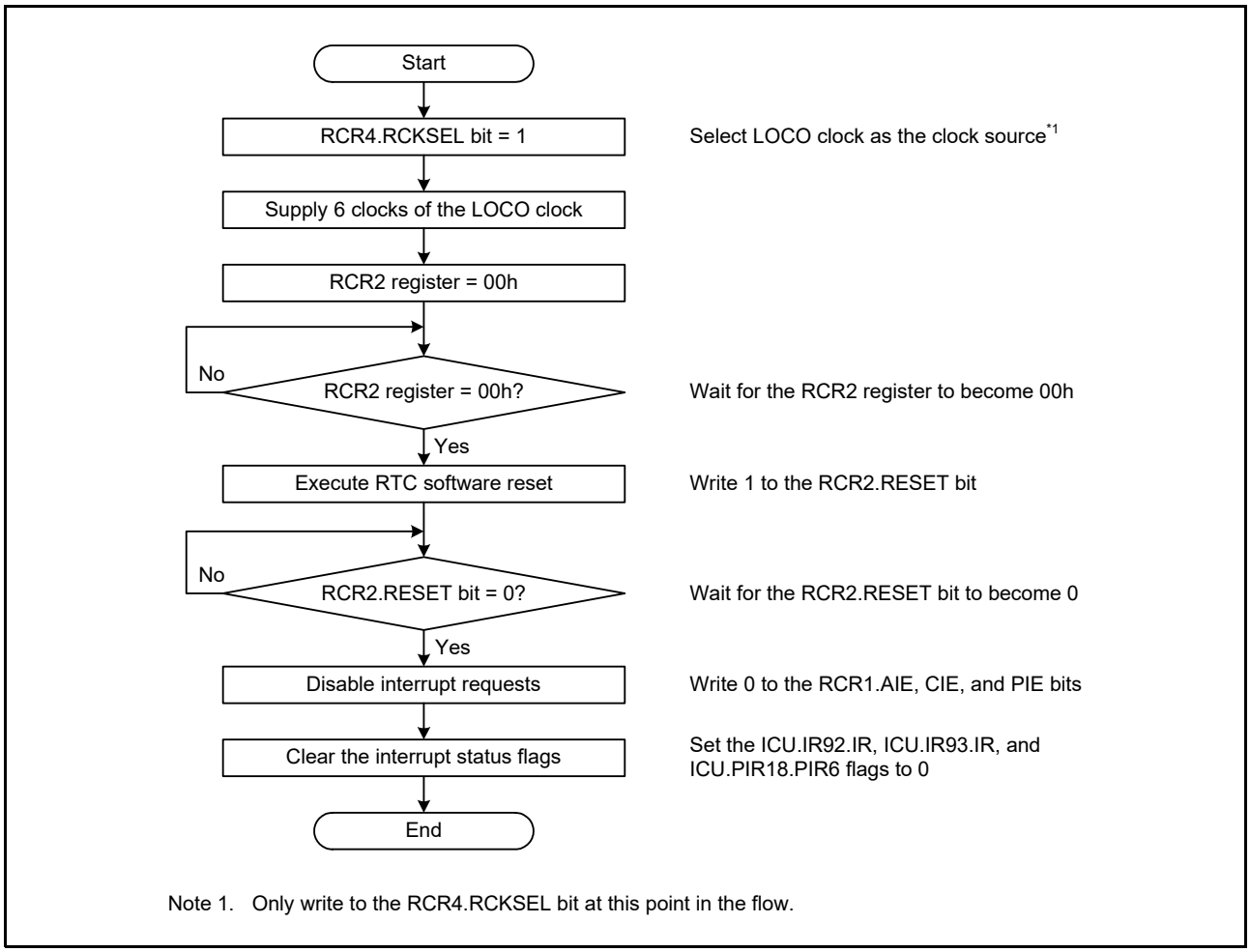
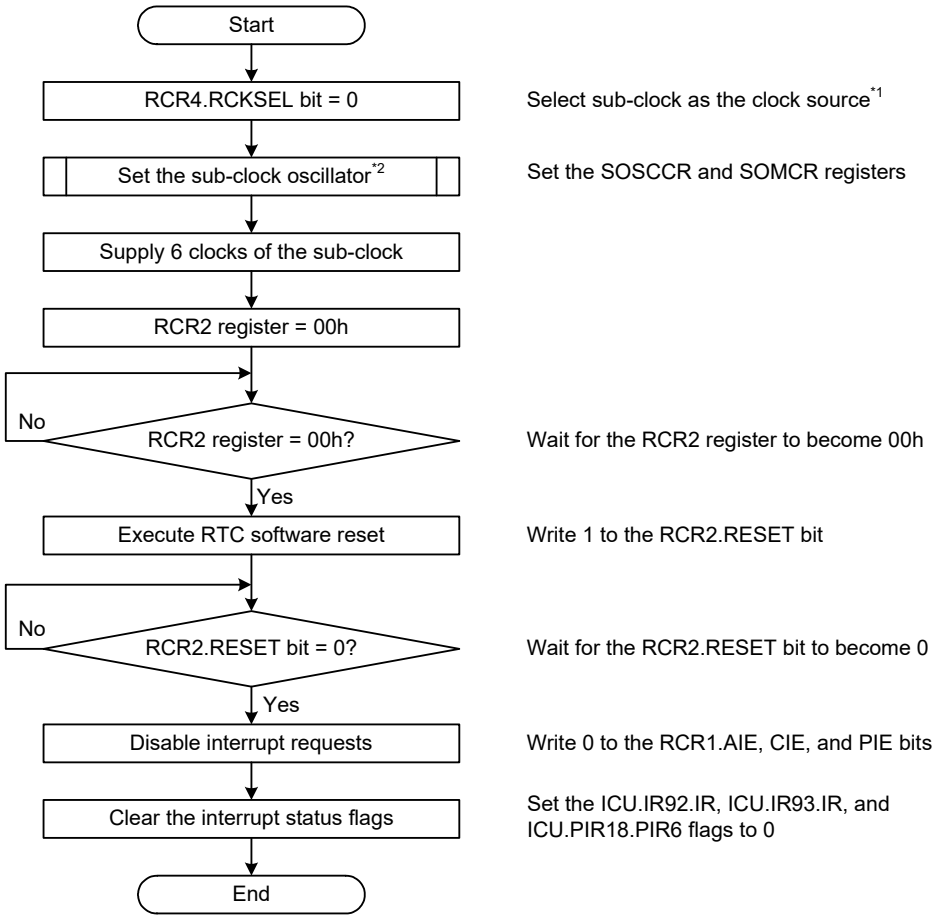


Figure 30.17 Initialization Procedure (When Using LOCO Clock)



Note 1. Only write to the RCR4.RCKSEL bit at this point in the flow.
Note 2. For the procedure for setting up a sub-clock oscillator, refer to section 9, Clock Generation Circuit.

Figure 30.18 Initialization Procedure (When Using Sub-Clock)

31. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.
In addition, a non-maskable or maskable interrupt can be generated on an underflow.
The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.

31.1 Overview

Table 31.1 lists the specifications of the WDT and Figure 31.1 shows a block diagram of the WDT.

Table 31.1 WDT Specifications

Item	Description
Count source	Peripheral module clock (PCLKB)
Clock divide ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the WDTRR register)
Conditions for stopping the counter	• Reset • In low power consumption modes (depends on the register setting*1) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/ maskable interrupt sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The counter value can be read by the WDTSR register.
Event link function (output)	• Counter underflow event output • Refresh error event output
ProtectZone filter	Security attribute can be set.*2

Note 1. When the OFS0.WDT0SLCSTP bit is 1 in auto-start mode, and when the WDTCSTPR.SLCSTP bit is 1 in register start mode.
Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

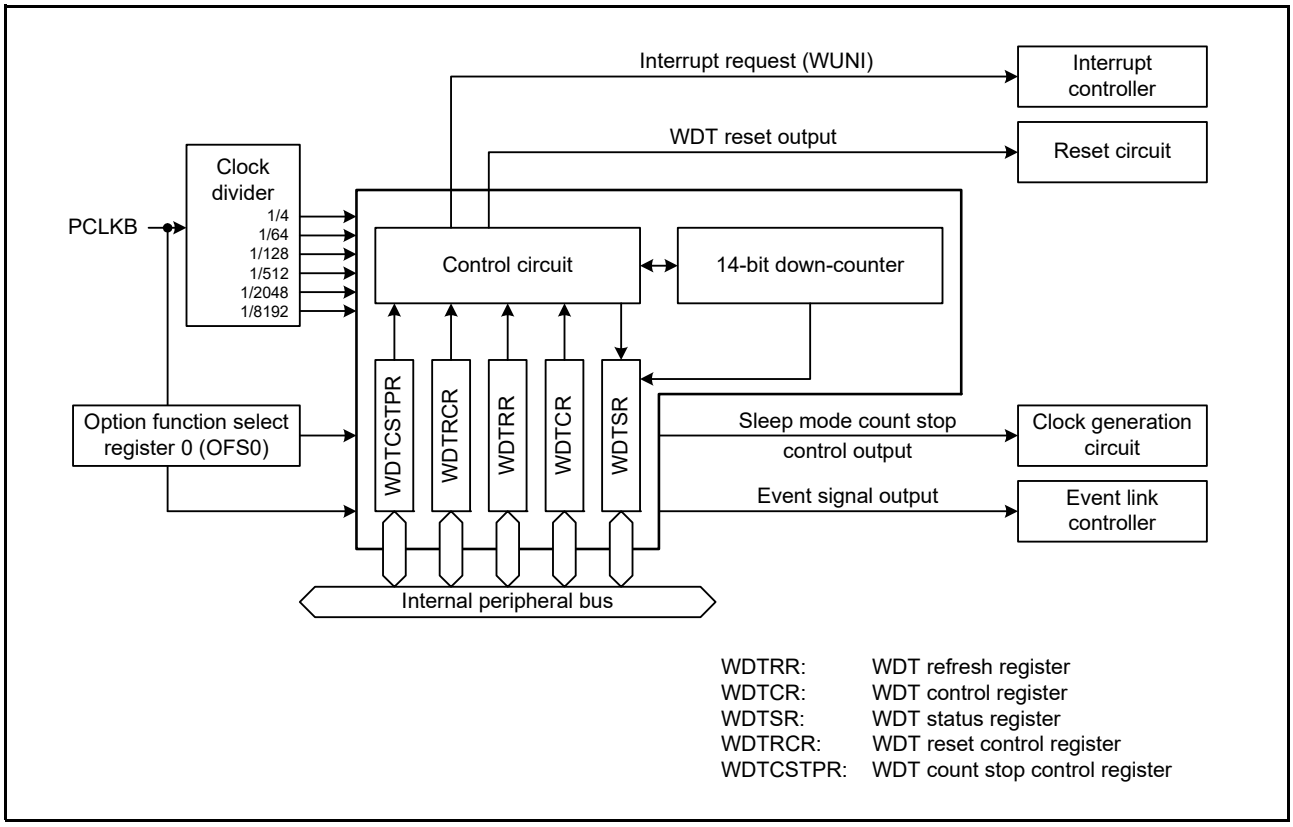
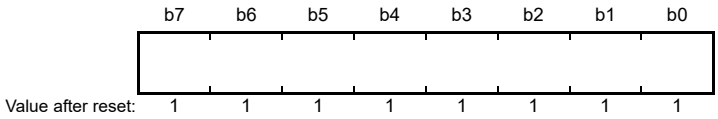


Figure 31.1 WDT Block Diagram

31.2 Register Descriptions

31.2.1 WDT Refresh Register (WDTRR)

Address(es): WDT0.WDTRR 8780 2600h, WDT0_NS.WDTRR 0780 2600h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register	R/W

The WDTRR register is used to refresh the counter of the WDT.
The counter of the WDT is refreshed by writing 00h and then writing FFh to the WDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by setting the WDT0TOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the WDTCR.TOPS[1:0] bits.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 31.3.2, Refresh Operation.

31.2.2 WDT Control Register (WDTCR)

Address(es): WDT0.WDTCR 8780 2602h, WDT0_NS.WDTCR 0780 2602h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Settings other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the time until a timeout occurs and the refresh-permitted period when using register start mode.

In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 31.3.6, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 1024, 4096, 8192, and 16384 cycles of the count clock.

After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLKB cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles are listed in Table 31.2.

Table 31.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of PCLKB
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits specify the frequency of the clock used for the counter (count clock). The count clock frequency can be selected from among the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLKB can be selected for the WDT.

RPES[1:0] Bits (Window End Position Select)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Select)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 31.3 lists the counter values for the window start and end positions and Figure 31.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 31.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note. If the window end setting is greater than or equal to the window start setting, the window end position is set to 0%.

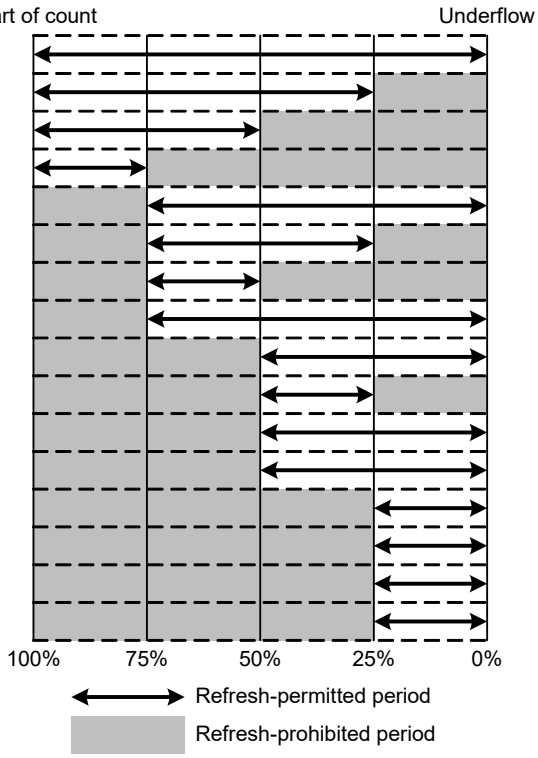


Figure 31.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

31.2.3 WDT Status Register (WDTSR)

Address(es): WDT0.WDTSR 8780 2604h, WDT0_NS.WDTSR 0780 2604h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This flag is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
To clear the flag, write 0 to the UNDFF flag. The flag will be cleared five PCLKB cycles after writing 0.
Note that after an underflow occurs, the flag cannot be cleared for a period of one count clock cycle and two PCLKB cycles. Wait for the period before clearing the flag.

REFEF Flag (Refresh Error Flag)

This flag is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
To clear the flag, write 0 to the REFEF flag. The flag will be cleared five PCLKB cycles after writing 0.
Note that after a refresh error occurs, the flag cannot be cleared for a period of one count clock cycle and two PCLKB cycles. Wait for the period before clearing the flag.

31.2.4 WDT Reset Control Register (WDTRCR)

Address(es): WDT0.WDTRCR 8780 2606h, WDT0_NS.WDTRCR 0780 2606h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or maskable interrupt request is output. 1: Reset signal is output.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the behavior when a timeout or refresh error occurs when using register start mode. In auto-start mode, the WDTRCR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTRCR register can also be made in the OFS0 register. For details, refer to section 31.3.6, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

31.2.5 WDT Count Stop Control Register (WDTCSTPR)

Address(es): WDT0.WDTCSTPR 8780 2608h, WDT0_NS.WDTCSTPR 0780 2608h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Counter continues counting in sleep mode or deep sleep mode. 1: Counter stops in low power consumption mode.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify whether the counter operates in sleep mode or deep sleep mode when using register start mode.

In auto-start mode, the WDTCSTPR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the WDTCSTPR register can also be made in the OFS0 register. For details, refer to section 31.3.6, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether the counter continues or stops counting in sleep mode or deep sleep mode.

31.2.6 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 31.3.6, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

31.3 Operation

31.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after a reset is released, and register start mode, in which counting is started by refresh operation (writing to the register).
In auto-start mode, counting automatically starts after a reset is released in accordance with the settings in option function select register 0 (OFS0).
In register start mode, counting is started by refresh operation (writing to the register) after the respective registers are set after a reset is released.
Select auto-start mode or register start mode by setting the OFS0.WDT0STRT bit.
When the auto-start mode is selected, the settings in the WDTCR, WDTRCR, and WDTCSTPR registers are disabled, and the settings in the OFS0 register are enabled.
On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDTCR, WDTRCR, and WDTCSTPR registers are enabled.

31.3.1.1 Register Start Mode

When the OFS0.WDT0STRT bit is 1, register start mode is selected, and the WDTCR, WDTRCR, and WDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the WDTCR register, the reset output or interrupt request output in the WDTRCR register, and the counter operation/stop in sleep mode or deep sleep mode in the WDTCSTPR register. Then, refresh the counter to start counting down from the value set by the WDTCR.TOPS[1:0] bits.
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or an interrupt request (WUNI). Reset signal output or interrupt request output can be selected by setting the WDTRCR.RSTIRQS bit.

Figure 31.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDT0STRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

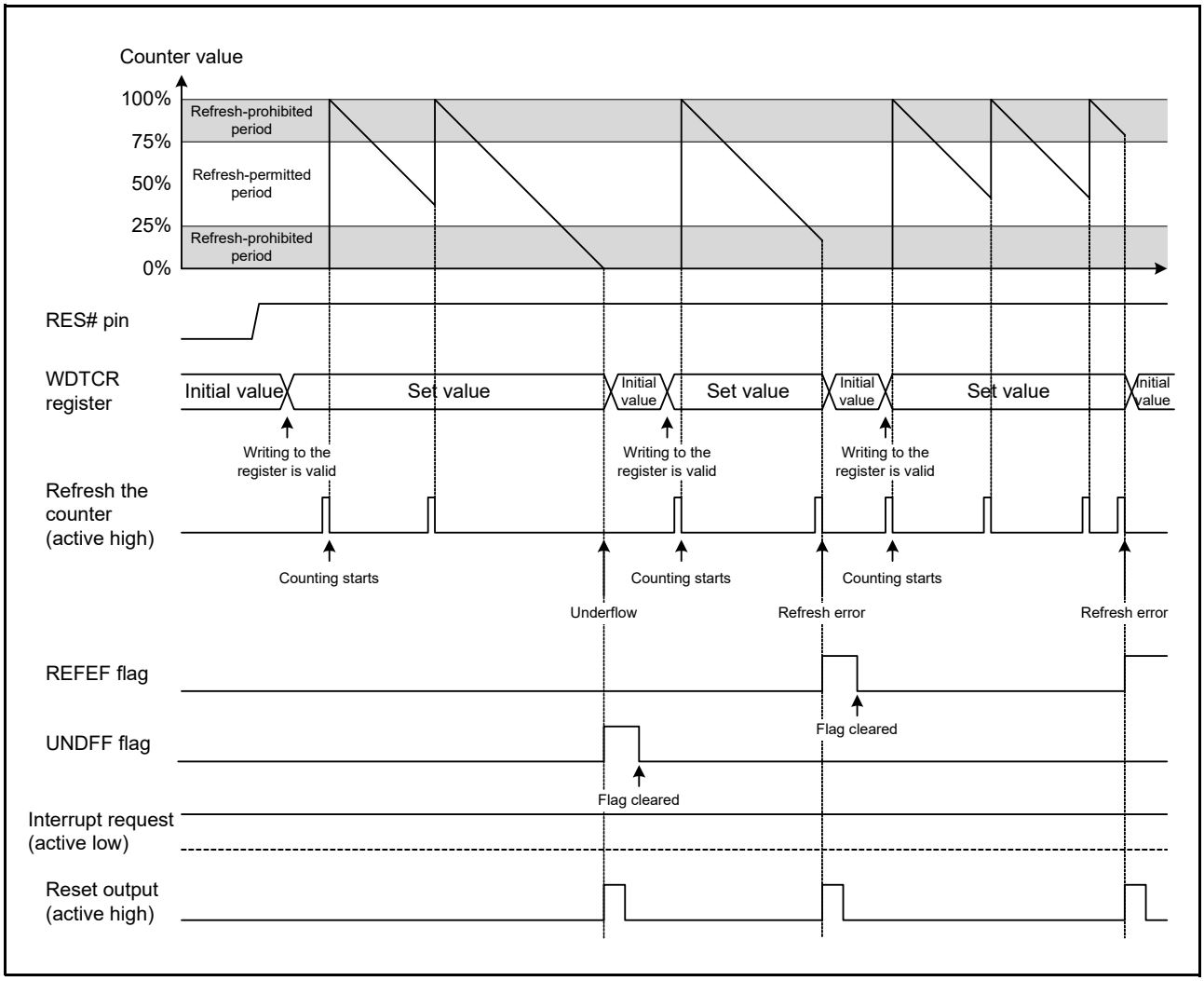


Figure 31.3 Operation Example in Register Start Mode

31.3.1.2 Auto-Start Mode

When the WDT0STRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDTCR, WDTRCR, and WDTCSTPR registers are disabled, and the settings in the OFS0 register are enabled.
During the reset state, the setting values (clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter operation/stop in sleep mode or deep sleep mode) of the OFS0 register are set in the WDT registers. When the reset is released, the counter automatically starts counting down from the value set by the OFS0.WDT0TOPS[1:0] bits.
After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or interrupt request (WUNI).
After the reset signal or interrupt request is output of for one cycle of counting, the value of the timeout period is set in the counter counting is restarted.
Reset signal output or interrupt request output can be selected by setting the OFS0.WDT0RSTIRQS bit.

Figure 31.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.WDT0STRT = 0)
- Interrupt request output is enabled (OFS0.WDT0RSTIRQS = 0)
- The window start position is 75% (OFS0.WDT0RPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDT0RPES[1:0] = 10b)

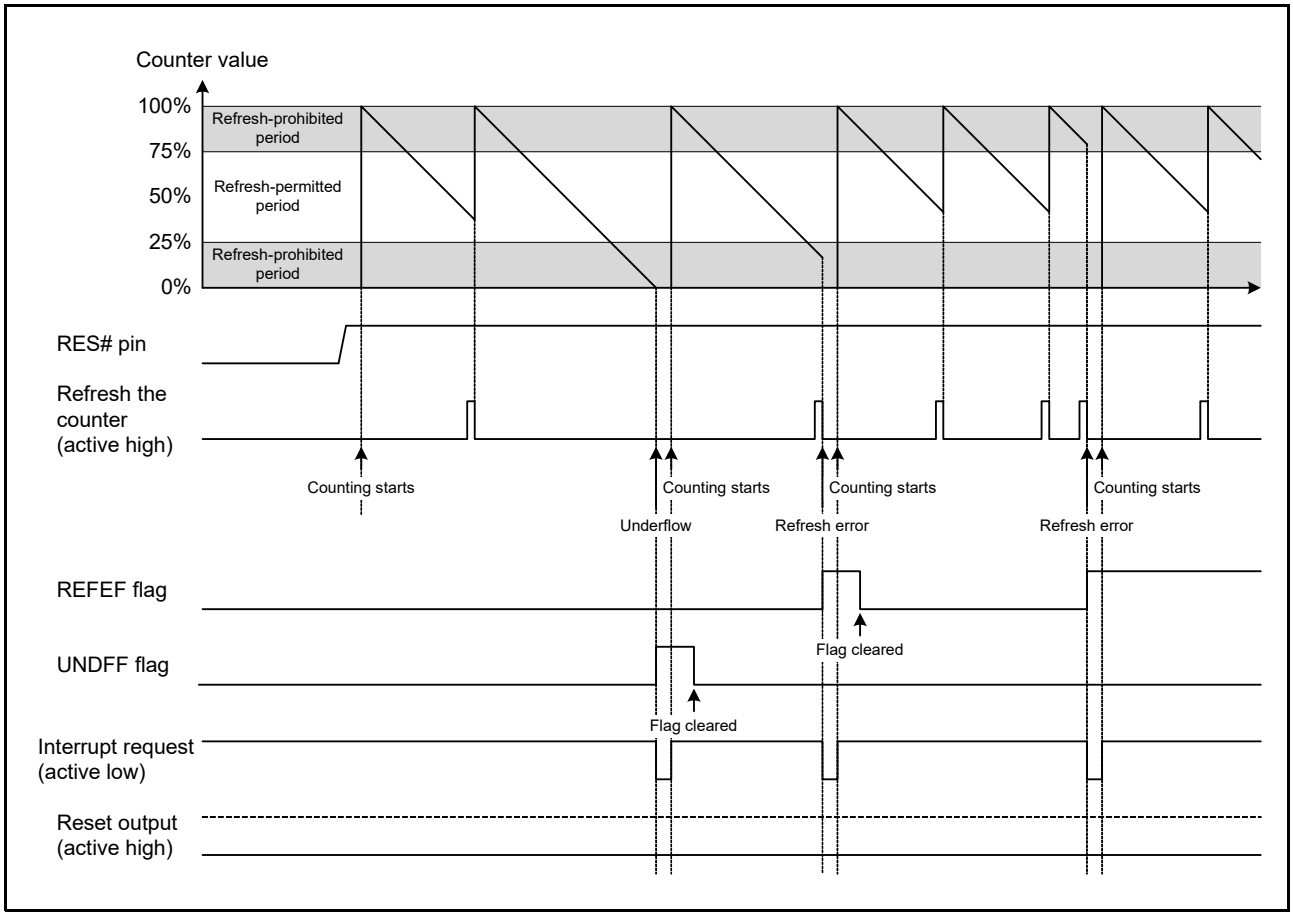


Figure 31.4 Operation Example in Auto-Start Mode

31.3.2 Refresh Operation

The counter is refreshed by writing the values 00h and then FFh to the WDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.
Even if a register other than the WDTRR register is accessed or the WDTRR register is read between writing 00h and writing FFh to the WDTRR register, correct refreshing will be done.
Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the WDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh

After FFh is written to the WDTRR register, refreshing the counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR register should be completed four-count cycles before the counter underflows.

Figure 31.5 shows the WDT refresh operation waveforms when the count clock is PCLKB/64.

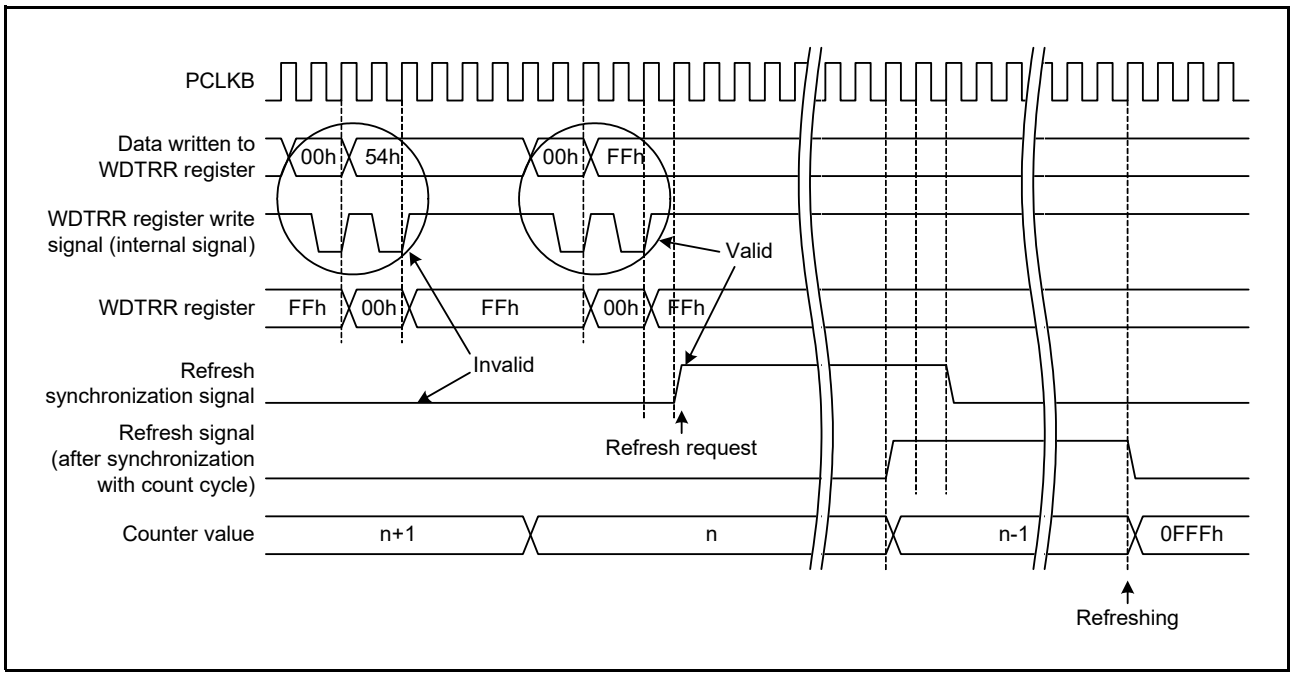


Figure 31.5 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

31.3.3 Reset Output

When the WDTRCR.RSTIRQS bit is set to 1 in register start mode or when the WDT0RSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, all registers are initialized after output of the reset signal. To start down-counting, set the registers again after the reset is released and then perform a refresh operation.
In auto-start mode, counting down automatically starts after the reset output.

31.3.4 Interrupt Source

When the WDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.WDT0RSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or a maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUM).

Table 31.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Counter underflow Refresh error	Not possible	Not possible

31.3.5 Reading the Counter Value

The WDT stores the counter value in the WDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.
Figure 31.6 shows the processing for reading the WDT counter value when the count clock is PCLKB/64.

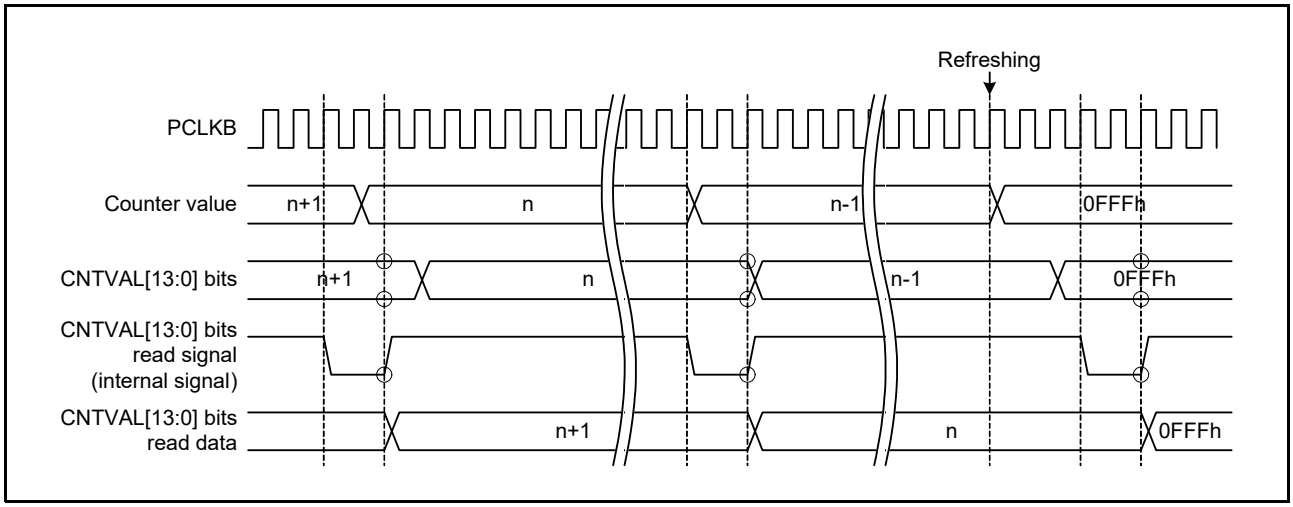


Figure 31.6 Processing for Reading WDT Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

31.3.6 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 31.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
For details on the OFS0 register, refer to section 7.2.2, Option Function Select Register 0 (OFS0).

Table 31.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDT0STRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDT0STRT = 1
Counter	Timeout period selection	OFS0.WDT0TOPS[1:0]	WDTCR.TOPS[1:0]
	Clock divide ratio selection	OFS0.WDT0CKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDT0RPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDT0RPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDT0RSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.WDT0SLCSTP	WDTCSTPR.SLCSTP

31.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the WDT as the event signal. Therefore, the ELC generates an event to the module specified previously when the WDT outputs an interrupt request. The event signal is output by the counter underflow and refresh error.
An event signal is output regardless of the setting of the WDTRCR.RSTIRQS bit in register start mode or the OFS0.WDT0RSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the WDTSR.REFEF or WDTSR.UNDFF flag is 1.
For details, see section 21, Event Link Controller (ELC).

32. Independent Watchdog Timer (IWDTa)

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT clock (IWDTCLK) is used as the count clock (not affected by the peripheral module clock).
- The IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select not to stop the counter not only in sleep mode and deep sleep mode but also in software standby mode and deep software standby mode 1.

32.1 Overview

Table 32.1 lists the specifications of the IWDT.

Table 32.1 IWDT Specifications

Item	Description
Count source*1	IWDT clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset • In low power consumption modes (depends on the register setting*2) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/ maskable interrupt sources	• Counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The counter value can be read by the IWDTSR register.
Event link function (output)	• Counter underflow event output • Refresh error event output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the counter operation/stop in low power consumption mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the counter operation/stop in low power consumption mode (IWDTCSTPR.SLCSTP bit)
ProtectZone filter	Security attribute can be set.*3

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 32 × (the frequency of the count clock).
Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSTPR.SLCSTP bit is 1 in register start mode.
Note 3. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 32.1 is a block diagram of the IWDT.
In IWDT, the bus interface and registers operate with the peripheral module clock (PCLKB), while the 14-bit down-counter and control circuits operate with IWDTCLK.

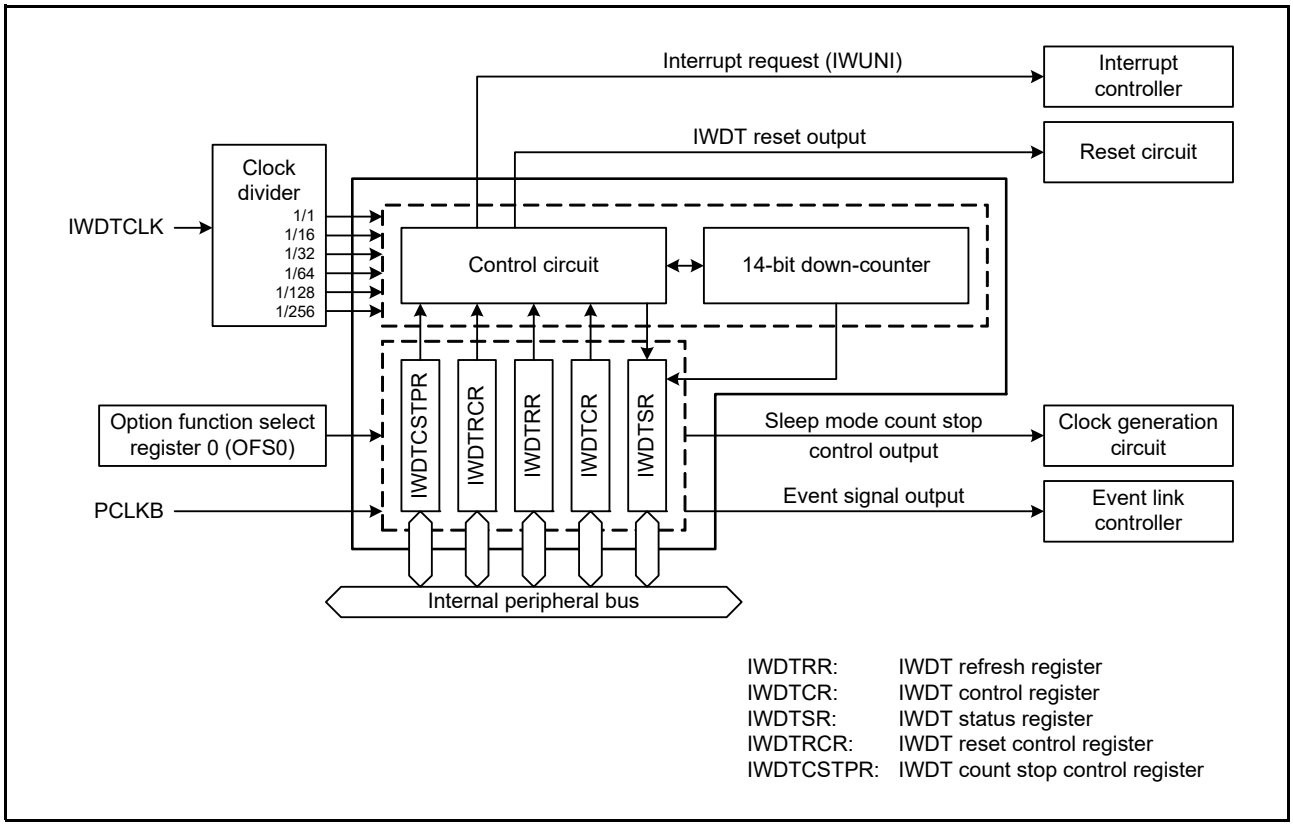
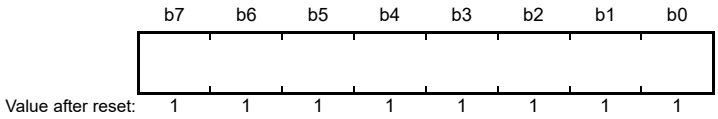


Figure 32.1 IWDT Block Diagram

32.2 Register Descriptions

32.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 8780 2200h, IWDT_NS.IWDTRR 0780 2200h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register is used to refresh the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 32.3.2, Refresh Operation.

32.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 8780 2202h, IWDT_NS.IWDTCR 0780 2202h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the time until a timeout occurs and the refresh-permitted period when using register start mode.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles of the count clock.

After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 32.2.

Table 32.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits specify the frequency of the clock used for the counter (count clock). The count clock frequency can be selected from among the IWDTCLK divided by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 32.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 32.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

RPSS[1:0] Bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 32.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note. If the window end setting is greater than or equal to the window start setting, the window end position is set to 0%.

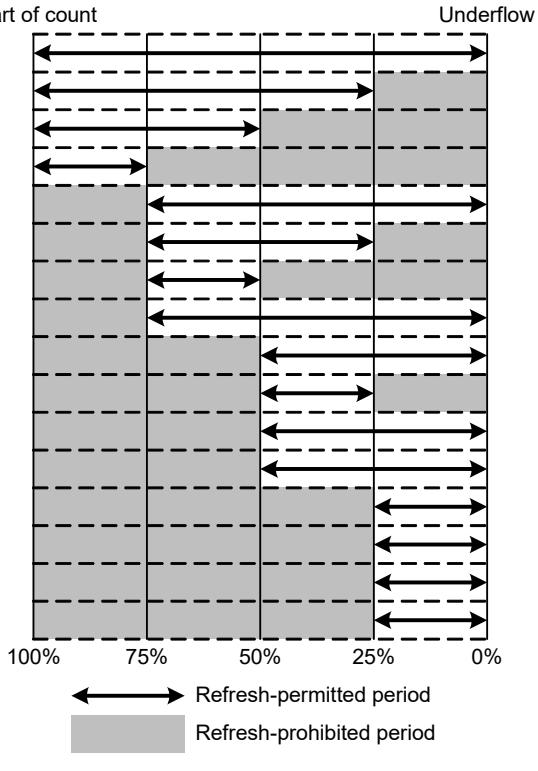


Figure 32.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

32.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 8780 2204h, IWDT_NS.IWDTSR 0780 2204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This flag is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
To clear the flag, write 0 to the UNDFF flag. The flag will be cleared three IWDTCLK cycles and 32 PCLKB cycles after writing 0.
Note that after an underflow occurs, the flag cannot be cleared for a period of one count clock cycle and two IWDTCLK cycles. Wait for the period before clearing the flag.

REFEF Flag (Refresh Error Flag)

This flag is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
To clear the flag, write 0 to the REFEF flag. The flag will be cleared three IWDTCLK cycles and 32 PCLKB cycles after writing 0.
Note that after a refresh error occurs, the flag cannot be cleared for a period of one count clock cycle and two IWDTCLK cycles. Wait for the period before clearing the flag.

32.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 8780 2206h, IWDT_NS.IWDTRCR 0780 2206h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or maskable interrupt request is output. 1: Reset signal is output.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.

This register is used to specify the behavior when a timeout or refresh error occurs when using register start mode. In auto-start mode, the IWDTRCR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

32.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 8780 2208h, IWDT_NS.IWDTCSTPR 0780 2208h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Counter continues counting even in low power consumption mode.*1 1: Counter stops in low power consumption mode.	R/W

Note: This register can be written only once after a reset, and cannot be written after the counter starts counting.
Note 1. The counter continues counting in sleep mode, deep sleep mode, software standby mode, or deep software standby mode 1. The MCU does not enter deep software standby mode 2 or 3.

This register is used to specify whether the counter operates in low power consumption mode when using register start mode.
In auto-start mode, the IWDTCSTPR register setting is disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether the counter continues or stops counting in low power consumption mode.

32.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

32.3 Operation

32.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDTSTRT bit in option function select register 0 (OFS0).
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDTRR register. When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

32.3.1.1 Register Start Mode

When the OFS0.IWDTSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and counter operation/stop in low power consumption mode in the IWDTCSTPR register. Then, refresh the counter to start counting down from the value selected by setting the IWDTCR.TOPS[1:0] bits.
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or an interrupt request (IWUNI). Set the IWDTRCR.RSTIRQS bit to select either reset signal output or interrupt request output.

Figure 32.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

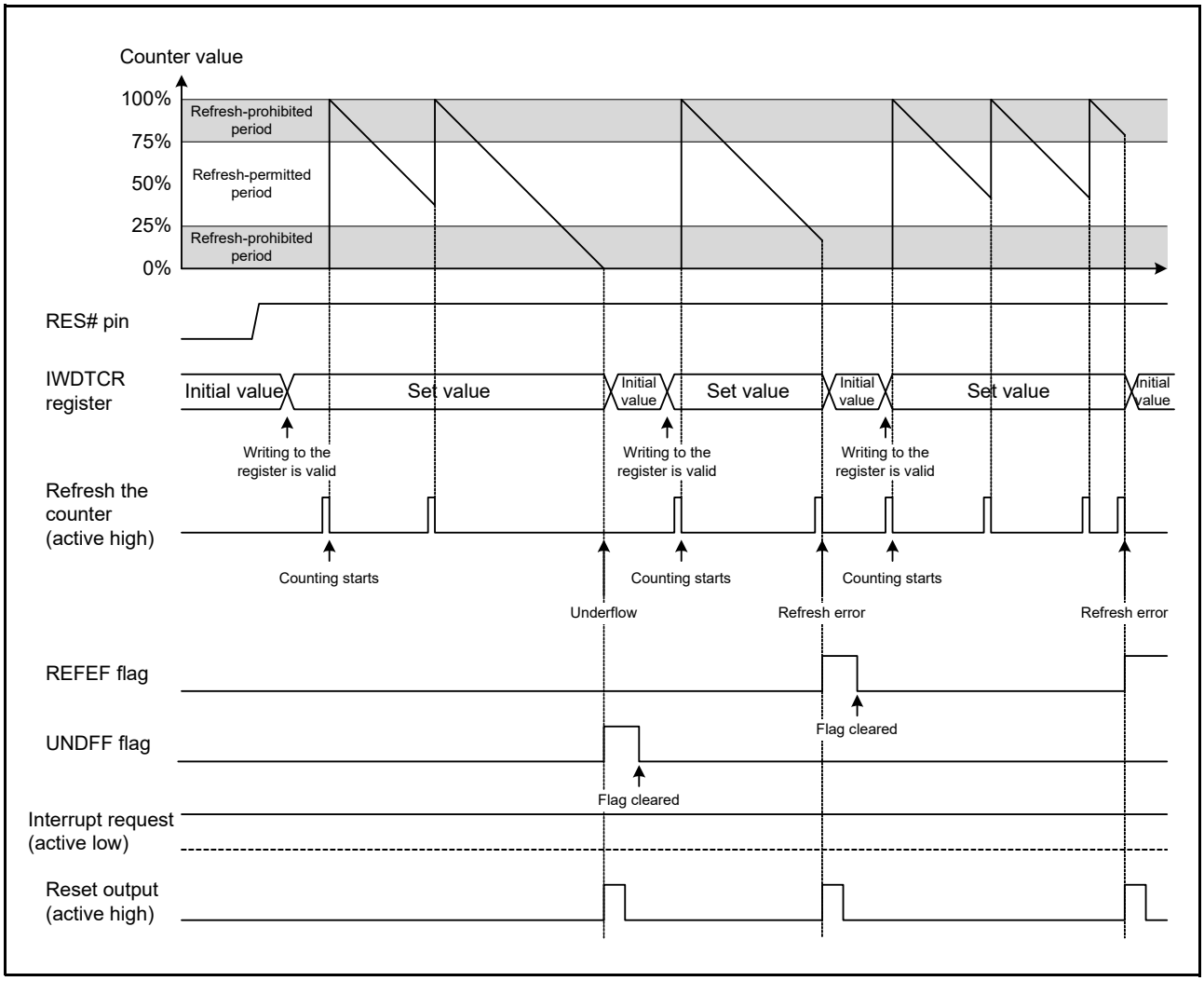


Figure 32.3 Operation Example in Register Start Mode

32.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.

During the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter operation/stop in low power consumption mode are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the OFS0.IWDTTOPS[1:0] bits.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or an interrupt request (IWUNI). After the reset signal or interrupt request (IWUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset signal output or interrupt request output.

Figure 32.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

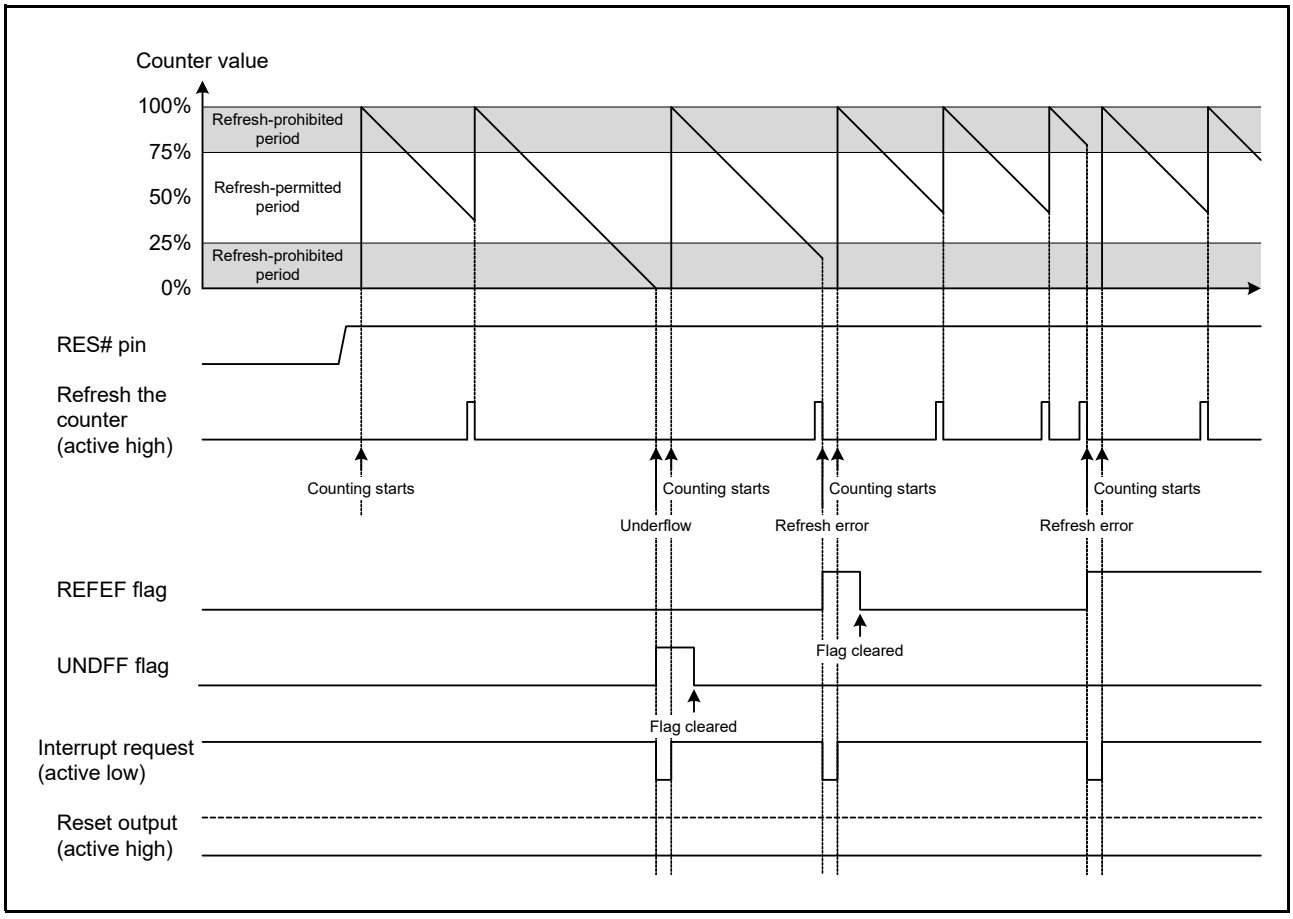


Figure 32.4 Operation Example in Auto-Start Mode

32.3.2 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDTCLK make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 01FFh, even if 00h is written to the IWDTRR register before 01FFh is reached (0202h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 01FFh.
- When the window end position is set to 01FFh, refreshing is done if 0203h (four-count cycles before 01FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 32.5 shows the IWDT refresh operation waveforms when PCLKB > IWDTCLK and the count clock is IWDTCLK/1.

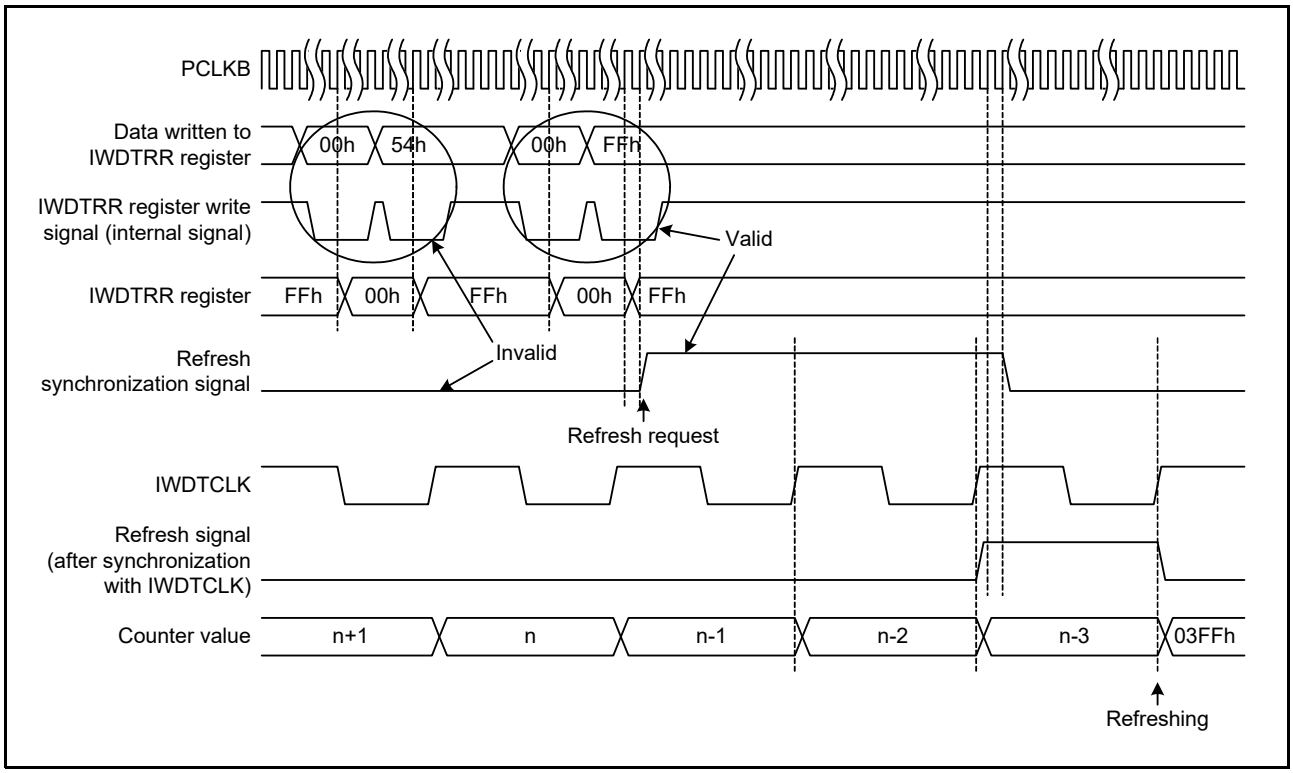


Figure 32.5 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 10b)

32.3.3 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDFF flags retain the source of the interrupt request from the IWDT.
Thus, after interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next interrupt request from the IWDT, the earlier interrupt source is cleared and the new interrupt source is written.

32.3.4 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, all registers are initialized after output of the reset signal. To start down-counting, set the registers again after the reset is released and then perform a refresh operation.
In auto-start mode, counting down automatically starts after the reset output.

32.3.5 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (IWUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or a maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUM).

Table 32.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

32.3.6 Reading the Counter Value

As the counter in IWDTCLK, the counter value cannot be read directly. The IWDT synchronizes the counter value with the PCLKB and stores it in the IWDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLKB clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 32.6 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the count clock is IWDTCLK/1.

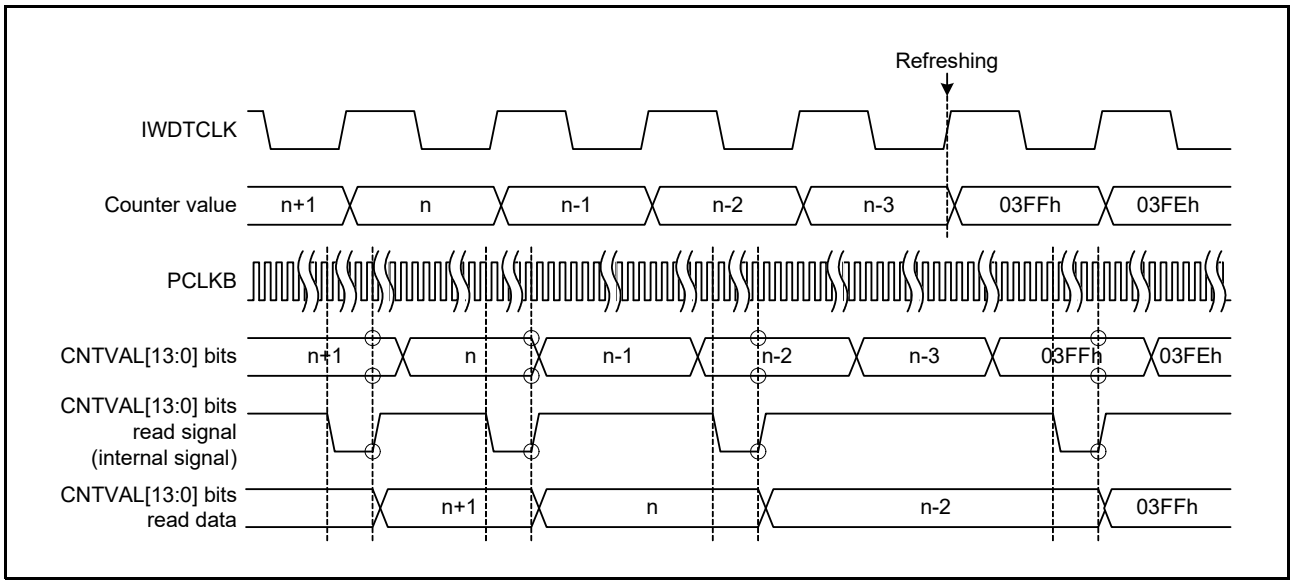


Figure 32.6 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 10b)

32.3.7 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 32.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
For details on the OFS0 register, refer to section 7.2.2, Option Function Select Register 0 (OFS0).

Table 32.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

32.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the IWDT as the event signal. Therefore, the ELC generates an event to the module specified previously when the IWDT outputs an interrupt request. The event signal is output by the counter underflow and refresh error.
An event signal is output regardless of the setting of the IWDTRCR.RSTIRQS bit in register start mode or the OFS0.IWDTRSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the IWDTSR.REFEF or IWDTSR.UNDFF flag is 1.
For details, see section 21, Event Link Controller (ELC).

32.5 Usage Notes

32.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLKB and IWDTCLK and set values which ensure that refreshing is possible.

32.5.2 Count Clock Frequency Setting

Satisfy the frequency of the PCLKB ≥ 32 × (the frequency of the count clock).

32.5.3 Reading Counter after Returning from Deep Software Standby Mode 1

After returning from deep software standby mode 1, the value of the IWDTSR.CNTVAL[13:0] bits will be 0000h until the first count clock is supplied. To obtain the down counter value, use one of the following methods:

- Continue reading the CNTVAL[13:0] bits until a value other than 0000h is read.
- Wait until the count clock is supplied, then read the CNTVAL[13:0] bits.
- If the entire period is set as the refresh-permitted period, refresh the counter and then read the CNTVAL[13:0] bits.

32.5.4 Transitions to Low Power Consumption Mode

If the timer-refresh procedure is started immediately before entry to low power consumption mode, complete the procedure, confirm that FFh has been written to the IWDTRR register, and then execute a WAIT instruction.

33. Ethernet Module

33.1 Overview

The Ethernet module (ESWM) system consists of the following six modules.

- Ethernet message forwarding engine (MFWD)
- Ethernet common agent (COMA)
- Ethernet agent (ETHA)
- Ethernet MAC (RMAC)
- Ethernet CPU agent (GWCA)
- Ethernet generalized PTP timer (GPTP)

This MCU has one Ethernet port compliant with the IEEE 802.3 Media Access Control (MAC) layer protocol.
Table 33.1 lists the specifications of ESWM. Figure 33.1 shows the block diagram of the ESWM system. Table 33.2 lists the I/O pins of ESWM.

Table 33.1 ESWM Specifications

Item		Description
MAC	Number or ports	1 port
	Supported standard	IEEE 802.3
	Supported PHY interfaces	MII, RMII, GMII, RGMII
	Data rate	10 Mbps, 100 Mbps, 1000 Mbps
	Mode	Full duplex Half duplex (10 Mbps or 100 Mbps only)
	Jumbo frame support	Up to 64 Kbytes
	Management interface	MDIO
	MAC-to-MAC connection	Full-duplex communication at 10 Mbps or 100 Mbps is possible by connecting MII to MII.
	Power reduction	Supports Energy Efficient Ethernet (IEEE 802.3az)
	Wake-on-LAN	Magic Packet™ detection, WOL signal output
Generalized PTP timer	Number of channels	1
	Media clock capture input	2
	Media clock recovery output	2
	PTP pulse generator	two output timers
Protect Zone filter	Security attribute can be set.*1	

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

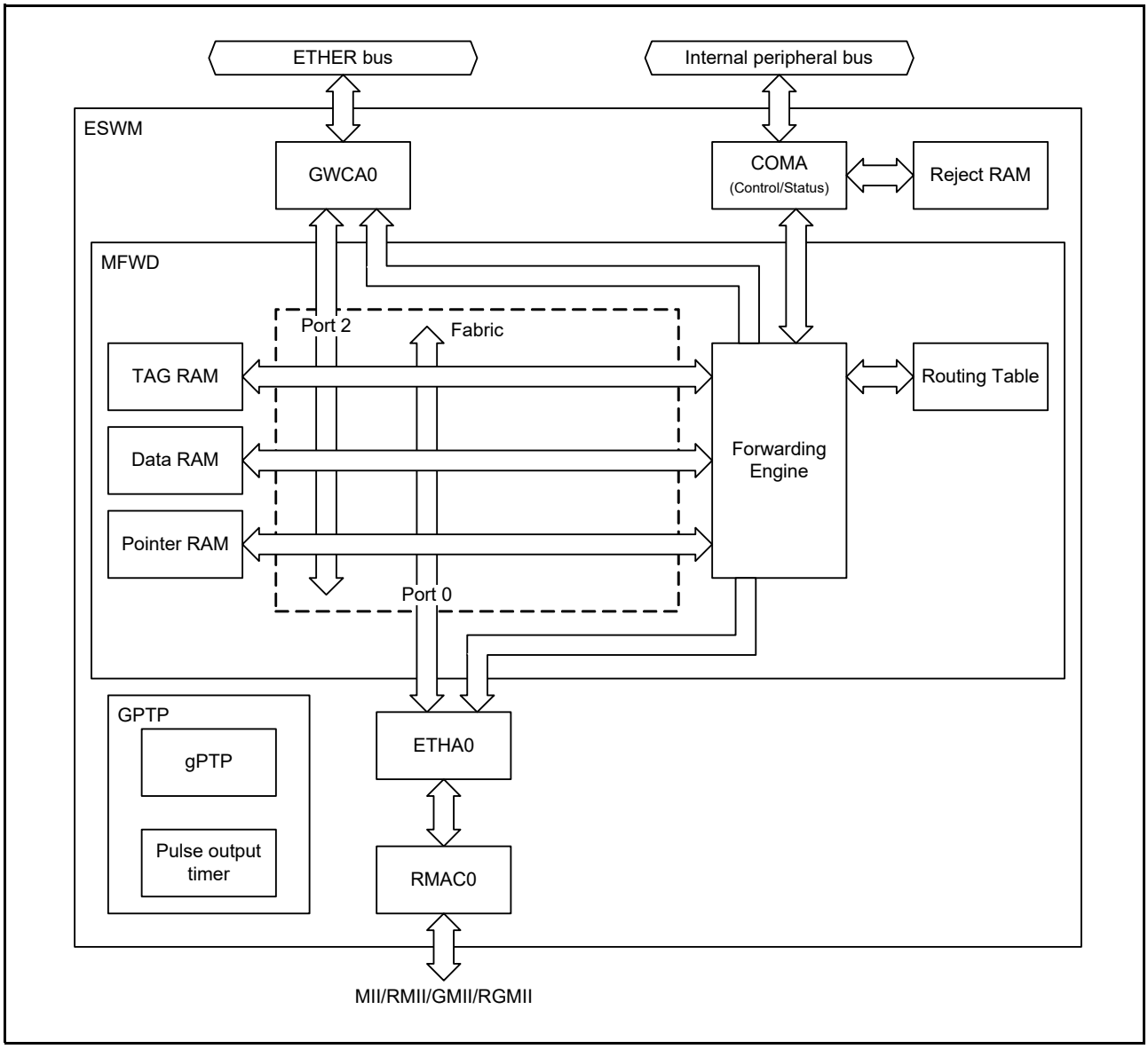


Figure 33.1 ESWM Block Diagram

Table 33.2 ESWM I/O Pins

Interface	Pin Name	I/O	Description
MII/GMII	ET0_GTX_CLK	Output	1000 Mbps transmit clock
	ET0_TX_CLK	I/O	100 Mbps, 10 Mbps transmit clock
	ET0_RX_CLK	I/O	Receive clock
	ET0_TX_EN	Output	Transmit enable
	ET0_TXD7 to ET0_TXD0	Output	Transmit data
	ET0_TX_ER	Output	Transmit coding error
	ET0_RX_DV	Input	Receive data valid
	ET0_RXD7 to ET0_RXD0	Input	Receive data
	ET0_RX_ER	Input	Receive error
	ET0_MDC	Output	Management data clock
	ET0_MDIO	I/O	Management data input/output
	ET0_CRS	Input	Receive Carrier Sense for MII Half Duplex
	ET0_COL	Input	Receive Collision detection for MII Half Duplex
RGMII	RGMII0_TXC	Output	Transmit clock
	RGMII0_RXC	Input	Receive clock
	RGMII0_TX_CTL	Output	Transmit control
	RGMII0_TXD3 to RGMII0_TXD0	Output	Transmit data
	RGMII0_RX_CTL	Input	Receive control
	RGMII0_RXD3 to RGMII0_RXD0	Input	Receive data
RMII	RMII0_REF50CK	Input	Synchronous clock reference
	RMII0_TX_EN	Output	Transmit enable
	RMII0_TXD1, RMII0_TXD0	Output	Transmit data
	RMII0_CRS_DV	Input	Carrier sense/Receive data valid
	RMII0_RXD1, RMII0_RXD0	Input	Receive data
	RMII0_RX_ER	Input	Receive error
Common	ET0_LINKSTA	Input	PHY link status
	ET0_INT	Input	PHY interrupt
	ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.
GPTP	GPTP_CAPTURE0, GPTP_CAPTURE1	Input	Media clock capture input
	GPTP_MATCH0, GPTP_MATCH1	Output	Media clock recovery output
	GPTP0_PPS	Output	PPS signal
	GPTP_PTPOUT0, GPTP_PTPOUT1	Output	PTP Pulse generator signal

Table 33.3 lists an overview of each modules that configures the ESWM system.

Table 33.3 ESWM Functional Blocks

Module	Function	Operating Clock*1	Address
ESWM	Top module	—	874C 8000h to 874C 8FFFh, 874E 1400h to 874E 1FFFh
MFWD	Core module of the ESWM system. Forwards descriptors received from a port (GWCA or ETHA) to other ports based on the destination MAC address, VLAN code, IP address, port number, etc.	ESWCLK	874C 0000h to 874C 7FFFh
COMA	Interface module between CPU and ESWM system. Access control of ESWM internal registers, pointer control of ESWM local RAM.	PCLKA ESWCLK	874C 9000h to 874C 9FFFh
ETHA	Ethernet frame generation module. Transmission: Generates Ethernet frames based on the descriptors received from MFWD. Reception: Forwards frames received from RMAC to MFWD.	ESWCLK	874C A000h to 874C AFFFh
RMAC	IEEE 802.3 compliant media access controller. Interfaces physical layer LSI chip (PHY-LSI) with ETHA and performs message filtering.	ESWCLK ESWPHYCLK (RGMII)	874C B000h to 874C BFFFh
GWCA	Interface module with ETHER bus (bus master). Transmission: Transfers descriptors and frame data read from CPU memory to MFWD. Reception: Transfers descriptors and frame data received from MFWD to CPU memory.	ICLK ESWCLK	874C E000h to 874C EFFFh
GPTP	gPTP timer module Pulse output timer	PCLKA ESWCLK	874E 0000h to 874E 13FFh

Note 1. Refer to section 33.3.1, Operating Clocks for the frequency of each clock.

33.2 Register Description

33.2.1 ESWM Interface Control Registers

33.2.1.1 Media Interface Reset Register (MIIRR)

Address(es): ESWM.MIIRR 874E 1400h, ESWM_NS.MIIRR 074E 1400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RMRST 0	—	—	—	—	—	—	—	RGRST 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RGRST0	RGMII0 Reset	0: Initiate RGMII0 reset 1: Release RGMII0 reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	RMRST0	RMII0 Reset	0: Initiate RMII0 reset 1: Release RMII0 reset	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.1.2 Media Interface Control Register 0 (MIICR0)

Address(es): ESWM.MIICR0 874E 1404h, ESWM_NS.MIICR0 074E 1404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TXCIDE	—	—	—	DIVSTP	—	—	M2MCS[1:0]		—	M2MEN	MIISEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MIISEL[1:0]	MII Select	b1 b0 0 0: MII/GMII 0 1: RGMII 1 0: RMII 1 1: Setting prohibited	R/W
b2	M2MEN	MAC-to-MAC Connection Enable	0: Setting for PHY-LSI connection (TX_CLK and RX_CLK are input). 1: Setting for MAC-to-MAC connection (TX_CLK and RX_CLK are output).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	M2MCS[1:0]	ESWPHYCLK Select for MAC-to-MAC Connection*1	b5 b4 0 0: 500 MHz 0 1: 200 MHz 1 0: 100 MHz 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DIVSTP	Clock Divider Stop	0: Enable 1: Stop	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	TXCIDE	RGMII TXC Internal Delay Enable	0: TXC Internal delay is disabled. 1: TXC Internal delay is enabled.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Refer to section 33.3.2, Interface Modes for the M2MCS[1:0] bit settings.

M2MEN Bit (MAC-to-MAC Connection Enable)

Set this bit to 1 when connecting to the MAC interface of another LSI. When this bit is set to 1, the ET0_TX_CLK and ET0_RX_CLK pins become outputs. If connecting ESWM modules together, set this bit to 1 for only one of the ESWM modules.

M2MCS[1:0] Bits (ESWPHYCLK Select for MAC-to-MAC Connection)

Set the frequency of the ESWPHYCLK generated by the clock generation circuit to these bits. These bits are valid when the M2MEN bit is set to 1.

DIVSTP Bit (Clock Divider Stop)

This bit is used to stop the frequency divider that generates each transmit clock from the ESWPHYCLK. this bit can be used in Low Power Idle (LPI) mode of Energy Efficient Ethernet.

33.2.1.3 Media Clock Capture Event Select Register (MCCESR)

Address(es): ESWM.MCCESR 874E 1410h, ESWM_NS.MCCESR 074E 1410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCES1	MCCES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCCES0	Media Clock Capture Event 0 Select	0: Input from the GPTP_CAPTURE0 pin 1: Event input from ELC	R/W
b1	MCCES1	Media Clock Capture Event 1 Select	0: Input from the GPTP_CAPTURE1 pin 1: Event input from ELC	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.1.4 TAS Status Monitor Signal Select Register (TASSTSR)

Address(es): ESWM.TASSTSR 874E 1420h, ESWM_NS.TASSTSR 074E 1420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSS3[4:0]					—	—	—	MSS2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MSS1[4:0]					—	—	—	MSS0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSS0[4:0]	ET_TAS_STA0 Monitor Signal Select	Select a signal to be output from the ET_TAS_STA0 pin. b4 b0 00000: Value of the ETHA0.EATASSM.TASGS0 flag 00001: Value of the ETHA0.EATASSM.TASGS1 flag 00010: Value of the ETHA0.EATASSM.TASGS2 flag 00011: Value of the ETHA0.EATASSM.TASGS3 flag 00100: Value of the ETHA0.EATASSM.TASGS4 flag 00101: Value of the ETHA0.EATASSM.TASGS5 flag 00110: Value of the ETHA0.EATASSM.TASGS6 flag 00111: Value of the ETHA0.EATASSM.TASGS7 flag 01000: Value of the ETHA0.EATASSM.TASCTGS flag Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	MSS1[4:0]	ET_TAS_STA1 Monitor Signal Select	Select a signal to be output from the ET_TAS_STA1 pin. Refer to the MSS0[4:0] bits for the setting values.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	MSS2[4:0]	ET_TAS_STA2 Monitor Signal Select	Select a signal to be output from the ET_TAS_STA2 pin. Refer to the MSS0[4:0] bits for the setting values.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	MSS3[4:0]	ET_TAS_STA3 Monitor Signal Select	Select a signal to be output from the ET_TAS_STA3 pin. Refer to the MSS0[4:0] bits for the setting values.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.1.5 Ether Bus Control Register (ETHBUSCR)

Address(es): ESWM.ETHBUSCR 874E 1424h, ESWM_NS.ETHBUSCR 074E 1424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENDIAN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENDIAN	Bus Endian Select	0: Little endian 1: Big endian	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ENDIAN Bit (Bus Endian Select)

This bit is used to set the endian of the ESWM bus interface. Set this bit to the same endian as the CPU.

33.2.2 Interrupt Mapping Function Registers

33.2.2.1 Error and Status Interrupt Mapping Register 1 (TPEMIMC1)

Address(es): ESWM.TPEMIMC1 874C 8004h, ESWM_NS.TPEMIMC1 074C 8004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSICM[2:0]			—	—	CSIGM	CSIM	—	CEICM[2:0]			—	—	CEIGM	CEIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FSICM[2:0]			—	—	FSIGM	FSIM	—	FEICM[2:0]			—	—	FEIGM	FEIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FEIM	MFWD Error Interrupt Mapping	0: Generates an FWEI interrupt. 1: Follows the setting of the FEIGM and FEICM[2:0] bits. MFWD error interrupt includes the following sources: • MFWD.FWEIS0i (i = 0, 2)	R/W
b1	FEIGM	MFWD Error Interrupt GWCA Mapping	This bit is valid only when the FEIM bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the FEICM[2:0] bits. 1: Does not output MFWD error interrupt.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b6 to b4	FEICM[2:0]	MFWD Error Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	FSIM	MFWD Status Interrupt Mapping	0: Generates an FWSI interrupt. 1: Follows the setting of the FSIGM and FSICM[2:0] bits. MFWD status interrupt includes the following sources: • MFWD.FWMIS0	R/W
b9	FSIGM	MFWD Status Interrupt GWCA Mapping	This bit is valid only when the FSIM bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the FSICM[2:0] bits. 1: Does not output MFWD status interrupt.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	FSICM[2:0]	MFWD Status Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	CEIM	COMA Error Interrupt Mapping	0: Generates a CAEI interrupt. 1: Follows the setting of the CEIGM and CEICM[2:0] bits. COMA error interrupt includes the following sources: • COMA.CAEIS0 • COMA.CAEIS1	R/W
b17	CEIGM	COMA Error Interrupt GWCA Mapping	This bit is valid only when the CEIM bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the CEICM[2:0] bits. 1: Does not output COMA error interrupt.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b20	CEICM[2:0]	COMA Error Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b24	CSIM	COMA Status Interrupt Mapping	0: Generates a CASI interrupt. 1: Follows the setting of the CSIGM and CSICM[2:0] bits. COMA status interrupt includes the following sources: • COMA.CAMIS0 • COMA.CAMIS1	R/W
b25	CSIGM	COMA Status Interrupt GWCA Mapping	This bit is valid only when the CSIM bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the CSICM[2:0] bits. 1: Does not output COMA status interrupt.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30 to b28	CSICM[2:0]	COMA Status Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

33.2.2.2 Error and Status Interrupt Mapping Register 2 (TPEMIMC2)

Address(es): ESWM.TPEMIMC2 874C 8008h, ESWM_NS.TPEMIMC2 074C 8008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	GEICM0[2:0]			—	—	GEIGM0	GEIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GEIM0	GWCA0 Error Interrupt Mapping	0: Generates a GWEI0 interrupt. 1: Follows the setting of the GEIGM0 and GEICM0[2:0] bits. GWCA0 error interrupt includes the following sources: • GWCA0.GWEIS0 (excluding the TDFES flag) • GWCA0.GWEIS1 • GWCA0.GWEIS4 • GWCA0.GWEIS5	R/W
b1	GEIGM0	GWCA0 Error Interrupt GWCA Mapping	This bit is valid only when the GEIM0 bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the GEICM0[2:0] bits. 1: Does not output GWCA0 error interrupt.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b6 to b4	GEICM0[2:0]	GWCA0 Error Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.2.3 Error and Status Interrupt Mapping Register 3 (TPEMIMC3)

Address(es): ESWM.TPEMIMC3 874C 800Ch, ESWM_NS.TPEMIMC3 074C 800Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ESICM0[2:0]			—	—	ESIGM0	ESIM0	—	EEICM0[2:0]			—	—	EEIGM0	EEIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EEIM0	ETHA0 Error Interrupt Mapping	0: Generates an EAEI0 interrupt. 1: Follows the setting of the EEIGM0 and EEICM0[2:0] bits. ETHA0 error interrupt includes the following sources: • ETHA0.EAEIS0 • ETHA0.EAEIS1 • ETHA0.EAEIS2 • RMAC0.MEIS	R/W
b1	EEIGM0	ETHA0 Error Interrupt GWCA Mapping	This bit is valid only when the EEIM0 bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the EEICM0[2:0] bits. 1: Does not output ETHA0 error interrupt.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b6 to b4	EEICM0[2:0]	ETHA0 Error Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	ESIM0	ETHA0 Status Interrupt Mapping	0: Generates an EASI0 interrupt. 1: Follows the setting of the ESIGM0 and ESICM0[2:0] bits. ETHA0 status interrupt includes the following sources: • RMAC0.MMIS0	R/W
b9	ESIGM0	ETHA0 Status Interrupt GWCA Mapping	This bit is valid only when the ESIM0 bit is 1. 0: Generates a GWDI0m interrupt. The value of m is specified by the ESICM0[2:0] bits. 1: Does not output ETHA0 status interrupt.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	ESICM0[2:0]	ETHA0 Status Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.2.4 Error and Status Interrupt Mapping Register 6n (TPEMIMC6n) (n = 0, 1)

Address(es): ESWM.TPEMIMC60 874C 8080h, ESWM.TPEMIMC61 874C 8084h,
ESWM_NS.TPEMIMC60 074C 8080h, ESWM_NS.TPEMIMC61 074C 8084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	GTSICM0[2:0]			GTSIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GTSIM0	GWCA0 Timestamp Interrupt Mapping	0: Generates a TSDI0n interrupt. 1: Generates a GWDI0m interrupt. The value of m is specified by the GTSICM0[2:0] bits. GWCA0 timestamp interrupt n includes the following sources: • GWCA0.GWTSDIS.TSDISn • GWCA0.GWEIS0.TDFESn	R/W
b3 to b1	GTSICM0[2:0]	GWCA0 Timestamp Interrupt Core Mapping	These bits specify the value of m for the GWDI0m interrupt.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.2.5 Error and Status Interrupt Mapping Register 7n (TPEMIMC7n) (n = 0 to 63)

Address(es): ESWM.TPEMIMC70 874C 8100h, ESWM.TPEMIMC71 874C 8104h, ESWM.TPEMIMC72 874C 8108h, ESWM.TPEMIMC73 874C 810Ch, ESWM.TPEMIMC74 874C 8110h, ESWM.TPEMIMC75 874C 8114h, ESWM.TPEMIMC76 874C 8118h, ESWM.TPEMIMC77 874C 811Ch, ESWM.TPEMIMC78 874C 8120h, ESWM.TPEMIMC79 874C 8124h, ESWM.TPEMIMC710 874C 8128h, ESWM.TPEMIMC711 874C 812Ch, ESWM.TPEMIMC712 874C 8130h, ESWM.TPEMIMC713 874C 8134h, ESWM.TPEMIMC714 874C 8138h, ESWM.TPEMIMC715 874C 813Ch, ESWM.TPEMIMC716 874C 8140h, ESWM.TPEMIMC717 874C 8144h, ESWM.TPEMIMC718 874C 8148h, ESWM.TPEMIMC719 874C 814Ch, ESWM.TPEMIMC720 874C 8150h, ESWM.TPEMIMC721 874C 8154h, ESWM.TPEMIMC722 874C 8158h, ESWM.TPEMIMC723 874C 815Ch, ESWM.TPEMIMC724 874C 8160h, ESWM.TPEMIMC725 874C 8164h, ESWM.TPEMIMC726 874C 8168h, ESWM.TPEMIMC727 874C 816Ch, ESWM.TPEMIMC728 874C 8170h, ESWM.TPEMIMC729 874C 8174h, ESWM.TPEMIMC730 874C 8178h, ESWM.TPEMIMC731 874C 817Ch, ESWM.TPEMIMC732 874C 8180h, ESWM.TPEMIMC733 874C 8184h, ESWM.TPEMIMC734 874C 8188h, ESWM.TPEMIMC735 874C 818Ch, ESWM.TPEMIMC736 874C 8190h, ESWM.TPEMIMC737 874C 8194h, ESWM.TPEMIMC738 874C 8198h, ESWM.TPEMIMC739 874C 819Ch, ESWM.TPEMIMC740 874C 81A0h, ESWM.TPEMIMC741 874C 81A4h, ESWM.TPEMIMC742 874C 81A8h, ESWM.TPEMIMC743 874C 81ACh, ESWM.TPEMIMC744 874C 81B0h, ESWM.TPEMIMC745 874C 81B4h, ESWM.TPEMIMC746 874C 81B8h, ESWM.TPEMIMC747 874C 81BCh, ESWM.TPEMIMC748 874C 81C0h, ESWM.TPEMIMC749 874C 81C4h, ESWM.TPEMIMC750 874C 81C8h, ESWM.TPEMIMC751 874C 81CCh, ESWM.TPEMIMC752 874C 81D0h, ESWM.TPEMIMC753 874C 81D4h, ESWM.TPEMIMC754 874C 81D8h, ESWM.TPEMIMC755 874C 81DCh, ESWM.TPEMIMC756 874C 81E0h, ESWM.TPEMIMC757 874C 81E4h, ESWM.TPEMIMC758 874C 81E8h, ESWM.TPEMIMC759 874C 81ECh, ESWM.TPEMIMC760 874C 81F0h, ESWM.TPEMIMC761 874C 81F4h, ESWM.TPEMIMC762 874C 81F8h, ESWM.TPEMIMC763 874C 81FCh,
ESWM_NS.TPEMIMC70 074C 8100h, ESWM_NS.TPEMIMC71 074C 8104h, ESWM_NS.TPEMIMC72 074C 8108h, ESWM_NS.TPEMIMC73 074C 810Ch, ESWM_NS.TPEMIMC74 074C 8110h, ESWM_NS.TPEMIMC75 074C 8114h, ESWM_NS.TPEMIMC76 074C 8118h, ESWM_NS.TPEMIMC77 074C 811Ch, ESWM_NS.TPEMIMC78 074C 8120h, ESWM_NS.TPEMIMC79 074C 8124h, ESWM_NS.TPEMIMC710 074C 8128h, ESWM_NS.TPEMIMC711 074C 812Ch, ESWM_NS.TPEMIMC712 074C 8130h, ESWM_NS.TPEMIMC713 074C 8134h, ESWM_NS.TPEMIMC714 074C 8138h, ESWM_NS.TPEMIMC715 074C 813Ch, ESWM_NS.TPEMIMC716 074C 8140h, ESWM_NS.TPEMIMC717 074C 8144h, ESWM_NS.TPEMIMC718 074C 8148h, ESWM_NS.TPEMIMC719 074C 814Ch, ESWM_NS.TPEMIMC720 074C 8150h, ESWM_NS.TPEMIMC721 074C 8154h, ESWM_NS.TPEMIMC722 074C 8158h, ESWM_NS.TPEMIMC723 074C 815Ch, ESWM_NS.TPEMIMC724 074C 8160h, ESWM_NS.TPEMIMC725 074C 8164h, ESWM_NS.TPEMIMC726 074C 8168h, ESWM_NS.TPEMIMC727 074C 816Ch, ESWM_NS.TPEMIMC728 074C 8170h, ESWM_NS.TPEMIMC729 074C 8174h, ESWM_NS.TPEMIMC730 074C 8178h, ESWM_NS.TPEMIMC731 074C 817Ch, ESWM_NS.TPEMIMC732 074C 8180h, ESWM_NS.TPEMIMC733 074C 8184h, ESWM_NS.TPEMIMC734 074C 8188h, ESWM_NS.TPEMIMC735 074C 818Ch, ESWM_NS.TPEMIMC736 074C 8190h, ESWM_NS.TPEMIMC737 074C 8194h, ESWM_NS.TPEMIMC738 074C 8198h, ESWM_NS.TPEMIMC739 074C 819Ch, ESWM_NS.TPEMIMC740 074C 81A0h, ESWM_NS.TPEMIMC741 074C 81A4h, ESWM_NS.TPEMIMC742 074C 81A8h, ESWM_NS.TPEMIMC743 074C 81ACh, ESWM_NS.TPEMIMC744 074C 81B0h, ESWM_NS.TPEMIMC745 074C 81B4h, ESWM_NS.TPEMIMC746 074C 81B8h, ESWM_NS.TPEMIMC747 074C 81BCh, ESWM_NS.TPEMIMC748 074C 81C0h, ESWM_NS.TPEMIMC749 074C 81C4h, ESWM_NS.TPEMIMC750 074C 81C8h, ESWM_NS.TPEMIMC751 074C 81CCh, ESWM_NS.TPEMIMC752 074C 81D0h, ESWM_NS.TPEMIMC753 074C 81D4h, ESWM_NS.TPEMIMC754 074C 81D8h, ESWM_NS.TPEMIMC755 074C 81DCh, ESWM_NS.TPEMIMC756 074C 81E0h, ESWM_NS.TPEMIMC757 074C 81E4h, ESWM_NS.TPEMIMC758 074C 81E8h, ESWM_NS.TPEMIMC759 074C 81ECh, ESWM_NS.TPEMIMC760 074C 81F0h, ESWM_NS.TPEMIMC761 074C 81F4h, ESWM_NS.TPEMIMC762 074C 81F8h, ESWM_NS.TPEMIMC763 074C 81FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	GDICM0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GDICM0[2:0]	GWCA0 Data Interrupt Core Mapping	GWCA0 data interrupt n is output as a GWDI0m interrupt. These bits specify the value of m. GWCA0 data interrupt n includes the following sources: • GWCA0.GWDISi.DISn (i = 0, 1) • GWCA0.GWEIS2i.DFESn (i = 0, 1) • GWCA0.GWEIS3.IAOESk (k = 0 to 3)	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.3 Interrupt Mirroring Registers

33.2.3.1 ESWM Interrupt Mirroring Register (TSIM)

Address(es): ESWM.TSIM 874C 8700h, ESWM_NS.TSIM 074C 8700h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	EIM0	—	GIM0	CIM	FIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FIM	MFWD Interrupt Mirroring Flag	0: No interrupt request from the MFWD module 1: An interrupt request from the MFWD module	R
b1	CIM	COMA Interrupt Mirroring Flag	0: No interrupt request from the COMA module 1: An interrupt request from the COMA module	R
b2	GIM0	GWCA0 Interrupt Monitoring Flag	0: No interrupt request from the GWCA0 module 1: An interrupt request from the GWCA0 module	R
b3	—	Reserved	This bit is read as 0.	R
b4	EIM0	ETHA0 Interrupt Monitoring Flag	0: No interrupt request from the ETHA0 module 1: An interrupt request from the ETHA0 module	R
b31 to b5	—	Reserved	These bits are read as 0.	R

33.2.3.2 MFWD Interrupt Mirroring Register (TFIM)

Address(es): ESWM.TFIM 874C 8704h, ESWM_NS.TFIM 074C 8704h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FWMISIM0	FWEISIM8	FWEISIM7	FWEISIM6	FWEISIM5	FWEISIM4	FWEISIM3	FWEISIM2	FWEISIM1	FWEISIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FWEISIM0	FWEIS0 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS0 register are set to 0. 1: Any flag in the MFWD.FWEIS0 register is set to 1.	R
b1	FWEISIM1	FWEIS1 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS1 register are set to 0. 1: Any flag in the MFWD.FWEIS1 register is set to 1.	R
b2	FWEISIM2	FWEIS2 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS2 register are set to 0. 1: Any flag in the MFWD.FWEIS2 register is set to 1.	R
b3	FWEISIM3	FWEIS3 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS3 register are set to 0. 1: Any flag in the MFWD.FWEIS3 register is set to 1.	R
b4	FWEISIM4	FWEIS4 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS4 register are set to 0. 1: Any flag in the MFWD.FWEIS4 register is set to 1.	R
b5	FWEISIM5	FWEIS5 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS5 register are set to 0. 1: Any flag in the MFWD.FWEIS5 register is set to 1.	R
b6	FWEISIM6	FWEIS6 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS6 register are set to 0. 1: Any flag in the MFWD.FWEIS6 register is set to 1.	R
b7	FWEISIM7	FWEIS7 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS7 register are set to 0. 1: Any flag in the MFWD.FWEIS7 register is set to 1.	R
b8	FWEISIM8	FWEIS8 Interrupt Mirroring Flag	0: All flags in the MFWD.FWEIS8 register are set to 0. 1: Any flag in the MFWD.FWEIS8 register is set to 1.	R
b9	FWMISIM0	FWMIS0 Interrupt Mirroring Flag	0: All flags in the MFWD.FWMIS0 register are set to 0. 1: Any flag in the MFWD.FWMIS0 register is set to 1.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

33.2.3.3 COMA Interrupt Mirroring Register (TCIM)

Address(es): ESWM.TCIM 874C 8708h, ESWM_NS.TCIM 074C 8708h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CAMISIM1	CAMISIM0	CAEISIM1	CAEISIM0	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0.	R
b1	CAEISIM0	CAEIS0 Interrupt Mirroring Flag	0: All flags in the COMA.CAEIS0 register are set to 0. 1: Any flag in the COMA.CAEIS0 register is set to 1.	R
b2	CAEISIM1	CAEIS1 Interrupt Mirroring Flag	0: All flags in the COMA.CAEIS1 register are set to 0. 1: Any flag in the COMA.CAEIS1 register is set to 1.	R
b3	CAMISIM0	CAMIS0 Interrupt Mirroring Flag	0: All flags in the COMA.CAMIS0 register are set to 0. 1: Any flag in the COMA.CAMIS0 register is set to 1.	R
b4	CAMISIM1	CAMIS1 Interrupt Mirroring Flag	0: All flags in the COMA.CAMIS1 register are set to 0. 1: Any flag in the COMA.CAMIS1 register is set to 1.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

33.2.3.4 GWCA0 Interrupt Mirroring Register (TGIM0)

Address(es): ESWM.TGIM0 874C 8710h, ESWM_NS.TGIM0 074C 8710h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GWEISIM5	GWEISIM4	GWEISIM3	GWEISIM2	GWEISIM1	GWEISIM0	GWTSDISIM	GWDISIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GWDISIM	GWDIS Interrupt Mirroring Flag	0: All flags in the GWCA0.GWDISi registers are set to 0 (i = 0, 1). 1: Any flag in the GWCA0.GWDISi registers is set to 1.	R
b1	GWTSDISIM	GWTSDIS Interrupt Mirroring Flag	0: All flags in the GWCA0.GWTSDIS register are set to 0. 1: Any flag in the GWCA0.GWTSDIS register is set to 1.	R
b2	GWEISIM0	GWEIS0 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS0 register are set to 0. 1: Any flag in the GWCA0.GWEIS0 register is set to 1.	R
b3	GWEISIM1	GWEIS1 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS1 register are set to 0. 1: Any flag in the GWCA0.GWEIS1 register is set to 1.	R
b4	GWEISIM2	GWEIS2 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS2i registers are set to 0 (i = 0, 1). 1: Any flag in the GWCA0.GWEIS2i registers is set to 1.	R
b5	GWEISIM3	GWEIS3 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS3 register are set to 0. 1: Any flag in the GWCA0.GWEIS3 register is set to 1.	R
b6	GWEISIM4	GWEIS4 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS4 register are set to 0. 1: Any flag in the GWCA0.GWEIS4 register is set to 1.	R
b7	GWEISIM5	GWEIS5 Interrupt Mirroring Flag	0: All flags in the GWCA0.GWEIS5 register are set to 0. 1: Any flag in the GWCA0.GWEIS5 register is set to 1.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

33.2.3.5 ETHA0 Interrupt Mirroring Register (TEIM0)

Address(es): ESWM.TEIM0 874C 8720h, ESWM_NS.TEIM0 074C 8720h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	EAEISIM3	MMISIM	MEISIM	EAEISIM2	EAEISIM1	EAEISIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EAEISIM0	EAEIS0 Interrupt Mirroring Flag	0: All flags in the ETHA0.EAEIS0 register are set to 0. 1: Any flag in the ETHA0.EAEIS0 register is set to 1.	R
b1	EAEISIM1	EAEIS1 Interrupt Mirroring Flag	0: All flags in the ETHA0.EAEIS1 register are set to 0. 1: Any flag in the ETHA0.EAEIS1 register is set to 1.	R
b2	EAEISIM2	EAEIS2 Interrupt Mirroring Flag	0: All flags in the ETHA0.EAEIS2 register are set to 0. 1: Any flag in the ETHA0.EAEIS2 register is set to 1.	R
b3	MEISIM	MEIS Interrupt Mirroring Flag	0: All flags in the RMAC0.MEIS register are set to 0. 1: Any flag in the RMAC0.MEIS register is set to 1.	R
b4	MMISIM	MMIS0 Interrupt Mirroring Flag	0: All flags in the RMAC0.MMIS0 register are set to 0. 1: Any flag in the RMAC0.MMIS0 register is set to 1.	R
b5	EAEISIM3	EAEIS3 Interrupt Mirroring Flag	0: All flags in the ETHA0.EAEIS3 register are set to 0. 1: Any flag in the ETHA0.EAEIS3 register is set to 1.	R
b31 to b6	—	Reserved	These bits are read as 0.	R

33.3 Operation

33.3.1 Operating Clocks

Table 33.4 lists the operating clocks for the ESWM system. These clocks are supplied by Clock Generation Circuit. Set the clock before starting operation of ESWM.

Table 33.4 Configurable Range of Clocks

Interface Mode	Operating Speed	ICLK	PCLKA	ESWCLK	ESWPHYCLK
GMII/RGMII	1000 Mbps	150 to 300 MHz	Up to 150 MHz, < ESWCLK, < ICLK	150 to 250 MHz, ≤ ICLK × 1.5	500 MHz
MII/RMII/RGMII	100/10 Mbps	50 to 300 MHz	Up to 150 MHz, < ESWCLK, < ICLK	50 to 250 MHz, ≤ ICLK × 1.5	500 MHz (RGMII only)
MAC-to-MAC connection	100/10 Mbps	50 to 300 MHz	Up to 150 MHz, < ESWCLK, < ICLK	50 to 250 MHz, ≤ ICLK × 1.5	100/200/500 MHz

33.3.2 Interface Modes

Table 33.5 lists the interface modes and their register settings. Set the MIICR0 register before setting the RMAC0.MPIC register.

Table 33.5 Interface Modes and Register Settings

		MIICR0 Register			RMAC0.MPIC Register	
Interface Mode	Speed	MIISEL[1:0] Bits	M2MEN Bit	M2MCS[1:0] Bits	LSC[2:0] Bits	PIS[2:0] Bits
MII	10 Mbps	00b	0	00b (not used)	000b	000b (MII)
MII	100 Mbps	00b	0	00b (not used)	001b	000b (MII)
GMII	1000 Mbps	00b	0	00b (500 MHz)	010b	010b (GMII)
RGMII	10 Mbps	01b	0	00b (500 MHz)	000b	000b (MII)
RGMII	100 Mbps	01b	0	00b (500 MHz)	001b	000b (MII)
RGMII	1000 Mbps	01b	0	00b (500 MHz)	010b	010b (GMII)
RMII	10 Mbps	10b	0	00b (not used)	000b	000b (MII)
RMII	100 Mbps	10b	0	00b (not used)	001b	000b (MII)
MAC-to-MAC	10 Mbps	00b	1	00b (500 MHz)	000b	000b (MII)
MAC-to-MAC	10 Mbps	00b	1	01b (200 MHz)	000b	000b (MII)
MAC-to-MAC	10 Mbps	00b	1	10b (100 MHz)	000b	000b (MII)
MAC-to-MAC	100 Mbps	00b	1	00b (500 MHz)	001b	000b (MII)
MAC-to-MAC	100 Mbps	00b	1	01b (200 MHz)	001b	000b (MII)
MAC-to-MAC	100 Mbps	00b	1	10b (100 MHz)	001b	000b (MII)

33.3.2.1 MII/GMII Interface

When the MIICR0.MIISEL[1:0] bits are set to 00b, MII/GMII interface is selected.
When using GMII, set the frequency of the ESWPHYCLK to 500 MHz. In this case, the ESWPHYCLK is divided by 4 to generate a 125 MHz GTX_CLK.

33.3.2.2 RGMII Interface

When the MIICR0.MIISEL[1:0] bits are set to 01b, RGMII interface is selected.
When using RGMII, set the frequency of the ESWPHYCLK to 500 MHz. In this case, the ESWPHYCLK is divided by 4 to generate a 125 MHz TXC.
The RGMII interface remains reset after the MCU reset is released. The RGMII interface can be enabled by setting the MIICR0.MIISEL[1:0] bits to 01b and then setting the MIIRR.RGRST0 bit to 1.

(1) RGMII Internal Delay Mode

The RGMII interface support internal delay mode for RGMII0_TXC.

33.3.2.3 RMII Interface

When the MIICR0.MIISEL[1:0] bits are set to 10b, RMII interface is selected.
The RMII interface remains reset after the MCU reset is released. The RMII interface can be enabled by setting the MIICR0.MIISEL[1:0] bits to 10b and then setting the MIIRR.RMRST0 bit to 1.

33.3.2.4 MAC-to-MAC Connection

ESWM has a function to directly connect and communicate between the MAC of this MCU and the MAC of another LSI. MAC-to-MAC connection only supports 10 Mbps or 100 Mbps full-duplex communication using MII. This function is enabled by setting the MIICR0.M2MEN bit to 1.
When using the MAC-to-MAC connection, set the frequency of the ESWPHYCLK to 100, 200, or 500 MHz and set the MIICR0.M2MCS[1:0] bits to match the frequency. At this time, the ESWPHYCLK is divided according to the settings of each bit to generate RX_CLK and TX_CLK. The generated clocks are output from the ET0_RX_CLK pin and ET0_TX_CLK pins.
Figure 33.2 shows an example of the MAC-to-MAC connection.

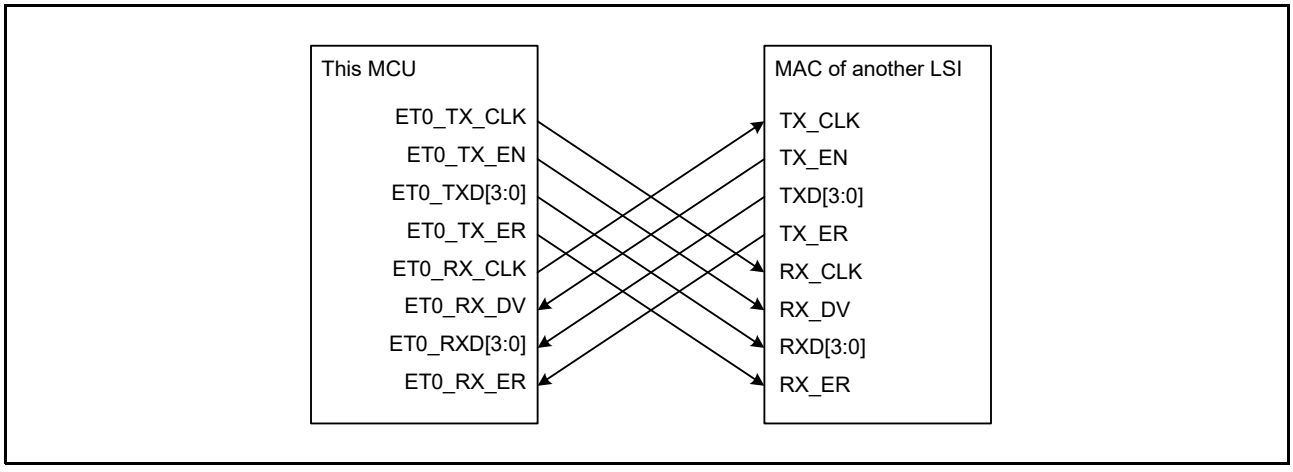


Figure 33.2 MAC-to-MAC Connection

33.3.3 Software Flows

33.3.3.1 Initialization Flow

Figure 33.3 shows the initialization flow for ESWM.

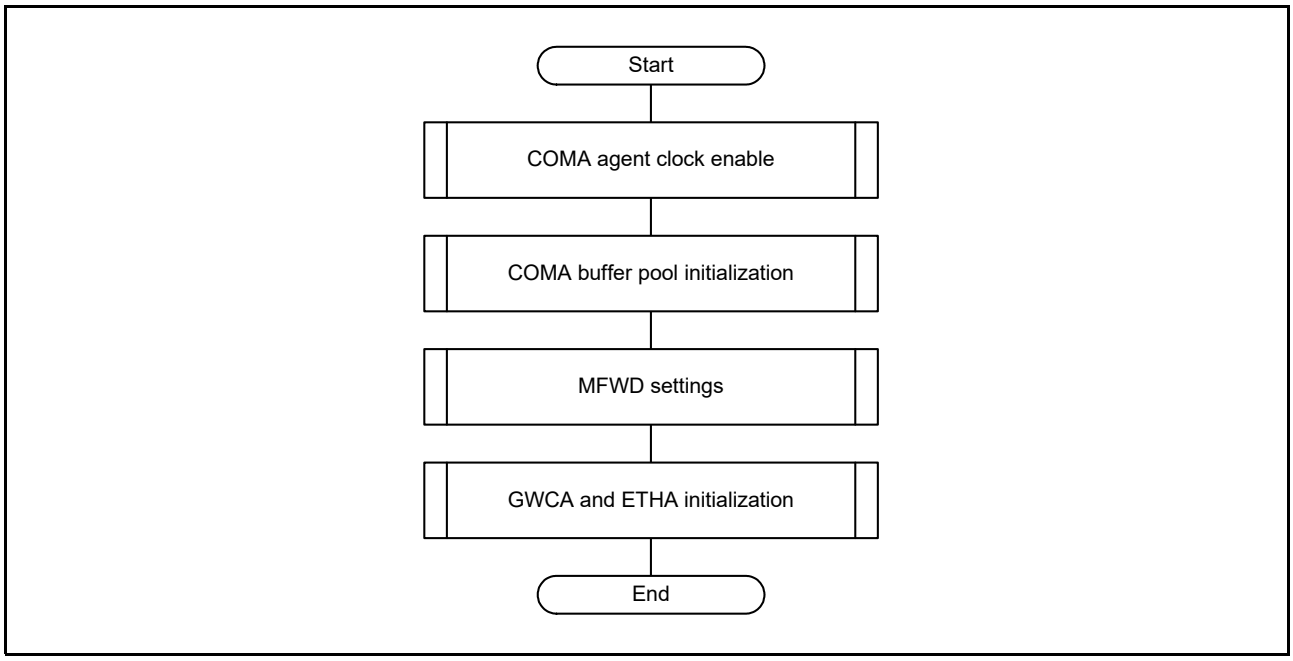


Figure 33.3 ESWM Initialization Flow

33.3.3.2 Disable Flow

Figure 33.4 shows the disable flow for ESWM.

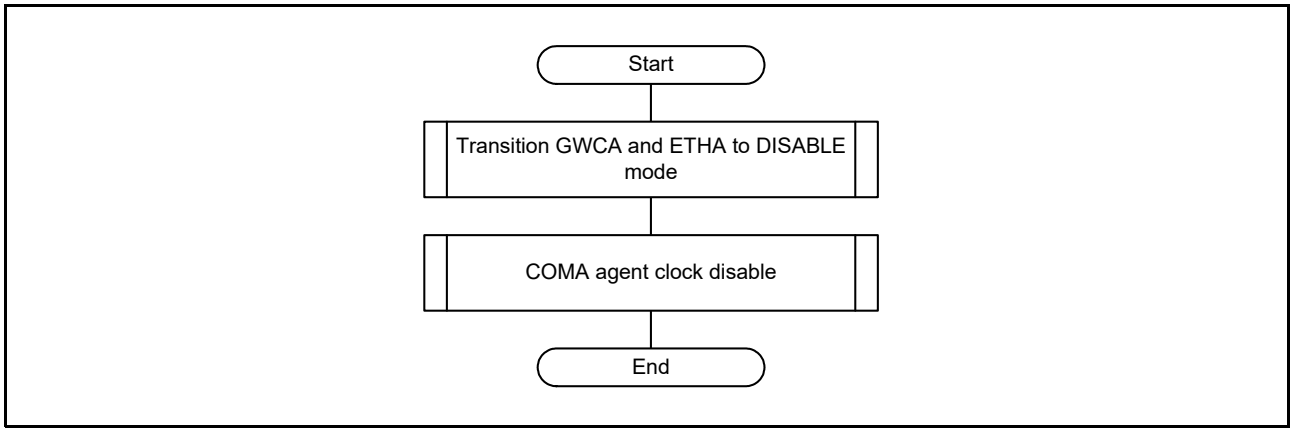


Figure 33.4 ESWM Disable Flow

33.3.3.3 Reset Flow

Figure 33.5 shows the reset flow for ESWM.

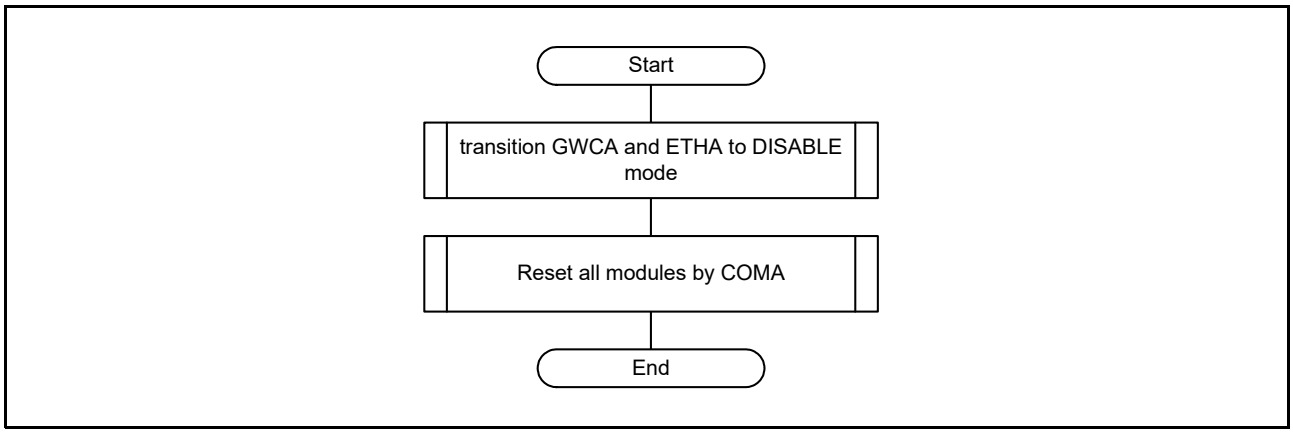


Figure 33.5 ESWM Reset Flow

33.3.3.4 Emergency Reset Flow

Figure 33.6 shows the emergency reset flow for ESWM.
This is used when ETHA is locked.
For example, when using MII, if an error occurs in the PHY-LSI and the transmit clock (ET0_TX_CLK) stops, ESWM cannot transmit data and cannot release the pointer (ETHA is locked in operation mode). This state will be resolved when the transmit clock resumes, but use an emergency reset if necessary.

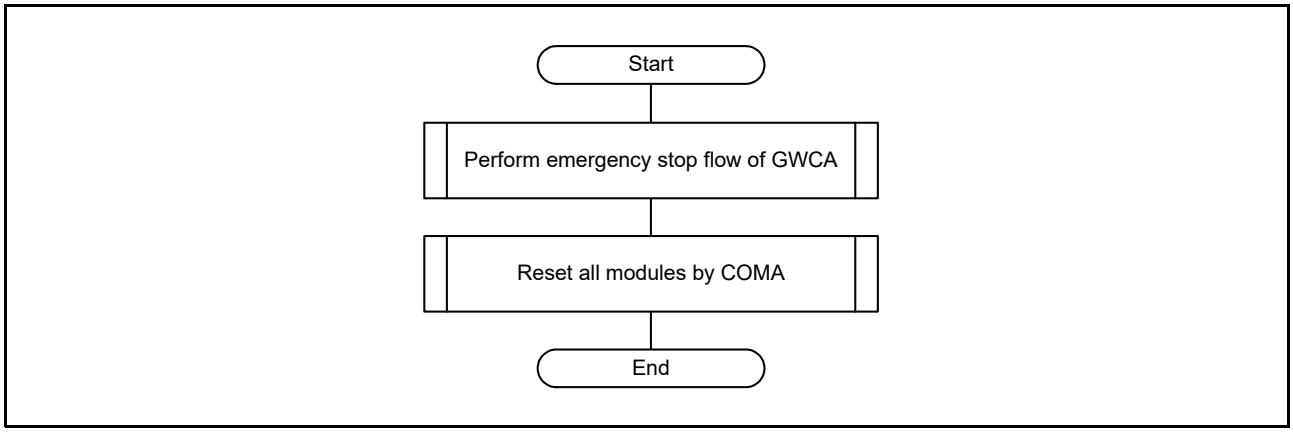


Figure 33.6 ESWM Emergency Reset Flow

33.4 Interrupts and Events

Table 33.6 Interrupt and Event Outputs

Module	Symbol	Name	Interrupt Source	Interrupt Detection
ESWM	FWEI	MFWD error interrupt	MFWD.FWEIS0i (i = 0, 2)	Level
	CAEI	COMA error interrupt	COMA.CAEIS0, COMA.CAEIS1	Level
	GWEI0	GWCA0 error interrupt	GWCA0.GWEIS0 (excluding TDFES), GWCA0.GWEIS1, GWCA0.GWEIS4, GWCA0.GWEIS5	Level
	EAEI0	ETHA0 error interrupt	ETHA0.EAEIS0, ETHA0.EAEIS1, ETHA0.EAEIS2, ETHA0.EAEIS3, RMAC0.MEIS	Level
	PTPSI0	gPTP status interrupt 0	GPTP.PTPIS0	Level
	PTPSI1	gPTP status interrupt 1	GPTP.PTPIS1	Level
	FWSI	MFWD status interrupt	MFWD.FWMIS0	Level
	CASI	COMA status interrupt	COMA.CAMIS0, COMA.CAMIS1	Level
	EASI0	ETHA0 status interrupt	RMAC0.MMIS0	Level
	GWDI00	GWCA0 data interrupt 0	GWCA0.GWDIS0, GWCA0.GWDIS1, GWCA0.GWEIS20, GWCA0.GWEIS21, GWCA0.GWEIS3	Level
	GWDI01	GWCA0 data interrupt 1		Level
	GWDI02	GWCA0 data interrupt 2		Level
	GWDI03	GWCA0 data interrupt 3		Level
	GWDI04	GWCA0 data interrupt 4		Level
	GWDI05	GWCA0 data interrupt 5		Level
	GWDI06	GWCA0 data interrupt 6		Level
	GWDI07	GWCA0 data interrupt 7		Level
	TSDI00	GWCA0 timestamp interrupt 0	GWCA0.GWTSDIS.TSDIS0, GWCA0.GWEIS0.TDFES0	Level
	TSDI01	GWCA0 timestamp interrupt 1	GWCA0.GWTSDIS.TSDIS1, GWCA0.GWEIS0.TDFES1	Level
	MDIO0	ETHA0 MDIO interrupt	RMAC0.MMIS1	Level
	RMPI0	RMAC0 PHY interrupt	RMAC0.MMIS2	Level
GPTP	MATCH0	Media clock 0 recovery interrupt/event	—	Edge
	MATCH1	Media clock 1 recovery interrupt/event	—	Edge
	PTPOUT0	Pulse output 0 interrupt/event	—	Edge
	PTPOUT1	Pulse output 1 interrupt/event	—	Edge

Table 33.7 Event Inputs

Module	Symbol	Name	Event Source
GPTP	CAPTURE0	gPTP media clock capture event 0	An event selected by the ELC.ELSR51 register
	CAPTURE1	gPTP media clock capture event 1	An event selected by the ELC.ELSR52 register

33.5 Usage Notes

33.5.1 Module-Stop Function Setting

Operation of the ESWM module can be disabled or enabled by setting a bit in the module stop control register C (MSTPCRC). The ESWM is initially disabled after a reset. Registers in the ESWM only become accessible after it has been released from the module-stop state. For details, refer to section 11., Low Power Consumption.

33.5.2 Transition to Software Standby Mode

Before transitioning to software standby mode, stop ESWM operation as described in section 33.3.3.2, Disable Flow. If RGMII or RMII interface is selected, it must be reset by the MIIRR register after the above procedure.

33.5.3 Return from Software Standby Mode

After returning from software standby mode, ESWM must be initialized as described in section 33.3.3.1, Initialization Flow.

33.5.4 Notes on Cutting Off the Power Supply to ESWM

Before cutting off the power supply to the ESWM power domain, stop the operation of ESWM as described in section 33.3.3.2, Disable Flow. If RGMII or RMII interface is selected, it must be reset by the MIIRR register after the above procedure.

33.5.5 Note on Resuming Power Supply to ESWM

After resuming power supply to the ESWM power domain, ESWM must be initialized as described in section 33.3.3.1, Initialization Flow.

33.5.6 Notes on Using Half-Duplex Mode

Notes on using half-duplex mode are listed below.

- The size of the transmit and receive frames must comply with IEEE 802.3.
- Do not use loopback mode.
- Do not use cut-through mode.
- Do not use with GMII or RGMII interface.
- The LPI function is disabled.

34. Ethernet Message Forwarding Engine (MFWD)

34.1 Overview

Ethernet message forwarding engine (MFWD) is part of ESWM system and aims at filtering, forwarding and routing the switch frames.
It snoops the frame information from Fabric to redistribute them to the agent after filtering, forwarding and routing mechanisms.

Table 34.1 shows the specifications of the MFWD, and Table 34.2 shows the specifications of the Fabric.

Table 34.1 MFWD Specifications

Function		Details
Overall function	Cut-through	Forwarding frames while receiving them depending on Layer 2
	Direct forwarding	Forwarding frames coming from GWCA by bypassing the MFWD
	Port based forwarding	Forwarding frames depending on source port

Table 34.2 Fabric Specifications

Function		Details
Overall function	Data transfer	• Function to transfer data and access to RAMs
Data provision		• Data for Fabric • Initial frame data + Intermediate data + Last frame data + Frame Descriptor • Data width: 185 bits
Control function	Time arbiter	• Arbitration between Time-critical agents [ETHA] for data bus. Arbitration scheme depends on each Time critical agents [ETHA] PHY link/Internal speed
	Arbitration	• LRU for Time critical-agents [ETHA], Slow agents [GWCA], and Common Agent (COMA) error read requests arbitration. • LRU between Slow agents [GWCA] for read/write bus arbitration. • Strict priority like arbitration between time arbiter, Error LRU, and slow LRU. Priority: time arbiter > slow LRU > Error LRU
	RAM access	• Function that writes and reads to RAM
	Transfer write request to MFWD	• Function to transfer write request sent from Ethernet Agents (ETHA) or GWCA to MFWD

Figure 34.1 shows MFWD block diagram. Figure 34.2 shows Fabric block diagram.

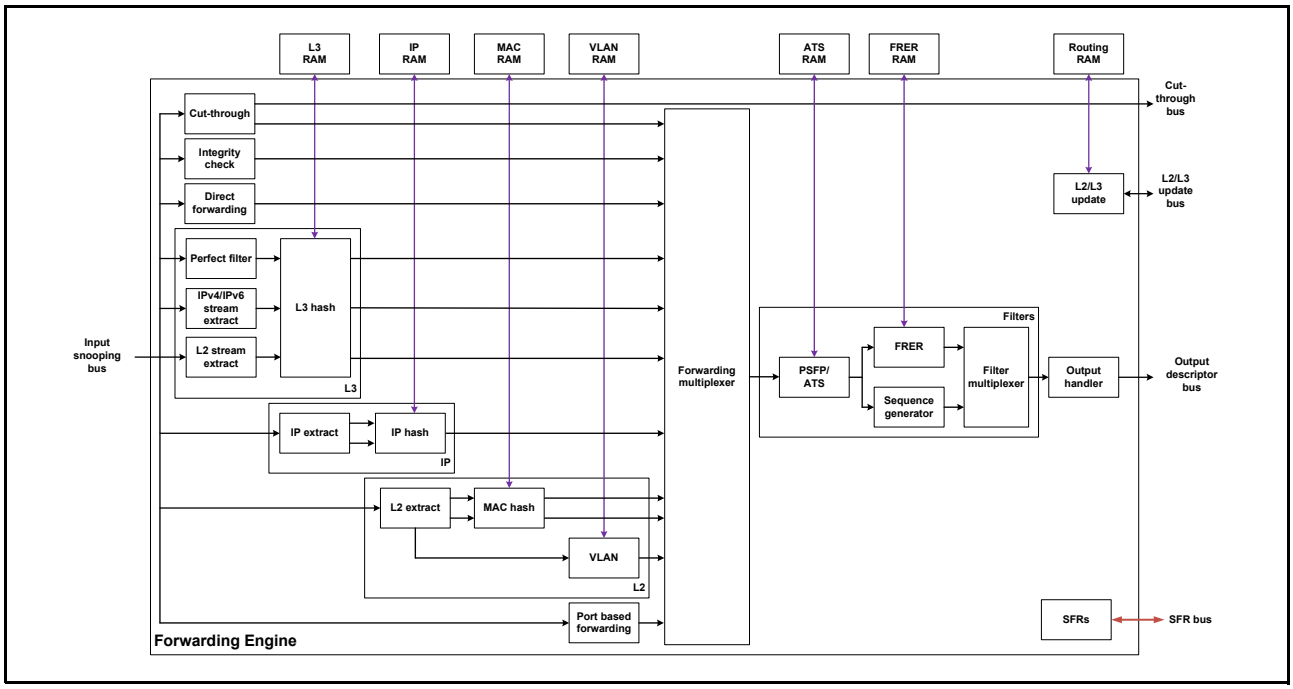


Figure 34.1 MFWD Block Diagram

Table 34.3 MFWD Functional Blocks

Block name	Function
Direct forwarding	Forward frames coming from a CPU with the forwarding information already present.
Port based forwarding	Source port-based forwarding
Forwarding multiplexer	Arbitrate between all the forwarding/ routing mechanisms
Filter multiplexer	Arbitrate between forwarding results and filtering results.
Output handler	Copy and send descriptors to targets. Manage overflow and security error recovery.

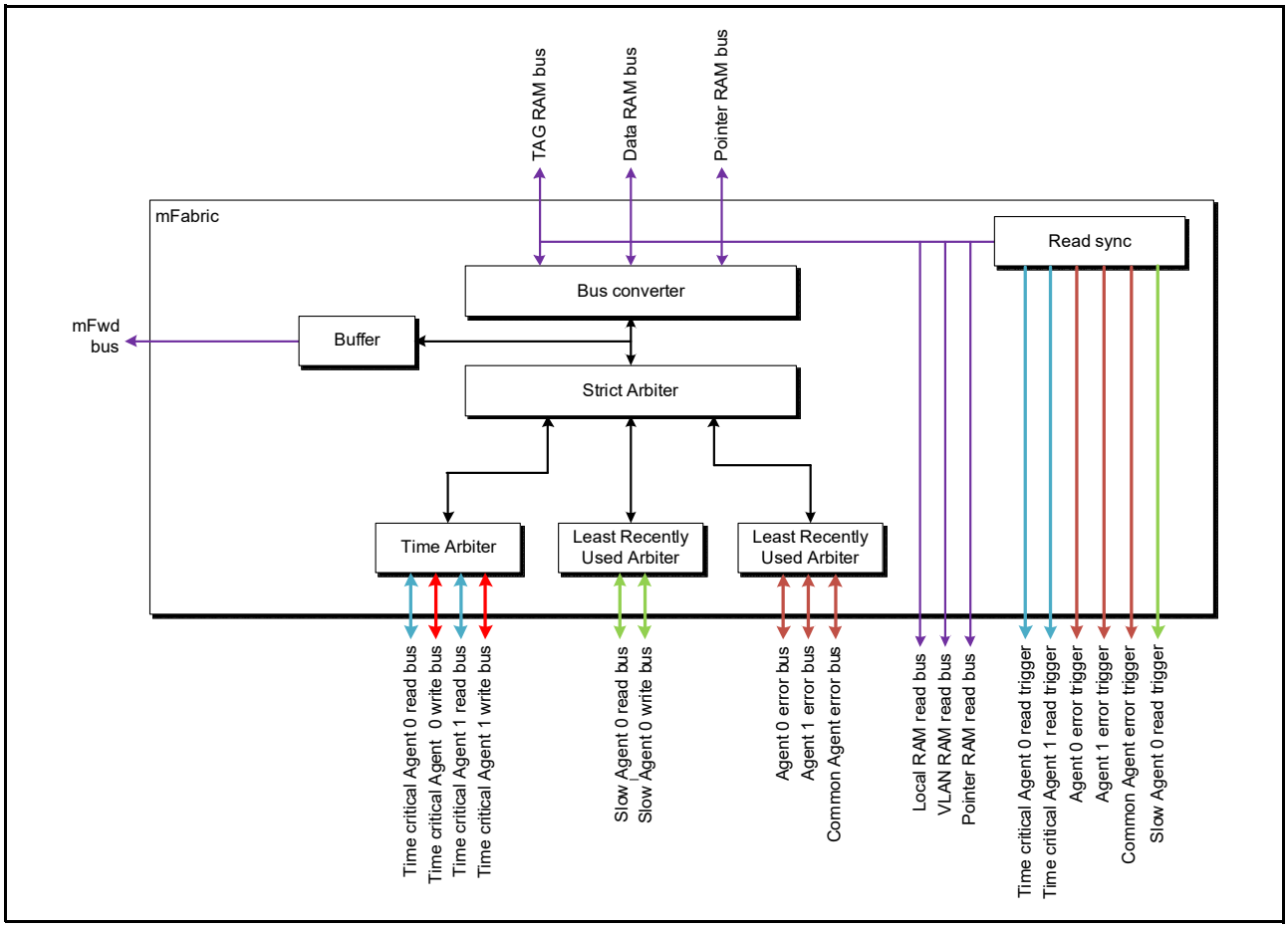


Figure 34.2 Fabric Block Diagram

Table 34.4 Fabric Functional Blocks

Block name	Function
Time arbiter	Function to share the Fabric throughput per agent depending on agent modes and PHY speed.
Least Recently Used arbiter	Arbiter between several buses. (LRU) If multiple error read requests collide, LRU arbitration is executed on a cycle by cycle basis.
Strict arbiter	Arbiter between several buses. If requests from Time arbiter, Round robin arbiter and GWCA collide, the higher priority request will be processed first. Priority ordering: Time-critical Write/Read request > Slow Write/Read request > Agent Error Read request
Read sync	Module that generates a signal indicating whether data read from RAM is valid or invalid.
Bus converter	Module that converts the internal access request in RAM protocol
Buffer	Register slices to meet design timings.

34.2 Register Descriptions

34.2.1 General Function Registers

34.2.1.1 General Configuration Register (FWGC)

Address(es): MFWD.FWGC 874C 0000h, MFWD_NS.FWGC 074C 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SVM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SVM[1:0]	Switch VLAN Mode	b1 b0 0 0: No VLAN mode (VLAN not used for forwarding) 0 1: C-TAG mode (C-TAG used for forwarding) 1 0: SC-TAG mode (S-TAG used for forwarding) 1 1: Reserved	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

SVM[1:0] Bits (Switch VLAN Mode)

[Functions]

- For more details, refer to section 33, Ethernet Module.
- HW: If FWGC.SVM[1:0] are set to 00b, incoming packet PCP and DEI fields will be considered as 0 by MFWD.
- For more details, refer to section 34.3.1.2, ESWM Modes.

34.2.1.2 TAG TPID Configuration Register 0 (FWTTC0)

Address(es): MFWD.FWTTC0 874C 0010h, MFWD_NS.FWTTC0 074C 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STT[15:0]															
Value after reset:	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTT[15:0]															
Value after reset:	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CTT[15:0]	C-TAG TPID [801.2Q]	Configure the C-TAG TPID.	R/W
b31 to b16	STT[15:0]	S-TAG TPID [801.2Q]	Configure the S-TAG TPID.	R/W

CTT[15:0] Bits (C-TAG TPID [801.2Q])

[Restrictions]

- SW: The FWTTC0.CTT[15:0], FWTTC0.STT[15:0] and FWTTC1.RTT[15:0] bits should always be set to different values.
- SW: These bits cannot be set to 0000h to 05FFh, 0800h nor 86DDh

STT[15:0] Bits (S-TAG TPID [801.2Q])

[Restrictions]

- SW: The FWTTC0.CTT[15:0], FWTTC0.STT[15:0] and FWTTC1.RTT[15:0] bits should always be set to different values.
- SW: These bits cannot be set to 0000h to 05FFh, 0800h nor 86DDh

34.2.1.3 TAG TPID Configuration Register 1 (FWTTC1)

Address(es): MFWD.FWTTC1 874C 0014h, MFWD_NS.FWTTC1 074C 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RTT[15:0]															
Value after reset:	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RTT[15:0]	R-TAG TPID [801.2CB]	Configure the R-TAG TPID.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

RTT[15:0] Bits (R-TAG TPID [801.2CB])

[Restrictions]

- SW: The FWTTC0.CTT[15:0], FWTTC0.STT[15:0] and FWTTC1.RTT[15:0] bits should always be set to different values.
- SW: These bits cannot be set to 0000h to 05FFh, 0800h nor 86DDh

34.2.1.4 CPU Exceptional Path Target Configuration Register (FWCEPTC)

Address(es): MFWD.FWCEPTC 874C 0020h, MFWD_NS.FWCEPTC 074C 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	EPSL	—	—	—	—	—	—	—	EPCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EPIPV[2:0]			—	—	—	—	—	—	EPCSD[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	EPCSD[5:0]	Exceptional Path CPU Sub Destination	Configures the CPU sub destination to which exceptional frames will be sent.	R/W
b11 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	EPIPV[2:0]	Exceptional Path Internal Priority Value	Configures the priority to which exceptional frames will be sent.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	EPCS	Exceptional Path CPU Select	Configures CPU to which exceptional frames will be sent.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	EPSL	Exceptional Path Security Level	Configures descriptor security level for exceptional path.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

EPCS Bit (Exceptional Path CPU Select)

Refer to Table 34.25 for port mapping.

[Restriction]

- This bit should be set to 0.

34.2.1.5 CPU Exceptional Path Reason Configuration Register 0 (FWCEPRC0)

Address(es): MFWD.FWCEPRC0 874C 0024h, MFWD_NS.FWCEPRC0 074C 0024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	DDFSFEF	—	DDEEF	—	—	GDNEEF	GTFEF	—	GSEQEEF	GAXEEF	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ETFEF	EOEEF	EUEEF	EBOEEF	ERPOOEF	ERFFEF	—	—	—	EFCSEEF	ENIBEEF	—	EPHYEEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EPHYEEF	Ethernet PHY Error Exceptional Forwarding	0: Frames received with PHY Error are discarded. 1: Frames received with PHY Error are forwarded to exceptional path.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	ENIBEEF	Ethernet Nibble Error Exceptional Forwarding	0: Frames received with Nibble Error are discarded. 1: Frames received with Nibble Error are forwarded to exceptional path.	R/W
b3	EFCSEEF	Ethernet FCS Error Exceptional Forwarding	0: Frames received with FCS Error are discarded. 1: Frames received with FCS Error are forwarded to exceptional path.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	ERFFEF	Ethernet RMAC Frame Filtered Exceptional Forwarding	0: Frames received with RMAC Frame Filtered flag are discarded. 1: Frames received with RMAC Frame Filtered flag are forwarded to exceptional path.	R/W
b8	ERPOOEF	Ethernet Reception Partially Out of Operation Exceptional Forwarding	0: Frames received with Reception Partially Out of Operation flag are discarded. 1: Frames received with Reception Partially Out of Operation flag are forwarded to exceptional path.	R/W
b9	EBOEEF	Ethernet Buffer Overflow Error Exceptional Forwarding	0: Frames received with Buffer Overflow Error are discarded. 1: Frames received with Buffer Overflow Error are forwarded to exceptional path.	R/W
b10	EUEEF	Ethernet Undersize Error Exceptional Forwarding	0: Frames received with Undersize Error are discarded. 1: Frames received with Undersize Error are forwarded to exceptional path.	R/W
b11	EOEEF	Ethernet Oversize Error Exceptional Forwarding	0: Frames received with Oversize Error are discarded. 1: Frames received with Oversize Error are forwarded to exceptional path.	R/W
b12	ETFEF	Ethernet TAG Filtering Exceptional Forwarding	0: Frames received with TAG Filtering flag are discarded. 1: Frames received with TAG Filtering flag are forwarded to exceptional path.	R/W
b16 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17	GAXEEF	GWCA AXI Error Exceptional Forwarding	0: Frames received with AXI Error are discarded. 1: Frames received with AXI Error are forwarded to exceptional path.	R/W
b18	GSEQEEF	GWCA Sequence Error Exceptional Forwarding	0: Frames received with Sequence Error are discarded. 1: Frames received with Sequence Error are forwarded to exceptional path.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b20	GTFEF	GWCA TAG Filtering Exceptional Forwarding	0: Frames received with TAG Filtering flag are discarded. 1: Frames received with TAG Filtering flag are forwarded to exceptional path.	R/W
b21	GDNEEF	GWCA Descriptor Number Error Exceptional Forwarding	0: Frames received with Descriptor Number Error are discarded. 1: Frames received with Descriptor Number Error are forwarded to exceptional path.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	DDEEF	Direct Descriptor Error Exceptional Forwarding	0: Frames with Direct Descriptor Error are discarded. 1: Frames with Direct Descriptor Error are forwarded to exceptional path.	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26	DDFSFEF	Direct Descriptor Format Security Filtering Exceptional Forwarding	0: Frames with Direct Descriptor Format Security Error are discarded. 1: Frames with Direct Descriptor Format Security Error are forwarded to exceptional path.	R/W
b31 to b27	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: For Ethernet error/flag explanations, refer to section 36, Ethernet Agent (ETHA).
Note: For GWCA error/flag explanations, refer to section 38, Ethernet CPU Agent (GWCA).

34.2.1.6 CPU Exceptional Path Reason Configuration Register 2 (FWCEPRC2)

Address(es): MFWD.FWCEPRC2 874C 002Ch, MFWD_NS.FWCEPRC2 074C 002Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	FWMFEF	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FPBNTFEF	—	—	—	FDDNTFEF	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	FDDNTFEF	Direct Descriptor No Target Filtering Exceptional Forwarding	0: Frames filtered because no target during direct forwarding is detected are discarded. 1: Frames filtered because no target during direct forwarding is detected are forwarded to exceptional path.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	FPBNTFEF	Port Based No Target Filtering Exceptional Forwarding	0: Frames filtered because no target during Port Based forwarding is detected are discarded. 1: Frames filtered because no target during Port Based forwarding is detected are forwarded to exceptional path.	R/W
b25 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26	FWMFEF	Watermark Filtering Exceptional Forwarding	0: Frames filtered because of watermark are discarded. 1: Frames filtered because of watermark are forwarded to exceptional path.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

34.2.1.7 CPU Mirroring Path Target Configuration Register (FWCMPTC)

Address(es): MFWD.FWCMPTC 874C 0040h, MFWD_NS.FWCMPTC 074C 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	CMPSL	—	—	—	—	—	—	—	CMPCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPIPU	CMPIPV[2:0]			—	—	—	—	—	—	CMPCSD[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCSD[5:0]	CPU Mirroring Path CPU Sub Destination	Configures the CPU sub destination to which CPU mirroring frames will be sent.	R/W
b11 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	CMPIPV[2:0]	CPU Mirroring Path Internal Priority Value	CPU Mirroring Path Internal Priority Value	R/W

Bit	Symbol	Bit Name	Description	R/W
b15	CMPIPU	CPU Mirroring Path Internal Priority Update	0: Frames mirrored to CPU path are sent with the same priority as for forwarding. 1: Frames mirrored to CPU path are sent with FWCMPTC.CMPIPV[2:0] priority.	R/W
b16	CMPCS	CPU Mirroring Path CPU Select	Configures CPU to which CPU mirroring frames will be sent.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	CMPSL	CPU Mirroring Path Security Level	Configures descriptor security level for CPU mirroring path.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

CMPCS Bit (CPU Mirroring Path CPU Select)

Refer to Table 34.25 for port mapping.

[Restriction]

- Set this bit to 0.

34.2.1.8 Ethernet Mirroring Path Target Configuration Register (FWEMPTC)

Address(es): MFWD.FWEMPTC 874C 0044h, MFWD_NS.FWEMPTC 074C 0044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	EMPSL	—	—	—	—	—	—	—	EMPPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EMPIPU	EMPIPV[2:0]			—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	EMPIPV[2:0]	Ethernet Mirroring Path Internal Priority Value	Ethernet Mirroring Path Internal Priority Value	R/W
b15	EMPIPU	Ethernet Mirroring Path Internal Priority Update	0: Frames mirrored to Ethernet path are sent with the same priority as for forwarding. 1: Frames mirrored to Ethernet path are sent with FWEMPTC.EMPIPV[2:0] priority.	R/W
b16	EMPPS	Ethernet Mirroring Path Port Select	Configures port to which Ethernet mirroring frames will be sent.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	EMPSL	Ethernet Mirroring Path Security Level	Configures descriptor security level for Ethernet mirroring path.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

EMPPS Bit (Ethernet Mirroring Path Port Select)

Refer to Table 34.25 for port mapping.

[Restriction]

- Set this bit to 0.

34.2.1.9 Source-Destination Mirroring Path Target Configuration Register (FWSDMPTC)

Address(es): MFWD.FWSDMPTC 874C 0050h, MFWD_NS.FWSDMPTC 074C 0050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SDMPSL	—	—	—	—	—	—	SDMPPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SDMPIPU	SDMPIPV[2:0]			—	—	—	—	—	—	SDMPCSD[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	SDMPCSD[5:0]	Source-Destination Mirroring Path CPU Sub Destination	Configures the CPU sub destination to which Source-Destination mirroring frames will be sent.	R/W
b11 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b14 to b12	SDMPIPV[2:0]	Source-Destination Mirroring Path Internal Priority Value	Source-Destination Mirroring Path Internal Priority Value	R/W
b15	SDMPIPU	Source-Destination Mirroring Path Internal Priority Update	0: Frames mirrored to Source-Destination path are sent with the same priority as for forwarding. 1: Frames mirrored to Source-Destination path are sent with FWSDMPTC.SDMPIPV[2:0] priority.	R/W
b17, b16	SDMPPS[1:0]	Source-Destination Mirroring Path Port Select	Configures Port to which Source-Destination mirroring frames will be sent.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SDMPSL	Source-Destination Mirroring Path Security Level	Configures descriptor security level for Source-Destination mirroring path.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

SDMPPS[1:0] Bits (Source-Destination Mirroring Path Port Select)

Refer to Table 34.25 for port mapping.

[Restriction]

- SW: These bits should not be set to a value larger than 2.

34.2.1.10 Source-Destination Mirroring Path Vector Configuration Register (FWSDMPVC)

Address(es): MFWD.FWSDMPVC 874C 0054h, MFWD_NS.FWSDMPVC 074C 0054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SDMSV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SDMDV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SDMDV[2:0]	Source-Destination Mirroring Destination Vector		R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	SDMSV[2:0]	Source-Destination Mirroring Source Vector		R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

SDMDV[2:0] Bits (Source-Destination Mirroring Destination Vector)

Frames coming from a port set in the FWSDMPVC.SDMSV[2:0] bits and sent to a port set in the FWSDMPVC.SDMDV[2:0] bits are also duplicated to Source-Destination mirroring path.
Setting these bits to 000b disable Source-Destination mirroring path.
Set the SDMDV[1] bit to 0.

SDMSV[2:0] Bits (Source-Destination Mirroring Source Vector)

Frames coming from a port set in the FWSDMPVC.SDMSV[2:0] bits and sent to a port set in the FWSDMPVC.SDMDV[2:0] bits are also duplicated to Source-Destination mirroring path.
Setting these bits to 000b disable Source-Destination mirroring path.
Set the SDMSV[1] bit to 0.

34.2.1.11 Port i Level Based Watermark Configuration Register (FWLBWMCi) (i = 0, 2)

Address(es): MFWD.FWLBWMC0 874C 0080h, MFWD.FWLBWMC2 874C 0088h,
MFWD_NS.FWLBWMC0 074C 0080h, MFWD_NS.FWLBWMC2 074C 0088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WMFL PR15	WMFL PR14	WMFL PR13	WMFL PR12	WMFL PR11	WMFL PR10	WMFL PR9	WMFL PR8	WMFL PR7	WMFL PR6	WMFL PR5	WMFL PR4	WMFL PR3	WMFL PR2	WMFL PR1	WMFL PR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WMCL PR15	WMCL PR14	WMCL PR13	WMCL PR12	WMCL PR11	WMCL PR10	WMCL PR9	WMCL PR8	WMCL PR7	WMCL PR6	WMCL PR5	WMCL PR4	WMCL PR3	WMCL PR2	WMCL PR1	WMCL PR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WMCLPR0	Watermark Critical Level Priority 0 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 0 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 0 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b1	WMCLPR1	Watermark Critical Level Priority 1 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 1 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 1 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b2	WMCLPR2	Watermark Critical Level Priority 2 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 2 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 2 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b3	WMCLPR3	Watermark Critical Level Priority 3 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 3 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 3 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b4	WMCLPR4	Watermark Critical Level Priority 4 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 4 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 4 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b5	WMCLPR5	Watermark Critical Level Priority 5 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 5 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 5 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b6	WMCLPR6	Watermark Critical Level Priority 6 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 6 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 6 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W

Bit	Symbol	Bit Name	Description	R/W
b7	WMCLPR7	Watermark Critical Level Priority 7 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 7 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 7 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b8	WMCLPR8	Watermark Critical Level Priority 8 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 8 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 8 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b9	WMCLPR9	Watermark Critical Level Priority 9 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 9 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 9 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b10	WMCLPR10	Watermark Critical Level Priority 10 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 10 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 10 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b11	WMCLPR11	Watermark Critical Level Priority 11 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 11 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 11 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b12	WMCLPR12	Watermark Critical Level Priority 12 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 12 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 12 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b13	WMCLPR13	Watermark Critical Level Priority 13 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 13 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 13 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b14	WMCLPR14	Watermark Critical Level Priority 14 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 14 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 14 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b15	WMCLPR15	Watermark Critical Level Priority 15 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 15 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 15 are discarded when watermark critical level is set for port i (the CABPPWMLCi.PWMCL[9:0] bits are set [COMA]).	R/W
b16	WMFLPR0	Watermark Flush Level Priority 0 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 0 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 0 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W

Bit	Symbol	Bit Name	Description	R/W
b17	WMFLPR1	Watermark Flush Level Priority 1 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 1 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 1 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b18	WMFLPR2	Watermark Flush Level Priority 2 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 2 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 2 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b19	WMFLPR3	Watermark Flush Level Priority 3 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 3 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 3 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b20	WMFLPR4	Watermark Flush Level Priority 4 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 4 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 4 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b21	WMFLPR5	Watermark Flush Level Priority 5 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 5 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 5 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b22	WMFLPR6	Watermark Flush Level Priority 6 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 6 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 6 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b23	WMFLPR7	Watermark Flush Level Priority 7 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 7 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 7 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b24	WMFLPR8	Watermark Flush Level Priority 8 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 8 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 8 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b25	WMFLPR9	Watermark Flush Level Priority 9 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 9 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 9 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b26	WMFLPR10	Watermark Flush Level Priority 10 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 10 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 10 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W

Bit	Symbol	Bit Name	Description	R/W
b27	WMFLPR11	Watermark Flush Level Priority 11 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 11 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 11 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b28	WMFLPR12	Watermark Flush Level Priority 12 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 12 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 12 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b29	WMFLPR13	Watermark Flush Level Priority 13 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 13 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 13 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b30	WMFLPR14	Watermark Flush Level Priority 14 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 14 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 14 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W
b31	WMFLPR15	Watermark Flush Level Priority 15 Reject	0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 15 are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to 15 are discarded when watermark flush level is set for port i (the CABPPWMLCi.PWMFL[9:0] bits are set [COMA]).	R/W

Note: {DEI,IPV} refers to the bit association {DEI,IPV[2:0]}, so in decimal it can be written 8 × DEI + IPV.

34.2.1.12 Port i Configuration Register 1 (FWPCi1) (i = 0, 2)

Address(es): MFWD.FWPC01 874C 0104h, MFWD.FWPC21 874C 0124h,
MFWD_NS.FWPC01 074C 0104h, MFWD_NS.FWPC21 074C 0124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0*2

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSL	DDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DDE	Direct Descriptor Enable*3	0: Direct Descriptor disabled for port i 1: Direct Descriptor enabled for port i	R/W
b1	DDSL	Direct Descriptor Security Level*3	0: Secured Direct Descriptor discarded for port i. 1: Secured Direct Descriptor forwarded for port i.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. The value after reset is 1 for the FWPC21 register.
Note 2. The value after reset is 1 for the FWPC01 register.
Note 3. This bit is only valid for i = 2.

34.2.2 Port Based Forwarding Function Registers

34.2.2.1 Port Based Forwarding Configuration Register i (FWPBFCi) (i = 0, 2)

Address(es): MFWD.FWPBFC0 874C 4A00h, MFWD.FWPBFC2 874C 4A20h,
MFWD_NS.FWPBFC0 074C 4A00h, MFWD_NS.FWPBFC2 074C 4A20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	FAIFP	IP6PDE	IP4PDM	IP4PDE	PBSL	PBCME	PBEME	PBIPU	PBIPV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PBDV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PBDV[2:0]	Port Based Destination Vector	Destination vector for port i port-based forwarding. Set the PBDV[1] bit to 0.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	PBIPV[2:0]	Port Based Internal Priority Value	Internal priority value for port i port-based forwarding.	R/W
b19	PBIPU	Port Based Internal Priority Update	Enables frame priority update to internal priority value (FWPBFCi.PBIPV[2:0]) for port i.	R/W
b20	PBEME	Port Based Ethernet Mirroring Enabled	Enables Ethernet mirroring for port i.	R/W
b21	PBCME	Port Based CPU Mirroring Enabled	Enables CPU mirroring for port i.	R/W
b22	PBSL	Port Based Security Level	Set descriptor security level for port i.	R/W
b23	IP4PDE	IPv4 Priority Decode Enable	0: For port i, Input frame priority is taken from VLAN TAGs for port i. 1: For port i, If an IPv4 frame is detected, input frame priority is taken from precedence ToS. Else Input frame priority is taken from VLAN TAGs.	R/W
b24	IP4PDM	IPv4 Priority Decode Mode	0: For port i, Input frame priority is taken from precedence field in ToS. 1: For port i, Input frame priority is decoded from DSCP field in ToS.	R/W
b25	IP6PDE	IPv6 Priority Decode Enable	0: For port i, Input frame priority is taken from VLAN TAGs. 1: For port i, If an IPv6 frame is detected, input frame priority is decoded from DSCP field in ToS. Else Input frame priority is taken from VLAN TAGs.	R/W
b26	FAIFP	Force All Input Frame Priority Enable	0: For port i, Input frame priority is given by frame decoding. 1: For port i, Input frame priority is forced to FWPBFCi.PBIPV[2:0]	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

34.2.2.2 Port i Port Based Forwarding CSD Configuration Register 0 (FWPBFCSDCi0) (i = 0, 2)

Address(es): MFWD.FWPBFCSDC00 874C 4A04h, MFWD.FWPBFCSDC20 874C 4A24h, MFWD_NS.FWPBFCSDC00 074C 4A04h, MFWD_NS.FWPBFCSDC20 074C 4A24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PBCSD[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PBCSD[5:0]	Port Based CPU Sub Destination	CPU Sub Destination for GWCA0 for port i port-based forwarding.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

34.2.3 MFWD Counter Registers

34.2.3.1 Port 2 Direct Descriptor Forwarded Descriptor Counter Register (FWDDFDCN2)

Address(es): MFWD.FWDDFDCN2 874C 6340h, MFWD_NS.FWDDFDCN2 074C 6340h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DDFDN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DDFDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DDFDN[31:0]	Direct Descriptor Forwarded Descriptor Number	These bits count the number of descriptors forwarded by Direct Descriptor forwarding for port 2.	R

DDFDN[31:0] Bits (Direct Descriptor Forwarded Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a frame coming from port 2 is forwarded by Direct Descriptor Forwarding and if these bits have a value different than FFFF FFFFh.

34.2.3.2 Port i Port Based Forwarded Descriptor Counter Register (FWPBFDCNi) (i = 0, 2)

Address(es): MFWD.FWPBFDCN0 874C 6310h, MFWD.FWPBFDCN2 874C 6350h,
MFWD_NS.FWPBFDCN0 074C 6310h, MFWD_NS.FWPBFDCN2 074C 6350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PBFDN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PBFDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	PBFDN[31:0]	Port Based Forwarded Descriptor Number	These bits count the number of descriptors forwarded by Port Based forwarding for port i.	R

PBFDN[31:0] Bits (Port Based Forwarded Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a frame coming from port i is forwarded by Port Based Forwarding and if these bits have a value different than FFFF FFFFh.

34.2.3.3 Port i Watermark Rejected Descriptor Counter Register (FWWMRDCNi) (i = 0 to 2)

Address(es): MFWD.FWWMRDCN0 874C 6504h, MFWD.FWWMRDCN2 874C 6544h, MFWD_NS.FWWMRDCN0 074C 6504h, MFWD_NS.FWWMRDCN2 074C 6544h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WMRDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WMRDN[15:0]	Watermark Rejected Descriptor Number	These bits count the number of descriptors rejected by Watermark for port i.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

WMRDN[15:0] Bits (Watermark Rejected Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a frame coming from port i is rejected by Watermark and if these bits have a value different than FFFFh.

34.2.3.4 Port 2 Direct Descriptor Rejected Descriptor Counter Register (FWDDRDCN2)

Address(es): MFWD.FWDDRDCN2 874C 6548h, MFWD_NS.FWDDRDCN2 074C 6548h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DDRDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DDRDN[15:0]	Direct Descriptor Rejected Descriptor Number	These bits count the number of descriptors rejected by Direct Descriptor forwarding for port 2.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DDRDN[15:0] Bits (Direct Descriptor Rejected Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a frame coming from port 2 is rejected by Direct Descriptor forwarding and if these bits have a value different than FFFFh.

34.2.3.5 Port i Port Based Rejected Descriptor Counter Register (FWPBRDCNi) (i = 0, 2)

Address(es): MFWD.FWPBRDCN0 874C 6518h, MFWD.FWPBRDCN2 874C 6558h, MFWD_NS.FWPBRDCN0 074C 6518h, MFWD_NS.FWPBRDCN2 074C 6558h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PBRDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PBRDN[15:0]	Port Based Rejected Descriptor Number	These bits count the number of descriptors rejected by Port Based forwarding for port i.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

PBRDN[15:0] Bits (Port Based Rejected Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a frame coming from port i is rejected by Port Based forwarding and if these bits have a value different than FFFFh.

34.2.4 MFWD Interrupt Registers

34.2.4.1 Port i Error Interrupt Status Register 0 (FWEIS0i) (i = 0, 2)

Address(es): MFWD.FWEIS00 874C 7900h, MFWD.FWEIS02 874C 7920h,
MFWD_NS.FWEIS00 074C 7900h, MFWD_NS.FWEIS02 074C 7920h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DDNTFS	DDSES	—	DDES	WMIUFS	WMISFS	WMFFS	WMCFS	—	—	—	—	PBNTFS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17	PBNTFS	Port Based No Target Filtering Status Flag	0: No Port Based No Target Filtering 1: Port Based No Target Filtering occurred.	R/(W)
b21 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22	WMCFS	Watermark Critical Filtering Status Flag	0: No Watermark Critical Filtering 1: Watermark Critical Filtering occurred.	R/(W)
b23	WMFFS	Watermark Flush Filtering Status Flag	0: No Watermark Flush Filtering 1: Watermark Flush Filtering occurred.	R/(W)
b24	WMISFS	Watermark IPV Secure Filtering Status Flag	0: No Watermark IPV Secure Filtering 1: Watermark IPV Secure Filtering occurred.	R/(W)
b25	WMIUFS	Watermark IPV Unsecure Filtering Status Flag	0: No Watermark IPV Unsecure Filtering 1: Watermark IPV Unsecure Filtering occurred.	R/(W)
b26	DDES	Direct Descriptor Error Status Flag	0: No Direct Descriptor Error 1: Direct Descriptor Error occurred.	R/(W)
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	DDSES	Direct Descriptor Security Error Status Flag	0: No Direct Descriptor Security Error 1: Direct Descriptor Security Error occurred.	R/(W)
b29	DDNTFS	Direct Descriptor No Target Filtering Status Flag	0: No Direct Descriptor No Target Filtering 1: Direct Descriptor No Target Filtering occurred.	R/(W)
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

PBNTFS Flag (Port Based No Target Filtering Status Flag)

[Setting Conditions]

- HW: When forwarding flow for a frame coming from port i goes until “Port based forwarding secure error” block in Figure 34.9 and FWPBFCi.PBDV[2:0] is null.
- HW: When forwarding flow for a frame coming from port i goes until “Port based forwarding unsecure error” block in Figure 34.10 and FWPBFCi.PBDV[2:0] is null.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

WMCFS Flag (Watermark Critical Filtering Status Flag)

[Setting Condition]

- HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and the FWLBWMCi.WMCLPRi bit {DEI,IPV} is set and coa_watermark_critical[i] pin is set (where DEI is set to input frame C-TAG DEI if the FWGC.SVM[1:0] bits are set to 01b, input frame S-TAG DEI if the FWGC.SVM[1:0] bits are set to 10b and set to 0 if the FWGC.SVM[1:0] bits are set to 00b and IPV is the Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

WMFFS Flag (Watermark Flush Filtering Status Flag)

[Setting Condition]

- HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and the FWLBWMCi.WMFLPRi bit {DEI,IPV} is set and coa_watermark_flush[i] pin is set (where DEI is set to input frame C-TAG DEI if the FWGC.SVM[1:0] bits are set to 01b, input frame S-TAG DEI if the FWGC.SVM[1:0] bits are set to 10b and set to 0 if the FWGC.SVM[1:0] bits are set to 00b and IPV is the Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

WMISFS Flag (Watermark IPV Secure Filtering Status Flag)

[Setting Condition]

- HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and corresponding IPV is rejected by IPV based unsecure watermark [COMA] (where IPV is Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value before filtering).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

WMIUFS Flag (Watermark IPV Unsecure Filtering Status Flag)

[Setting Condition]

- HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and SEC is not set and corresponding IPV is rejected by IPV based secure watermark [COMA] (where SEC and IPV are respectively Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.SEC and FDESCR.IPV values before filtering).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

DDES Flag (Direct Descriptor Error Status Flag)

[Setting Condition]

- HW: When a direct descriptor (LDESCR.FMT [GWCA] set to 1) is received from port 2 + i and FWPCi1.DDE is not set.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

[Restriction]

- HW: This flag is only valid for i = 2.

DDSES Flag (Direct Descriptor Security Error Status Flag)

[Setting Condition]

- HW: When a secure direct descriptor (LDESCR.FMT [GWCA] and LDESCR.SEC [GWCA] are both set to 1) is received from port 2 + i, FWPCi1.DDE is set and FWPCi1.DDSL is not set.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

[Restriction]

- HW: This flag is only valid for i = 2.

DDNTFS Flag (Direct Descriptor No Target Filtering Status Flag)

[Setting Condition]

- HW: HW: When a directed descriptor (LDESCR.FMT [GWCA] set to 1) is received from port 2 + i and LDESCR.DV [GWCA] is set to ALL0.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- HW: Descriptor corresponding frame is rejected.

[Restriction]

- HW: This flag is only valid for i = 2.

34.2.4.2 Port i Error Interrupt Enable Register 0 (FWEIE0i) (i = 0, 2)

Address(es): MFWD.FWEIE00 874C 7904h, MFWD.FWEIE02 874C 7924h,
MFWD_NS.FWEIE00 074C 7904h, MFWD_NS.FWEIE02 074C 7924h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DDNTFE	DDSEE	—	DDEE	WMIUFE	WMISFE	WMFFE	WMCFE	—	—	—	—	PBNTFE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17	PBNTFE	Port Based No Target Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22	WMCFE	Watermark Critical Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	WMFFE	Watermark Flush Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	WMISFE	Watermark IPV Secure Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	WMIUFE	Watermark IPV Unsecure Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b26	DDEE	Direct Descriptor Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27	—	Reserved	This bit is read as 0.	R
b28	DDSEE	Direct Descriptor Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b29	DDNTFE	Direct Descriptor No Target Filtering Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the FWEID0i register.

PBNTFE Bit (Port Based No Target Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.PBNTFD bit will clear this bit.

WMCFE Bit (Watermark Critical Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.WMCFD bit will clear this bit.

WMFFE Bit (Watermark Flush Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.WMFFD bit will clear this bit.

WMISFE Bit (Watermark IPV Secure Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.WMISFD bit will clear this bit.

WMIUFE Bit (Watermark IPV Unsecure Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.WMIUFD bit will clear this bit.

DDEE Bit (Direct Descriptor Error Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.DDED bit will clear this bit.

DDSEE Bit (Direct Descriptor Security Error Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.DDSED bit will clear this bit.

DDNTFE Bit (Direct Descriptor No Target Filtering Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the FWEID0i.DDNTFD bit will clear this bit.

34.2.4.3 Port i Error Interrupt Disable Register 0 (FWEID0i) (i = 0, 2)

Address(es): MFWD.FWEID00 874C 7908h, MFWD.FWEID02 874C 7928h,
MFWD_NS.FWEID00 074C 7908h, MFWD_NS.FWEID02 074C 7928h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DDNTFD	DDSED	—	DDED	WMIUFD	WMISFD	WMFFD	WMCFD	—	—	—	—	PBNTFD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17	PBNTFD	Port Based No Target Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.PBNTFE bit to 0.	R/W
b21 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22	WMCFD	Watermark Critical Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.WMCFE bit to 0.	R/W
b23	WMFFD	Watermark Flush Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.WMFFE bit to 0.	R/W
b24	WMISFD	Watermark IPV Secure Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.WMISFE bit to 0.	R/W
b25	WMIUFD	Watermark IPV Unsecure Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.WMIUFE bit to 0.	R/W
b26	DDED	Direct Descriptor Error Disable	0: Has no effects. 1: Sets FWEIE0i.DDEE bit to 0.	R/W
b27	—	Reserved	This bit is read as 0.	R
b28	DDSED	Direct Descriptor Security Error Disable	0: Has no effects. 1: Sets FWEIE0i.DDSEE bit to 0.	R/W
b29	DDNTFD	Direct Descriptor No Target Filtering Disable	0: Has no effects. 1: Sets FWEIE0i.DDNTFE bit to 0.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

34.3 Operations

34.3.1 General Functionalities

General functionalities are separated in several types:

- Descriptors
- ESWM modes.
- ESWM TPIDs.
- ESWM debug functions.
- ESWM watermark.

34.3.1.1 Descriptors

To forward a frame to one or several port, the MFWD emits descriptors through the descriptor bus. All descriptor emitted by the MFWD shall follow the descriptor format described in Table 34.5. In following sections the format when a function emits a descriptor will be explained using the field name. Only fields that are not determined by the input local descriptor or the input frame will be explained.

Table 34.5 Descriptor Field

Field name FDESCR.	Bit width	Field explanation	Values
DV	3	Bit i set to 0: Descriptor not valid for port i Bit i set to 1: Descriptor valid for port i	Depends on descriptor. Will be explained per descriptor in following sections.
SEC	1	0: Descriptor is unsecure 1: Descriptor is secure	Depends on descriptor. Will be explained per descriptor in following sections.
CSD	6	FDESCR.CSD0 defines the CPU Sub destination for GWCA0 (CPU Sub destination is equivalent to AXI descriptor queue number)	Depends on descriptor. Will be explained per descriptor in following sections.
IPV	3	Internal priority value before agent internal priority remapping	Depends on descriptor. Will be explained per descriptor in following sections.
SPN	2	Port number from which the frame came from	Set to the source agent port number given by mfw_spn_fwd pin
TPL	16	Input frame total payload length.	Set to the TPL value given by agent through local descriptors
MINFO	24	Meta-information associated to a frame for CPU forwarding	Depends on descriptor. Will be explained per descriptor in following sections.
RV	1	Routing valid. Informs agents that routing rules should be fetched to MFWD through L23 update bus to update frame information	Depends on descriptor. Will be explained per descriptor in following sections.
RN	8	Routing number. Informs agents that routing number should be used when FDESCR.RV is set	Depends on descriptor. Will be explained per descriptor in following sections.
VCTRL	3	VLAN control information.	Set to the VCTRL value given by agent through local descriptors
RTGI	1	R-TAG in	Set to the RTGI value given by agent through local descriptors
SEQN	16	Frame sequence number	Depends on descriptor. Will be explained per descriptor in following sections.
FI	1	0: Input frame does not contain its FCS 1: Input frame contains its FCS	Set to the FI value given by agent through local descriptors
FW	1	0: If input frame contains its FCS, it was correct when given to the switch 1: If input frame contains its FCS, it was incorrect when given to the switch	Set to the FW value given by agent through local descriptors

34.3.1.2 ESWM Modes

ESWM modes define the general behavior of the ESWM module using FWGC registers.
[Function]

- FWGC register sets the VLAN format that will be used by the MFWD. Depending on this setting all the agents should convert Ethernet frames in the required format (for format conversion refer to section 38, Ethernet CPU Agent (GWCA)).

34.3.1.3 ESWM TPIDs

ESWM TPIDs for C-TAG, S-TAG and R-TAG [801.2Q] [801.2CB] can be set by the user using FWTTC0 and FWTTC1 registers.

[Functions]

- FWTTC0 register sets the S-TAG and C-TAG TPIDs [801.2Q].
- FWTTC1 register sets the R-TAG TPID [801.2CB].
- FWTTC0 and FWTTC1 registers reset values are as per defined in IEEE standards [801.2Q] [801.2CB].

34.3.1.4 ESWM Frame Input Priority

Any incoming frame will have an input priority given by MFWD depending on switch mode FWGC.SVM[1:0], and the incoming priority settings in FWPBFCi register and the frame format. Figure 34.3 describes frame input priority assignment for a frame coming from port i (the ToS DSCP field decoding is fixed in hardware and described in Table 34.6).

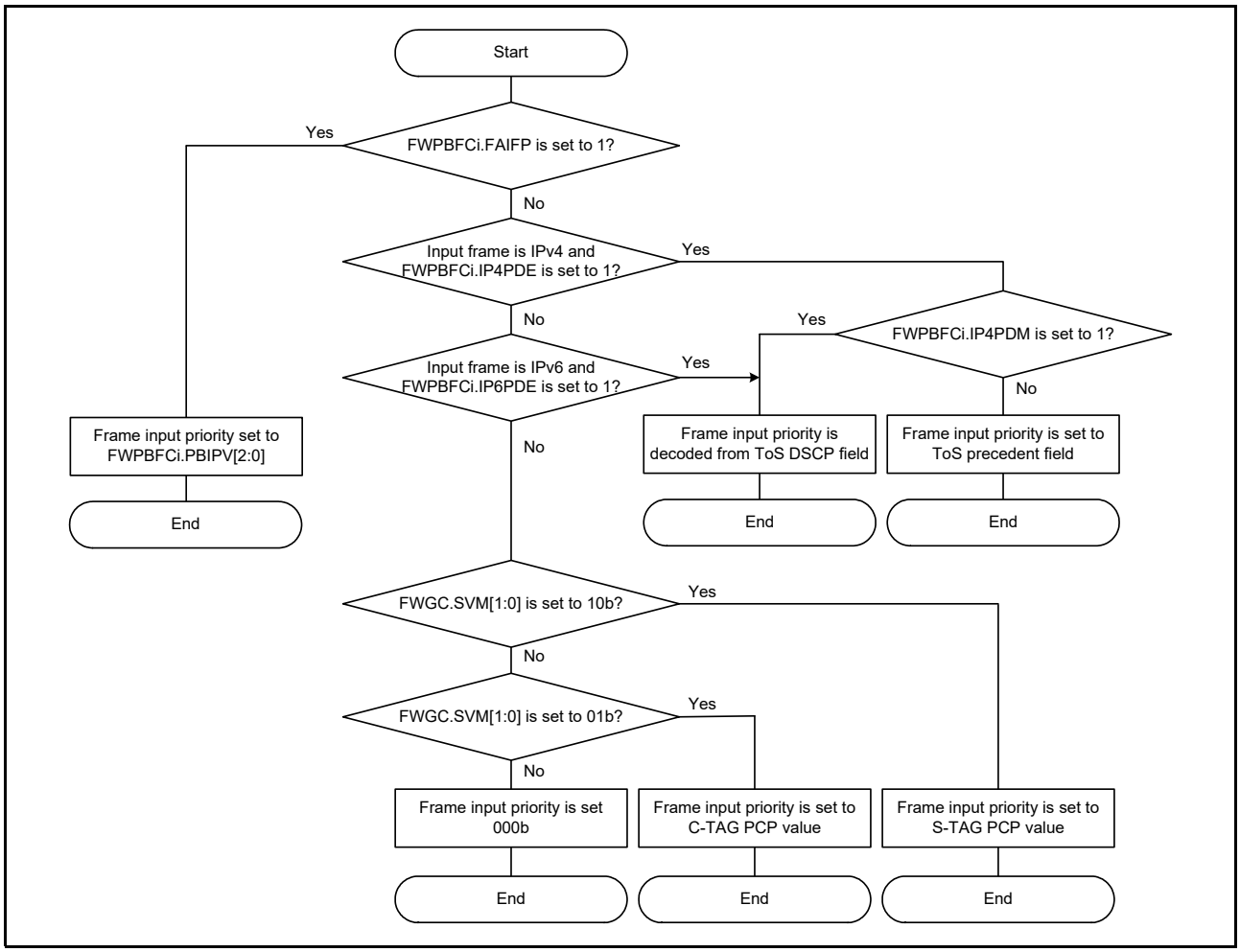


Figure 34.3 Frame Input Priority Assignment for Port i

[Restriction]

- HW: For cut-through forwarding, ToS and DSCP are not considered as frame input priority. Frame input priority is always extracted from S-TAG or set to 0.

Table 34.6 DSCP to Priority Mapping

DSCP values in decimal	Frame input priority
56	7
48	6
40, 46	5
32, 34, 36, 38	4
24, 26, 28, 30	3
16, 18, 20, 22	2
8, 10, 12, 14	1
Others	0

34.3.1.5 ESWM Debug Functions

(1) CPU Exceptional Path

Exceptional path is used to forward faulty frames to CPU for diagnosis using FWCEPTC, FWCEPRC0, FWCEPRC1, and FWCEPRC2 registers. When a frame is selected by exceptional path an exceptional descriptor is emitted.

[Functions]

- FWCEPTC register sets the target to which type of faulty frames will be forwarded.
- FWCEPRC0, FWCEPRC1, and FWCEPRC2 registers set which error frame will be forwarded to the exceptional path.

Note: For exceptional path descriptor formats and associated error details, refer to section 34.3.2.1 (3) Direct Exceptional Descriptor and section 34.3.2.2 (3) Port Based Exceptional Descriptor.

(2) Mirroring Paths

MFWD contains 3 mirroring paths:

- CPU mirroring path
- Ethernet mirroring path
- Source-destination mirroring path

(a) CPU Mirroring Path

CPU Mirroring path is used to copy frames to a CPU for monitoring purpose using FWCMPTC register. Table 34.7 describes CPU mirroring selection conditions depending on the forwarding mechanism selected by forwarding flow. When a frame is selected for CPU mirroring, a mirroring descriptor is emitted (refer to section 34.3.1.5 (2) (d) Mirroring Descriptor).

[Function]

- FWCMPTC register sets the target to which mirrored frames will be forwarded.

Table 34.7 CPU Mirroring Selection Conditions

Descriptor selected by forwarding flow	Mirroring selection conditions
Port based normal descriptor (Figure 34.9, Figure 34.10)	If FWPBFCi.PBCME is set for incoming frame source port.

(b) Ethernet Mirroring Path

Ethernet Mirroring path is used to copy frames to an Ethernet agent for monitoring purpose using FWEMPTC register. Table 34.8 describes Ethernet mirroring selection conditions depending on the forwarding mechanism selected by forwarding flow. When a frame is selected for Ethernet mirroring, a mirroring descriptor is emitted (refer to section 34.3.1.5 (2) (d) Mirroring Descriptor).

[Function]

- FWEMPTC register sets the target to which mirrored frames will be forwarded.

Table 34.8 Ethernet Mirroring Selection Conditions

Descriptor selected by forwarding flow	Mirroring selection conditions
Port based normal descriptor (Figure 34.9, Figure 34.10)	If FWPBFCi.PBEME is set for incoming frame source port.

(c) Source-Destination Mirroring Path

Mirroring path is used to copy frames to an Ethernet port or a CPU for monitoring purpose using FWSDMPTC and FWSDMPVC registers. When a frame is selected for Source-destination mirroring, a mirroring descriptor is emitted (refer to section 34.3.1.5 (2) (d) Mirroring Descriptor).

[Functions]

- FWSDMPTC register sets the target to which mirrored frames will be forwarded.
- FWSDMPVC register sets the source from which should come and the destination where should go a frame to be mirrored.
- A frame is selected for source-destination mirroring if the descriptor selected by forwarding flow is Port based normal descriptor (Figure 34.9, Figure 34.10)}, if the frame source port is set in FWSDMPVC.SDMSV[2:0] bits and if FDESCR.DV & FWSDMPVC.SDMDV[2:0] is not 000b.

(d) Mirroring Descriptor

Mirroring descriptor fields are described in Table 34.9.

Table 34.9 Mirroring Descriptor Fields

Field name FDESCR.	Values for CPU mirroring	Values for Ethernet mirroring	Values for Source-destination mirroring
DV	Mirroring descriptors are forwarded to GWCA number FWCMPTC.CMPCS (port number 2 + FWCMPTC.CMPCS)	Mirroring descriptors are forwarded to ETHA number FWEMPTC.EMPPS (port number FWEMPTC.EMPPS)	Mirroring descriptors are forwarded to port number FWSDMPTC.SDMPPS[1:0]
SEC	FWCMPTC.CMPSL	FWEMPTC.EMPSL	FWSDMPTC.SDMPSL
CSDj	FWCMPTC.CMPCSD[5:0]	ALL0	ALL0
IPV	If FWCMPTC.CMPIPU is set, FWCMPTC.CMPIPV[2:0] If FWCMPTC.CMPIPU is not set, same IPV as normal descriptor	If FWEMPTC.EMPIPU is set, FWEMPTC.EMPIPV[2:0] If FWEMPTC.EMPIPU is not set, same IPV as normal descriptor	If FWSDMPTC.SDMPIPU is set, FWSDMPTC.SDMPIPV[2:0] If FWSDMPTC.SDMPIPU is not set, same IPV as normal descriptor
MINFO	Refer to Figure 34.4 and Table 34.10.	Refer to Figure 34.4 and Table 34.10.	Refer to Figure 34.4 and Table 34.10.
RV	0	0	0
RN	ALL0	ALL0	ALL0
SEQN	Set to corresponding normal descriptor sequence number	Set to corresponding normal descriptor sequence number	Set to corresponding normal descriptor sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	NDFB[7:0]							
2	RSV							

Figure 34.4 Mirroring Descriptor MINFO Format

Table 34.10 Mirroring Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 14 for Mirroring descriptors
NDFB	8	Normal Descriptor First Byte	Set to mirrored frame corresponding normal descriptor FDESCR.MINFO[7:0] value

34.3.1.6 Watermark

ESWM watermark defines the switch behavior in case of overflow using FWLBWMCi and FWCEPRC2 registers and can be monitored using FWEIS0i register.

[Functions]

- FWLBWMCi is used to set which frame should be discarded by level-based watermark [COMA].
- FWCEPRC2 register enables exceptional path for error descriptors.
- FWEIS0i register notifies that a frame coming from port i has been rejected because of watermark.
- Watermark uses as frame priority the IPV value given by forwarding mechanism before filtering. This correspond to the FDESCR.IPV in the normal descriptor emitted by the selected forwarding.
- Watermark uses as DEI value the value extracted from the input frame which is the S-TAG DEI value when FWGC.SVM[1:0] is set to 10b, the C-TAG DEI value when FWGC.SVM[1:0] is set to 01b, and 0 when FWGC.SVM[1:0] is set to 00b.

[Caution]

- SW: The watermark is an indispensable functionality for switch QoS behavior definition, it is not advised to disable it.

Note: Error frames are not going through watermark.

(1) Watermark Errors

Table 34.11 describes the Watermark errors. Instead of being discarded by MFWD, an error frame can be forwarded to exceptional path using Watermark exceptional descriptor described in section 34.3.1.6 (2) Watermark Exceptional Descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 34.11 Watermark Errors

Error	Error name	Error interrupt	Error exceptional bit
WMCF	WaterMark Critical Filtering	FWEIS0i.WMCFS	FWCEPRC2.FWMFEF
WMFF	WaterMark Flush Filtering	FWEIS0i.WMFFS	FWCEPRC2.FWMFEF
WMISF	WaterMark IPV Secure Filtering	FWEIS0i.WMISFS	FWCEPRC2.FWMFEF
WMIUF	WaterMark IPV Unsecure Filtering	FWEIS0i.WMIUFS	FWCEPRC2.FWMFEF

(2) Watermark Exceptional Descriptor

Watermark exceptional descriptor fields are described in Table 34.12.

Table 34.12 Watermark Exceptional Descriptor Fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD[5:0]
IPV	FWCEPTC.EPIPV[2:0]
MINFO	Refer to Figure 34.5 and Table 34.13.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	RSV				WMIUF	WMISF	WMFF	WMCF
2	RSV							

Figure 34.5 Watermark Exceptional Descriptor MINFO Format

Table 34.13 Watermark Exceptional Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 4 for Watermark exceptional descriptors
WMCF	1	WaterMark Critical Filtering	Refer to section 34.3.1.6 (1) Watermark Errors.
WMFF	1	WaterMark Flush Filtering	Refer to section 34.3.1.6 (1) Watermark Errors.
WMISF	1	WaterMark IPV Secure Filtering	Refer to section 34.3.1.6 (1) Watermark Errors.
WMIUF	1	WaterMark IPV Unsecure Filtering	Refer to section 34.3.1.6 (1) Watermark Errors.

34.3.2 Forwarding/Routing

MFWD contains different types of forwarding which are:

- Direct descriptor forwarding
- Port based forwarding

34.3.2.1 Direct Descriptor Forwarding

Direct descriptor forwarding is used to forward local direct descriptors (LDESCR.FMT [GWCA] set to 1) received from a slow interface [GWCA] using FWPCi1 and FWCEPRC0 registers and can be monitored using FWEIS0i register. Direct descriptor forwarding flow is described in Figure 34.6.

[Functions]

- FWPCi1.DDE bit is used to enable direct descriptor reception from port i.
- FWPCi1.DDSL bit is used to enable secure direct descriptor reception from port i when direct descriptor reception from port i is enabled.
- FWCEPRC0 register enables exceptional path for error descriptors.
- FWEIS0i register notifies that a direct descriptor has been rejected.

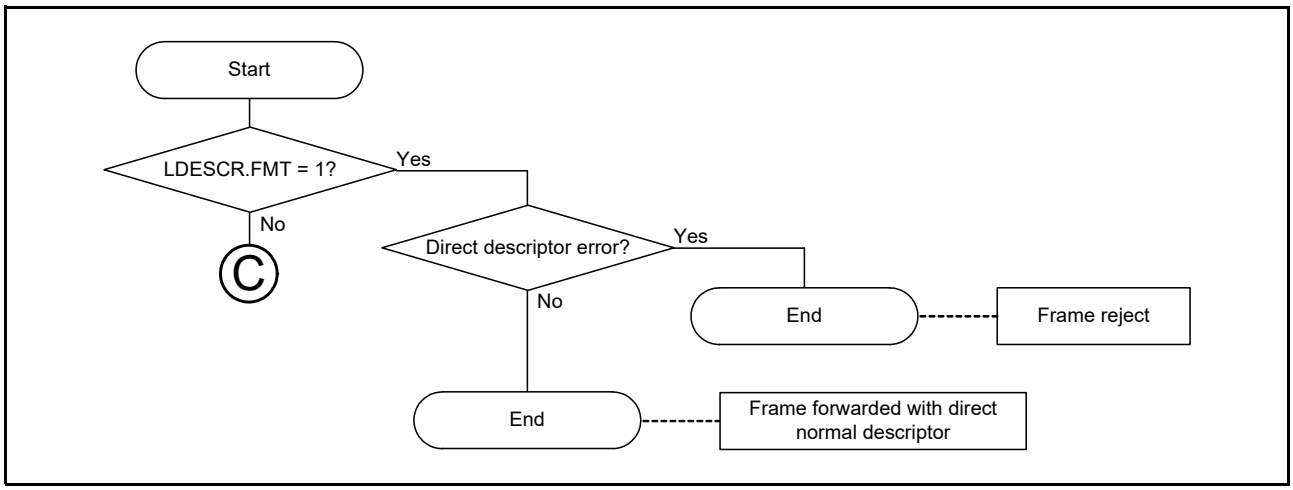


Figure 34.6 Direct Descriptor Forwarding Flow

Note: Direct descriptor errors are described in section 34.3.2.1 (1) Direct Descriptor Forwarding Errors.
Note: Direct normal descriptor format is described in section 34.3.2.1 (2) Direct Normal Descriptor.
Note: Link C links to Figure 34.41.

(1) Direct Descriptor Forwarding Errors

Table 34.14 describes the Direct descriptor forwarding errors. Instead of being discarded by MFWD, an error frame can be forwarded to exceptional path using Direct exceptional descriptor described in section 34.3.2.1 (3) Direct Exceptional Descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 34.14 Direct Descriptor Errors

Error	Error name	Error interrupt	Error exceptional bit
DDE	Direct Descriptor Error	FWEIS0i.DDES	FWCEPRC0.DDEEF
DDSF	Direct Descriptor Security Filtering	FWEIS0i.DDSES	FWCEPRC0.DDFSFEF
NTF	No Target Filtering	FWEIS0i.DDNTFS	FWCEPRC2.DDNTFEF

(2) Direct Normal Descriptor

Direct normal descriptor fields are described in Table 34.15. For LDESCR explanation, refer to section 38, Ethernet CPU Agent (GWCA).

Table 34.15 Direct Normal Descriptor Fields

Field name FDESCR.	Values
DV	LDESCR.DV
SEC	LDESCR.SEC
CSD	LDESCR.CSD
IPV	LDESCR.IPV
MINFO	Refer to Figure 34.7 and Table 34.16.
RV	LDESCR.RV
RN	If LDESCR.RV is set, set to LDESCR.RN If LDESCR.RV is not set, set to 0
SEQN	Set to frame input sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	RSV							
2								

Figure 34.7 Direct Normal Descriptor MINFO Format

Table 34.16 Direct Normal Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 2 for Direct normal descriptors
RSV	—	Reserved field	Set to 0

(3) Direct Exceptional Descriptor

Direct exceptional descriptor fields are described in Table 34.17.

Table 34.17 Direct Exceptional Descriptor Fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD[5:0]
IPV	FWCEPTC.EPIPV[2:0]
MINFO	Refer to Figure 34.8 and Table 34.18.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	RSV				NTF	DDFF	DDSF	DDF
2	RSV							

Figure 34.8 Direct Exceptional Descriptor MINFO Format

Table 34.18 Direct Exceptional Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 3 for Direct exceptional descriptors
DDF	1	Direct Descriptor Filtered	Refer to section 34.3.2.1 (1) Direct Descriptor Forwarding Errors.
DDSF	1	Direct Descriptor Security Filtered	Refer to section 34.3.2.1 (1) Direct Descriptor Forwarding Errors.
DDFF	1	Direct Descriptor Format Filtered	Refer to section 34.3.2.1 (1) Direct Descriptor Forwarding Errors.
NTF	1	No Target Filtered	Refer to section 34.3.2.1 (1) Direct Descriptor Forwarding Errors.

34.3.2.2 Port Based Forwarding

Port based forwarding aims at forwarding all frames from a port to other ones. It used FWPBFCi and FWPBFCSDCi0 registers. Port based forwarding will forward frames following Port based forwarding secure flow in Figure 34.9 and Port based forwarding unsecure flow in Figure 34.10.

[Functions]

- FWPBFCi and FWPBFCSDCi0 registers are used to set forwarding information for frames coming from port i.

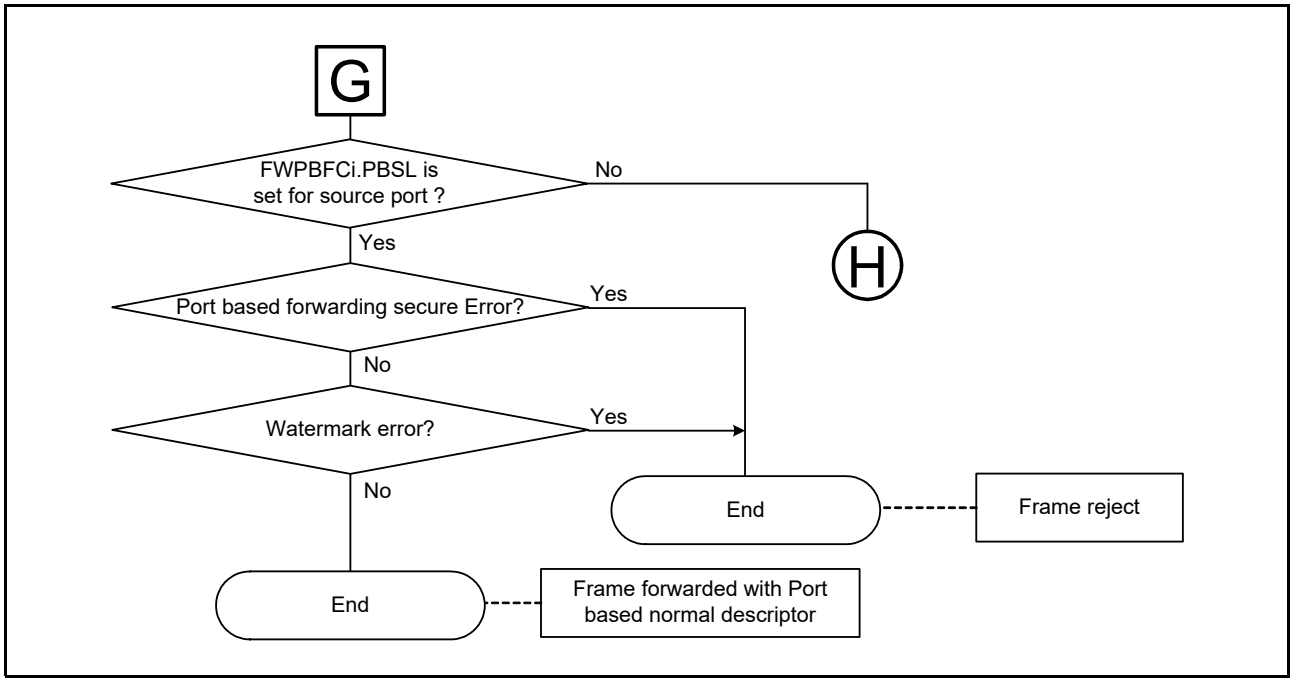


Figure 34.9 Port Based Forwarding Secure Flow

Note: Port based forwarding secure errors are described in section 34.3.2.2 (1) Port Based Forwarding Errors.
Note: Watermark errors are described in section 34.3.1.6 (1) Watermark Errors.
Note: Link G is linked from Figure 34.51.
Note: Link H links to Figure 34.42.

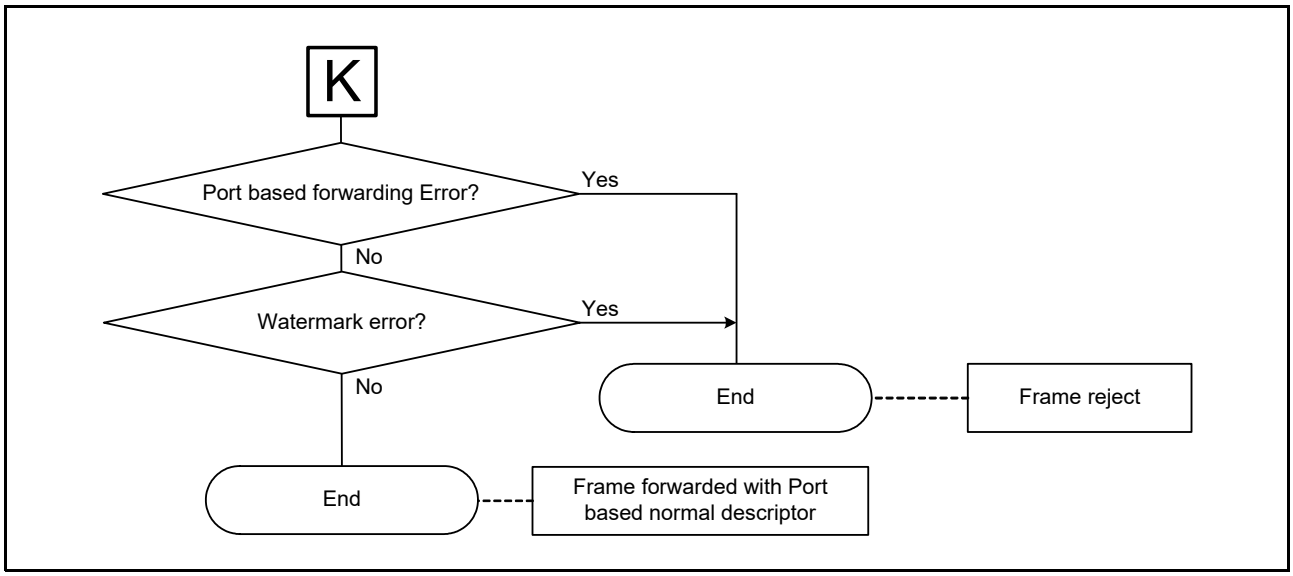


Figure 34.10 Port Based Forwarding Unsecure Flow

Note: Port based forwarding errors are described in section 34.3.2.2 (1) Port Based Forwarding Errors.
Note: Watermark errors are described in section 34.3.1.6 (1) Watermark Errors.
Note: Link G is linked from Figure 34.52.

(1) Port Based Forwarding Errors

Table 34.19 describes the Port based forwarding errors. Instead of being discarded by MFWD, an error frame can be forwarded to exceptional path using Port based exceptional descriptor described in section 34.3.2.2 (3) Port Based Exceptional Descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 34.19 Port Based Forwarding Errors

Error	Error name	Error interrupt	Error exceptional bit
Port based forwarding secure errors			
SNTF	Secure No Target Filtering	FWEIS0i.PBNTFS	FWCEPRC2.FPBNTFEF
Port based forwarding errors			
NTF	No Target Filtering	FWEIS0i.PBNTFS	FWCEPRC2.FPBNTFEF

(2) Port Based Normal Descriptor

Port based normal descriptor fields are described in Table 34.20.

Table 34.20 Port Based Normal Descriptor Fields

Field name FDESCR.	Values
DV	FWPBFCi.PBDV[2:0]
SEC	FWPBFCi.PBSL
CSDj	FWPBFCSDCi0.PBCSD[5:0]
IPV	If FWPBFCi.PBIPU is set, set to FWPBFCi.PBIPV[2:0] If FWPBFCi.PBIPU is not set, set to the frame input priority (refer to section 34.3.1.4, ESWM Frame Input Priority)
MINFO	Refer to Figure 34.11 and Table 34.21.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	RSV							
2								

Figure 34.11 Port Based Normal Descriptor MINFO Format

Table 34.21 Port Based Normal Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 11 for Port based normal descriptors
RSV	—	Reserved field	Set to 0

(3) Port Based Exceptional Descriptor

Port based exceptional descriptor fields are described in Table 34.22.

Table 34.22 Port Based Exceptional Descriptor Fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD[5:0]
IPV	FWCEPTC.EPIPV[2:0]
MINFO	Refer to Figure 34.12 and Table 34.23.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RSV				FWDC[3:0]			
1	RSV						NTF	SNTF
2	RSV							

Figure 34.12 Port Based Exceptional Descriptor MINFO Format

Table 34.23 Port Based Exceptional Descriptor MINFO Field Explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allows to identify the type of forwarding	Fixed to 12 for Port based exceptional descriptors
SNTF	1	Secure No Target Filtering	Refer to section 34.3.2.2 (1) Port Based Forwarding Errors.
NTF	1	No Target Filtering	Refer to section 34.3.2.2 (1) Port Based Forwarding Errors.

34.3.3 Data Representation in RAMs

Agents connected to Fabric will use TAG, Pointer and Data RAM to store their data. Because of peripheral IPs such as MFWD, the data saving/fetching patterns described in this section should be respected by Agents.

34.3.3.1 RAM Address Dependencies

When an agent access to Fabric, it can decide to access to one, two or all the RAMs in one access (thanks to*_den_*,*_pen_*and*_ten_*signals) but since it can only give one address. The address at which RAMs will be accessed will be calculated from the address given to the Fabric. The address calculation for Data, TAG and Pointer RAMs are respectively described in equations (1), (2) and (3). Figure 34.13 describes the first addresses mapping to RAMs.

(1) *dataAddress = fabricAddress*
(2) *tagAddress = fabricAddress[11:3]*
(3) *pointerAddress = fabricAddress[11:3]*

[Where]

- *fabricAddress* is the address given by any agent to Fabric.

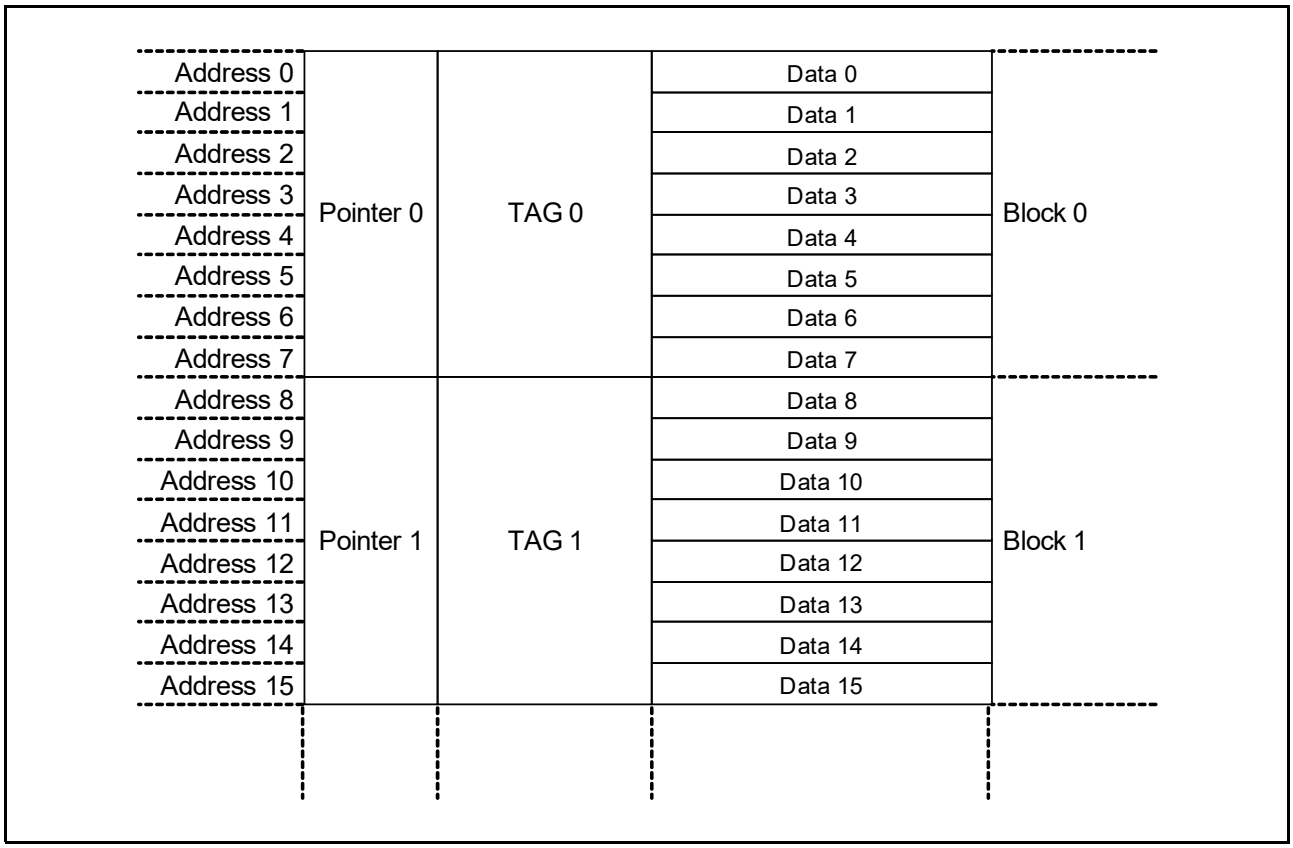


Figure 34.13 RAM First Address Representation

34.3.3.2 RAM Access by Agents

While written in RAMs, the Agents should respect a given structure defined by the following rules and described in Figure 34.14, Figure 34.15 and Figure 34.16.

[Restrictions]

- HW: A frame data should be written in a block in order and from smallest address to biggest address.
- HW: When a frame is contained in several blocks, the next block pointer should be written in the previous block pointer RAM to create a block chain.
- HW: In any block containing the first part of a frame, the frame data should be written from beat 1 address. The beat 0 address is reserved to write the descriptor.
- HW: While writing in a frame first block, the TAGs should be written with the first beat of data.
- HW: A pointer should always been written/read while accessing a block for the first time if needed.
- HW: The descriptor should always be written last and read first.
- HW: Type signals should be constant inside a frame. Value 0 is used for E-frames and 1 for P-frames.
- HW: Size signal can only be used for the last beat of data.
- HW: Cmd signals is used to signal what is being written to the RAMs. Value 0 is used for start-of-frame, Value 2 for end-of-frame and Value 3 for descriptor. All other accesses are done with value 1.

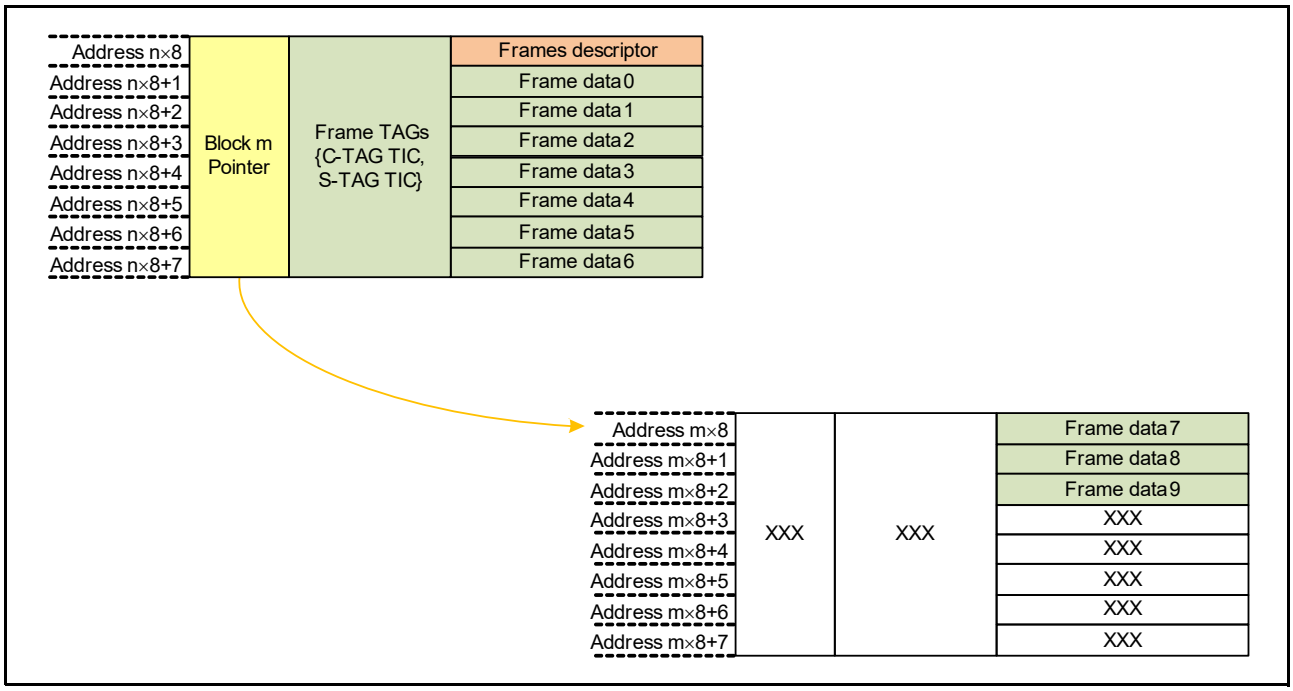


Figure 34.14 Fabric Data Structure

Note: Only S-TAGs and C-TAGs will be saved in the TAG RAM. The R-TAG will be taken by MFWD and saved in the agent descriptor RAM [GWCA][ETHA].

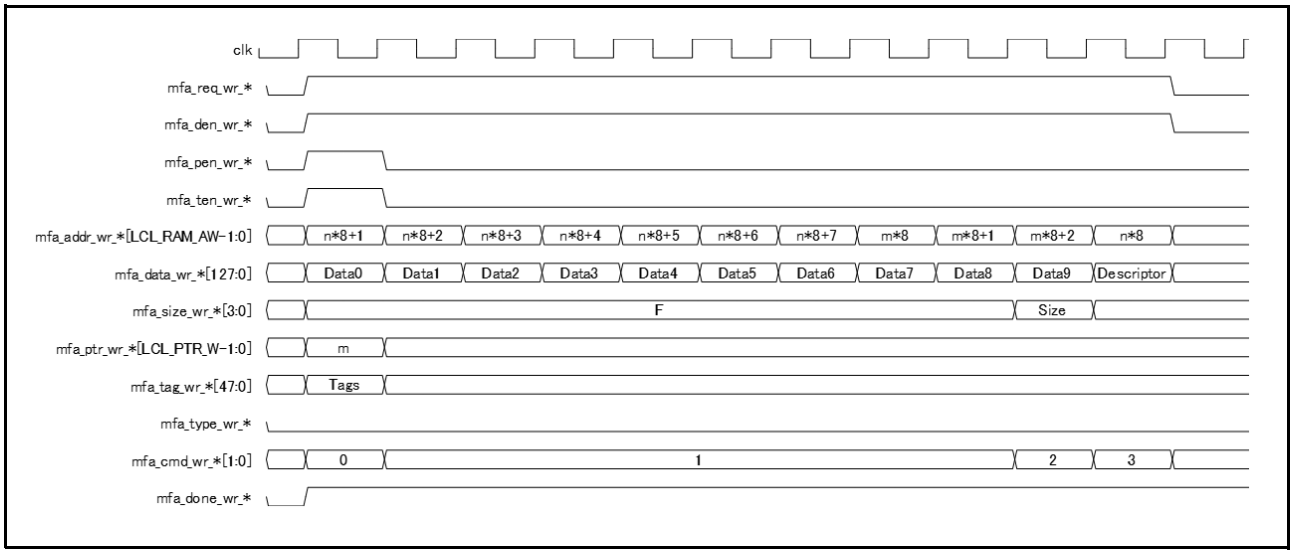


Figure 34.15 Frame Corresponding Fabric Write Accesses

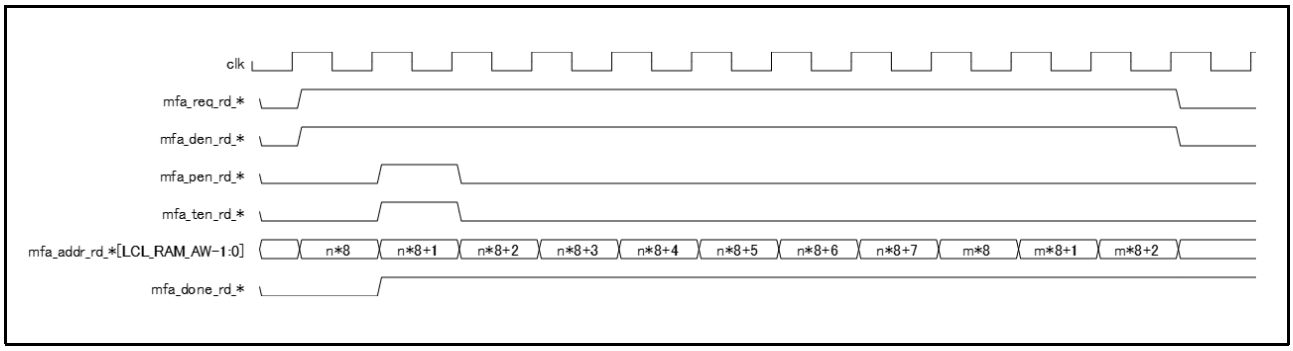


Figure 34.16 Frame Corresponding Fabric Read Accesses

34.3.4 Fabric Time Arbiter

Time arbiter controls done signal of Ethernet Agent [ETHA] Write and Read buses depending on link speed.
The faster a link is the more the corresponding agent will be able access to the local RAM.

34.3.4.1 Agents Done Signal Distribution

The Table 34.24 shows done signal distribution depending on PHY link speed possible combinations. Any disabled agent has a *doneSignalDistributionValue* equal to 0.

Table 34.24 Done Signal Output Ratio by Combination of PHY Link Speed

PHY link speeds of time-critical agents in OPERATION mode [ETHA]	*doneSignalDistributionValue* **[cycle]**
Only one Agent enabled	1
PHY link speed of all Agents is the same	1
100 Mbps : 1 Gbps	1 : 10

34.3.4.2 Done Signal Patterns

Figure 34.17 shows a done signal pattern example where time-critical port 0 has a 100 Mbps link and time-critical port 1 has a 1 Gbps link. When a request signals is not asserted at the timing when the corresponding done signal should have been asserted the done signals is not asserted and the handshake does not happen. In Figure 34.17 the request signals are always asserted to make the done pattern visible but, in a real system, request signals can be asserted anytime.

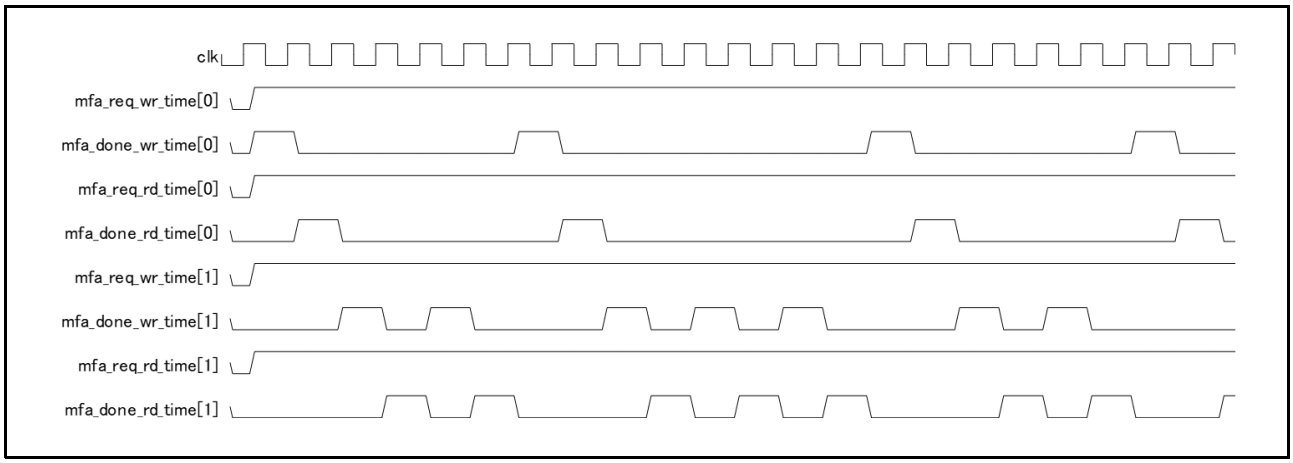


Figure 34.17 Done Signal Pattern Example

The done signal pattern has inherent properties that are always respected:

- For a static configuration the done pattern is cyclic with a constant cycle time *doneCycleClk* described in equation (4).
- During a period of *doneCycleClk* clocks starting from anywhere, a time-critical agent receives for both read and write buses a number of done signals equal to its associated done signal distribution value (Table 34.24).

(4) $doneCycleClk[cycle] = (\Sigma_2\ doneSignalDistributionValue[cycle]) \times 2$

[Where]

- Σ_2 *doneSignalDistributionValue* is the sum of all time-critical agents in OPERATION mode *doneSignalDistributionValue* (Table 34.24).

[Restriction]

- HW: Done signals are only given to Ethernet agents in OPERATION mode [ETHA].
- HW: If an unknown link speed is set for time-critical port i [ETHA] (mfa_xmii_speed[3 × (i + 1) – 1:3 × i] > 101b [RMAC]), Fabric will consider port i as disabled and port i will not be able to access the time arbiter.

34.3.4.3 Fabric Jitter

When an Ethernet Agent reads data using Fabric time arbiter, some jitter is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv] or PSFP [802.1Qci]. The fabric jitter can be calculated for an agent using equation (6).

(6) $fabricJitter[ns] = (\lceil doneCycleClk[cycle] / doneSignalDistributionValue[cycle] \rceil + (PortTimeOperation[number] - 1) \times 2) \times clk_period[ns]$

[Where]

- *doneCycleClk* is detailed in section 34.3.4.2, Done Signal Patterns.
- *doneSignalDistributionValue* is corresponding agent value detailed in section 34.3.4.1, Agents Done Signal Distribution
- *PortTimeOperation* is the number of Ethernet Agents in OPERATION mode.
- *clk_period* is the ESWCLK clock period.

[Restrictions]

- SW: The *fabricJitter* equation takes in account the number of Ethernet Agents [ETHA] in OPERATION mode and also *doneCycleClk*. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric jitter value. However, reconfiguration of TAS [802.1Qbv] or PSFP [802.1Qci] jitter in OPERATION is not a recommended option so, the maximum number of Ethernet Agents in OPERATION mode and their corresponding maximum PHY speed should be determined to calculate the worst *fabricJitter* per agent and it should be kept to the same value all the time.

34.3.4.4 Fabric Minimum Latency

When an Ethernet Agent reads data using Fabric time arbiter, some latency is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv] or PSFP [802.1Qci]. The fabric minimum latency can be calculated for an agent using equation (7).

(7) $fabricLatency[ns] = (\lfloor doneCycleClk[cycle] / doneSignalDistributionValue[cycle] \rfloor - 1) \times clk_period[ns]$

[Where]

- *doneCycleClk* is detailed in section 34.3.4.2, Done Signal Patterns.
- *doneSignalDistributionValue* is corresponding agent value detailed in section 34.3.4.1, Agents Done Signal Distribution.
- *clk_period* is the ESWCLK clock period.

[Restrictions]

- SW: The *fabricLatency* equation takes in account *doneCycleClk*. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric latency value. However, reconfiguration of TAS [802.1Qbv] or PSFP [802.1Qci] latency in OPERATION is not a recommended option so, if Ethernet agent are disabled or PHY speed is changed during OPERATION, it is recommended to set the *fabricLatency* to 2 × *clk_period[ns]* and add the *fabricLatency* calculated value in the worst case to *fabricJitter*.
- SW: When preemption is enabled for a port, *fabricLatency* should be set to 0ns and *fabricLatency* calculated value when all agents are enabled should be added to *fabricJitter*.

34.3.5 Source Port Generation

Fabric gives automatically a source port number to each Agent depending on the bus they are accessing to. Table 34.25 shows the relation between the Bus accessed by an agent and its source port number.

Table 34.25 Source Port Number Assignment

Bus from which agent access the Fabric	Source port number
ETHA0	0
ETHA1	1
GWCA0	2

34.3.6 Software Flows

[Restrictions]

- The registers within MFWD can only be written using the flows specified in this section.
- All software flows described in this section should be used in sequence.

34.3.6.1 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except RESET mode.
The interrupt handling flow is described in Figure 34.18.

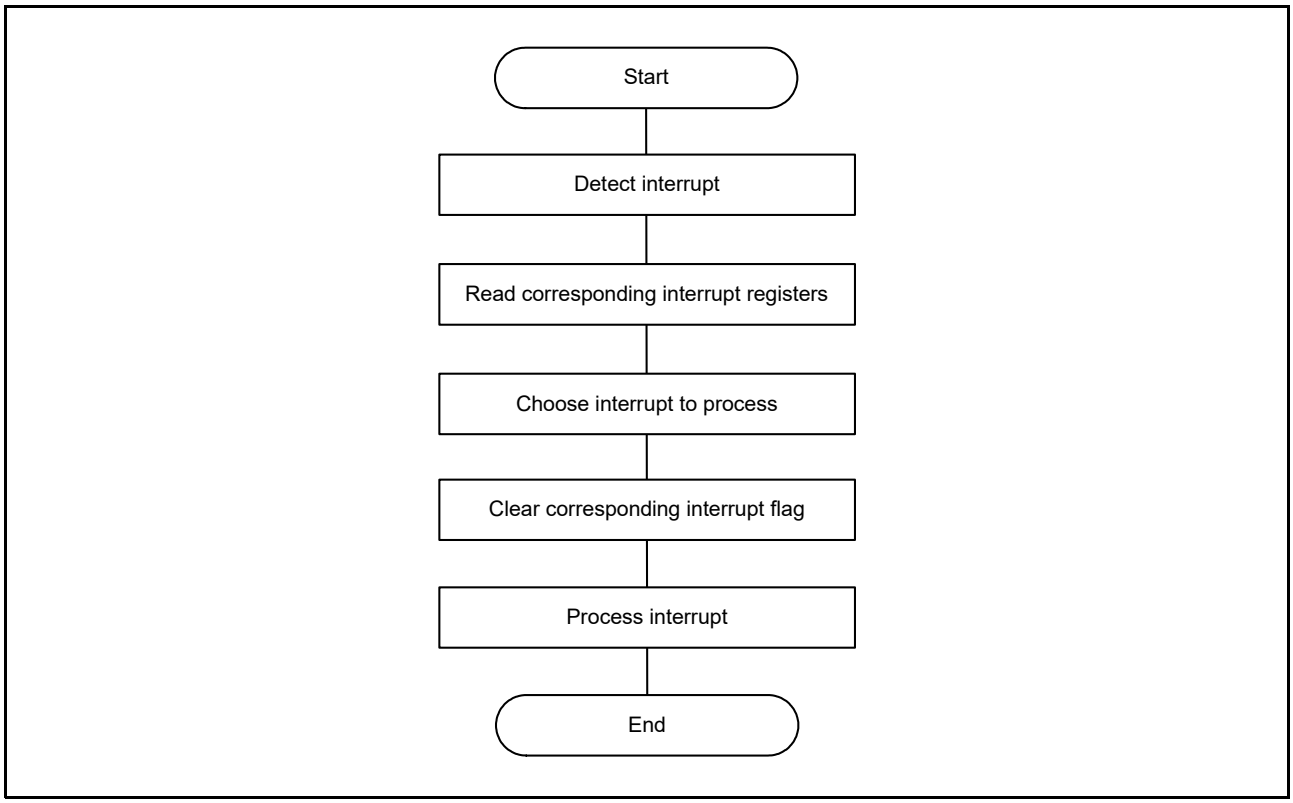


Figure 34.18 Interrupt Handling Flow

34.3.6.2 Port i Setting Flow

Port i setting flow is described in Figure 34.19.

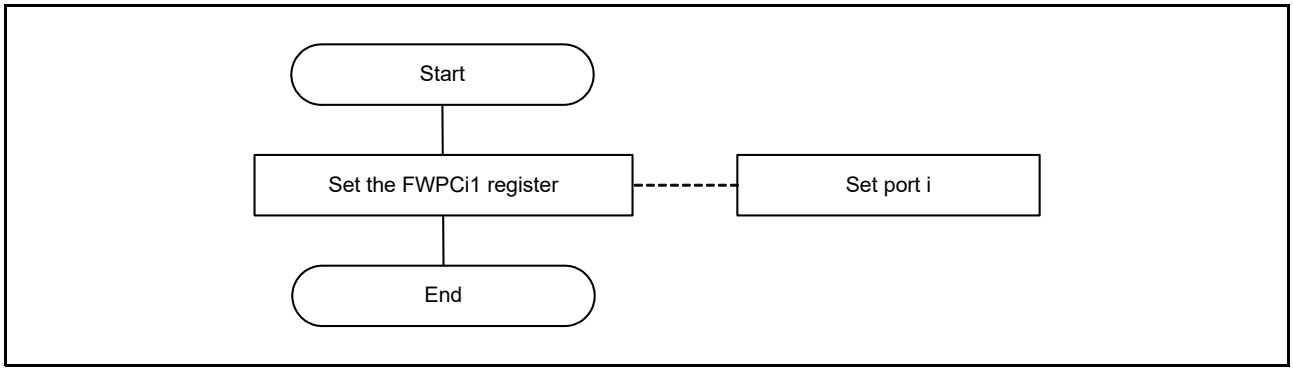


Figure 34.19 Port i Setting Flow (i = 0, 2)

34.3.6.3 Port Based Forwarding Flows

(1) Port i Port-Based Setting Flow

Port i port-based setting flow is described in Figure 34.20.

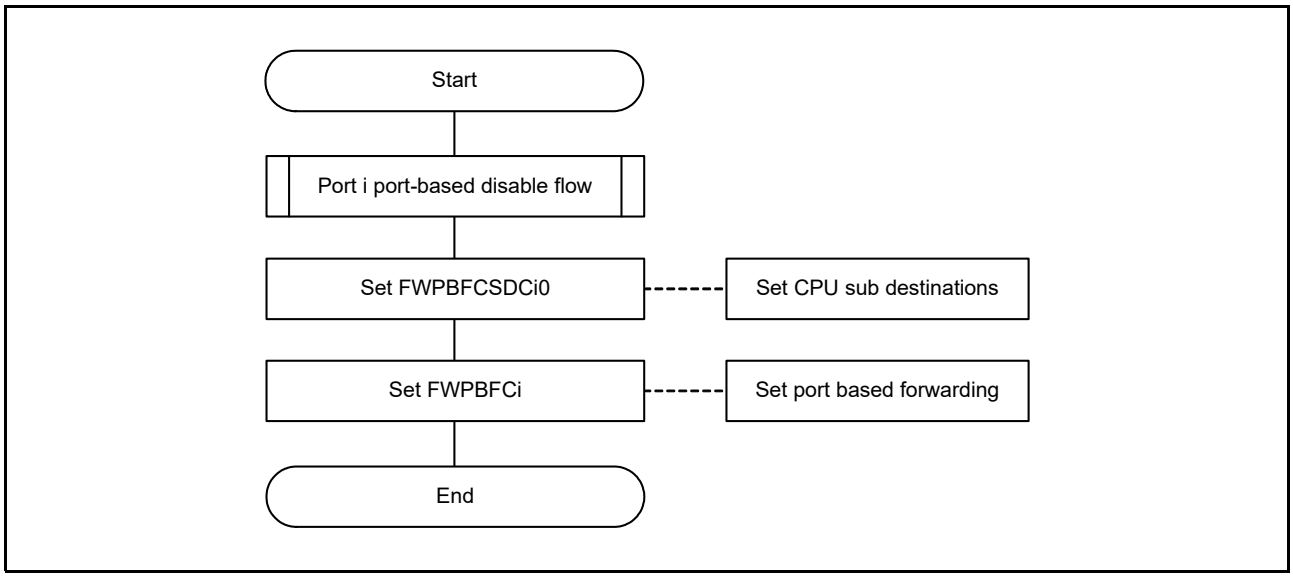


Figure 34.20 Port i Port-Based Setting Flow

(2) Port i Port-Based Disable Flow

Port i port-based disable flow is described in Figure 34.21.

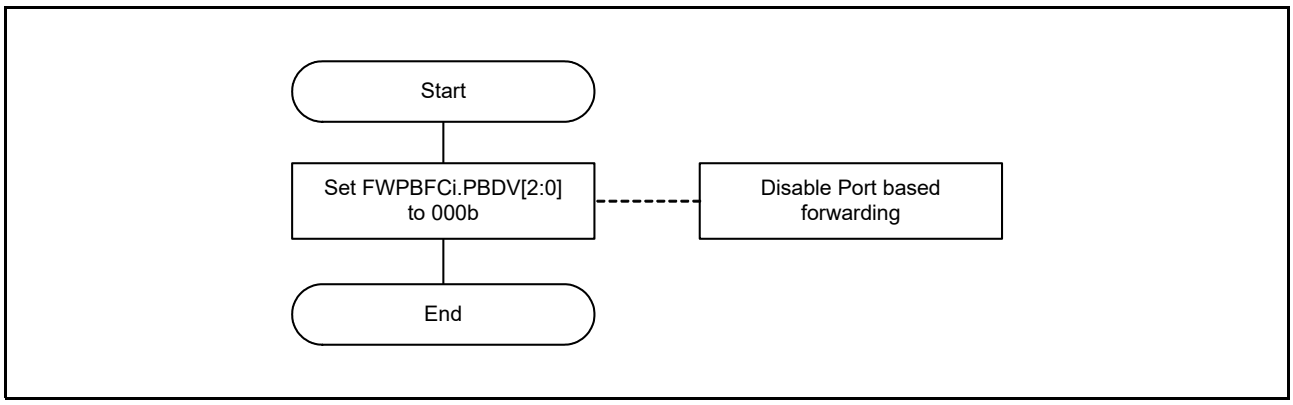


Figure 34.21 Port i Port-Based Disable Flow

34.3.6.4 Called Software Flows

The flows described in this section can only be called from other flows and cannot be used alone.

(1) Overall Setting Flow

Overall setting flow is described in Figure 34.22.

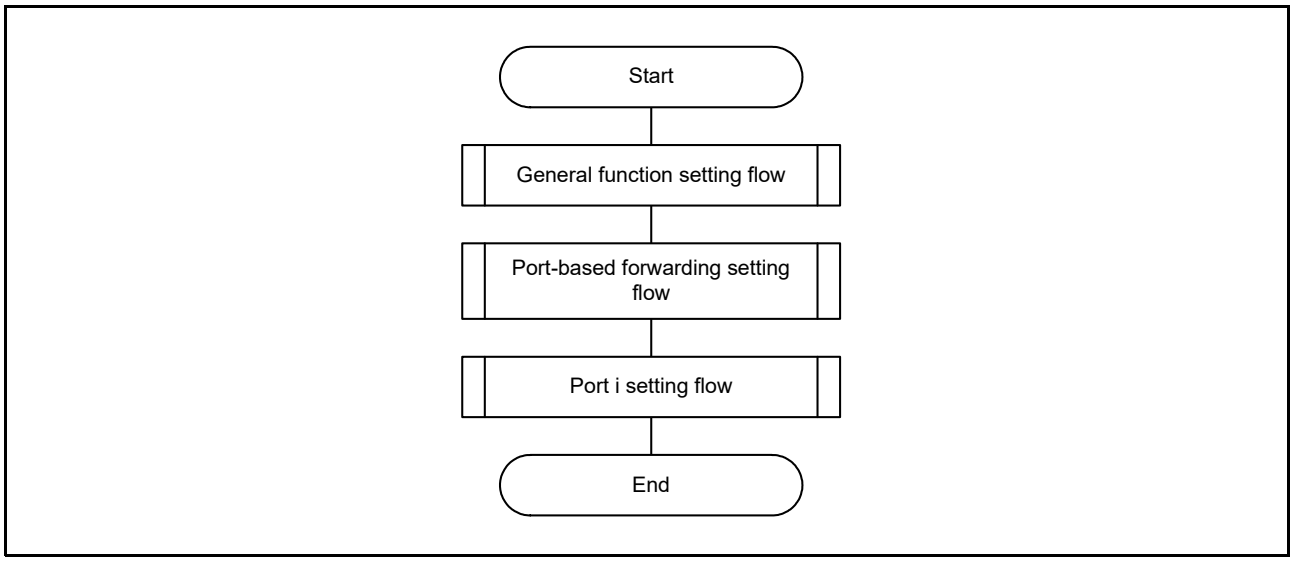


Figure 34.22 Overall Setting Flow (i = 0, 2)

(2) General Function Setting Flow

General function setting flow is described in Figure 34.23.

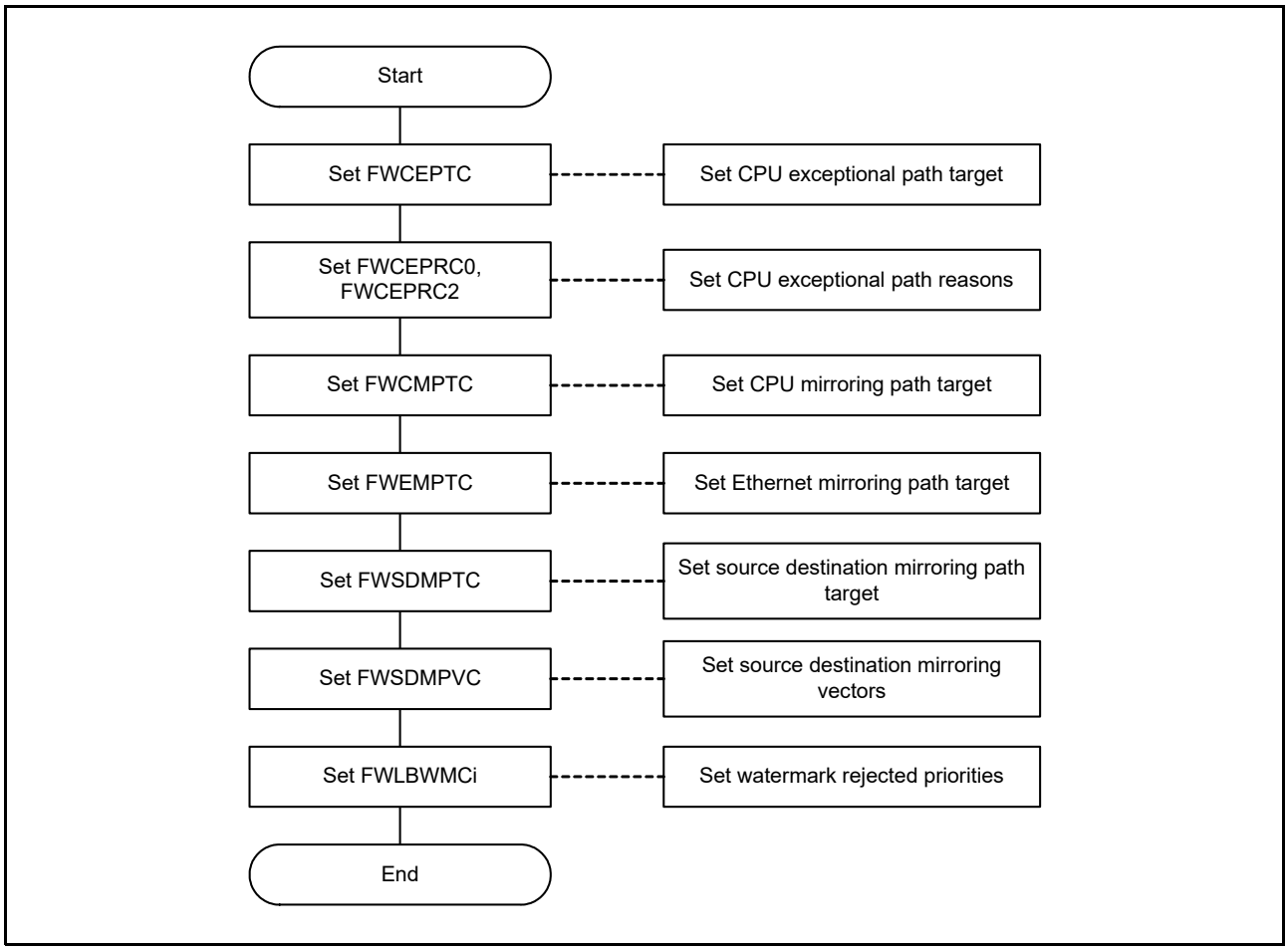


Figure 34.23 General Function Setting Flow

(3) Port-Based Forwarding Setting Flow

Port-based forwarding setting flow is described in Figure 34.24.

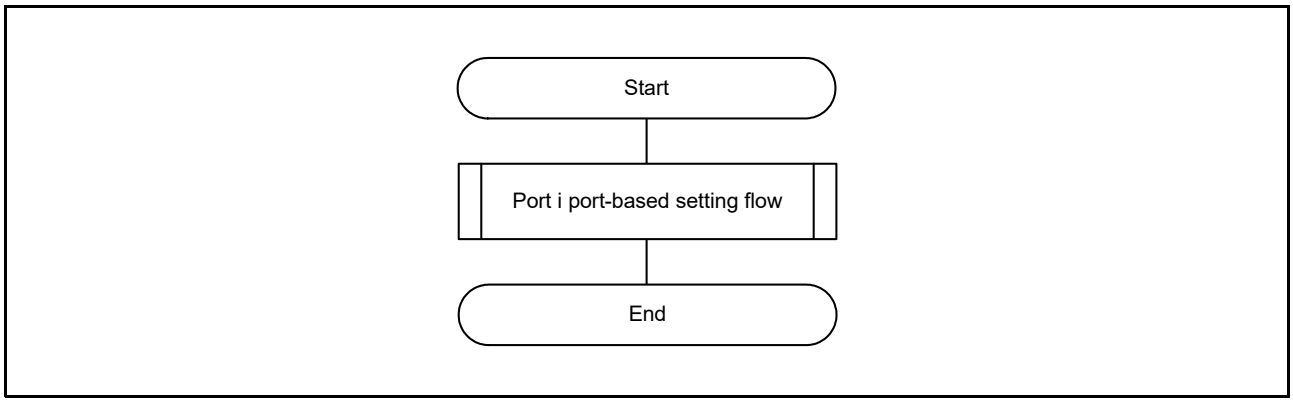


Figure 34.24 Port-Based Forwarding Setting Flow (i = 0, 2)

34.3.6.5 Register Writable Without Software Flow After Initialization

The table below describes the registers writable without software flow and in which mode they can be written without software flow.

Table 34.26 Writable Without Software Flow Registers

Register name	Writable anytime
FWCEPTC	✓
FWCEPRC0, FWCEPRC2	✓
FWCMPTC	✓
FWEMPTC	✓
FWSDMPTC	✓
FWSDMPVC	✓
FWLBWMCi	✓
FWPCi1	✓

✓ = Writable, × = not Writable

35. Ethernet Common Agent (COMA)

35.1 Overview

Ethernet common agent (COMA) is an agent of ESWM system which aims at gathering common functionalities for other agents.
It handles the APB to SFR bus conversion, the buffer pointer release for rejected frames and the pointer handling for Local RAM. Software involvement required for common agent configuration.

35.1.1 Features

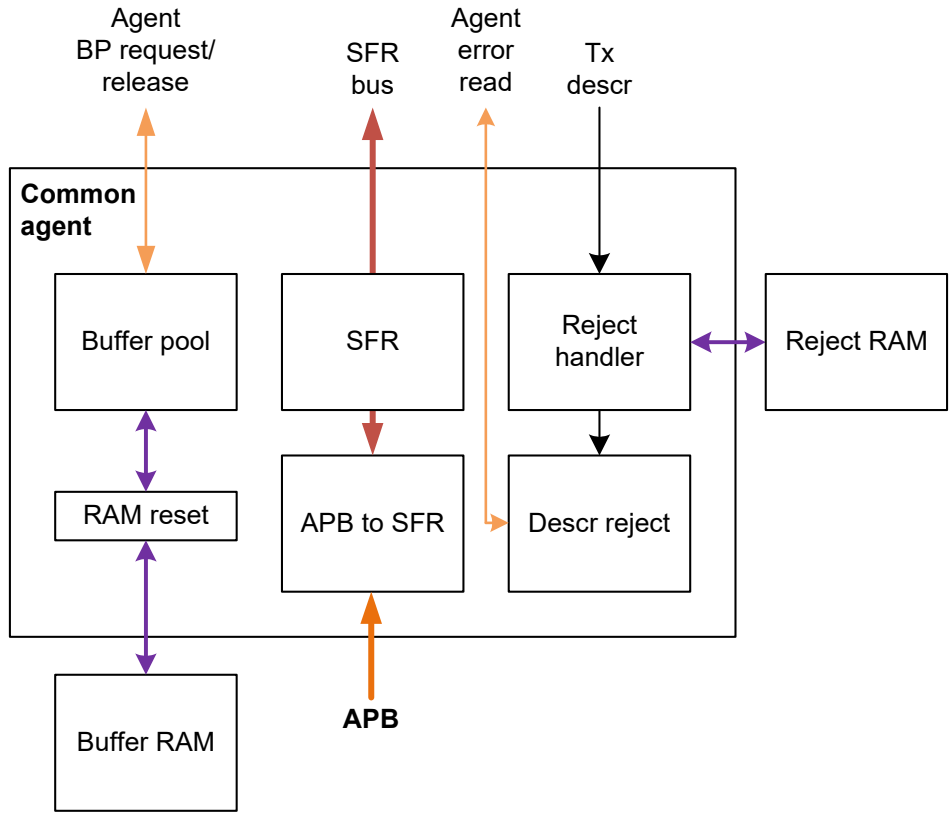
COMA features are described in Table 35.1.

Table 35.1 COMA Feature List

Function		Details
Overall function	Pointer management	• Manage the Local RAM pointers and their distribution
	APB to SFR	• Convert APB protocol in SFR bus • Distribute SFR access among other Ethernet-related modules
	Descriptor reject	• Reject the descriptors unused by agents for transmission
Data provision	Fabric error bus	• Pointer received from fabric through error bus
	Descriptor bus	• Descriptor received from forwarding engine (MFWD) for rejection
Control function	RAM access	• Function that writes and reads to RAM

35.1.2 COMA Block Diagram

Figure 35.1 shows common agent block diagram.



APB to SFR : Convert APB bus to SFR bus for every IP in ESWM system
SFR : COMA SFR
Reject handler : Receive descriptors to be rejected from forwarding engine and save them in reject RAM
Descr reject : Reject the descriptors
Free Buffer pool : Handle the free buffer pointers
RAM reset : Handle RAM reset

Figure 35.1 COMA Block Diagram

35.2 Register Description

35.2.1 Global Registers

35.2.1.1 IP Version Register (RIPV)

Address(es): COMA.RIPV 874C 9000h, COMA_NS.RIPV 074C 9000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CAIPV[3:0]				FBIPV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EAIPV[3:0]				FWIPV[3:0]				GWIPV[3:0]				TIPV[3:0]			
Value after reset:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	TIPV[3:0]	Top Module IP Version Number		R
b7 to b4	GWIPV[3:0]	Gateway CPU Agent IP Version Number		R
b11 to b8	FWIPV[3:0]	Forwarding Engine IP Version Number		R
b15 to b12	EAIPV[3:0]	Ethernet Agent IP Version Number		R
b19 to b16	FBIPV[3:0]	Fabric Bus IP Version Number		R
b23 to b20	CAIPV[3:0]	Common Agent IP Version Number		R
b31 to b24	—	Reserved	These bits are read as 0.	R

35.2.1.2 Reset Configuration Register (RRC)

Address(es): COMA.RRC 874C 9004h, COMA_NS.RRC 074C 9004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RR	Software Reset*1	0: Release ESWM reset. 1: Initiate ESWM reset.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. After writing “1”, change the value back to “0” as soon as possible.

RR Bit (Software Reset)

When this bit is set to 1, the entire ESWM system is reset.

A synchronous reset with this bit consumes a lot of power, so it is only asserted internally for seven clock cycles. The synchronous reset is also asserted internally for seven clock cycles after the hardware reset is released.

35.2.1.3 Clock Enable Configuration Register (RCEC)

Address(es): COMA.RCEC 874C 9008h, COMA_NS.RCEC 074C 9008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACE2	ACE1	ACE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACE0	Agent 0 Clock Enable	0: Agent 0 (ETHA0) clock is disabled. 1: Agent 0 (ETHA0) clock is enabled.	R/(W) *1
b1	ACE1	Agent 1 Clock Enable	0: Agent 1 (ETHA1) clock is disabled. 1: Agent 1 (ETHA1) clock is enabled.	R/(W) *1
b2	ACE2	Agent 2 Clock Enable	0: Agent 2 (GWCA0) clock is disabled. 1: Agent 2 (GWCA0) clock is enabled.	R/(W) *1
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RCE	Clock Enable	0: ESWCLK is disabled. 1: ESWCLK is enabled.	R/(W) *1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the RCDC register.

The COMA registers are controlled by PCLKA, so register read and write operations are possible regardless of the setting of this register. Each agent (ETHA, GWCA) is controlled by the agent clock, so when the ACEi bit is 0, register read operations are possible but write operations are not. Also, MFWD is controlled by ESWCLK, so when the RCE bit is 0, register read operations are possible but write operations are not.

ACEi Bit (Agent i Clock Enable) (i = 0 to 2)

This bit enables agent i clock.

When the RRC.RR bit is set to 1, all agent clocks are internally enabled to allow software reset even if the ACEi bits are not set to 1.

When the RCEC.RCE bit is not set to 1, agent clocks are internally disabled even if the ACEi bits are set to 1.

RCE Bit (Clock Enable)

This bit enables ESWCLK.

When the RRC.RR bit is set to 1, ESWCLK is internally enabled to allow software reset even if the RCE bit is not set to 1.

35.2.1.4 Clock Disable Configuration Register (RCDC)

Address(es): COMA.RCDC 874C 900Ch, COMA_NS.RCDC 074C 900Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACD2	ACD1	ACD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACD0	Agent 0 Clock Disable	Writing 1 to this bit clears the RCEC.ACE0 bit. Writing 0 to this bit has no effect. The read value is 0.	R/W
b1	ACD1	Agent 1 Clock Disable	Writing 1 to this bit clears the RCEC.ACE1 bit. Writing 0 to this bit has no effect. The read value is 0.	R/W
b2	ACD2	Agent 2 Clock Disable	Writing 1 to this bit clears the RCEC.ACE2 bit. Writing 0 to this bit has no effect. The read value is 0.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RCD	Clock Disable*1	Writing 1 to this bit clears the RCEC.RCE bit. Writing 0 to this bit has no effect. The read value is 0.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Write 1 to this bit when all agents are disabled and the CABPPCM.RPC[9:0] bits are equal to 512.

35.2.2 Buffer Pool Function Registers

35.2.2.1 Buffer Pool IPV Based Watermark Configuration Register i (CABPIBWMCi) (i = 0 to 7)

Address(es): COMA.CABPIBWMC0 874C 9020h, COMA.CABPIBWMC1 874C 9024h, COMA.CABPIBWMC2 874C 9028h, COMA.CABPIBWMC3 874C 902Ch, COMA.CABPIBWMC4 874C 9030h, COMA.CABPIBWMC5 874C 9034h, COMA.CABPIBWMC6 874C 9038h, COMA.CABPIBWMC7 874C 903Ch, COMA_NS.CABPIBWMC0 074C 9020h, COMA_NS.CABPIBWMC1 074C 9024h, COMA_NS.CABPIBWMC2 074C 9028h, COMA_NS.CABPIBWMC3 074C 902Ch, COMA_NS.CABPIBWMC4 074C 9030h, COMA_NS.CABPIBWMC5 074C 9034h, COMA_NS.CABPIBWMC6 074C 9038h, COMA_NS.CABPIBWMC7 074C 903Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	IBSWMPN[9:0]									
Value after reset:	0	0	0	0	0	0	*1									

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IBUWMPN[9:0]									
Value after reset:	0	0	0	0	0	0	*1									

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	IBUWMPN[9:0]	IPV Based Unsecure Watermark Pointer Number	• CABPIBWMC7 register: Set a value of 512 or less. Note that setting these bits to 512 disables the IPV-based watermark function for IPV 7. • CABPIBWMCi register (i = 0 to 6): Set a value equal to or less than the CABPIBWMC[i+1].IBUWMPN[9:0] bits. Note that setting these bits to the same value as the CABPIBWMC[i+1].IBUWMPN[9:0] bits disables the IPV-based watermark function for IPV i.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	IBSWMPN[9:0]	IPV Based Secure Watermark Pointer Number	• CABPIBWMC7 register: Set a value of 512 or less. Note that setting these bits to 512 disables the IPV-based watermark function for IPV 7. • CABPIBWMCi register (i = 0 to 6): Set a value equal to or less than the CABPIBWMC[i+1].IBSWMPN[9:0] bits. Note that setting these bits to the same value as the CABPIBWMC[i+1].IBSWMPN[9:0] bits disables the IPV-based watermark function for IPV i.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. The value after reset of the IBUWMPN[9:0] and IBSWMPN[9:0] bits is the same; and the value for the CABPIBWMCi register is 430 + 8 × i.

These registers are used to set watermarks based on the Internal Priority Value (IPV) assigned by the MFWD. The CABPIBWMCi register corresponds to IPV i.

Refer to section 35.3.1.1, IPV Based Watermark for details.

35.2.2.2 Buffer Pool Watermark Level Configuration Register (CABPWMLC)

Address(es): COMA.CABPWMLC 874C 9040h, COMA_NS.CABPWMLC 074C 9040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	WMCL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	WMFL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	WMFL[9:0]	Watermark Flush Level	Set a value of 512 or less. Note that setting these bits to 512 disables the watermark flush level function.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	WMCL[9:0]	Watermark Critical Level	Set a value equal to or less than the WMFL[9:0] bits. Setting these bits to 512 disables the watermark critical level function.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to set a watermark based on the global level.
Refer to section 35.3.1.2, Global Level-Based Watermark for details.

35.2.2.3 Buffer Pointer Pause Frame Level i Configuration Register (CABPPFLCi) (i = 0, 1)

Address(es): COMA.CABPPFLC0 874C 9050h, COMA.CABPPFLC1 874C 9054h, COMA_NS.CABPPFLC0 074C 9050h, COMA_NS.CABPPFLC1 074C 9054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	PAL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PDL[9:0]	Pause De-assertion Level		R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	PAL[9:0]	Pause Assertion Level	Setting this register to 512 will disable pause frames.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

PDL[9:0] Bits (Pause De-assertion Level)

Refer to section 35.3.2.1, Global Pause Function for details.

[Restrictions]

- SW: The value of these bits should always be less than or equal to the CABPPFLCi.PAL[9:0] bits.
- SW: These bits should not be set to a value greater than 512.
- SW: These bits should not be set to a value less than 24.

[Precautions]

- If CABPPFLCi.PAL[9:0] value is too close to CABPPFLCi.PDL[9:0] value, corresponding PAUSE signals will change state rapidly and if zero automatic pause frames are activated [RMAC], it can result on a succession of pause frames on RMAC TX side which will decrease the link performance.

PAL[9:0] Bits (Pause Assertion Level)

Refer to section 35.3.2.1, Global Pause Function for details.

[Restrictions]

- SW: These bits should not be set to a value greater than 512.
- SW: These bits should not be set to a value less than 24.

35.2.2.4 Port i Buffer Pool Watermark Level Configuration Register (CABPPWMLCi) (i = 0 to 2)

Address(es): COMA.CABPPWMLC0 874C 9060h, COMA.CABPPWMLC1 874C 9064h, COMA.CABPPWMLC2 874C 9068h, COMA_NS.CABPPWMLC0 074C 9060h, COMA_NS.CABPPWMLC1 074C 9064h, COMA_NS.CABPPWMLC2 074C 9068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	PWMCL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PWMFL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PWMFL[9:0]	Watermark Flush Level	Setting these bits to 512 disables the watermark flush level function.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	PWMCL[9:0]	Watermark Critical Level	Setting these bits to 512 disables the watermark critical level function.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

PWMFL[9:0] Bits (Watermark Flush Level)

Refer to section 35.3.1.3, Per-Port Level-Based Watermark for details.

[Restrictions]

- SW: These bits should not be set to a value greater than 512.

PWMCL[9:0] Bits (Watermark Critical Level)

Refer to section 35.3.1.3, Per-Port Level-Based Watermark for details.

[Restrictions]

- SW: The value of these bits should always be less than or equal to the CABPPWMLCi.PWMFL[9:0] bits.
- SW: These bits should not be set to a value greater than 512.

35.2.2.5 Port i Buffer Pointer Pause Frame Level j Configuration Register (CABPPPFLCij) (i = 0 to 2; j = 0, 1)

Address(es): COMA.CABPPPFLC00 874C 90A0h, COMA.CABPPPFLC01 874C 90A4h, COMA.CABPPPFLC10 874C 90A8h, COMA.CABPPPFLC11 874C 90ACh, COMA.CABPPPFLC20 874C 90B0h, COMA.CABPPPFLC21 874C 90B4h, COMA_NS.CABPPPFLC00 074C 90A0h, COMA_NS.CABPPPFLC01 074C 90A4h, COMA_NS.CABPPPFLC10 074C 90A8h, COMA_NS.CABPPPFLC11 074C 90ACh, COMA_NS.CABPPPFLC20 074C 90B0h, COMA_NS.CABPPPFLC21 074C 90B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	PPAL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PPDL[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PPDL[9:0]	Pause De-Assertion Level		R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	PPAL[9:0]	Pause Assertion Level	Setting these bits to 512 will disable pause frames.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

PPDL[9:0] Bits (Pause De-Assertion Level)

Refer to section 35.3.2.2, Per-Port Pause Function for details.

[Restrictions]

- SW: The value of these bits should always be less than or equal to the CABPPPFLCij.PPAL[9:0] bits.
- SW: These bits should not be set to a value greater than 512.
- SW: These bits should not be set to a value less than 6.

[Precautions]

- If CABPPFLCi.PAL[9:0] value is too close to CABPPFLCi.PDL[9:0] value, corresponding PAUSE signals will change state rapidly and if zero automatic pause frames are activated [RMAC], it can result on a succession of pause frames on RMAC TX side which will decrease the link performance.

PPAL[9:0] Bits (Pause Assertion Level)

Refer to section 35.3.2.2, Per-Port Pause Function for details.

[Restrictions]

- SW: These bits should not be set to a value greater than 512.
- SW: These bits should not be set to a value less than 6.

35.2.2.6 Port i Buffer Pointer Utilization Level Configuration Register (CABPULCi) (i = 0 to 2)

Address(es): COMA.CABPULC0 874C 9100h, COMA.CABPULC1 874C 9104h, COMA.CABPULC2 874C 9108h, COMA_NS.CABPULC0 074C 9100h, COMA_NS.CABPULC1 074C 9104h, COMA_NS.CABPULC2 074C 9108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	MNNPN[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MXNPN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	MXNPN[9:0]	Maximum Number of Pointer	Maximum number of pointers that can be used by port i.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25 to b16	MNNPN[9:0]	Minimum Number of Pointer	Minimum number of pointers that can be used by port i.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: If CABPULCi.MNNPN[9:0] of port i is changed (reconfigured) from its default value, then it restricts the reception maximum frame size of other ports(i.e. ports other than port i). The restricted reception maximum frame size of ports other than port i in such case is given by below expression.
Restricted reception maximum frame size = ((512 – CABPULCi.MNNPN[9:0])/2) × 128 bytes. This is because other ports (ports other than port i) have to share remaining pointers (512 – CABPULCi.MNNPN[9:0]).
If this restriction is not followed, then the consumed buffer pointers will not be released by other ports and the frame will be stuck inside ESWM. Because the total buffer pointers are insufficient for the port in the middle of reception. To resolve this, reconfigure CABPULCi.MNNPN[9:0] value to a smaller value.

MXNPN[9:0] Bits (Maximum Number of Pointer)

If the value of the CABPCPMi.RPCP[9:0] bits is equal to the value of these bits, port i will not receive pointer anymore.
Refer to section 35.3.3, Per Port Memory Allocation Function for details.
[Restrictions]

- SW: The maximum number of pointers should be set to a value big enough to save 2 frames with a size equal to RMAC reception maximum frame size (set in RMAC [RMAC]).

MNNPN[9:0] Bits (Minimum Number of Pointer)

If SUM(CABPULCi.MNNPN[9:0]) ≥ CABPPCM.RPC[9:0] + SUM(CABPCPMi.RPCP[9:0]) only ports for which CABPCPMi.RPCP[9:0] < CABPULCi.MNNPN[9:0] will be able to receive pointers.
Refer to section 35.3.3, Per Port Memory Allocation Function for details.
[Restrictions]

- SW: The sum of all CABPULCi.MNNPN[9:0] bits should be less than or equal to 512.
- SW: These bits should always be less than or equal to the CABPULCi.MXNPN[9:0] bits.

35.2.2.7 Buffer Pool Initialization Register Monitoring Register (CABPIRM)

Address(es): COMA.CABPIRM 874C 9140h, COMA_NS.CABPIRM 074C 9140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BPR	BPIOG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BPIOG	Buffer Pool Initialization Ongoing		R/W
b1	BPR	Buffer Pool Ready	0: Buffer pool cannot be used. 1: Buffer pool can be used. The write value should be 0.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

BPIOG Bit (Buffer Pool Initialization Ongoing)

[Setting Condition]

- SW: By writing 1 to this bit.

[Clearing Condition]

- HW: This bit is cleared when buffer pool initialization is finished.

[Restriction]

- HW: CPU cannot write 1 to this bit when the CABPIRM.BPR bit is set.

BPR Bit (Buffer Pool Ready)

[Setting Condition]

- When the CABPIRM.BPIOG bit is getting cleared.
- HW: This bit is set at “clk_period[ns]*512” time from buffer pool initialization starting.

35.2.2.8 Buffer Pool Pointer Count Monitoring Register (CABPPCM)

Address(es): COMA.CABPPCM 874C 9144h, COMA_NS.CABPPCM 074C 9144h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TPC[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RPC[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	RPC[9:0]	Remaining Pointer Count		R
b15 to b10	—	Reserved	These bits are read as 0.	R
b25 to b16	TPC[9:0]	Total Pointer Count	The value of these bits is fixed to 512.	R
b31 to b26	—	Reserved	These bits are read as 0.	R

RPC[9:0] Bits (Remaining Pointer Count)

[Increment Condition]

- HW: Incremented by 1 when a pointer is released.

[Decrement Condition]

- HW: Decremented by 1 when a pointer is given to an agent.

TPC[9:0] Bits (Total Pointer Count)

Reading these bits gives the number of pointers contained in the buffer pool. The ESWM local data RAM size will be equal to 128 × CABPPCM.TPC[9:0] bytes.

35.2.2.9 Buffer Pool Pointer Least Count Monitoring Register (CABPLCM)

Address(es): COMA.CABPLCM 874C 9148h, COMA_NS.CABPLCM 074C 9148h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LRC[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	LRC[9:0]	Least Remaining Pointer Count	These bits store the minimum value taken by the pointer remaining counter.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

RPC[9:0] Bits (Least Remaining Pointer Count)

[Update Condition]

- HW: These bits are updated to the CABPPCM.RPC[9:0] bits when CABPPCM.RPC[9:0] < CABPLCM.LRC[9:0].

[Clearing Condition]

- Reading these bits clears it to CABPPCM.RPC[9:0] value.

35.2.2.10 Port i Buffer Pointer Count Monitoring Register (CABPCPMi) (i = 0 to 2)

Address(es): COMA.CABPCPM0 874C 9180h, COMA.CABPCPM1 874C 9184h, COMA.CABPCPM2 874C 9188h, COMA_NS.CABPCPM0 074C 9180h, COMA_NS.CABPCPM1 074C 9184h, COMA_NS.CABPCPM2 074C 9188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RPCP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	RPCP[9:0]	Received Pointer Count	These bits show the number of pointers that are used by agent i to store data.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

RPCP[9:0] Bits (Received Pointer Count)

[Increment Condition]

- HW: Incremented by 1 when a pointer is given to port i.

[Decrement Condition]

- HW: Decremented by 1 when a pointer which has been given to port i is being released.

35.2.2.11 Port i Buffer Pointer Maximum Count Monitoring Register (CABPMCPMi) (i = 0 to 2)

Address(es): COMA.CABPMCPM0 874C 9200h, COMA.CABPMCPM1 874C 9204h, COMA.CABPMCPM2 874C 9208h, COMA_NS.CABPMCPM0 074C 9200h, COMA_NS.CABPMCPM1 074C 9204h, COMA_NS.CABPMCPM2 074C 9208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RPMCP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	RPMCP[9:0]	Received Pointer Maximum Count	These bits store the maximum value taken by the CABPCPMi.RPCP[9:0] bits since last read access.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

RPMCP[9:0] Bits (Received Pointer Maximum Count)

[Update Condition]

- HW: These bits are updated to the CABPCPMi.RPCP[9:0] bits when CABPCPMi.RPCP[9:0] > CABPMCPMi.RPMCP[9:0].

[Clearing Condition]

- Reading these bits clears it to CABPCPMi.RPCP[9:0] value.

[Restrictions]

- It may momentarily exceed several upper and lower limit settings.

35.2.3 Reject RAM Function Registers

35.2.3.1 Rejected Descriptor Number Monitoring Register (CARDNM)

Address(es): COMA.CARDNM 874C 9300h, COMA_NS.CARDNM 074C 9300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RDNRR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	RDNRR[9:0]	Rejected Descriptor Number in Reject RAM	These bits show the number of descriptors currently contained in the reject RAM.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

RDNRR[9:0] Bits (Rejected Descriptor Number in Reject RAM)

[Increment Condition]

- HW: Incremented by 1 a descriptor is received from forwarding engine.

[Decrement Condition]

- HW: Decremented by 1 when a descriptor is read from reject RAM.

35.2.3.2 Rejected Descriptor Maximum Number Monitoring Register (CARDMNM)

Address(es): COMA.CARDMNM 874C 9304h, COMA_NS.CARDMNM 074C 9304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RDMNRR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	RDMNRR[9:0]	Rejected Descriptor Maximum Number in Reject RAM	These bits store the maximum value taken by the CARDNM.RDNRR[9:0] bits since last read access.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

RDMNRR[9:0] Bits (Rejected Descriptor Maximum Number in Reject RAM)

[Update Condition]

- Set to the CARDNM.RDNRR[9:0] bits when CARDNM.RDNRR[9:0] > CARDMNM.RDMNRR[9:0].

[Clearing Condition]

- Reading these bits clears it to CARDNM.RDNRR[9:0] value.

35.2.4 Reject RAM Counter Registers

35.2.4.1 Rejected Descriptor Counter Register (CARDCN)

Address(es): COMA.CARDCN 874C 9310h, COMA_NS.CARDCN 074C 9310h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RDN[31:0]	Rejected Descriptor Number	These bits count the number of descriptors that have been rejected since the last read.	R

RDN[31:0] Bits (Rejected Descriptor Number)

[Clearing Condition]

- SW: Reading these bits clears it.

[Increment Condition]

- HW: Incremented by 1 when a rejected descriptor is received from forwarding engine [MFWD] and if this register has a value different than FFFF FFFFh.

35.2.5 Error Interrupt Registers

35.2.5.1 Error Interrupt Status Register 0 (CAEIS0)

Address(es): COMA.CAEIS0 874C 9400h, COMA_NS.CAEIS0 074C 9400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	WMFLOS	WMCLOS	BPOPS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BPOPS	Buffer Pool Out of Pointer Status Flag	0: No Buffer Pool Out of Pointer 1: Buffer Pool Out of Pointer occurred.	R/W
b9	WMCLOS	Watermark Critical Level Overtook Status Flag	0: No Watermark Critical Level Overtook 1: Watermark Critical Level Overtook occurred.	R/W
b10	WMFLOS	Watermark Flush Level Overtook Status Flag	0: No Watermark Flush Level Overtook 1: Watermark Flush Level Overtook occurred.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

BPOPS Flag (Buffer Pool Out of Pointer Status Flag)

[Setting Condition]

- HW: When the CABPPCM.RPC[9:0] bits become null.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- SW: No recovery, ESWM could overflow [RMAC]
  When using cut through, please use the watermark function so that this status does not occur. If get this error (it may get hung-up in the transmit state [ETHA and RMAC]), initialize the state with Emergency reset flow [ESWM].

WMCLOS Flag (Watermark Critical Level Overtook Status Flag)

[Setting Condition]

- HW: When watermark critical level (WM.CRITICAL) gets asserted to All1 because of global level-based watermark (refer to Figure 35.3).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- SW: No recovery, data can be lost [MFWD].

WMFLOS Flag (Watermark Flush Level Overtook Status Flag)

[Setting Condition]

- HW: When watermark flush level (WM.FLUSH) gets asserted to All1 because of global level-based watermark (refer to Figure 35.3).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- SW: No recovery, data can be lost [MFWD].

35.2.5.2 Error Interrupt Enable Register 0 (CAEIE0)

Address(es): COMA.CAEIE0 874C 9404h, COMA_NS.CAEIE0 074C 9404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	WMFLOE	WMCLOE	BPOPE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BPOPE	Buffer Pool Out of Pointer Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b9	WMCLOE	Watermark Critical Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	WMFLOE	Watermark Flush Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the CAEID0 register.

BPOPE Bit (Buffer Pool Out of Pointer Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the CAEID0.BPOPD bit will clear this bit.

WMCLOE Bit (Watermark Critical Level Overtook Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the CAEID0.WMCLOD bit will clear this bit.

WMFLOE Bit (Watermark Flush Level Overtook Enable)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- Writing 1 to the CAEID0.WMFLOD bit will clear this bit.

35.2.5.3 Error Interrupt Disable Register 0 (CAEID0)

Address(es): COMA.CAEID0 874C 9408h, COMA_NS.CAEID0 074C 9408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	WMFLOD	WMCLOD	BPOPD	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BPOPD	Buffer Pool Out of Pointer Disable	0: Has no effects. 1: Sets the CAEIE0.BPOPE bit to 0.	R/W
b9	WMCLOD	Watermark Critical Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE0.WMCLOE bit to 0.	R/W
b10	WMFLOD	Watermark Flush Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE0.WMFLOE bit to 0.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.2.5.4 Error Interrupt Status Register 1 (CAEIS1)

Address(es): COMA.CAEIS1 874C 9410h, COMA_NS.CAEIS1 074C 9410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOS2	PWMFLOS1	PWMFLOS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOS2	PWMCLOS1	PWMCLOS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PWMCLOS0	Port 0 Watermark Critical Level Overtook Status Flag	0: No Port 0 Watermark Critical Level Overtook 1: Port 0 Watermark Critical Level Overtook occurred.	R/W
b1	PWMCLOS1	Port 1 Watermark Critical Level Overtook Status Flag	0: No Port 1 Watermark Critical Level Overtook 1: Port 1 Watermark Critical Level Overtook occurred.	R/W
b2	PWMCLOS2	Port 2 Watermark Critical Level Overtook Status Flag	0: No Port 2 Watermark Critical Level Overtook 1: Port 2 Watermark Critical Level Overtook occurred.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PWMFLOS0	Port 0 Watermark Flush Level Overtook Status Flag	0: No Port 0 Watermark Flush Level Overtook 1: Port 0 Watermark Flush Level Overtook occurred.	R/W
b17	PWMFLOS1	Port 1 Watermark Flush Level Overtook Status Flag	0: No Port 1 Watermark Flush Level Overtook 1: Port 1 Watermark Flush Level Overtook occurred.	R/W
b18	PWMFLOS2	Port 2 Watermark Flush Level Overtook Status Flag	0: No Port 2 Watermark Flush Level Overtook 1: Port 2 Watermark Flush Level Overtook occurred.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

PWMCLOSi Flag (Port i Watermark Critical Level Overtook Status Flag) (i = 0 to 2)

[Setting Condition]

- HW: When watermark critical level for port i (WM.CRITICAL[i]) gets asserted to 1 because of per-port level-based watermark (refer to Figure 35.4).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- No recovery, data can be lost [MFWD].

PWMFLOSi Flag (Port i Watermark Flush Level Overtook Status Flag) (i = 0 to 2)

[Setting Condition]

- HW: When watermark flush level for port i (WM.FLUSH[i]) gets asserted to 1 because of per-port level-based watermark (refer to Figure 35.4).

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

[Error Recovery]

- No recovery, data can be lost [MFWD].

35.2.5.5 Error Interrupt Enable Register 1 (CAEIE1)

Address(es): COMA.CAEIE1 874C 9414h, COMA_NS.CAEIE1 074C 9414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOE2	PWMFLOE1	PWMFLOE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOE2	PWMCLOE1	PWMCLOE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PWMCLOE0	Port 0 Watermark Critical Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	PWMCLOE1	Port 1 Watermark Critical Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	PWMCLOE2	Port 2 Watermark Critical Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PWMFLOE0	Port 0 Watermark Flush Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	PWMFLOE1	Port 1 Watermark Flush Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	PWMFLOE2	Port 2 Watermark Flush Level Overtook Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the CAEID1 register.

PWMCLOEi Bit (Port i Watermark Critical Level Overtook Enable) (i = 0 to 2)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- SW: Writing 1 to the CAEID1.PWMCLODi bit will clear this bit.

PWMFLOEi Bit (Port i Watermark Flush Level Overtook Enable) (i = 0 to 2)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- SW: Writing 1 to the CAEID1.PWMFLODi bit will clear this bit.

35.2.5.6 Error Interrupt Disable Register 1 (CAEID1)

Address(es): COMA.CAEID1 874C 9418h, COMA_NS.CAEID1 074C 9418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOD2	PWMFLOD1	PWMFLOD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOD2	PWMCLOD1	PWMCLOD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PWMCLOD0	Port 0 Watermark Critical Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMCLOE0 bit to 0.	R/W
b1	PWMCLOD1	Port 1 Watermark Critical Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMCLOE1 bit to 0.	R/W
b2	PWMCLOD2	Port 2 Watermark Critical Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMCLOE2 bit to 0.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PWMFLOD0	Port 0 Watermark Flush Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMFLOE0 bit to 0.	R/W
b17	PWMFLOD1	Port 1 Watermark Flush Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMFLOE1 bit to 0.	R/W
b18	PWMFLOD2	Port 2 Watermark Flush Level Overtook Disable	0: Has no effects. 1: Sets the CAEIE1.PWMFLOE2 bit to 0.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.2.6 Monitoring Interrupt Registers

35.2.6.1 Monitoring Interrupt Status Register 0 (CAMIS0)

Address(es): COMA.CAMIS0 874C 9440h, COMA_NS.CAMIS0 074C 9440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFS1	PFS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFS0	Pause Frame Status Flag 0	0: No Pause Frame 0 1: Pause Frame 0 occurred.	R/W
b1	PFS1	Pause Frame Status Flag 1	0: No Pause Frame 1 1: Pause Frame 1 occurred.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

PFSi Flag (Pause Frame Status Flag i) (i = 0, 1)

[Setting Condition]

- HW: When pause frame i starts to happen (pause assertion timing). Refer to section 35.3.2, Pause Function.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

35.2.6.2 Monitoring Interrupt Enable Register 0 (CAMIE0)

Address(es): COMA.CAMIE0 874C 9444h, COMA_NS.CAMIE0 074C 9444h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFE1	PFE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFE0	Pause Frame Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	PFE1	Pause Frame Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the CAMID0 register.

PFEi Bit (Pause Frame Enable i) (i = 0, 1)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- SW: Writing 1 to the CAMID0.PFDi bit will clear this bit.

35.2.6.3 Monitoring Interrupt Disable Register 0 (CAMID0)

Address(es): COMA.CAMID0 874C 9448h, COMA_NS.CAMID0 074C 9448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFD1	PFD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFD0	Pause Frame Disable 0	0: Has no effects. 1: Sets the CAMIE0.PFE0 bit to 0.	R/W
b1	PFD1	Pause Frame Disable 1	0: Has no effects. 1: Sets the CAMIE0.PFE1 bit to 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.2.6.4 Monitoring Interrupt Status Register 1 (CAMIS1)

Address(es): COMA.CAMIS1 874C 9450h, COMA_NS.CAMIS1 074C 9450h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PPFS21	PPFS20	PPFS11	PPFS10	PPFS01	PPFS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PPFS00	Port 0 Pause Frame Status Flag 0	0: No Port 0 Pause Frame 0 1: Port 0 Pause Frame 0 occurred.	R/W
b1	PPFS01	Port 0 Pause Frame Status Flag 1	0: No Port 0 Pause Frame 1 1: Port 0 Pause Frame 1 occurred.	R/W
b2	PPFS10	Port 1 Pause Frame Status Flag 0	0: No Port 1 Pause Frame 0 1: Port 1 Pause Frame 0 occurred.	R/W
b3	PPFS11	Port 1 Pause Frame Status Flag 1	0: No Port 1 Pause Frame 1 1: Port 1 Pause Frame 1 occurred.	R/W
b4	PPFS20	Port 2 Pause Frame Status Flag 0	0: No Port 2 Pause Frame 0 1: Port 2 Pause Frame 0 occurred.	R/W
b5	PPFS21	Port 2 Pause Frame Status Flag 1	0: No Port 2 Pause Frame 1 1: Port 2 Pause Frame 1 occurred.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

PPFSij Flag (Port i Pause Frame Status Flag j) (i = 0 to 2) (j = 0, 1)

[Setting Condition]

- HW: When the port i pause frame j starts to happen (pause assertion timing). Refer to section 35.3.2, Pause Function.

[Clearing Condition]

- SW: Writing 1 to this flag will clear it.

35.2.6.5 Monitoring Interrupt Enable Register 1 (CAMIE1)

Address(es): COMA.CAMIE1 874C 9454h, COMA_NS.CAMIE1 074C 9454h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PPFE21	PPFE20	PPFE11	PPFE10	PPFE01	PPFE00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PPFE00	Port 0 Pause Frame Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	PPFE01	Port 0 Pause Frame Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	PPFE10	Port 1 Pause Frame Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	PPFE11	Port 1 Pause Frame Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	PPFE20	Port 2 Pause Frame Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	PPFE21	Port 2 Pause Frame Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the CAMID1 register.

PPFEij Bit (Port i Pause Frame Enable j) (i = 0 to 2) (j = 0, 1)

[Setting Condition]

- Writing 1 to this bit will set it.

[Clearing Condition]

- SW: Writing 1 to the CAMID1.PPFDij bit will clear this bit.

35.2.6.6 Monitoring Interrupt Disable Register 1 (CAMID1)

Address(es): COMA.CAMID1 874C 9458h, COMA_NS.CAMID1 074C 9458h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PPFD21	PPFD20	PPFD11	PPFD10	PPFD01	PPFD00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PPFD00	Port 0 Pause Frame Disable 0	0: Has no effects. 1: Sets the CAMIE1.PPFE00 bit to 0.	R/W
b1	PPFD01	Port 0 Pause Frame Disable 1	0: Has no effects. 1: Sets the CAMIE1.PPFE01 bit to 0.	R/W
b2	PPFD10	Port 1 Pause Frame Disable 0	0: Has no effects. 1: Sets the CAMIE1.PPFE10 bit to 0.	R/W
b3	PPFD11	Port 1 Pause Frame Disable 1	0: Has no effects. 1: Sets the CAMIE1.PPFE11 bit to 0.	R/W
b4	PPFD20	Port 2 Pause Frame Disable 0	0: Has no effects. 1: Sets the CAMIE1.PPFE20 bit to 0.	R/W
b5	PPFD21	Port 2 Pause Frame Disable 1	0: Has no effects. 1: Sets the CAMIE1.PPFE21 bit to 0.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.3 Operations

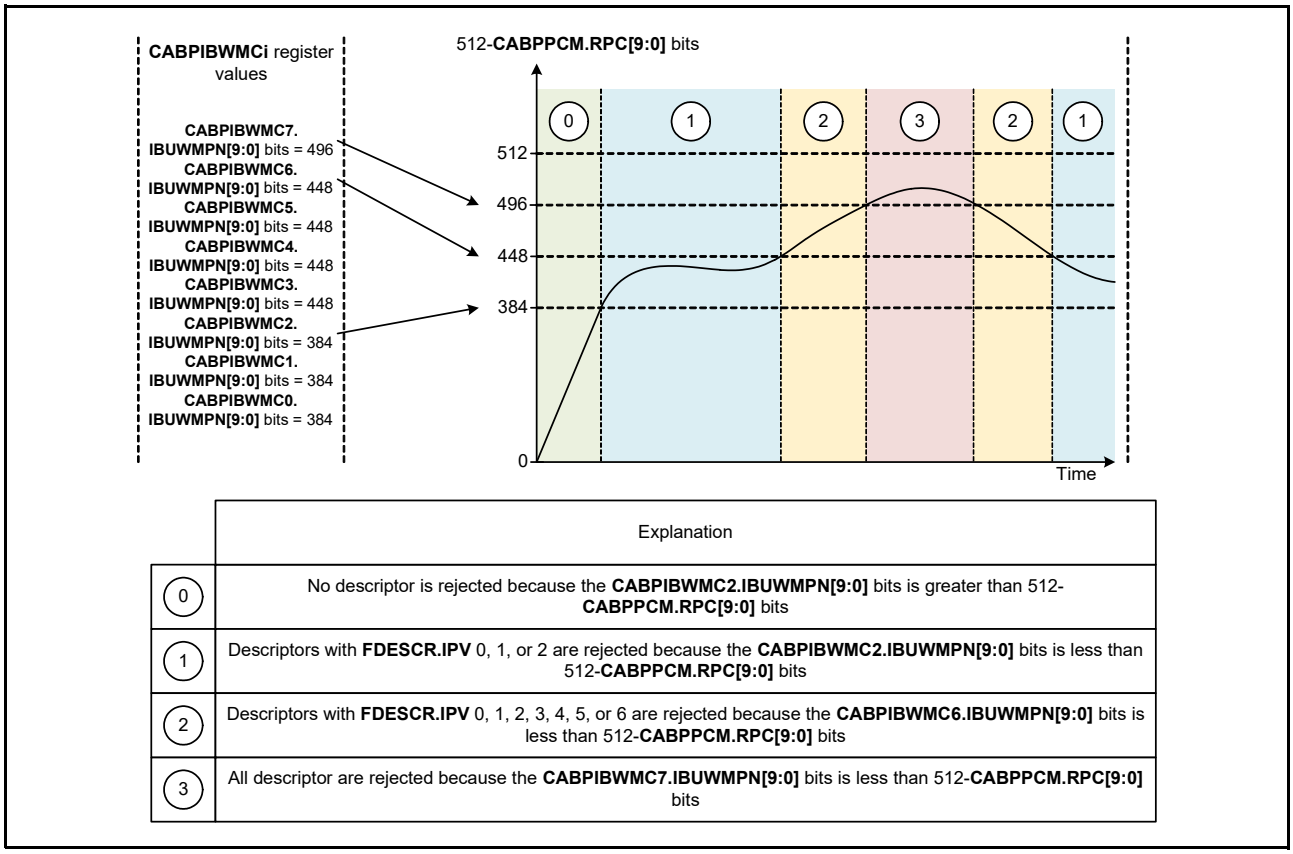
35.3.1 Watermark Function

If a ESWM overflow occurs, frames in the ESWM will be loss at the ingress part of agents [GWCA] [ETHA]. It will result in a random loss of data independently from the actual priority of frames. To avoid so, the watermark function allows to selectively reject frames in forwarding engine [MFWD] before a ESWM overflow occurs.

COMA includes three watermark functions, the IPV based watermark, the global level-based watermark and the per-port level-based watermark. The watermark functions aim at controlling the ESWM behavior in case of overflow by generating data loss. All watermark functions are running in parallel and a frame will be rejected if one or more of the watermark functions decide so.

35.3.1.1 IPV Based Watermark

The IPV based watermark ensure the minimum QoS requirement for a ESWM. It is enabled by the default configurations for a ESWM maximum frame size of 256 bytes. This function is controlled by the common agent, however, the data loss is performed by the forwarding engine [MFWD]. Depending on the combination with other functions, “IPV based watermark” is affected by “Queue pause” regardless of “Frame size(256)”.
Figure 35.2 gives an example for IPV based unsecure watermark setting (The IPV based secure watermark works the same way but using the CABPIBWMCi.IBSWMPN[9:0] bits). Note that 512 – CABPPCM.RPC[9:0] bits variations in Figure 35.2 are arbitrary and do not correspond to anything relevant.



	Explanation
0	No descriptor is rejected because the **CABPIBWMC2.IBUWMPN[9:0]** bits is greater than 512-**CABPPCM.RPC[9:0]** bits
1	Descriptors with **FDESCR.IPV** 0, 1, or 2 are rejected because the **CABPIBWMC2.IBUWMPN[9:0]** bits is less than 512-**CABPPCM.RPC[9:0]** bits
2	Descriptors with **FDESCR.IPV** 0, 1, 2, 3, 4, 5, or 6 are rejected because the **CABPIBWMC6.IBUWMPN[9:0]** bits is less than 512-**CABPPCM.RPC[9:0]** bits
3	All descriptor are rejected because the **CABPIBWMC7.IBUWMPN[9:0]** bits is less than 512-**CABPPCM.RPC[9:0]** bits

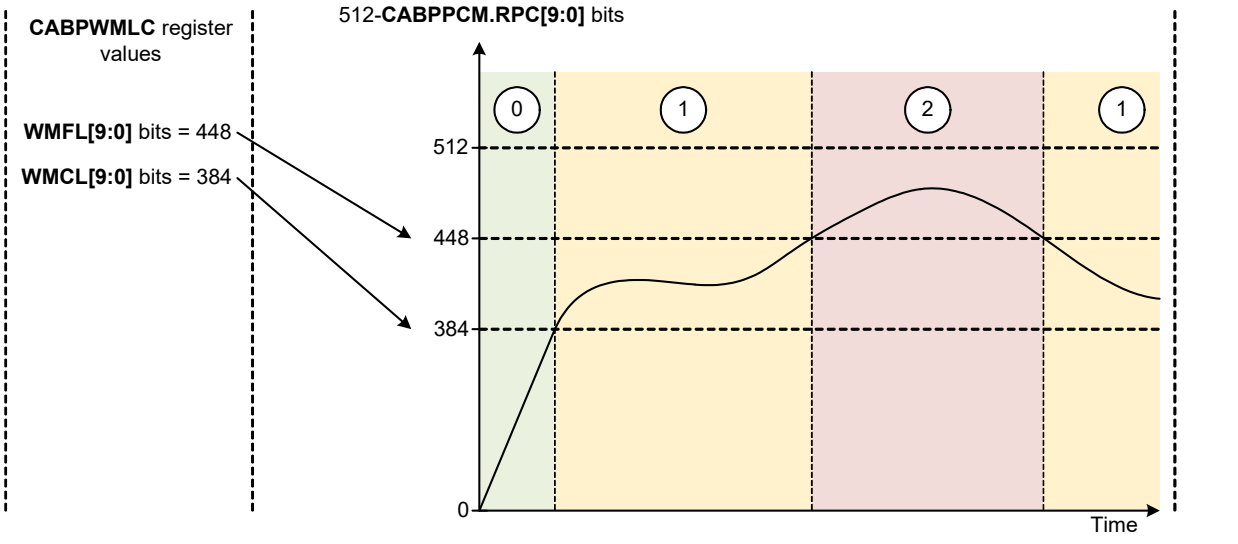
Figure 35.2 Example for IPV Based Watermark Setting

Note: FDESCR.IPV refers to the internal priority value given to a frame by forwarding engine in the forwarding descriptor but before filtering. For more details, refer to forwarding engine specification document [MFWD].

35.3.1.2 Global Level-Based Watermark

The global level-based watermark aims at rejecting frames with targeted IPVs depending on the number of pointers are currently used by the ESWM (the CABPPCM.RPC[9:0] bits). This function is controlled by the common agent (through WM.CRITICAL[2:0] and WM.FLUSH[2:0]), however, the data loss is performed by the forwarding engine [MFWD]. Note that WM.CRITICAL[2:0] and WM.FLUSH[2:0] interfaces can also be set by the per-port level-based watermark (one of these pins will be set if the per-port level-based watermark OR the global level-based watermark is in the required conditions to set it).

Figure 35.3 gives an example for global level-based watermark setting. Note that 512-CABPPCM.RPC[9:0] bits variations in Figure 35.3 are arbitrary and do not correspond to anything relevant.



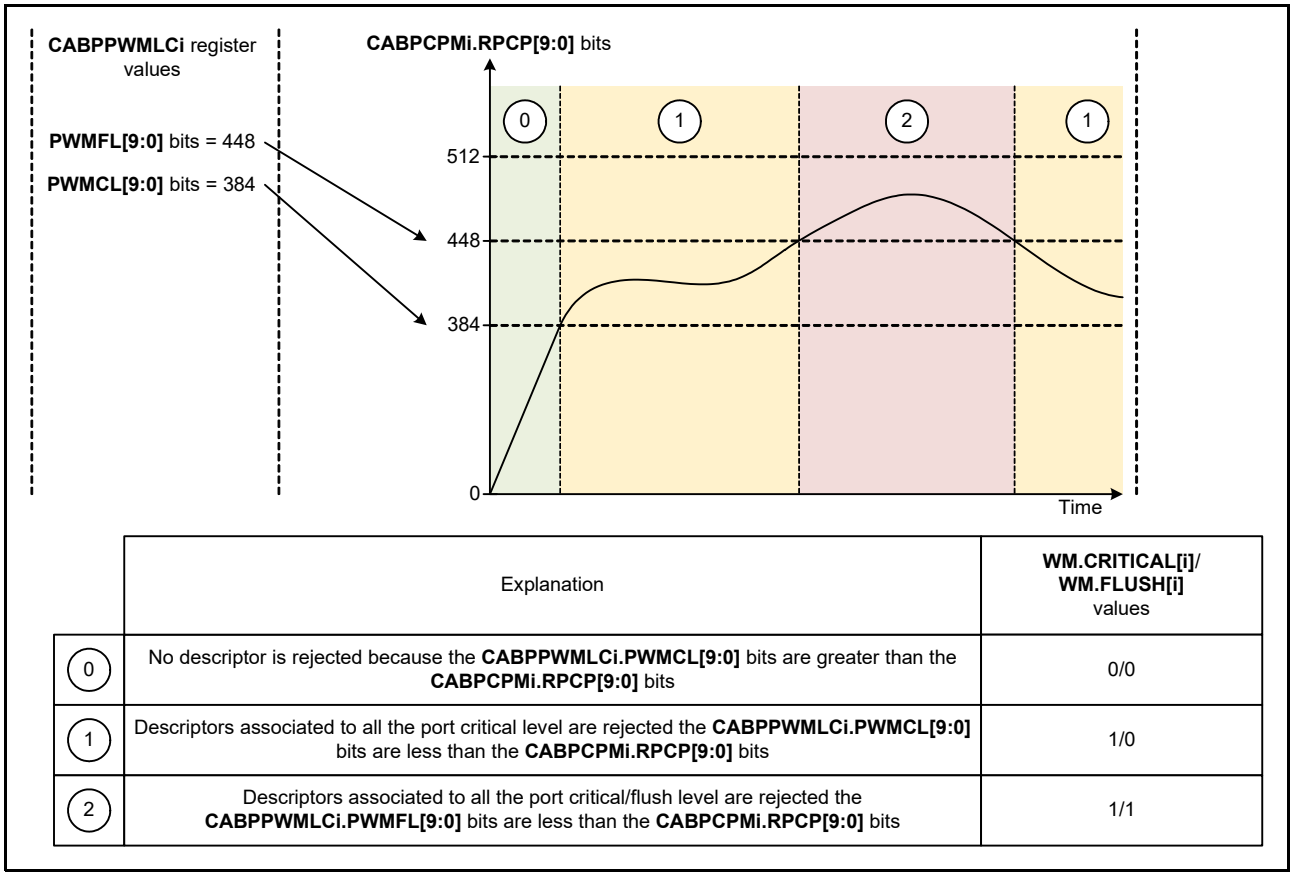
	Explanation	WM.CRITICAL[i]/ WM.FLUSH[i] values
0	No descriptor is rejected because the **CABPWMLC.WMCL[9:0]** bits are greater than 512-**CABPPCM.RPC[9:0]** bits	All0/All0
1	Descriptors associated to all the port critical level are rejected because 512-**CABPPCM.RPC[9:0]** bits are greater than the **CABPWMLC.WMCL[9:0]** bits	All1/All0
2	Descriptors associated to all the port critical/flush level are rejected because 512-**CABPPCM.RPC[9:0]** bits are greater than the **CABPWMLC.WMFL[9:0]** bits	All1/All1

Figure 35.3 Example for Global Level-Based Watermark Setting

35.3.1.3 Per-Port Level-Based Watermark

The per-port level-based watermark aims at rejecting frames with targeted IPVs depending on the CABPPWMLCi.PWMCL[9:0] bits. This function is controlled by the common agent (through WM.CRITICAL[i] and WM.FLUSH[i]), however, the data loss is performed by the forwarding engine [MFWD]. Note that WM.CRITICAL[i] and WM.FLUSH[i] interfaces can also be set by the Global level-based watermark (one of these pins will be set if the per-port level-based watermark OR the global level-based watermark is in the required conditions to set it).

Figure 35.4 gives an example for per-port level-based watermark setting for port i. Note that the CABPCPMi.RPCP[9:0] bits variations in Figure 35.4 are arbitrary and do not correspond to anything relevant.



	Explanation	WM.CRITICAL[i]/ WM.FLUSH[i] values
0	No descriptor is rejected because the **CABPPWMLCi.PWMCL[9:0]** bits are greater than the **CABPCPMi.RPCP[9:0]** bits	0/0
1	Descriptors associated to all the port critical level are rejected the **CABPPWMLCi.PWMCL[9:0]** bits are less than the **CABPCPMi.RPCP[9:0]** bits	1/0
2	Descriptors associated to all the port critical/flush level are rejected the **CABPPWMLCi.PWMFL[9:0]** bits are less than the **CABPCPMi.RPCP[9:0]** bits	1/1

Figure 35.4 Example for Per-Port Level-Based Watermark Setting for Port i

35.3.2 Pause Function

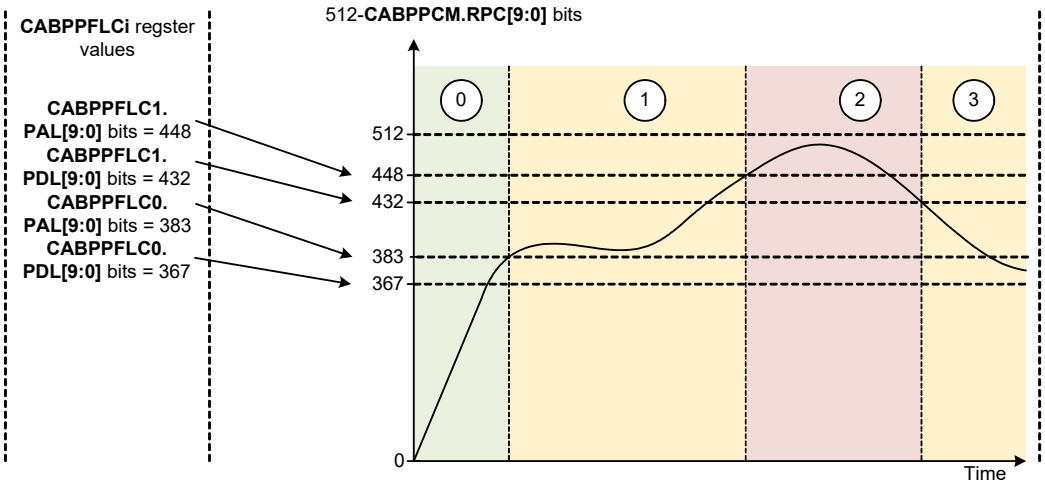
If the ESWM overflow occurs, frames in the ESWM will be loss at the ingress part of agents [GWCA] [ETHA]. It will result in a random loss of data independently from the actual priority of frames. To avoid so, the pause function allows to selectively pause frame injection by agents [GWCA] [ETHA].

COMA includes two pause functions, the global pause function and the per-port pause. The pause functions aim at controlling the ESWM behavior in case of overflow by generating backpressure. All pause functions are running in parallel and backpressure will be applied if one or more of the pause functions decide so.

35.3.2.1 Global Pause Function

The global pause function aims at applying backpressure on agents for targeted IPVs depending on the CABPPCM.RPC [9:0] bits. The pause function can pause agents with 2 combinations of IPVs.
This function is controlled by the common agent PAUSE[2:0][1:0] interface, however, the back pressure is performed by the agents [ETHA] [GWCA]. Note that PAUSE interface can also be set by the per-port pause function (one of these pins will be set if the per-port pause function OR the global pause function is in the required conditions to set it).

Figure 35.5 gives an example for global pause function setting. Note that 512-CABPPCM.RPC[9:0] bits variations in Figure 35.5 are arbitrary and do not correspond to anything relevant.



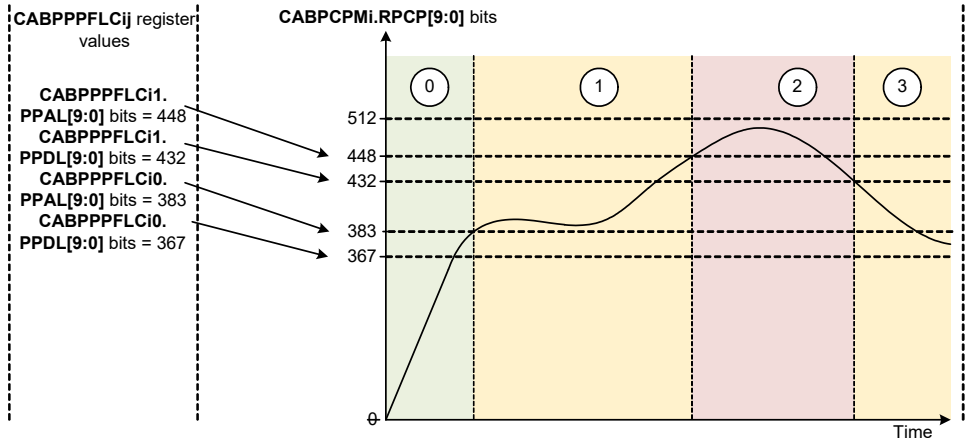
	Explanation	PAUSE[i][0]/ PAUSE[i][1] values (for all I = 1..PORT_N-1)
0	No agent paused because 512-**CABPPCM.RPC[9:0]** bits never exceeded the **CABPPFLC0.PAL[9:0]** bits level nor the **CABPPFLC1.PAL[9:0]** bits level	0/0
1	Agents are paused with level 0 because 512-**CABPPCM.RPC[9:0]** bits exceeded the **CABPPFLC0.PAL[9:0]** bits level and not the **CABPPFLC1.PAL[9:0]** bits level	1/0
2	Agents are paused with level 0 and 1 because 512-**CABPPCM.RPC[9:0]** bits exceeded the **CABPPFLC0.PAL[9:0]** bits level and the **CABPPFLC1.PAL[9:0]** bits level	1/1
3	Agents are paused with level 0 because 512-**CABPPCM.RPC[9:0]** bits exceeded the **CABPPFLC0.PAL[9:0]** bits and dropped under the **CABPPFLC1.PDL[9:0]** bits	1/0

Figure 35.5 Example for Global Pause Function Setting

35.3.2.2 Per-Port Pause Function

The per-port function aims at applying backpressure on agents for targeted IPVs depending on the CABPPPFLCij.PPAL[9:0] bits and PPDL[9:0] bits. The pause function can pause agents with 2 combinations of IPVs. This function is controlled by the common agent PAUSE[i] interface, however, the back pressure is performed by the agents [ETHA] [GWCA]. Note that PAUSE interface can also be set by the global pause function (one of these pins will be set if the per-port pause function OR the global pause function is in the required conditions to set it).

Figure 35.6 gives an example for per-port pause function setting for port i. Note that the CABPCPMi.RPCP[9:0] bits variations in Figure 35.6 are arbitrary and do not correspond to anything relevant.



	Explanation	PAUSE[i][0]/ PAUSE[i][0] values
0	No agent paused because the **CABPCPMi.RPCP[9:0]** bits never exceeded the **CABPPPFLCi0.PPAL[9:0]** bits level nor the **CABPPPFLCi1.PPAL[9:0]** bits level	0/0
1	Agents are paused with level 0 because the **CABPCPMi.RPCP[9:0]** bits exceeded the **CABPPPFLCi0.PPAL[9:0]** bits level and not the **CABPPPFLCi1.PPAL[9:0]** bits level	1/0
2	Agents are paused with level 0 and level 1 because the **CABPCPMi.RPCP[9:0]** bits exceeded the **CABPPPFLCi0.PPAL[9:0]** bits level and the **CABPPPFLCi1.PPAL[9:0]** bits level	1/1
3	Agents are paused with level 0 because the **CABPCPMi.RPCP[9:0]** bits exceeded the **CABPPPFLCi0.PPAL[9:0]** bits and dropped under the **CABPPPFLCi1.PPDL[9:0]** bits	1/0

Figure 35.6 Example for Per-Port Pause Function Setting for Port i

35.3.3 Per Port Memory Allocation Function

It is possible to control the maximum and the minimum number of pointers that can be used per port thanks to the CABPULCi.MNNPN[9:0] bits and the CABPULCi.MXNPN[9:0] bits. Any agent will be able to use at least its minimum of pointers and at most its maximum number of pointers. In this section are given some setting examples for the memory allocation function.

35.3.3.1 Shared Memory Setting

In shared memory, the data RAM is shared by all the ports. It is the reset state of the common agent.
Figure 35.7 describes the shared memory setting.

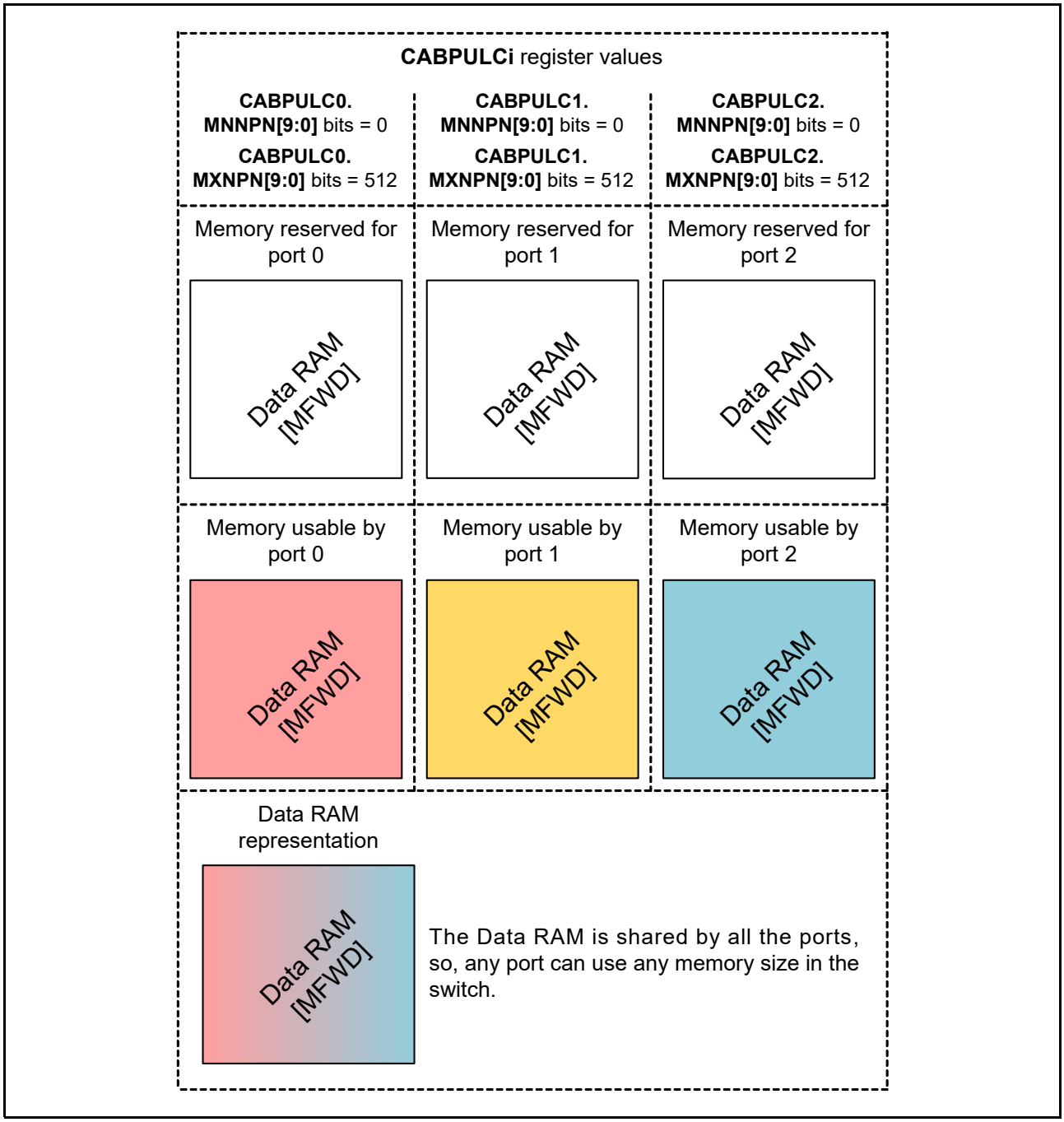


Figure 35.7 Shared Memory Setting

35.3.3.2 Reduced Memory Setting 1

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they are able to make the ESWM overflow (especially for CPU ports [GWCA]). To set this mode, the CABPULCi.MXNPN[9:0] bits should be used.
Follow figure describes one example of reduced memory setting. For example, if port 2 is a fast CPU port, the CABPULCi.MXNPN[9:0] bits can limit the number of reception frames.

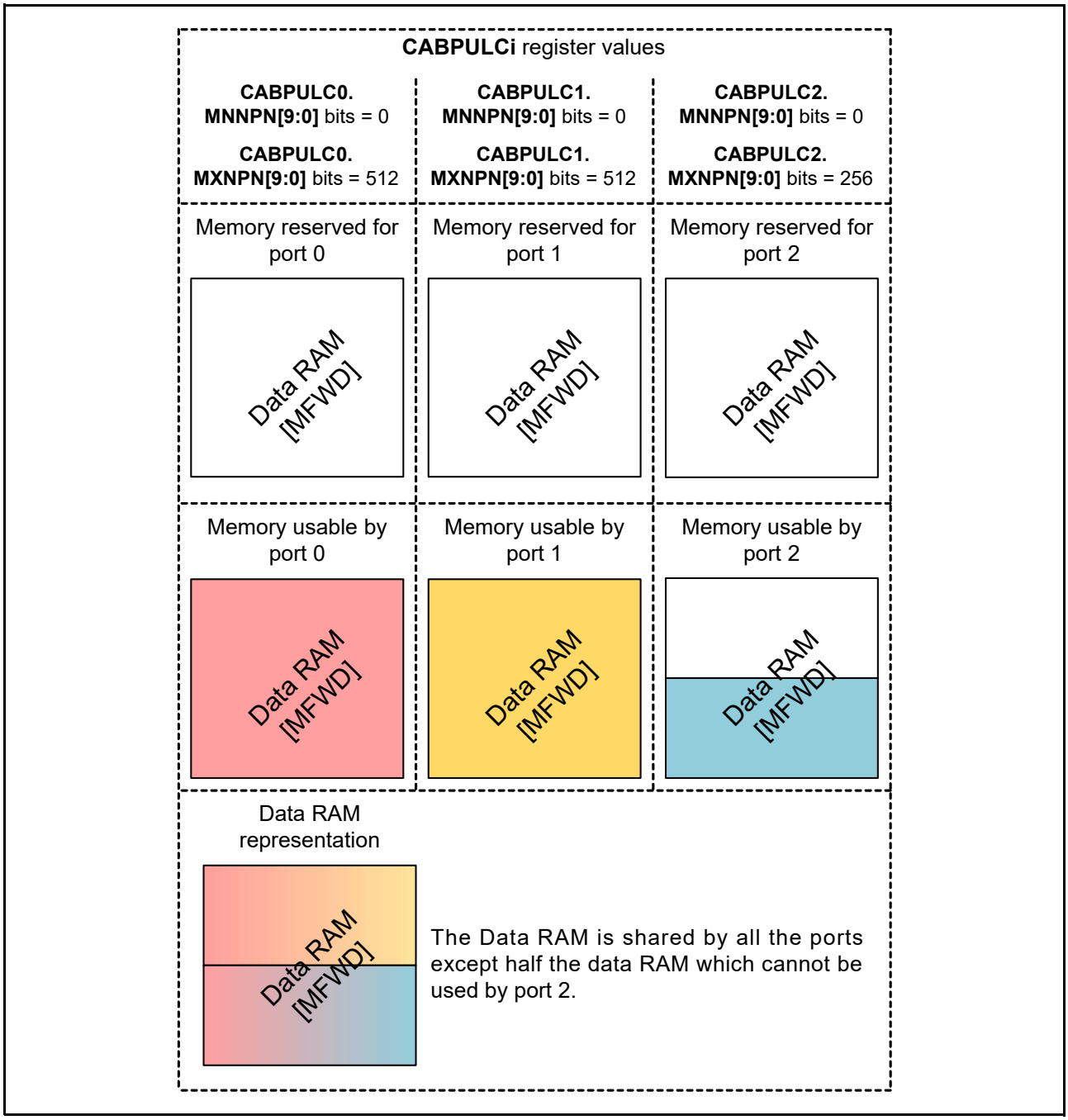


Figure 35.8 Reduced Memory Setting Example (1)

35.3.3.3 Reduced Memory Setting 2

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they are able to make the ESWM overflow (especially for CPU ports [GWCA]). To set this mode, the CABPULCi.MNNPN[9:0] bits should be used.
Follow figure describes one example of reduced memory setting. For example, if port 0 and 1 are a slow ether port, the CABPULCi.MNNPN[9:0] bits can protect the number of frame buffer.

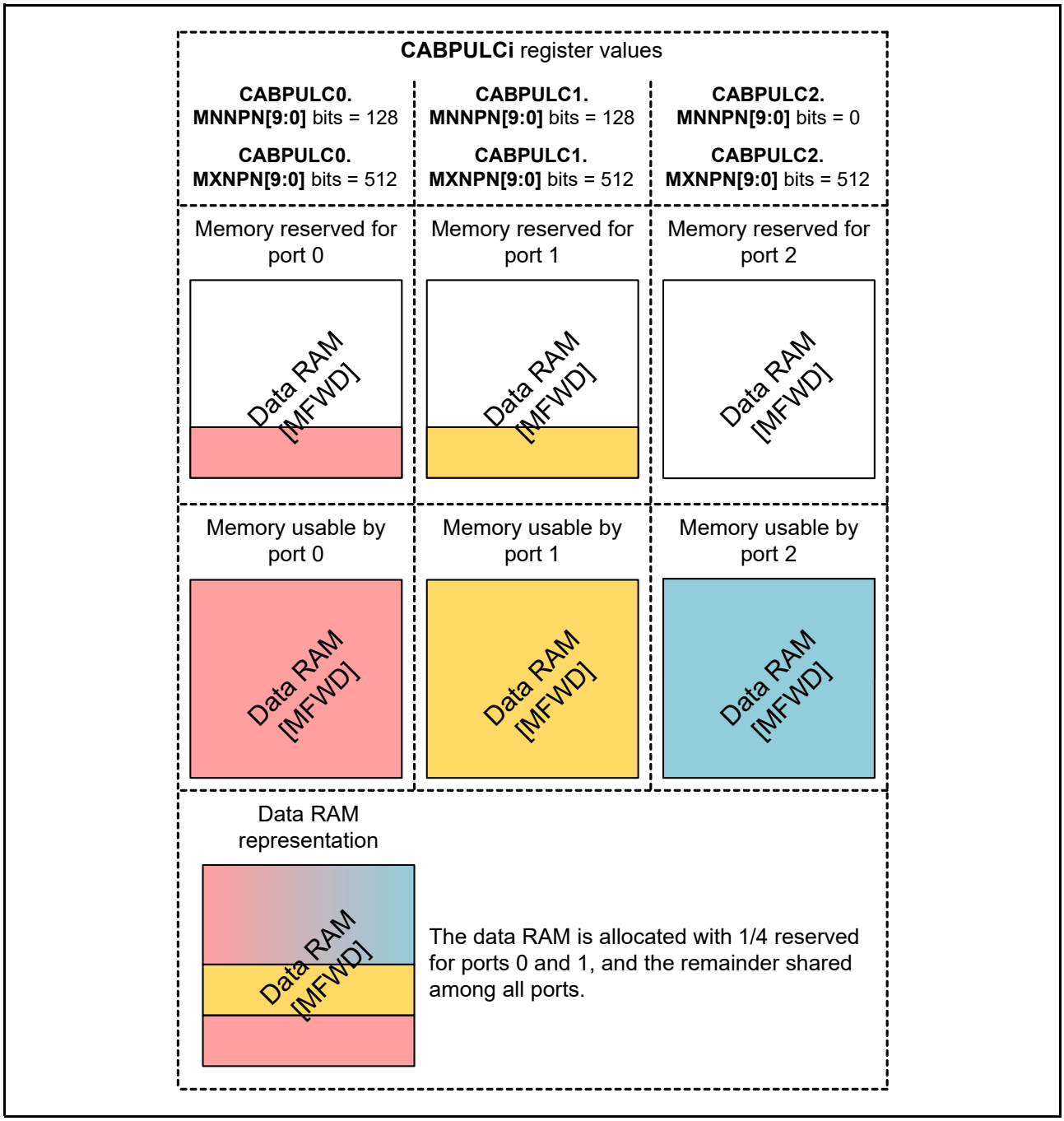


Figure 35.9 Reduced Memory Setting Example (2)

35.3.3.4 Hybrid Memory Setting

All the setting described in section 35.3.3.1 and 35.3.3.2 can be used together to elaborate a complex memory. It is the hybrid memory setting.

Figure 35.10 describes one example of hybrid memory setting.

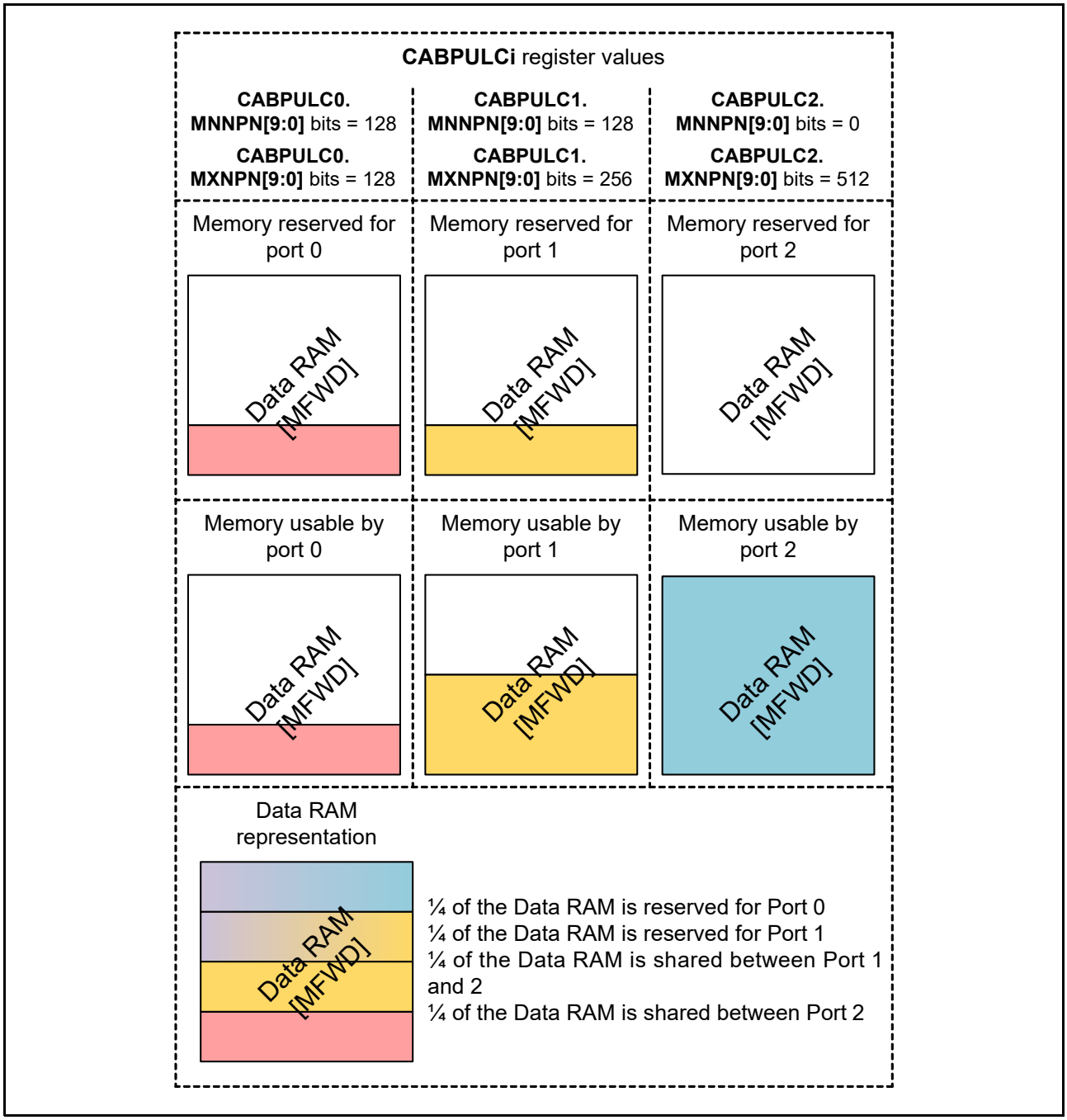


Figure 35.10 Hybrid Memory Setting Example

35.3.4 Software Flows

35.3.4.1 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except RESET mode.
The interrupt handling flow is described in Figure 35.11.

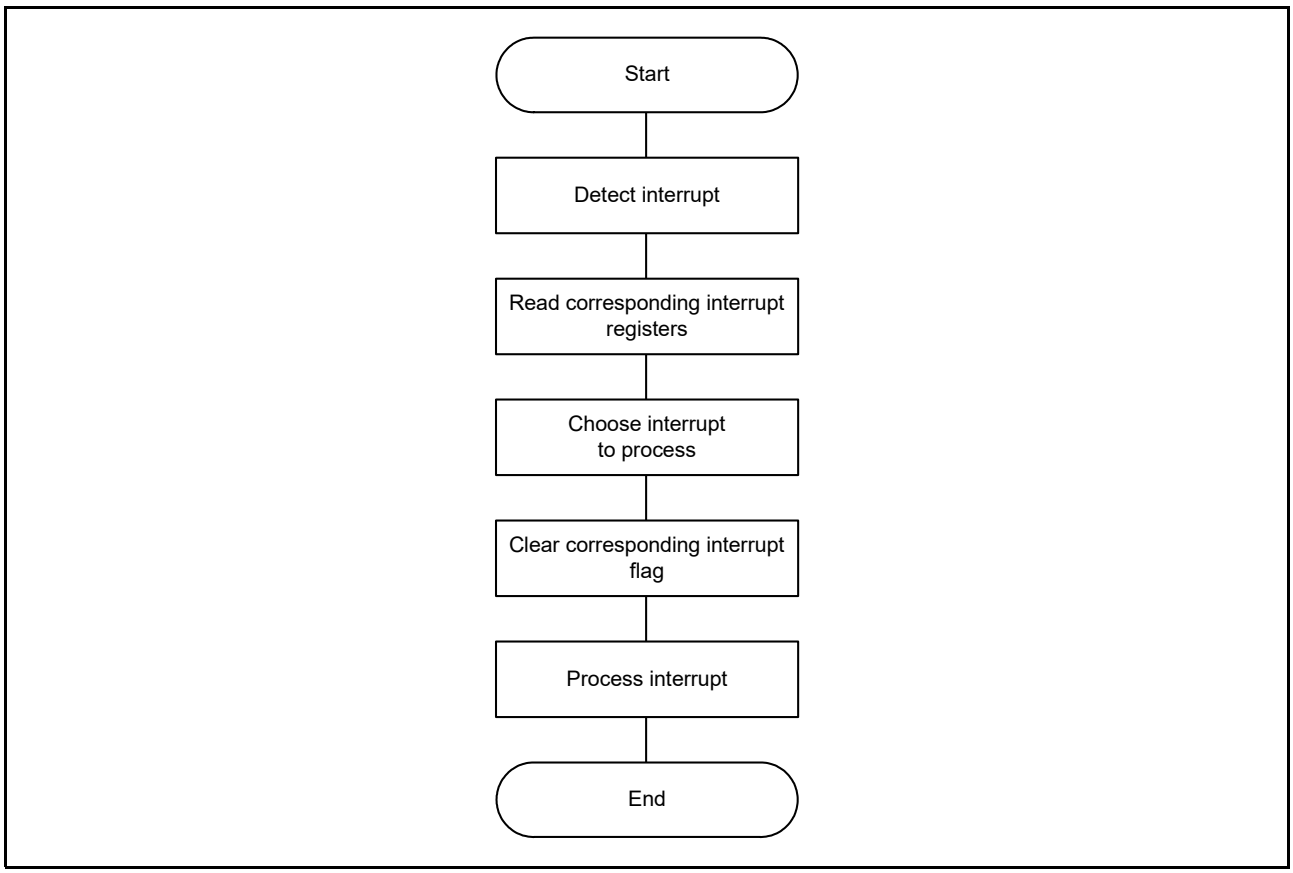


Figure 35.11 Interrupt Handling Flow

35.3.4.2 Called Software Flows

The flows described in this section can only be called from other flows and cannot be used alone.

(1) Buffer Pool Initialization Flow

The buffer pool initialization flow is described in Figure 35.12.

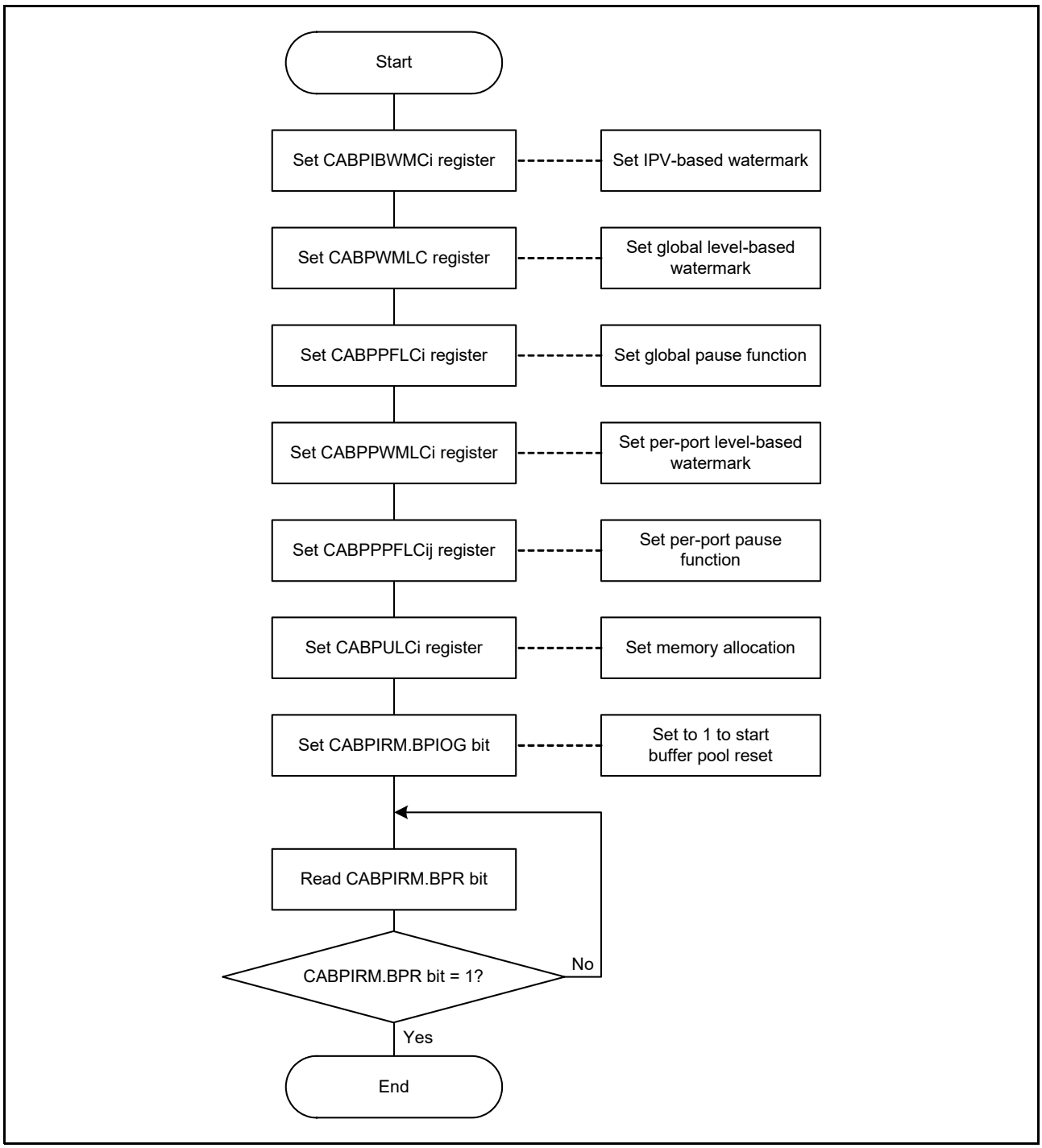


Figure 35.12 Buffer Pool Initialization Flow

(2) Software Reset Flow

The ESWM reset flow is described in Figure 35.13.

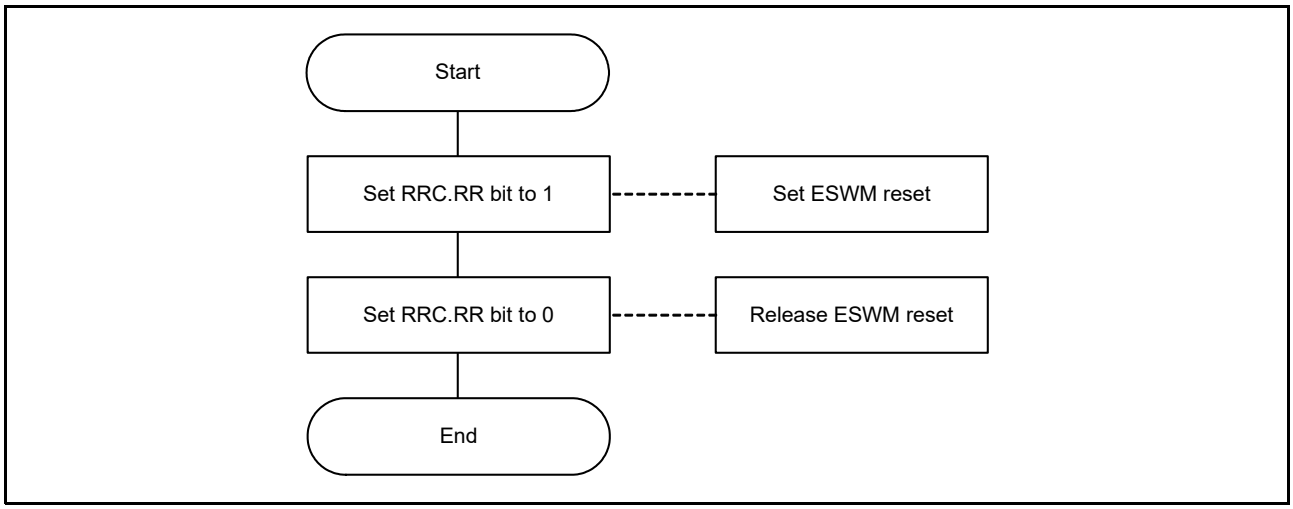


Figure 35.13 ESWM Reset Flow

(3) Agent i Clock Control Flows (i = 0 to 2)

The agent i clock flows are described in Figure 35.14, Figure 35.15, and Figure 35.16.

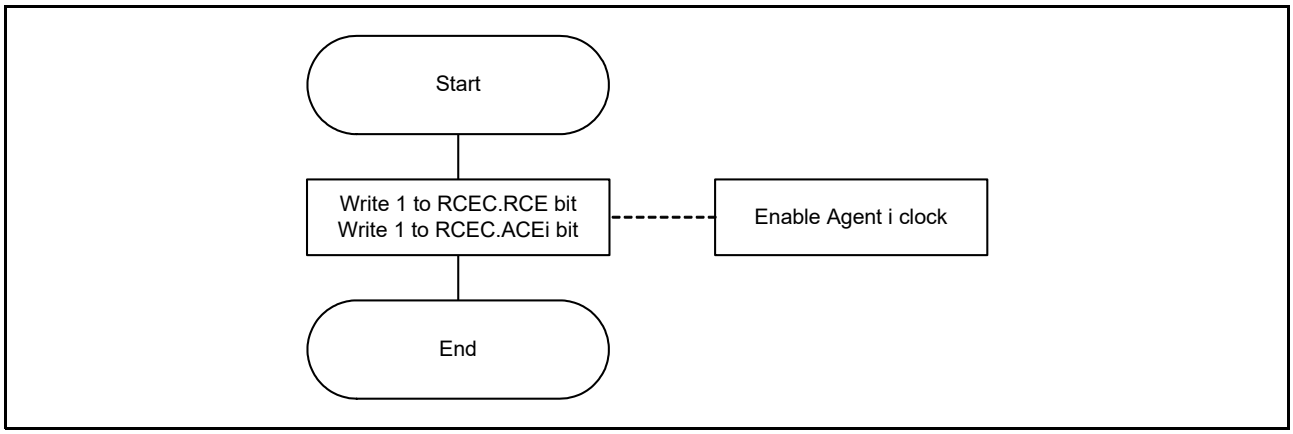


Figure 35.14 Agent i Clock Enable Flow

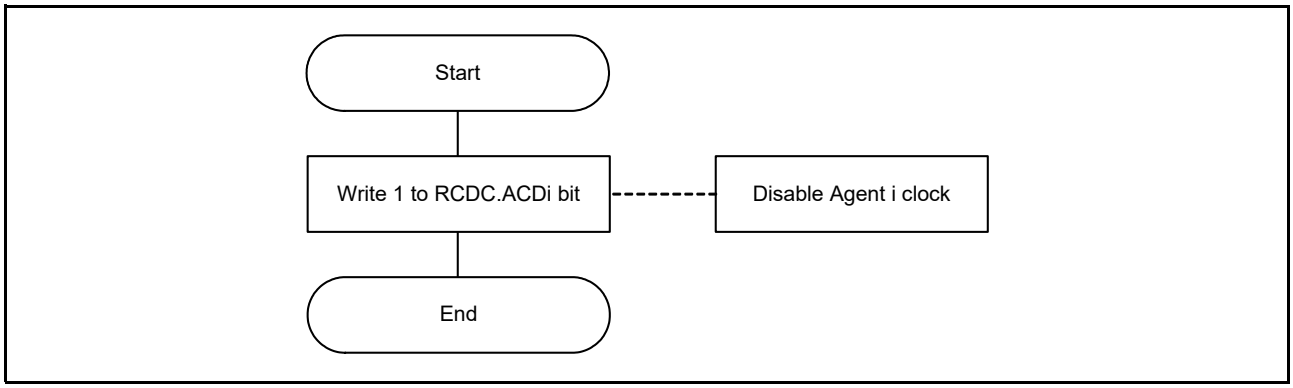


Figure 35.15 Agent i Clock Disable Flow

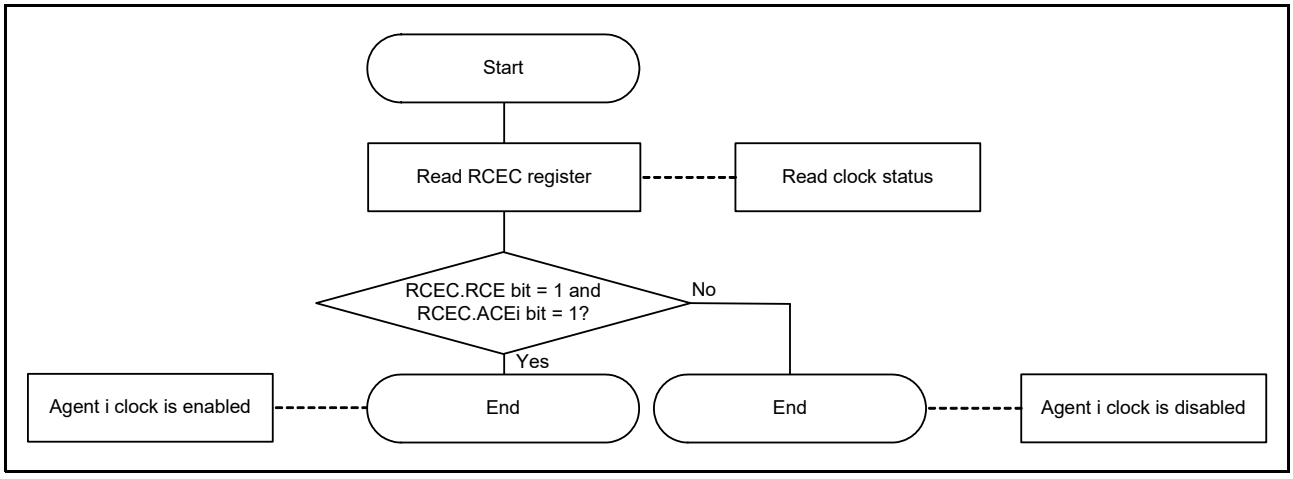


Figure 35.16 Agent i Clock Status Check Flow

(4) ESWCLK Clock Control Flows

The ESWCLK clock flows are described in Figure 35.17 and Figure 35.18.

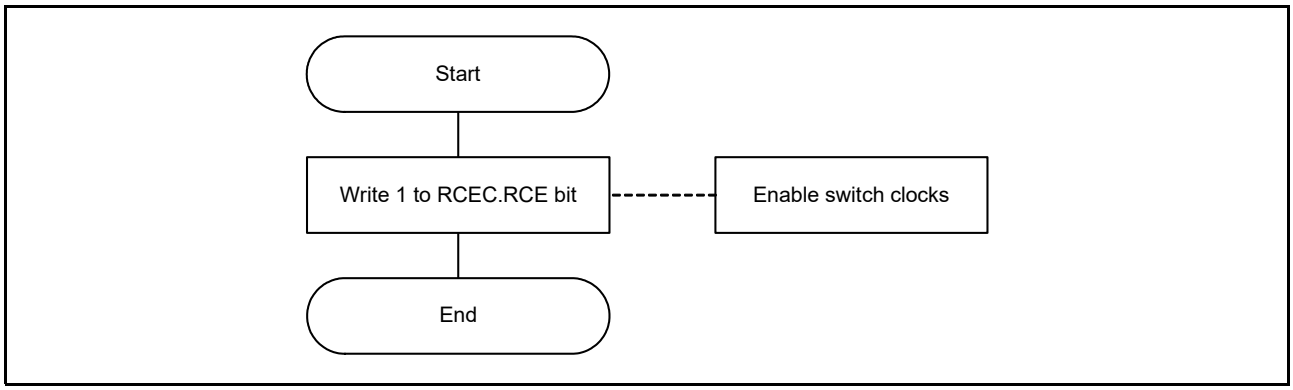


Figure 35.17 ESWCLK Clock Enable Flow

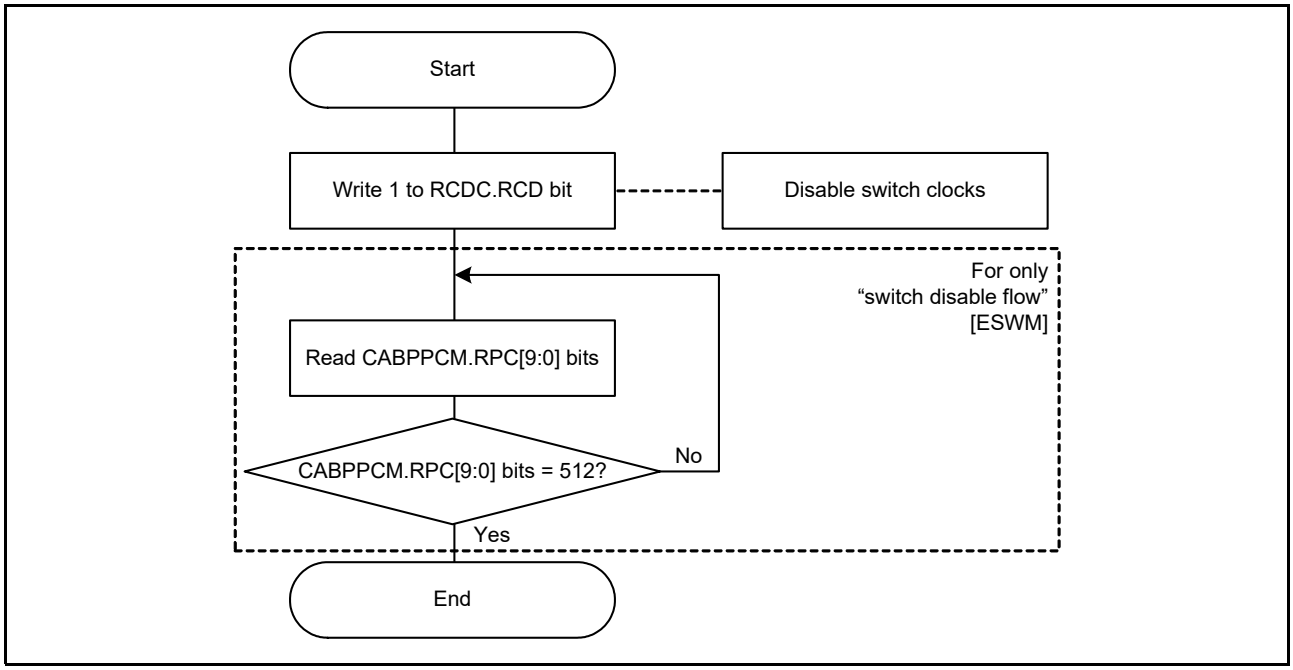


Figure 35.18 ESWCLK Clock Disable Flow

36. Ethernet Agent (ETHA)

36.1 Overview

The ethernet agent consists of an agent interface module to allowing communication within the ESWM. It handles the data exchange between the ESWM and an Ethernet PHY.

36.1.1 Features

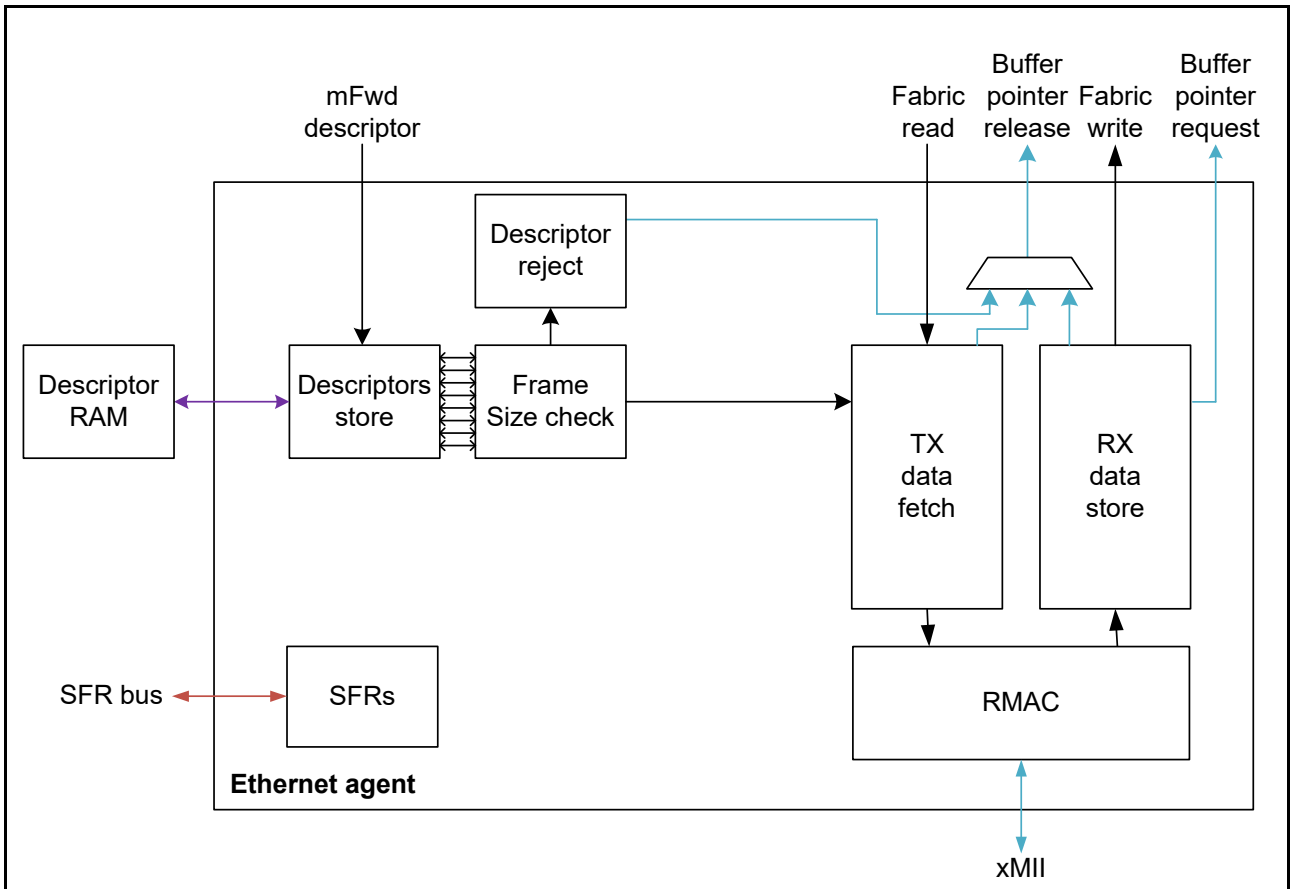
Ethernet agent Features are described Table 36.1.

Table 36.1 Ethernet Agent Feature List

Function		Details
Interfaces	Clock/Reset interface	Clock/Reset interface [ESWM]
	Scan interface	Scan interface [ESWM]
	Switch mode	Interface to receive the switch mode [MFWD]
	Pause interface (PAUSE)	Interface to pause TX queues [COMA]
	xMII	GMII 100 Mbps, 1 Gbps [RMAC]
	SFR interface	Interface to access GWCA SFRs [COMA]
	Interrupts	GWCA interrupt to CPU
	Fabric interface	Interface to communicate with the data, TAG and pointer RAM [MFWD].
	Descriptor bus	Interface to receive frame information to send them to CPU [MFWD].
	BP Request	Interface to receive BPs to store frames in the data, TAG and pointer RAM [COMA].
	BP Release	Interface to release the BPs that has been used [COMA].
	TX Timestamp Capture interface	Interface to send TX timestamps to CPU [RMAC].
	RAM interfaces	Interface to communication with RAMs [ESWM]

36.1.2 Ethernet Agent Block Diagram

Figure 36.1 shows ethernet agent block diagram.



RX data store : Transmits frames from RMAC to the local RAM
TX data fetch : Transmits frame from local RAM to RMAC
Descriptor store : Receives the descriptors from forwarding engine and controls the priority between descriptors
Frame size check : Check the frame size
Descriptor reject : Reject descriptors
RMAC : Mac interface [RMAC]
SFRs : ETHA SFRs

Figure 36.1 Ethernet Agent Block Diagram

36.2 Register Description

36.2.1 ETHA Mode Function Registers

36.2.1.1 Mode Configuration Register (EAMC)

Address(es): ETHA0.EAMC 874C A000h, ETHA0_NS.EAMC 074C A000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OPC[1:0]	Operating Mode Command	b1 b0 0 0: Enter RESET mode 0 1: Enter DISABLE mode 1 0: Enter CONFIG mode 1 1: Enter OPERATION mode	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

OPC[1:0] Bits (Operating Mode Command)

Used to set ETHA mode.

[Restriction]

- HW: This register is not writable if its value is different than EAMS.OPS[1:0].

36.2.1.2 Mode Status Register (EAMS)

Address(es): ETHA0.EAMS 874C A004h, ETHA0_NS.EAMS 074C A004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OPS[1:0]	Operating Mode Status Flag	b1 b0 0 0: RESET mode 0 1: DISABLE mode 1 0: CONFIG mode 1 1: OPERATION mode	R
b31 to b2	—	Reserved	These bits are read as 0.	R

OPS[1:0] Flag (Operating Mode Status Flag)

Indicates the current ETHA mode.

[Update conditions]

- HW: Mode change completed.
- HW: This register is changed at “Maximum frame communication time (exp: 64 Kbyte/100 Mbps = 5.12 ms)” from OPERATION to DISABLE.

36.2.2 Transmission Function Registers

36.2.2.1 IPV Remapping Configuration Register [802.1Q] (EAIRC)

Address(es): ETHA0.EAIRC 874C A010h, ETHA0_NS.EAIRC 074C A010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	IPVR7[2:0]			—	IPVR6[2:0]			—	IPVR5[2:0]			—	IPVR4[2:0]		
Value after reset:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IPVR3[2:0]			—	IPVR2[2:0]			—	IPVR1[2:0]			—	IPVR0[2:0]		
Value after reset:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IPVR0[2:0]	IPV Remapping 0	Configure to which descriptor queue descriptor received with IPV 0 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 0).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6 to b4	IPVR1[2:0]	IPV Remapping 1	Configure to which descriptor queue descriptor received with IPV 1 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 1).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b10 to b8	IPVR2[2:0]	IPV Remapping 2	Configure to which descriptor queue descriptor received with IPV 2 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 2).	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14 to b12	IPVR3[2:0]	IPV Remapping 3	Configure to which descriptor queue descriptor received with IPV 3 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 3).	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b18 to b16	IPVR4[2:0]	IPV Remapping 4	Configure to which descriptor queue descriptor received with IPV 4 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 4).	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b22 to b20	IPVR5[2:0]	IPV Remapping 5	Configure to which descriptor queue descriptor received with IPV 5 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 5).	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26 to b24	IPVR6[2:0]	IPV Remapping 6	Configure to which descriptor queue descriptor received with IPV 6 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 6).	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b30 to b28	IPVR7[2:0]	IPV Remapping 7	Configure to which descriptor queue descriptor received with IPV 7 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 7).	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

36.2.2.2 TX Descriptor Queue Security Configuration Register (EATDQSC)

Address(es): ETHA0.EATDQSC 874C A014h, ETHA0_NS.EATDQSC 074C A014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TDQSL7	TDQSL6	TDQSL5	TDQSL4	TDQSL3	TDQSL2	TDQSL1	TDQSL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDQSL0	TX Descriptor Queue 0 Security Level	0: Queue i unsecure 1: Queue i secure (i = 0 to 7)	R/W
b1	TDQSL1	TX Descriptor Queue 1 Security Level		R/W
b2	TDQSL2	TX Descriptor Queue 2 Security Level		R/W
b3	TDQSL3	TX Descriptor Queue 3 Security Level		R/W
b4	TDQSL4	TX Descriptor Queue 4 Security Level		R/W
b5	TDQSL5	TX Descriptor Queue 5 Security Level		R/W
b6	TDQSL6	TX Descriptor Queue 6 Security Level		R/W
b7	TDQSL7	TX Descriptor Queue 7 Security Level		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TDQSLi Bit (TX Descriptor Queue i Security Level) (i = 0 to 7)

When a queue is secured, an unsecure descriptor cannot enter it (when a descriptor is received through Descriptor Bus [MFWD] with eha_descr_sec from forwarding engine with FDESCR.SEC [MFWD] equal to 0).

36.2.2.3 TX Descriptor Queue Configuration Register (EATDQC)

Address(es): ETHA0.EATDQC 874C A018h, ETHA0_NS.EATDQC 074C A018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TDQP7	TDQP6	TDQP5	TDQP4	TDQP3	TDQP2	TDQP1	TDQP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TDQD7	TDQD6	TDQD5	TDQD4	TDQD3	TDQD2	TDQD1	TDQD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDQD0	TX Descriptor Queue 0 Disable	0: Queue n enabled 1: Queue n disabled (n = 0 to 7)	R/W
b1	TDQD1	TX Descriptor Queue 1 Disable		R/W
b2	TDQD2	TX Descriptor Queue 2 Disable		R/W
b3	TDQD3	TX Descriptor Queue 3 Disable		R/W
b4	TDQD4	TX Descriptor Queue 4 Disable		R/W
b5	TDQD5	TX Descriptor Queue 5 Disable		R/W
b6	TDQD6	TX Descriptor Queue 6 Disable		R/W
b7	TDQD7	TX Descriptor Queue 7 Disable		R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	TDQP0	TX Descriptor Queue 0 Pause	0: Queue m active 1: Queue m paused (m = 0 to 7)	R/W
b17	TDQP1	TX Descriptor Queue 1 Pause		R/W
b18	TDQP2	TX Descriptor Queue 2 Pause		R/W
b19	TDQP3	TX Descriptor Queue 3 Pause		R/W
b20	TDQP4	TX Descriptor Queue 4 Pause		R/W
b21	TDQP5	TX Descriptor Queue 5 Pause		R/W
b22	TDQP6	TX Descriptor Queue 6 Pause		R/W
b23	TDQP7	TX Descriptor Queue 7 Pause		R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

TDQDn Bit (TX Descriptor Queue n Disable) (n = 0 to 7)

Avoid descriptor from forwarding engine [MFWD] to enter the corresponding queue. The Forwarding Engine will reject the corresponding descriptor.

If disabled queue data is not needed anymore, the EATMFSCn.MFS[15:0] bits can be set to 0 in order to avoid more frames to go to the RMAC.

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing conditions]

- SW: Writing 0 to this bit will clear it.
- HW: Going out of OPERATION mode will clear this bit (EAMC.OPC[1:0] ≠ 11b).

TDQPm Bit (TX Descriptor Queue m Pause) (m = 0 to 7)

Avoid frames to be sent to the RMAC by stopping descriptor fetching from the descriptor RAM.

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing conditions]

- SW: Writing 0 to this bit will clear it.
- HW: Going out of OPERATION mode will clear this bit (EAMC.OPC[1:0] ≠ 11b).

[Restriction]

- SW: Pausing a queue during a long time could result in switch overflow. In case of switch overflow [COMA] this register should be set to 0.

[Cautions]

- This register is used to stop a queue but forwarding engine [MFWD] will not stop sending descriptor to the corresponding queue. In this case the queue might overflow.

36.2.2.4 TX Descriptor Queue Arbitration Configuration Register (EATDQAC)

Address(es): ETHA0.EATDQAC 874C A01Ch, ETHA0_NS.EATDQAC 074C A01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDQA7[3:0]				TDQA6[3:0]				TDQA5[3:0]				TDQA4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDQA3[3:0]				TDQA2[3:0]				TDQA1[3:0]				TDQA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0 to b3	TDQA0[3:0]	TX Descriptor Queue 0 Arbitration	b3 b0 0 0 0 0: Queue 0 strict arbitration Others: Queue 0 WRR arbitration	R/W
b7 to b4	TDQA1[3:0]	TX Descriptor Queue 1 Arbitration	b7 b4 0 0 0 0: Queue 1 strict arbitration Others: Queue 1 WRR arbitration	R/W
b11 to b8	TDQA2[3:0]	TX Descriptor Queue 2 Arbitration	b11 b8 0 0 0 0: Queue 2 strict arbitration Others: Queue 2 WRR arbitration	R/W
b15 to b12	TDQA3[3:0]	TX Descriptor Queue 3 Arbitration	b15 b12 0 0 0 0: Queue 3 strict arbitration Others: Queue 3 WRR arbitration	R/W
b19 to b16	TDQA4[3:0]	TX Descriptor Queue 4 Arbitration	b19 b16 0 0 0 0: Queue 4 strict arbitration Others: Queue 4 WRR arbitration	R/W
b23 to b20	TDQA5[3:0]	TX Descriptor Queue 5 Arbitration	b23 b20 0 0 0 0: Queue 5 strict arbitration Others: Queue 5 WRR arbitration	R/W
b27 to b24	TDQA6[3:0]	TX Descriptor Queue 6 Arbitration	b27 b24 0 0 0 0: Queue 6 strict arbitration Others: Queue 6 WRR arbitration	R/W
b31 to b28	TDQA7[3:0]	TX Descriptor Queue 7 Arbitration	b31 b28 0 0 0 0: Queue 7 strict arbitration Others: Queue 7 WRR arbitration	R/W

TDQAi[3:0] Bits (TX Descriptor Queue i Arbitration) (i = 0 to 7)

For details, refer to section 38.2.2.4, RX Descriptor Queue Arbitration Control Register (GWRDQAC).

[Restrictions]

- SW: In hybrid arbitration mode, all the queues with an EATDQAC.TDQAi[3:0] value different than 0 should have a consecutive priority.
- SW: In all arbitration mode except strict priority arbitration mode, at least two queues should have an EATDQAC.TDQAi[3:0] different than 0000b.

36.2.2.5 Transmission Maximum Frame Size Configuration Register q (EATMFSCq) (q = 0 to 7)

Address(es): ETHA0.EATMFSC0 874C A040h, ETHA0.EATMFSC1 874C A044h, ETHA0.EATMFSC2 874C A048h, ETHA0.EATMFSC3 874C A04Ch, ETHA0.EATMFSC4 874C A050h, ETHA0.EATMFSC5 874C A054h, ETHA0.EATMFSC6 874C A058h, ETHA0.EATMFSC7 874C A05Ch, ETHA0_NS.EATMFSC0 074C A040h, ETHA0_NS.EATMFSC1 074C A044h, ETHA0_NS.EATMFSC2 074C A048h, ETHA0_NS.EATMFSC3 074C A04Ch, ETHA0_NS.EATMFSC4 074C A050h, ETHA0_NS.EATMFSC5 074C A054h, ETHA0_NS.EATMFSC6 074C A058h, ETHA0_NS.EATMFSC7 074C A05Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MFS[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MFS[15:0]	Maximum Frame Size	Maximum frame size for descriptor queue q.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

MFS[15:0] Bits (Maximum Frame Size)

All bigger frames will be rejected.

[Restriction]

- HW: The MFS[15:0] size comparison takes the frame size on the PHY interface.

36.2.2.6 Transmission Descriptor Queue Depth Configuration Register q (EATDQDCq) (q = 0 to 7)

Address(es): ETHA0.EATDQDC0 874C A060h, ETHA0.EATDQDC1 874C A064h, ETHA0.EATDQDC2 874C A068h, ETHA0.EATDQDC3 874C A06Ch, ETHA0.EATDQDC4 874C A070h, ETHA0.EATDQDC5 874C A074h, ETHA0.EATDQDC6 874C A078h, ETHA0.EATDQDC7 874C A07Ch, ETHA0_NS.EATDQDC0 074C A060h, ETHA0_NS.EATDQDC1 074C A064h, ETHA0_NS.EATDQDC2 074C A068h, ETHA0_NS.EATDQDC3 074C A06Ch, ETHA0_NS.EATDQDC4 074C A070h, ETHA0_NS.EATDQDC5 074C A074h, ETHA0_NS.EATDQDC6 074C A078h, ETHA0_NS.EATDQDC7 074C A07Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DQD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DQD[9:0]	Descriptor Queue Depth	Set the number of descriptors that can be contained in descriptor queue q.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

DQD[9:0] Bits (Descriptor Queue Depth)

For details, refer to section 38.2.2.7, Reception Descriptor Queue q Depth Configuration Register (GWRDQDCq) (q = 0 to 7).

[Restriction]

- SW: The sum of DQD[9:0] fields should always be smaller or equal to 512.

36.2.2.7 Transmission Descriptor Queue q Monitoring Register (EATDQMq) (q = 0 to 7)

Address(es): ETHA0.EATDQM0 874C A080h, ETHA0.EATDQM1 874C A084h, ETHA0.EATDQM2 874C A088h, ETHA0.EATDQM3 874C A08Ch, ETHA0.EATDQM4 874C A090h, ETHA0.EATDQM5 874C A094h, ETHA0.EATDQM6 874C A098h, ETHA0.EATDQM7 874C A09Ch, ETHA0_NS.EATDQM0 074C A080h, ETHA0_NS.EATDQM1 074C A084h, ETHA0_NS.EATDQM2 074C A088h, ETHA0_NS.EATDQM3 074C A08Ch, ETHA0_NS.EATDQM4 074C A090h, ETHA0_NS.EATDQM5 074C A094h, ETHA0_NS.EATDQM6 074C A098h, ETHA0_NS.EATDQM7 074C A09Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DNQ[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DNQ[9:0]	Descriptor Number in Queue	These bits indicate the current number of descriptors stored in TX Descriptor Queue q.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

DNQ[9:0] Bits (Descriptor Number in Queue)

All descriptor stored here will be rejected/dropped at the out of OPERATION.

[Increment condition]

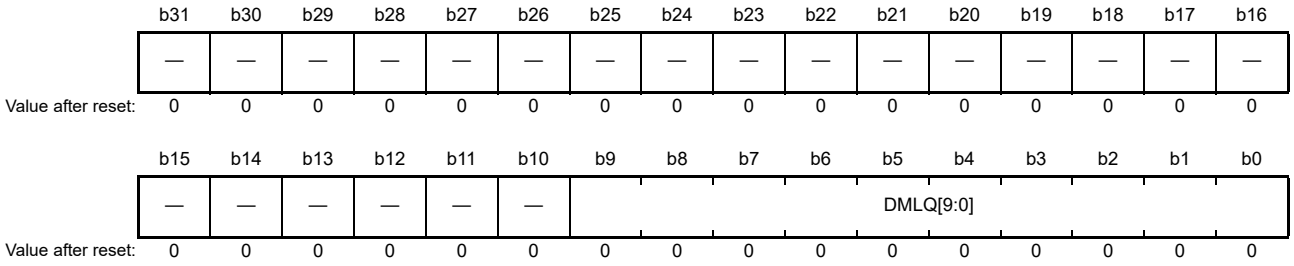
- HW: Incremented by 1 when a descriptor is received from the forwarding engine [MFWD] for the corresponding queue and, the queue is not full and there is no descriptor security error.

[Decrement condition]

- HW: Decremented by 1 when a descriptor is read by Ethernet Agent to send data to RMAC or to reject data.

36.2.2.8 Transmission Descriptor Queue q Max Level Monitoring Register (EATDQMLMq) (q = 0 to 7)

Address(es): ETHA0.EATDQMLM0 874C A0A0h, ETHA0.EATDQMLM1 874C A0A4h, ETHA0.EATDQMLM2 874C A0A8h, ETHA0.EATDQMLM3 874C A0ACh, ETHA0.EATDQMLM4 874C A0B0h, ETHA0.EATDQMLM5 874C A0B4h, ETHA0.EATDQMLM6 874C A0B8h, ETHA0.EATDQMLM7 874C A0BCh, ETHA0_NS.EATDQMLM0 074C A0A0h, ETHA0_NS.EATDQMLM1 074C A0A4h, ETHA0_NS.EATDQMLM2 074C A0A8h, ETHA0_NS.EATDQMLM3 074C A0ACh, ETHA0_NS.EATDQMLM4 074C A0B0h, ETHA0_NS.EATDQMLM5 074C A0B4h, ETHA0_NS.EATDQMLM6 074C A0B8h, ETHA0_NS.EATDQMLM7 074C A0BCh



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DMLQ[9:0]	Descriptor Max Level in Queue	These bits indicate the maximum number of descriptors that has been stored in TX Descriptor Queue q.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

DMLQ[9:0] Bits (Descriptor Max Level in Queue)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits (EAMC.OPS[1:0] = 00b).
- SW: By reading these bits, it is cleared to EATDQMq.DNQ[9:0].

[Increment condition]

- HW: Increments to EATDQMq.DNQ[9:0] value when smaller than EATDQMq.DNQ[9:0].

36.2.3 TAG Function Registers

36.2.3.1 VLAN Control Configuration Register (EAVCC)

Address(es): ETHA0.EAVCC 874C A130h, ETHA0_NS.EAVCC 074C A130h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	VEM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VIM	VLAN Ingress Mode	0: Incoming VLAN mode 1: Port based VLAN mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	VEM[2:0]	VLAN Egress Mode	b18 b16 0 0 0: No VLAN mode 0 0 1: C-TAG VLAN mode 0 1 0: HW C-TAG VLAN mode 0 1 1: SC-TAG VLAN mode 1 0 0: HW SC-TAG VLAN mode	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.3.2 VLAN TAG Configuration Register (EAVTC)

Address(es): ETHA0.EAVTC 874C A134h, ETHA0_NS.EAVTC 074C A134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STD	STP[2:0]			STV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTD	CTP[2:0]			CTV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	CTV[11:0]	C-TAG VLAN	Set the C-TAG VID values.	R/W
b14 to b12	CTP[2:0]	C-TAG PCP	Set the C-TAG PCP values.	R/W
b15	CTD	C-TAG DEI	Set the C-TAG DEI values.	R/W
b27 to b16	STV[11:0]	S-TAG VLAN	Set the S-TAG VID values.	R/W
b30 to b28	STP[2:0]	S-TAG PCP	Set the S-TAG PCP values.	R/W
b31	STD	S-TAG DEI	Set the S-TAG DEI values.	R/W

36.2.3.3 Reception TAG Filtering Configuration Register (EARTFC)

Address(es): ETHA0.EARTFC 874C A138h, ETHA0_NS.EARTFC 074C A138h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	UT	SCRT	SCT	CRT	CT	CSRT	CST	RT	NT
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NT	No Tag	0: No Tag frame passed 1: No Tag frame rejected	R/W
b1	RT	R-TAG	0: R-TAG frame passed 1: R-TAG frame rejected	R/W
b2	CST	CoS-TAG	0: CoS-TAG frame passed 1: CoS-TAG frame rejected	R/W
b3	CSRT	CoSR-TAG	0: CoSR-TAG frame passed 1: CoSR-TAG frame rejected	R/W
b4	CT	C-TAG	0: C-TAG frame passed 1: C-TAG frame rejected	R/W
b5	CRT	CR-TAG	0: CR-TAG frame passed 1: CR-TAG frame rejected	R/W
b6	SCT	SC-TAG	0: SC-TAG frame passed 1: SC-TAG frame rejected	R/W
b7	SCRT	SCR-TAG	0: SCR-TAG frame passed 1: SCR-TAG frame rejected	R/W
b8	UT	Unknown TAG	0: Unknown Tag frame passed 1: Unknown Tag frame rejected	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.4 ETHA Counter Registers

36.2.4.1 Under Minimum Frame Size Error Counter Register (EAUSMFSECN)

Address(es): ETHA0.EAUSMFSECN 874C A400h, ETHA0_NS.EAUSMFSECN 074C A400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USMFSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	USMFSEN[15:0]	Under Minimum Frame Size Error Number	These bits count the number of received data lost because of under switch minimum size error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

USMFSEN[15:0] Bits (Under Minimum Frame Size Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading this register clears it.

[Increment condition]

- HW: Incremented by 1 when a Under Switch Minimum Frame Size Error happens and if this register has a value different than FFFFh.

36.2.4.2 TAG Filtering Error Counter Register (EATFECN)

Address(es): ETHA0.EATFECN 874C A404h, ETHA0_NS.EATFECN 074C A404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TFEN[15:0]	TAG Filtering Error Number	These bits count the number of received data lost because of TAG filtering.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TFEN[15:0] Bits (TAG Filtering Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a TAG filter error happens and if these bits have a value different than FFFFh.

36.2.4.3 Frame Size Error Counter Register (EAFSECN)

Address(es): ETHA0.EAFSECN 874C A408h, ETHA0_NS.EAFSECN 074C A408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	FSEN[15:0]	Frame Size Error Number	These bits count the number of received data lost because of a Frame Size Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

FSEN[15:0] Bits (Frame Size Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Frame Size Error happens and if these bits has a value different than FFFFh.

36.2.4.4 Descriptor Queue Overflow Error Counter Register (EADQOECN)

Address(es): ETHA0.EADQOECN 874C A40Ch, ETHA0_NS.EADQOECN 074C A40Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DQOEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DQOEN[15:0]	Descriptor Queue Overflow Error Number	These bits count the number of transmit data lost because of a Descriptor Queue Overflow Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DQOEN[15:0] Bits (Descriptor Queue Overflow Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Overflow Error happens and if these bits have a value different than FFFFh.

36.2.4.5 Descriptor Queue Security Error Counter Register (EADQSECN)

Address(es): ETHA0.EADQSECN 874C A410h, ETHA0_NS.EADQSECN 074C A410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DQSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DQSEN[15:0]	Descriptor Queue Security Error Number	These bits count the number of transmit data lost because of a Descriptor Queue Security Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DQSEN[15:0] Bits (Descriptor Queue Security Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Security Error happens and if these bits have a value different than FFFFh.

36.2.5 Error Interrupt Registers

36.2.5.1 Error Interrupt Status Register 0 (EAEIS0)

Address(es): ETHA0.EAEIS0 874C A500h, ETHA0_NS.EAEIS0 074C A500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSES7	FSES6	FSES5	FSES4	FSES3	FSES2	FSES1	FSES0	—	TFES	USMFSES	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	USMFSES	Under Switch Minimum Frame Size Error Status Flag	0: No Under Switch Minimum Frame Size Error 1: Under Switch Minimum Frame Size Error occurred.	R/W
b6	TFES	TAG Filtering Error Status Flag	0: No TAG Filtering Error 1: TAG Filtering Error occurred.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	FSES0	Frame Size Error Status Flag 0	0: No Frame Size Error 0 1: Frame Size Error 0 occurred.	R/W
b9	FSES1	Frame Size Error Status Flag 1	0: No Frame Size Error 1 1: Frame Size Error 1 occurred.	R/W
b10	FSES2	Frame Size Error Status Flag 2	0: No Frame Size Error 2 1: Frame Size Error 2 occurred.	R/W
b11	FSES3	Frame Size Error Status Flag 3	0: No Frame Size Error 3 1: Frame Size Error 3 occurred.	R/W
b12	FSES4	Frame Size Error Status Flag 4	0: No Frame Size Error 4 1: Frame Size Error 4 occurred.	R/W
b13	FSES5	Frame Size Error Status Flag 5	0: No Frame Size Error 5 1: Frame Size Error 5 occurred.	R/W
b14	FSES6	Frame Size Error Status Flag 6	0: No Frame Size Error 6 1: Frame Size Error 6 occurred.	R/W
b15	FSES7	Frame Size Error Status Flag 7	0: No Frame Size Error 7 1: Frame Size Error 7 occurred.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

USMFSES Flag (Under Switch Minimum Frame Size Error Status Flag)

[Setting condition]

- HW: A frame smaller than 32 bytes has been received from RMAC.
  This error will happen with [RMAC] MEIS.RPOES.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to one of this flag will clear it.

[Error recovery]

- HW: The frame is discarded.
- SW: System dependent, cannot be defined here.

TFES Flag (TAG Filtering Error Status Flag)

[Setting condition]

- HW: An unauthorized TAG format has been detected. (Refer to section 36.3.2.1, RX Data Store).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to one of this flag will clear it.

[Error recovery]

- HW: Frame is sent to Forwarding Engine [MFWD] with local descriptor DESCR.TFE bit set (Refer to section 36.3.2.1, RX Data Store).
- SW: System dependent, cannot be defined here.

FSESq Flag (Frame Size Error Status Flag q) (q = 0 to 7)

[Setting condition]

- HW: FSESq is set when a frame bigger than EATMFSCq.MFS[15:0] has been received for descriptor queue q.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The current frame will be lost. Following data will be processed normally.
- SW: System dependent, cannot be defined here.

36.2.5.2 Error Interrupt Enable Register 0 (EAEIE0)

Address(es): ETHA0.EAEIE0 874C A504h, ETHA0_NS.EAEIE0 074C A504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSEE7	FSEE6	FSEE5	FSEE4	FSEE3	FSEE2	FSEE1	FSEE0	—	TFEE	USMFSEE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	USMFSEE	Under Switch Minimum Frame Size Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b6	TFEE	TAG Filtering Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	FSEE0	Frame Size Error Enable 0	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b9	FSEE1	Frame Size Error Enable 1	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b10	FSEE2	Frame Size Error Enable 2	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b11	FSEE3	Frame Size Error Enable 3	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b12	FSEE4	Frame Size Error Enable 4	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b13	FSEE5	Frame Size Error Enable 5	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b14	FSEE6	Frame Size Error Enable 6	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b15	FSEE7	Frame Size Error Enable 7	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the EAEID0 register.

USMFSEE Bit (Under Switch Minimum Frame Size Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID0.USMFSED bit will clear this bit.

TFEE Bit (TAG Filtering Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID0.TFED bit will clear this bit.

FSEEi Bit (Frame Size Error Enable i) (i = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID0.FSEDi bit will clear this bit.

36.2.5.3 Error Interrupt Disable Register 0 (EAEID0)

Address(es): ETHA0.EAEID0 874C A508h, ETHA0_NS.EAEID0 074C A508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSED7	FSED6	FSED5	FSED4	FSED3	FSED2	FSED1	FSED0	—	TFED	USMFSED	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	USMFSED	Under Switch Minimum Frame Size Error Disable	0: Has no effects. 1: Sets the EAEIE0.USMFSEE bit to 0.	R/W
b6	TFED	TAG Filtering Error Disable	0: Has no effects. 1: Sets the EAEIE0.TFEE bit to 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	FSED0	Frame Size Error Disable 0	0: Has no effects. 1: Sets the EAEIE0.FSEE0 bit to 0.	R/W
b9	FSED1	Frame Size Error Disable 1	0: Has no effects. 1: Sets the EAEIE0.FSEE1 bit to 0.	R/W
b10	FSED2	Frame Size Error Disable 2	0: Has no effects. 1: Sets the EAEIE0.FSEE2 bit to 0.	R/W
b11	FSED3	Frame Size Error Disable 3	0: Has no effects. 1: Sets the EAEIE0.FSEE3 bit to 0.	R/W
b12	FSED4	Frame Size Error Disable 4	0: Has no effects. 1: Sets the EAEIE0.FSEE4 bit to 0.	R/W
b13	FSED5	Frame Size Error Disable 5	0: Has no effects. 1: Sets the EAEIE0.FSEE5 bit to 0.	R/W
b14	FSED6	Frame Size Error Disable 6	0: Has no effects. 1: Sets the EAEIE0.FSEE6 bit to 0.	R/W
b15	FSED7	Frame Size Error Disable 7	0: Has no effects. 1: Sets the EAEIE0.FSEE7 bit to 0.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.5.4 Error Interrupt Status Register 2 (EAEIS2)

Address(es): ETHA0.EAEIS2 874C A520h, ETHA0_NS.EAEIS2 074C A520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSES7	DQSES6	DQSES5	DQSES4	DQSES3	DQSES2	DQSES1	DQSES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOES7	DQOES6	DQOES5	DQOES4	DQOES3	DQOES2	DQOES1	DQOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOES0	Descriptor Queue 0 Overflow Error Status Flag	0: No Descriptor Queue 0 Overflow Error 1: Descriptor Queue 0 Overflow Error occurred.	R/W
b1	DQOES1	Descriptor Queue 1 Overflow Error Status Flag	0: No Descriptor Queue 1 Overflow Error 1: Descriptor Queue 1 Overflow Error occurred.	R/W
b2	DQOES2	Descriptor Queue 2 Overflow Error Status Flag	0: No Descriptor Queue 2 Overflow Error 1: Descriptor Queue 2 Overflow Error occurred.	R/W
b3	DQOES3	Descriptor Queue 3 Overflow Error Status Flag	0: No Descriptor Queue 3 Overflow Error 1: Descriptor Queue 3 Overflow Error occurred.	R/W
b4	DQOES4	Descriptor Queue 4 Overflow Error Status Flag	0: No Descriptor Queue 4 Overflow Error 1: Descriptor Queue 4 Overflow Error occurred.	R/W
b5	DQOES5	Descriptor Queue 5 Overflow Error Status Flag	0: No Descriptor Queue 5 Overflow Error 1: Descriptor Queue 5 Overflow Error occurred.	R/W
b6	DQOES6	Descriptor Queue 6 Overflow Error Status Flag	0: No Descriptor Queue 6 Overflow Error 1: Descriptor Queue 6 Overflow Error occurred.	R/W
b7	DQOES7	Descriptor Queue 7 Overflow Error Status Flag	0: No Descriptor Queue 7 Overflow Error 1: Descriptor Queue 7 Overflow Error occurred.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSES0	Descriptor Queue 0 Security Error Status Flag	0: No Descriptor Queue 0 Security Error 1: Descriptor Queue 0 Security Error occurred.	R/W
b17	DQSES1	Descriptor Queue 1 Security Error Status Flag	0: No Descriptor Queue 1 Security Error 1: Descriptor Queue 1 Security Error occurred.	R/W
b18	DQSES2	Descriptor Queue 2 Security Error Status Flag	0: No Descriptor Queue 2 Security Error 1: Descriptor Queue 2 Security Error occurred.	R/W
b19	DQSES3	Descriptor Queue 3 Security Error Status Flag	0: No Descriptor Queue 3 Security Error 1: Descriptor Queue 3 Security Error occurred.	R/W
b20	DQSES4	Descriptor Queue 4 Security Error Status Flag	0: No Descriptor Queue 4 Security Error 1: Descriptor Queue 4 Security Error occurred.	R/W
b21	DQSES5	Descriptor Queue 5 Security Error Status Flag	0: No Descriptor Queue 5 Security Error 1: Descriptor Queue 5 Security Error occurred.	R/W
b22	DQSES6	Descriptor Queue 6 Security Error Status Flag	0: No Descriptor Queue 6 Security Error 1: Descriptor Queue 6 Security Error occurred.	R/W
b23	DQSES7	Descriptor Queue 7 Security Error Status Flag	0: No Descriptor Queue 7 Security Error 1: Descriptor Queue 7 Security Error occurred.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

DQOESq Flag (Descriptor Queue q Overflow Error Status Flag) (q = 0 to 7)

[Setting condition]

- HW: This flag is set...

[ETHA is ongoing] (EAMC.OPC[1:0] = 11b and EAMC.OPS[1:0] = 11b)
Descriptor queue is “full (EATDQDC0.DQD[9:0] = EATDQM0.DNQ[9:0])” or “not disabled (EATDQC.TDQDq is not set)” when a descriptor is received for queue q.
[ETHA is going out] (EAMC.OPC[1:0] ≠ 11b and EAMC.OPS[1:0] = 11b)
When a descriptor is received for queue q.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Any descriptor received for a full Descriptor queue will not be accepted by ETHA and will not be forwarded to RMAC.
- SW: Too many descriptors are received for the corresponding queue. EATDQDC0.DQD[9:0] setting should be reviewed.

DQSESq Flag (Descriptor Queue q Security Error Status Flag) (q = 0 to 7)

[Setting condition]

- HW: DQSESq is set when a non-Secure descriptor is received (FDESCR.SEC is not set [MFWD]) and queue q is a secure queue (EATDQSC.TDQSLq is set)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Any non-Secure descriptor received for secure Descriptor queue will not be accepted by ETHA and will not be forwarded to CPU.
- SW: System dependent, cannot be defined here.

36.2.5.5 Error Interrupt Enable Register 2 (EAEIE2)

Address(es): ETHA0.EAEIE2 874C A524h, ETHA0_NS.EAEIE2 074C A524h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSEE7	DQSEE6	DQSEE5	DQSEE4	DQSEE3	DQSEE2	DQSEE1	DQSEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOEE7	DQOEE6	DQOEE5	DQOEE4	DQOEE3	DQOEE2	DQOEE1	DQOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOEE0	Descriptor Queue 0 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b1	DQOEE1	Descriptor Queue 1 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b2	DQOEE2	Descriptor Queue 2 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b3	DQOEE3	Descriptor Queue 3 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b4	DQOEE4	Descriptor Queue 4 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b5	DQOEE5	Descriptor Queue 5 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b6	DQOEE6	Descriptor Queue 6 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b7	DQOEE7	Descriptor Queue 7 Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSEE0	Descriptor Queue 0 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b17	DQSEE1	Descriptor Queue 1 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b18	DQSEE2	Descriptor Queue 2 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b19	DQSEE3	Descriptor Queue 3 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b20	DQSEE4	Descriptor Queue 4 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b21	DQSEE5	Descriptor Queue 5 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b22	DQSEE6	Descriptor Queue 6 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b23	DQSEE7	Descriptor Queue 7 Security Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the EAEID2 register.

DQOEEi Bit (Descriptor Queue i Security Error Enable) (i = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID2.DQOEDi bit will clear this bit.

DQSEEi Bit (Descriptor Queue i Security Error Enable) (i = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID2.DQSEDi bit will clear this bit.

36.2.5.6 Error Interrupt Disable Register 2 (EAEID2)

Address(es): ETHA0.EAEID2 874C A528h, ETHA0_NS.EAEID2 074C A528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSED7	DQSED6	DQSED5	DQSED4	DQSED3	DQSED2	DQSED1	DQSED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOED7	DQOED6	DQOED5	DQOED4	DQOED3	DQOED2	DQOED1	DQOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOED0	Descriptor Queue 0 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE0 bit to 0.	R/W
b1	DQOED1	Descriptor Queue 1 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE1 bit to 0.	R/W
b2	DQOED2	Descriptor Queue 2 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE2 bit to 0.	R/W
b3	DQOED3	Descriptor Queue 3 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE3 bit to 0.	R/W
b4	DQOED4	Descriptor Queue 4 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE4 bit to 0.	R/W
b5	DQOED5	Descriptor Queue 5 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE5 bit to 0.	R/W
b6	DQOED6	Descriptor Queue 6 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE6 bit to 0.	R/W
b7	DQOED7	Descriptor Queue 7 Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQOEE7 bit to 0.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSED0	Descriptor Queue 0 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE0 bit to 0.	R/W
b17	DQSED1	Descriptor Queue 1 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE1 bit to 0.	R/W
b18	DQSED2	Descriptor Queue 2 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE2 bit to 0.	R/W
b19	DQSED3	Descriptor Queue 3 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE3 bit to 0.	R/W
b20	DQSED4	Descriptor Queue 4 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE4 bit to 0.	R/W
b21	DQSED5	Descriptor Queue 5 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE5 bit to 0.	R/W
b22	DQSED6	Descriptor Queue 6 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE6 bit to 0.	R/W
b23	DQSED7	Descriptor Queue 7 Security Error Disable	0: Has no effects. 1: Sets the EAEIE2.DQSEE7 bit to 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.5.7 Error Interrupt Status Register 3 (EAEIS3)

Address(es): ETHA0.EAEIS3 874C A530h, ETHA0_NS.EAEIS3 074C A530h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PROVES	—	—	—	—	—	—	—	EXCES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LATCES	—	—	—	—	—	—	—	CLSES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLSES	Carrier Loss Error Status Flag	0: No Carrier Loss Error 1: Carrier Loss Error occurred	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	LATCES	Late Collision Error Status Flag	0: No Late Collision Error 1: Late Collision Error occurred	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	EXCES	Excessive Collision Error Status Flag	0: No Excessive Collision Error 1: Excessive Collision Error occurred	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PROVES	Pointer Release Overflow Error Status Flag	0: No Pointer Release Overflow Error 1: Pointer Release Overflow Error occurred	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

CLSES Flag (Carrier Loss Error Status Flag)

[Setting condition]

- HW: This flag is set when a carrier loss error occurred.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The frame being sent that caused the error is discarded.
- SW: System dependent, cannot be defined here.

LATCES Flag (Late Collision Error Status Flag)

[Setting condition]

- HW: This flag is set when an late collision error occurred.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The frame being sent that caused the error is discarded.
- SW: System dependent, cannot be defined here.

EXCES Flag (Excessive Collision Error Status Flag)

[Setting condition]

- HW: This flag is set when an excessive collision error occurred.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The frame being sent that caused the error is discarded.
- SW: System dependent, cannot be defined here.

PROVES Flag (Pointer Release Overflow Error Status Flag)

[Setting condition]

- HW: This flag is set when the transmitted frame size is greater than 2048 bytes.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: None
- SW: Reset ESWM by COMA.RRC.RR bit. Don’t transmit the frame whose size is greater than 2048 bytes.

36.2.5.8 Error Interrupt Enable Register 3 (EAEIE3)

Address(es): ETHA0.EAEIE3 874C A534h, ETHA0_NS.EAEIE3 074C A534h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PROVEE	—	—	—	—	—	—	—	EXCEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LATCEE	—	—	—	—	—	—	—	CLSEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLSEE	Carrier Loss Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	LATCEE	Late Collision Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	EXCEE	Excessive Collision Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PROVEE	Pointer Release Overflow Error Enable	0: Interrupt disabled 1: Interrupt Enabled	R/(W) *1
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the EAEID3 register.

CLSEE Bit (Carrier Loss Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID3.CLSED bit will clear this bit.

LATCEE Bit (Late Collision Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID3.LATCED bit will clear this bit.

EXCEE Bit (Excessive Collision Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID3.EXCED bit will clear this bit.

PROVEE Bit (Pointer Release Overflow Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the EAEID3.PROVED bit will clear this bit.

36.2.5.9 Error Interrupt Disable Register 3 (EAEID3)

Address(es): ETHA0.EAEID3 874C A538h, ETHA0_NS.EAEID3 074C A538h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PROVED	—	—	—	—	—	—	—	EXCED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LATCED	—	—	—	—	—	—	—	CLSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLSED	Carrier Loss Error Disable	0: Has no effects. 1: Sets the EAEIE3.CLSEE bit to 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	LATCED	Late Collision Error Disable	0: Has no effects. 1: Sets the EAEIE3.LATCEE bit to 0.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	EXCED	Excessive Collision Error Disable	0: Has no effects. 1: Sets the EAEIE3.EXCEE bit to 0.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PROVED	Pointer Release Overflow Error Disable	0: Has no effects. 1: Sets the EAEIE3.PROVEE bit to 0.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.3 Operation

36.3.1 Data Transmission

Ethernet Agent allows data transmission through the ETHA TX data path, The TX data path is described in Figure 36.2.

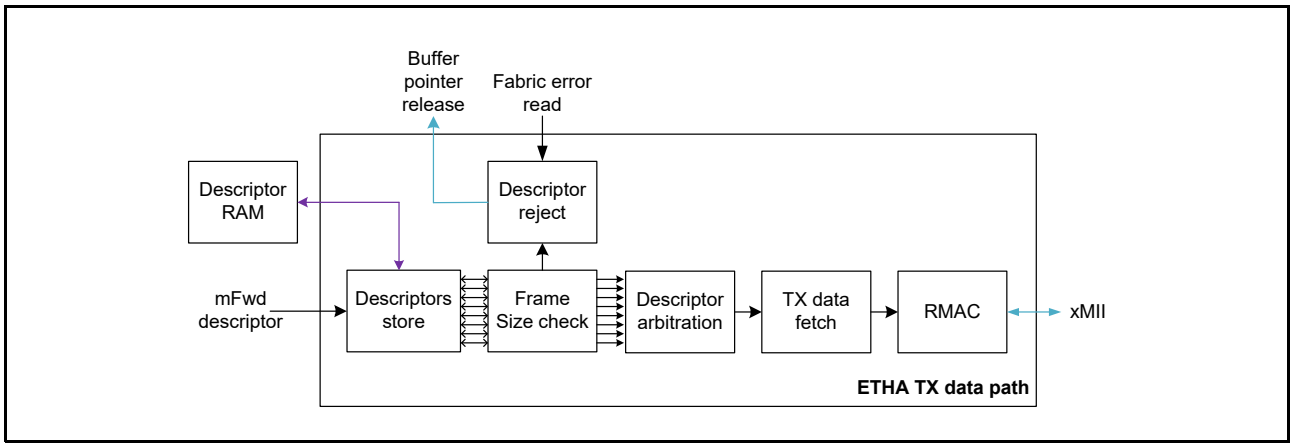


Figure 36.2 ETHA TX Data Path Block Diagram

The TX data path is separated in eleven blocks:

- Descriptor store: This block aims are storing the descriptor received from the Forwarding Engine [MFWD] in the Descriptor RAM.
- Frame size check: This block checks the frame size and decide to forward or discard frames.
- Descriptor reject: This block aims at releasing the pointer of the frames rejected by Frame size control block. This block is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- Descriptor arbitration: Arbitrate between descriptors based on their descriptor queue using strict priority and/or WRR arbitration.
- TX data fetch: This block fetches the frame data from the local RAM, update it depending on the information obtained by L2/L3 update module and on VLAN control information and release the frame pointers. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- RMAC: This block handles the data exchange with the Ethernet PHY. This block functionalities are described in the RMAC specification document [RMAC] so it will not be described.

36.3.1.1 Descriptor Store

Descriptor store aims at storing the descriptors coming from the descriptor bus [MFWD] in the descriptor RAM. This function is the same as the “Descriptor storage” function in the “Descriptor store” block in GWCA. Refer to GWCA specification for more details [GWCA] (The correspondence between ETHA and GWCA registers is described in Table 36.2).

Table 36.2 Descriptor Store ETHA/GWCA Register Correspondence (1/2)

Register name in GWCA	Register name in ETHA
GWIRC.IPVRi[2:0] (i = 0 to 7)	EAIRC.IPVRi[2:0] (i = 0 to 7)
GWRDQSC.RDQSLq (q = 0 to 7)	EATDQSC.TDQSLi (i = 0 to 7)
GWRDQC.RDQDq (q = 0 to 7)	EATDQC.TDQDn (n = 0 to 7)
GWRDQC.RDQPq (q = 0 to 7)	EATDQC.TDQPn (n = 0 to 7)

Table 36.2 Descriptor Store ETHA/GWCA Register Correspondence (2/2)

Register name in GWCA	Register name in ETHA
GWRDQDCq.DQD[9:0] (q = 0 to 7)	EATDQDCq.DQD[9:0] (q = 0 to 7)
GWRDQMq.DNQ[9:0] (q = 0 to 7)	EATDQMq.DNQ[9:0] (q = 0 to 7)
GWRDQMLMq.DMLQ[9:0] (q = 0 to 7)	EATDQMLMq.DMLQ[9:0] (q = 0 to 7)
GWEIS1.DQOESq (q = 0 to 7)	EAEIS2.DQOESq (q = 0 to 7)
GWEIS1.DQSESq (q = 0 to 7)	EAEIS2.DQSESq (q = 0 to 7)

36.3.1.2 Frame Size Check

Frame size check aims at checking if a frame has the right properties to be received by CPU using EATMFSCq (q = 0 to 7) registers.

[Function]

- EATMFSCq (q = 0 to 7) registers are used to set the maximum frame size accepted for each descriptor queue. Any frame received for queue q with a size bigger than EATMFSCq.MFS[15:0] will be discarded and EAEIS0.FSESq flag will be set. The frame size considered is the size of frame on the PHY.

(1) Maximum Interference Size Calculation

The maximum interface *maxInterferenceSize* size can be calculated for a queue q using per equation (1).
Each interference size includes preamble and inter frame gap (IFG).

(1) maxInterferenceSize = queueInterferenceSizeLow + queueInterfereSize + Σi (queueInterferenceSizeHigh[i])

[Where]

- *queueInterferenceSizeLow* is the interference created by all the queues which have a priority smaller than q. This interference size is equal to (*queueLowMaxSize*+20)*8 where *queueLowMaxSize* is EATMFSCi.MFS[15:0] maximum values for all i smaller than q.
- *queueInterfereSize* is the interference created by queue q and is equal to (EATMFSCq.MFS[15:0]+20)*8
- *queueInterferenceSizeHigh*[i] is the interference created by queue i with a higher priority than q. *queueInterferenceSizeHigh*[i] is equal to (*MaxFrameBurst**EATMFSCi.MFS[15:0] + *queueLowMaxSize* + 40)*8 where *queueLowMaxSize* is EATMFSCj.MFS[15:0] maximum values for all j smaller than i and where *MaxFrameBurst* is the maximum number of frames that can be expected in a burst. *MaxFrameBurst* is a variable which is system dependent and cannot be defined here.

36.3.1.3 TX Data Fetch

TX data fetch is the same as “RX data fetch” in the GWCA except that it can handle pre-emption and that it also handles the data fetching for cut-through queues. Refer to GWCA specification for more details [GWCA] about data update in case of none-cut-through frames (The correspondence between ETHA and GWCA registers is described in Table 36.2).

Table 36.3 TX Data Fetch ETHA/GWCA Register Correspondence

Register name in GWCA	Register name in ETHA
GWVCC.VEM	EAVCC.VEM
{GWMAC0.MAU[15:0], GWMAC1.MAL[31:0]}	{MRMAC0.MAU[15:0], MRMAC1.MAL[31:0]} [RMAC]
GWRGC.RCPT	Fixed to 1 (RMAC will add an FCS to all frames sent without it)

36.3.1.4 RMAC TX Interface Control

After TX data fetch, data is sent to the RMAC through ETHA TX interface [RMAC]. Table 36.4 shows how ETHA TX interface status information are provided to the RMAC.

Table 36.4 ETHA TX Interface Status Provided by ETHA to RMAC

Signal name in RMAC	Bus width	Value provided by ETHA
FCS in	1	Set to 1 if corresponding transmit frames have an FCS (descriptor is received from forwarding engine with FDESCR.FI equal to 1) and if the frame has not been modified by the switch (No update due to VLAN tagging/un-tagging and no update requested by L3 routing [MFWD]).
Timestamp capture	1	Set to LDESCR.TXC
Insert egress timestamp	1	Set to LDESCR.IET
Calculate residence time	1	Set to LDESCR.CRT
Timer number	1	Set to LDESCR.TN
Timestamp unique number	8+PTP_TUNES = 10	Set to {SPN,LDESCR.TSUN} where SPN is the source port number of the corresponding frame.

36.3.2 Data Reception

Ethernet Agent allows data reception through the ETHA RX data path, The RX data path is described in Figure 36.3.

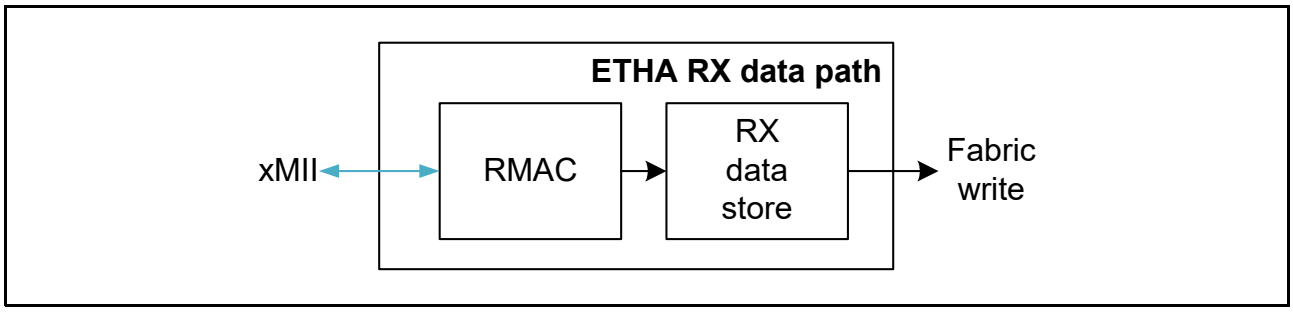


Figure 36.3 ETHA RX Data Path Block Diagram

The RX data path is separated in two blocks:

- RMAC: This block handles the data exchange with the Ethernet PHY. This block functionalities are described in the RMAC specification document [RMAC] so it will not described.
- RX data store: This block extracts TAG information for frames, applies VLAN tagging and save the frames in the local RAM. This block can also release pointers while going out of OPERATION mode. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.

[Restriction]

- SW: The switch maximum input frame size is 60 Kbytes. RMAC should be set so it does not send frames bigger than 60 Kbytes (MRFSCE.EMXS and MRFSCP.PMXS[15:0] should be set to a value smaller or equal to 61440 [RMAC]).

36.3.2.1 RX Data Store

RX data store has follow functions, it extracts TAG information for frames, applies VLAN tagging, and save the frames in the local RAM. This function is the same as the “TX data store” function in GWCA except for the local descriptor format. Refer to GWCA specification for more details [GWCA] (The correspondence between ETHA and GWCA registers is described in Figure 36.5).

Table 36.5 Data Store ETHA/GWCA Register Correspondence

Register name in GWCA	Register name in ETHA
GWTTFC	EARTFC
GWVCC.VIM	EAVCC.VIM
GWVTC	EAVTC
GWEIS0.USMFSES	EAEIS0.USMFSES

As local descriptor format, ETHA can only use ethernet local descriptors. Ethernet local descriptor is explained in GWCA specification and the field corresponding values for ethernet agent is described in Table 36.6.

Table 36.6 Ethernet Local Descriptor Field Description (1/2)

Field name	Bit width	Value for ETHA
FI	1	Set to RMAC ETHA Rx Interface “FCS included” field [RMAC]
FMT	1	Set to 0 for ethernet descriptors
TXC	1	Set to RMAC ETHA Rx Interface “Timestamp capture at TX side” field [RMAC]

Table 36.6 Ethernet Local Descriptor Field Description (2/2)

Field name	Bit width	Value for ETHA
IET	1	Set to 0
CRT	1	Set to 0
TN	1	Set to RMAC ETHA Rx Interface “Time domain number” field [RMAC]
TSUN	8	Set to RMAC ETHA Rx Interface “Timestamp unique number” field [RMAC]
SAEF	8	Refer to Figure 36.4 and Table 36.7
VCTRL	3	VCTRL[2] values: • Set to EAVCC.VIM. VCTRL[1:0] values 00b: For No TAG, R-TAG and Unknown TAG frames. 01b: For C-TAG and CR-TAG frames. 10b: For SC-TAG and SCR-TAG frames. 11b: For CoS-TAG and CoSR-TAG frames.
RTGI	1	Values: 0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames. 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames.
TPL	16	Payload length of frame. It only includes the data saved in the data RAM [MFWD].
TSV	1	Values: 0: Timestamp has not been received along with the frame from RMAC 1: Timestamp has been received along with the frame from RMAC
TSD	1	Set to RMAC ETHA Rx Interface “Timestamp is a default timestamp” field [RMAC]
TSNS	30	Set to 00000000h if timestamp has not been received. Set to timestamps nanosecond part if timestamp has not been received
TSS	32	Set to 00000000h if timestamp has not been received. Set to timestamps second part if timestamp has not been received
RSV	—	Set to 0

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	EC1[1:0]		EC0[3:0]				RSV	TFE

Figure 36.4 Source Agent Error Flag Format

Table 36.7 Source Agent Error Flag Field Description

Field name	Bit width	Description
TFE	1	TAG Filtering Error [Setting conditions] HW: When a frame does not fit the required TAG ingress format. Refer to EARTFC register explanation.
EC0	4	Error Code 0 [RMAC] Values: 0 0 0 0: No error 0 0 0 1: Frame received from RMAC with “PHY error” bit set 0 0 1 1: Frame received from RMAC with “Nibble error” bit set 0 1 0 0: Frame received from RMAC with “FCS (mCRC) error” bit set 0 1 0 1: Frame received from RMAC with “Final fragment missing error” bit set 0 1 1 0: Frame received from RMAC with “C Fragment SMD Error” bit set 0 1 1 1: Frame received from RMAC with “C Fragment FRAG_COUNT Error” bit set 1 0 0 0: Frame received from RMAC with “RMAC Frame filtered” bit set 1 0 0 1: Frame received from RMAC with “Reception partially out of operation” bit set Others: Reserved Restrictions: • HW: Because of the coding mechanism only one error can be flagged in this field. • HW: If several errors happen for the same frame, the error with the smallest EC0 value will be flagged.
EC1	2	Error Code 1 [RMAC] Values: 0 0: No error 0 1: Frame received from RMAC with “Buffer overflow error” bit set 1 0: Frame received from RMAC with “Undersize error” bit set 1 1: Frame received from RMAC with “Oversize error” bit set Restrictions: • HW: Because of the coding mechanism only one error can be flagged in this field. • HW: If several errors happen for the same frame, the error with the smallest EC0 value will be flagged.
RSV	—	Reserved Set to 0

36.3.3 Operating Modes

Table 36.8 describes ETHA operation modes.

Table 36.8 ETHA Operation Modes

Operation mode	GWMS.OPS[1:0] value	Description
DISABLE	0 1	• No transaction is ongoing. • Only status registers are accessible for writing when the agent clock is enabled.
RESET	0 0	• No transaction is ongoing. • An internal Reset is asserted to reset ETHA logic with status registers (When a register is reset in RESET mode, it is mentioned in its description). • No register is accessible for writing. • RAM values are held.
CONFIG	1 0	• No transaction is ongoing. • Static and some dynamic registers are accessible for writing.
OPERATION	1 1	• Transactions are ongoing. • Dynamic registers are accessible for writing.

36.3.3.1 Operating Mode Transitions

Figure 36.5 shows the operating mode transitions.
A mode transition can be trigger by:

- Hardware reset.
- Software reset.
- Configuration of EAMC.OPC[1:0]. In that case, mode transition will be confirmed by EAMS.OPS[1:0].

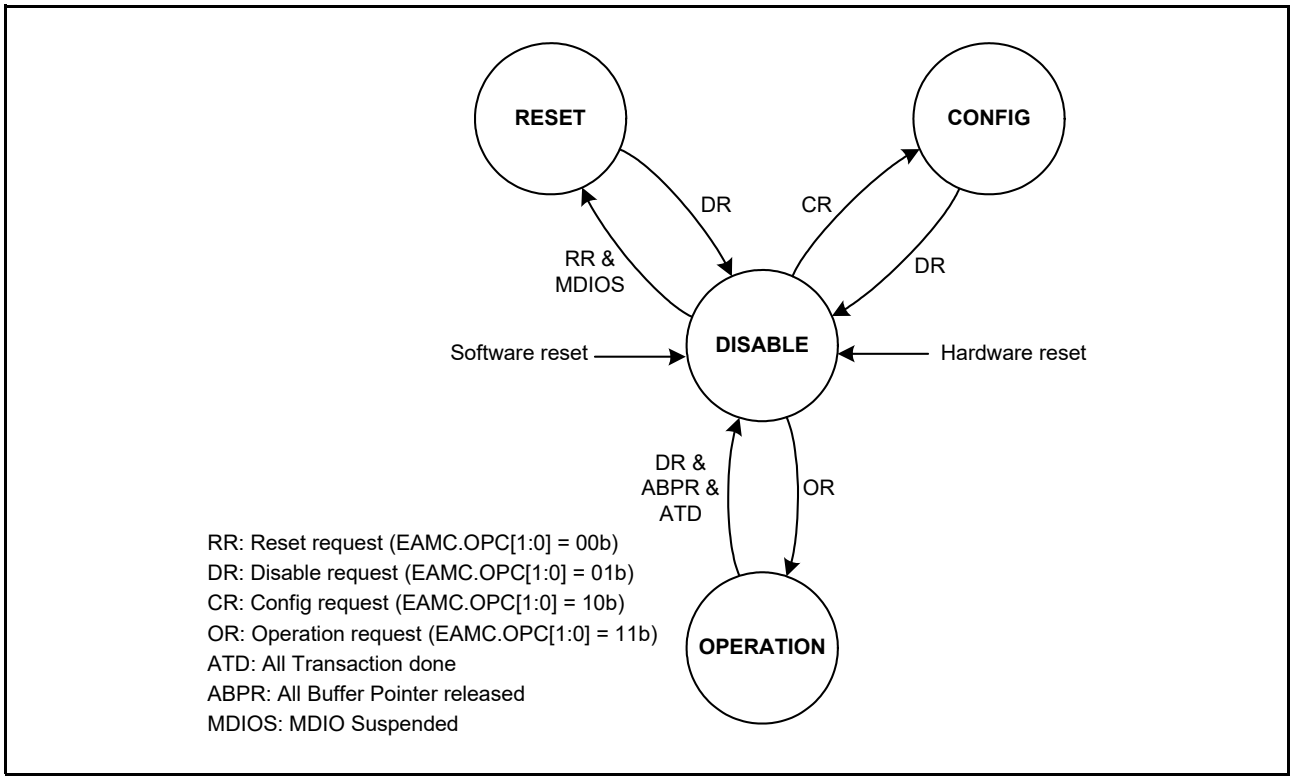


Figure 36.5 Operating Mode Transitions

Complementary information on transition conditions
ABPR:

- All the buffer pointers contained in the ETHA are release to the forwarding engine [MFWD].

ATD:

- All the descriptors already received are processed and corresponding frames are transmitted to the Ethernet PHY.
- All ongoing reception frames are fully sent to the forwarding engine [MFWD].

MDIOS:

- MDIO has no ongoing access.

[Restrictions]

- Software shall only trigger transitions shown to Figure 36.5.
  Exp: Do not directly transition from OPERATION to CONFIG.

36.3.4 Software Flows

[Restriction]

- SW: Please follow to the flow in this section.

36.3.4.1 Mode Transition Flow

The mode transition flow is described in Figure 36.6.

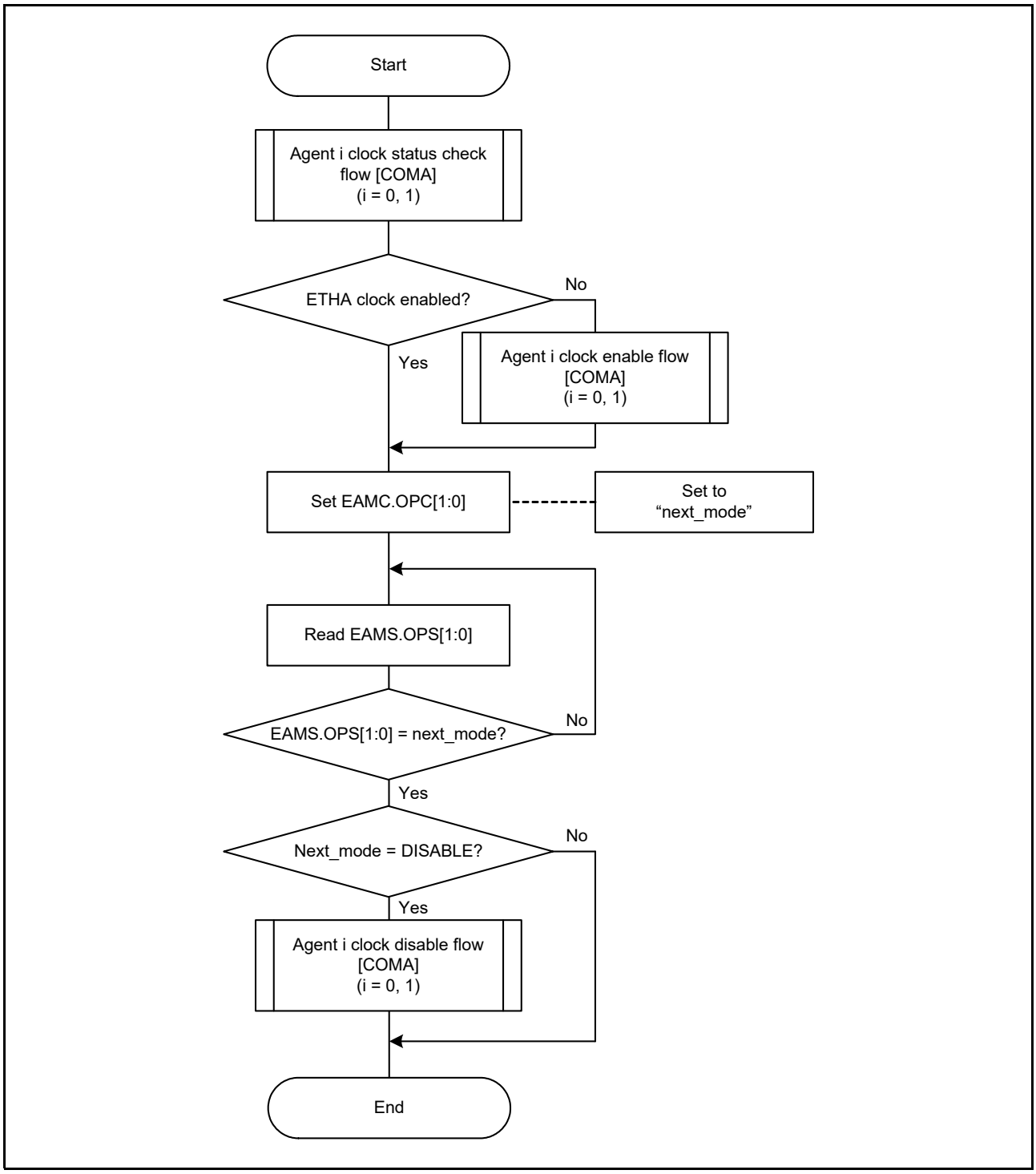


Figure 36.6 Mode Transition Flow

Note: If SW already know that the clock is enabled, clock status check step can be skipped.
Note: If SW does not need to disable the agent clock in DISABLE mode (because it will go directly to another mode or because the clock never needs to be disabled), clock disable step can be skipped.

36.3.4.2 Reset Flow

The reset flow is described in Figure 36.7.

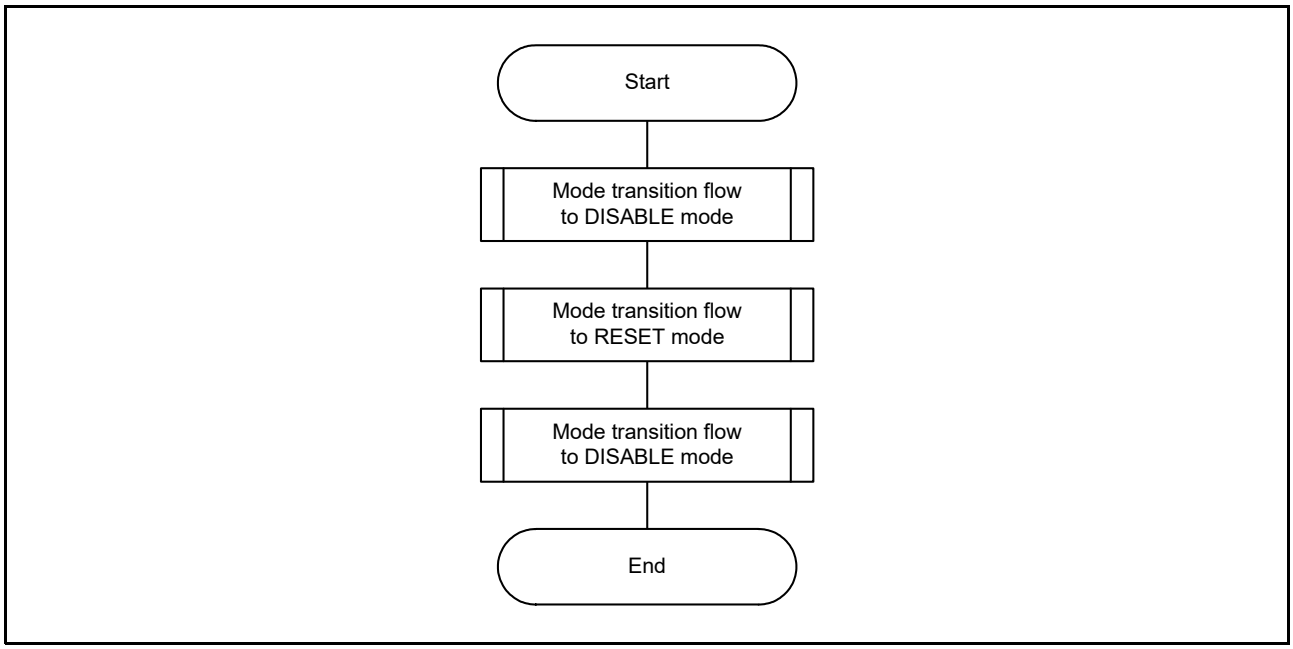


Figure 36.7 Reset Flow

36.3.4.3 Initialization Flow

The initialization flow is described in Figure 36.8.

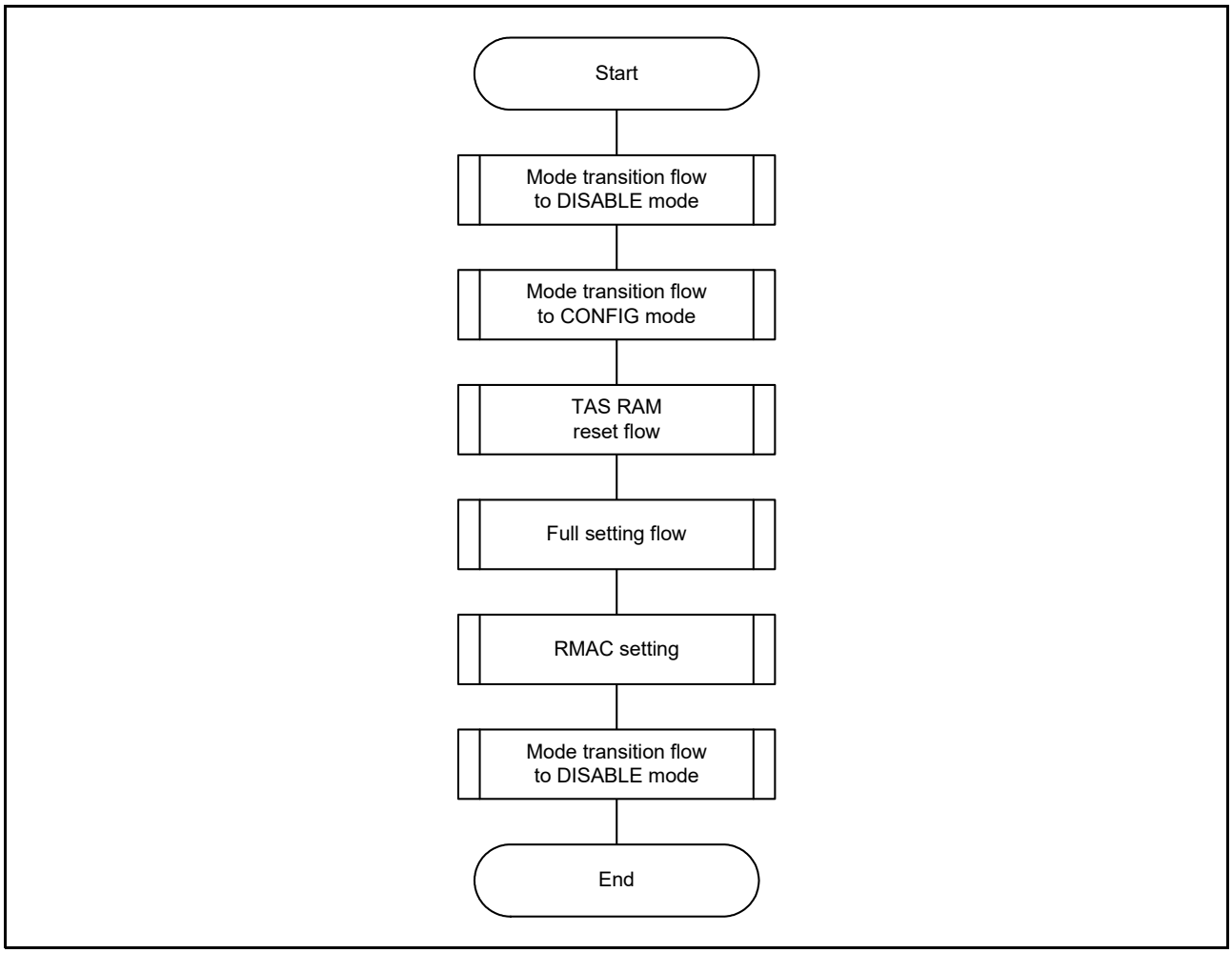


Figure 36.8 Initialization Flow

Note: For RMAC setting refer to section 37, Ethernet MAC (RMAC)

36.3.4.4 Reinitialization Flow

The reinitialization flow is described in Figure 36.9.

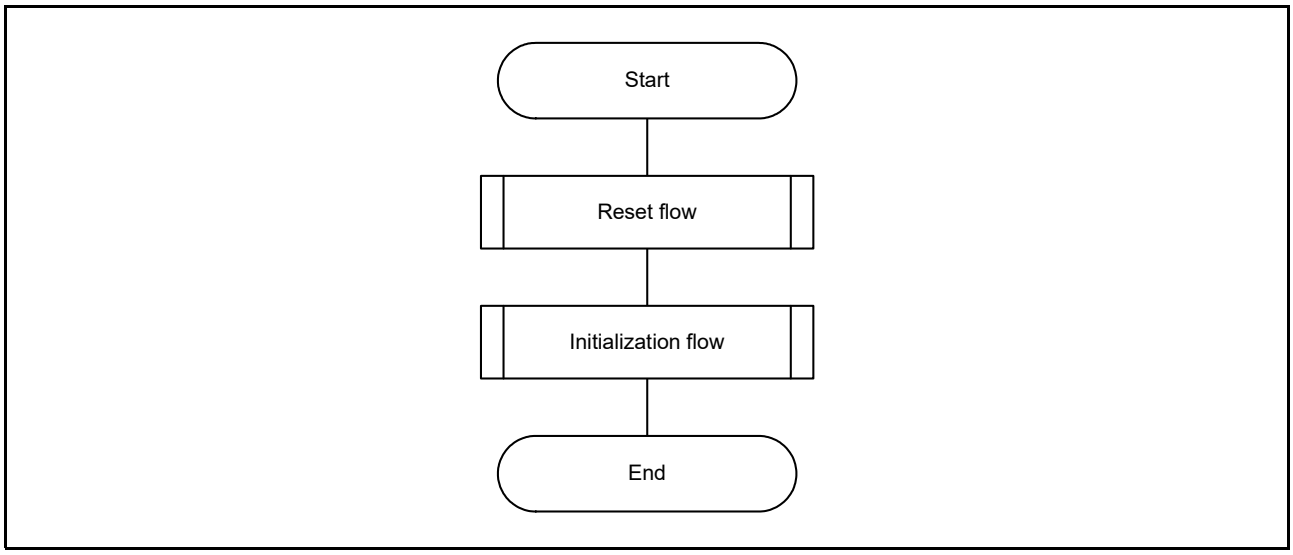


Figure 36.9 Reinitialization Flow

36.3.4.5 Interrupt Handling Flow

The interrupt handling flow is described in Figure 36.10.
[Restriction]

- This flow is not usable in RESET mode.

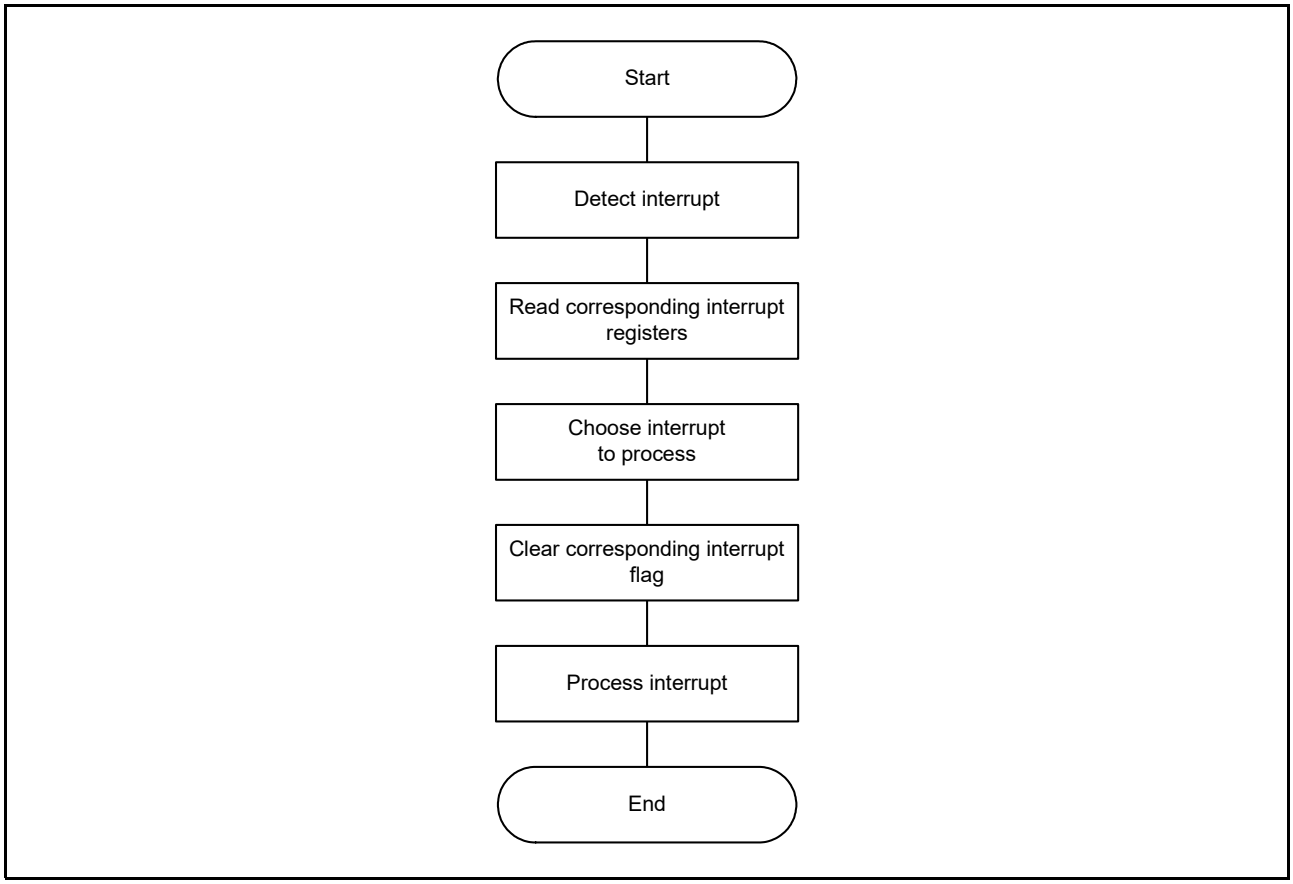


Figure 36.10 Interrupt Handling Flow

36.3.4.6 Called Software Flows

The flows described in this section can only be called from other flows thanks to a “flow link” box (Figure 36.xx) and cannot be used alone.

(1) Full Setting Flow

The full setting flow is described in Figure 36.11.

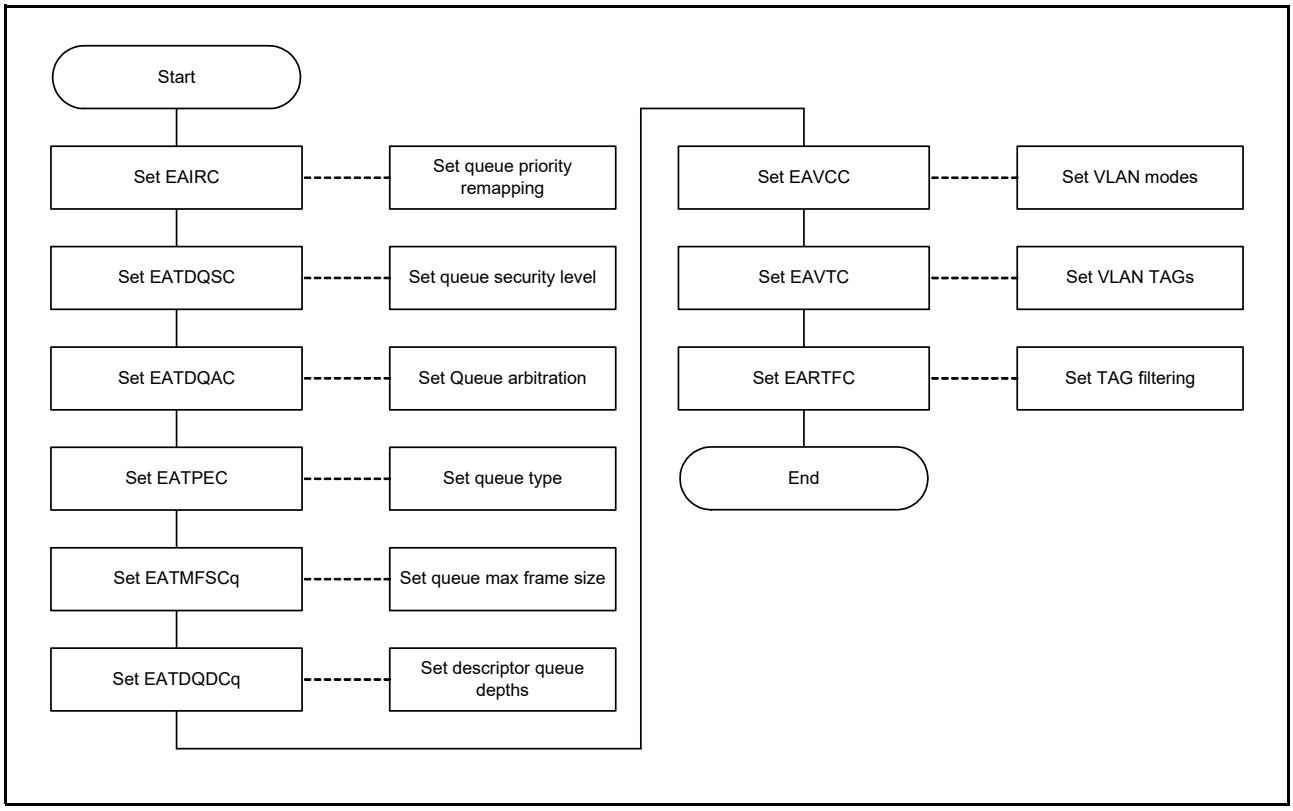


Figure 36.11 Full Setting Flow

37. Ethernet MAC (RMAC)

37.1 Overview

This IP is an Ethernet controller that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard.
When connected with a physical-layer LSI chip (PHY-LSI) that complies with the standard, it is able to transmit and receive Ethernet (IEEE 802.3) frames. It has a single MAC layer interface.

37.1.1 Features

Table 37.1 lists the specifications.

Table 37.1 Specifications (Functions)

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
	Priority-based Flow Control with 802.1Qbb
	Support for IEEE 1588 time synchronization, Support for IEEE AS and IEEE AS-Rev acceleration
Transfer speed	Supports transfer at 10,100,1000 Mbps
Mode	Full-duplex mode
	Energy Efficient Ethernet (Low Power Idle)
Interface	Supports the IEEE 802.3 standard MII (Media Independent Interface) and GMII (Gigabit Media Independent Interface)
	PHY MDIO programming Interface
	Supports internal register interface (32bit data width) Supports internal data interface (64bit data width)
Original function	Configurable frame data padding A variety of flexible receive address filtering modes Configurable CRC pass through for transmit and receive Network statistic counter (IEEE 802.3, ROMN RFC2819)
Magic Packet™	Detection of Magic Packets™*1 and output of a detected signal

Note 1. Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

Table 37.2 References

Title	Contents
IEEE Std 802.3™-2008	Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications
IEEE 802.3az™-2010	Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet
IEEE Std 1588™-2008	Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
IEEE802.1Qbb	Priority-based Flow Control

37.1.2 RMAC Block Diagram

Figure 37.1 represents RMAC block diagram.

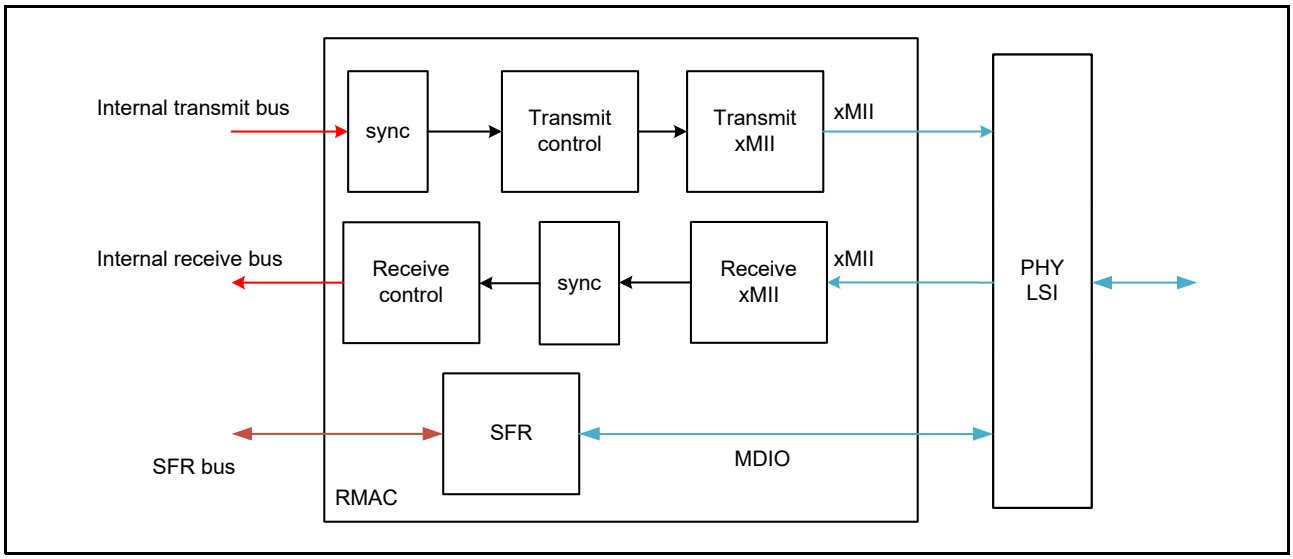


Figure 37.1 Block Diagram

Block	Function
SFR	Contains all the RMAC SFRs and the bridging system to access PHY registers. RMAC registers are on clk.
Receive control	Consists of the receive FSM and Data path logic, which disassembles the received Ethernet frame. It controls functions like checking and removing FCS, Destination Address filtering, Flow Control etc.
Transmit control	Consists of the transmission FSM and Data path logic, which facilitates the transmitted Ethernet frame. It controls functions like adding the Preamble and SFD, adding PAD bytes, calculating the FCS etc.
Receive xMII/ Transmit xMII	Provides the connectivity to the PHY. It supports several types of xMII (MII, GMII, XGMII).
RMAC	IP described in this documentation.
Sync	Synchronized between bus clock domain and mii clock domain.

37.2 Register Description

37.2.1 Function Registers

37.2.1.1 PHY Station Management Register (MPSM)

Address(es): RMAC0.MPSM 874C B000h, RMAC0_NS.MPSM 074C B000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PRD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	POP[1:0]		PRA[4:0]					PDA[4:0]					MFF	—	PSME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSME	PHY Station Management Enable	Set this bit to access PHY registers. 0: PHY register access is not on progress. 1: PHY register access is requested and on progress.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	MFF	Management Frame Format	0: Normal management frame format defined in Clause 22 of IEEE802.3 describe “MDIO” in this document. 1: Extension management frame format defined in Clause 45 of IEEE802.3 describe “eMDIO” in this document.	R/W
b7 to b3	PDA[4:0]	PHY Device Address	MPSM.MFF = 0: The PHY Address field MPSM.MFF = 1: Port address field	R/W
b12 to b8	PRA[4:0]	PHY Register Address	MPSM.MFF = 0: Register address MPSM.MFF = 1: Device address	R/W
b14, b13	POP[1:0]	PHY Operation Code	These bits specify following value according to MPSM.MFF. • MPSM.MFF = 0 10 Read Frame 01 Write Frame Others reserved • MPSM.MFF = 1 00 Address Frame 01 Write Frame 11 Read Frame 10 Post-read-increment-address frame	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31 to b16	PRD[15:0]	PHY Register Data	MPSM.MFF = 0: Data field of the PHY register MPSM.MFF = 1: Data/Address field of the PHY register	R/W

Provides an access to the PHY-LSI internal registers via the MDIO.

PSME Bit (PHY Station Management Enable)

[Setting condition]

- SW: By writing 1 to this bit. It starts phy access.

[Clearing condition]

- HW: When PHY register access is completed, this bit returns automatically to 0.

[Restrictions]

- SW: Accessing to MPSM register while this bit is asserted has no effect.
- SW: This bit cannot be set when MPIC.PSMCS[6:0] = 0.
- SW: Before resetting RMAC, this bit should always be checked. If this bit is set, no reset should be performed.

MFF Bit (Management Frame Format)

This bit selects the format of management frame.

PDA[4:0] Bits (PHY Device Address)

These bits specify following value according to MPSM.MFF.
Refer to section 37.3.4, Energy Efficient Ethernet for more details.

PRA[4:0] Bits (PHY Register Address)

These bits specify following value according to MPSM.MFF.
Refer to section 37.3.4, Energy Efficient Ethernet for more details.

POP[1:0] Bits (PHY Operation Code)

These bits specify the operation code.
Refer to section 37.3.4, Energy Efficient Ethernet for more details.
[Restriction]

- SW: Value 00 or 11 should not be set when MPSM.MFF = 0.

PRD[15:0] Bits (PHY Register Data)

These bits specify following value according to MPSM.MFF.
Refer to section 37.3.4, Energy Efficient Ethernet for more details.
[Functions]

- Write operation: HW send these bits value to PHY in write operation.
- Read operation: HW write read data to these bits from PHY in read operation.

37.2.1.2 PHY Interfaces Configuration Register (MPIC)

Address(es): RMAC0.MPIC 874C B004h, RMAC0_NS.MPIC 074C B004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	PSMCT[2:0]			—	PSMHT[2:0]			PSMDP	PSMCS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PLSPP	PIPP	PIP	FHDM	—	LSC[2:0]			PIS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PIS[2:0]	PHY Interface Select	b2 b0 0 0 0: MII 0 1 0: GMII Settings other than above are prohibited.	R/W
b5 to b3	LSC[2:0]	Link Speed Configuration	b5 b3 0 0 0: 10 Mbps 0 0 1: 100 Mbps 0 1 0: 1 Gbps Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R
b7	FHDM	Full/Half-Duplex Mode Select	0: Full-Duplex mode (default) 1: Half-Duplex mode	R/W
b8	PIP	PHY Interrupt Polarity	0: PHY interrupt is active low (default). 1: PHY interrupt is active high.	R/W
b9	PIPP	PHY Interrupt Pin Plugged	0: Unplugged 1: Plugged	R/W
b10	PLSPP	PHY Link Status Pin Plugged	0: Unplugged 1: Plugged	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b16	PSMCS[6:0]	PHY Station Management Clock Selection	b22 b16 0 0 0 0 0 0 0: MDC clock is off 0 0 0 0 0 0 1: clk/4 0 0 0 0 0 1 0: clk/6 : : 1 1 1 1 1 1 0: clk/254 1 1 1 1 1 1 1: clk/256	R/W
b23	PSMDP	PHY Station Management Disable Preamble	0: PHY Station Management Preamble enabled 1: PHY Station Management Preamble disabled	R/W
b26 to b24	PSMHT[2:0]	PHY Station Management Hold Time Adjustment	b26 b24 0 0 0: No adjusted hold time (i.e. change MDO at the rising edge of MDC) 0 0 1: 1 clk cycle extra hold time : : 1 1 1: 7 clk cycle extra hold time	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b30 to b28	PSMCT[2:0]	PHY Station Management Capture Time Adjustment	b30 b28 0 0 0: No adjusted capture time (i.e. capture MDI at the rising edge of MDC) 0 0 1: Capture before 1 clk cycle : : 1 1 1: Capture before 7 clk cycle	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

This register configures MDIO and xMII interface.

PIS[2:0] Bits (PHY Interface Select)

These bits select the xMII interface connected to the PHY.

[Notes]

- The value for LSC and PIS are only available shown in Table 33.5, Interface Modes and Register Settings.

LSC[2:0] Bits (Link Speed Configuration)

[Note]

- The value for LSC and PIS are only available shown in Table 33.5, Interface Modes and Register Settings.

FHDM Bit (Full/Half-Duplex Mode Select)

[Note]

- Half-duplex mode supports 10/100Mbps@MII/RMII.
- In half-duplex mode, should not use all TSN features (TAS, CBS, Qci, 802.1as (Time stamp), Frame Preemption), LPI mode, Pause/PFC frame, Loop back mode, Cut-through mode.

PIP Bit (PHY Interrupt Polarity)

This bit sets the active level of the PHY interrupt pin (rmc_phy_int).

[Note]

- For the active level, refer to the specifications of PHY-LSI to be connected.

PSMCS[6:0] Bits (PHY Station Management Clock Selection)

These bits configure the cycle of MDC clock. MDC clock is divided from clk clock. The duty cycle of MDC clock is 50%. Refer to section 37.3.3, Station Management for details.

[Restriction]

- SW: These bits cannot be set if MPSM.PSME is set.

PSMDP Bit (PHY Station Management Disable Preamble)

This bit could cut the preamble pattern (sequence of 32 contiguous one bits). When it is asserted, the preamble pattern is skipped and the management frame starts with the ST (Start of Frame) pattern.

[Restriction]

- SW: This bit cannot be set if MPSM.PSME is set.

PSMHT[2:0] Bits (PHY Station Management Hold Time Adjustment)

The configured value will shift the data change timing of MDO with respect to the rising edge of MDC clock.

[Restrictions]

- SW: These bits cannot be set if MPSM.PSME is set.
- SW: The configured value must be less than MDC clock cycle (less than MPIC.PSMCS[6:0]*2).

PSMCT[2:0] Bits (PHY Station Management Capture Time Adjustment)

The configured value will shift the data capture timing of MDI with respect to the rising edge of MDC clock.

[Restrictions]

- SW: These bits cannot be set if MPSM.PSME is set.
- SW: The configured value must be less than MDC clock cycle (less than MPIC.PSMCS[6:0]*2).

37.2.1.3 PHY Interfaces Monitoring Register (MPIM)

Address(es): RMAC0.MPIM 874C B008h, RMAC0_NS.MPIM 074C B008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPIA	PLS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PLS	PHY Link Status Flag	0: PHY link signal is at the low level. 1: PHY link signal is at the high level.	R
b1	LPIA	LPI Active	0: No LPI 1: LPI active	R
b31 to b2	—	Reserved	These bits are read as 0.	R

PLS Flag (PHY Link Status Flag)

This flag indicates PHY link state. It could be used only when the link status pin (rmc_phy_link) is connected to the link signal from PHY-LSI.

For the active level, refer to the specifications of PHY-LSI to be connected.

LPIA Bit (LPI Active)

This bit indicates that LPI is active on receive side.

[Setting condition]

- HW: When LPI code is appeared on detected on xMII receive data signals (rmc_rx_data_valid_in, rmc_rx_data_data_in[31:0], rmc_rx_data_err_in). Refer xMII specification for more details.

[Clearing condition]

- HW: When LPI code is not appeared on detected on xMII receive data signals.

37.2.1.4 I/O Configuration Register (MIOC)

Address(es): RMAC0.MIOC 874C B010h, RMAC0_NS.MIOC 074C B010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIOC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MIOC0	I/O Configuration 0 (ETn_TX_CLK/ETn_RX_CLK mode)	clk_phy_tx / clk_phy_rx mode for mii phy. 0: Normal mode. clk_phy_tx and clk_phy_rx are provided. 1: Emergency clock recovery mode. clk_phy_tx and clk_phy_rx are provided free-run clock(clk_common). (So transmission is not possible in this state.)	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

MIOC0 Bit (I/O Configuration 0 (ETn_TX_CLK/ETn_RX_CLK mode))

[Restriction]

- SW can write 1 to MIOC0 bit when clock derivation from PHY is unexpectedly stopped. In this case, write 1 to this bit to flush all internal condition in RMAC. After waiting to flush and change DISABLE mode, write 0 to this bit and change RESET mode.

37.2.1.5 Transmission Frame Format Configuration Register (MTFFC)

Address(es): RMAC0.MTFFC 874C B020h, RMAC0_NS.MTFFC 074C B020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCM	DPAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPAD	Data Padding Disable	0: Padding is inserted. 1: Padding is not inserted.	R/W
b1	FCM	Flow Control Mode	0: PAUSE 1: PFC	R/W
b30 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to configure the frame format to be transmitted.

DPAD Bit (Data Padding Disable)

Disables the automatic padding of too small transmit frames (less than 60 bytes) with no FCS.

[Note]

- If a too small transmit frame (less than 60 bytes) with no FCS is received when this bit is set, RMAC adds FCS and sends such a frame.
- When half-duplex mode, set 0 (padding is inserted) to keep slot time to detect collision.

[Restrictions]

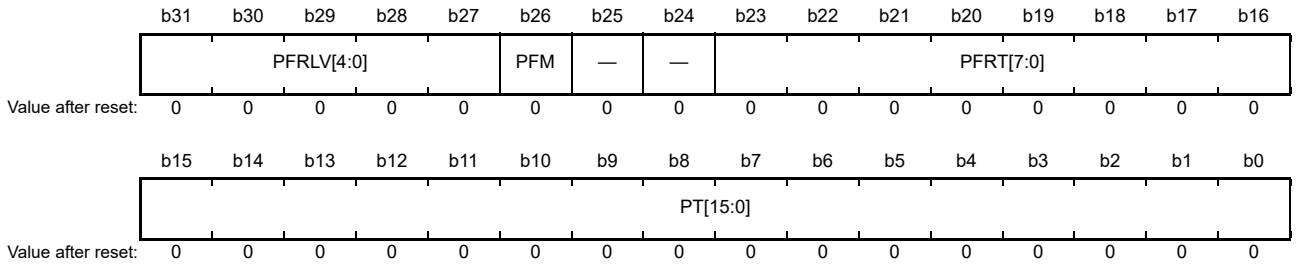
- HW: A Verify frame, a Pause frame and a PFC frame are not affected by this bit.
- HW: Padding is not executed when “fcs included (FI = 1)” indication is applied by upper layer.

FCM Bit (Flow Control Mode)

This bit specifies the flow control mode.

37.2.1.6 Transmission Pause or PFC Frame Configuration Register (MTPFC)

Address(es): RMAC0.MTPFC 874C B024h, RMAC0_NS.MTPFC 074C B024h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PT[15:0]	Pause Time*1, *2	b15 b0 0000h: Pause frame disable 0001h: 512 × 1 bit-time : : FFFFh: 512 × 65,535 bit-time	R/W
b23 to b16	PFRT[7:0]	Pause Frame Retransmission Time *1, *2	b23 b16 00h: 512 × 0 bit-time 01h: 512 × 1 bit-time : : FFh: 512 × 255 bit-time	R/W
b25, b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26	PFM	Pause Frame Mode*2	0: Automatic pause frame. Sending a pause frame will be triggered by a hardware pause request. 1: Manual pause frame. Sending a pause frame will be triggered by MTPFC2.MPFR, MTPFC2.MPFCFR0, or MTPFC2.MPFCFR1.	R/W
b31 to b27	PFRLV[4:0]	Pause or PFC Frame Retry Limit Value*2	b31 b27 00h: No interrupt 01h: Interrupt at 1 pause frame : : 1Fh: Interrupt at 31 pause frames	R/W

Note 1. The unit for the setting is 512 bit-time. 1 bit-time is equal to 1 ns in 1000 Mbps, 10 ns in 100 Mbps, 100 ns in 10 Mbps.
Note 2. These bits are common bit for PAUSE and PFC.

The register is used to configure the TIME parameter value for pause frames.

PT[15:0] Bits (Pause Time)

These bits set the TIME parameter value for pause frames.
The value of these bits is set to Time field of a PAUSE or PFC frame. If the value of these bits is 0, PAUSE of PFC is not sent.
[Restriction]

- SW: These bits are only writable in CONFIG mode.

PFRT[7:0] Bits (Pause Frame Retransmission Time)

These bits set the TIME parameter value of pause frame retransmission.
A pause frame will be retransmitted when the previous frame has been sent more than (PT[15:0] - PFRT[7:0]) ± 10 cycles bit-time ago.
[Restrictions]

- SW: These bits should be set to a smaller value than one of MTPFC.PT[15:0].
- SW: These bits are only writable in CONFIG mode.

PFM Bit (Pause Frame Mode)

[Restriction]

- SW: These bits are only writable in CONFIG mode.

PFRLV[4:0] Bits (Pause or PFC Frame Retry Limit Value)

Value of pause frame sent without pause frame request signal de-assertion from which at which RMAC will emit an interrupt.

[Restriction]

- SW: These bits are only writable in CONFIG mode.

[Note]

- The number of pause (pfc) frames in this field indicates the total number of transmission attempt.

37.2.1.7 Transmission Pause or PFC Frame Configuration Register 2 (MTPFC2)

Address(es): RMAC0.MTPFC2 874C B028h, RMAC0_NS.MTPFC2 074C B028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPFR	PFTTZ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MPFCFR1	MPFCFR0	—	—	—	—	—	—	PFCTTZ1	PFCTTZ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFCTTZ0	PFC Frame 0 Transmission with TIME = 0	0: The transmission of PFC frames with TIME value zero is disabled. 1: The transmission of PFC frames with TIME value zero is enabled	R/W
b1	PFCTTZ1	PFC Frame 1 Transmission with TIME = 0		
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MPFCFR0	Manual PFC Frame 0 Request	Writing 1 to this bit request manual pfc frame.	R/W
b9	MPFCFR1	Manual PFC Frame 1 Request		R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PFTTZ	Pause Frame Transmission with TIME = 0	0: The transmission of Pause frames with TIME value zero is disabled. 1: The transmission of Pause frames with TIME value zero is enabled	R/W
b17	MPFR	Manual Pause Frame Request	Writing 1 to this bit requests a manual pause frame.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to set the TIME parameter value for PFC frames.

PFCTTZn Bit (PFC Frame n Transmission with TIME = 0) (n = 0, 1)

This bit enables the transmission of automatic PFC frames with TIME value zero. It is transmitted when the ETHA negated the PFC request in automatic PFC frame mode or when the software negate the MTPFC2.MPFCFRn bit in manual PFC frame mode and the remaining PFC time is more than PFRT.

[Restrictions]

- SW: This bit has no effect for Manual PFC frame transmission.
- SW: This bit is only writable in CONFIG mode

MPFCFRn Bit (Manual PFC Frame n Request) (n = 0, 1)

If a pfc frame is requested so fast that the hardware did not have time to send the previous one (because a long frame is ongoing or because the software requests too many pfc frames), the hardware will ignore the pfc frame requests.
After completion of non-pause frame transmission the next attempt for PFC frame will be only when timer expires if request for PFC frame is still high.

[Setting condition]

- SW: Writing 1 to this bit will set.

[Clearing conditions]

- SW: Writing 0 to this bit will clear.
- HW: Going out of OPERATION mode will clear this bit.

[Restriction]

- SW: This bit is only writable in OPERATION mode.

PFTTZ Bit (Pause Frame Transmission with TIME = 0)

This bit enables the transmission of automatic Pause frames with TIME value zero. It is transmitted when the ETHA negates the pause request in automatic pause frame mode or when the software negates the MTPFC2.MPFR bit in manual pause frame mode and the remaining pause time is more than PFRT.
This bit has no effect for Manual Pause frame transmission.
[Restriction]

- SW: This bit is only writable in CONFIG mode.

[Note]

- This bit is used for PAUSE frame only.

MPFR Bit (Manual Pause Frame Request)

If a pause frame is requested so fast that the hardware has have time to send the previous one (because a long frame is ongoing or because the software requests too many pause frames), the hardware will ignore the pause frame requests.
[Setting condition]

- SW: Writing 1 to this bit will set the bit.

[Clearing conditions]

- SW: Writing 0 to this bit will clear the bit.
- HW: Going out of OPERATION mode will clear this bit.

[Restriction]

- SW: This bit is only writable in OPERATION mode.

[Note]

- This bit is used for a PAUSE frame only.

37.2.1.8 Transmission Pause or PFC Frame Configuration Register 3t (MTPFC3t) (t = 0, 1)

Address(es): RMAC0.MTPFC30 874C B030h, RMAC0.MTPFC31 874C B034h, RMAC0_NS.MTPFC30 074C B030h, RMAC0_NS.MTPFC31 074C B034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PFCPG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PFCPG[7:0]	PFC Priority Group	0: Priority x is not assigned to priority group t. 1: Priority x is assigned to group t.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to set the TIME parameter value for PFC frames.

PFCPG[7:0] Bits (PFC Priority Group)

[When MTFFC.FCM = 1 (PFC)]

- When PFC frame transmission request (hardware request for automatic request, MTPFC2.MPFCFR for manual request) is set, PFC frame is generated. Then, the Field of Class-Enable vector is set to these bits value.
- When all bits of PFCPG[7:0] = 00h, the transmission request is ignored.

[When MTFFC.FCM = 0 (PAUSE)]

- PAUSE frame transmission request is set by hardware pause request or MTPFC2.MPFR.
- When PFCPG[0] = 0, the transmission request is ignored.

[Restriction]

- SW: These bits are only writable in CONFIG mode.

37.2.1.9 Transmission Flow Control JAM Configuration Register (MTFCJC)

Address(es): RMAC0.MTFCJC 874C B070h, RMAC0_NS.MTFCJC 074C B070h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MJOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	JM	—	—	—	—	—	—	—	JE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	JE	JAM Enable	0: JAM disabled 1: JAM enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	JM	JAM Mode Select	0: Automatic JAM 1: Manual JAM	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	MJOE	Manual JAM Output Enable	0: Transmission disabled 1: Transmission enabled	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to set jam output configuration for back pressure for flow control on half-duplex mode.

JE Bit (JAM Enable)

[Restrictions]

- SW: This bit is only writable in CONFIG mode.

[Note]

- This bit should be used only half-duplex mode.

JM Bit (JAM Mode Select)

[Restriction]

- SW: This bit is only writable in CONFIG mode.

[Note]

- This bit should be used only half-duplex mode.
- Assertion/De-assertion levels are set by CABPPFLC and CABPPPFLC registers. In manual mode, CABPPFLC and CABPPPFLC are ignored.

MJOE Bit (Manual JAM Output Enable)

[Restriction]

- SW: This bit is only writable in OPERATION mode.

[Note]

- This bit should be used only half-duplex mode.

37.2.1.10 Reception General Configuration Register (MRGC)

Address(es): RMAC0.MRGC 874C B080h, RMAC0_NS.MRGC 074C B080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	PFCRC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	RFCFE	MPDE	PFRTZ	PFRC	RCPT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCPT	Receive CRC Pass Through	0: Correct FCS is not passed to the ETHA. Incorrect FCS is passed to the ETHA. 1: Both Correct and Incorrect FCS are passed to the ETHA.	R/W
b1	PFRC	Pause Frame Reception Control	0: Flow control for the receiving port is disabled. 1: Flow control for the receiving port is enabled.	R/W
b2	PFRTZ	Pause or PFC Frame Reception with Time = 0	0: Reception of Pause or PFC frames with the TIME value 0 is disabled. 1: Reception of Pause or PFC frames with the TIME value 0 is enabled.	R/W
b3	MPDE	Magic Packet Detection Enable	0: Magic Packet detection is disabled. 1: Magic Packet detection is enabled.	R/W
b4	RFCFE	Reception Flow Control Forwarding Enable	0: Flow Control Frame (PAUSE or PFC frame) is not forward to receive ETHA interface. 1: Flow Control Frame (PAUSE or PFC frame) is forward to receive ETHA interface.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23 to b16	PFCRC[7:0]	PFC Frame Reception Control	0: Flow control for the receiving priority level x(7:0) is disabled. 1: Flow control for the receiving priority level x(7:0) is enabled.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to configure the frame reception.

RCPT Bit (Receive CRC Pass Through)

This bit selects to pass FCS field of the received frame. The CRC is checked regardless of this configuration.

PFRC Bit (Pause Frame Reception Control)

This bit enables the Flow Control (Pause frame detection) function.

[Restrictions]

- HW: Only for e-frames
- SW: When any bits of MRGC.PFCRC[7:0] is set, SW should not set this bit.

PFRTZ Bit (Pause or PFC Frame Reception with Time = 0)

When this bit is set to 1, the reception of Pause or PFC frame with the Timer value 0 releases the transmission wait state.

MPDE Bit (Magic Packet Detection Enable)

This bit enables Magic Packet detection by hardware to allow activation via the Ethernet connection.
[Restriction]

- HW: Only for e-frames

RFCFE Bit (Reception Flow Control Forwarding Enable)

This bit enables the Flow Control Frame (PAUSE or PFC frame) forwarding function.

PFCRC[7:0] Bits (PFC Frame Reception Control)

This bit enables the Flow Control (Pause frame detection) function.
[Restrictions]

- HW: Only for e-frames
- SW: When MRGC.PFRC is set, SW should not set this bit.

37.2.1.11 Reception MAC Address Configuration Register 0 (MRMAC0)

Address(es): RMAC0.MRMAC0 874C B084h, RMAC0_NS.MRMAC0 074C B084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MAU[15:0]	MAC Address Upper Part	These bits are used to set the upper 16 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 0123h in this register.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Specifies the upper 16 bits of the 48-bit unicast MAC address. This register is also used as source address field when the IP transmits a Pause frame or a PFC frame.

37.2.1.12 Reception MAC Address Configuration Register 1 (MRMAC1)

Address(es): RMAC0.MRMAC1 874C B088h, RMAC0_NS.MRMAC1 074C B088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MAL[31:0]	MAC Address Lower Part	These bits are used to set the lower 32 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 456789ABh in this register.	R/W

The register is to use to specify the lower 32 bits of the 48-bit unicast MAC address. This register is also used as source address field when the IP transmits a Pause frame or a PFC frame.

37.2.1.13 Reception Address Filter Configuration Register (MRAFC)

Address(es): RMAC0.MRAFC 874C B08Ch, RMAC0_NS.MRAFC 074C B08Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSAREE	NSAREE	SDSFREE	NDAREE	BCACE	MCACE	BSTENE	MSTENE	BCENE	MCENE	UCENE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	UCENE	Unicast Reception Enable E-Frames	0: Unicast reception disabled 1: Unicast reception enabled	R/W
b1	MCENE	Multicast Reception Enable E-Frames	0: Multicast reception disabled 1: Multicast reception enabled	R/W
b2	BCENE	Broadcast Reception Enable E-Frames	0: Broadcast reception disabled 1: Broadcast reception enabled	R/W
b3	MSTENE	Multicast Storm Filter Reception Enable E-Frames	0: Multicast storm filter reception disabled 1: Multicast storm filter reception enabled	R/W
b4	BSTENE	Broadcast Storm Filter Reception Enable E-Frames	0: Broadcast storm filter reception disabled 1: Broadcast storm filter reception enabled	R/W
b5	MCACE	Multicast Storm Autoclear E-Frames	0: Multicast Storm AutoClear disabled 1: Multicast Storm AutoClear enabled	R/W
b6	BCACE	Broadcast Storm Autoclear E-Frames	0: Broadcast Storm AutoClear disabled 1: Broadcast Storm AutoClear enabled	R/W
b7	NDAREE	Null Destination Address Reception Enable E-Frames	0: Disable 1: Enable	R/W
b8	SDSFREE	Same DA and SA Frames Reception Enable E-Frames	0: Disable 1: Enable	R/W
b9	NSAREE	Null Source Address Reception Enable E-Frames	0: Disable 1: Enable	R/W
b10	MSAREE	Multicast Source Address Reception Enable E-Frames	0: Disable 1: Enable	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

UCENE Bit (Unicast Reception Enable E-Frames)

This bit enables the unicast filtering based on MAC address (MRMAC0, MRMAC1).

MCENE Bit (Multicast Reception Enable E-Frames)

This bit enables the multicast filtering.

BCENE Bit (Broadcast Reception Enable E-Frames)

This bit enables the broadcast filtering.

MSTENE Bit (Multicast Storm Filter Reception Enable E-Frames)

This bit enables the multicast storm filtering. When it is asserted, Multicast frames are rejected after the number of received multicast e-frames reaches the configured threshold (MRSCE.CMFE[15:0]).

[Restriction]

- SW: It is valid only when Multicast reception is enabled (MRAFC.MCENE = 1).

BSTENE Bit (Broadcast Storm Filter Reception Enable E-Frames)

This bit enables the broadcast storm filtering. When it is asserted, Broadcast frames are rejected after the number of received broadcast e-frames reaches to the configured threshold (MRSCE.CBFE[15:0]).

[Restriction]

- SW: It is valid only when Broadcast reception is enabled (MRAFC.BCENE = 1).

MCACE Bit (Multicast Storm Autoclear E-Frames)

This bit enables auto-clear of multicast frame storm counter. When it is asserted, multicast frame storm counter is cleared by detecting a frame with non-multicast destination address. After the counter is cleared, the IP can receive multicast frame again.

[Restriction]

- HW: Broadcast frames are not taken in account by multicast storm counter.

BCACE Bit (Broadcast Storm Autoclear E-Frames)

This bit enables auto-clear of broadcast frame storm counter. When it is asserted, broadcast frames storm counter is cleared by detecting a frame with non-broadcast destination address. After the counter is cleared, the IP can receive Broadcast frame again.

NDAREE Bit (Null Destination Address Reception Enable E-Frames)

This bit enables the reception of e-frames with destination MAC address null.

SDSFREE Bit (Same DA and SA Frames Reception Enable E-Frames)

This bit enables the reception of e-frame with equal destination address and source address.

NSAREE Bit (Null Source Address Reception Enable E-Frames)

This bit enables the reception of e-frames with source MAC address null.

MSAREE Bit (Multicast Source Address Reception Enable E-Frames)

This bit enables the reception of e-frame with multicast and broadcast source address.

37.2.1.14 Reception Storm Configuration for E-Frames Register (MRSCE)

Address(es): RMAC0.MRSCE 874C B090h, RMAC0_NS.MRSCE 074C B090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CBFE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMFE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CMFE[15:0]	Consecutive Multicast Frame Reception Count Setting for E-Frames	b15 b0 0000h: 1 Multicast frame accepted until multicast frame counter reset. 0001h: 2 Multicast frames accepted until multicast frame counter reset. : : FFFEh: 65535 Multicast frames accepted until multicast frame counter reset. Settings other than above are prohibited.	R/W
b31 to b16	CBFE[15:0]	Consecutive Broadcast Frame Reception Count Setting for E-Frames	b31 b16 0000h: 1 Broadcast frame accepted until broadcast frame counter reset. 0001h: 2 Broadcast frames accepted until broadcast frame counter reset. : : FFFEh: 65535 Broadcast frames accepted until broadcast frame counter reset. Settings other than above are prohibited.	R/W

This register configures the number of times how many broadcast/multicast frames can be received consecutively.

CMFE[15:0] Bits (Consecutive Multicast Frame Reception Count Setting for E-Frames)

These bits specify the upper-limitation of received consecutive multicast frames. After the number of received consecutive multicast frames reaches the set value, the subsequent multicast frames are discarded.

CBFE[15:0] Bits (Consecutive Broadcast Frame Reception Count Setting for E-Frames)

These bits specify the upper-limitation of received consecutive broadcast frames. After the number of received consecutive broadcast frames reaches the set value, the subsequent broadcast frames are discarded.

37.2.1.15 Reception Storm Counter Configuration Register (MRSCC)

Address(es): RMAC0.MRSCC 874C B098h, RMAC0_NS.MRSCC 074C B098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BSCCE	MSCCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSCCE	Multicast Storm Counter Clear E-Frames	Writing 1 to this bit clears the multicast storm counter. This bit is read as 0.	R/W
b1	BSCCE	Broadcast Storm Counter Clear E-Frames	Writing 1 to this bit clears the broadcast storm counter. This bit is read as 0.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17, b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

Register to clear Broadcast and Multicast Storm Counters.

MSCCE Bit (Multicast Storm Counter Clear E-Frames)

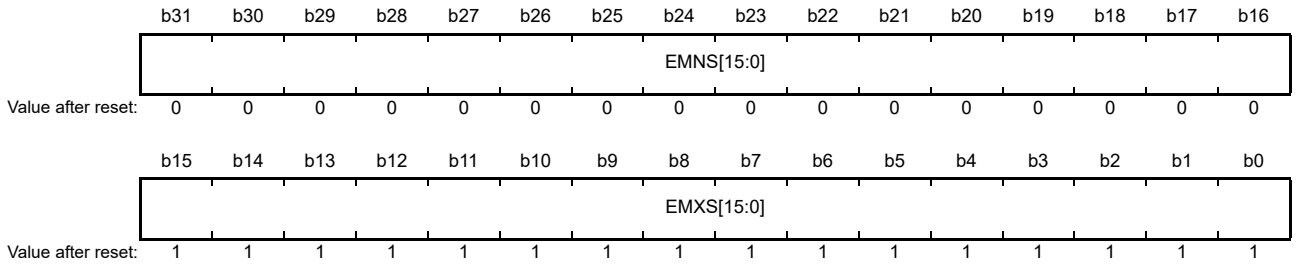
This bit requests to clear the multicast frames storm counter.
After the counter is cleared, the IP can receive multicast frames again until the upper-limitation of received consecutive multicast frames (MRSCE.CMFE[15:0]).
It should be used with multicast Storm AutoClear disabled (MRAFC.MCACE = 0).

BSCCE Bit (Broadcast Storm Counter Clear E-Frames)

This bit requests to clear the broadcast frames storm counter.
After the counter is cleared, the IP can receive Broadcast frames again until the upper-limitation of received consecutive Broadcast frames (MRSCE.CBFE[15:0]).
It should be used with Broadcast Storm AutoClear disabled (MRAFC.BCACE = 0).

37.2.1.16 Reception Frame Size Configuration for E-Frames Register (MRFSCE)

Address(es): RMAC0.MRFSCE 874C B09Ch, RMAC0_NS.MRFSCE 074C B09Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EMXS[15:0]	E-Frame Maximum Size	E-Frame maximum size	R/W
b31 to b16	EMNS[15:0]	E-Frame Minimum Size	E-Frame minimum size	R/W

Note: The size limitations concern the frame size RMAC outputs to an upper layer (e.g. ETHA), and so the FCS removal impacts the register values.

EMXS[15:0] Bits (E-Frame Maximum Size)

A part of received frame which exceeds the set value are truncated.

Do not set a value under 16 bytes

37.2.1.17 Timestamp Reception Configuration Register (MTRC)

Address(es): RMAC0.MTRC 874C B0A4h,RMAC0_NS.MTRC 074C B0A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	DTN	—	TCTSE	—	TRDDE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRHFME1	TRHFME0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRHFME0	Timestamp Reception Hardware Filter Match Enable 0	0: Disable 1: Enable	R/W
b1	TRHFME1	Timestamp Reception Hardware Filter Match Enable 1	0: Disable 1: Enable	R/W
b23 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	TRDDE	Timestamp Reception Default Disable E-Frame	0: Enable 1: Disable	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26	TCTSE	Timestamp Capture on Transmit Side E-Frame	0: Disable 1: Enable	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b28	DTN	Default timer number	The bit set Timer used for timestamp forwarding on receive ETHA interface when no timer is detected.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

TRHFME0 Bit (Timestamp Reception Hardware Filter Match Enable 0)

Setting this bit to 1 will enable hardware filtering for timer 0.
When this bit is set to 1, at least one of the in the MPFCt.TEF0 bits (t = 0 to 15) should be set to 1.
When hardware filter matches, the timestamp from the matched timer will be sent along with the frame.

TRHFME1 Bit (Timestamp Reception Hardware Filter Match Enable 1)

Setting this bit to 1 will enable hardware filtering for timer 1.
When this bit is set to 1, at least one of the in the MPFCt.TEF1 bits (t = 0 to 15) should be set to 1.
When hardware filter matches, the timestamp from the matched timer will be sent along with the frame.

TRDDE Bit (Timestamp Reception Default Disable E-Frame)

A timestamp for every frame will be transmitted to the ETHA. This timestamp will be taken on the default timer except when TRHFMEn is set and the hardware filter matches.

TCTSE Bit (Timestamp Capture on Transmit Side E-Frame)

The bit enables the timestamp capture on transmit side for e-frames

37.2.1.18 Reception Pause or PFC Frame Monitoring Register (MRPFM)

Address(es): RMAC0.MRPFM 874C B0ACh, RMAC0_NS.MRPFM 074C B0ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	PFCTCA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PTCA	Pause Time Counting Active	0: No pause state 1: Pause state	R
b15 to b1	—	Reserved	These bits are read as 0.	R
b23 to b16	PFCTCA[7:0]	PFC Time Counting Active	0: No pause state 1: Pause state	R
b31 to b24	—	Reserved	These bits are read as 0.	R

PTCA Bit (Pause Time Counting Active)

This bit indicates that the Pause time of a received Pause frame has not been completed.

PFCTCA[7:0] Bits (PFC Time Counting Active)

This bit indicates that the Pause time of a received PFC frame has not been completed.

37.2.1.19 PTP Filtering Register Configuration Register t (MPFCt) (t = 0 to 15)

Address(es): RMAC0.MPFC0 874C B100h, RMAC0.MPFC1 874C B104h, RMAC0.MPFC2 874C B108h, RMAC0.MPFC3 874C B10Ch, RMAC0.MPFC4 874C B110h, RMAC0.MPFC5 874C B114h, RMAC0.MPFC6 874C B118h, RMAC0.MPFC7 874C B11Ch, RMAC0.MPFC8 874C B120h, RMAC0.MPFC9 874C B124h, RMAC0.MPFC10 874C B128h, RMAC0.MPFC11 874C B12Ch, RMAC0.MPFC12 874C B130h, RMAC0.MPFC13 874C B134h, RMAC0.MPFC14 874C B138h, RMAC0.MPFC15 874C B13Ch, RMAC0_NS.MPFC0 074C B100h, RMAC0_NS.MPFC1 074C B104h, RMAC0_NS.MPFC2 074C B108h, RMAC0_NS.MPFC3 074C B10Ch, RMAC0_NS.MPFC4 074C B110h, RMAC0_NS.MPFC5 074C B114h, RMAC0_NS.MPFC6 074C B118h, RMAC0_NS.MPFC7 074C B11Ch, RMAC0_NS.MPFC8 074C B120h, RMAC0_NS.MPFC9 074C B124h, RMAC0_NS.MPFC10 074C B128h, RMAC0_NS.MPFC11 074C B12Ch, RMAC0_NS.MPFC12 074C B130h, RMAC0_NS.MPFC13 074C B134h, RMAC0_NS.MPFC14 074C B138h, RMAC0_NS.MPFC15 074C B13Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TEF1	TEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFBV[7:0]								PFBN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PFBN[7:0]	PTP Filtering Byte Number	0: Byte number 0 will be filtered : 255: Byte number 255 will be filtered	R/W
b15 to b8	PFBV[7:0]	PTP Filtering Byte Value	When the filtered byte has a value equal to these bits, the byte will be eligible for the corresponding timers	R/W
b16	TEF0	Timer Enable for Filtering 0	Setting this bit to 1 will apply the filter to timer 0.	R/W
b17	TEF1	Timer Enable for Filtering 1	Setting this bit to 1 will apply the filter to timer 1.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

The register is used to configure the PTP frame filtering for time domain selection.

Note: This register is used for receive time domain selection.
When an overlapping configuration is done, HW will always choose the timer with the smaller number.

37.2.1.20 Energy Efficient Ethernet Configuration Register (MEEEC)

Address(es): RMAC0.MEEEC 874C B184h, RMAC0_NS.MEEEC 074C B184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPITR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LPITR	LPI Transmit Request	0: LPI mode is not requested 1: LPI mode is requested	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

LPITR Bit (LPI Transmit Request)

This bit requests to enter Low Power Idle (LPI) mode. RMAC enters to LPI state after all ongoing transmissions have been completed. When this bit is negated, this IP leaves LPI state.
LPI signal should be transmitted only when transmit is disabled.
[Restriction]

- SW: LPITR bit can only be set when transmission request to RMAC is stopped, and transmission request can only be allowed after LPITR is cleared. It is prohibited to request transmission from ETHA when LPITR bit is set.

Going out of OPERATION mode will clear this register.

37.2.1.21 Loopback Configuration Register (MLBC)

Address(es): RMAC0.MLBC 874C B188h, RMAC0_NS.MLBC 074C B188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LBME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LBME	Loopback Mode Enable	0: Disable 1: Enable	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

LBME Bit (Loopback Mode Enable)

Transmit xMII signals are connected to reception xMII signals internally.

[Restrictions]

- When loopback mode is enabled, frame received by external PHY Interface are ignored.
- If this bit is set, clk_phy_tx and clk_phy_rx must be synchronized with each other.

37.2.2 Counter Registers

Note: All counter registers in RMAC are read clear registers.
When counters reach their own maximum value, they stop.

37.2.2.1 Manual Pause Frame Transmit Counter Register (MMPFTCT)

Address(es): RMAC0.MMPFTCT 874C B300h, RMAC0_NS.MMPFTCT 074C B300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MPFTC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MPFTC[15:0]	Manual Pause Frame Transmit Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

MPFTC[15:0] Bits (Manual Pause Frame Transmit Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when manual Pause frame is transmitted.

37.2.2.2 Automatic Pause Frame Transmit Counter Register (MAPFTCT)

Address(es): RMAC0.MAPFTCT 874C B304h, RMAC0_NS.MAPFTCT 074C B304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	APFTC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	APFTC[15:0]	Automatic Pause Frame Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

APFTC[15:0] Bits (Automatic Pause Frame Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when automatic Pause frame is transmitted.

37.2.2.3 Pause Frame Receive Counter Register (MPFRCT)

Address(es): RMAC0.MPFRCT 874C B308h, RMAC0_NS.MPFRCT 074C B308h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFRC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PFRC[15:0]	Pause Frame Receive Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

PFRC[15:0] Bits (Pause Frame Receive Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when automatic Pause frame is received.

[Restriction]

- SW: Set condition is only available when Flow Control is enabled in reception (MRGC.PFRC = 1).

37.2.2.4 False Carrier Indication Counter Register (MFCICT)

Address(es): RMAC0.MFCICT 874C B30Ch, RMAC0_NS.MFCICT 074C B30Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCIC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	FCIC[15:0]	False Carrier Indication Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

Note: This register is counter of “False Carrier Indication” event. This event goes through the synchronizer between clk and clk_phy_rx. If the event occurs continuously, there is a possibility of miss counting.

FCIC[15:0] Bits (False Carrier Indication Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when False Carrier Indication pattern is detected.

37.2.2.5 Energy Efficient Ethernet Counter Register (MEEECT)

Address(es): RMAC0.MEEECT 874C B310h, RMAC0_NS.MEEECT 074C B310h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EEERC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EEERC[15:0]	Energy Efficient Ethernet Receive Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

EEERC[15:0] Bit (Energy Efficient Ethernet Receive Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when LPI signal is detected on xMII reception bus.

37.2.2.6 Manual PFC Frame Transmit Counter Register t (MMPCFTCTt) (t = 0, 1)

Address(es): RMAC0.MMPCFTCT0 874C B320h, RMAC0.MMPCFTCT1 874C B324h, RMAC0_NS.MMPCFTCT0 074C B320h, RMAC0_NS.MMPCFTCT1 074C B324h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MPCFCTC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MPCFCTC[15:0]	Manual PFC Frame Transmit Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

MPCFCTC[15:0] Bits (Manual PFC Frame Transmit Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when manual PFC frames is transmitted.

37.2.2.7 Automatic PFC Frame Transmit Counter Register t (MAPCFTCTt) (t = 0, 1)

Address(es): RMAC0.MAPCFTCT0 874C B330h, RMAC0.MAPCFTCT1 874C B334h, RMAC0_NS.MAPCFTCT0 074C B330h, RMAC0_NS.MAPCFTCT1 074C B334h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	APCFCTC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	APCFCTC[15:0]	Automatic PFC Frame Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

APCFCTC[15:0] Bits (Automatic PFC Frame Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when automatic PFC frames is transmitted.

37.2.2.8 PFC Frame Receive Counter Register t (MPCFRCTt) (t = 0 to 7)

Address(es): RMAC0.MPCFRCT0 874C B340h, RMAC0.MPCFRCT1 874C B344h, RMAC0.MPCFRCT2 874C B348h, RMAC0.MPCFRCT3 874C B34Ch, RMAC0.MPCFRCT4 874C B350h, RMAC0.MPCFRCT5 874C B354h, RMAC0.MPCFRCT6 874C B358h, RMAC0.MPCFRCT7 874C B35Ch,
RMAC0_NS.MPCFRCT0 074C B340h, RMAC0_NS.MPCFRCT1 074C B344h, RMAC0_NS.MPCFRCT2 074C B348h, RMAC0_NS.MPCFRCT3 074C B34Ch, RMAC0_NS.MPCFRCT4 074C B350h, RMAC0_NS.MPCFRCT5 074C B354h, RMAC0_NS.MPCFRCT6 074C B358h, RMAC0_NS.MPCFRCT7 074C B35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCFCRC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PCFCRC[15:0]	PFC Frame Receive Counter		R
b31 to b16	—	Reserved	These bits are read as 0.	R

PCFCRC[15:0] Bits (PFC Frame Receive Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when PFC frames is received.

[Restriction]

- SW: Set condition is only available when Flow Control is enabled in reception (MRGC.PFRC = 1).

37.2.2.9 Receive Overflow Counter Register (MROVFC)

Address(es): RMAC0.MROVFC 874C B360h, RMAC0_NS.MROVFC 074C B360h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ROVFC[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ROVFC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	ROVFC[31:0]	Receive Overflow Counter		R

ROVFC[31:0] Bits (Receive Overflow Counter)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- HW: Incremented by 1 when frame receive overflow occurred (when set condition for MEIS.REOES or MEIS.RPOES are met).

37.2.2.10 Received Byte Counter E-Frames Upper Side Register (MRXBCEU)

Address(es): RMAC0.MRXBCEU 874C B44Ch, RMAC0_NS.MRXBCEU 074C B44Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RBNEU[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBNEU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBNEU[31:0]	Received Byte Number E-Frames Upper Side		R

RBNEU[31:0] Bits (Received Byte Number E-Frames Upper Side)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

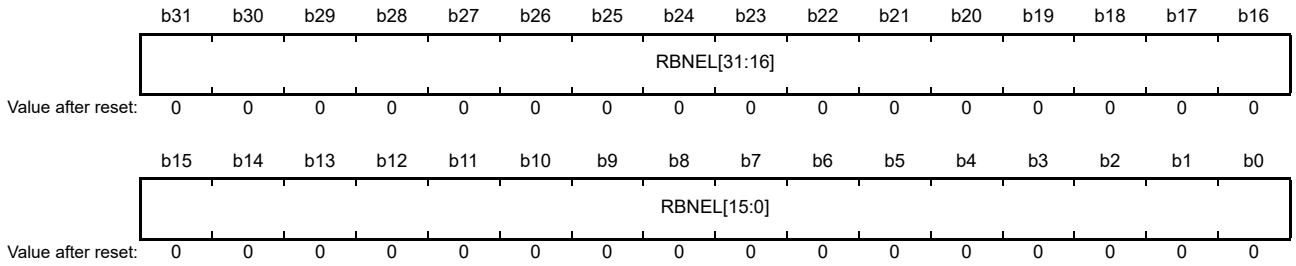
- Incremented each time the received data of e-Frame is sent to the receive ETHA interface regardless error is detected.

[Notes]

- When the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation.
- Read before reading MRXBCEL.

37.2.2.11 Received Byte Counter E-Frames Lower Side Register (MRXBCEL)

Address(es): RMAC0.MRXBCEL 874C B450h, RMAC0_NS.MRXBCEL 074C B450h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBNEL[31:0]	Received Byte Number E-Frames Lower Side		R

RBNEL[31:0] Bits (Received Byte Number E-Frames Lower Side)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.
- SW: None. (Reading MRXBCEL register holds it.)

[Increment condition]

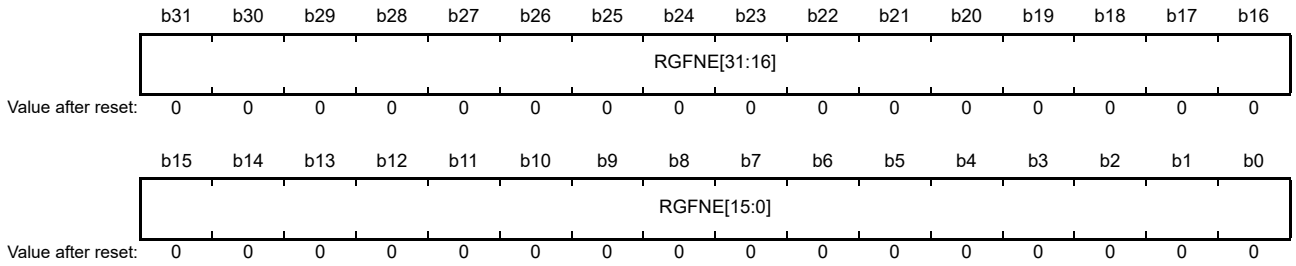
- Incremented each time the received data of e-frame is sent to the receive ETHA interface regardless the error detection.

[Notes]

- When the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation.
- Read after reading MRXBCEU.

37.2.2.12 Received Good Frame Counter E-Frames Register (MRGFCE)

Address(es): RMAC0.MRGFCE 874C B408h, RMAC0_NS.MRGFCE 074C B408h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RGFNE[31:0]	Received Good Frame Number E-Frames		R

RGFNE[31:0] Bits (Received Good Frame Number E-Frames)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when the reception of e-frame is completed without error.

37.2.2.13 Received Good Broadcast Frame Counter Register (MRBFC)

Address(es): RMAC0.MRBFC 874C B410h, RMAC0_NS.MRBFC 074C B410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RBFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBFN[31:0]	Received Good Broadcast Frame Number		R

RBFN[31:0] Bits (Received Good Broadcast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when destination address field of broadcast frame is sent to receive ETHA interface without error.

37.2.2.14 Received Good Multicast Frame Counter Register (MRMFC)

Address(es): RMAC0.MRMFC 874C B414h, RMAC0_NS.MRMFC 074C B414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RMFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RMFN[31:0]	Received Good Multicast Frame Number		R

RMFN[31:0] Bits (Received Good Multicast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when destination address field of multicast frame is sent to receive ETHA interface without error.

37.2.2.15 Received Good Unicast Frame Counter Register (MRUFC)

Address(es): RMAC0.MRUFC 874C B418h, RMAC0_NS.MRUFC 074C B418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RUFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RUFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RUFN[31:0]	Received Good Unicast Frame Number		R

RUFN[31:0] Bits (Received Good Unicast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when destination address field of unicast frame is sent to receive ETHA interface without error.

37.2.2.16 Received PHY Error Frame Count Register (MRPEFC)

Address(es): RMAC0.MRPEFC 874C B41Ch, RMAC0_NS.MRPEFC 074C B41Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RPEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RPEFN[15:0]	Received PHY Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RPEFN[15:0] Bits (Received PHY Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with phy error.

37.2.2.17 Received Nibble Error Frame Count Register (MRNEFC)

Address(es): RMAC0.MRNEFC 874C B420h, RMAC0_NS.MRNEFC 074C B420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RNEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RNEFN[15:0]	Received Nibble Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RNEFN[15:0] Bits (Received Nibble Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with nibble error.

37.2.2.18 Received FCS/mCRC Error Frame Count Register (MRFMEFC)

Address(es): RMAC0.MRFMEFC 874C B424h, RMAC0_NS.MRFMEFC 074C B424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFMEFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFMEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RFMEFN[31:0]	Received FCS/mCRC Error Frame Number		R

RFMEFN[31:0] Bits (Received FCS/mCRC Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with FCS/mCRC error.

37.2.2.19 Received Final Fragment Missing Error Frame Count Register (MRFFMEFC)

Address(es): RMAC0.MRFFMEFC 874C B428h, RMAC0_NS.MRFFMEFC 074C B428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFFMEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFFMEFN[15:0]	Received Final Fragment Missing Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RFFMEFN[15:0] Bits (Received Final Fragment Missing Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with final fragment error.

37.2.2.20 Received C-Fragment Count Error Frame Count Register (MRCFCEFC)

Address(es): RMAC0.MRCFCEFC 874C B42Ch, RMAC0_NS.MRCFCEFC 074C B42Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCFCEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RCFCEFN[15:0]	Received C-Fragment Count Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RCFCEFN[15:0] Bits (Received C-Fragment Count Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with C-fragment count error.

37.2.2.21 Received Fragment Count Error Frame Count Register (MRFCEFC)

Address(es): RMAC0.MRFCEFC 874C B430h, RMAC0_NS.MRFCEFC 074C B430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFCEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFCEFN[15:0]	Received Fragment Count Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RFCEFN[15:0] Bits (Received Fragment Count Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with fragment count error.

37.2.2.22 Received MAC Filter Error Frame Count Register (MRRCFEFC)

Address(es): RMAC0.MRRCFEFC 874C B434h, RMAC0_NS.MRRCFEFC 074C B434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RRCFEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RRCFEFN[15:0]	Received RMAC Filter Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

RRCFEFN[15:0] Bits (Received RMAC Filter Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with RMAC filtered.

37.2.2.23 Received Frame Count Register (MRFC)

Address(es): RMAC0.MRFC 874C B438h, RMAC0_NS.MRFC 074C B438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RFN[31:0]	Received Frame Number		R

RFN[31:0] Bits (Received Frame Number)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received frame is sent to receive ETHA interface.

37.2.2.24 Received Good Undersize Error Frame Count Register (MRGUEFC)

Address(es): RMAC0.MRGUEFC 874C B43Ch, RMAC0_NS.MRGUEFC 074C B43Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RGUEFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RGUEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RUEFN[31:0]	Received Good Undersize Error Frame Number		R

RUEFN[31:0] Bits (Received Good Undersize Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with undersize error but without fcs/mcrc error.

37.2.2.25 Received Bad Undersize Error Frame Count Register (MRBUEFC)

Address(es): RMAC0.MRBUEFC 874C B440h, RMAC0_NS.MRBUEFC 074C B440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RBUEFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBUEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBUEFN[31:0]	Received Bad Undersize Error Frame Number		R

RBUEFN[31:0] Bits (Received Bad Undersize Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when received data is sent to receive ETHA interface with undersize error and fcs/mcrc error.

37.2.2.26 Received Good Oversize Error Frame Count Register (MRGOEFC)

Address(es): RMAC0.MRGOEFC 874C B444h, RMAC0_NS.MRGOEFC 074C B444h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RGOEFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RGOEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RGOEFN[31:0]	Received Good Oversize Error Frame Number		R

RGOEFN[31:0] Bits (Received Good Oversize Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with oversize error, but without fcs/mcrc error.

37.2.2.27 Received Bad Oversize Error Frame Count Register (MRBOEFC)

Address(es): RMAC0.MRBOEFC 874C B448h, RMAC0_NS.MRBOEFC 074C B448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RBOEFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBOEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBOEFN[31:0]	Received Bad Oversize Error Frame Number		R

RBOEFN[31:0] Bits (Received Bad Oversize Error Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Incremented when received data is sent to receive ETHA interface with oversize error and fcs/mcrc error.

37.2.2.28 Transmitted Byte Counter E-Frames Upper Side Register (MTXBCEU)

Address(es): RMAC0.MTXBCEU 874C B520h, RMAC0_NS.MTXBCEU 074C B520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TBNEU[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TBNEU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBNEU[31:0]	Transmitted Byte Number E-Frames Upper Side		R

Note: If the FCS is not passed to the upper layer FCS is not taken in account in total byte calculation.

TBNEU[31:0] Bits (Transmitted Byte Number E-Frames Upper Side)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment conditions]

- Increment every time transmit data for e-frame is received on transmit ETHA interface.
- Read before reading MTXBCEL.

37.2.2.29 Transmitted Byte Counter E-Frames Lower Side Register (MTXBCEL)

Address(es): RMAC0.MTXBCEL 874C B524h, RMAC0_NS.MTXBCEL 074C B524h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TBNEL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TBNEL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBNEL[31:0]	Transmitted Byte Number E-Frames Lower Side		R

Note: If the FCS is not passed to the upper layer FCS is not taken in account in total byte calculation.

TBNEL[31:0] Bits (Transmitted Byte Number E-Frames Lower Side)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: None. (Reading MTXBCEL register holds it.)

[Increment conditions]

- Increment every time transmit data for e-frame is received on transmit ETHA interface.
- Read after reading MTXBCEU.

37.2.2.30 Transmitted Good Frame Counter E-Frames Register (MTGFCE)

Address(es): RMAC0.MTGFCE 874C B508h, RMAC0_NS.MTGFCE 074C B508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TGFNE[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TGFNE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TGFNE[31:0]	Transmitted Good Frame Number E-Frames		R

TGFNE[31:0] Bits (Transmitted Good Frame Number E-Frames)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when transmission of e-frame is completed on transmit ETHA interface without error.

37.2.2.31 Transmitted Broadcast Frame Counter Register (MTBFC)

Address(es): RMAC0.MTBFC 874C B510h, RMAC0_NS.MTBFC 074C B510h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TBFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TBFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBFN[31:0]	Transmitted Broadcast Frame Number		R

TBFN[31:0] Bits (Transmitted Broadcast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when destination address field of broadcast frame is sent on transmit ETHA interface.

37.2.2.32 Transmitted Multicast Frame Counter Register (MTMFC)

Address(es): RMAC0.MTMFC 874C B514h, RMAC0_NS.MTMFC 074C B514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TMFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TMFN[31:0]	Transmitted Multicast Frame Number		R

TMFN[31:0] Bits (Transmitted Multicast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when destination address field of multicast frame is sent on transmit ETHA interface.

37.2.2.33 Transmitted Unicast Frame Counter Register (MTUFC)

Address(es): RMAC0.MTUFC 874C B518h, RMAC0_NS.MTUFC 074C B518h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TUFN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TUFN[31:0]	Transmitted Unicast Frame Number		R

TUFN[31:0] Bits (Transmitted Unicast Frame Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when destination address field of unicast frame is sent on transmit ETHA interface.

37.2.2.34 Transmitted Error Frame Counter Register (MTEFC)

Address(es): RMAC0.MTEFC 874C B51Ch, RMAC0_NS.MTEFC 074C B51Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEFN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TEFN[15:0]	Transmitted Error Frame Number		R
b31 to b16	—	Reserved	These bits are read as 0.	R

TEFN[15:0] Bits (Transmitted Error Frame Number)

[Clearing conditions]

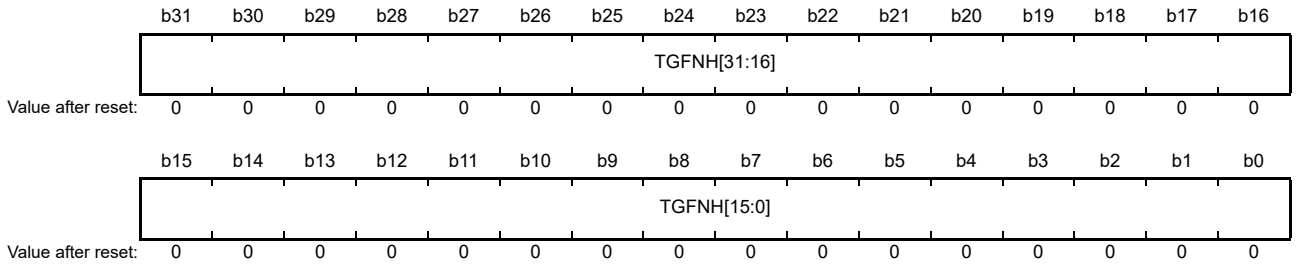
- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when frame transmission is completed on transmit ETHA interface with error.

37.2.2.35 Transmitted Half Duplex Good Frame Counter Register (MTHDGFC)

Address(es): RMAC0.MTHDGFC 874C B530h, RMAC0_NS.MTHDGFC 074C B530h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TGFNH[31:0]	Transmitted Good Frame Number		R

Note: The good frame is a successfully transmitted frame without collisions or carrier loss.

TGFNH[31:0] Bits

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears them.

[Increment condition]

- Increment when transmission of half-duplex-frame is completed on transmit ETHA interface without error.

37.2.3 Interrupt Registers

37.2.3.1 Error Interrupt Status Register (MEIS)

Address(es): RMAC0.MEIS 874C B200h, RMAC0_NS.MEIS 074C B200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FOES	FUES	FFS	RPOOMS	FRCES	CFCES	FFMES	FCMCES	PNAES	PDES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTLES1	CTLES0	—	—	REOES	FCES	BFES	TBCIS	TCES	FCDS	PFRROS	PRES	—	TSLS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSLS	Transmission Stream Lost Status Flag	0: No Transmission Stream Lost 1: Transmission Stream Lost occurred.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	PRES	Pause or PFC Frame Reception Error Status Flag	0: No Pause or PFC Frame Reception Error 1: Pause or PFC Frame Reception Error occurred.	R/W
b3	PFRROS	Pause or PFC Frame Retransmit Retry Over Status Flag	0: No Pause or PFC Frame Retransmit Retry Over 1: Pause or PFC Frame Retransmit Retry Over occurred.	R/W
b4	FCDS	False Carrier Detection Status Flag	0: No False Carrier Detection 1: False Carrier Detection occurred.	R/W
b5	TCES	Transmit CRC Error Status Flag	0: No Transmit CRC Error 1: Transmit CRC Error occurred.	R/W
b6	TBCIS	Transmit Bad CRC Insertion Status Flag	0: No Transmit Bad CRC Insertion 1: Transmit Bad CRC Insertion occurred.	R/W
b7	BFES	Bad Fragment Error Status Flag	0: No Bad Fragment Error 1: Bad Fragment Error occurred.	R/W
b8	FCES	Frame Count Error Status Flag	0: No Frame Count Error 1: Frame Count Error occurred.	R/W
b9	REOES	E-Frame Overflow Error Status Flag	0: No overflow error 1: Overflow error occurred during e-frame reception.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	CTLES0	Captured Timestamp Lost Error Status Flag 0	0: No Captured Timestamp Lost Error 1: Captured Timestamp Lost Error occurred.	R/W
b13	CTLES1	Captured Timestamp Lost Error Status Flag 1	0: No Captured Timestamp Lost Error 1: Captured Timestamp Lost Error occurred.	R/W
b19 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	PDES	PHY Data Error Status Flag	0: No PHY Data Error 1: PHY Data Error occurred.	R/W
b21	PNAES	PHY Nibble Alignment Error Status Flag	0: No PHY Nibble Alignment Error 1: PHY Nibble Alignment Error occurred.	R/W
b22	FCMCES	FCS/mCRC Error Status Flag	0: No FCS/mCRC Error 1: FCS/mCRC Error occurred.	R/W
b23	FFMES	Final Fragment Missing Error Status Flag	0: No Final Fragment Missing Error 1: Final Fragment Missing Error occurred.	R/W
b24	CFCES	C Fragment Count Error Status Flag	0: No C Fragment Count Error 1: C Fragment Count Error occurred.	R/W
b25	FRCES	Fragment Count Error Status Flag	0: No Fragment Count Error 1: Fragment Count Error occurred.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26	RPOOMS	Reception Partially out of Operation Mode Status Flag	0: No Reception Partially out of Operation Mode 1: Reception Partially out of Operation Mode occurred.	R/W
b27	FFS	Frame Filtered Status Flag	0: No Frame Filtered 1: Frame Filtered occurred.	R/W
b28	FUES	Undersize Error Status Flag	0: No Undersize Error 1: Undersize Error occurred.	R/W
b29	FOES	Oversize Error Status Flag	0: No Oversize Error 1:Oversize Error occurred.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This interrupt register is mapped to rmc_meis_int interrupt wire.

TSLS Flag (Transmission Stream Lost Status Flag)

[Setting condition]

- HW: When ETHA do not provide the transmitted data timely. In this case an invalid FCS is transmitted. (only for e-frames)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Note and Restriction]

- When using Cut through, this flag set condition may be due to the state ([COMA] CAEIS0.BPOPS) where all Buffer pointer are used. Please use the Watermark function so that Buffer Pool Out of Pointer Status does not occur. If get this error with Buffer Pool Out of Pointer Status using Cut through (it may get hung-up in the transmit state), initialize the state with Emergency reset flow [ESWM].

PRES Flag (Pause or PFC Frame Reception Error Status Flag)

[Setting conditions]

- HW: When Pause or PFC frame is received with lower than minimum pause/pfc frame length.
- Minimum PAUSE Length: 18 byte (MRGC.RCPT = 0) or 22 byte ((MRGC.RCPT = 1)
- Minimum PFC Length: 34 byte (MRGC.RCPT = 0) or 38 byte ((MRGC.RCPT = 1)
- HW: When PAUSE/PFC frame is received with fcs error in MRGC.RFCFE = 0.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

PFRROS Flag (Pause or PFC Frame Retransmit Retry Over Status Flag)

[Setting condition]

- HW: When the total number of Pause (pfc) transmit count reaches the configured upper limit and Pause time expires.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

FCDS Flag (False Carrier Detection Status Flag)

[Setting condition]

- HW: When PHY-LSI has detected a false carrier on the line in case of MII and GMII.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.

- SW: Writing 1 to this flag will clear it.

TCES Flag (Transmit CRC Error Status Flag)

[Setting condition]

- HW: When CRC error occurred

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

TBCIS Flag (Transmit Bad CRC Insertion Status Flag)

[Setting condition]

- HW: A bad CRC is inserted from the upper layer (e.g. ETHA).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

BFES Flag (Bad Fragment Error Status Flag)

[Setting condition]

- HW: C fragment has been detected without receiving S fragment.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Note]

- When C-fragment is received at very first time before any valid S-fragment is received or C-fragment is received after frame termination is occurred before any valid S-fragment is received, then RMAC will discard the fragment and this error status flag will not be set. For Error status flag to get set, RMAC should have received one valid S-fragment before error condition (bad fragment error) has occurred.

FCES Flag (Frame Count Error Status Flag)

[Setting condition]

- When S fragment has been received in the wrong order.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Note]

- When C-fragment is received at very first time before any valid S-fragment is received or C-fragment is received after frame termination is occurred before any valid S-fragment is received, then RMAC will discard the fragment and this error status flag will not be set. For Error status flag to get set, RMAC should have received one valid S-fragment before error condition (frame count error) has occurred.

REOES Flag (E-Frame Overflow Error Status Flag)

[Setting condition]

- HW: The acknowledge signal was not received successfully from an upper layer (e.g. ETHA) during e-frame reception.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.

- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: Pulse is generated per 1 frame although multiple de-assertion is generated.

[Notes]

- When frame reception is already ongoing and buffer overflow occurs i.e CABPPCM.RPC = 0, RMAC adds one byte padding to the reception error Frames to terminate the protocol internally.
- When new frame is received, during buffer overflow i.e CABPPCM.RPC = 0, the incoming frame will be rejected completely.

CTLES0 Flag (Captured Timestamp Lost Error Status Flag 0)

[Setting condition]

- HW: When captured timestamp is lost.
  This flag is fixed to 0.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- No recovery. Timestamp is lost.

CTLES1 Flag (Captured Timestamp Lost Error Status Flag 1)

[Setting condition]

- HW: When captured timestamp is lost.
  This bit is fixed to 0.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- No recovery. Timestamp is lost.

PDES Flag (PHY Data Error Status Flag)

[Setting condition]

- HW: When frame is received with error.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

PNAES Flag (PHY Nibble Alignment Error Status Flag)

[Setting condition]

- HW: When residual-bit is received.

[Clearing conditions]

- SW: Writing 1 to this flag will clear it.

FCMCES Flag (FCS/mCRC Error Status Flag)

[Setting condition]

- HW: When unexpected FCS/mCRC are received.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

FFMES Flag (Final Fragment Missing Error Status Flag)

[Setting condition]

- HW: When a S-fragment is received even though the previous fragment has been terminated by a mCRC.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

CFCES Flag (C Fragment Count Error Status Flag)

[Setting condition]

- HW: When a C fragment not corresponding to the current frame is received
  For example: S0-C0-C0-C1

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

FRCES Flag (Fragment Count Error Status Flag)

[Setting condition]

- HW: When a C-fragment is received with a bad fragment count field

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

RPOOMS Flag (Reception Partially out of Operation Mode Status Flag)

[Setting condition]

- HW: When a frame is being received while RMAC is moving out of operation mode (suspending).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Note]

- This error has nothing to do with partial rejection due to buffer overflow. E-Frames can only be fragmented/ partialized by buffer overflows and it is not affected by partial out of operation.

FFS Flag (Frame Filtered Status Flag)

[Setting condition]

- HW: When frame did not pass the RMAC filters

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

FUES Flag (Undersize Error Status Flag)

[Setting condition]

- HW: When too short frame is received. (when the received frame size is shorter than the setting in

MRFSCE.EMNS[15:0])

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

FOES Flag (Oversize Error Status Flag)

[Setting condition]

- HW: When too big frame is received. (when the received frame size is bigger than the setting in MRFSCE.EMXS[15:0])

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

Table 37.3 Transmission Error List

Error type	Event	Status Bit
Transmit bad CRC insertion status	A packet has been received for transmission by RMAC with an error and a wrong FCS has been inserted.	MEIS.TBCIS
Transmit CRC error status	A packet which included a presupposed valid FCS had a faulty FCS	MEIS.TCES
Pause frame Retransmit Retry Over Status	A Pause or PFC retransmit count reaches the configured upper limit(MTPFC.PFRLV[4:0]) and Pause time expires.	MEIS.PFRROS
Transmission Stream Lost Status	An e-frame data has not been provided on time to RMAC.	MEIS.TSLS

Table 37.4 Reception Error List

No	Error type	Event	Frame status bit	Status register
1	PHY data error	Received a data with error.	Bit 0	MEIS.PDES
2	PHY nibble alignment error	Received residual-bit.	Bit 1	MEIS.PNAES
3	FCS/mCRC error	Received unexpected FCS field value.	Bit 2	MEIS.FCMCES
4	Final fragment missing error	Occurs when a S-fragment is received even though the previous fragment has been terminated by a MCRC	Bit 3	MEIS.FFMES
5	C fragment count error	Occurs when a C fragment not corresponding to the current frame is received. For example: S0-C0-C0-C1	Bit 4	MEIS.CFCES
6	Fragment count error	Occurs when a C-fragment is received with a bad fragment count field	Bit 5	MEIS.FRCES
7	Reception partially out of operation mode	Occurs when a frame is being received while RMAC is moving out of operation mode	Bit 6	MEIS.RPOOMS
8	Frame filtered	The frame did not pass the RMAC filters	Bit 7	MEIS.FFS
9	Undersize error	The frame is too small	Bit 8	MEIS.FUES
10	Oversize error	The frame is too big	Bit 9	MEIS.FOES
11	Buffer overflow error	Occurs when the rx done signal has been de-asserted during frame reception	Bit 12	MEIS.REOES MEIS.RPOES
12	Frame count error status	Occurs when a wrong S-frame order is received. For example: S0-S1-S2-S1	—	MEIS.FCES
13	Bad fragment error status	Occurs when a C fragment has been detected without receiving S fragment	—	MEIS.BFES
14	False carrier detection status	Occurs when PHY-LSI has detected a false carrier on the line.	—	MEIS.FCDS

37.2.3.2 Error Interrupt Enable Register (MEIE)

Address(es): RMAC0.MEIE 874C B204h, RMAC0_NS.MEIE 074C B204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FOEE	FUEE	FFE	RPOOME	FRCEE	CFCEE	FFMEE	FCMCEE	PNAEE	PDEE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTLEE1	CTLEE0	—	RPOEE	REOEE	FCEE	BFEE	TBCIE	TCEE	FCDE	PFRROE	PREE	—	TSLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSLE	Transmission Stream Lost Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	PREE	Pause or PFC Frame Reception Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	PFRROE	Pause or PFC Frame Retransmit Retry Over Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	FCDE	False Carrier Detection Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	TCEE	Transmit CRC Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b6	TBCIE	Transmit Bad CRC Insertion Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b7	BFEE	Bad Fragment Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b8	FCEE	Frame Count Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b9	REOEE	E-Frame Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	RPOEE	P-Frame Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R
b12	CTLEE0	Captured Timestamp Lost Error Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b13	CTLEE1	Captured Timestamp Lost Error Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b19 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	PDEE	PHY Data Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	PNAEE	PHY Nibble Alignment Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	FCMCEE	FCS/mCRC Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	FFMEE	Final Fragment Missing Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	CFCEE	C Fragment Count Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	FRCEE	Fragment Count Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b26	RPOOME	Reception Partially out of Operation Mode Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27	FFE	Frame Filtered Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b28	FUEE	Undersize Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b29	FOEE	Oversize Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the MEID register.

TSLE Bit (Transmission Stream Lost Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.TSLD will clear this bit.

PREE Bit (Pause or PFC Frame Reception Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.PRED will clear this bit.

PFRROE Bit (Pause or PFC Frame Retransmit Retry Over Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.PFRROD will clear this bit.

FCDE Bit (False Carrier Detection Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FCDD will clear this bit.

TCEE Bit (Transmit CRC Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.TCED will clear this bit.

TBCIE Bit (Transmit Bad CRC Insertion Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.TBCID will clear this bit.

BFEE Bit (Bad Fragment Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.BFED will clear this bit.

FCEE Bit (Frame Count Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FCED will clear this bit.

REOEE Bit (E-Frame Overflow Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.REOED will clear this bit.

RPOEE Bit (P-Frame Overflow Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.RPOED will clear this bit.

CTLEE0 Bit (Captured Timestamp Lost Error Enable 0)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.CTLED0 will clear this bit.

CTLEE1 Bit (Captured Timestamp Lost Error Enable 1)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.CTLED1 will clear this bit.

PDEE Bit (PHY Data Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.PDED will clear this bit.

PNAEE Bit (PHY Nibble Alignment Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.PNAED will clear this bit.

FCMCEE Bit (FCS/mCRC Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FCMCED will clear this bit.

FFMEE Bit (Final Fragment Missing Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FFMED will clear this bit.

CFCEE Bit (C Fragment Count Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.CFCED will clear this bit.

FRCEE Bit (Fragment Count Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FRCED will clear this bit.

RPOOME Bit (Reception Partially out of Operation Mode Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.RPOOMD will clear this bit.

FFE Bit (Frame Filtered Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FFD will clear this bit.

FUEE Bit (Undersize Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to MEID.FUED will clear this bit.

FOEE Bit (Oversize Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- SW: Writing 1 to MEID.FOED will clear this bit.

37.2.3.3 Error Interrupt Disable Register (MEID)

Address(es): RMAC0.MEID 874C B208h, RMAC0_NS.MEID 074C B208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FOED	FUED	FFD	RPOOMD	FRCED	CFCED	FFMED	FCMCED	PNAED	PDED	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTLED1	CTLED0	—	RPOED	REOED	FCED	BFED	TBCID	TCED	FCDD	PFRROD	PRED	—	TSLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSLD	Transmission Stream Lost Disable	0: Has no effects. 1: Sets MEIE.TSLE bit to 0.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	PRED	Pause or PFC Frame Reception Error Disable	0: Has no effects. 1: Sets MEIE.PREE bit to 0.	R/W
b3	PFRROD	Pause or PFC Frame Frame Retransmit Retry Over Disable	0: Has no effects. 1: Sets MEIE.PFRROE bit to 0.	R/W
b4	FCDD	False Carrier Detection Disable	0: Has no effects. 1: Sets MEIE.FCDE bit to 0.	R/W
b5	TCED	Transmit CRC Error Disable	0: Has no effects. 1: Sets MEIE.TCEE bit to 0.	R/W
b6	TBCID	Transmit Bad CRC Insertion Disable	0: Has no effects. 1: Sets MEIE.TBCIE bit to 0.	R/W
b7	BFED	Bad Fragment Error Disable	0: Has no effects. 1: Sets MEIE.BFEE bit to 0.	R/W
b8	FCED	Frame Count Error Disable	0: Has no effects. 1: Sets MEIE.FCEE bit to 0.	R/W
b9	REOED	E-Frame Overflow Error Disable	0: Has no effects. 1: Sets MEIE.REOEE bit to 0.	R/W
b10	RPOED	P-Frame Overflow Error Disable	0: Has no effects. 1: Sets MEIE.RPOEE bit to 0.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R
b12	CTLED0	Captured Timestamp Lost Error Disable 0	0: Has no effects. 1: Sets MEIE.CTLEE0 bit to 0.	R/W
b13	CTLED1	Captured Timestamp Lost Error Disable 1	0: Has no effects. 1: Sets MEIE.CTLEE1 bit to 0.	R/W
b19 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	PDED	PHY Data Error Disable	0: Has no effects. 1: Sets MEIE.PDEE bit to 0.	R/W
b21	PNAED	PHY Nibble Alignment Error Disable	0: Has no effects. 1: Sets MEIE.PNAEE bit to 0.	R/W
b22	FCMCED	FCS/mCRC Error Disable	0: Has no effects. 1: Sets MEIE.FCMCEE bit to 0.	R/W
b23	FFMED	Final Fragment Missing Error Disable	0: Has no effects. 1: Sets MEIE.FFMEE bit to 0.	R/W
b24	CFCED	C Fragment Count Error Disable	0: Has no effects. 1: Sets MEIE.CFCEE bit to 0.	R/W
b25	FRCED	Fragment Count Error Disable	0: Has no effects. 1: Sets MEIE.FCEE bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26	RPOOMD	Reception Partially out of Operation Mode Disable	0: Has no effects. 1: Sets MEIE.RPOOME bit to 0.	R/W
b27	FFD	Frame Filtered Disable	0: Has no effects. 1: Sets MEIE.FFE bit to 0.	R/W
b28	FUED	Undersize Error Disable	0: Has no effects. 1: Sets MEIE.FUEE bit to 0.	R/W
b29	FOED	Oversize Error Disable	0: Has no effects. 1: Sets MEIE.FOEE bit to 0.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

37.2.3.4 Monitoring Interrupt Status Register 0 (MMIS0)

Address(es): RMAC0.MMIS0 874C B210h, RMAC0_NS.MMIS0 074C B210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	VFRS	—	LVSS	PIDS	PLSCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PLSCS	PHY Link Signal Change Status Flag	0: No PHY Link Signal Change 1: PHY Link Signal Change occurred.	R/W
b1	PIDS	PHY Interrupt Detection Status Flag	0: No PHY Interrupt Detection 1: PHY Interrupt Detection occurred.	R/W
b2	LVSS	Link Verification Succeed Status Flag	0: No Link Verification Succeed 1: Link Verification Succeed occurred.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	VFRS	Verify Frame Reception Status Flag	0: No Verify Frame Reception 1: Verify Frame Reception occurred.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This interrupt register is mapped to rmc_mmis0_int interrupt wire.

PLSCS Flag (PHY Link Signal Change Status Flag)

[Setting condition]

- HW: When the value of link status pin (rmc_phy_link port) is changed.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has not effect.
- SW: To check the current link state, refer to the link status pin state bit in the PHY link status register (MPIC.PLSPP).

PIDS Flag (PHY Interrupt Detection Status Flag)

[Setting condition]

- HW: PHY interrupt pin (rmc_phy_int port) is active level.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has not effect.
- HW: This flag could be asserted unintentionally before configuring PHY interrupt polarity (MPIC.PIP) into the right value. In the phase, this flag should be ignored.

LVSS Flag (Link Verification Succeed Status Flag)

[Setting condition]

- HW: When Link Verification process succeed.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has not effect.
- It’s no effect when suspending, e.g. an upper layer (e.g. ETHA) is not in OPERATION state.

VFRS Flag (Verify Frame Reception Status Flag)

[Setting condition]

- HW: When valid verify frame has been received.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has not effect.
- It’s no effect when suspending, e.g. an upper layer (e.g. ETHA) is not in OPERATION state.

37.2.3.5 Monitoring Interrupt Enable Register 0 (MMIE0)

Address(es): RMAC0.MMIE0 874C B214h, RMAC0_NS.MMIE0 074C B214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	VFRE	LVFE	LVSE	PIDE	PLSCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PLSCE	PHY Link Signal Change Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	PIDE	PHY Interrupt Detection Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	LVSE	Link Verification Succeed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	LVFE	Link Verification Failed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	VFRE	Verify Frame Reception Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the MMID0 register.

37.2.3.6 Monitoring Interrupt Disable Register 0 (MMID0)

Address(es): RMAC0.MMID0 874C B218h, RMAC0_NS.MMID0 074C B218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	VFRD	LVFD	LVSD	PIDD	PLSCD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PLSCD	PHY Link Signal Change Disable	0: Has no effects. 1: Sets MMIE0.PLSCE bit to 0.	W
b1	PIDD	PHY Interrupt Detection Disable	0: Has no effects. 1: Sets MMIE0.PIDE bit to 0.	W
b2	LVSD	Link Verification Succeed Disable	0: Has no effects. 1: Sets MMIE0.LVSE bit to 0.	W
b3	LVFD	Link Verification Failed Disable	0: Has no effects. 1: Sets MMIE0.LVFE bit to 0.	W
b4	VFRD	Verify Frame Reception Disable	0: Has no effects. 1: Sets MMIE0.VFRE bit to 0.	W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

37.2.3.7 Monitoring Interrupt Status Register 1 (MMIS1)

Address(es): RMAC0.MMIS1 874C B220h, RMAC0_NS.MMIS1 074C B220h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PPRACS	PAACS	PWACS	PRACS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRACS	PHY Read Access Completed Status Flag	0: No PHY Read Access Completed 1: PHY Read Access Completed occurred.	R/W
b1	PWACS	PHY Write Access Completed Status Flag	0: No PHY Write Access Completed 1: PHY Write Access Completed occurred.	R/W
b2	PAACS	PHY Address Access Completed Status Flag	0: No PHY Address Access Completed 1: PHY Address Access Completed occurred.	R/W
b3	PPRACS	PHY Post-Read Access Completed Status Flag	0: No PHY Post-Read Access Completed 1: PHY Post-Read Access Completed occurred.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This interrupt register is mapped to rmc_mdio_int interrupt wire.

PRACS Flag (PHY Read Access Completed Status Flag)

[Setting condition]

- HW: When PHY read access has completed. Refer to section 37.3.3, Station Management.

[Clearing conditions]

- HW: Being in RESET mode will clear this bit.
- SW: Writing 1 to this bit will clear it.

[Restriction]

- Writing 0 to this bit has no effect.

PWACS Flag (PHY Write Access Completed Status Flag)

[Setting condition]

- HW: When PHY write access has completed. Refer to section 37.3.3, Station Management.

[Clearing conditions]

- HW: Being in RESET mode will clear this bit.
- SW: Writing 1 to this bit will clear it.

[Restriction]

- Writing 0 to this bit has no effect.

PAACS Flag (PHY Address Access Completed Status Flag)

[Setting condition]

- HW: When PHY address access has completed. Refer to section 37.3.3, Station Management.

[Clearing conditions]

- HW: Being in RESET mode will clear this bit.
- SW: Writing 1 to this bit will clear it.

[Restriction]

- Writing 0 to this bit has no effect.

PPRACS Flag (PHY Post-Read Access Completed Status Flag)

[Setting condition]

- HW: When PHY post-read access has completed. Refer to section 37.3.3, Station Management.

[Clearing conditions]

- HW: Being in RESET mode will clear this bit.
- SW: Writing 1 to this bit will clear it.

37.2.3.8 Monitoring Interrupt Enable Register 1 (MMIE1)

Address(es): RMAC0.MMIE1 874C B224h, RMAC0_NS.MMIE1 074C B224h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PPRACE	PAACE	PWACE	PRACE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRACE	PHY Read Access Completed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	PWACE	PHY Write Access Completed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	PAACE	PHY Address Access Completed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	PPRACE	PHY Post-Read Access Completed Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the MMID1 register.

PRACE Bit (PHY Read Access Completed Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- HW: Writing 1 to the MMID1.PRACD bit will clear this bit.

[Restriction]

- SW: Writing 0 to this bit has no effect.

PWACE Bit (PHY Write Access Completed Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- HW: Writing 1 to the MMID1.PWACD bit will clear this bit.

[Restriction]

- SW: Writing 0 to this bit has no effect.

PAACE Bit (PHY Address Access Completed Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- HW: Writing 1 to the MMID1.PAACD bit will clear this bit.

[Restriction]

- SW: Writing 0 to this bit has no effect.

PPRACE Bit (PHY Post-Read Access Completed Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- HW: Writing 1 to the MMID1.PPRACD bit will clear this bit.

[Restriction]

- SW: Writing 0 to this bit has no effect.

37.2.3.9 Monitoring Interrupt Disable Register 1 (MMID1)

Address(es): RMAC0.MMID1 874C B228h, RMAC0_NS.MMID1 074C B228h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PPRACD	PAACD	PWACD	PRACD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRACD	PHY Read Access Completed Disable	0: Has no effects. 1: Sets MMIE1.PRACE bit to 0.	W
b1	PWACD	PHY Write Access Completed Disable	0: Has no effects. 1: Sets MMIE1.PWACE bit to 0.	W
b2	PAACD	PHY Address Access Completed Disable	0: Has no effects. 1: Sets MMIE1.PAACE bit to 0.	W
b3	PPRACD	PHY Post-Read Access Completed Disable	0: Has no effects. 1: Sets MMIE1.PPRACE bit to 0.	W
b31 to b4	—	Reserved	The write value should be 0.	W

PRACD Bit (PHY Read Access Completed Disable)

[Restriction]

- Writing 0 to this bit has no effect.

PWACD Bit (PHY Write Access Completed Disable)

[Restriction]

- Writing 0 to this bit has no effect.

PAACD Bit (PHY Address Access Completed Disable)

[Restriction]

- Writing 0 to this bit has no effect.

PPRACD Bit (PHY Post-Read Access Completed Disable)

[Restriction]

- Writing 0 to this bit has no effect.

37.2.3.10 Monitoring Interrupt Status Register 2 (MMIS2)

Address(es): RMAC0.MMIS2 874C B230h, RMAC0_NS.MMIS2 074C B230h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LPIDIS	LPIAIS	MPDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPDIS	Magic Packet Detection Interrupt Status Flag	0: No Magic Packet Detection Interrupt 1: Magic Packet Detection Interrupt occurred.	R/W
b1	LPIAIS	LPI Assertion Interrupt Status Flag	0: No LPI Assertion Interrupt 1: LPI Assertion Interrupt occurred.	R/W
b2	LPIDIS	LPI De-Assertion Interrupt Status Flag	0: No LPI De-Assertion Interrupt 1: LPI De-Assertion Interrupt occurred.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This interrupt register is mapped to rmc_mp_int, rmc_lpi_start_int, and rmc_lpi_stop_int interrupt wire.
Note: This register's value goes through the synchronizer between clk and clk_phy_rx.
The result of write access is reflected to each SFR bit after "5clk + 4clk_phy_rx" in worst case in RMAC.
Do not make write access continuously. Please wait following time for re-write access.

MPDIS Flag (Magic Packet Detection Interrupt Status Flag)

[Setting condition]

- HW: When a Magic Packet has been detected.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has no effect.
- HW: CRC of magic packet is not checked.
- HW: It's no effect when suspending

LPIAIS Flag (LPI Assertion Interrupt Status Flag)

[Setting condition]

- HW: When LPI request is detected. This flag is continuously asserted when LPI request continues.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has no effect.
- HW: It's no effect when suspending.

LPIDIS Flag (LPI De-Assertion Interrupt Status Flag)

[Setting condition]

- HW: When LPI request de-assertion.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restrictions]

- SW: Writing 0 to this flag has no effect.

HW: It’s no effect when suspending.

37.2.3.11 Monitoring Interrupt Enable Register 2 (MMIE2)

Address(es): RMAC0.MMIE2 874C B234h, RMAC0_NS.MMIE2 074C B234h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LPIDIE	LPIAIE	MPDIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPDIE	Magic Packet Detection Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	LPIAIE	LPI Assertion Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	LPIDIE	LPI De-Assertion Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register's value goes through the synchronizer between clk and clk_phy_rx.
The result of write access is reflected to each SFR bit after "5clk + 4clk_phy_rx" in worst case in RMAC.
Do not make write access continuously. Please wait following time for re-write access.

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the MMID2 register.

37.2.3.12 Monitoring Interrupt Disable Register 2 (MMID2)

Address(es): RMAC0.MMID2 874C B238h, RMAC0_NS.MMID2 074C B238h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LPIDID	LPIAID	MPDID
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPDID	Magic Packet Detection Interrupt Disable	0: Has no effects. 1: Sets MMIE2.MPDIE bit to 0.	W
b1	LPIAID	LPI Assertion Interrupt Disable	0: Has no effects. 1: Sets MMIE2.LPIAIE bit to 0.	W
b2	LPIDID	LPI De-Assertion Interrupt Disable	0: Has no effects. 1: Sets MMIE2.LPIDIE bit to 0.	W
b31 to b3	—	Reserved	The write value should be 0.	W

Note: This register's value goes through the synchronizer between clk and clk_phy_rx.
The result of write access is reflected to each SFR bit after "5clk + 4clk_phy_rx" in worst case in RMAC.
Do not make write access continuously. Please wait following time for re-write access.

37.3 Operation

37.3.1 Flow Control

This IP supports the following two types of Flow Control functions.

1. Flow Control specified in IEEE 802.3 for a full-duplex operation using a PAUSE frame.
2. Priority Based Flow Control specified in IEEE 802.1Qbb for a full-duplex operation using a PFC frame.

37.3.1.1 Flow Control by PAUSE Frame

This Flow Control is only for a full-duplex operation and this consists of Transmission and Reception of a PAUSE frame.

Transmission of PAUSE frame

(1) During MTFFC.FCM = 0 and MTPFC.PT[15:0] ≠ 0, the transmission function of a PAUSE frame is enabled. This IP transmits a PAUSE frame as described in Figure 37.2 by asserting a transmission request. The value of MTPFC.PT[15:0] is used for “PAUSE_TIME” field. And MRMAC0 and MRMAC1 are used for “SA” field.



Figure 37.2 PAUSE Frame Format

- Auto PAUSE Frame
  In MTPFC.PFM = 0, this IP uses a hardware pause transmission requests a transmit request. After transmitting the PAUSE Frame, MAPFTCT is incremented.
- Manual PAUSE Frame
  In MTPFC.PFM = 1, this IP uses MTPFC2.MPFR as a transmit request. After transmitting the PAUSE Frame, MMPFTCT is incremented.
  When the transmit request of the PAUSE frame is asserted while the IP is transmitting the other frame, the PAUSE frame is transmitted after the transmitting frame.

(2) While the transmit request is asserted, a PAUSE frame is continuously re-transmitted at a fixed interval of time configured by MTPFC.PFRT[7:0]. If the number of re-transmission reaches the value of MTPFC.PFRLV[4:0], this IP sets MEIS.PFRROS to 1 for an interrupt. Even if the interrupt is detected, this IP continues re-transmission of a PAUSE frame while the transmit request is asserted.

(3) After de-asserting the transmission request, this IP stops a transmission of PAUSE frame. At that time, when MTPFC2.PFTTZ = 1 and an internal counter to count a pausing-period exceeds the value of MTPFC.PFRT[7:0] (*), this IP transmits one more a PAUSE frame with “PAUSE_TIME” field as 0.

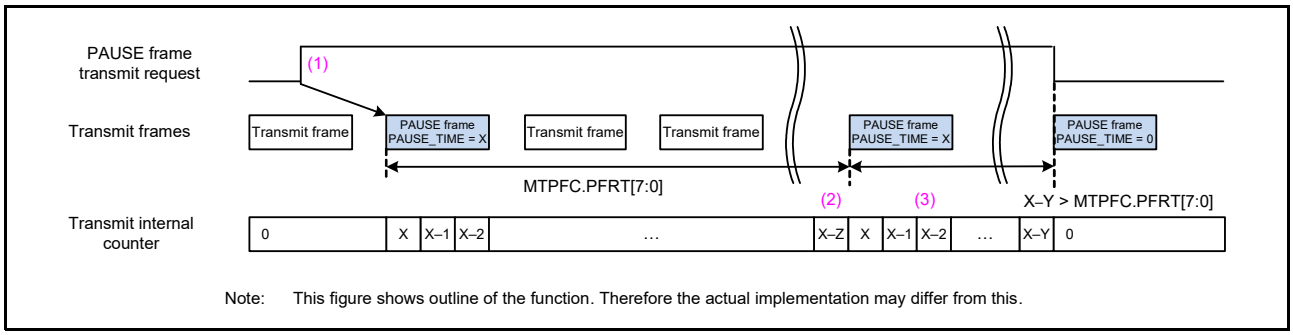


Figure 37.3 Transmission of PAUSE Frame

Reception of PAUSE frame

1. During MRGC.PFRC = 1, this IP receives a PAUSE frame and it starts internally counting a pausing-period specified by “PAUSE_TIME” field of a received PAUSE frame. While counting the pausing-period, notification of pause frame reception to an upper layer (e.g. ETHA) and MRPFM.PTCA is asserted. And any transmissions except for a Verify_packet, a Respond_packet and a PAUSE frame are inhibited. When this IP receives a PAUSE frame while transmitting any frames, transmissions are inhibited after the transmission completion.
2. When a PAUSE frame with “PAUSE_TIME ≠ 0” is received again before expiring the internal pausing-period, the internal counter is updated to that value.
3. When the internal pausing-period is expired, notification of pause frame reception to an upper layer (e.g. ETHA) and MRPFM.PTCA is de-asserted.
4. When MRGC.PFRTZ = 1 and PAUSE frame with “PAUSE_TIME = 0” is received before expiring the internal pause-period, the internal counter is cleared. Then notification of pause frame reception to an upper layer (e.g. ETHA) and MRPFM.PTCA is de-asserted.

In any case, MPFRCT is incremented after receiving a PAUSE frame in MRGC.PFRC = 1. And the received PAUSE frames are discarded internally.

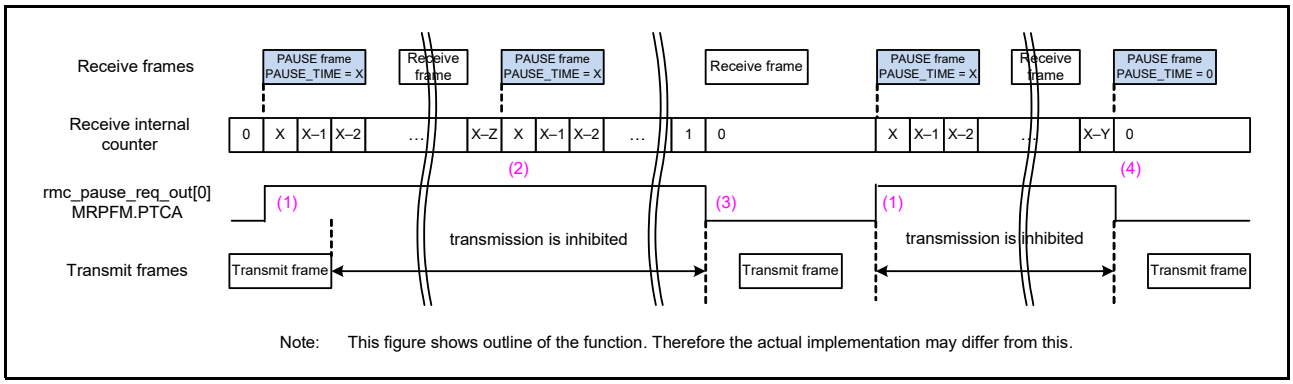


Figure 37.4 Reception of PAUSE Frame

37.3.1.2 Flow Control by PFC Frame

This Flow Control is for a full-duplex operation and this consists of Transmission and Reception of a PFC frame.

Transmission of PFC frame

InMTFFC.FCM = 1 and MTPFC.PT[15:0] ≠ 0, the transmission function of a PFC frame is enabled.
This IP treats several priorities as one priority-group to transmission of the PFC frame. The number of priority-group is configured by hardware parameter 2 and all descriptions in this section are written as 2 = 2. Each priority-group is configured by MTPFC30 and MTPFC31.

This IP transmits a PFC frame as described in Figure 37.5 for each priority-group by asserting a transmission request. The value of MTPFC.PT[15:0] is used for all “PAUSE_TIME” fields. “Priority_Enable” field value is different for each priority-group and the value of MTPFC30.PFCPG[7:0] and MTPFC31.PFCPG[7:0] are used for these. And MRMAC0 and MRMAC1 are used for “SA” field. When MTPFC30.PFCPG[7:0] (or MTPFC31.PFCPG[7:0]) is set to all 0,this IP does not transmit a PFC frame for the priority-group0 (or priority-group1) even when a transmission request is asserted.

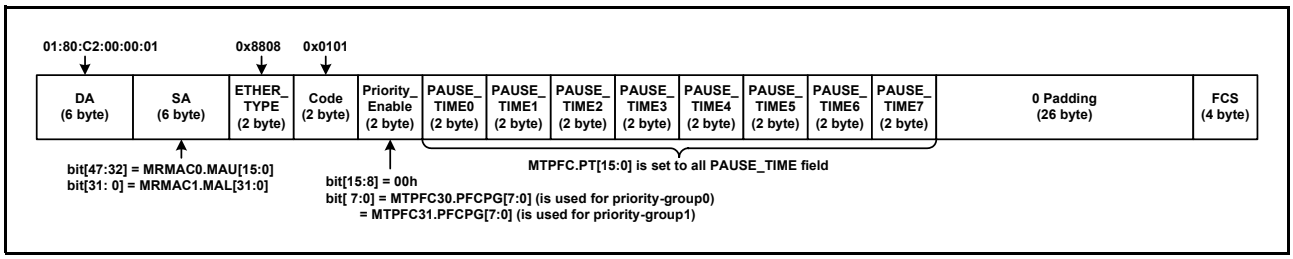


Figure 37.5 PFC Frame Format

- Auto PFC Frame
  InMTPFC.PFM = 0, this IP uses a hardware pause transmission request as a transmit request. The bit[0] is for priority-group0 and the bit[1] is for priority-group1. MAPCFTCT0 and MAPCFTCT1 are incremented respectively after transmitting a PFC Frame of each priority-group.
- Manual PFC Frame
  InMTPFC.PFM = 1, this IP uses MTPFC2.MPFCFR1, MPFCFR0 as a transmit request. The bit[0] is for priority-group0 and the bit[1] is for priority-group1. MMPCFTCT0 and MMPCFTCT1 are incremented respectively after transmitting a PFC Frame of each priority-group.

If several transmit requests are asserted at same time, one PFC frame which has merged priority_enable information is transmitted.
If several transmit requests are asserted while transmitting other frames, an internal counter is cleared and one PFC frame which has merged priority_enable information is transferred.

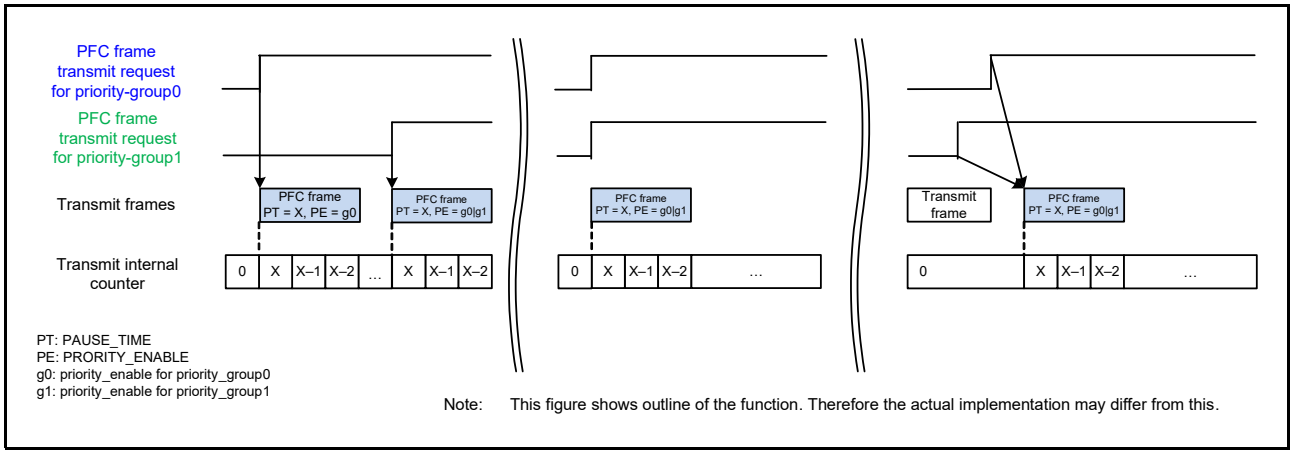


Figure 37.6 Transmitting PFC Frame (part1)

While the transmit request is asserted, a PFC frame corresponding to that request is continuously re-transmitted at a fixed interval configured by MTPFC.PFRT[7:0]. When the number of re-transmissions for one of priority-groups reaches the value of MTPFC.PFRLV[4:0], this IP sets MEIS.PFRROS to 1 for an interrupt. Even when the interrupt is detected, this IP continues re-transmission of a PFC frame while the transmit request is asserted.

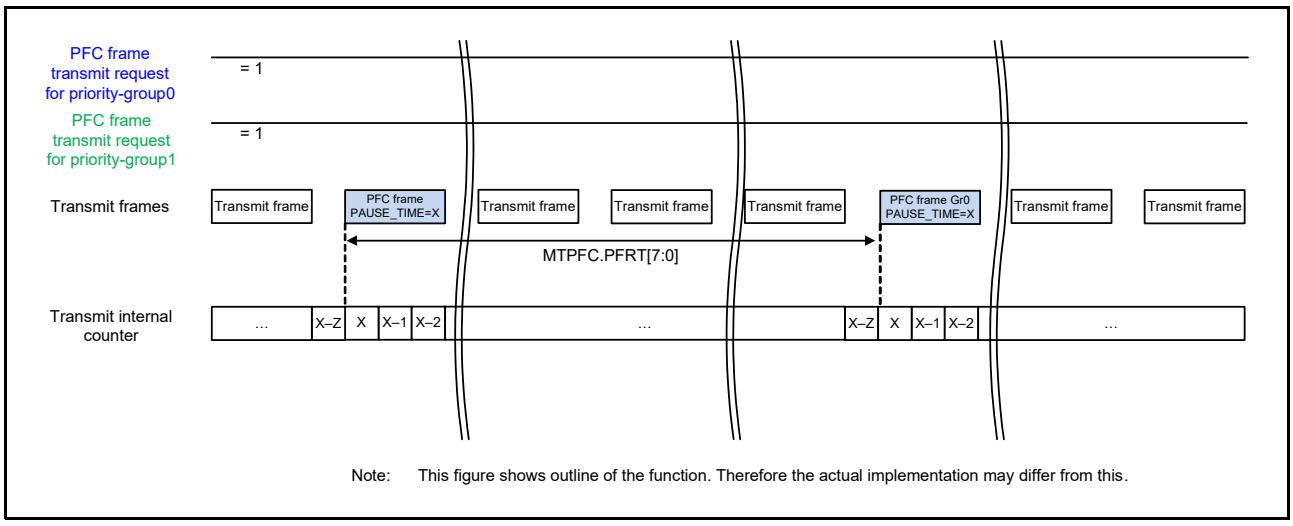


Figure 37.7 Transmitting PFC Frame (part2)

After de-asserting the transmission request, this IP stops the transmission of the PFC frame corresponding to that request. At that time, if the corresponding bit of MTPFC2.PFCTTZ1, PFCTTZ0 for de-asserted priority-group request is 1 and an internal counter to count the pausing-period exceeds MTPFC.PFRT[7:0], this IP transmits one more PFC frame with all “PAUSE_TIME” fields as 0 for that priority-group.

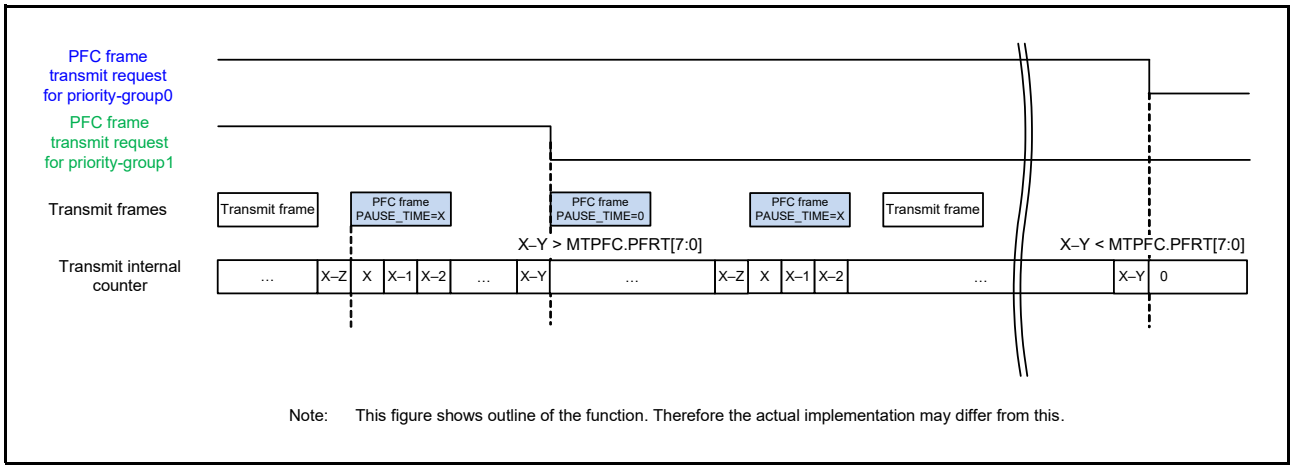


Figure 37.8 Transmitting PFC Frame (part3)

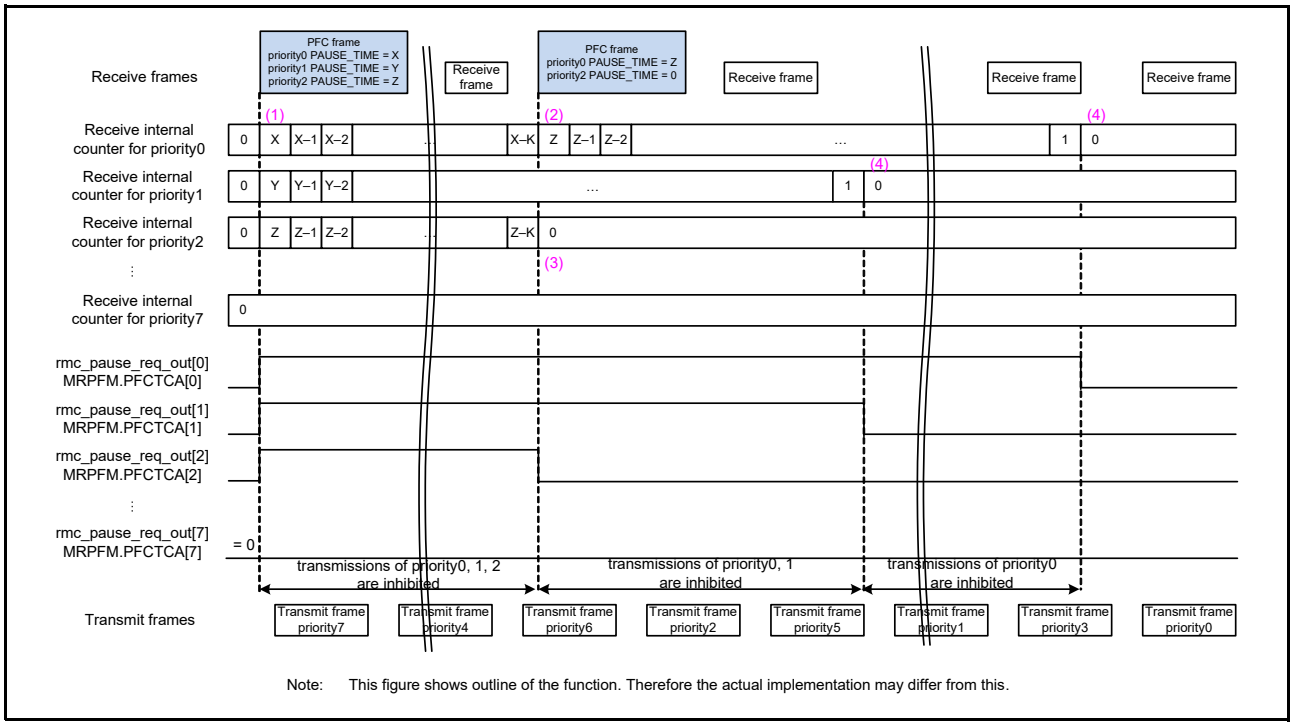
Reception of PFC frame

This IP treats several priorities specified in received PFC frames. The number of supported priorities is configured by hardware parameter 8 and all descriptions in this section are written as 8 = 8.

(1) When this IP receives a PFC frame with a priority-enable field which is corresponding to apriority configured by MRGC.PFCRC[7:0], it starts internally counting a pausing-period for each priority respectively specified by each “PAUSE_TIME” fields of a received PFC frame. During counting the pausing-period, notification of PFC frame reception to an upper layer (e.g. ETHA) and MRPFM.PFCTCA[7:0] corresponding to each priority are asserted. The bit number is corresponding to priority number.

(2) When PFC frame with “PAUSE_TIME≠0” with corresponding to valid “priority_enable” field is received again before expiring the internal pausing-period, the internal counter for that priority is updated to that value.
(3) When MRGC.PFRTZ = 1 and PFC frame with “PAUSE_TIME = 0” with corresponding to valid “priority_enable” field is received before expiring the internal pause-period, the internal counter for that priority is cleared and notification of pause frame reception to an upper layer (e.g. ETHA) and MRPFM.PFCTCA[7:0] for that priority is de-asserted.
(4) When the internal pausing-period is expired, notification of pause frame reception to an upper layer (e.g. ETHA) and MRPFM.PFCTCA[7:0] for that priority are de-asserted.

In any case, MPCFRCT0 to MPCFRCT7 are incremented respectively after receiving a PFC frame for enabled priority by MRGC.PFCRC[7:0]. And all received PFC frames are discarded internally.



Note: This figure shows outline of the function. Therefore the actual implementation may differ from this.

Figure 37.9 Reception of PAUSE Frame

37.3.1.3 Flow Control by JAM

This flow control is only for a half-duplex operation and this consists of transmission of a JAM data.

Transmission of JAM data

(1) During MTFCJC.JE = 1, the transmission function of a JAM data is enabled.
This IP transmits a JAM data as described in Figure 37.10 with preamble by asserting a transmission request.

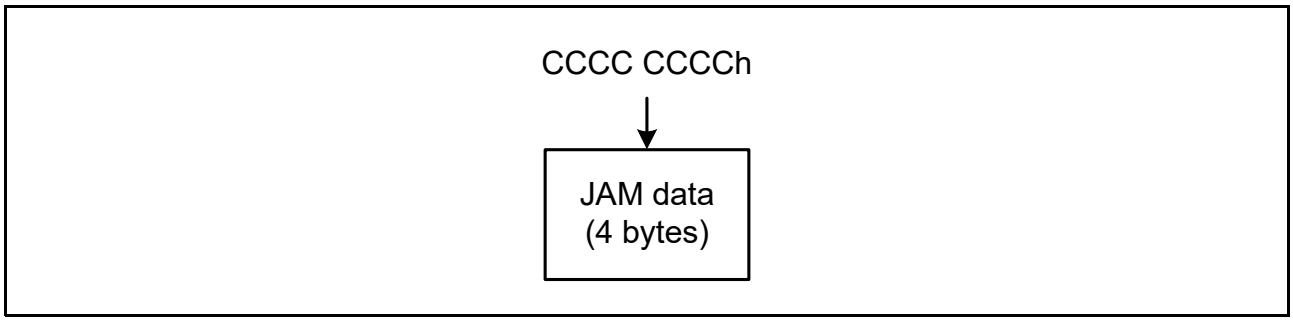


Figure 37.10 JAM Data

- Auto JAM Data
  In MTFCJC.JM = 0, this IP uses a hardware JAM data transmission request as a transmit request.
- Manual JAM Data
  In MTFCJC.JM = 1, this IP uses MTFCJC.MJOE as a transmit request.

(1) When the transmit request of the JAM data is asserted while the IP is transmitting the other frame, the JAM data is transmitted after the transmitting frame.
(2) While the transmit request is asserted, a JAM data is continuously re-transmitted at a fixed interval of time (IPG). After that, while this bit is asserted, the transmission of JAM data is prioritized and normal frames are not transmitted.

37.3.2 Timestamping

This IP supports time synchronization for IEEE 1588. It supports time synchronization accuracy in the sub-microsecond range.

37.3.2.1 Timestamp Capture

This IP can capture timestamps on both transmit and receive side. gPTP timer value includes two 62bit timers, bit[61:0] is a Timer0, bit[123:62] is a Timer1.

[Timestamp capture on transmit side]

“Timestamp capture” and “Timer number” fields in Table 36.4 control this function for both e-frame and p-frame. These are configured by a transmit descriptor. If “Timestamp capture” = 1, the timer specified by “Timer number” field is captured.

The captured timestamp is sent to an upper module(e.g. ETHA) with “Timestamp unique number” in Table 36.4 via ETHA transmit timestamp capture interface.

Note: On the transmit side, the timestamp capture function by this IP and PHY can be enabled at same time.

[Timestamp capture on receive side]

MTRC.TRDDE, MTRC.TRDDP, and MTRC.TRHFME1, TRHFME0 control this function.
The configuration of the timestamp capture is following.

- MTRC.TRDDE controls this function for e-frames.
  MTRC.TRDDE = 0
  the default timer configured by MTRC.DTN is captured for a received e-frame.

- MTRC.TRDDP controls this function for p-frames.
  MTRC.TRDDP = 0
  the default timer configured by MTRC.DTN is captured for a received p-frame.

- MTRC.TRHFME1, TRHFME0 controls this function for e-frames using the HW PTP message filtering.
  The TRHFME0 is for Timer0 and TRHFME1 is for Timer1.
  MTRC.TRHFME0 = 1 and the received e-frame matched the HW PTP messaged filter for Timer0
  The Timer0 is captured for that received e-frames.

  MTRC.TRHFME1 = 1 and MTRC.TRHFME0 = 0 and the received e-frame matched the HW PTP messaged filter for Timer1
  The Timer1 is captured for that received e-frames.

  MTRC.TRHFME1 = 1 and MTRC.TRHFME0 = 1 and the received e-frame matches the HW PTP messaged filter for Timer0 and 1
  The Timer0 is captured for that received e-frames.

A captured timestamp is outputted from this IP with the received frame.

.

37.3.3 Station Management

The Station Management controls the access to Management registers in PHY.
rmc_mdi, rmc_mdo, rmc_mde, rmc_mdc, and rmc_mdio_int are used for this function.

37.3.3.1 Management Register Access

This IP has two types of access methods, one is specified by clause 22 the other is specified by clause 45 in IEEE802.3.

Access Method specified in clause 22

To write or read of Management register of PHY which supports this access method using management frame format, execute the following steps.

[Write access]

Set MPSM as following.

PRD[15:0] = “Write Data”
POP[1:0] = 01b
PRA[4:0] = “Register Address”
PDA[4:0] = “PHY Address”
MFF = 0
PSME = 1

After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PWACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PWACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear the interrupt.

[Read access]

Set MPSM as following.

PRD[15:0] = 0000h
POP[1:0] = 10b
PRA[4:0] = “Register Address”
PDA[4:0] = “PHY Address”
MFF = 0
PSME = 1

After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PRACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PRACE is set to 1, the interrupt appears on rmc_mdio_int.
After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it.
Set MMIS1.PRACS to 0 to clear the interrupt.

Note: When MPSM.PSME is set to 1, this IP outputs preamble code of fixed length specified in standard on rmc_mdo before write or read access. IEEE802.3 clause 22 allows skipping preamble code. When the preamble code is not needed and PHY also supports to skip the preamble code, this IP does not output the preamble code by setting MPIC.PSMDP to 1.

Access Method specified in clause 45

To write or read Management register of PHY which supports this access method using extended management frame format, execute the following steps.

[Write access]

(1) Set MPSM as following. (“Register Address” is set to a PHY)
PRD[15:0] = “Register Address”
POP[1:0] = 00b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear interrupt.

(2) Set MPSM as following. (“Write Data” is set to the “Register Address” which was configured by step 1)
PRD[15:0] = “Write Data”
POP[1:0] = 01b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PWACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, if MMIE1.PWACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear interrupt.

[Read access]

(1) Set MPSM as following. (“Register Address” is set to a PHY)
PRD[15:0] = “Register Address”
POP[1:0] = 00b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear interrupt.

(2) Set MPSM as following. (Read access of the Register Address which was pointed by previous step)
PRD[15:0] = 0000h
POP[1:0] = 11b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PRACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.

At this time, when MMIE1.PRACE is set to 1, the interrupt appears on rmc_mdio_int.
After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it.
Set MMIS1.PRACS to 0 to clear interrupt.

[Post read incremental access]

Read access needs setting “Register Address” every time. Only when continuous Register Address fields are read continuously, using this access (POP[1:0] = 10) is efficient.

(1) Set MPSM as following. (“Register Address” is set to a PHY)
PRD[15:0] = “Register Address”
POP[1:0] = 00b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1

After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear the interrupt.

(2) Set MPSM as following. (Read access of the Register Address which was pointed by previous step)
PRD[15:0] = 0000h
POP[1:0] = 10b
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1

After setting MPSM, PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PPRACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PPRACE is set to 1, the interrupt appears on rmc_mdio_int.
After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it. And “Register Address” set by step 1) is automatically changed in the PHY to point a next register address field. So do not need to re-configuration of “Register Address” for next read access.
Set MMIS1.PPRACS to 0 to clear the interrupt.

(3) Repeat step (2) for the read access from the Next Register Address field.

Note: When MPSM.PSME set to 1, this IP outputs preamble code of fixed length specified in standard on rmc_mdo before write or read access. IEEE802.3 clause 45 does not allow skipping the preamble code. But skipping preamble code is also supported in this access method by setting MPIC.PSMDP to 1. To use this function, check a PHY specification.

37.3.3.2 Management Data Clock (MDC) Setting

To access Management registers in PHY, this IP outputs MDC. This is generated by dividing CHI clock (clk) in this IP and outputted from rmc_mdc. The division ratio can be set by the MPIC.PSMCS[6:0].
The following table shows the programmed MDC frequency depending on MPIC.PSMCS[6:0]. MDC frequency is calculated by clk[MHz] / ((MPIC.PSMCS[6:0] +1)*2).

Table 37.5 Programming Example for MPIC.PSMCS[6:0]

clk frequency	MPIC.PSMCS[6:0]	MDC frequency
100 MHz	13h	2.5 MHz
200 MHz	27h	2.5 MHz

Note: IEEE 802.3 specifies as the clock should be slower than 2.5 MHz. But some PHY could operate with a higher frequency. Refer to the PHY specification for detail.

37.3.3.3 Management Data Timing Adjustment

This IP can adjust the drive timing of MDO (rmc_mdo and rmc_mde) and the capture timing of MDI (rmc_mdi). It may be used for the propagation delay on the PCB.
The drive timing could be adjusted by PHY Station Management Hold time adjustment (MPIC.PSMHT[2:0]). The capture timing could be adjusted by PHY Station Management Capture time adjustment (MPIC.PSMCT[2:0]). The figure below is an example shifted by 2clk cycles.

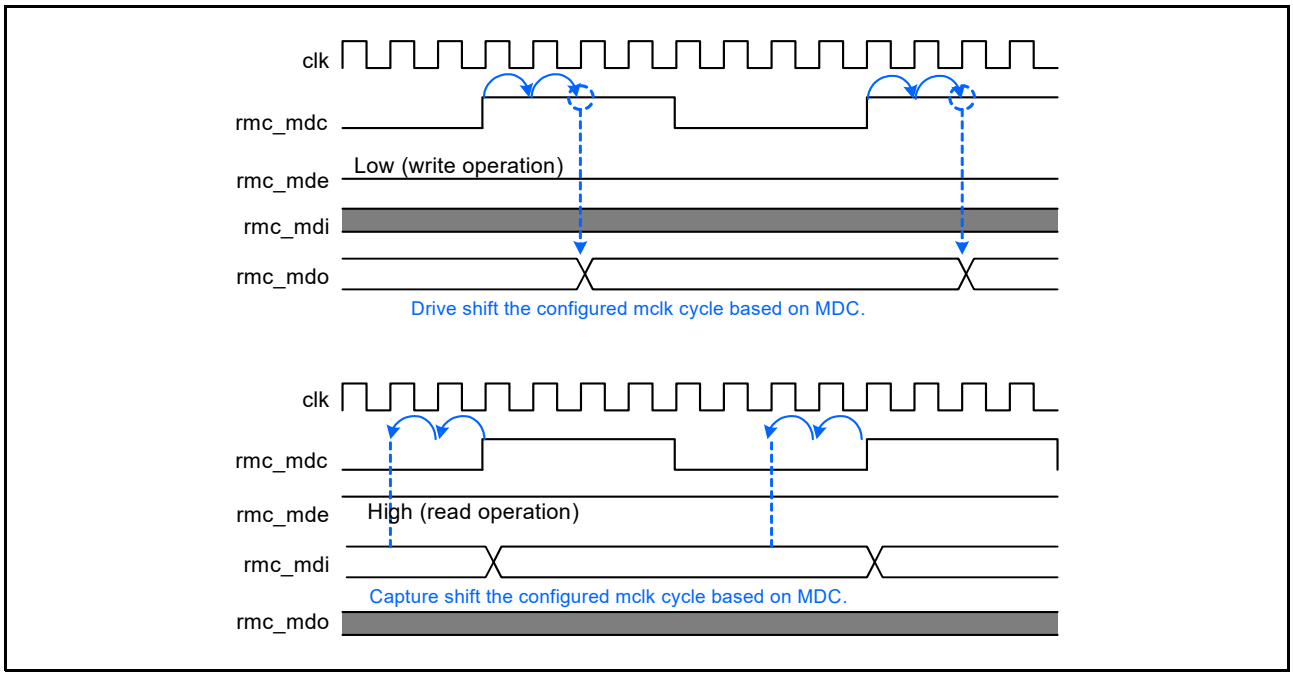


Figure 37.11 Management Data Timing Adjustment

37.3.4 Energy Efficient Ethernet

This IP supports Energy Efficient Ethernet (EEE) which is standardized inIEEE 802.3. Transmit-LPI mode and receive-LPI mode can be activated independently.

37.3.4.1 LPI Control

This IP can request LPI mode for transmit port to the PHY when the LPI request bit (MEEEC.LPITR) is asserted. LPI mode is released when the bit is negated. The operation flow is showed in section 37.3.7.4, LPI Transition for Transmit Port in detail.

And LPI mode for receive port can be recognized by the reception LPI interrupt status bit (MMIS2.LPIAIS). When the bit (MMIS2.LPIDIS) is asserted, it means RMAC detected de-assertion of LPI.
The table below is LPI specification on each xMII interface. The specification for MII or GMII is specified.

Table 37.6 Receive LPI Specification on Each xMII Interface

PHY IF	RX_DV	RX_ER	RXD
MII	rmc_rx_valid_gmii = 0	rmc_rx_err_gmii = 1	rmc_rx_data_gmii[3:0] = 0001
GMII	rmc_rx_valid_gmii = 0	rmc_rx_err_gmii = 1	rmc_rx_data_gmii[7:0] = 00000001

Table 37.7 Transmit LPI Specification on Each xMII Interface

PHY IF	TX_EN	TX_ER	TXD
MII	rmc_tx_valid_gmii = 0	rmc_tx_err_gmii = 1	rmc_tx_data_gmii [3:0] = 0001
GMII	rmc_tx_valid_gmii = 0	rmc_tx_err_gmii = 1	rmc_tx_data_gmii [7:0] = 00000001

37.3.5 MAC Address Filtering on Reception

This IP supports the frame reception in a full-duplex operation mode.
This IP can pass all received Ethernet frames data to an upper layer via xMII Interface.
This IP strips off automatically SFDs/SMDs, preambles, FRAG_COUNTs and mCRCs from all the frames.
The FCS reception by the upper layer can be set though MRGC.RCPT register.
This IP identifies the frames with the special format and informs to the upper layer. It is indicated with the dedicated flag in status information followed payload data. The table below indicates the frames with the special format.

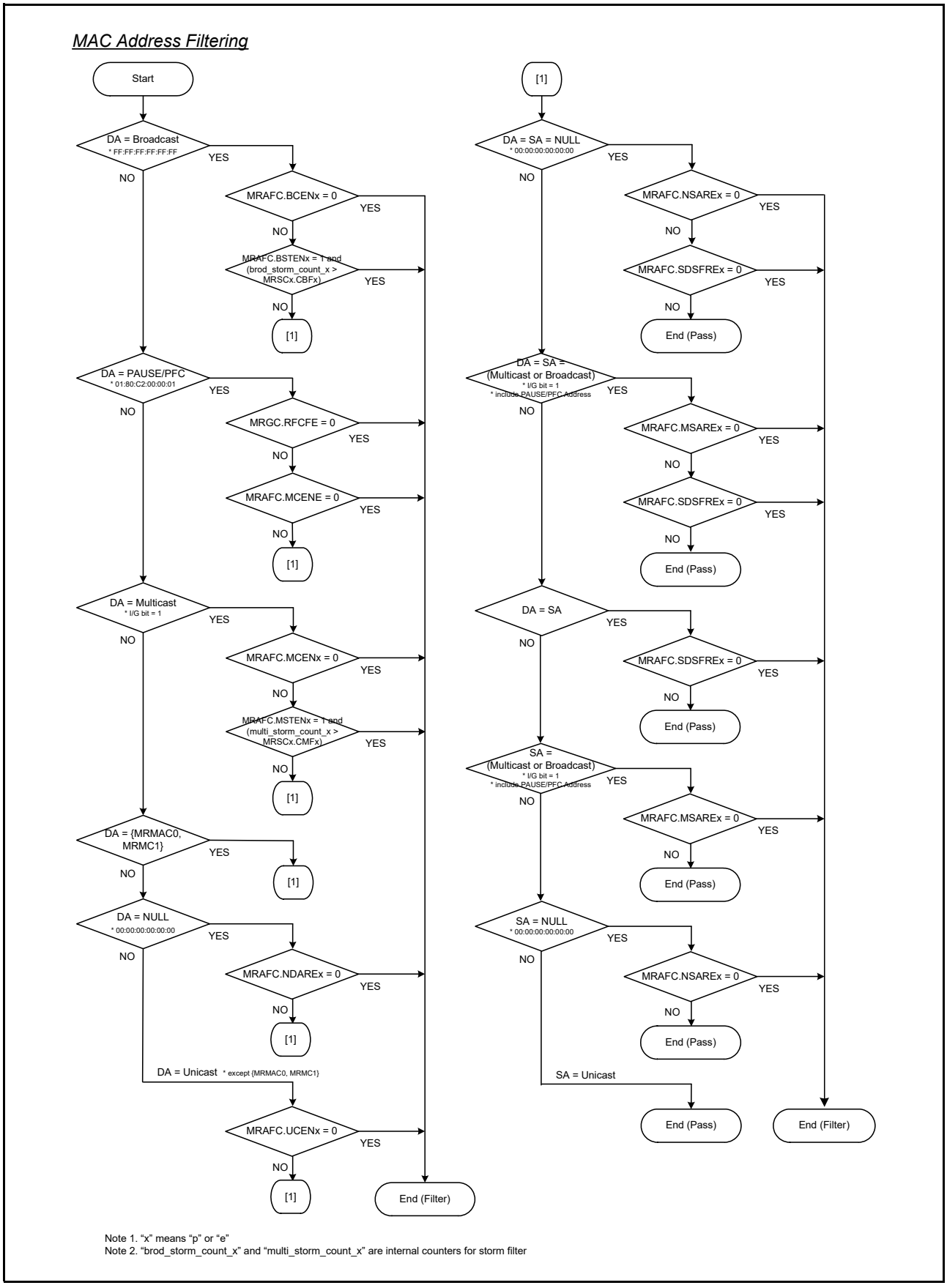
Table 37.8 Operation for Special Format Frame

Frame	Format	Operation
Broadcast frame	DA: FF_FF_FF_FF_FF_FF	Filtering possible.
Multicast frame	DA: B'xxxx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx	Filtering possible.
Pause frame	DA: 01_80_C2_00_00_01 TYPE: 88_08 OPCODE: 00_01	This IP handles Flow Control, refer to section 37.3.1, Flow Control. Pause information is not sent to upper layer.
PFC frame	DA: 01_80_C2_00_00_01 TYPE: 88_08 OPCODE: 01_01	This IP handles Flow Control, refer to section 37.3.1, Flow Control. Pause information is not sent to upper layer.
Magic packet	When the MAC address (MRMAC0.MAU[15:0], MRMAC1.MAL[31:0]) is 00-11-22-33-44-55h, this IP would scan for the data sequence: Destination-Address Source-Address .....MISC..... FF FF FF FF FF FF 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 .....MISC..... FCS This sequence can be located anywhere in the Magic Packet payload. The Magic Packet is formed with a standard Ethernet header and FCS.	Interrupt. Provide information to upper layer. Provide magic packet interrupt (MMIS2.MPDIS). Magic packets are only detected as E-frame.

This IP supports the MAC address filtering for Unicast, Multicast, Broadcast, Broadcast storm (continuous Broadcast frame reception) and Multicast storm (continuous Multicast frame reception). It uses the destination address (DA) and source Address (SA) of a received frame for the address filtering. When the filter condition is satisfied, the received frames are transferred to an upper layer. If not satisfied, the received frames are transferred to the upper layer with the error flag of "Frame filtered".
The filter condition is configured in Reception filter configuration register (MRAFC). It can be individually configured for Unicast address, Multicast address, and Broadcast address.
And this IP also supports Promiscuous mode.
When MRAFC.UCENx, MCENx, BCENx, NDAREx, SDSFREx, NSAREx and MSAREx (x = P, N) are set to 1, all frames are transferred to the upper layer without error flag of "Frame filtered".

Note: Regardless of MRAFC configuration, Verify and Respond are not transferred to the upper layer.

The flow of next page shows the MAC Address filtering.



Note 1. “x” means “p” or “e”
Note 2. “brod_storm_count_x” and “multi_storm_count_x” are internal counters for storm filter

Figure 37.12 MAC Address Filtering

37.3.6 Operation Modes

Operational mode is controlled. Please refer to IPS. RMAC received a mode control signal for controlling operational mode of RMAC.

37.3.6.1 Frame Handling

In transmission, this IP transmits all frames from an upper module (e.g. ETHA) when this IP is in operation state,e.g. when ETHA is in OPERATION state.
In reception, this IP only receives frames from PHY when this IP is in operation state, e.g. when ETHA is in OPERATION state, and outputs them to the upper module(e.g. ETHA). In other modes, this IP discards all received frames from PHY internally, and does not output them to the upper module.

37.3.7 Software Flows

Restrictions:

- SW: Please follow to the flow in this section.

37.3.7.1 Initialization Flow

The figure below shows the initialization flow.

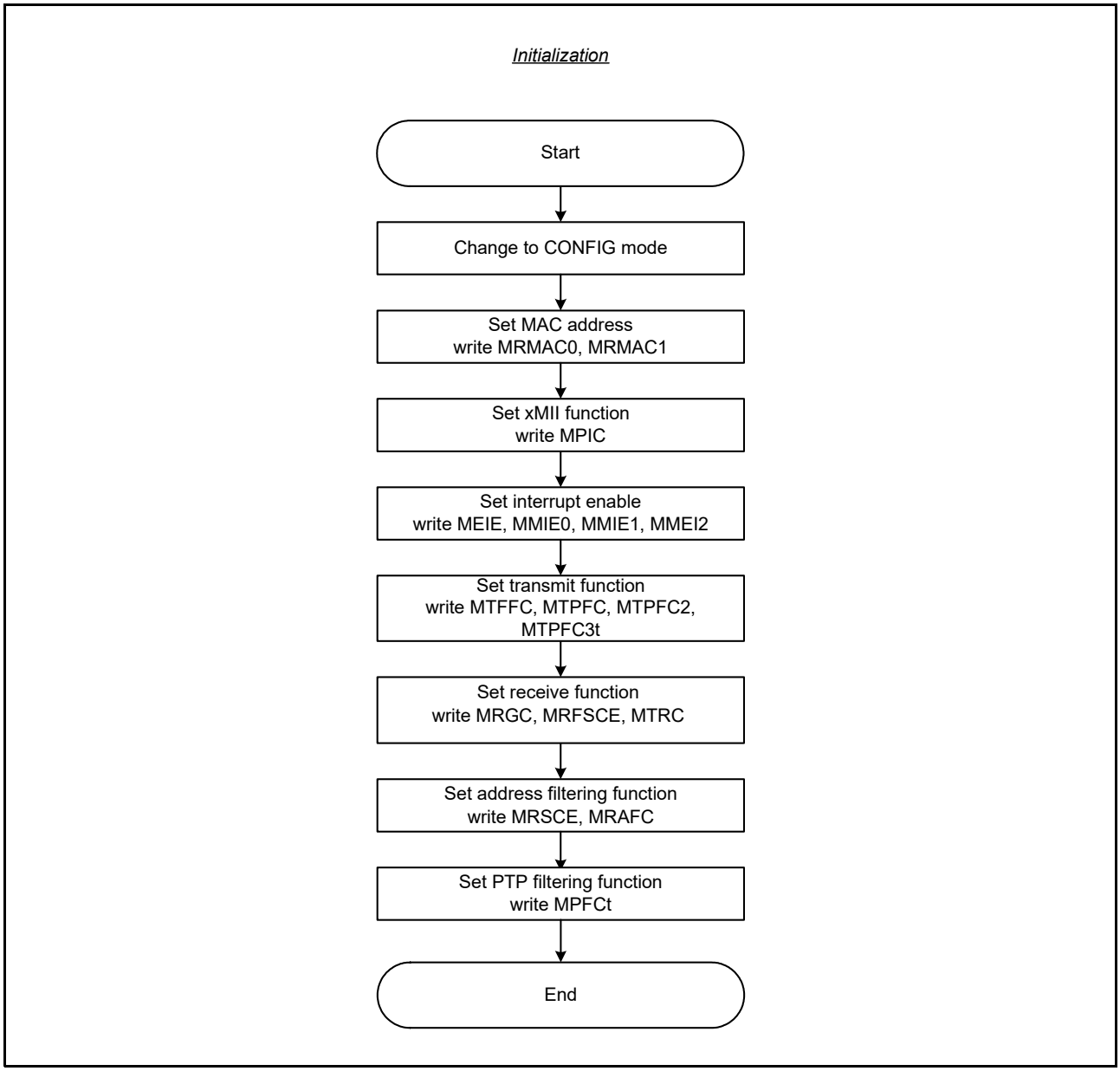


Figure 37.13 Initialization Flow

37.3.7.2 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except RESET mode.
The figure below shows the interrupt handling flow.

Restrictions:

- This flow is not usable in RESET modes.

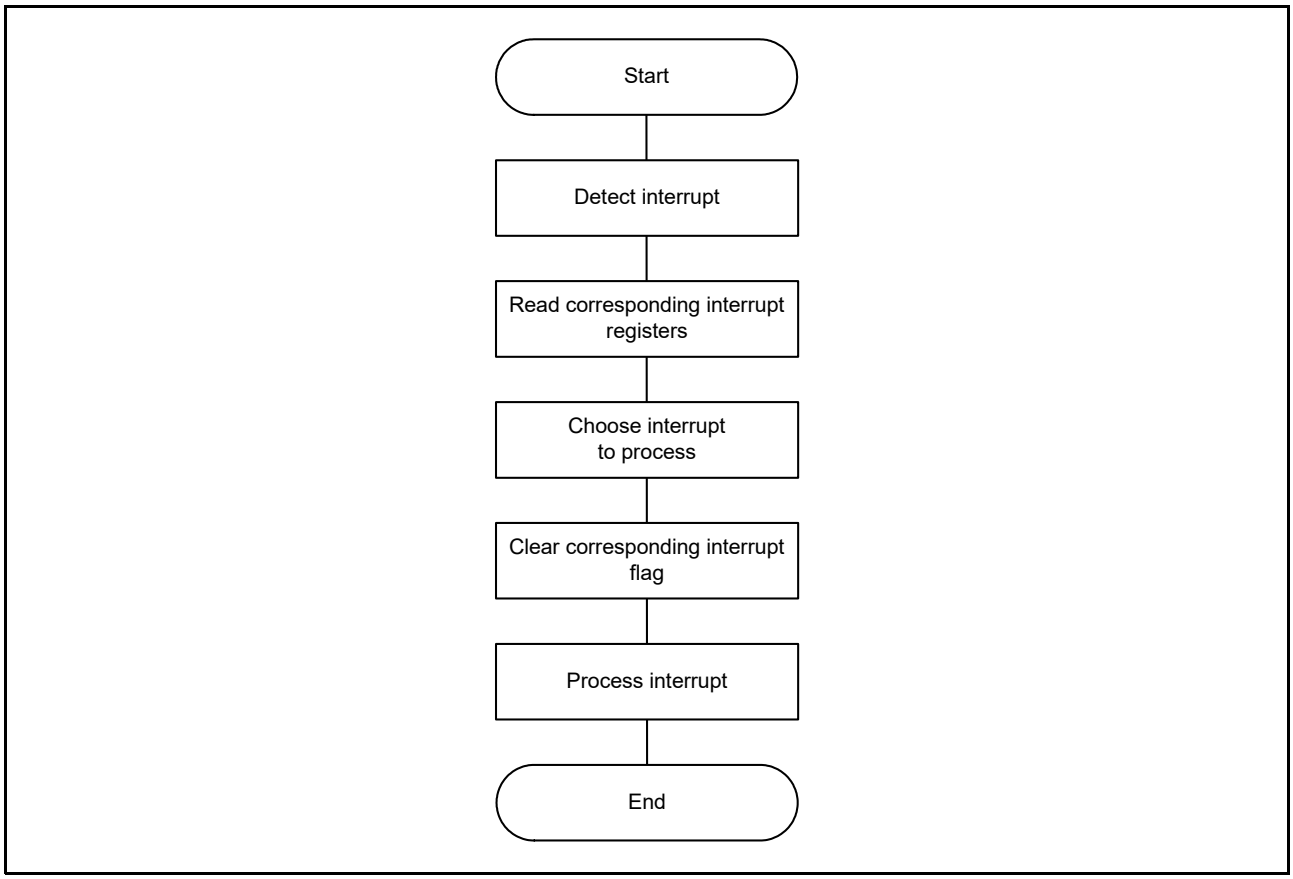


Figure 37.14 Interrupt Handling Flow

37.3.7.3 PHY MDIO and eMDIO Access Flows

The figure below shows the PHY MDIO and eMDIO access flow.

Restrictions:

- This flow is not usable in RESET modes.

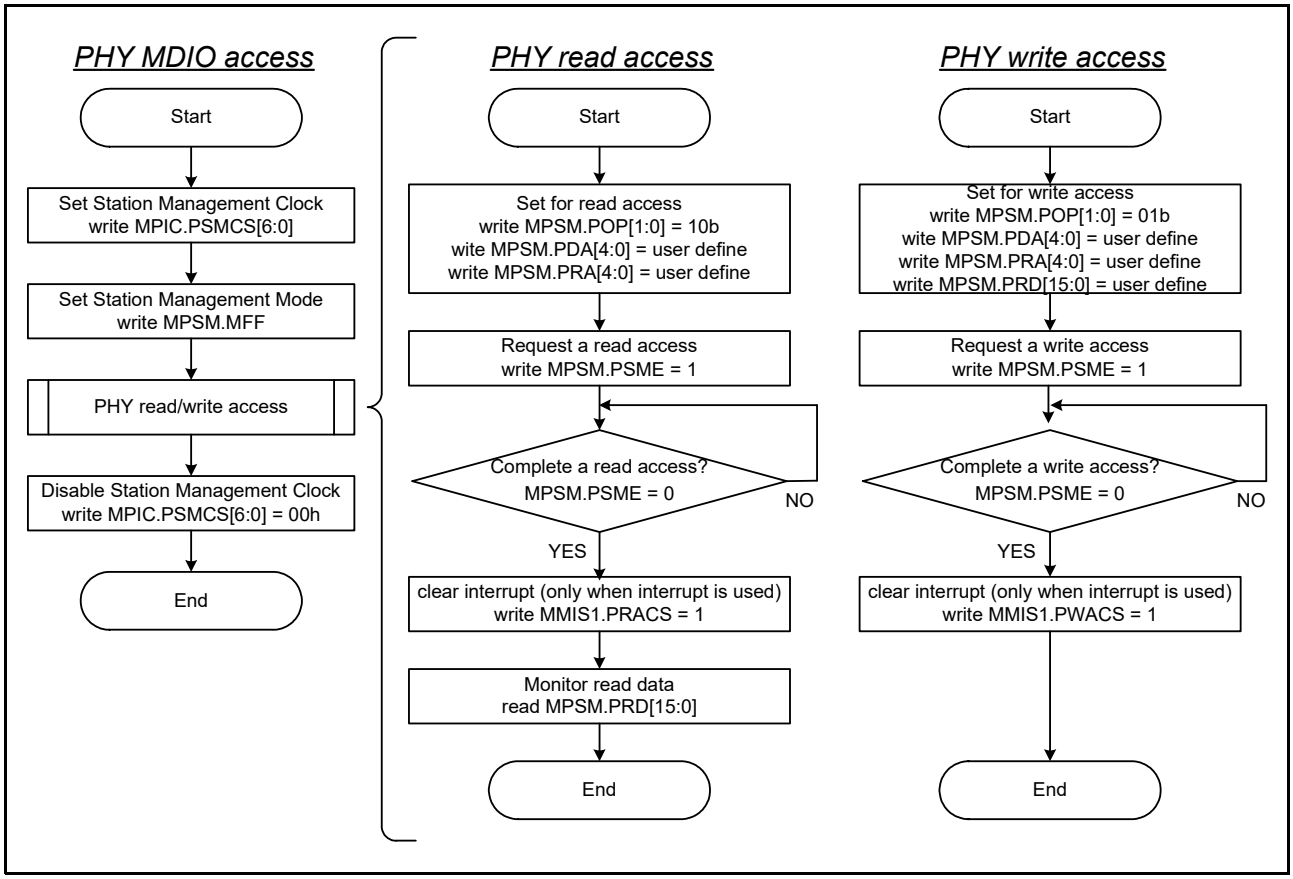


Figure 37.15 PHY MDIO Access Flow

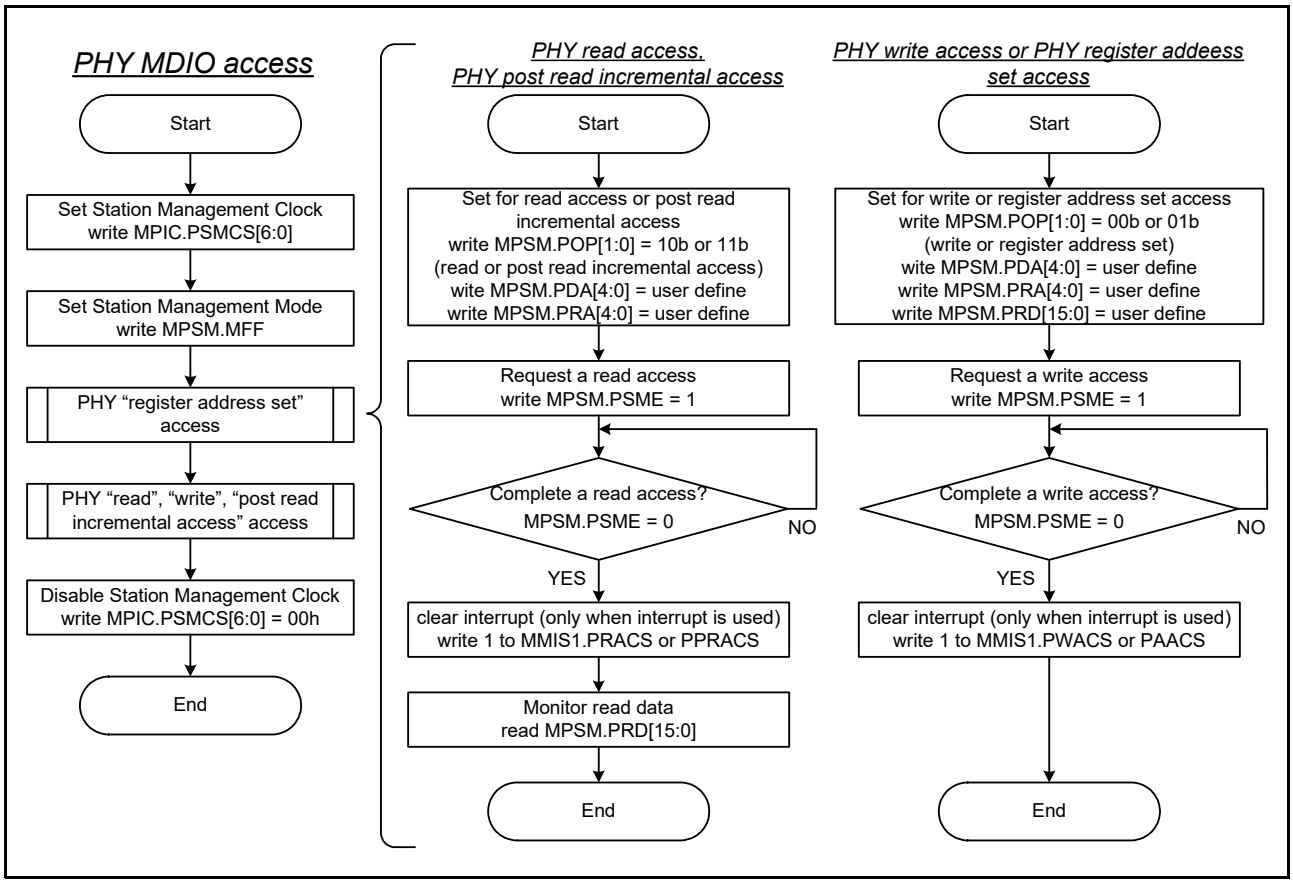


Figure 37.16 PHY eMDIO Access Flow

37.3.7.4 LPI Transition for Transmit Port

The figure below shows the LPI transition flow. In case of resumed from LPI, the transmission should be requested until PHY wake-up completed.

Restrictions:

- This flow is not usable in RESET modes.

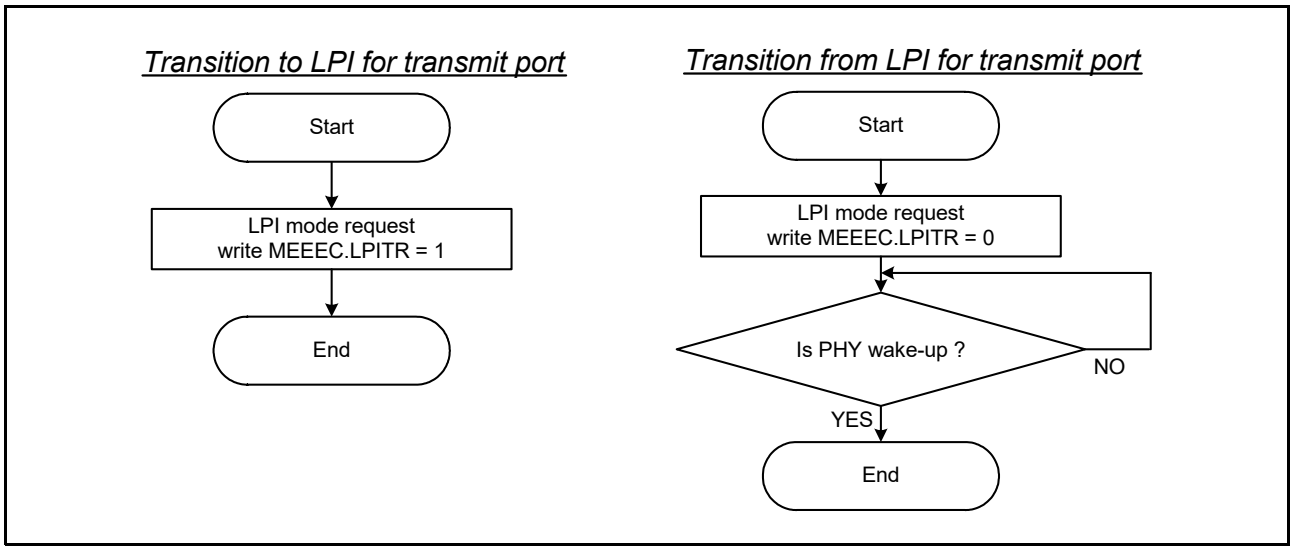


Figure 37.17 LPI Transition for Transmit Port

37.3.7.5 LINK Verification

The figure below shows Link Verification flow.

Restrictions:

- This flow is usable in OPERATIONAL mode.

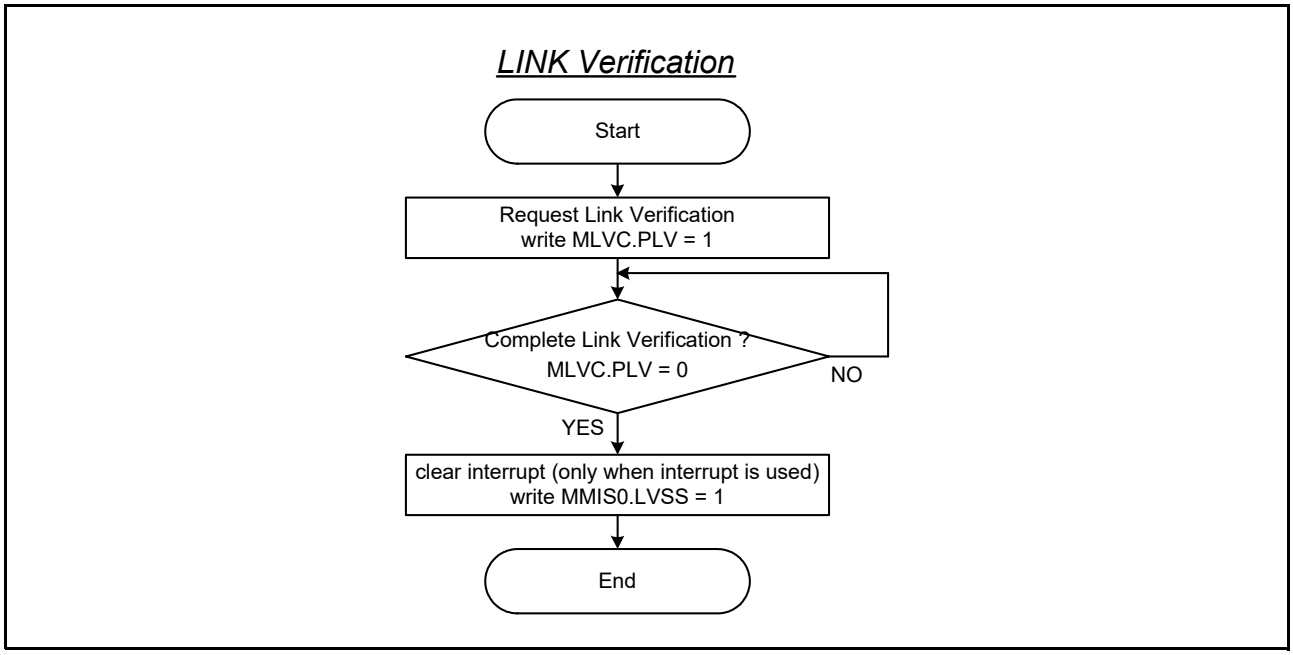


Figure 37.18 Link Verification Flow

38. Ethernet CPU Agent (GWCA)

38.1 Overview

GWCA (Ethernet CPU Agent) is a CPU interface.
The GWCA consists of an agent interface module allowing communication within the ESWM [ESWM]. It handles the data exchange between the ESWM and the GWCPU subsystem. SW involvement is required for configuration of the GateWay AXI bus interface.

38.1.1 Features

GWCA features are described Table 38.1.

Table 38.1 GWCA Feature List

Function		Details
Interfaces	Switch mode	Interface to receive the switch mode [MFWD]
	Pause interface	Interface to pause TX queues [COMA]
	AXI Master interface	Master interface conforming to AXI 3 AMBA protocol. It is used for data transfer between GWCPU and GWCA. • Data bit width: 128 bits • Number of outstanding read: 8 • Number of outstanding write: 8 • Number of ID read: 8 (from 0 to 7) • Number of ID write: 8 (from 0 to 7) • Incremental Access only • Supports unaligned transactions • Supports out of order • Supports interleave on read bus
	SFR interface	Interface to access GWCA SFRs [COMA]
	Interrupts	GWCA interrupt to CPU
	Fabric interface	Interface to communicate with the data, TAG and pointer RAM [MFWD].
	Descriptor bus	Interface to receive frame information to send them to CPU [MFWD].
	BP Request	Interface to receive BPs to store frames in the data, TAG and pointer RAM [COMA].
	BP Release	Interface to release the BPs that has been used [COMA].
	TX Timestamp Capture interface	Interface to receive TX timestamps and send it to CPU [RMAC].
	RAM interfaces	Interface to communication with RAMs [ESWM]
Data provision	Timestamp	TX timestamp for transmission to CPU RAM [RMAC]

38.1.2 GWCA Block Diagram

Figure 38.1 shows GWCA block diagram.

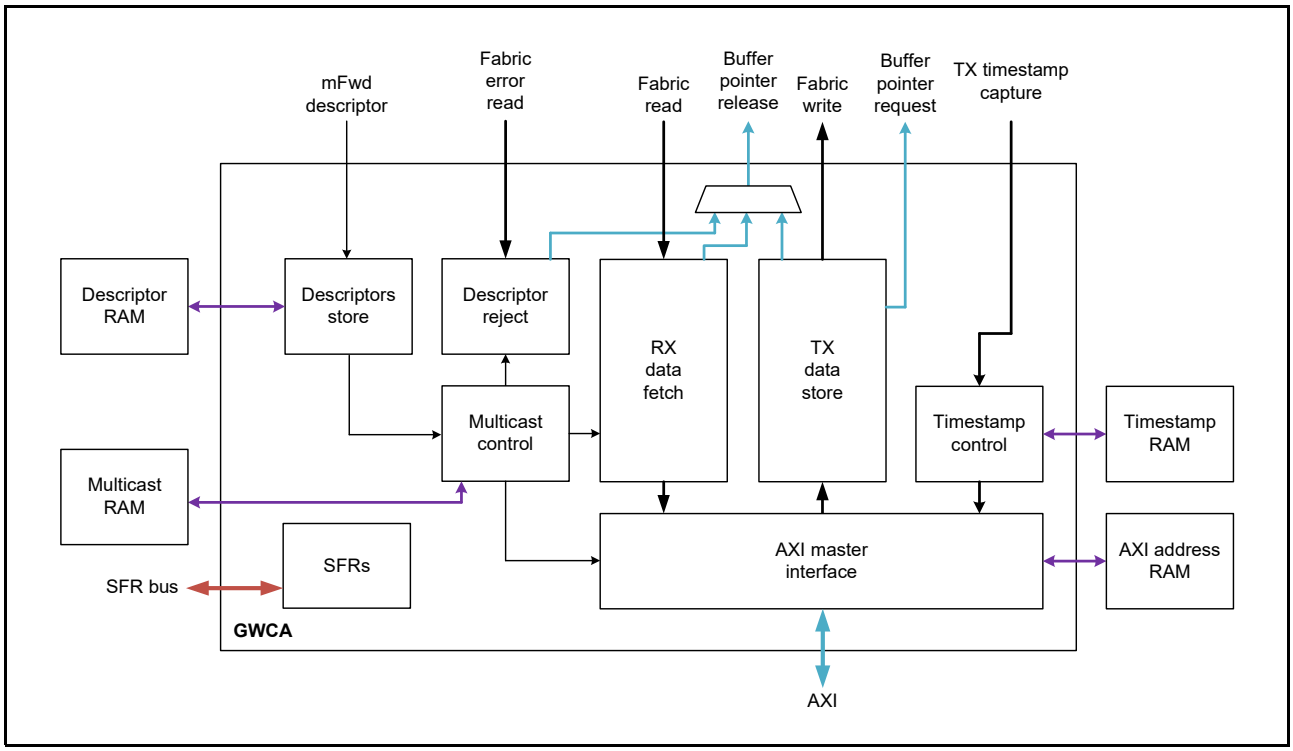


Figure 38.1 GWCA Block Diagram

Table 38.2 GWCA Functional Blocks

Block Name	Function
TX data store	Transmits frames from AXI master to the local RAM
RX data fetch	Transmits frame from local RAM to AXI master
Descriptor store	Receives the descriptors from forwarding engine and controls the priority between descriptors
Descriptor reject	Reject descriptors
Multicast control	Controls multicast transactions
Timestamp control	Receives timestamps for Ethernet agents, saves them and transmits them to the AXI master
AXI master interface	AXI master for data exchange with the CPU
SFRs	GWCA SFRs

38.2 Register Descriptions

38.2.1 GWCA Mode Function Registers

38.2.1.1 Mode Configuration Register (GWMC)

Address(es): GWCA0.GWMC 874C E000h, GWCA0_NS.GWMC 074C E000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OPC[1:0]	Operating Mode Command	b1 b0 0 0: Enter RESET mode 0 1: Enter DISABLE mode 1 0: Enter CONFIG mode 1 1: Enter OPERATION mode	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

OPC[1:0] Bits (Operating Mode Command)

Used to set GWCA mode. For more details, refer to section 38.3.7, Operation Modes.

[Restriction]

- HW: These bits are not writable if its value is different than GWMS.OPS[1:0].

38.2.1.2 Mode Status Register (GWMS)

Address(es): GWCA0.GWMS 874C E004h, GWCA0_NS.GWMS 074C E004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OPS[1:0]	Operating Mode Status	b1 b0 0 0: RESET mode 0 1: DISABLE mode 1 0: CONFIG mode 1 1: OPERATION mode	R
b31 to b2	—	Reserved	These bits are read as 0.	R

OPS[1:0] Bits (Operating Mode Status)

Indicates the current GWCA mode. For more details, refer to section 38.3.7, Operation Modes.

[Update conditions]

- HW: Mode change completed.
- These bits are change at “Maximum frame communication time (exp: 64 Kbyte/100 Mbps = 5.12 ms)” from OPERATION to DISABLE.

38.2.2 Reception Function Registers

38.2.2.1 IPV Remapping Configuration Register [802.1Q] (GWIRC)

Address(es): GWCA0.GWIRC 874C E010h, GWCA0_NS.GWIRC 074C E010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	IPVR7[2:0]			—	IPVR6[2:0]			—	IPVR5[2:0]			—	IPVR4[2:0]		
Value after reset:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IPVR3[2:0]			—	IPVR2[2:0]			—	IPVR1[2:0]			—	IPVR0[2:0]		
Value after reset:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IPVR0[2:0]	IPV Remapping 0	Configure to which descriptor queue descriptor received with IPV 0 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 0).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6 to b4	IPVR1[2:0]	IPV Remapping 1	Configure to which descriptor queue descriptor received with IPV 1 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 1).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b10 to b8	IPVR2[2:0]	IPV Remapping 2	Configure to which descriptor queue descriptor received with IPV 2 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 2).	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14 to b12	IPVR3[2:0]	IPV Remapping 3	Configure to which descriptor queue descriptor received with IPV 3 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 3).	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b18 to b16	IPVR4[2:0]	IPV Remapping 4	Configure to which descriptor queue descriptor received with IPV 4 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 4).	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b22 to b20	IPVR5[2:0]	IPV Remapping 5	Configure to which descriptor queue descriptor received with IPV 5 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 5).	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26 to b24	IPVR6[2:0]	IPV Remapping 6	Configure to which descriptor queue descriptor received with IPV 6 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 6).	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b30 to b28	IPVR7[2:0]	IPV Remapping 7	Configure to which descriptor queue descriptor received with IPV 7 will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 7).	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

38.2.2.2 RX Descriptor Queue Security Configuration Register (GWRDQSC)

Address(es): GWCA0.GWRDQSC 874C E014h, GWCA0_NS.GWRDQSC 074C E014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RDQSL7	RDQSL6	RDQSL5	RDQSL4	RDQSL3	RDQSL2	RDQSL1	RDQSL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RDQSL0	RX Descriptor Queue 0 Security Level	0: Queue q unsecure 1: Queue q secure (q = 0 to 7)	R/W
b1	RDQSL1	RX Descriptor Queue 1 Security Level		R/W
b2	RDQSL2	RX Descriptor Queue 2 Security Level		R/W
b3	RDQSL3	RX Descriptor Queue 3 Security Level		R/W
b4	RDQSL4	RX Descriptor Queue 4 Security Level		R/W
b5	RDQSL5	RX Descriptor Queue 5 Security Level		R/W
b6	RDQSL6	RX Descriptor Queue 6 Security Level		R/W
b7	RDQSL7	RX Descriptor Queue 7 Security Level		R/W
b31 to b8	—	Reserved	This bit is read as 0. The write value should be 0.	R

RDQSLq Bit (RX Descriptor Queue q Security Level) (q = 0 to 7)

When a queue is secured, an unsecure descriptor cannot enter it (when a descriptor is received from forwarding engine with FDESCR.SEC [MFWD] equal to 0).

38.2.2.3 RX Descriptor Queue Control Register (GWRDQC)

Address(es): GWCA0.GWRDQC 874C E018h, GWCA0_NS.GWRDQC 074C E018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	RDQP7	RDQP6	RDQP5	RDQP4	RDQP3	RDQP2	RDQP1	RDQP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RDQD7	RDQD6	RDQD5	RDQD4	RDQD3	RDQD2	RDQD1	RDQD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RDQD0	RX Descriptor Queue 0 Disable	0: Queue q enabled 1: Queue q disabled (q = 0 to 7)	R/W
b1	RDQD1	RX Descriptor Queue 1 Disable		R/W
b2	RDQD2	RX Descriptor Queue 2 Disable		R/W
b3	RDQD3	RX Descriptor Queue 3 Disable		R/W
b4	RDQD4	RX Descriptor Queue 4 Disable		R/W
b5	RDQD5	RX Descriptor Queue 5 Disable		R/W
b6	RDQD6	RX Descriptor Queue 6 Disable		R/W
b7	RDQD7	RX Descriptor Queue 7 Disable		R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RDQP0	RX Descriptor Queue 0 Pause	0: Queue q active 1: Queue q paused (q = 0 to 7)	R/W
b17	RDQP1	RX Descriptor Queue 1 Pause		R/W
b18	RDQP2	RX Descriptor Queue 2 Pause		R/W
b19	RDQP3	RX Descriptor Queue 3 Pause		R/W
b20	RDQP4	RX Descriptor Queue 4 Pause		R/W
b21	RDQP5	RX Descriptor Queue 5 Pause		R/W
b22	RDQP6	RX Descriptor Queue 6 Pause		R/W
b23	RDQP7	RX Descriptor Queue 7 Pause		R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

RDQDq Bit (RX Descriptor Queue q Disable) (q = 0 to 7)

Avoid descriptor from forwarding engine [MFWD] to enter the corresponding queue. The Forwarding Engine will reject the corresponding descriptor.

If disabled queue data is not needed anymore, the GWRMFSCq.MFS[15:0] bits can be set to 0 in order to avoid more frames to go to the CPU RAM.

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing conditions]

- SW: Writing 0 to this bit will clear it.
- HW: Going out of OPERATION mode will clear this register (GWMC.OPC[1:0] ≠ 11b).

RDQPq Bit (RX Descriptor Queue q Pause) (q = 0 to 7)

Avoid frames to be sent to the CPU by stopping descriptor fetching from the descriptor RAM.

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing conditions]

- SW: Writing 0 to this bit will clear it.
- HW: Going out of OPERATION mode will clear this bit (GWMC.OPC[1:0] ≠ 11b).

[Restriction]

- SW: Pausing a queue during a long time could result in switch overflow. In case of switch overflow [COMA] this bit should be set to 0.

[Caution]

- This bit is used to stop a queue but forwarding engine [MFWD] will not stop sending descriptor to the corresponding queue. In this case the queue might overflow.

38.2.2.4 RX Descriptor Queue Arbitration Control Register (GWRDQAC)

Address(es): GWCA0.GWRDQAC 874C E01Ch, GWCA0_NS.GWRDQAC 074C E01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDQA7[3:0]				RDQA6[3:0]				RDQA5[3:0]				RDQA4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDQA3[3:0]				RDQA2[3:0]				RDQA1[3:0]				RDQA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RDQA0[3:0]	RX Descriptor Queue 0 Arbitration	0 0 0 0: Queue i strict arbitration Others: Queue i WRR arbitration (i = 0 to 7)	R/W
b7 to b4	RDQA1[3:0]	RX Descriptor Queue 1 Arbitration		R/W
b11 to b8	RDQA2[3:0]	RX Descriptor Queue 2 Arbitration		R/W
b15 to b12	RDQA3[3:0]	RX Descriptor Queue 3 Arbitration		R/W
b19 to b16	RDQA4[3:0]	RX Descriptor Queue 4 Arbitration		R/W
b23 to b20	RDQA5[3:0]	RX Descriptor Queue 5 Arbitration		R/W
b27 to b24	RDQA6[3:0]	RX Descriptor Queue 6 Arbitration		R/W
b31 to b28	RDQA7[3:0]	RX Descriptor Queue 7 Arbitration		R/W

RDQAi[3:0] Bits (RX Descriptor Queue i Arbitration) (i = 0 to 7)

For more details, refer to (2) Descriptor Arbitration.

[Restrictions]

- SW: In hybrid arbitration mode, all the queues with a GWRDQAC.RDQAi[3:0] value different than 0 should have a consecutive priority (priority refers to the queue number i).
- SW: In all arbitration mode except strict priority arbitration mode, at least two queues should have a GWRDQAC.RDQAi[3:0] different than 0.

38.2.2.5 RX General Configuration Register (GWRGC)

Address(es): GWCA0.GWRGC 874C E020h, GWCA0_NS.GWRGC 074C E020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCPT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCPT	Receive CRC Pass Through [802.3]	0: FCS is not passed to GWCPU. If a descriptor and is received from forwarding engine with FDESCR.FI equal to 1, GWCA remove the last 4 bytes of the Frame Data (FCS). 1: FCS is passed to GWCPU. GWCA do not remove FCS from the Frame which has one (a descriptor and is received from forwarding engine with FDESCR.FI equal to 1) if the frame has not been modified by the switch (No update due to VLAN tagging/un-tagging and no update requested by L3 routing [MFWD]).	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

RCPT Bit (Receive CRC Pass Through [802.3])

This bit selects to pass FCS field on the reception frame.

[Restriction]

- HW: GWCA does not check FCS integrity.

38.2.2.6 Reception Maximum Frame Size Configuration Register q (GWRMFSCq) (q = 0 to 7)

Address(es): GWCA0.GWRMFSC0 874C E040h, GWCA0.GWRMFSC1 874C E044h, GWCA0.GWRMFSC2 874C E048h, GWCA0.GWRMFSC3 874C E04Ch, GWCA0.GWRMFSC4 874C E050h, GWCA0.GWRMFSC5 874C E054h, GWCA0.GWRMFSC6 874C E058h, GWCA0.GWRMFSC7 874C E05Ch, GWCA0_NS.GWRMFSC0 074C E040h, GWCA0_NS.GWRMFSC1 074C E044h, GWCA0_NS.GWRMFSC2 074C E048h, GWCA0_NS.GWRMFSC3 074C E04Ch, GWCA0_NS.GWRMFSC4 074C E050h, GWCA0_NS.GWRMFSC5 074C E054h, GWCA0_NS.GWRMFSC6 074C E058h, GWCA0_NS.GWRMFSC7 074C E05Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MFS[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MFS[15:0]	Maximum Frame Size	Maximum frame size for descriptor queue q. All bigger frames will be rejected.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

MFS[15:0] Bits (Maximum Frame Size)

[Restriction]

- HW: The maximum frame size comparison takes in account the frame size which will actually be written by the AXI master (after VLAN un-tagging, R-TAG insertion and L23 update) but always counts in this size FCS even if not received by CPU.

38.2.2.7 Reception Descriptor Queue q Depth Configuration Register (GWRDQDCq) (q = 0 to 7)

Address(es): GWCA0.GWRDQDC0 874C E060h, GWCA0.GWRDQDC1 874C E064h, GWCA0.GWRDQDC2 874C E068h, GWCA0.GWRDQDC3 874C E06Ch, GWCA0.GWRDQDC4 874C E070h, GWCA0.GWRDQDC5 874C E074h, GWCA0.GWRDQDC6 874C E078h, GWCA0.GWRDQDC7 874C E07Ch, GWCA0_NS.GWRDQDC0 074C E060h, GWCA0_NS.GWRDQDC1 074C E064h, GWCA0_NS.GWRDQDC2 074C E068h, GWCA0_NS.GWRDQDC3 074C E06Ch, GWCA0_NS.GWRDQDC4 074C E070h, GWCA0_NS.GWRDQDC5 074C E074h, GWCA0_NS.GWRDQDC6 074C E078h, GWCA0_NS.GWRDQDC7 074C E07Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DQD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DQD[9:0]	Descriptor Queue Depth	Number of descriptors that can be contained in descriptor queue q.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

DQD[9:0] Bits (Descriptor Queue Depth)

For details refer to (1) Descriptor Storage.

[Restriction]

- SW: The sum of DQD[9:0] fields should always be less or equal to 512.

38.2.2.8 RX Descriptor Queue q Monitoring Register (GWRDQMq) (q = 0 to 7)

Address(es): GWCA0.GWRDQM0 874C E080h, GWCA0.GWRDQM1 874C E084h, GWCA0.GWRDQM2 874C E088h, GWCA0.GWRDQM3 874C E08Ch, GWCA0.GWRDQM4 874C E090h, GWCA0.GWRDQM5 874C E094h, GWCA0.GWRDQM6 874C E098h, GWCA0.GWRDQM7 874C E09Ch, GWCA0_NS.GWRDQM0 074C E080h, GWCA0_NS.GWRDQM1 074C E084h, GWCA0_NS.GWRDQM2 074C E088h, GWCA0_NS.GWRDQM3 074C E08Ch, GWCA0_NS.GWRDQM4 074C E090h, GWCA0_NS.GWRDQM5 074C E094h, GWCA0_NS.GWRDQM6 074C E098h, GWCA0_NS.GWRDQM7 074C E09Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DNQ[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DNQ[9:0]	Descriptor Number in Queue	Indicate the current number of descriptors stored in RX Descriptor Queue q.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

DNQ[9:0] Bits (Descriptor Number in Queue)

[Increment condition]

- HW: Incremented by 1 when a descriptor is received from the forwarding engine [MFWD] for the corresponding queue and, the queue is not full, there is no descriptor security error and the queue is not disabled.

[Decrement condition]

- HW: Decremented by 1 when a descriptor is read by GWCA to send data to AXI Master Interface or to reject data.

38.2.2.9 RX Descriptor Queue q Max Level Monitoring Register (GWRDQMLMq) (q = 0 to 7)

Address(es): GWCA0.GWRDQMLM0 874C E0A0h, GWCA0.GWRDQMLM1 874C E0A4h, GWCA0.GWRDQMLM2 874C E0A8h, GWCA0.GWRDQMLM3 874C E0ACh, GWCA0.GWRDQMLM4 874C E0B0h, GWCA0.GWRDQMLM5 874C E0B4h, GWCA0.GWRDQMLM6 874C E0B8h, GWCA0.GWRDQMLM7 874C E0BCh, GWCA0_NS.GWRDQMLM0 074C E0A0h, GWCA0_NS.GWRDQMLM1 074C E0A4h, GWCA0_NS.GWRDQMLM2 074C E0A8h, GWCA0_NS.GWRDQMLM3 074C E0ACh, GWCA0_NS.GWRDQMLM4 074C E0B0h, GWCA0_NS.GWRDQMLM5 074C E0B4h, GWCA0_NS.GWRDQMLM6 074C E0B8h, GWCA0_NS.GWRDQMLM7 074C E0BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DMLQ[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	DMLQ[9:0]	Descriptor Max Level in Queue	Indicates the maximum number of descriptors that has been stored in RX Descriptor Queue q.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

DMLQ[9:0] Bits (Descriptor Max Level in Queue)

[Clearing conditions]

- HW: Being in RESET mode will clear this register (GWMC.OPC[1:0] = 00b).
- SW: By reading this register, it is cleared to GWRDQMq.DNQ[9:0].

[Increment condition]

- HW: Increment to GWRDQMq.DNQ[9:0] value when smaller than GWRDQMq.DNQ[9:0].

38.2.3 Reception Multicast Function Registers

38.2.3.1 Multicast Table Initialization Register Monitoring Register (GWMTIRM)

Address(es): GWCA0.GWMTIRM 874C E100h, GWCA0_NS.GWMTIRM 074C E100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MTR	MTIOG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTIOG	Multicast Table Initialization Ongoing		R/W
b1	MTR	Multicast Table Ready		R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

MTIOG Bit (Multicast Table Initialization Ongoing)

[Setting condition]

- SW: By writing 1 to this bit. It starts Multicast Table initialization.

[Clearing condition]

- HW: This bit is cleared when Multicast Table initialization is finished.
  Min clk_period*64 = [exp] (3.125)*128 = 400 ns

MTR Bit (Multicast Table Ready)

[Setting condition]

- When GWMTIRM.MTIOG is getting cleared.

[Clearing condition]

- By writing 1 to GWMTIRM.MTIOG.

38.2.3.2 Multicast Table Learning Setting Register (GWMSTLS)

Address(es): GWCA0.GWMSTLS 874C E104h, GWCA0_NS.GWMSTLS 074C E104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	MSENL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MNL[2:0]			—	—	MNRCNL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	MNRCNL[5:0]	Multicast Next Descriptor Chain Number Learn		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	MNL[2:0]	Multicast Number Learn	b10 b8 0 0 0: multicast 0 (1 target, no multicast) 0 0 1: multicast 1 (2 targets) : 1 1 1: multicast 7 (8 targets)	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b21 to b16	MSENL[5:0]	Multicast Setting Entry Number Learn		R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R

MNRCNL[5:0] Bits (Multicast Next Descriptor Chain Number Learn)

These bits are used to set the next address in the multicast table to be learnt in GWMSTLS.MSENL[5:0] multicast table entry, refer to (3) Multicast Learning.

MNL[2:0] Bits (Multicast Number Learn)

These bits are used to set the multicast number that should be learnt in GWMSTLS.MSENL[5:0] multicast table entry.

MSENL[5:0] Bits (Multicast Setting Entry Number Learn)

These bits are used to set in which address the current learning should happen.

38.2.3.3 Multicast Table Learning Result Register (GWMSTLR)

Address(es): GWCA0.GWMSTLR 874C E108h, GWCA0_NS.GWMSTLS 074C E108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MTLF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTLF	Multicast Table Learning Fail	0: Entry learning did not fail because the Multicast table is not ready. 1: Entry learning failed because the Multicast table is not ready.	R
b30 to b1	—	Reserved	This bit is read as 0.	R
b31	MTL	Multicast Table Learning		R

MTLF Bit (Multicast Table Learning Fail)

[Update condition]

- HW: GWMSTLR.MTL clear event.

MTL Bit (Multicast Table Learning)

[Setting condition]

- SW: Writing GWMSTLS register will set this bit.

[Clearing condition]

- HW: This bit will be de-asserted when the Multicast RAM write is completed.

38.2.3.4 Multicast Table Searching Setting Register (GWMSTSS)

Address(es): GWCA0.GWMSTSS 874C E10Ch, GWCA0_NS.GWMSTSS 074C E10Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	MSENS[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	MSENS[5:0]	Multicast Setting Entry Number Search	This register is used to read multicast table.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.3.5 Multicast Table Searching Result Register (GWMSTSR)

Address(es): GWCA0.GWMSTSR 874C E110h, GWCA0_NS.GWMSTSR 074C E110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MTS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MNR[2:0]			—	—	MNRCNR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	MNRCNR[5:0]	Multicast Next RX Descriptor Chain Number Result		R
b7, b6	—	Reserved	These bits are read as 0.	R
b10 to b8	MNR[2:0]	Multicast Number Result		R
b30 to b11	—	Reserved	These bits are read as 0.	R
b31	MTS	Multicast Table Searching		R

MNRCNR[5:0] Bits (Multicast Next RX Descriptor Chain Number Result)

These bits are used to read multicast table.

[Update condition]

- HW: GWMSTSR.MTS clear event.

MNR[2:0] Bits (Multicast Number Result)

These bits are used to read multicast table.

[Update condition]

- HW: GWMSTSR.MTS clear event.

MTS Bit (Multicast Table Searching)

This bit is used to read multicast table.

[Setting condition]

- SW: Writing GWMSTSS register will set this bit.

[Clearing condition]

- HW: This bit will be de-asserted when the Multicast RAM read is completed.

38.2.4 MAC Function Registers

38.2.4.1 MAC Address Configuration Register 0 (GWMAC0)

Address(es): GWCA0.GWMAC0 874C E120h, GWCA0_NS.GWMAC0 074C E120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MAU[15:0]	MAC Address Upper Part	These bits are used to set the upper 16 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 0123h in this register.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.4.2 MAC Address Configuration Register 1 (GWMAC1)

Address(es): GWCA0.GWMAC1 874C E124h, GWCA0_NS.GWMAC1 074C E124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MAL[31:0]	MAC Address Lower Part	These bits are used to set the lower 32 bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 456789ABh in this register.	R/W

38.2.5 TAG Function Registers

38.2.5.1 VLAN Control Configuration Register (GWVCC)

Address(es): GWCA0.GWVCC 874C E130h, GWCA0_NS.GWVCC 074C E130h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	VEM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VIM	VLAN Ingress Mode	0: Incoming VLAN mode 1: Port based VLAN mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	VEM[2:0]	VLAN Egress Mode	b18 b16 0 0 0: No VLAN mode, frames are transmitted with no VLAN. 0 0 1: C-TAG VLAN mode, frames are transmitted with ingress C-TAG if there is one stored in the Local RAM. 0 1 0: HW C-TAG VLAN mode, frames are transmitted with the C-TAG stored in local RAM. 0 1 1: SC-TAG VLAN mode, frames are transmitted with ingress C-TAG and ingress S-TAG if they were stored in the Local RAM. 1 0 0: HW SC-TAG VLAN mode, frames are transmitted with the C-TAG and S-TAG stored in local RAM. Settings other than above are prohibited.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

VIM Bit (VLAN Ingress Mode)

[Restriction]

- SW: This bit should not be set to 1 if the switch is in no VLAN mode (In forwarding engine, FWGC.SVM is set to 00b [MFWD])

38.2.5.2 VLAN TAG Configuration Register (GWVTC)

Address(es): GWCA0.GWVTC 874C E134h, GWCA0_NS.GWVTC 074C E134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STD	STP[2:0]			STV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTD	CTP[2:0]			CTV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	CTV[11:0]	C-TAG VLAN		R/W
b14 to b12	CTP[2:0]	C-TAG PCP		R/W
b15	CTD	C-TAG DEI		R/W
b27 to b16	STV[11:0]	S-TAG VLAN		R/W
b30 to b28	STP[2:0]	S-TAG PCP		R/W
b31	STD	S-TAG DEI		R/W

38.2.5.3 Transmission TAG Filtering Configuration Register (GWTTFC)

Address(es): GWCA0.GWTTFC 874C E138h, GWCA0_NS.GWTTFC 074C E138h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	UT	SCRT	SCT	CRT	CT	CSRT	CST	RT	NT
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NT	No Tag	0: No Tag frame passed 1: No Tag frame rejected	R/W
b1	RT	R-TAG	0: R-TAG frame passed 1: R-TAG frame rejected	R/W
b2	CST	CoS-TAG	0: CoS-TAG frame passed 1: CoS-TAG frame rejected	R/W
b3	CSRT	CoSR-TAG	0: CoSR-TAG frame passed 1: CoSR-TAG frame rejected	R/W
b4	CT	C-TAG	0: C-TAG frame passed 1: C-TAG frame rejected	R/W
b5	CRT	CR-TAG	0: CR-TAG frame passed 1: CR-TAG frame rejected	R/W
b6	SCT	SC-TAG	0: SC-TAG frame passed 1: SC-TAG frame rejected	R/W
b7	SCRT	SCR-TAG	0: SCR-TAG frame passed 1: SCR-TAG frame rejected	R/W
b8	UT	Unknown TAG	0: Unknown Tag frame passed 1: Unknown Tag frame rejected	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.6 Timestamp Function Registers

38.2.6.1 Timestamp Descriptor Chain s Address Configuration Register 0 (GWTDCACs0) (s = 0, 1)

Address(es): GWCA0.GWTDCAC00 874C E140h, GWCA0.GWTDCAC10 874C E148h, GWCA0_NS.GWTDCAC00 074C E140h, GWCA0_NS.GWTDCAC10 074C E148h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TSCCAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSCCAU[7:0]	TS Descriptor Chain s Current Address Upper Part		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TSCCAU[7:0] Bits (TS Descriptor Chain s Current Address Upper Part)

[Update conditions]

- SW: Writing to this field with update it to the write value.
- HW: When an FEMPTY_ND descriptor is read for descriptor queue s, this field value is automatically updated to the next descriptor address upper part.
- HW: When a LINK or LINKFIX descriptor is read, this register value is updated to DESCR.PTR[39:32].

[Restriction]

- SW: To overwrite the current address, GWTDCACs0 and GWTDCACs1 should always both be written starting by GWTDCACs1 and finishing by GWTDCACs0.

38.2.6.2 Timestamp Descriptor Chain s Address Configuration Register 1 (GWTDCACs1) (s = 0, 1)

Address(es): GWCA0.GWTDCAC01 874C E144h, GWCA0.GWTDCAC11 874C E14Ch, GWCA0_NS.GWTDCAC01 074C E144h, GWCA0_NS.GWTDCAC11 074C E14Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSCCAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSCCAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TSCCAL[31:0]	TS Descriptor Chain s Current Address Lower Part		R/W

TSCCAL[31:0] Bits (TS Descriptor Chain s Current Address Lower Part)

[Update conditions]

- SW: Writing GWTDCACs0 register with update this field to the previous value written on it by the same APB.
- HW: When an FEMPTY_ND descriptor is read for descriptor queue s, this field value is automatically updated to the next descriptor address lower part.
- HW: When a LINK or LINKFIX descriptor is read, this register value is updated to DESCR.PTR[31:0].

[Restriction]

- SW: To overwrite the current address, GWTDCACs0 and GWTDCACs1 should always both be written starting by GWTDCACs1 and finishing by GWTDCACs0.

38.2.6.3 Timestamp Descriptor Chain s Configuration Register (GWTSDCCs) (s = 0, 1)

Address(es): GWCA0.GWTSDCC0 874C E160h, GWCA0.GWTSDCC1 874C E164h, GWCA0_NS.GWTSDCC0 074C E160h, GWCA0_NS.GWTSDCC1 074C E164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OSID[2:0]			—	—	—	—	—	—	DCS	TE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TE	Timer Enable	Enable timestamp reception for timestamps taken with timer s.	R/W
b1	DCS	Descriptor Chain Select	Select to which chain timestamps taken with timer s will be received.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	OSID[2:0]	OS ID	When a memory access is done for timer number s, the OSID set in this register will be used.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

TE Bit (Timer Enable)

[Restriction]

- HW: If a timestamp is received from RMAC for a timer which is disabled for timestamp reception, the timestamp will be ignored and lost. This is a valid setting (another GWCA could have taken the timestamp) so there is no flag to monitor this event.

38.2.6.4 Timestamp Number Monitoring Register (GWTSNM)

Address(es): GWCA0.GWTSNM 874C E180h, GWCA0_NS.GWTSNM 074C E180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TNTR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TNTR[4:0]	Timestamp Number in Timestamp RAM		R
b31 to b5	—	Reserved	These bits are read as 0.	R

TNTR[4:0] Bits (Timestamp Number in Timestamp RAM)

[Increment condition]

- HW: Incremented by 1 when a timestamp is received from an Ethernet Agent.

[Decrement condition]

- HW: Decremented by 1 when a timestamp is read from timestamp RAM.

38.2.6.5 Timestamp Maximum Number Monitoring Register (GWTSMNM)

Address(es): GWCA0.GWTSMNM 874C E184h, GWCA0_NS.GWTSMNM 074C E184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TMNTR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TMNTR[4:0]	Timestamp Maximum Number in Timestamp RAM		R
b31 to b5	—	Reserved	These bits are read as 0.	R

TMNTR[4:0] Bits (Timestamp Maximum Number in Timestamp RAM)

[Update condition]

- HW: Set to GWTSNM.TNTR[4:0] when GWTSNM.TNTR[4:0] > GWTSNM.TMNTR[4:0].

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clear it to GWTSNM.TNTR[4:0] value.

38.2.7 AXI Master Registers

38.2.7.1 AXI Control Register (GWAC)

Address(es): GWCA0.GWAC 874C E190h, GWCA0_NS.GWAC 074C E190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AMP	AMPR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AMPR	AXI Master Pause Request	0: AXI master pause not requested 1: AXI master pause requested	R/W
b1	AMP	AXI Master Paused	0: AXI master not paused (transaction ongoing) 1: AXI master paused (no transaction ongoing)	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

AMPR Bit (AXI Master Pause Request)

This register blocks the AXI master from making new accesses (because the AXI master has register slices for timing purpose few AXI accesses can be emitted before stopping).

AMP Bit (AXI Master Paused)

“AXI Master Pause” stop only the final processing of the AXI interface. (Internal processing cannot be stopped.) For example, the AXI address for chain [i] is loaded immediately before “AXI Master Paused”. This address will used after “AXI Master not paused” regardless of AXI address re-configuration (by GWDCCi) during “AXI Master Pause”.

[Setting condition]

- HW: When GWAC.AMPR is set to 1 and all ongoing AXI transactions are completed, this bit will be set.

[Clearing condition]

- HW: When GWAC.AMPR is set to 0, this bit will be set to 0 (it can take time because of asynchronous modules).

38.2.7.2 Descriptor Chain Base Address Configuration Register 0 (GWDCBAC0)

Address(es): GWCA0.GWDCBAC0 874C E194h, GWCA0_NS.GWDCBAC0 074C E194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DCBAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DCBAU[7:0]	Descriptor Chain Base Address Upper Part		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

DCBAU[7:0] Bits (Descriptor Chain Base Address Upper Part)

For further setting information, refer to (1) LINKFIX Table Initialization in section 38.3.1.3, LINKFIX Table.

38.2.7.3 Descriptor Chain Base Address Configuration Register 1 (GWDCBAC1)

Address(es): GWCA0.GWDCBAC1 874C E198h, GWCA0_NS.GWDCBAC1 074C E198h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DCBAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DCBAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DCBAL[31:0]	Descriptor Chain Base Address Lower Part		R/W

DCBAL[30:0] Bits (Descriptor Chain Base Address Lower Part)

For further setting information, refer to (1) LINKFIX Table Initialization in section 38.3.1.3, LINKFIX Table.

38.2.7.4 Maximum Descriptor Number Configuration Register (GWMDNC)

Address(es): GWCA0.GWMDNC 874C E1A0h, GWCA0_NS.GWMDNC 074C E1A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDMN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TXDMN[4:0]					—	—	—	RXDMN[4:0]				
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RXDMN[4:0]	RX Descriptor Maximum Number	Configures the maximum number of descriptors that can be processed at once for an RX queue (it limits to GWMDNC.RXDMN[4:0] + 1 the number of descriptors read that can be done for an RX frame). Refer to the GWEIS5.RXDNES flag explanation.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	TXDMN[4:0]	TX Descriptor Maximum Number	Configures the maximum number of descriptors that can be processed at once for a TX queue (it limits to GWMDNC.TXDMN[4:0] + 1 the number of descriptors read that can be done for a TX frame). Refer to the GWEIS0.TXDNES flag explanation.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b17, b16	TSDMN[1:0]	Timestamp Descriptor Maximum Number	Configures the maximum number of descriptors that can be processed at once for a TS queue (it limits to GWMDNC.TSDMN[1:0] + 1 the number of descriptors read that can be done for a timestamp). Refer to the GWEIS0.TSDNESi (i = 0, 1) flag explanation.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

RXDMN[4:0] Bits (RX Descriptor Maximum Number)

[Caution]

- SW: This maximum descriptor number takes in account all descriptors including LINK, LINKFIX descriptors and descriptors remaining from a previous error frame like FMID_EMPTY and FEND_EMPTY descriptors. It should be taken in account while setting this register.

TXDMN[4:0] Bits (TX Descriptor Maximum Number)

[Restriction]

- SW: The switch maximum input frame size is 60 Kbytes so this register should be set to a value less than or equal to 30. To avoid any unexpected frame loss due to this register, any software sending frames larger than 58 Kbytes should avoid having more than one consecutive LINK/LINKFIX descriptor in a descriptor chain.

[Caution]

- SW: This maximum descriptor number takes in account all descriptors including LINK, LINKFIX descriptors and descriptors remaining from a previous error frame like FMID and FEND descriptors. It should be taken in account while setting this register.

38.2.7.5 Transmission Request Configuration Register i (GWTRCi) (i = 0, 1)

- GWCA0.GWTRC0

Address(es): GWCA0.GWTRC0 874C E200h, GWCA0_NS.GWTRC0 074C E200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSR31	TSR30	TSR29	TSR28	TSR27	TSR26	TSR25	TSR24	TSR23	TSR22	TSR21	TSR20	TSR19	TSR18	TSR17	TSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSR0	Transmission Start Request 0	0: TX Descriptor Chain is empty. 1: Transmission request is ongoing.	R/W
b1	TSR1	Transmission Start Request 1		R/W
b2	TSR2	Transmission Start Request 2		R/W
b3	TSR3	Transmission Start Request 3		R/W
b4	TSR4	Transmission Start Request 4		R/W
b5	TSR5	Transmission Start Request 5		R/W
b6	TSR6	Transmission Start Request 6		R/W
b7	TSR7	Transmission Start Request 7		R/W
b8	TSR8	Transmission Start Request 8		R/W
b9	TSR9	Transmission Start Request 9		R/W
b10	TSR10	Transmission Start Request 10		R/W
b11	TSR11	Transmission Start Request 11		R/W
b12	TSR12	Transmission Start Request 12		R/W
b13	TSR13	Transmission Start Request 13		R/W
b14	TSR14	Transmission Start Request 14		R/W
b15	TSR15	Transmission Start Request 15		R/W
b16	TSR16	Transmission Start Request 16		R/W
b17	TSR17	Transmission Start Request 17		R/W
b18	TSR18	Transmission Start Request 18		R/W
b19	TSR19	Transmission Start Request 19		R/W
b20	TSR20	Transmission Start Request 20		R/W
b21	TSR21	Transmission Start Request 21		R/W
b22	TSR22	Transmission Start Request 22		R/W
b23	TSR23	Transmission Start Request 23		R/W
b24	TSR24	Transmission Start Request 24		R/W
b25	TSR25	Transmission Start Request 25		R/W
b26	TSR26	Transmission Start Request 26		R/W
b27	TSR27	Transmission Start Request 27		R/W
b28	TSR28	Transmission Start Request 28		R/W
b29	TSR29	Transmission Start Request 29		R/W
b30	TSR30	Transmission Start Request 30		R/W
b31	TSR31	Transmission Start Request 31		R/W

- GWCA0.GWTRC1

Address(es): GWCA0.GWTRC1 874C E204h, GWCA0_NS.GWTRC1 074C E204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSR63	TSR62	TSR61	TSR60	TSR59	TSR58	TSR57	TSR56	TSR55	TSR54	TSR53	TSR52	TSR51	TSR50	TSR49	TSR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSR47	TSR46	TSR45	TSR44	TSR43	TSR42	TSR41	TSR40	TSR39	TSR38	TSR37	TSR36	TSR35	TSR34	TSR33	TSR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSR32	Transmission Start Request 32	0: TX Descriptor Chain is empty. 1: Transmission request is ongoing.	R/W
b1	TSR33	Transmission Start Request 33		R/W
b2	TSR34	Transmission Start Request 34		R/W
b3	TSR35	Transmission Start Request 35		R/W
b4	TSR36	Transmission Start Request 36		R/W
b5	TSR37	Transmission Start Request 37		R/W
b6	TSR38	Transmission Start Request 38		R/W
b7	TSR39	Transmission Start Request 39		R/W
b8	TSR40	Transmission Start Request 40		R/W
b9	TSR41	Transmission Start Request 41		R/W
b10	TSR42	Transmission Start Request 42		R/W
b11	TSR43	Transmission Start Request 43		R/W
b12	TSR44	Transmission Start Request 44		R/W
b13	TSR45	Transmission Start Request 45		R/W
b14	TSR46	Transmission Start Request 46		R/W
b15	TSR47	Transmission Start Request 47		R/W
b16	TSR48	Transmission Start Request 48		R/W
b17	TSR49	Transmission Start Request 49		R/W
b18	TSR50	Transmission Start Request 50		R/W
b19	TSR51	Transmission Start Request 51		R/W
b20	TSR52	Transmission Start Request 52		R/W
b21	TSR53	Transmission Start Request 53		R/W
b22	TSR54	Transmission Start Request 54		R/W
b23	TSR55	Transmission Start Request 55		R/W
b24	TSR56	Transmission Start Request 56		R/W
b25	TSR57	Transmission Start Request 57		R/W
b26	TSR58	Transmission Start Request 58		R/W
b27	TSR59	Transmission Start Request 59		R/W
b28	TSR60	Transmission Start Request 60		R/W
b29	TSR61	Transmission Start Request 61		R/W
b30	TSR62	Transmission Start Request 62		R/W
b31	TSR63	Transmission Start Request 63		R/W

TSRt Bit (Transmission Start Request t) (t = 0 to 63)

This bit is used to start transfer from CPU to switch for its corresponding TX queue.

[Setting conditions (only for TX queues)]

- SW: Writing 1 to this bit will set this bit set if queue t is a transmission queue (GWDCCi.DQT is set).
- HW: A transmission request received through INTER_TX[t] interface will set this bit if queue t is a transmission queue (GWDCCi.DQT is set).

[Clearing conditions]

- HW: When HW read a descriptor other than FSINGLE, FSTART, FMID, FEND, LINK and LINKFIX.
- HW: When an AXI error occurs while reading a descriptor in the corresponding AXI descriptor queue.
- When a descriptor number error occurs while reading the corresponding AXI descriptor queue.
- HW: Going out of OPERATION mode will clear this register.

[Restriction]

- HW: For RX queues, this bit cannot be set.

38.2.7.6 Transmission Pause Configuration Register p (GWTPCp) (p = 0, 1)

Address(es): GWCA0.GWTPC0 874C E300h, GWCA0.GWTPC1 874C E304h,
GWCA0_NS.GWTPC0 074C E300h, GWCA0_NS.GWTPC1 074C E304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PPPL7	PPPL6	PPPL5	PPPL4	PPPL3	PPPL2	PPPL1	PPPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PPPL0	Per Priority Pause Level 0	0: Transmission queues with their priority GWDCCi.DCP[2:0] set to i are not paused when pause level p is set (PAUSE[2 for GWCA0][p] is set [COMA]) 1: Transmission queues with the priority GWDCCi.DCP[2:0] set to i are paused when pause level p is set (PAUSE[2 for GWCA0][p] is set [COMA]) (i = 0 to 7) (p = 0, 1)	R/W
b1	PPPL1	Per Priority Pause Level 1		R/W
b2	PPPL2	Per Priority Pause Level 2		R/W
b3	PPPL3	Per Priority Pause Level 3		R/W
b4	PPPL4	Per Priority Pause Level 4		R/W
b5	PPPL5	Per Priority Pause Level 5		R/W
b6	PPPL6	Per Priority Pause Level 6		R/W
b7	PPPL7	Per Priority Pause Level 7		R/W
b31 to b8	—	Reserved	This bit is read as 0. The write value should be 0.	R

38.2.7.7 AXI RAM Initialization Register Monitoring Register (GWARIRM)

Address(es): GWCA0.GWARIRM 874C E380h, GWCA0_NS.GWARIRM 074C E380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARR	ARIOG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ARIOG	AXI RAM Initialization Ongoing		R/W
b1	ARR	AXI RAM Ready		R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

ARIOG Bit (AXI RAM Initialization Ongoing)

[Setting condition]

- SW: By writing 1 to this bit. It starts AXI RAM initialization.

[Clearing condition]

- HW: This bit is cleared when AXI RAM initialization is finished.
  Min clk_period × 64 = [exp] 4 × 64 = 256 ns

ARR Bit (AXI RAM Ready)

[Setting condition]

- When GWARIRM.ARIOG is getting cleared.

[Clearing condition]

- By writing 1 to GWARIRM.ARIOG.

38.2.7.8 Descriptor Chain i Configuration Register (GWDCCi) (i = 0 to 63)

Address(es): GWCA0.GWDCC0 874C E400h, GWCA0.GWDCC1 874C E404h, GWCA0.GWDCC2 874C E408h, GWCA0.GWDCC3 874C E40Ch, GWCA0.GWDCC4 874C E410h, GWCA0.GWDCC5 874C E414h, GWCA0.GWDCC6 874C E418h, GWCA0.GWDCC7 874C E41Ch, GWCA0.GWDCC8 874C E420h, GWCA0.GWDCC9 874C E424h, GWCA0.GWDCC10 874C E428h, GWCA0.GWDCC11 874C E42Ch, GWCA0.GWDCC12 874C E430h, GWCA0.GWDCC13 874C E434h, GWCA0.GWDCC14 874C E438h, GWCA0.GWDCC15 874C E43Ch, GWCA0.GWDCC16 874C E440h, GWCA0.GWDCC17 874C E444h, GWCA0.GWDCC18 874C E448h, GWCA0.GWDCC19 874C E44Ch, GWCA0.GWDCC20 874C E450h, GWCA0.GWDCC21 874C E454h, GWCA0.GWDCC22 874C E458h, GWCA0.GWDCC23 874C E45Ch, GWCA0.GWDCC24 874C E460h, GWCA0.GWDCC25 874C E464h, GWCA0.GWDCC26 874C E468h, GWCA0.GWDCC27 874C E46Ch, GWCA0.GWDCC28 874C E470h, GWCA0.GWDCC29 874C E474h, GWCA0.GWDCC30 874C E478h, GWCA0.GWDCC31 874C E47Ch, GWCA0.GWDCC32 874C E480h, GWCA0.GWDCC33 874C E484h, GWCA0.GWDCC34 874C E488h, GWCA0.GWDCC35 874C E48Ch, GWCA0.GWDCC36 874C E490h, GWCA0.GWDCC37 874C E494h, GWCA0.GWDCC38 874C E498h, GWCA0.GWDCC39 874C E49Ch, GWCA0.GWDCC40 874C E4A0h, GWCA0.GWDCC41 874C E4A4h, GWCA0.GWDCC42 874C E4A8h, GWCA0.GWDCC43 874C E4ACh, GWCA0.GWDCC44 874C E4B0h, GWCA0.GWDCC45 874C E4B4h, GWCA0.GWDCC46 874C E4B8h, GWCA0.GWDCC47 874C E4BCh, GWCA0.GWDCC48 874C E4C0h, GWCA0.GWDCC49 874C E4C4h, GWCA0.GWDCC50 874C E4C8h, GWCA0.GWDCC51 874C E4CCh, GWCA0.GWDCC52 874C E4D0h, GWCA0.GWDCC53 874C E4D4h, GWCA0.GWDCC54 874C E4D8h, GWCA0.GWDCC55 874C E4DCh, GWCA0.GWDCC56 874C E4E0h, GWCA0.GWDCC57 874C E4E4h, GWCA0.GWDCC58 874C E4E8h, GWCA0.GWDCC59 874C E4ECh, GWCA0.GWDCC60 874C E4F0h, GWCA0.GWDCC61 874C E4F4h, GWCA0.GWDCC62 874C E4F8h, GWCA0.GWDCC63 874C E4FCh,
GWCA0_NS.GWDCC0 074C E400h, GWCA0_NS.GWDCC1 074C E404h, GWCA0_NS.GWDCC2 074C E408h, GWCA0_NS.GWDCC3 074C E40Ch, GWCA0_NS.GWDCC4 074C E410h, GWCA0_NS.GWDCC5 074C E414h, GWCA0_NS.GWDCC6 074C E418h, GWCA0_NS.GWDCC7 074C E41Ch, GWCA0_NS.GWDCC8 074C E420h, GWCA0_NS.GWDCC9 074C E424h, GWCA0_NS.GWDCC10 074C E428h, GWCA0_NS.GWDCC11 074C E42Ch, GWCA0_NS.GWDCC12 074C E430h, GWCA0_NS.GWDCC13 074C E434h, GWCA0_NS.GWDCC14 074C E438h, GWCA0_NS.GWDCC15 074C E43Ch, GWCA0_NS.GWDCC16 074C E440h, GWCA0_NS.GWDCC17 074C E444h, GWCA0_NS.GWDCC18 074C E448h, GWCA0_NS.GWDCC19 074C E44Ch, GWCA0_NS.GWDCC20 074C E450h, GWCA0_NS.GWDCC21 074C E454h, GWCA0_NS.GWDCC22 074C E458h, GWCA0_NS.GWDCC23 074C E45Ch, GWCA0_NS.GWDCC24 074C E460h, GWCA0_NS.GWDCC25 074C E464h, GWCA0_NS.GWDCC26 074C E468h, GWCA0_NS.GWDCC27 074C E46Ch, GWCA0_NS.GWDCC28 074C E470h, GWCA0_NS.GWDCC29 074C E474h, GWCA0_NS.GWDCC30 074C E478h, GWCA0_NS.GWDCC31 074C E47Ch, GWCA0_NS.GWDCC32 074C E480h, GWCA0_NS.GWDCC33 074C E484h, GWCA0_NS.GWDCC34 074C E488h, GWCA0_NS.GWDCC35 074C E48Ch, GWCA0_NS.GWDCC36 074C E490h, GWCA0_NS.GWDCC37 074C E494h, GWCA0_NS.GWDCC38 074C E498h, GWCA0_NS.GWDCC39 074C E49Ch, GWCA0_NS.GWDCC40 074C E4A0h, GWCA0_NS.GWDCC41 074C E4A4h, GWCA0_NS.GWDCC42 074C E4A8h, GWCA0_NS.GWDCC43 074C E4ACh, GWCA0_NS.GWDCC44 074C E4B0h, GWCA0_NS.GWDCC45 074C E4B4h, GWCA0_NS.GWDCC46 074C E4B8h, GWCA0_NS.GWDCC47 074C E4BCh, GWCA0_NS.GWDCC48 074C E4C0h, GWCA0_NS.GWDCC49 074C E4C4h, GWCA0_NS.GWDCC50 074C E4C8h, GWCA0_NS.GWDCC51 074C E4CCh, GWCA0_NS.GWDCC52 074C E4D0h, GWCA0_NS.GWDCC53 074C E4D4h, GWCA0_NS.GWDCC54 074C E4D8h, GWCA0_NS.GWDCC55 074C E4DCh, GWCA0_NS.GWDCC56 074C E4E0h, GWCA0_NS.GWDCC57 074C E4E4h, GWCA0_NS.GWDCC58 074C E4E8h, GWCA0_NS.GWDCC59 074C E4ECh, GWCA0_NS.GWDCC60 074C E4F0h, GWCA0_NS.GWDCC61 074C E4F4h, GWCA0_NS.GWDCC62 074C E4F8h, GWCA0_NS.GWDCC63 074C E4FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	OSID[2:0]			—	—	—	BALR	—	—	—	—	—	DCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	DQT	SL	ETS	EDE	—	—	—	—	—	—	SM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SM[1:0]	Synchronization Mode	b1 b0 0 0: Normal mode (full descriptor write back) 0 1: No-write-back mode (no descriptor write back) 1 0: Keep-DT mode (no update of DT field at descriptor write back) 1 1: Reserved	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	EDE	Extended Descriptor Enable	0: Basic Descriptor 1: Extended Descriptor	R/W
b9	ETS	Enable Timestamp Storage	0: Timestamp is not added in the Descriptor. 1: Timestamp is added in the Descriptor.	R/W
b10	SL	Security Level	0: Chain i unsecure 1: Chain i secure	R/W
b11	DQT	Descriptor Queue Type	0: Descriptor queue i is a reception queue. 1: Descriptor queue i is a transmission queue.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b18 to b16	DCP[2:0]	Descriptor Chain Priority	b18 b16 0 0 0: lowest priority : 1 1 1: highest priority	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	BALR	Base Address Load Request	This bit is used to reset current_address field in the AXI address RAM to {{GWDCBAC0.DCBAU[7:0], GWDCBAC1.DCBAL[31:0]} + i*8}	R/W
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30 to b28	OSID[2:0]	OS ID	When a memory access is done for descriptor queue number i the OSID set in this register will be used.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

ETS Bit (Enable Timestamp Storage)

[Restriction]

- HW: This bit is only valid for RX queues.

SL Bit (Security Level)

When a chain is secured, an unsecure descriptor cannot enter it (when a descriptor is received from forwarding engine [MFWD] with FDESCR.SEC equal to 0).

[Restriction]

- HW: This bit is only valid for RX queues.

DCP[2:0] Bits (Descriptor Chain Priority)

[Restrictions]

- HW: These bits are only valid for TX queues.
- SW: These bits value should not be changed for a TX queue when GWTRCi.TSRj is set. (When transmission is ongoing for the corresponding queue.)

BALR Bit (Base Address Load Request)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- HW: This bit will be cleared when the current_address field in the AXI address RAM is reset.

38.2.7.9 AXI Address RAM Searching Setting Register (GWAARSS)

Address(es): GWCA0.GWAARSS 874C E800h, GWCA0_NS.GWAARSS 074C E800h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	AARA[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	AARA[5:0]	AXI Address RAM Address	This register is used to read the current address used in AXI master for GWAARSS.AARA[5:0] descriptor queue.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

AARA[5:0] Bits (AXI Address RAM Address)

[Restriction]

- HW: Due to the pipeline architecture of the AXI master, the AXI read can just give an idea of which address is processing the AXI master and is not precise enough to be used for HW/SW synchronization. This register is only here for debug purpose.

38.2.7.10 AXI Address RAM Searching Result Register 0 (GWAARSR0)

Address(es): GWCA0.GWAARSR0 874C E804h, GWCA0_NS.GWAARSR0 074C E804h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AARS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ACARU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ACARU[7:0]	AXI Current Address Result Upper Part	Displays AXI address read value.	R
b30 to b8	—	Reserved	These bits are read as 0.	R
b31	AARS	AXI Address RAM Searching	This bit is used to read an AXI address RAM.	R

ACARU[7:0] Bits (AXI Current Address Result Upper Part)

[Update condition]

- HW: GWAARSR0.AARS clear event.

Note: AXI address read will not restart from the base address of chain after RESET mode (because being in RESET mode will not clear this register).

AARS Bit (AXI Address RAM Searching)

[Setting condition]

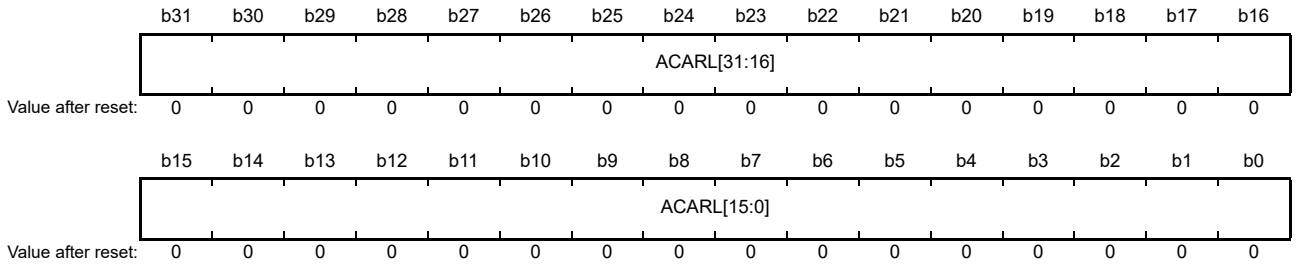
- SW: Writing GWAARSS will set this bit.

[Clearing condition]

- HW: This bit will be de-asserted when the AXI address RAM search is completed.

38.2.7.11 AXI Address RAM Searching Result Register 1 (GWAARSR1)

Address(es): GWCA0.GWAARSR1 874C E808h, GWCA0_NS.GWAARSR1 074C E808h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	ACARL[31:0]	AXI Current Address Result Lower Part	Displays AXI address read value.	R

ACARL[31:0] Bits (AXI Current Address Result Lower Part)

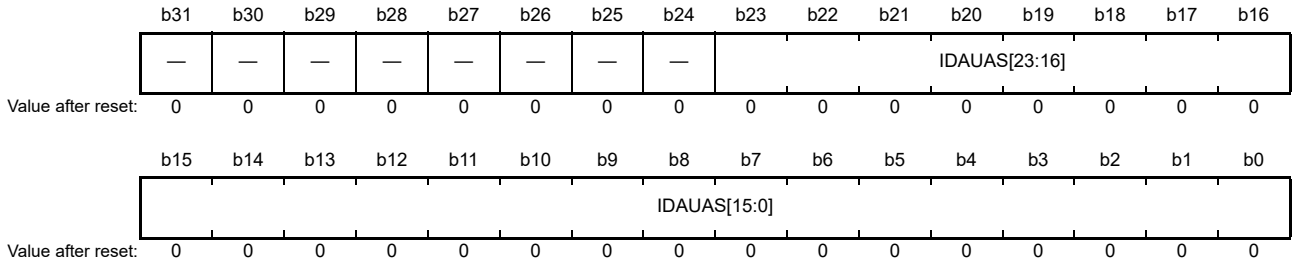
[Update condition]

- HW: GWAARSR0.AARS clear event.

Note: AXI address read will not restart from the base address of chain after RESET mode (because being in RESET mode will not clear this register).

38.2.7.12 Incremental Data Area i Used Area Size Register (GWIDAUASi) (i = 0 to 3)

Address(es): GWCA0.GWIDAUAS0 874C E840h, GWCA0.GWIDAUAS1 874C E844h, GWCA0.GWIDAUAS2 874C E848h, GWCA0.GWIDAUAS3 874C E84Ch,
GWCA0_NS.GWIDAUAS0 074C E840h, GWCA0_NS.GWIDAUAS1 074C E844h, GWCA0_NS.GWIDAUAS2 074C E848h, GWCA0_NS.GWIDAUAS3 074C E84Ch



Bit	Symbol	Bit Name	Description	R/W
b23 to b0	IDAUAS[23:0]	Incremental Data Area Used Size of RX Descriptor Chain i	Represents the byte numbers that have been used in the corresponding incremental area.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

IDAUAS[23:0] Bits (Incremental Data Area Used Size of RX Descriptor Chain i)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, these bits get cleared.

[Increment condition]

- HW: When a frame has been stored in the corresponding incremental area, these bits are incremented by the frame size.

[Decrement condition]

- SW: By writing a value, SW will decrement these bits by this value.

[Restriction]

- SW: These bits should always write a value smaller than the current register value.

38.2.7.13 Incremental Data Area i Size Monitoring Register (GWIDASMi) (i = 0 to 3)

Address(es): GWCA0.GWIDASM0 874C E880h, GWCA0.GWIDASM1 874C E884h, GWCA0.GWIDASM2 874C E888h, GWCA0.GWIDASM3 874C E88Ch,
GWCA0_NS.GWIDASM0 074C E880h, GWCA0_NS.GWIDASM1 074C E884h, GWCA0_NS.GWIDASM2 074C E888h, GWCA0_NS.GWIDASM3 074C E88Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IDAS[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDAS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	IDAS[23:0]	Incremental Data Area Size of RX Descriptor Chain i	This register indicates the size of RX descriptor chain i incremental area in bytes.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

IDAS[23:0] Bits (Incremental Data Area Size of RX Descriptor Chain i)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update condition]

- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, these bits are update to DESCR.DS*4096.

38.2.7.14 Incremental Data Area i Start Address Monitoring Register 0 (GWIDASAMi0) (i = 0 to 3)

Address(es): GWCA0.GWIDASAM00 874C E900h, GWCA0.GWIDASAM10 874C E908h, GWCA0.GWIDASAM20 874C E910h, GWCA0.GWIDASAM30 874C E918h,
GWCA0_NS.GWIDASAM00 074C E900h, GWCA0_NS.GWIDASAM10 074C E908h, GWCA0_NS.GWIDASAM20 074C E910h, GWCA0_NS.GWIDASAM30 074C E918h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	IDASAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IDASAU[7:0]	Incremental Data Area Start Address Upper Part of RX Descriptor Chain i		R
b31 to b8	—	Reserved	These bits are read as 0.	R

IDASAU[7:0] Bits (Incremental Data Area Start Address Upper Part of RX Descriptor Chain i)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update condition]

- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, these bits are update to DESCR.PTR[39:32].

38.2.7.15 Incremental Data Area i Start Address Monitoring Register 1 (GWIDASAMi1) (i = 0 to 3)

Address(es): GWCA0.GWIDASAM01 874C E904h, GWCA0.GWIDASAM11 874C E90Ch, GWCA0.GWIDASAM21 874C E914h, GWCA0.GWIDASAM31 874C E91Ch, GWCA0_NS.GWIDASAM01 074C E904h, GWCA0_NS.GWIDASAM11 074C E90Ch, GWCA0_NS.GWIDASAM21 074C E914h, GWCA0_NS.GWIDASAM31 074C E91Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDASAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDASAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	IDASAL[31:0]	Incremental Data Area Start Address Lower Part of RX Descriptor Chain i		R

IDASAL[31:0] Bits (Incremental Data Area Start Address Lower Part of RX Descriptor Chain i)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update condition]

- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, these bits are update to DESCR.PTR[31:0].

38.2.7.16 Incremental Data Area i Current Address Monitoring Register 0 (GWIDACAMi0) (i = 0 to 3)

Address(es): GWCA0.GWIDACAM00 874C E980h, GWCA0.GWIDACAM10 874C E988h, GWCA0.GWIDACAM20 874C E990h, GWCA0.GWIDACAM30 874C E998h,
GWCA0_NS.GWIDACAM00 074C E980h, GWCA0_NS.GWIDACAM10 074C E988h, GWCA0_NS.GWIDACAM20 074C E990h, GWCA0_NS.GWIDACAM30 074C E998h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	IDACAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IDACAU[7:0]	Incremental Data Area Current Address Upper Part of RX Descriptor Chain i		R
b31 to b8	—	Reserved	These bits are read as 0.	R

IDACAU[7:0] Bits (Incremental Data Area Current Address Upper Part of RX Descriptor Chain i)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update conditions]

- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, these bits are update to DESCR.PTR[39:32].
- HW: When a burst is decided to be sent in the incremental area, {GWIDACAMi0.IDACAU[7:0], GWIDACAMi1.IDACAL[31:0]} is incremented by the burst size.
- HW: {GWIDACAMi0.IDACAU[7:0], GWIDACAMi1.IDACAL[31:0]} overflows at {GWIDASAMi0.IDASAU[7:0], GWIDASAMi1.IDASAL[31:0]} + GWIDASMi.IDAS[23:0] – 1.

38.2.7.17 Incremental Data Area i Current Address Monitoring Register 1 (GWIDACAMi1) (i = 0 to 3)

Address(es): GWCA0.GWIDACAM01 874C E984h, GWCA0.GWIDACAM11 874C E98Ch, GWCA0.GWIDACAM21 874C E994h, GWCA0.GWIDACAM31 874C E99Ch,
GWCA0_NS.GWIDACAM01 074C E984h, GWCA0_NS.GWIDACAM11 074C E98Ch, GWCA0_NS.GWIDACAM21 074C E994h, GWCA0_NS.GWIDACAM31 074C E99Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDACAL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDACAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	IDACAL[31:0]	Incremental Data Area Current Address Lower Part of RX Descriptor Chain i		R

IDACAL[31:0] Bits (Incremental Data Area Current Address Lower Part of RX Descriptor Chain i)

[Clearing condition]

- HW: Being in RESET mode will clear this register.

[Update conditions]

- HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is update to DESCR.PTR[31:0].
- HW: When a burst is decided to be sent in the incremental area, {GWIDACAMi0.IDACAU[7:0], GWIDACAMi1.IDACAL[31:0]} is incremented by the burst size.
- HW: {GWIDACAMi0.IDACAU[7:0], GWIDACAMi1.IDACAL[31:0]} overflows at {GWIDASAMi0.IDASAU[7:0], GWIDASAMi1.IDASAL[31:0]} + GWIDASMi.IDAS[23:0] – 1.

38.2.8 Rate Limiter Function Registers

38.2.8.1 Global Rate Limiter Configuration Register (GWGRLC)

Address(es): GWCA0.GWGRLC 874C EA00h, GWCA0_NS.GWGRLC 074C EA00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GRLULRS	GRLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GRLIV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	GRLIV[15:0]	Global Rate Limiter Incremental Value		R/W
b16	GRLE	Global Rate Limiter Enable	0: Global Rate limiter disabled 1: Global Rate limiter enabled	R/W
b17	GRLULRS	Global Rate Limiter Upper Limit Reached Status Flag		R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

GRLIV[15:0] Bits (Global Rate Limiter Incremental Value)

For further setting information refer to (2) Rate Limiters in section 38.3.2.1, TX Queue Arbitration.

[Caution]

- SW: Setting this register to 0 or a value close to 0 would set GWCA to a very low throughput or could stuck GWCA in transmission.

GRLE Bit (Global Rate Limiter Enable)

The global rate limiter limits the throughput through all TX queues to avoid the switch to take in too much data at once.

GRLULRS Flag (Global Rate Limiter Upper Limit Reached Status Flag)

If this flag gets set when only the global rate limiter is enabled (per port rate limiters disabled), it means that the AXI master is too slow to provide the throughput set in the global rate limiter or that the AXI latency has been underestimated.

This flag can get set while using per port rate limiters. In this case it should be ignored.

[Setting condition]

- HW: When the global rate limit reaches its upper limit GWGRLULC.GRLUL[23:0], this flag gets set.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it

38.2.8.2 Global Rate Limiter Upper Limit Configuration Register (GWGRLULC)

Address(es): GWCA0.GWGRLULC 874C EA04h, GWCA0_NS.GWGRLULC 074C EA04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	GRLUL[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GRLUL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	GRLUL[23:0]	Global Rate Limiter Upper Limit	Set the maximum number of credits in bit rate limiter that can accumulate before stopping.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

GRLUL[23:0] Bits (Global Rate Limiter Upper Limit)

For further setting information refer to (2) Rate Limiters in section 38.3.2.1, TX Queue Arbitration.

[Restriction]

- SW: These bits should not be set to a value bigger than 800000h.

38.2.8.3 Rate Limiter i Configuration Register (GWRLCi) (i = 0 to 7)

Address(es): GWCA0.GWRLC0 874C EA80h, GWCA0.GWRLC1 874C EA88h, GWCA0.GWRLC2 874C EA90h, GWCA0.GWRLC3 874C EA98h, GWCA0.GWRLC4 874C EAA0h, GWCA0.GWRLC5 874C EAA8h, GWCA0.GWRLC6 874C EAB0h, GWCA0.GWRLC7 874C EAB8h, GWCA0_NS.GWRLC0 074C EA80h, GWCA0_NS.GWRLC1 074C EA88h, GWCA0_NS.GWRLC2 074C EA90h, GWCA0_NS.GWRLC3 074C EA98h, GWCA0_NS.GWRLC4 074C EAA0h, GWCA0_NS.GWRLC5 074C EAA8h, GWCA0_NS.GWRLC6 074C EAB0h, GWCA0_NS.GWRLC7 074C EAB8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RLIV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RLIV[11:0]	Rate Limiter Incremental Value		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RLE	Rate Limiter Enable	0: Rate limiter i disabled 1: Rate limiter i enabled	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

GRLUL[23:0] Bits (Rate Limiter Incremental Value)

For further setting information refer to (2) Rate Limiters in section 38.3.2.1, TX Queue Arbitration.

[Caution]

- SW: Setting these bits to 0 or a value close to 0 would set corresponding queue to a very low throughput or could stuck the corresponding queue.

RLE Bit (Rate Limiter Enable)

Rate limiter i limit the through put for AXI descriptor queue number 63-I. For details, refer to section 38.3.1.1, AXI Descriptor Queue Mapping.

38.2.8.4 Rate Limiter i Upper Limit Configuration Register (GWRLULCi) (i = 0 to 7)

Address(es): GWCA0.GWRLULC0 874C EA84h, GWCA0.GWRLULC1 874C EA8Ch, GWCA0.GWRLULC2 874C EA94h, GWCA0.GWRLULC3 874C EA9Ch, GWCA0.GWRLULC4 874C EAA4h, GWCA0.GWRLULC5 874C EAACh, GWCA0.GWRLULC6 874C EAB4h, GWCA0.GWRLULC7 874C EABCh, GWCA0_NS.GWRLULC0 074C EA84h, GWCA0_NS.GWRLULC1 074C EA8Ch, GWCA0_NS.GWRLULC2 074C EA94h, GWCA0_NS.GWRLULC3 074C EA9Ch, GWCA0_NS.GWRLULC4 074C EAA4h, GWCA0_NS.GWRLULC5 074C EAACh, GWCA0_NS.GWRLULC6 074C EAB4h, GWCA0_NS.GWRLULC7 074C EABCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	RLUL[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RLUL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	RLUL[23:0]	Rate Limiter Upper Limit	Set the maximum number of credits in bit rate limiter i can accumulate before stopping.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

RLUL[23:0] Bits (Rate Limiter Upper Limit)

These bits can be used if a lot of traffic happens on the AXI bus so burst on queues i make the switch overflow. If not required, set this register to 800000h.

For further setting information refer to (2) Rate Limiters in section 38.3.2.1, TX Queue Arbitration.

[Restriction]

- SW: These bits should not be set to a value bigger than 800000h.

38.2.9 Interrupt Function Registers

38.2.9.1 Interrupt Delay Prescaler Configuration Register (GWIDPC)

Address(es): GWCA0.GWIDPC 874C EB80h, GWCA0_NS.GWIDPC 074C EB80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IDPV[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

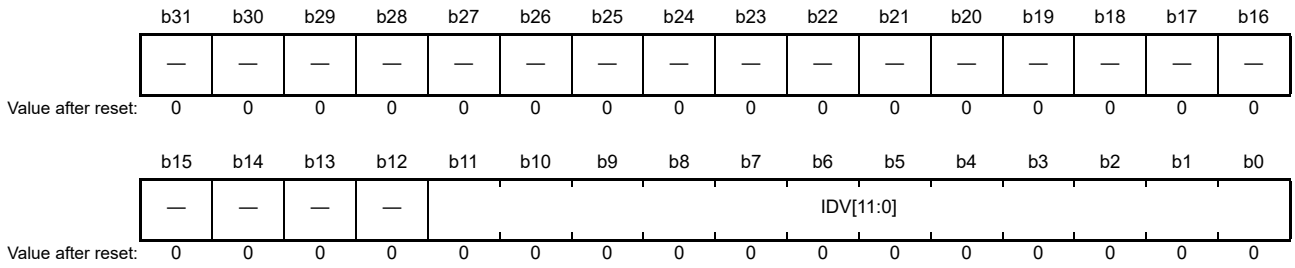
Bit	Symbol	Bit Name	Description	R/W
b9 to b0	IDPV[9:0]	Interrupt Delay Prescaler Value	This register is used to create an internal clock.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

IDPV[9:0] Bits (Interrupt Delay Prescaler Value)

This internal clock period will be clk_delay_period = clk_period*(GWIDPC.IDPV[9:0] + 1) where clk_period is the period of clk. For further information, refer to section 38.3.6, Interrupt Delay Function.

38.2.9.2 Interrupt Delay Configuration Register (GWIDCi) (i = 0 to 63)

Address(es): GWCA0.GWIDC0 874C EC00h, GWCA0.GWIDC1 874C EC04h, GWCA0.GWIDC2 874C EC08h, GWCA0.GWIDC3 874C EC0Ch, GWCA0.GWIDC4 874C EC10h, GWCA0.GWIDC5 874C EC14h, GWCA0.GWIDC6 874C EC18h, GWCA0.GWIDC7 874C EC1Ch, GWCA0.GWIDC8 874C EC20h, GWCA0.GWIDC9 874C EC24h, GWCA0.GWIDC10 874C EC28h, GWCA0.GWIDC11 874C EC2Ch, GWCA0.GWIDC12 874C EC30h, GWCA0.GWIDC13 874C EC34h, GWCA0.GWIDC14 874C EC38h, GWCA0.GWIDC15 874C EC3Ch, GWCA0.GWIDC16 874C EC40h, GWCA0.GWIDC17 874C EC44h, GWCA0.GWIDC18 874C EC48h, GWCA0.GWIDC19 874C EC4Ch, GWCA0.GWIDC20 874C EC50h, GWCA0.GWIDC21 874C EC54h, GWCA0.GWIDC22 874C EC58h, GWCA0.GWIDC23 874C EC5Ch, GWCA0.GWIDC24 874C EC60h, GWCA0.GWIDC25 874C EC64h, GWCA0.GWIDC26 874C EC68h, GWCA0.GWIDC27 874C EC6Ch, GWCA0.GWIDC28 874C EC70h, GWCA0.GWIDC29 874C EC74h, GWCA0.GWIDC30 874C EC78h, GWCA0.GWIDC31 874C EC7Ch, GWCA0.GWIDC32 874C EC80h, GWCA0.GWIDC33 874C EC84h, GWCA0.GWIDC34 874C EC88h, GWCA0.GWIDC35 874C EC8Ch, GWCA0.GWIDC36 874C EC90h, GWCA0.GWIDC37 874C EC94h, GWCA0.GWIDC38 874C EC98h, GWCA0.GWIDC39 874C EC9Ch, GWCA0.GWIDC40 874C ECA0h, GWCA0.GWIDC41 874C ECA4h, GWCA0.GWIDC42 874C ECA8h, GWCA0.GWIDC43 874C ECACh, GWCA0.GWIDC44 874C ECB0h, GWCA0.GWIDC45 874C ECB4h, GWCA0.GWIDC46 874C ECB8h, GWCA0.GWIDC47 874C ECBCh, GWCA0.GWIDC48 874C ECC0h, GWCA0.GWIDC49 874C ECC4h, GWCA0.GWIDC50 874C ECC8h, GWCA0.GWIDC51 874C ECCCh, GWCA0.GWIDC52 874C ECD0h, GWCA0.GWIDC53 874C ECD4h, GWCA0.GWIDC54 874C ECD8h, GWCA0.GWIDC55 874C ECDCh, GWCA0.GWIDC56 874C ECE0h, GWCA0.GWIDC57 874C ECE4h, GWCA0.GWIDC58 874C ECE8h, GWCA0.GWIDC59 874C ECECh, GWCA0.GWIDC60 874C ECF0h, GWCA0.GWIDC61 874C ECF4h, GWCA0.GWIDC62 874C ECF8h, GWCA0.GWIDC63 874C ECFCh, GWCA0_NS.GWIDC0 074C EC00h, GWCA0_NS.GWIDC1 074C EC04h, GWCA0_NS.GWIDC2 074C EC08h, GWCA0_NS.GWIDC3 074C EC0Ch, GWCA0_NS.GWIDC4 074C EC10h, GWCA0_NS.GWIDC5 074C EC14h, GWCA0_NS.GWIDC6 074C EC18h, GWCA0_NS.GWIDC7 074C EC1Ch, GWCA0_NS.GWIDC8 074C EC20h, GWCA0_NS.GWIDC9 074C EC24h, GWCA0_NS.GWIDC10 074C EC28h, GWCA0_NS.GWIDC11 074C EC2Ch, GWCA0_NS.GWIDC12 074C EC30h, GWCA0_NS.GWIDC13 074C EC34h, GWCA0_NS.GWIDC14 074C EC38h, GWCA0_NS.GWIDC15 074C EC3Ch, GWCA0_NS.GWIDC16 074C EC40h, GWCA0_NS.GWIDC17 074C EC44h, GWCA0_NS.GWIDC18 074C EC48h, GWCA0_NS.GWIDC19 074C EC4Ch, GWCA0_NS.GWIDC20 074C EC50h, GWCA0_NS.GWIDC21 074C EC54h, GWCA0_NS.GWIDC22 074C EC58h, GWCA0_NS.GWIDC23 074C EC5Ch, GWCA0_NS.GWIDC24 074C EC60h, GWCA0_NS.GWIDC25 074C EC64h, GWCA0_NS.GWIDC26 074C EC68h, GWCA0_NS.GWIDC27 074C EC6Ch, GWCA0_NS.GWIDC28 074C EC70h, GWCA0_NS.GWIDC29 074C EC74h, GWCA0_NS.GWIDC30 074C EC78h, GWCA0_NS.GWIDC31 074C EC7Ch, GWCA0_NS.GWIDC32 074C EC80h, GWCA0_NS.GWIDC33 074C EC84h, GWCA0_NS.GWIDC34 074C EC88h, GWCA0_NS.GWIDC35 074C EC8Ch, GWCA0_NS.GWIDC36 074C EC90h, GWCA0_NS.GWIDC37 074C EC94h, GWCA0_NS.GWIDC38 074C EC98h, GWCA0_NS.GWIDC39 074C EC9Ch, GWCA0_NS.GWIDC40 074C ECA0h, GWCA0_NS.GWIDC41 074C ECA4h, GWCA0_NS.GWIDC42 074C ECA8h, GWCA0_NS.GWIDC43 074C ECACh, GWCA0_NS.GWIDC44 074C ECB0h, GWCA0_NS.GWIDC45 074C ECB4h, GWCA0_NS.GWIDC46 074C ECB8h, GWCA0_NS.GWIDC47 074C ECBCh, GWCA0_NS.GWIDC48 074C ECC0h, GWCA0_NS.GWIDC49 074C ECC4h, GWCA0_NS.GWIDC50 074C ECC8h, GWCA0_NS.GWIDC51 074C ECCCh, GWCA0_NS.GWIDC52 074C ECD0h, GWCA0_NS.GWIDC53 074C ECD4h, GWCA0_NS.GWIDC54 074C ECD8h, GWCA0_NS.GWIDC55 074C ECDCh, GWCA0_NS.GWIDC56 074C ECE0h, GWCA0_NS.GWIDC57 074C ECE4h, GWCA0_NS.GWIDC58 074C ECE8h, GWCA0_NS.GWIDC59 074C ECECh, GWCA0_NS.GWIDC60 074C ECF0h, GWCA0_NS.GWIDC61 074C ECF4h, GWCA0_NS.GWIDC62 074C ECF8h, GWCA0_NS.GWIDC63 074C ECFCh



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	IDV[11:0]	Interrupt Delay Value	Configures the delay between the interrupt status flag GWDISi.DISt setting and the actual interrupt event in µs.	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R

IDV[11:0] Bits (Interrupt Delay Value)

If these bits are set to 0, data interrupt will occur without delay.

If value X is set to this register, the interrupt will be delayed by a time between X-1*clk_delay_period µs and X*clk_delay_period µs where clk_delay_period is the internal clock created using GWIDPC register period.

For further information, refer to section 38.3.6, Interrupt Delay Function.

38.2.10 Counter Registers

38.2.10.1 Received Data Counter Register (GWRDCN)

Address(es): GWCA0.GWRDCN 874C F000h, GWCA0_NS.GWRDCN 074C F000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RDN[31:0]	Received Data Number	These bits counts data that has been received by GWCA (data aborted by AXI error are accounted here).	R

RDN[31:0] Bits (Received Data Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a frame is passed from multicast control to AXI master interface and if these bits have a value different than FFFF FFFFh.

38.2.10.2 Transmitted Data Counter Register (GWTDCN)

Address(es): GWCA0.GWTDCN 874C F004h, GWCA0_NS.GWTDCN 074C F004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TDN[31:0]	Transmitted Data Number	These bits count data that has been transmit by GWCA (this includes data transmitted with an error and frames smaller than 32 bytes).	R

TDN[31:0] Bits (Transmitted Data Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a frame is passed from AXI master interface to TX data store and if these bits have a value different than FFFF FFFFh.

38.2.10.3 Timestamp Counter Register (GWTSCN)

Address(es): GWCA0.GWTSCN 874C F008h, GWCA0_NS.GWTSCN 074C F008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TN[31:0]	Timestamp Number	These bits count timestamp processed by GWCA (TS lost because of AXI errors are accounted here)	R

TN[31:0] Bits (Timestamp Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Timestamp is saved in timestamp RAM and if these bits have a value different than FFFF FFFFh.

38.2.10.4 Timestamp Overflow Error Counter Register (GWTSOVFECN)

Address(es): GWCA0.GWTSOVFECN 874C F00Ch, GWCA0_NS.GWTSOVFECN 074C F00Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSOVFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TSOVFEN[15:0]	Timestamp Overflow Error Number	These bits count the number of timestamps lost because of timestamp RAM overflow.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TSOVFEN[15:0] Bits (Timestamp Overflow Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a timestamp overflow error happens and if these bits have a value different than FFFFh.

38.2.10.5 Under Minimum Frame Size Error Counter Register (GWUSMFSECN)

Address(es): GWCA0.GWUSMFSECN 874C F010h, GWCA0_NS.GWUSMFSECN 074C F010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USMFSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	USMFSEN[15:0]	Under Minimum Frame Size Error Number	These bits count the number of transmit data lost because of under switch minimum size error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

USMFSEN[15:0] Bits (Under Minimum Frame Size Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Under Switch Minimum Frame Size Error happens and if these bits have a value different than FFFFh.

38.2.10.6 TAG Filtering Error Counter Register (GWTFECN)

Address(es): GWCA0.GWTFECN 874C F014h, GWCA0_NS.GWTFECN 074C F014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TFEN[15:0]	TAG Filtering Error Number	These bits count the number of transmit data lost because of TAG filtering.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TFEN[15:0] Bits (TAG Filtering Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a TAG filter error happens and if these bits have a value different than FFFFh.

38.2.10.7 Sequence Error Counter Register (GWSEQECN)

Address(es): GWCA0.GWSEQECN 874C F018h, GWCA0_NS.GWSEQECN 074C F018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SEQEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SEQEN[15:0]	Sequence Error Number	These bits count the number of transmit data lost because of a sequence error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

SEQEN[15:0] Bits (Sequence Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a sequence error happens and if these bits have a value different than FFFFh.

38.2.10.8 TX Descriptor Number Error Counter Register (GWTXDNECN)

Address(es): GWCA0.GWTXDNECN 874C F020h, GWCA0_NS.GWTXDNECN 074C F020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TXDNEN[15:0]	TX Descriptor Number Error Number	These bits count the number of transmit data lost because of a TX Descriptor Number Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TXDNEN[15:0] Bits (TX Descriptor Number Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a TX Descriptor Number Error happens and if these bits have a value different than FFFFh.

38.2.10.9 Frame Size Error Counter Register (GWFSECN)

Address(es): GWCA0.GWFSECN 874C F024h, GWCA0_NS.GWFSECN 074C F024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	FSEN[15:0]	Frame Size Error Number	These bits count the number of receive data lost because of a Frame Size Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

FSEN[15:0] Bits (Frame Size Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Frame Size Error happens and if these bits have a value different than FFFFh.

38.2.10.10 Timestamp Descriptor Full Error Counter Register (GWTDFECN)

Address(es): GWCA0.GWTDFECN 874C F028h, GWCA0_NS.GWTDFECN 074C F028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TDFEN[15:0]	Timestamp Descriptor Full Error Number	These bits count the number of timestamps lost because of a Timestamp Descriptor Full Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TDFEN[15:0] Bits (Timestamp Descriptor Full Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Timestamp Descriptor Full Error happens and if these bits have a value different than FFFFh.

38.2.10.11 Timestamp Descriptor Number Error Counter Register (GWTSDNECN)

Address(es): GWCA0.GWTSDNECN 874C F02Ch, GWCA0_NS.GWTSDNECN 074C F02Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TSDNEN[15:0]	Timestamp Descriptor Number Error Number	These bits count the number of timestamps lost because of a Timestamp Descriptor Number Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

TSDNEN[15:0] Bits (Timestamp Descriptor Number Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Timestamp Descriptor Number Error happens and if these bits have a value different than FFFFh.

38.2.10.12 Descriptor Queue Overflow Error Counter Register (GWDQOECN)

Address(es): GWCA0.GWDQOECN 874C F030h, GWCA0_NS.GWDQOECN 074C F030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DQOEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DQOEN[15:0]	Descriptor Queue Overflow Error Number	These bits count the number of receive data lost because of a Descriptor Queue Overflow Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DQOEN[15:0] Bits (Descriptor Queue Overflow Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Overflow Error happens and if these bits have a value different than FFFFh.

38.2.10.13 Descriptor Queue Security Error Counter Register (GWDQSECN)

Address(es): GWCA0.GWDQSECN 874C F034h, GWCA0_NS.GWDQSECN 074C F034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DQSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DQSEN[15:0]	Descriptor Queue Security Error Number	These bits count the number of receive data lost because of a Descriptor Queue Security Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DQSEN[15:0] Bits (Descriptor Queue Security Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Security Error happens and if these bits have a value different than FFFFh.

38.2.10.14 Descriptor Full Error Counter Register (GWDFECN)

Address(es): GWCA0.GWDFECN 874C F038h, GWCA0_NS.GWDFECN 074C F038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DFEN[15:0]	Descriptor Full Error Number	These bits count the number of receive data lost because of a Descriptor Full Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DFEN[15:0] Bits (Descriptor Full Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Full Error happens and if these bits have a value different than FFFFh.

38.2.10.15 Descriptor Security Error Counter Register (GWDSECN)

Address(es): GWCA0.GWDSECN 874C F03Ch, GWCA0_NS.GWDSECN 074C F03Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DSEN[15:0]	Descriptor Security Error Number	These bits count the number of receive data lost because of a Descriptor Security Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DSEN[15:0] Bits (Descriptor Security Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Security Error happens and if these bits have a value different than FFFFh.

38.2.10.16 Data Size Error Counter Register (GWDSZECN)

Address(es): GWCA0.GWDSZECN 874C F040h, GWCA0_NS.GWDSZECN 074C F040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSZEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DSZEN[15:0]	Data Size Error Number	These bits count the number of receive data lost because of a Data Size Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DSZEN[15:0] Bits (Data Size Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Data Size Error happens and if these bits have a value different than FFFFh.

38.2.10.17 Descriptor Chain Type Error Counter Register (GWDCTECN)

Address(es): GWCA0.GWDCTECN 874C F044h, GWCA0_NS.GWDCTECN 074C F044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DCTEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DCTEN[15:0]	Descriptor Chain Type Error Number	These bits count the number of receive data lost because of a Descriptor Chain Type Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

DCTEN[15:0] Bits (Descriptor Chain Type Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Chain Type Error happens and if these bits have a value different than FFFFh.

38.2.10.18 RX Descriptor Number Error Counter Register (GWRXDNECN)

Address(es): GWCA0.GWRXDNECN 874C F048h, GWCA0_NS.GWRXDNECN 074C F048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RXDNEN[15:0]	RX Descriptor Number Error Number	These bits count the number of receive data lost because of a RX Descriptor Number Error.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

RXDNEN[15:0] Bits (RX Descriptor Number Error Number)

[Clearing conditions]

- HW: Being in RESET mode will clear these bits.
- SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a RX Descriptor Number Error happens and if these bits have a value different than FFFFh.

38.2.11 Data Interrupt Registers

38.2.11.1 Data Interrupt Status Register i (GWDISi) (i = 0, 1)

- GWCA0.GWDIS0

Address(es): GWCA0.GWDIS0 874C F100h, GWCA0_NS.GWDIS0 074C F100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIS31	DIS30	DIS29	DIS28	DIS27	DIS26	DIS25	DIS24	DIS23	DIS22	DIS21	DIS20	DIS19	DIS18	DIS17	DIS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIS15	DIS14	DIS13	DIS12	DIS11	DIS10	DIS9	DIS8	DIS7	DIS6	DIS5	DIS4	DIS3	DIS2	DIS1	DIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIS0	Data 0 Interrupt Status Flag	0: No Data 0 Interrupt 1: Data 0 Interrupt occurred.	R/W
b1	DIS1	Data 1 Interrupt Status Flag	0: No data 1 Interrupt 1: Data 1 Interrupt occurred.	R/W
b2	DIS2	Data 2 Interrupt Status Flag	0: No data 2 Interrupt 1: Data 2 Interrupt occurred.	R/W
b3	DIS3	Data 3 Interrupt Status Flag	0: No data 3 Interrupt 1: Data 3 Interrupt occurred.	R/W
b4	DIS4	Data 4 Interrupt Status Flag	0: No data 4 Interrupt 1: Data 4 Interrupt occurred.	R/W
b5	DIS5	Data 5 Interrupt Status Flag	0: No data 5 Interrupt 1: Data 5 Interrupt occurred.	R/W
b6	DIS6	Data 6 Interrupt Status Flag	0: No data 6 Interrupt 1: Data 6 Interrupt occurred.	R/W
b7	DIS7	Data 7 Interrupt Status Flag	0: No data 7 Interrupt 1: Data 7 Interrupt occurred.	R/W
b8	DIS8	Data 8 Interrupt Status Flag	0: No data 8 Interrupt 1: Data 8 Interrupt occurred.	R/W
b9	DIS9	Data 9 Interrupt Status Flag	0: No data 9 Interrupt 1: Data 9 Interrupt occurred.	R/W
b10	DIS10	Data 10 Interrupt Status Flag	0: No data 10 Interrupt 1: Data 10 Interrupt occurred.	R/W
b11	DIS11	Data 11 Interrupt Status Flag	0: No data 11 Interrupt 1: Data 11 Interrupt occurred.	R/W
b12	DIS12	Data 12 Interrupt Status Flag	0: No data 12 Interrupt 1: Data 12 Interrupt occurred.	R/W
b13	DIS13	Data 13 Interrupt Status Flag	0: No data 13 Interrupt 1: Data 13 Interrupt occurred.	R/W
b14	DIS14	Data 14 Interrupt Status Flag	0: No data 14 Interrupt 1: Data 14 Interrupt occurred.	R/W
b15	DIS15	Data 15 Interrupt Status Flag	0: No data 15 Interrupt 1: Data 15 Interrupt occurred.	R/W
b16	DIS16	Data 16 Interrupt Status Flag	0: No data 16 Interrupt 1: Data 16 Interrupt occurred.	R/W
b17	DIS17	Data 17 Interrupt Status Flag	0: No data 17 Interrupt 1: Data 17 Interrupt occurred.	R/W

Bit	Symbol	Bit Name	Description	R/W
b18	DIS18	Data 18 Interrupt Status Flag	0: No data 18 Interrupt 1: Data 18 Interrupt occurred.	R/W
b19	DIS19	Data 19 Interrupt Status Flag	0: No data 19 Interrupt 1: Data 19 Interrupt occurred.	R/W
b20	DIS20	Data 20 Interrupt Status Flag	0: No data 20 Interrupt 1: Data 20 Interrupt occurred.	R/W
b21	DIS21	Data 21 Interrupt Status Flag	0: No data 21 Interrupt 1: Data 21 Interrupt occurred.	R/W
b22	DIS22	Data 22 Interrupt Status Flag	0: No data 22 Interrupt 1: Data 22 Interrupt occurred.	R/W
b23	DIS23	Data 23 Interrupt Status Flag	0: No data 23 Interrupt 1: Data 23 Interrupt occurred.	R/W
b24	DIS24	Data 24 Interrupt Status Flag	0: No data 24 Interrupt 1: Data 24 Interrupt occurred.	R/W
b25	DIS25	Data 25 Interrupt Status Flag	0: No data 25 Interrupt 1: Data 25 Interrupt occurred.	R/W
b26	DIS26	Data 26 Interrupt Status Flag	0: No data 26 Interrupt 1: Data 26 Interrupt occurred.	R/W
b27	DIS27	Data 27 Interrupt Status Flag	0: No data 27 Interrupt 1: Data 27 Interrupt occurred.	R/W
b28	DIS28	Data 28 Interrupt Status Flag	0: No data 28 Interrupt 1: Data 28 Interrupt occurred.	R/W
b29	DIS29	Data 29 Interrupt Status Flag	0: No data 29 Interrupt 1: Data 29 Interrupt occurred.	R/W
b30	DIS30	Data 30 Interrupt Status Flag	0: No data 30 Interrupt 1: Data 30 Interrupt occurred.	R/W
b31	DIS31	Data 31 Interrupt Status Flag	0: No data 31 Interrupt 1: Data 31 Interrupt occurred.	R/W

- GWCA0.GWDIS1

Address(es): GWCA0.GWDIS1 874C F110h, GWCA0_NS.GWDIS1 074C F110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIS63	DIS62	DIS61	DIS60	DIS59	DIS58	DIS57	DIS56	DIS55	DIS54	DIS53	DIS52	DIS51	DIS50	DIS49	DIS48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIS47	DIS46	DIS45	DIS44	DIS43	DIS42	DIS41	DIS40	DIS39	DIS38	DIS37	DIS36	DIS35	DIS34	DIS33	DIS32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIS32	Data 32 Interrupt Status Flag	0: No data 32 Interrupt 1: Data 32 Interrupt occurred.	R/W
b1	DIS33	Data 33 Interrupt Status Flag	0: No data 33 Interrupt 1: Data 33 Interrupt occurred.	R/W
b2	DIS34	Data 34 Interrupt Status Flag	0: No data 34 Interrupt 1: Data 34 Interrupt occurred.	R/W
b3	DIS35	Data 35 Interrupt Status Flag	0: No data 35 Interrupt 1: Data 35 Interrupt occurred.	R/W
b4	DIS36	Data 36 Interrupt Status Flag	0: No data 36 Interrupt 1: Data 36 Interrupt occurred.	R/W
b5	DIS37	Data 37 Interrupt Status Flag	0: No data 37 Interrupt 1: Data 37 Interrupt occurred.	R/W
b6	DIS38	Data 38 Interrupt Status Flag	0: No data 38 Interrupt 1: Data 38 Interrupt occurred.	R/W
b7	DIS39	Data 39 Interrupt Status Flag	0: No data 39 Interrupt 1: Data 39 Interrupt occurred.	R/W
b8	DIS40	Data 40 Interrupt Status Flag	0: No data 40 Interrupt 1: Data 40 Interrupt occurred.	R/W
b9	DIS41	Data 41 Interrupt Status Flag	0: No data 41 Interrupt 1: Data 41 Interrupt occurred.	R/W
b10	DIS42	Data 42 Interrupt Status Flag	0: No data 42 Interrupt 1: Data 42 Interrupt occurred.	R/W
b11	DIS43	Data 43 Interrupt Status Flag	0: No data 43 Interrupt 1: Data 43 Interrupt occurred.	R/W
b12	DIS44	Data 44 Interrupt Status Flag	0: No data 44 Interrupt 1: Data 44 Interrupt occurred.	R/W
b13	DIS45	Data 45 Interrupt Status Flag	0: No data 45 Interrupt 1: Data 45 Interrupt occurred.	R/W
b14	DIS46	Data 46 Interrupt Status Flag	0: No data 46 Interrupt 1: Data 46 Interrupt occurred.	R/W
b15	DIS47	Data 47 Interrupt Status Flag	0: No data 47 Interrupt 1: Data 47 Interrupt occurred.	R/W
b16	DIS48	Data 48 Interrupt Status Flag	0: No data 48 Interrupt 1: Data 48 Interrupt occurred.	R/W
b17	DIS49	Data 49 Interrupt Status Flag	0: No data 49 Interrupt 1: Data 49 Interrupt occurred.	R/W
b18	DIS50	Data 50 Interrupt Status Flag	0: No data 50 Interrupt 1: Data 50 Interrupt occurred.	R/W
b19	DIS51	Data 51 Interrupt Status Flag	0: No data 51 Interrupt 1: Data 51 Interrupt occurred.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	DIS52	Data 52 Interrupt Status Flag	0: No data 52 Interrupt 1: Data 52 Interrupt occurred.	R/W
b21	DIS53	Data 53 Interrupt Status Flag	0: No data 53 Interrupt 1: Data 53 Interrupt occurred.	R/W
b22	DIS54	Data 54 Interrupt Status Flag	0: No data 54 Interrupt 1: Data 54 Interrupt occurred.	R/W
b23	DIS55	Data 55 Interrupt Status Flag	0: No data 55 Interrupt 1: Data 55 Interrupt occurred.	R/W
b24	DIS56	Data 56 Interrupt Status Flag	0: No data 56 Interrupt 1: Data 56 Interrupt occurred.	R/W
b25	DIS57	Data 57 Interrupt Status Flag	0: No data 57 Interrupt 1: Data 57 Interrupt occurred.	R/W
b26	DIS58	Data 58 Interrupt Status Flag	0: No data 58 Interrupt 1: Data 58 Interrupt occurred.	R/W
b27	DIS59	Data 59 Interrupt Status Flag	0: No data 59 Interrupt 1: Data 59 Interrupt occurred.	R/W
b28	DIS60	Data 60 Interrupt Status Flag	0: No data 60 Interrupt 1: Data 60 Interrupt occurred.	R/W
b29	DIS61	Data 61 Interrupt Status Flag	0: No data 61 Interrupt 1: Data 61 Interrupt occurred.	R/W
b30	DIS62	Data 62 Interrupt Status Flag	0: No data 62 Interrupt 1: Data 62 Interrupt occurred.	R/W
b31	DIS63	Data 63 Interrupt Status Flag	0: No data 63 Interrupt 1: Data 63 Interrupt occurred.	R/W

DISt Flag (Data t Interrupt Status Flag) (t = 0 to 63)

[Setting condition TX]

- HW: A descriptor has been processed in the corresponding queue and DESCR.DIE was set in the corresponding read descriptor.
  [About set timing information for normal TX]
  A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed.

[Setting condition RX]

- HW: A frame has been processed in the corresponding queue and DESCR.DIE was set in at least one of the corresponding read descriptors.
  [About set timing information for normal RX]
  A frame is considered as processed when frame FSTART or FSINGLE descriptor is processed. A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed.
  All unknown descriptors are never considered as processed (refer to GWEIS2i.DFESt).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

38.2.11.2 Data Interrupt Enable Register i (GWDIEi) (i = 0, 1)

- GWCA0.GWDIE0

Address(es): GWCA0.GWDIE0 874C F104h, GWCA0_NS.GWDIE0 074C F104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIE31	DIE30	DIE29	DIE28	DIE27	DIE26	DIE25	DIE24	DIE23	DIE22	DIE21	DIE20	DIE19	DIE18	DIE17	DIE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIE15	DIE14	DIE13	DIE12	DIE11	DIE10	DIE9	DIE8	DIE7	DIE6	DIE5	DIE4	DIE3	DIE2	DIE1	DIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIE0	Data 0 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DIE1	Data 1 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	DIE2	Data 2 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	DIE3	Data 3 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	DIE4	Data 4 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	DIE5	Data 5 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b6	DIE6	Data 6 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b7	DIE7	Data 7 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b8	DIE8	Data 8 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b9	DIE9	Data 9 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	DIE10	Data 10 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b11	DIE11	Data 11 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b12	DIE12	Data 12 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b13	DIE13	Data 13 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b14	DIE14	Data 14 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15	DIE15	Data 15 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b16	DIE16	Data 16 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	DIE17	Data 17 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	DIE18	Data 18 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b19	DIE19	Data 19 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b20	DIE20	Data 20 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	DIE21	Data 21 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	DIE22	Data 22 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	DIE23	Data 23 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	DIE24	Data 24 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	DIE25	Data 25 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b26	DIE26	Data 26 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27	DIE27	Data 27 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b28	DIE28	Data 28 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b29	DIE29	Data 29 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b30	DIE30	Data 30 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31	DIE31	Data 31 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWDID0 register.

- GWCA0.GWDIE1

Address(es): GWCA0.GWDIE1 874C F114h, GWCA0_NS.GWDIE1 074C F114h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIE63	DIE62	DIE61	DIE60	DIE59	DIE58	DIE57	DIE56	DIE55	DIE54	DIE53	DIE52	DIE51	DIE50	DIE49	DIE48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIE47	DIE46	DIE45	DIE44	DIE43	DIE42	DIE41	DIE40	DIE39	DIE38	DIE37	DIE36	DIE35	DIE34	DIE33	DIE32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIE32	Data 32 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b1	DIE33	Data 33 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b2	DIE34	Data 34 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b3	DIE35	Data 35 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b4	DIE36	Data 36 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b5	DIE37	Data 37 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b6	DIE38	Data 38 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b7	DIE39	Data 39 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b8	DIE40	Data 40 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b9	DIE41	Data 41 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b10	DIE42	Data 42 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b11	DIE43	Data 43 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b12	DIE44	Data 44 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b13	DIE45	Data 45 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b14	DIE46	Data 46 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b15	DIE47	Data 47 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b16	DIE48	Data 48 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b17	DIE49	Data 49 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b18	DIE50	Data 50 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b19	DIE51	Data 51 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b20	DIE52	Data 52 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b21	DIE53	Data 53 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b22	DIE54	Data 54 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b23	DIE55	Data 55 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b24	DIE56	Data 56 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b25	DIE57	Data 57 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b26	DIE58	Data 58 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b27	DIE59	Data 59 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b28	DIE60	Data 60 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b29	DIE61	Data 61 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b30	DIE62	Data 62 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1
b31	DIE63	Data 63 Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/(W) *1

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWDID1 register.

DIEt Bit (Data t Interrupt Enable) (t = 0 to 63)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to GWDIDi.DIDt bit will clear this bit.

38.2.11.3 Data Interrupt Disable Register i (GWDIDi) (i = 0, 1)

- GWCA0.GWDID0

Address(es): GWCA0.GWDID0 874C F108h, GWCA0_NS.GWDID0 074C F108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DID0	Data 0 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE0 bit to 0.	R/W
b1	DID1	Data 1 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE1 bit to 0.	R/W
b2	DID2	Data 2 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE2 bit to 0.	R/W
b3	DID3	Data 3 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE3 bit to 0.	R/W
b4	DID4	Data 4 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE4 bit to 0.	R/W
b5	DID5	Data 5 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE5 bit to 0.	R/W
b6	DID6	Data 6 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE6 bit to 0.	R/W
b7	DID7	Data 7 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE7 bit to 0.	R/W
b8	DID8	Data 8 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE8 bit to 0.	R/W
b9	DID9	Data 9 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE9 bit to 0.	R/W
b10	DID10	Data 10 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE10 bit to 0.	R/W
b11	DID11	Data 11 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE11 bit to 0.	R/W
b12	DID12	Data 12 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE12 bit to 0.	R/W
b13	DID13	Data 13 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE13 bit to 0.	R/W
b14	DID14	Data 14 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE14 bit to 0.	R/W
b15	DID15	Data 15 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE15 bit to 0.	R/W
b16	DID16	Data 16 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE16 bit to 0.	R/W
b17	DID17	Data 17 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE17 bit to 0.	R/W
b18	DID18	Data 18 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE18 bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	DID19	Data 19 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE19 bit to 0.	R/W
b20	DID20	Data 20 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE20 bit to 0.	R/W
b21	DID21	Data 21 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE21 bit to 0.	R/W
b22	DID22	Data 22 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE22 bit to 0.	R/W
b23	DID23	Data 23 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE23 bit to 0.	R/W
b24	DID24	Data 24 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE24 bit to 0.	R/W
b25	DID25	Data 25 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE25 bit to 0.	R/W
b26	DID26	Data 26 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE26 bit to 0.	R/W
b27	DID27	Data 27 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE27 bit to 0.	R/W
b28	DID28	Data 28 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE28 bit to 0.	R/W
b29	DID29	Data 29 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE29 bit to 0.	R/W
b30	DID30	Data 30 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE30 bit to 0.	R/W
b31	DID31	Data 31 Interrupt Disable	0: Has no effects. 1: Sets GWDIE0.DIE31 bit to 0.	R/W

- GWCA0.GWDID1

Address(es): GWCA0.GWDID1 874C F118h, GWCA0_NS.GWDID1 074C F118h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DID63	DID62	DID61	DID60	DID59	DID58	DID57	DID56	DID55	DID54	DID53	DID52	DID51	DID50	DID49	DID48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DID47	DID46	DID45	DID44	DID43	DID42	DID41	DID40	DID39	DID38	DID37	DID36	DID35	DID34	DID33	DID32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DID32	Data 32 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE32 bit to 0.	R/W
b1	DID33	Data 33 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE33 bit to 0.	R/W
b2	DID34	Data 34 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE34 bit to 0.	R/W
b3	DID35	Data 35 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE35 bit to 0.	R/W
b4	DID36	Data 36 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE36 bit to 0.	R/W
b5	DID37	Data 37 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE37 bit to 0.	R/W
b6	DID38	Data 38 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE38 bit to 0.	R/W
b7	DID39	Data 39 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE39 bit to 0.	R/W
b8	DID40	Data 40 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE40 bit to 0.	R/W
b9	DID41	Data 41 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE41 bit to 0.	R/W
b10	DID42	Data 42 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE42 bit to 0.	R/W
b11	DID43	Data 43 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE43 bit to 0.	R/W
b12	DID44	Data 44 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE44 bit to 0.	R/W
b13	DID45	Data 45 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE45 bit to 0.	R/W
b14	DID46	Data 46 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE46 bit to 0.	R/W
b15	DID47	Data 47 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE47 bit to 0.	R/W
b16	DID48	Data 48 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE48 bit to 0.	R/W
b17	DID49	Data 49 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE49 bit to 0.	R/W
b18	DID50	Data 50 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE50 bit to 0.	R/W
b19	DID51	Data 51 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE51 bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	DID52	Data 52 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE52 bit to 0.	R/W
b21	DID53	Data 53 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE53 bit to 0.	R/W
b22	DID54	Data 54 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE54 bit to 0.	R/W
b23	DID55	Data 55 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE55 bit to 0.	R/W
b24	DID56	Data 56 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE56 bit to 0.	R/W
b25	DID57	Data 57 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE57 bit to 0.	R/W
b26	DID58	Data 58 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE58 bit to 0.	R/W
b27	DID59	Data 59 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE59 bit to 0.	R/W
b28	DID60	Data 60 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE60 bit to 0.	R/W
b29	DID61	Data 61 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE61 bit to 0.	R/W
b30	DID62	Data 62 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE62 bit to 0.	R/W
b31	DID63	Data 63 Interrupt Disable	0: Has no effects. 1: Sets GWDIE1.DIE63 bit to 0.	R/W

38.2.11.4 Data Interrupt Delayed Status Register i (GWDIDSi) (i = 0, 1)

- GWCA0.GWDIDS0

Address(es): GWCA0.GWDIDS0 874C F10Ch, GWCA0_NS.GWDIDS0 074C F10Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIDS31	DIDS30	DIDS29	DIDS28	DIDS27	DIDS26	DIDS25	DIDS24	DIDS23	DIDS22	DIDS21	DIDS20	DIDS19	DIDS18	DIDS17	DIDS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIDS15	DIDS14	DIDS13	DIDS12	DIDS11	DIDS10	DIDS9	DIDS8	DIDS7	DIDS6	DIDS5	DIDS4	DIDS3	DIDS2	DIDS1	DIDS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIDS0	Data 0 Interrupt Delayed Status Flag	0: No Data 0 interrupt delayed 1: Data 0 interrupt delayed occurred.	R
b1	DIDS1	Data 1 Interrupt Delayed Status Flag	0: No Data 1 interrupt delayed 1: Data 1 interrupt delayed occurred.	R
b2	DIDS2	Data 2 Interrupt Delayed Status Flag	0: No Data 2 interrupt delayed 1: Data 2 interrupt delayed occurred.	R
b3	DIDS3	Data 3 Interrupt Delayed Status Flag	0: No Data 3 interrupt delayed 1: Data 3 interrupt delayed occurred.	R
b4	DIDS4	Data 4 Interrupt Delayed Status Flag	0: No Data 4 interrupt delayed 1: Data 4 interrupt delayed occurred.	R
b5	DIDS5	Data 5 Interrupt Delayed Status Flag	0: No Data 5 interrupt delayed 1: Data 5 interrupt delayed occurred.	R
b6	DIDS6	Data 6 Interrupt Delayed Status Flag	0: No Data 6 interrupt delayed 1: Data 6 interrupt delayed occurred.	R
b7	DIDS7	Data 7 Interrupt Delayed Status Flag	0: No Data 7 interrupt delayed 1: Data 7 interrupt delayed occurred.	R
b8	DIDS8	Data 8 Interrupt Delayed Status Flag	0: No Data 8 interrupt delayed 1: Data 8 interrupt delayed occurred.	R
b9	DIDS9	Data 9 Interrupt Delayed Status Flag	0: No Data 9 interrupt delayed 1: Data 9 interrupt delayed occurred.	R
b10	DIDS10	Data 10 Interrupt Delayed Status Flag	0: No Data 10 interrupt delayed 1: Data 10 interrupt delayed occurred.	R
b11	DIDS11	Data 11 Interrupt Delayed Status Flag	0: No Data 11 interrupt delayed 1: Data 11 interrupt delayed occurred.	R
b12	DIDS12	Data 12 Interrupt Delayed Status Flag	0: No Data 12 interrupt delayed 1: Data 12 interrupt delayed occurred.	R
b13	DIDS13	Data 13 Interrupt Delayed Status Flag	0: No Data 13 interrupt delayed 1: Data 13 interrupt delayed occurred.	R
b14	DIDS14	Data 14 Interrupt Delayed Status Flag	0: No Data 14 interrupt delayed 1: Data 14 interrupt delayed occurred.	R
b15	DIDS15	Data 15 Interrupt Delayed Status Flag	0: No Data 15 interrupt delayed 1: Data 15 interrupt delayed occurred.	R
b16	DIDS16	Data 16 Interrupt Delayed Status Flag	0: No Data 16 interrupt delayed 1: Data 16 interrupt delayed occurred.	R
b17	DIDS17	Data 17 Interrupt Delayed Status Flag	0: No Data 17 interrupt delayed 1: Data 17 interrupt delayed occurred.	R
b18	DIDS18	Data 18 Interrupt Delayed Status Flag	0: No Data 18 interrupt delayed 1: Data 18 interrupt delayed occurred.	R

Bit	Symbol	Bit Name	Description	R/W
b19	DIDS19	Data 19 Interrupt Delayed Status Flag	0: No Data 19 interrupt delayed 1: Data 19 interrupt delayed occurred.	R
b20	DIDS20	Data 20 Interrupt Delayed Status Flag	0: No Data 20 interrupt delayed 1: Data 20 interrupt delayed occurred.	R
b21	DIDS21	Data 21 Interrupt Delayed Status Flag	0: No Data 21 interrupt delayed 1: Data 21 interrupt delayed occurred.	R
b22	DIDS22	Data 22 Interrupt Delayed Status Flag	0: No Data 22 interrupt delayed 1: Data 22 interrupt delayed occurred.	R
b23	DIDS23	Data 23 Interrupt Delayed Status Flag	0: No Data 23 interrupt delayed 1: Data 23 interrupt delayed occurred.	R
b24	DIDS24	Data 24 Interrupt Delayed Status Flag	0: No Data 24 interrupt delayed 1: Data 24 interrupt delayed occurred.	R
b25	DIDS25	Data 25 Interrupt Delayed Status Flag	0: No Data 25 interrupt delayed 1: Data 25 interrupt delayed occurred.	R
b26	DIDS26	Data 26 Interrupt Delayed Status Flag	0: No Data 26 interrupt delayed 1: Data 26 interrupt delayed occurred.	R
b27	DIDS27	Data 27 Interrupt Delayed Status Flag	0: No Data 27 interrupt delayed 1: Data 27 interrupt delayed occurred.	R
b28	DIDS28	Data 28 Interrupt Delayed Status Flag	0: No Data 28 interrupt delayed 1: Data 28 interrupt delayed occurred.	R
b29	DIDS29	Data 29 Interrupt Delayed Status Flag	0: No Data 29 interrupt delayed 1: Data 29 interrupt delayed occurred.	R
b30	DIDS30	Data 30 Interrupt Delayed Status Flag	0: No Data 30 interrupt delayed 1: Data 30 interrupt delayed occurred.	R
b31	DIDS31	Data 31 Interrupt Delayed Status Flag	0: No Data 31 interrupt delayed 1: Data 31 interrupt delayed occurred.	R

- GWCA0.GWDIDS1

Address(es): GWCA0.GWDIDS1 874C F11Ch, GWCA0_NS.GWDIDS1 074C F11Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIDS63	DIDS62	DIDS61	DIDS60	DIDS59	DIDS58	DIDS57	DIDS56	DIDS55	DIDS54	DIDS53	DIDS52	DIDS51	DIDS50	DIDS49	DIDS48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIDS47	DIDS46	DIDS45	DIDS44	DIDS43	DIDS42	DIDS41	DIDS40	DIDS39	DIDS38	DIDS37	DIDS36	DIDS35	DIDS34	DIDS33	DIDS32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIDS32	Data 32 Interrupt Delayed Status Flag	0: No Data 32 interrupt delayed 1: Data 32 interrupt delayed occurred.	R
b1	DIDS33	Data 33 Interrupt Delayed Status Flag	0: No Data 33 interrupt delayed 1: Data 33 interrupt delayed occurred.	R
b2	DIDS34	Data 34 Interrupt Delayed Status Flag	0: No Data 34 interrupt delayed 1: Data 34 interrupt delayed occurred.	R
b3	DIDS35	Data 35 Interrupt Delayed Status Flag	0: No Data 35 interrupt delayed 1: Data 35 interrupt delayed occurred.	R
b4	DIDS36	Data 36 Interrupt Delayed Status Flag	0: No Data 36 interrupt delayed 1: Data 36 interrupt delayed occurred.	R
b5	DIDS37	Data 37 Interrupt Delayed Status Flag	0: No Data 37 interrupt delayed 1: Data 37 interrupt delayed occurred.	R
b6	DIDS38	Data 38 Interrupt Delayed Status Flag	0: No Data 38 interrupt delayed 1: Data 38 interrupt delayed occurred.	R
b7	DIDS39	Data 39 Interrupt Delayed Status Flag	0: No Data 39 interrupt delayed 1: Data 39 interrupt delayed occurred.	R
b8	DIDS40	Data 40 Interrupt Delayed Status Flag	0: No Data 40 interrupt delayed 1: Data 40 interrupt delayed occurred.	R
b9	DIDS41	Data 41 Interrupt Delayed Status Flag	0: No Data 41 interrupt delayed 1: Data 41 interrupt delayed occurred.	R
b10	DIDS42	Data 42 Interrupt Delayed Status Flag	0: No Data 42 interrupt delayed 1: Data 42 interrupt delayed occurred.	R
b11	DIDS43	Data 43 Interrupt Delayed Status Flag	0: No Data 43 interrupt delayed 1: Data 43 interrupt delayed occurred.	R
b12	DIDS44	Data 44 Interrupt Delayed Status Flag	0: No Data 44 interrupt delayed 1: Data 44 interrupt delayed occurred.	R
b13	DIDS45	Data 45 Interrupt Delayed Status Flag	0: No Data 45 interrupt delayed 1: Data 45 interrupt delayed occurred.	R
b14	DIDS46	Data 46 Interrupt Delayed Status Flag	0: No Data 46 interrupt delayed 1: Data 46 interrupt delayed occurred.	R
b15	DIDS47	Data 47 Interrupt Delayed Status Flag	0: No Data 47 interrupt delayed 1: Data 47 interrupt delayed occurred.	R
b16	DIDS48	Data 48 Interrupt Delayed Status Flag	0: No Data 48 interrupt delayed 1: Data 48 interrupt delayed occurred.	R
b17	DIDS49	Data 49 Interrupt Delayed Status Flag	0: No Data 49 interrupt delayed 1: Data 49 interrupt delayed occurred.	R
b18	DIDS50	Data 50 Interrupt Delayed Status Flag	0: No Data 50 interrupt delayed 1: Data 50 interrupt delayed occurred.	R
b19	DIDS51	Data 51 Interrupt Delayed Status Flag	0: No Data 51 interrupt delayed 1: Data 51 interrupt delayed occurred.	R

Bit	Symbol	Bit Name	Description	R/W
b20	DIDS52	Data 52 Interrupt Delayed Status Flag	0: No Data 52 interrupt delayed 1: Data 52 interrupt delayed occurred.	R
b21	DIDS53	Data 53 Interrupt Delayed Status Flag	0: No Data 53 interrupt delayed 1: Data 53 interrupt delayed occurred.	R
b22	DIDS54	Data 54 Interrupt Delayed Status Flag	0: No Data 54 interrupt delayed 1: Data 54 interrupt delayed occurred.	R
b23	DIDS55	Data 55 Interrupt Delayed Status Flag	0: No Data 55 interrupt delayed 1: Data 55 interrupt delayed occurred.	R
b24	DIDS56	Data 56 Interrupt Delayed Status Flag	0: No Data 56 interrupt delayed 1: Data 56 interrupt delayed occurred.	R
b25	DIDS57	Data 57 Interrupt Delayed Status Flag	0: No Data 57 interrupt delayed 1: Data 57 interrupt delayed occurred.	R
b26	DIDS58	Data 58 Interrupt Delayed Status Flag	0: No Data 58 interrupt delayed 1: Data 58 interrupt delayed occurred.	R
b27	DIDS59	Data 59 Interrupt Delayed Status Flag	0: No Data 59 interrupt delayed 1: Data 59 interrupt delayed occurred.	R
b28	DIDS60	Data 60 Interrupt Delayed Status Flag	0: No Data 60 interrupt delayed 1: Data 60 interrupt delayed occurred.	R
b29	DIDS61	Data 61 Interrupt Delayed Status Flag	0: No Data 61 interrupt delayed 1: Data 61 interrupt delayed occurred.	R
b30	DIDS62	Data 62 Interrupt Delayed Status Flag	0: No Data 62 interrupt delayed 1: Data 62 interrupt delayed occurred.	R
b31	DIDS63	Data 63 Interrupt Delayed Status Flag	0: No Data 63 interrupt delayed 1: Data 63 interrupt delayed occurred.	R

38.2.11.5 Timestamp Data Interrupt Status Register (GWTSDIS)

Address(es): GWCA0.GWTSDIS 874C F180h, GWCA0_NS.GWTSDIS 074C F180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDIS1	TSDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSDIS0	Timestamp Data 0 Interrupt Status Flag	0: No Timestamp Data 0 Interrupt 1: Timestamp Data 0 Interrupt occurred.	R/W
b1	TSDIS1	Timestamp Data 1 Interrupt Status Flag	0: No Timestamp Data 1 Interrupt 1: Timestamp Data 1 Interrupt occurred.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

TSDISt Flag (Timestamp Data t Interrupt Status Flag) (t = 0, 1)

[Setting condition]

- HW: A descriptor is processed in the corresponding queue and DESCR.DIE was set in the corresponding read descriptor. A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed. All unknown descriptors are never considered as processed (refer to GWEIS0.TDFES).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

38.2.11.6 Timestamp Data Interrupt Enable Register (GWTSDIE)

Address(es): GWCA0.GWTSDIE 874C F184h, GWCA0_NS.GWTSDIE 074C F184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDIE1	TSDIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSDIE0	Timestamp Data 0 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	TSDIE1	Timestamp Data 1 Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWTSDID register.

TSDIEt Bit (Timestamp Data Interrupt Enable t) (t = 0, 1)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWTSDID.TSDIDt bit will clear this bit.

38.2.11.7 Timestamp Data Interrupt Disable Register (GWTSDID)

Address(es): GWCA0.GWTSDID 874C F188h, GWCA0_NS.GWTSDID 074C F188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDID1	TSDID0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSDID0	Timestamp Data 0 Interrupt Disable	0: Has no effects. 1: Sets GWTSDIE.TSDIE0 bit to 0.	R/W
b1	TSDID1	Timestamp Data 1 Interrupt Disable	0: Has no effects. 1: Sets GWTSDIE.TSDIE1 bit to 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.12 Error Interrupt Registers

38.2.12.1 Error Interrupt Status Register 0 (GWEIS0)

Address(es): GWCA0.GWEIS0 874C F190h, GWCA0_NS.GWEIS0 074C F190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TSDNES1	TSDNES0	—	—	TDFES1	TDFES0	FSES7	FSES6	FSES5	FSES4	FSES3	FSES2	FSES1	FSES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSHES	TXDNES	—	SEQES	TFES	USMFSES	TSOVFES	—	—	—	—	—	—	—	—	AES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AES	AXI Error Status Flag	0: No AXI Error 1: AXI Error occurred.	R/W
b8 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b9	TSOVFES	Timestamp Overflow Error Status Flag	0: No Timestamp Overflow Error 1: Timestamp Overflow Error occurred.	R/W
b10	USMFSES	Under Minimum Frame Size Error Status Flag	0: No Under Minimum Frame Size Error 1: Under Minimum Frame Size Error occurred.	R/W
b11	TFES	TAG Filtering Error Status Flag	0: No TAG Filtering Error 1: TAG Filtering Error occurred.	R/W
b12	SEQES	Sequence Error Status Flag	0: No Sequence Error 1: Sequence Error occurred.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	TXDNES	TX Descriptor Number Error Status Flag	0: No TX Descriptor Number Error 1: TX Descriptor Number Error occurred.	R/W
b15	TSHES	Timestamp Hardware Error Status Flag	0: No Timestamp Hardware Error 1: Timestamp Hardware Error occurred.	R/W
b16	FSES0	Frame Size Error Status Flag 0	0: No Frame Size Error 0 1: Frame Size Error 0 occurred.	R/W
b17	FSES1	Frame Size Error Status Flag 1	0: No Frame Size Error 1 1: Frame Size Error 1 occurred.	R/W
b18	FSES2	Frame Size Error Status Flag 2	0: No Frame Size Error 2 1: Frame Size Error 2 occurred.	R/W
b19	FSES3	Frame Size Error Status Flag 3	0: No Frame Size Error 3 1: Frame Size Error 3 occurred.	R/W
b20	FSES4	Frame Size Error Status Flag 4	0: No Frame Size Error 4 1: Frame Size Error 4 occurred.	R/W
b21	FSES5	Frame Size Error Status Flag 5	0: No Frame Size Error 5 1: Frame Size Error 5 occurred.	R/W
b22	FSES6	Frame Size Error Status Flag 6	0: No Frame Size Error 6 1: Frame Size Error 6 occurred.	R/W
b23	FSES7	Frame Size Error Status Flag 7	0: No Frame Size Error 7 1: Frame Size Error 7 occurred.	R/W
b24	TDFES0	Timestamp Descriptor 0 Full Error Status Flag	0: No Timestamp Descriptor 0 Full Error 1: Timestamp Descriptor 0 Full Error occurred.	R/W
b25	TDFES1	Timestamp Descriptor 1 Full Error Status Flag	0: No Timestamp Descriptor 1 Full Error 1: Timestamp Descriptor 1 Full Error occurred.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b28	TSDNES0	Timestamp Descriptor 0 Number Error Status Flag	0: No Timestamp Descriptor 0 Number Error 1: Timestamp Descriptor 0 Number Error occurred.	R/W
b29	TSDNES1	Timestamp Descriptor 1 Number Error Status Flag	0: No Timestamp Descriptor 1 Number Error 1: Timestamp Descriptor 1 Number Error occurred.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

AES Flag (AXI Error Status Flag)

[Setting condition]

- HW: When an error is detected on an AXI access (xRESP ≠ 00b)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: For reception, if an AXI error happens during descriptor read before a frame/timestamp transfer started, the frame/timestamp is rejected.
- HW: For reception, if an AXI error happens during descriptor read after a frame transfer started, the remaining part of the frame is rejected and DESCR.AXIE is set in the frame corresponding first written back data descriptor, which is FSTART in normal synchronization mode (for GWDCCi.SM[1:0] = 00b).
- HW: For reception, if an AXI error happens during data write, DESCR.AXIE is set in the frame corresponding first written back data descriptor, which is a FSTART or a FSINGLE descriptor in normal synchronization mode (for GWDCCi.SM[1:0] = 00b).
- HW: For reception, if an AXI error happens during RX/TS descriptor write, nothing happens.
- HW: For transmission, if an AXI error happens during descriptor read and no frame is ongoing, corresponding GWTRCi.TSRt bit is cleared and no frame is transmitted.
- HW: For transmission, if an AXI error happens during descriptor read and a frame is ongoing, the ongoing frame is padded with 0h, sent to Forwarding Engine [MFWD] with AXE error set (refer to (c) SAEF Format) and corresponding GWTRCi.TSRt bit is cleared.
- HW: For transmission, if an AXI error happens during data read, the ongoing frame is sent to Forwarding Engine [MFWD] with AXE error set (refer to (c) SAEF Format).
- HW: For transmission, if an AXI error happens during descriptor write, nothing happens.
- SW: System reset.

TSOVFES Bit (Timestamp Overflow Error Status Flag)

[Setting condition]

- HW: When a timestamp is received when the timestamp RAM is full.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: A timestamp is lost.
- SW: Nothing if it is acceptable to lose a timestamp else switch reset.

USMFSES Flag (Under Minimum Frame Size Error Status Flag)

[Setting condition]

- HW: A frame smaller than 32 bytes has been received from CPU for transmission.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The frame is discarded.
- SW: Software error, software should be reviewed.

TFES Flag (TAG Filtering Error Status Flag)

[Setting condition]

- HW: An unauthorized TAG format has been detected. (Refer to (1) TAG Information Extraction).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Frame is sent to Forwarding Engine [MFWD] with local descriptor DESCR.TFE bit set (refer to (3) Frame Saving in Local RAM).
- SW: Software error, software should be reviewed.

SEQES Flag (Sequence Error Status Flag)

[Setting conditions]

- HW: A FMID, FEND has been received before a frame transfer started.
- HW: A descriptor different than FMID, FEND, LINK, and LINKFIX has been received after a frame transfer started.
- HW: An FSTART, FMID, FEND, or FSINGLE descriptor has been received with DESCR.DS set to 0.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: This flag is only available for TX descriptor queues.

[Error recovery]

- HW: If a Sequence Error happens before a frame transfer started, nothing happens.
- HW: If a Sequence Error happens after a frame transfer started, the ongoing frame is padded with 0h and sent to Forwarding Engine [MFWD] with SEQE error set (refer to (c) SAEF Format).
- SW: Software error, software should be reviewed.

TXDNES Flag (TX Descriptor Number Error Status Flag)

[Setting condition]

- HW: GWMDNC.TXDMN[4:0] + 1 descriptor has already been used to process one frame and the frame processing is not finished.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: This flag is only available for TX descriptor queues.

[Error recovery]

- HW: If a TX Descriptor Number Error happens before a frame transfer started, GWTRCi.TSRt bit is cleared.
- HW: If a TX Descriptor Number Error happens after a frame transfer started, the ongoing frame is padded with 0h,

sent to Forwarding Engine [MFWD] with DNE error set (refer to (c) SAEF Format)) and corresponding GWTRCi.TSRt bit is cleared.

- SW: Software error, software should be reviewed.

TSHES Flag (Timestamp Hardware Error Status Flag)

[Setting condition]

- HW: This error happens when timestamps are coming too fast to GWCA to be stored in time, so timestamp are lost even if the timestamp RAM is not full. This error should never happen and tells that the number of captured timestamps is too high or the bus clock clk has a too low frequency.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Timestamp lost.
- SW: No recovery, this error should not happen.

FSESq Flag (Frame Size Error Status Flag q) (q = 0 to 7)

[Setting condition]

- HW: This flag is set when a frame larger than GWRMFSCq.MFS[15:0] has been received for descriptor queue q.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The current frame will be lost. Following data will be processed normally.
- SW: System dependent, cannot be defined here.

TDFESi Flag (Timestamp Descriptor i Full Error Status Flag) (i = 0, 1)

[Setting condition]

- HW: This flag sets when a timestamp has been received for descriptor chain i which is full (read any other descriptor than a FEMPTY_ND, LINKFIX or LINK descriptor).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Timestamp lost.
- SW: Nothing if it is acceptable to lose a timestamp else switch reset.
- SW: AXI is too slow to save all timestamps in the CPU RAM. Less timestamps should be captured or more bandwidth on AXI should be reserved for ESWM.

TSDNESi Flag (Timestamp Descriptor i Number Error Status Flag) (i = 0, 1)

[Setting condition]

- HW: This flag sets when GWMDNC.TSDMN[1:0] + 1 descriptor has already been used to process one timestamp for descriptor chain i and the timestamp processing is not finished.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Timestamp lost
- SW: Software error, software should be reviewed.

38.2.12.2 Error Interrupt Enable Register 0 (GWEIE0)

Address(es): GWCA0.GWEIE0 874C F194h, GWCA0_NS.GWEIE0 074C F194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TSDNEE1	TSDNEE0	—	—	TDFEE1	TDFEE0	FSEE7	FSEE6	FSEE5	FSEE4	FSEE3	FSEE2	FSEE1	FSEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSHEE	TXDNEE	—	SEQEE	TFEE	USMFSEE	TSOVFEE	—	—	—	—	—	—	—	—	AEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AEE	AXI Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b8 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b9	TSOVFEE	Timestamp Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	USMFSEE	Under Minimum Frame Size Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b11	TFEE	TAG Filtering Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b12	SEQEE	Sequence Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	TXDNEE	TX Descriptor Number Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15	TSHEE	Timestamp Hardware Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b16	FSEE0	Frame Size Error Enable 0	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	FSEE1	Frame Size Error Enable 1	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	FSEE2	Frame Size Error Enable 2	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b19	FSEE3	Frame Size Error Enable 3	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b20	FSEE4	Frame Size Error Enable 4	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	FSEE5	Frame Size Error Enable 5	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	FSEE6	Frame Size Error Enable 6	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	FSEE7	Frame Size Error Enable 7	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	TDFEE0	Timestamp Descriptor 0 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	TDFEE1	Timestamp Descriptor 1 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	TSDNEE0	Timestamp Descriptor 0 Number Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b29	TSDNEE1	Timestamp Descriptor 1 Number Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31, b30	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID0 register.

AEE Bit (AXI Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.AED bit will clear this bit.

TSOVFEE Bit (Timestamp Overflow Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TSOVFED bit will clear this bit.

USMFSEE Bit (Under Minimum Frame Size Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.USMFSED bit will clear this bit.

TFEE Bit (TAG Filtering Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TFED bit will clear this bit.

SEQEE Bit (Sequence Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.SEQED bit will clear this bit.

TXDNEE Bit (TX Descriptor Number Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TXDNED bit will clear this bit.

TSHEE Bit (Timestamp Hardware Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TSHED bit will clear this bit.

FSEEq Bit (Frame Size Error Enable q) (q = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.FSEDq bit will clear this bit.

TDFEEi Bit (Timestamp Descriptor i Full Error Enable) (i = 0, 1)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TDFEDi bit will clear this bit.

TSDNEEi Bit (Timestamp Descriptor i Number Error Enable) (i = 0, 1)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID0.TSDNEDi bit will clear this bit.

38.2.12.3 Error Interrupt Disable Register 0 (GWEID0)

Address(es): GWCA0.GWEID0 874C F198h, GWCA0_NS.GWEID0 074C F198h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TSDNED1	TSDNED0	—	—	TDFED1	TDFED0	FSED7	FSED6	FSED5	FSED4	FSED3	FSED2	FSED1	FSED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSHED	TXDNED	—	SEQED	TFED	USMFSED	TSOVFED	—	—	—	—	—	—	—	—	AED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AED	AXI Error Disable	0: Has no effects. 1: Sets GWEIE0.AEE bit to 0.	R/W
b8 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b9	TSOVFED	Timestamp Overflow Error Disable	0: Has no effects. 1: Sets GWEIE0.TSOVFEE bit to 0.	R/W
b10	USMFSED	Under Minimum Frame Size Error Disable	0: Has no effects. 1: Sets GWEIE0.USMFSEE bit to 0.	R/W
b11	TFED	TAG Filtering Error Disable	0: Has no effects. 1: Sets GWEIE0.TFEE bit to 0.	R/W
b12	SEQED	Sequence Error Disable	0: Has no effects. 1: Sets GWEIE0.SEQEE bit to 0.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	TXDNED	TX Descriptor Number Error Disable	0: Has no effects. 1: Sets GWEIE0.TXDNEE bit to 0.	R/W
b15	TSHED	Timestamp Hardware Full Error Disable	0: Has no effects. 1: Sets GWEIE0.TSHEE bit to 0.	R/W
b16	FSED0	Frame Size Error Disable 0	0: Has no effects. 1: Sets GWEIE0.FSEE0 bit to 0.	R/W
b17	FSED1	Frame Size Error Disable 1	0: Has no effects. 1: Sets GWEIE0.FSEE1 bit to 0.	R/W
b18	FSED2	Frame Size Error Disable 2	0: Has no effects. 1: Sets GWEIE0.FSEE2 bit to 0.	R/W
b19	FSED3	Frame Size Error Disable 3	0: Has no effects. 1: Sets GWEIE0.FSEE3 bit to 0.	R/W
b20	FSED4	Frame Size Error Disable 4	0: Has no effects. 1: Sets GWEIE0.FSEE4 bit to 0.	R/W
b21	FSED5	Frame Size Error Disable 5	0: Has no effects. 1: Sets GWEIE0.FSEE5 bit to 0.	R/W
b22	FSED6	Frame Size Error Disable 6	0: Has no effects. 1: Sets GWEIE0.FSEE6 bit to 0.	R/W
b23	FSED7	Frame Size Error Disable 7	0: Has no effects. 1: Sets GWEIE0.FSEE7 bit to 0.	R/W
b24	TDFED0	Timestamp Descriptor 0 Full Error Disable	0: Has no effects. 1: Sets GWEIE0.TDFEE0 bit to 0.	R/W
b25	TDFED1	Timestamp Descriptor 1 Full Error Disable	0: Has no effects. 1: Sets GWEIE0.TDFEE1 bit to 0.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	TSDNED0	Timestamp Descriptor 0 Number Error Disable	0: Has no effects. 1: Sets GWEIE0.TSDNEE0 bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b29	TSDNED1	Timestamp Descriptor 1 Number Error Disable	0: Has no effects. 1: Sets GWEIE0.TSDNEE1 bit to 0.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.12.4 Error Interrupt Status Register 1 (GWEIS1)

Address(es): GWCA0.GWEIS1 874C F1A0h, GWCA0_NS.GWEIS1 074C F1A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSES7	DQSES6	DQSES5	DQSES4	DQSES3	DQSES2	DQSES1	DQSES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOES7	DQOES6	DQOES5	DQOES4	DQOES3	DQOES2	DQOES1	DQOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOES0	Descriptor Queue 0 Overflow Error Status Flag	0: No Descriptor Queue 0 Overflow Error 1: Descriptor Queue 0 Overflow Error occurred.	R/W
b1	DQOES1	Descriptor Queue 1 Overflow Error Status Flag	0: No Descriptor Queue 1 Overflow Error 1: Descriptor Queue 1 Overflow Error occurred.	R/W
b2	DQOES2	Descriptor Queue 2 Overflow Error Status Flag	0: No Descriptor Queue 2 Overflow Error 1: Descriptor Queue 2 Overflow Error occurred.	R/W
b3	DQOES3	Descriptor Queue 3 Overflow Error Status Flag	0: No Descriptor Queue 3 Overflow Error 1: Descriptor Queue 3 Overflow Error occurred.	R/W
b4	DQOES4	Descriptor Queue 4 Overflow Error Status Flag	0: No Descriptor Queue 4 Overflow Error 1: Descriptor Queue 4 Overflow Error occurred.	R/W
b5	DQOES5	Descriptor Queue 5 Overflow Error Status Flag	0: No Descriptor Queue 5 Overflow Error 1: Descriptor Queue 5 Overflow Error occurred.	R/W
b6	DQOES6	Descriptor Queue 6 Overflow Error Status Flag	0: No Descriptor Queue 6 Overflow Error 1: Descriptor Queue 6 Overflow Error occurred.	R/W
b7	DQOES7	Descriptor Queue 7 Overflow Error Status Flag	0: No Descriptor Queue 7 Overflow Error 1: Descriptor Queue 7 Overflow Error occurred.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSES0	Descriptor Queue 0 Security Error Status Flag	0: No Descriptor Queue 0 Security Error 1: Descriptor Queue 0 Security Error occurred.	R/W
b17	DQSES1	Descriptor Queue 1 Security Error Status Flag	0: No Descriptor Queue 1 Security Error 1: Descriptor Queue 1 Security Error occurred.	R/W
b18	DQSES2	Descriptor Queue 2 Security Error Status Flag	0: No Descriptor Queue 2 Security Error 1: Descriptor Queue 2 Security Error occurred.	R/W
b19	DQSES3	Descriptor Queue 3 Security Error Status Flag	0: No Descriptor Queue 3 Security Error 1: Descriptor Queue 3 Security Error occurred.	R/W
b20	DQSES4	Descriptor Queue 4 Security Error Status Flag	0: No Descriptor Queue 4 Security Error 1: Descriptor Queue 4 Security Error occurred.	R/W
b21	DQSES5	Descriptor Queue 5 Security Error Status Flag	0: No Descriptor Queue 5 Security Error 1: Descriptor Queue 5 Security Error occurred.	R/W
b22	DQSES6	Descriptor Queue 6 Security Error Status Flag	0: No Descriptor Queue 6 Security Error 1: Descriptor Queue 6 Security Error occurred.	R/W
b23	DQSES7	Descriptor Queue 7 Security Error Status Flag	0: No Descriptor Queue 7 Security Error 1: Descriptor Queue 7 Security Error occurred.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

DQOESq Flag (Descriptor Queue q Overflow Error Status Flag) (q = 0 to 7)

[Setting condition]

- HW: DQOESq is set when a descriptor is received for queue q when it is full (GWRDQDCq.DQD[9:0] = GWRDQMq.DNQ[9:0]), not disabled (GWRDQC.RDQDq is not set) and GWCA is not going out or out of

OPERATION mode (GWMC.OPC[1:0] = 11b).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Any descriptor received for a full Descriptor queue will not be accepted by GWCA and will not be forwarded to CPU.
- SW: Too many descriptors are received for the corresponding queue. GWRDQDCq.DQD[9:0] setting should be reviewed.

DQSESq Flag (Descriptor Queue q Security Error Status Flag) (q = 0 to 7)

[Setting condition]

- HW: DQSESq is set when a non-secure descriptor is received (FDESCR.SEC is not set [MFWD]) and queue q is a secure queue (GWRDQSC.RDQSLq is set).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Any non-secure descriptor received for secure Descriptor queue will not be accepted by GWCA and will not be forwarded to CPU.
- SW: System dependent, cannot be defined here.

38.2.12.5 Error Interrupt Enable Register 1 (GWEIE1)

Address(es): GWCA0.GWEIE1 874C F1A4h, GWCA0_NS.GWEIE1 074C F1A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSEE7	DQSEE6	DQSEE5	DQSEE4	DQSEE3	DQSEE2	DQSEE1	DQSEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOEE7	DQOEE6	DQOEE5	DQOEE4	DQOEE3	DQOEE2	DQOEE1	DQOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOEE0	Descriptor Queue 0 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DQOEE1	Descriptor Queue 1 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	DQOEE2	Descriptor Queue 2 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	DQOEE3	Descriptor Queue 3 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	DQOEE4	Descriptor Queue 4 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	DQOEE5	Descriptor Queue 5 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b6	DQOEE6	Descriptor Queue 6 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b7	DQOEE7	Descriptor Queue 7 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSEE0	Descriptor Queue 0 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	DQSEE1	Descriptor Queue 1 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	DQSEE2	Descriptor Queue 2 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b19	DQSEE3	Descriptor Queue 3 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b20	DQSEE4	Descriptor Queue 4 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	DQSEE5	Descriptor Queue 5 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	DQSEE6	Descriptor Queue 6 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	DQSEE7	Descriptor Queue 7 Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID1 register.

DQOEEq Bit (Descriptor Queue q Overflow Error Enable) (q = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID1.DQOEDq bit will clear the DQOEEq bit.

DQSEEq Bit (Descriptor Queue q Security Error Enable) (q = 0 to 7)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID1.DQSEDq bit will clear the DQSEEq bit.

38.2.12.6 Error Interrupt Disable Register 1 (GWEID1)

Address(es): GWCA0.GWEID1 874C F1A8h, GWCA0_NS.GWEID1 074C F1A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DQSED7	DQSED6	DQSED5	DQSED4	DQSED3	DQSED2	DQSED1	DQSED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DQOED7	DQOED6	DQOED5	DQOED4	DQOED3	DQOED2	DQOED1	DQOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DQOED0	Descriptor Queue 0 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE0 bit to 0.	R/W
b1	DQOED1	Descriptor Queue 1 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE1 bit to 0.	R/W
b2	DQOED2	Descriptor Queue 2 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE2 bit to 0.	R/W
b3	DQOED3	Descriptor Queue 3 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE3 bit to 0.	R/W
b4	DQOED4	Descriptor Queue 4 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE4 bit to 0.	R/W
b5	DQOED5	Descriptor Queue 5 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE5 bit to 0.	R/W
b6	DQOED6	Descriptor Queue 6 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE6 bit to 0.	R/W
b7	DQOED7	Descriptor Queue 7 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE1.DQOEE7 bit to 0.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DQSED0	Descriptor Queue 0 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE0 bit to 0.	R/W
b17	DQSED1	Descriptor Queue 1 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE1 bit to 0.	R/W
b18	DQSED2	Descriptor Queue 2 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE2 bit to 0.	R/W
b19	DQSED3	Descriptor Queue 3 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE3 bit to 0.	R/W
b20	DQSED4	Descriptor Queue 4 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE4 bit to 0.	R/W
b21	DQSED5	Descriptor Queue 5 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE5 bit to 0.	R/W
b22	DQSED6	Descriptor Queue 6 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE6 bit to 0.	R/W
b23	DQSED7	Descriptor Queue 7 Security Error Disable	0: Has no effects. 1: Sets GWEIE1.DQSEE7 bit to 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.12.7 Error Interrupt Status Register 2i (GWEIS2i) (i = 0, 1)

- GWCA0.GWEIS20

Address(es): GWCA0.GWEIS20 874C F200h, GWCA0_NS.GWEIS20 074C F200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFES31	DFES30	DFES29	DFES28	DFES27	DFES26	DFES25	DFES24	DFES23	DFES22	DFES21	DFES20	DFES19	DFES18	DFES17	DFES16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFES15	DFES14	DFES13	DFES12	DFES11	DFES10	DFES9	DFES8	DFES7	DFES6	DFES5	DFES4	DFES3	DFES2	DFES1	DFES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFES0	Descriptor 0 Full Error Status Flag	0: No Descriptor 0 Full Error 1: Descriptor 0 Full Error occurred.	R/W
b1	DFES1	Descriptor 1 Full Error Status Flag	0: No Descriptor 1 Full Error 1: Descriptor 1 Full Error occurred.	R/W
b2	DFES2	Descriptor 2 Full Error Status Flag	0: No Descriptor 2 Full Error 1: Descriptor 2 Full Error occurred.	R/W
b3	DFES3	Descriptor 3 Full Error Status Flag	0: No Descriptor 3 Full Error 1: Descriptor 3 Full Error occurred.	R/W
b4	DFES4	Descriptor 4 Full Error Status Flag	0: No Descriptor 4 Full Error 1: Descriptor 4 Full Error occurred.	R/W
b5	DFES5	Descriptor 5 Full Error Status Flag	0: No Descriptor 5 Full Error 1: Descriptor 5 Full Error occurred.	R/W
b6	DFES6	Descriptor 6 Full Error Status Flag	0: No Descriptor 6 Full Error 1: Descriptor 6 Full Error occurred.	R/W
b7	DFES7	Descriptor 7 Full Error Status Flag	0: No Descriptor 7 Full Error 1: Descriptor 7 Full Error occurred.	R/W
b8	DFES8	Descriptor 8 Full Error Status Flag	0: No Descriptor 8 Full Error 1: Descriptor 8 Full Error occurred.	R/W
b9	DFES9	Descriptor 9 Full Error Status Flag	0: No Descriptor 9 Full Error 1: Descriptor 9 Full Error occurred.	R/W
b10	DFES10	Descriptor 10 Full Error Status Flag	0: No Descriptor 10 Full Error 1: Descriptor 10 Full Error occurred.	R/W
b11	DFES11	Descriptor 11 Full Error Status Flag	0: No Descriptor 11 Full Error 1: Descriptor 11 Full Error occurred.	R/W
b12	DFES12	Descriptor 12 Full Error Status Flag	0: No Descriptor 12 Full Error 1: Descriptor 12 Full Error occurred.	R/W
b13	DFES13	Descriptor 13 Full Error Status Flag	0: No Descriptor 13 Full Error 1: Descriptor 13 Full Error occurred.	R/W
b14	DFES14	Descriptor 14 Full Error Status Flag	0: No Descriptor 14 Full Error 1: Descriptor 14 Full Error occurred.	R/W
b15	DFES15	Descriptor 15 Full Error Status Flag	0: No Descriptor 15 Full Error 1: Descriptor 15 Full Error occurred.	R/W
b16	DFES16	Descriptor 16 Full Error Status Flag	0: No Descriptor 16 Full Error 1: Descriptor 16 Full Error occurred.	R/W
b17	DFES17	Descriptor 17 Full Error Status Flag	0: No Descriptor 17 Full Error 1: Descriptor 17 Full Error occurred.	R/W
b18	DFES18	Descriptor 18 Full Error Status Flag	0: No Descriptor 18 Full Error 1: Descriptor 18 Full Error occurred.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	DFES19	Descriptor 19 Full Error Status Flag	0: No Descriptor 19 Full Error 1: Descriptor 19 Full Error occurred.	R/W
b20	DFES20	Descriptor 20 Full Error Status Flag	0: No Descriptor 20 Full Error 1: Descriptor 20 Full Error occurred.	R/W
b21	DFES21	Descriptor 21 Full Error Status Flag	0: No Descriptor 21 Full Error 1: Descriptor 21 Full Error occurred.	R/W
b22	DFES22	Descriptor 22 Full Error Status Flag	0: No Descriptor 22 Full Error 1: Descriptor 22 Full Error occurred.	R/W
b23	DFES23	Descriptor 23 Full Error Status Flag	0: No Descriptor 23 Full Error 1: Descriptor 23 Full Error occurred.	R/W
b24	DFES24	Descriptor 24 Full Error Status Flag	0: No Descriptor 24 Full Error 1: Descriptor 24 Full Error occurred.	R/W
b25	DFES25	Descriptor 25 Full Error Status Flag	0: No Descriptor 25 Full Error 1: Descriptor 25 Full Error occurred.	R/W
b26	DFES26	Descriptor 26 Full Error Status Flag	0: No Descriptor 26 Full Error 1: Descriptor 26 Full Error occurred.	R/W
b27	DFES27	Descriptor 27 Full Error Status Flag	0: No Descriptor 27 Full Error 1: Descriptor 27 Full Error occurred.	R/W
b28	DFES28	Descriptor 28 Full Error Status Flag	0: No Descriptor 28 Full Error 1: Descriptor 28 Full Error occurred.	R/W
b29	DFES29	Descriptor 29 Full Error Status Flag	0: No Descriptor 29 Full Error 1: Descriptor 29 Full Error occurred.	R/W
b30	DFES30	Descriptor 30 Full Error Status Flag	0: No Descriptor 30 Full Error 1: Descriptor 30 Full Error occurred.	R/W
b31	DFES31	Descriptor 31 Full Error Status Flag	0: No Descriptor 31 Full Error 1: Descriptor 31 Full Error occurred.	R/W

- GWCA0.GWEIS21

Address(es): GWCA0.GWEIS21 874C F210h, GWCA0_NS.GWEIS21 074C F210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFES63	DFES62	DFES61	DFES60	DFES59	DFES58	DFES57	DFES56	DFES55	DFES54	DFES53	DFES52	DFES51	DFES50	DFES49	DFES48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFES47	DFES46	DFES45	DFES44	DFES43	DFES42	DFES41	DFES40	DFES39	DFES38	DFES37	DFES36	DFES35	DFES34	DFES33	DFES32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFES32	Descriptor 32 Full Error Status Flag	0: No Descriptor 32 Full Error 1: Descriptor 32 Full Error occurred.	R/W
b1	DFES33	Descriptor 33 Full Error Status Flag	0: No Descriptor 33 Full Error 1: Descriptor 33 Full Error occurred.	R/W
b2	DFES34	Descriptor 34 Full Error Status Flag	0: No Descriptor 34 Full Error 1: Descriptor 34 Full Error occurred.	R/W
b3	DFES35	Descriptor 35 Full Error Status Flag	0: No Descriptor 35 Full Error 1: Descriptor 35 Full Error occurred.	R/W
b4	DFES36	Descriptor 36 Full Error Status Flag	0: No Descriptor 36 Full Error 1: Descriptor 36 Full Error occurred.	R/W
b5	DFES37	Descriptor 37 Full Error Status Flag	0: No Descriptor 37 Full Error 1: Descriptor 37 Full Error occurred.	R/W
b6	DFES38	Descriptor 38 Full Error Status Flag	0: No Descriptor 38 Full Error 1: Descriptor 38 Full Error occurred.	R/W
b7	DFES39	Descriptor 39 Full Error Status Flag	0: No Descriptor 39 Full Error 1: Descriptor 39 Full Error occurred.	R/W
b8	DFES40	Descriptor 40 Full Error Status Flag	0: No Descriptor 40 Full Error 1: Descriptor 40 Full Error occurred.	R/W
b9	DFES41	Descriptor 41 Full Error Status Flag	0: No Descriptor 41 Full Error 1: Descriptor 41 Full Error occurred.	R/W
b10	DFES42	Descriptor 42 Full Error Status Flag	0: No Descriptor 42 Full Error 1: Descriptor 42 Full Error occurred.	R/W
b11	DFES43	Descriptor 43 Full Error Status Flag	0: No Descriptor 43 Full Error 1: Descriptor 43 Full Error occurred.	R/W
b12	DFES44	Descriptor 44 Full Error Status Flag	0: No Descriptor 44 Full Error 1: Descriptor 44 Full Error occurred.	R/W
b13	DFES45	Descriptor 45 Full Error Status Flag	0: No Descriptor 45 Full Error 1: Descriptor 45 Full Error occurred.	R/W
b14	DFES46	Descriptor 46 Full Error Status Flag	0: No Descriptor 46 Full Error 1: Descriptor 46 Full Error occurred.	R/W
b15	DFES47	Descriptor 47 Full Error Status Flag	0: No Descriptor 47 Full Error 1: Descriptor 47 Full Error occurred.	R/W
b16	DFES48	Descriptor 48 Full Error Status Flag	0: No Descriptor 48 Full Error 1: Descriptor 48 Full Error occurred.	R/W
b17	DFES49	Descriptor 49 Full Error Status Flag	0: No Descriptor 49 Full Error 1: Descriptor 49 Full Error occurred.	R/W
b18	DFES50	Descriptor 50 Full Error Status Flag	0: No Descriptor 50 Full Error 1: Descriptor 50 Full Error occurred.	R/W
b19	DFES51	Descriptor 51 Full Error Status Flag	0: No Descriptor 51 Full Error 1: Descriptor 51 Full Error occurred.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	DFES52	Descriptor 52 Full Error Status Flag	0: No Descriptor 52 Full Error 1: Descriptor 52 Full Error occurred.	R/W
b21	DFES53	Descriptor 53 Full Error Status Flag	0: No Descriptor 53 Full Error 1: Descriptor 53 Full Error occurred.	R/W
b22	DFES54	Descriptor 54 Full Error Status Flag	0: No Descriptor 54 Full Error 1: Descriptor 54 Full Error occurred.	R/W
b23	DFES55	Descriptor 55 Full Error Status Flag	0: No Descriptor 55 Full Error 1: Descriptor 55 Full Error occurred.	R/W
b24	DFES56	Descriptor 56 Full Error Status Flag	0: No Descriptor 56 Full Error 1: Descriptor 56 Full Error occurred.	R/W
b25	DFES57	Descriptor 57 Full Error Status Flag	0: No Descriptor 57 Full Error 1: Descriptor 57 Full Error occurred.	R/W
b26	DFES58	Descriptor 58 Full Error Status Flag	0: No Descriptor 58 Full Error 1: Descriptor 58 Full Error occurred.	R/W
b27	DFES59	Descriptor 59 Full Error Status Flag	0: No Descriptor 59 Full Error 1: Descriptor 59 Full Error occurred.	R/W
b28	DFES60	Descriptor 60 Full Error Status Flag	0: No Descriptor 60 Full Error 1: Descriptor 60 Full Error occurred.	R/W
b29	DFES61	Descriptor 61 Full Error Status Flag	0: No Descriptor 61 Full Error 1: Descriptor 61 Full Error occurred.	R/W
b30	DFES62	Descriptor 62 Full Error Status Flag	0: No Descriptor 62 Full Error 1: Descriptor 62 Full Error occurred.	R/W
b31	DFES63	Descriptor 63 Full Error Status Flag	0: No Descriptor 63 Full Error 1: Descriptor 63 Full Error occurred.	R/W

DFESt Flag (Descriptor t Full Error Status Flag) (t = 0 to 63)

[Setting condition]

- HW: When a frame has been received for descriptor queue t which is full (read a descriptor different than FEMPTY, FEMPTY_IS, FEMPTY_IC, FEMPTY_ND, FEMPTY_START, FEMPTY_MID, FEMPTY_END, LINKFIX and LINK). This flag is set at the timing the frame last descriptor write-back should have been completed.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- This flag is only valid for RX queues.

[Error recovery]

- HW: Set of DESCR.ERR bit in RX descriptor if a part of the frame has been already sent.
- HW: Current frame or remaining part of the frame will be discarded, and next frames will be processed normally (as long as software does not insert new descriptors for reception the queue will continue overflowing).
- SW: Discard data which have DESCR.ERR bit set.

38.2.12.8 Error Interrupt Enable Register 2i (GWEIE2i) (i = 0, 1)

- GWCA0.GWEIE20

Address(es): GWCA0.GWEIE20 874C F204h, GWCA0_NS.GWEIE20 074C F204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFEE31	DFEE30	DFEE29	DFEE28	DFEE27	DFEE26	DFEE25	DFEE24	DFEE23	DFEE22	DFEE21	DFEE20	DFEE19	DFEE18	DFEE17	DFEE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFEE15	DFEE14	DFEE13	DFEE12	DFEE11	DFEE10	DFEE9	DFEE8	DFEE7	DFEE6	DFEE5	DFEE4	DFEE3	DFEE2	DFEE1	DFEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFEE0	Descriptor 0 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DFEE1	Descriptor 1 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	DFEE2	Descriptor 2 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	DFEE3	Descriptor 3 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	DFEE4	Descriptor 4 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	DFEE5	Descriptor 5 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b6	DFEE6	Descriptor 6 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b7	DFEE7	Descriptor 7 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b8	DFEE8	Descriptor 8 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b9	DFEE9	Descriptor 9 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	DFEE10	Descriptor 10 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b11	DFEE11	Descriptor 11 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b12	DFEE12	Descriptor 12 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b13	DFEE13	Descriptor 13 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b14	DFEE14	Descriptor 14 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15	DFEE15	Descriptor 15 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b16	DFEE16	Descriptor 16 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	DFEE17	Descriptor 17 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	DFEE18	Descriptor 18 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b19	DFEE19	Descriptor 19 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b20	DFEE20	Descriptor 20 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	DFEE21	Descriptor 21 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	DFEE22	Descriptor 22 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	DFEE23	Descriptor 23 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	DFEE24	Descriptor 24 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	DFEE25	Descriptor 25 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b26	DFEE26	Descriptor 26 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27	DFEE27	Descriptor 27 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b28	DFEE28	Descriptor 28 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b29	DFEE29	Descriptor 29 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b30	DFEE30	Descriptor 30 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31	DFEE31	Descriptor 31 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID20 register.

- GWCA0.GWEIE21

Address(es): GWCA0.GWEIE21 874C F214h, GWCA0_NS.GWEIE21 074C F214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFEE63	DFEE62	DFEE61	DFEE60	DFEE59	DFEE58	DFEE57	DFEE56	DFEE55	DFEE54	DFEE53	DFEE52	DFEE51	DFEE50	DFEE49	DFEE48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFEE47	DFEE46	DFEE45	DFEE44	DFEE43	DFEE42	DFEE41	DFEE40	DFEE39	DFEE38	DFEE37	DFEE36	DFEE35	DFEE34	DFEE33	DFEE32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFEE32	Descriptor 32 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DFEE33	Descriptor 33 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	DFEE34	Descriptor 34 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	DFEE35	Descriptor 35 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b4	DFEE36	Descriptor 36 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b5	DFEE37	Descriptor 37 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b6	DFEE38	Descriptor 38 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b7	DFEE39	Descriptor 39 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b8	DFEE40	Descriptor 40 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b9	DFEE41	Descriptor 41 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b10	DFEE42	Descriptor 42 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b11	DFEE43	Descriptor 43 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b12	DFEE44	Descriptor 44 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b13	DFEE45	Descriptor 45 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b14	DFEE46	Descriptor 46 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15	DFEE47	Descriptor 47 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b16	DFEE48	Descriptor 48 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	DFEE49	Descriptor 49 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b18	DFEE50	Descriptor 50 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b19	DFEE51	Descriptor 51 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b20	DFEE52	Descriptor 52 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b21	DFEE53	Descriptor 53 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b22	DFEE54	Descriptor 54 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b23	DFEE55	Descriptor 55 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b24	DFEE56	Descriptor 56 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b25	DFEE57	Descriptor 57 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b26	DFEE58	Descriptor 58 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b27	DFEE59	Descriptor 59 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b28	DFEE60	Descriptor 60 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b29	DFEE61	Descriptor 61 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b30	DFEE62	Descriptor 62 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31	DFEE63	Descriptor 63 Full Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID21 register.

DFEEt Bit (Descriptor t Full Error Enable) (t = 0 to 63)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to GWEID2i.DFEDt bit will clear this bit.

38.2.12.9 Error Interrupt Disable Register 2i (GWEID2i) (i = 0, 1)

- GWCA0.GWEID20

Address(es): GWCA0.GWEID20 874C F208h, GWCA0_NS.GWEID20 074C F208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFED31	DFED30	DFED29	DFED28	DFED27	DFED26	DFED25	DFED24	DFED23	DFED22	DFED21	DFED20	DFED19	DFED18	DFED17	DFED16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFED15	DFED14	DFED13	DFED12	DFED11	DFED10	DFED9	DFED8	DFED7	DFED6	DFED5	DFED4	DFED3	DFED2	DFED1	DFED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFED0	Descriptor 0 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE0 bit to 0.	R/W
b1	DFED1	Descriptor 1 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE1 bit to 0.	R/W
b2	DFED2	Descriptor 2 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE2 bit to 0.	R/W
b3	DFED3	Descriptor 3 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE3 bit to 0.	R/W
b4	DFED4	Descriptor 4 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE4 bit to 0.	R/W
b5	DFED5	Descriptor 5 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE5 bit to 0.	R/W
b6	DFED6	Descriptor 6 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE6 bit to 0.	R/W
b7	DFED7	Descriptor 7 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE7 bit to 0.	R/W
b8	DFED8	Descriptor 8 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE8 bit to 0.	R/W
b9	DFED9	Descriptor 9 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE9 bit to 0.	R/W
b10	DFED10	Descriptor 10 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE10 bit to 0.	R/W
b11	DFED11	Descriptor 11 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE11 bit to 0.	R/W
b12	DFED12	Descriptor 12 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE12 bit to 0.	R/W
b13	DFED13	Descriptor 13 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE13 bit to 0.	R/W
b14	DFED14	Descriptor 14 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE14 bit to 0.	R/W
b15	DFED15	Descriptor 15 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE15 bit to 0.	R/W
b16	DFED16	Descriptor 16 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE16 bit to 0.	R/W
b17	DFED17	Descriptor 17 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE17 bit to 0.	R/W
b18	DFED18	Descriptor 18 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE18 bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	DFED19	Descriptor 19 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE19 bit to 0.	R/W
b20	DFED20	Descriptor 20 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE20 bit to 0.	R/W
b21	DFED21	Descriptor 21 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE21 bit to 0.	R/W
b22	DFED22	Descriptor 22 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE22 bit to 0.	R/W
b23	DFED23	Descriptor 23 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE23 bit to 0.	R/W
b24	DFED24	Descriptor 24 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE24 bit to 0.	R/W
b25	DFED25	Descriptor 25 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE25 bit to 0.	R/W
b26	DFED26	Descriptor 26 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE26 bit to 0.	R/W
b27	DFED27	Descriptor 27 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE27 bit to 0.	R/W
b28	DFED28	Descriptor 28 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE28 bit to 0.	R/W
b29	DFED29	Descriptor 29 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE29 bit to 0.	R/W
b30	DFED30	Descriptor 30 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE30 bit to 0.	R/W
b31	DFED31	Descriptor 31 Full Error Disable	0: Has no effects. 1: Sets GWEIE20.DFEE31 bit to 0.	R/W

- GWCA0.GWEID21

Address(es): GWCA0.GWEID21 874C F218h, GWCA0_NS.GWEID21 074C F218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFED63	DFED62	DFED61	DFED60	DFED59	DFED58	DFED57	DFED56	DFED55	DFED54	DFED53	DFED52	DFED51	DFED50	DFED49	DFED48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DFED47	DFED46	DFED45	DFED44	DFED43	DFED42	DFED41	DFED40	DFED39	DFED38	DFED37	DFED36	DFED35	DFED34	DFED33	DFED32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFED32	Descriptor 32 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE32 bit to 0.	R/W
b1	DFED33	Descriptor 33 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE33 bit to 0.	R/W
b2	DFED34	Descriptor 34 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE34 bit to 0.	R/W
b3	DFED35	Descriptor 35 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE35 bit to 0.	R/W
b4	DFED36	Descriptor 36 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE36 bit to 0.	R/W
b5	DFED37	Descriptor 37 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE37 bit to 0.	R/W
b6	DFED38	Descriptor 38 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE38 bit to 0.	R/W
b7	DFED39	Descriptor 39 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE39 bit to 0.	R/W
b8	DFED40	Descriptor 40 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE40 bit to 0.	R/W
b9	DFED41	Descriptor 41 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE41 bit to 0.	R/W
b10	DFED42	Descriptor 42 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE42 bit to 0.	R/W
b11	DFED43	Descriptor 43 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE43 bit to 0.	R/W
b12	DFED44	Descriptor 44 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE44 bit to 0.	R/W
b13	DFED45	Descriptor 45 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE45 bit to 0.	R/W
b14	DFED46	Descriptor 46 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE46 bit to 0.	R/W
b15	DFED47	Descriptor 47 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE47 bit to 0.	R/W
b16	DFED48	Descriptor 48 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE48 bit to 0.	R/W
b17	DFED49	Descriptor 49 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE49 bit to 0.	R/W
b18	DFED50	Descriptor 50 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE50 bit to 0.	R/W
b19	DFED51	Descriptor 51 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE51 bit to 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	DFED52	Descriptor 52 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE52 bit to 0.	R/W
b21	DFED53	Descriptor 53 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE53 bit to 0.	R/W
b22	DFED54	Descriptor 54 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE54 bit to 0.	R/W
b23	DFED55	Descriptor 55 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE55 bit to 0.	R/W
b24	DFED56	Descriptor 56 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE56 bit to 0.	R/W
b25	DFED57	Descriptor 57 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE57 bit to 0.	R/W
b26	DFED58	Descriptor 58 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE58 bit to 0.	R/W
b27	DFED59	Descriptor 59 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE59 bit to 0.	R/W
b28	DFED60	Descriptor 60 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE60 bit to 0.	R/W
b29	DFED61	Descriptor 61 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE61 bit to 0.	R/W
b30	DFED62	Descriptor 62 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE62 bit to 0.	R/W
b31	DFED63	Descriptor 63 Full Error Disable	0: Has no effects. 1: Sets GWEIE21.DFEE63 bit to 0.	R/W

38.2.12.10 Error Interrupt Status Register 3 (GWEIS3)

Address(es): GWCA0.GWEIS3 874C F280h, GWCA0_NS.GWEIS3 074C F280h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IAOES3	IAOES2	IAOES1	IAOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IAOES0	Incremental Area 0 Overflow Error Status Flag	0: No Incremental Area 0 Overflow Error 1: Incremental Area 0 Overflow Error occurred.	R/W
b1	IAOES1	Incremental Area 1 Overflow Error Status Flag	0: No Incremental Area 1 Overflow Error 1: Incremental Area 1 Overflow Error occurred.	R/W
b2	IAOES2	Incremental Area 2 Overflow Error Status Flag	0: No Incremental Area 2 Overflow Error 1: Incremental Area 2 Overflow Error occurred.	R/W
b3	IAOES3	Incremental Area 3 Overflow Error Status Flag	0: No Incremental Area 3 Overflow Error 1: Incremental Area 3 Overflow Error occurred.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

IAOESt Flag (Incremental Area t Overflow Error Status Flag) (t = 0 to 3)

[Setting condition]

- HW: When incremental area t overflows, for further information refer to section (3), Single Page Incremental Data Reception. This flag is only set when GWIDAUASt.IDAUAS[23:0] crosses GWIDASMi.IDAS[23:0] value (if incremental area is already overflowing this flag would not be set again except if GWIDAUASt.IDAUAS[23:0] overflows).

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: This register is only valid for RX incremental queues.

[Error recovery]

- HW: The data coming for this incremental area will still be written.
- SW: Set a new incremental area.

38.2.12.11 Error Interrupt Enable Register 3 (GWEIE3)

Address(es): GWCA0.GWEIE3 874C F284h, GWCA0_NS.GWEIE3 074C F284h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IAOEE3	IAOEE2	IAOEE1	IAOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IAOEE0	Incremental Area 0 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	IAOEE1	Incremental Area 1 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b2	IAOEE2	Incremental Area 2 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b3	IAOEE3	Incremental Area 3 Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID3 register.

IAOEEt Bit (Incremental Area t Overflow Error Enable) (t = 0 to 3)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID3.IAOEDt bit will clear this bit.

38.2.12.12 Error Interrupt Disable Register 3 (GWEID3)

Address(es): GWCA0.GWEID3 874C F288h, GWCA0_NS.GWEID3 074C F288h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IAOED3	IAOED2	IAOED1	IAOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IAOED0	Incremental Area 0 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE3.IAOEE0 bit to 0.	R/W
b1	IAOED1	Incremental Area 1 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE3.IAOEE1 bit to 0.	R/W
b2	IAOED2	Incremental Area 2 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE3.IAOEE2 bit to 0.	R/W
b3	IAOED3	Incremental Area 3 Overflow Error Disable	0: Has no effects. 1: Sets GWEIE3.IAOEE3 bit to 0.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.12.13 Error Interrupt Status Register 4 (GWEIS4)

Address(es): GWCA0.GWEIS4 874C F290h, GWCA0_NS.GWEIS4 074C F290h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DSECN[5:0]						—	—	—	—	—	—	DSEIOS	DSES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DSSECN[5:0]						—	—	—	—	—	—	DSSEIOS	DSSES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSSES	Descriptor Security Error Status Flag	0: No Descriptor Security Error 1: Descriptor Security Error occurred.	R/W
b1	DSSEIOS	Descriptor Security Error Interrupt Overflow Status Flag	0: No Descriptor Security Error Interrupt Overflow 1: Descriptor Security Error Interrupt Overflow occurred.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13 to b8	DSSECN[5:0]	Descriptor Security Error Chain Number		R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DSES	Data Size Error Status Flag	0: No Data Size Error 1: Data Size Error occurred.	R/W
b17	DSEIOS	Data Size Error Interrupt Overflow Status Flag	0: No Data Size Error Interrupt Overflow 1: Data Size Error Interrupt Overflow occurred.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	DSECN[5:0]	Data Size Error Chain Number		R
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

DSSES Flag (Descriptor Security Error Status Flag)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Setting condition]

- HW: During reception, when an unsecure descriptor has been received for a secure queue (GWDCCi.SL set).

[Error recovery]

- HW: During reception, the current frame will be lost. Following data will be processed normally.
- SW: Secure CPU should read the unsecure forwarding engine registers to find which entry is trying to use a secure queue for unsecure traffic and suppress it.
- SW: Secure CPU could block access to switch from unsecure CPU.

DSSEIOS Flag (Descriptor Security Error Interrupt Overflow Status Flag)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Setting condition]

- When a Descriptor Security Error occurs and GWEIS4.DSSES is already set to one. In this case, the descriptor

chain number in which the new error occurred is lost.

[Restriction]

- HW: This flag is only valid for RX queues.

[Error recovery]

- SW: Switch reset.

DSSECN[5:0] Bits (Descriptor Security Error Chain Number)

[Restriction]

- HW: This flag is only valid for RX queues.

[Clearing condition]

- HW: Being in RESET mode will clear this flag.

[Update condition]

- This field is update to the descriptor chain number in which a Descriptor Security Error occurred when GWEIS4.DSSES is being set.

DSES Flag (Data Size Error Status Flag)

[Setting condition]

- HW: When a frame has been received with an unexpected size, for more information refer to section (2), Size-Controlled Data Reception. This flag is set after the frame last descriptor write back is completed.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: This flag is only valid for RX queues.

[Error recovery]

- HW: Set of DESCR.DSE bit in FSTART RX descriptor.
- HW: If the frame was an undersized frame the frame is written with the normal descriptor sequence (FSINGLE or FSTART/ FEND or FSTART/n*FMID/ FEND)
- HW: If the frame was an undersized frame, when a new frame comes for the corresponding queue, the AXI master will skip all following descriptors until a empty descriptor which is not FEMPTY_MID or FEMPTY_END (at start of a frame, FEMPTY_MID and FEMPTY_END descriptors are always ignored).
- If the frame is an oversized frame, the frame will be truncated.
- SW: Discard data which have DESCR.DSE bit set.
- SW: Because DESCR.DSE is set in RX descriptors with a size error, it is not mandatory to read this flag for error recovery.

DSEIOS Flag (Data Size Error Interrupt Overflow Status Flag)

[Setting condition]

- When a Data Size Error occurs and GWEIS4.DSES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Restriction]

- HW: This flag is only valid for RX queues.

[Error recovery]

- SW: Switch reset.

DSECN[5:0] Bits (Data Size Error Chain Number)

[Clearing condition]

- HW: Being in RESET mode will clear this flag.

[Update condition]

- This field is update to the descriptor chain number in which a Data Size Error occurred when GWEIS4.DSES is being set.

[Restriction]

- HW: These bits are only valid for RX queues.

38.2.12.14 Error Interrupt Enable Register 4 (GWEIE4)

Address(es): GWCA0.GWEIE4 874C F294h, GWCA0_NS.GWEIE4 074C F294h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEIOE	DSEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSEIOE	DSSEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSSEE	Descriptor Security Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DSSEIOE	Descriptor Security Error Interrupt Overflow Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DSEE	Data Size Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	DSEIOE	Data Size Error Interrupt Overflow Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID4 register.

DSSEE Bit (Descriptor Security Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID4.DSSED bit will clear this bit.

DSSEIOE Bit (Descriptor Security Error Interrupt Overflow Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID4.DSSEIOD bit will clear this bit.

DSEE Bit (Data Size Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID4.DSED bit will clear this bit.

DSEIOE Bit (Data Size Error Interrupt Overflow Interrupt Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID4.DSEIOD bit will clear this bit.

38.2.12.15 Error Interrupt Disable Register 4 (GWEID4)

Address(es): GWCA0.GWEID4 874C F298h, GWCA0_NS.GWEID4 074C F298h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEIOD	DSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSEIOD	DSSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSSED	Descriptor Security Error Disable	0: Has no effects. 1: Sets GWEIE4.DSSEE bit to 0.	R/W
b1	DSSEIOD	Descriptor Security Error Interrupt Overflow Disable	0: Has no effects. 1: Sets GWEIE4.DSSEIOE bit to 0.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DSED	Data Size Error Disable	0: Has no effects. 1: Sets GWEIE4.DSEE bit to 0.	R/W
b17	DSEIOD	Data Size Error Interrupt Overflow Disable	0: Has no effects. 1: Sets GWEIE4.DSEIOE bit to 0.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.12.16 Error Interrupt Status Register 5 (GWEIS5)

Address(es): GWCA0.GWEIS5 874C F2A0h, GWCA0_NS.GWEIS5 074C F2A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	RXDNECN[5:0]						—	—	—	—	—	—	RXDNEIOS	RXDNES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DCTECN[5:0]						—	—	—	—	—	—	DCTEIOS	DCTES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCTES	Descriptor Chain Type Error Status Flag	0: No Descriptor Chain Type Error 1: Descriptor Chain Type Error occurred.	R/W
b1	DCTEIOS	Descriptor Chain Type Error Interrupt Overflow Status Flag	0: No Descriptor Chain Type Error Interrupt Overflow 1: Descriptor Chain Type Error Interrupt Overflow occurred.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13 to b8	DCTECN[5:0]	Descriptor Chain Type Error Chain Number		R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RXDNES	RX Descriptor Number Error Status Flag	0: No RX Descriptor Number Error 1: RX Descriptor Number Error occurred.	R/W
b17	RXDNEIOS	RX Descriptor Number Error Interrupt Overflow Status Flag	0: No RX Descriptor Number Error Interrupt Overflow 1: RX Descriptor Number Error Interrupt Overflow occurred.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	RXDNECN[5:0]	RX Descriptor Number Error Chain Number		R
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

DCTES Flag (Descriptor Chain Type Error Status Flag)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Setting condition]

- HW: When a descriptor is received for a transmission descriptor queue (GWDCCt.DQT set).

[Error recovery]

- HW: When a descriptor is received for a transmission descriptor queue, the corresponding frame is discarded.
- SW: It is a software error. Software should be reviewed.

DCTEIOS Flag (Descriptor Chain Type Error Interrupt Overflow Status Flag)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Setting condition]

- Descriptor Chain Type Error occurs and GWEIS5.DCTES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Error recovery]

- SW: Switch reset.

DCTECN[5:0] Bits (Descriptor Chain Type Error Chain Number)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update condition]

- This field is update to the descriptor chain number in which a Descriptor Chain Type Error occurred when GWEIS5.DCTES is being set.

RXDNES Flag (RX Descriptor Number Error Status Flag)

[Setting condition]

- HW: GWMDNC.RXDMN[4:0] + 1 descriptor has already been used to process one frame and the frame processing is not finished.

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: Set of DESCR.ERR bit in RX descriptor if a part of the frame has been already sent.
- HW: Current frame or remaining part of the frame will be discarded, and next frames will be processed normally.
- SW: Discard data which have DESCR.ERR bit set.
- SW: Software error, software should be reviewed.

RXDNEIOS Flag (RX Descriptor Number Error Interrupt Overflow Status Flag)

[Clearing conditions]

- HW: Being in RESET mode will clear this flag.
- SW: Writing 1 to this flag will clear it.

[Setting condition]

- RX Descriptor Number Error occurs and GWEIS5.RXDNES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Error recovery]

- SW: Switch reset.

RXDNECN[5:0] Bits (RX Descriptor Number Error Chain Number)

[Clearing condition]

- HW: Being in RESET mode will clear these bits.

[Update condition]

- This field is update to the descriptor chain number in which a RX Descriptor Number Error occurred when GWEIS5.RXDNES is being set.

38.2.12.17 Error Interrupt Enable Register 5 (GWEIE5)

Address(es): GWCA0.GWEIE5 874C F2A4h, GWCA0_NS.GWEIE5 074C F2A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXDNEIOE	RXDNEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCTEIOE	DCTEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCTEE	Descriptor Chain Type Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	DCTEIOE	Descriptor Chain Type Error Interrupt Overflow Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RXDNEE	RX Descriptor Number Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	RXDNEIOE	RX Descriptor Number Error Interrupt Overflow Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the GWEID5 register.

DCTEE Bit (Descriptor Chain Type Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID5.DCTED bit will clear this bit.

DCTEIOE Bit (Descriptor Chain Type Error Interrupt Overflow Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID5.DCTEIOD bit will clear this bit.

RXDNEE Bit (RX Descriptor Number Error Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID5.RXDNED bit will clear this bit.

RXDNEIOE Bit (RX Descriptor Number Error Interrupt Overflow Enable)

[Setting condition]

- Writing 1 to this bit will set it.

[Clearing condition]

- Writing 1 to the GWEID5.RXDNEIOD bit will clear this bit.

38.2.12.18 Error Interrupt Disable Register 5 (GWEID5)

Address(es): GWCA0.GWEID5 874C F2A8h, GWCA0_NS.GWEID5 074C F2A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXDNEIOD	RXDNED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCTEIOD	DCTED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCTED	Descriptor Chain Type Error Disable	0: Has no effects. 1: Sets GWEIE5.DCTEE bit to 0.	R/W
b1	DCTEIOD	Descriptor Chain Type Error Interrupt Overflow Disable	0: Has no effects. 1: Sets GWEIE5.DCTEIOE bit to 0.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RXDNED	RX Descriptor Number Error Disable	0: Has no effects. 1: Sets GWEIE5.RXDNEE bit to 0.	R/W
b17	RXDNEIOD	RX Descriptor Number Error Interrupt Overflow Disable	0: Has no effects. 1: Sets GWEIE5.RXDNEIOE bit to 0.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.3 Operations

38.3.1 AXI Master General Information

In this section is described the knowledge required to utilize the GWCA AXI master. It contains the AXI address table handling, the AXI descriptor general description, and the AXI IDs utilization description.

38.3.1.1 AXI Descriptor Queue Mapping

The AXI descriptor queue mapping is described along with its mapping to the INTER_TX interface, the incremental queues and the rate limiters in Figure 38.2. The global rate limiter is not represented because it limits throughput for all the TX queues.

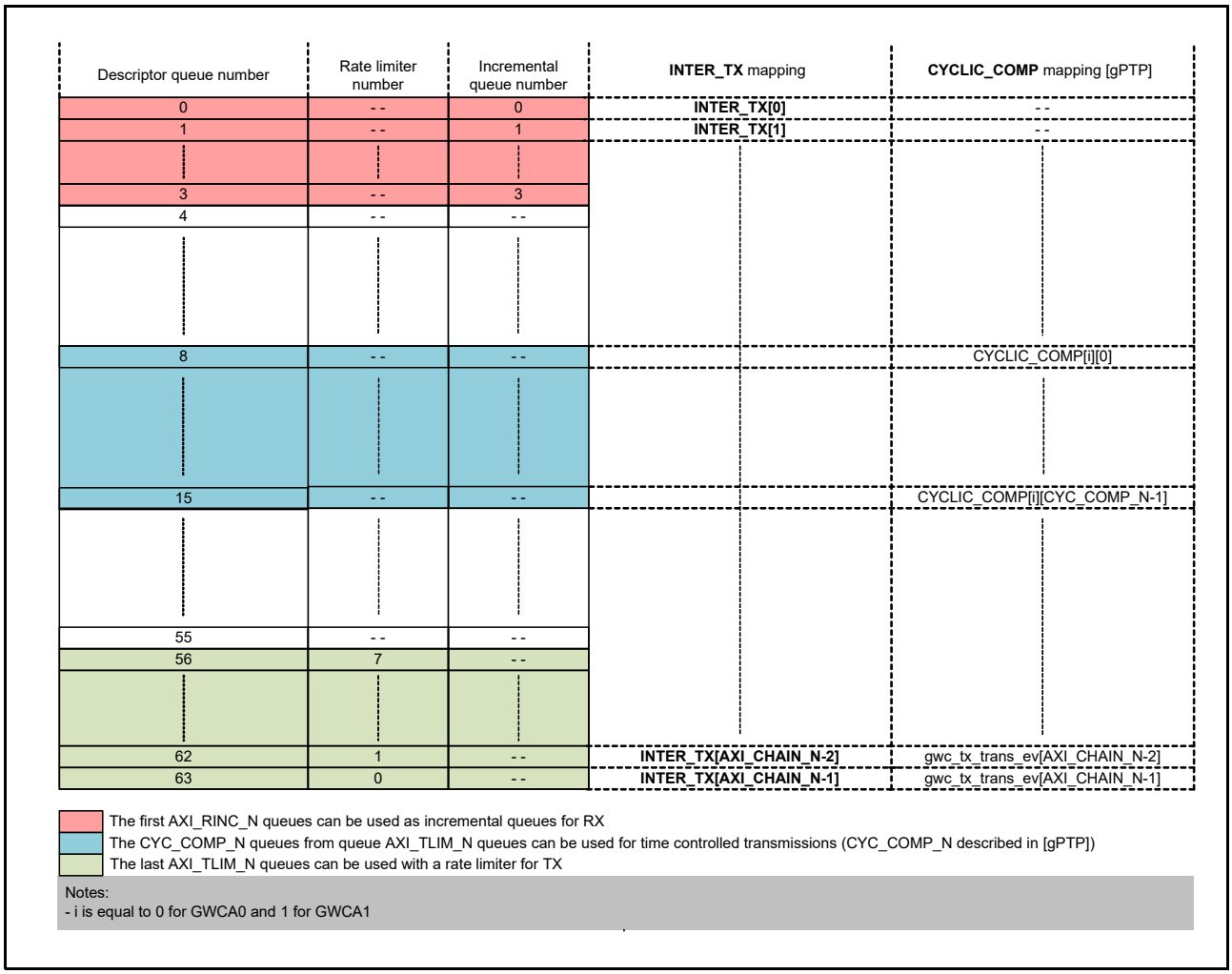


Figure 38.2 AXI Descriptor Queue Mapping

38.3.1.2 AXI Descriptor General Description

(1) General Formats

AXI descriptor general formats are described in Figure 38.3, Figure 38.4, Figure 38.5 and Figure 38.6. The fields are explained in Table 38.3. The fields utilization will be explained per transaction type (Data reception, Data transmission and Timestamp reception).

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

Figure 38.3 Basic Descriptor (GWDCCi.ETS = 0 and GWDCCi.EDE = 0)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
8	TS[63:0]							
9								
10								
11								
12								
13								
14								
15								

Figure 38.4 Timestamp Descriptor (GWDCCi.ETS = 1 and GWDCCi.EDE = 0)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
8	INFO1[63:0]							
9								
10								
11								
12								
13								
14								
15								

Figure 38.5 Extended Descriptor (GWDCCi.ETS = 0 and GWDCCi.EDE = 1)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
8	INFO1[63:0]							
9								
10								
11								
12								
13								
14								
15								
16	TS[63:0]							
17								
18								
19								
20								
21								
22								
23								

Figure 38.6 Extended Timestamp Descriptor (GWDCCi.ETS = 1 and GWDCCi.EDE = 1)

Table 38.3 General Format Descriptor Field Explanation

Field name	Bit width	Description
DS	12	Descriptor size. Restrictions: • SW: DS field maximum value is 2048. • SW: For other descriptors than LINKFIX, LEMPTY, EEMPTY, LINK and EOS, DS should not be set to 0.
INFO0	4	Information 0.
ERR	1	Error
DSE	1	Data Size Error.
AXIE	1	AXI Bus Error
DIE	1	Descriptor Interrupt Enable This (DESCR.DIE) field should not be set for LINK/LINKFIX descriptors in Rx chain.
DT	4	Descriptor Type, refer to (2) Descriptor Types.
PTR	40	Pointer
INFO1	64	Information 1.
TS	64	Timestamp.

Note: General format descriptor field utilization is described per use-case later in the following sections.

(2) Descriptor Types

The table below gives an overview of all descriptor types used by GWCA. The first column shows the descriptor type name and the second column the 4-bit number to be written to DESCR.DT.

Table 38.4 Descriptor Type

Name	DT	Description
Empty data descriptors		
FEMPTY_IS	1	**Frame Empty Incremental Start** Frame data related descriptor without valid frame data. Used to set an incremental area.
FEMPTY_IC	2	**Frame Empty Incremental Continue** Frame data related descriptor without valid frame data. Used to continue using an incremental area.
FEMPTY_ND	3	**Frame Empty No Data Storage** Frame data related descriptor without valid frame data. Used to reject data.
FEMPTY	4	**Frame Empty** Frame complete data related descriptor without valid frame data. Used to save frame data.
FEMPTY_START	5	**Frame Empty start** Frame start data related descriptor without valid frame data. Used to save first part of a frame.
FEMPTY_MID	6	**Frame Empty Middle** Frame continues data related descriptor without valid frame data. Used to save middle part of a frame.
FEMPTY_END	7	**Frame Empty End** Frame end data related descriptor without valid frame data. Used to save end part of a frame.
Data descriptors		
FSINGLE	8	**Frame Single** Storage Element has valid frame data of a complete frame.
FSTART	9	**Frame Start** Storage Element has valid frame data. Frame starts with this descriptor and continues in next descriptor.
FMID	10	**Frame Middle** Storage Element has valid frame data. Frame has started with a previous descriptor and continues in next descriptor.
FEND	11	**Frame End** Storage Element has valid frame data. Frame has started with a previous descriptor and ends with this descriptor.
Chain control/HW/SW arbitration		
LINKFIX	0	**Linkfix** Control element pointing to next descriptor in chain.
LEMPTY	12	**Link Empty** Used control element pointing to next descriptor in chain.
EEMPTY	13	**EOS Empty** Used control element pausing a transmit chain.
LINK	14	**Link** Control element pointing to next descriptor in chain.
EOS	15	**End Of Set** Control element pausing a transmit chain.

38.3.1.3 LINKFIX Table

(1) LINKFIX Table Initialization

The usage of AXI address table requires the setting of a LINKFIX table in the CPU USER RAM at {GWDCBAC0.DCBAU[7:0], GWDCBAC1.DCBAL[31:0]} location. This table will be used to set the first address of descriptor chains. It can be used any time to change a chain base address or to return to the previous base address. The LINKFIX table setting is described in Figure 38.7 through an example. Note that SW and HW initialization are happening simultaneously.

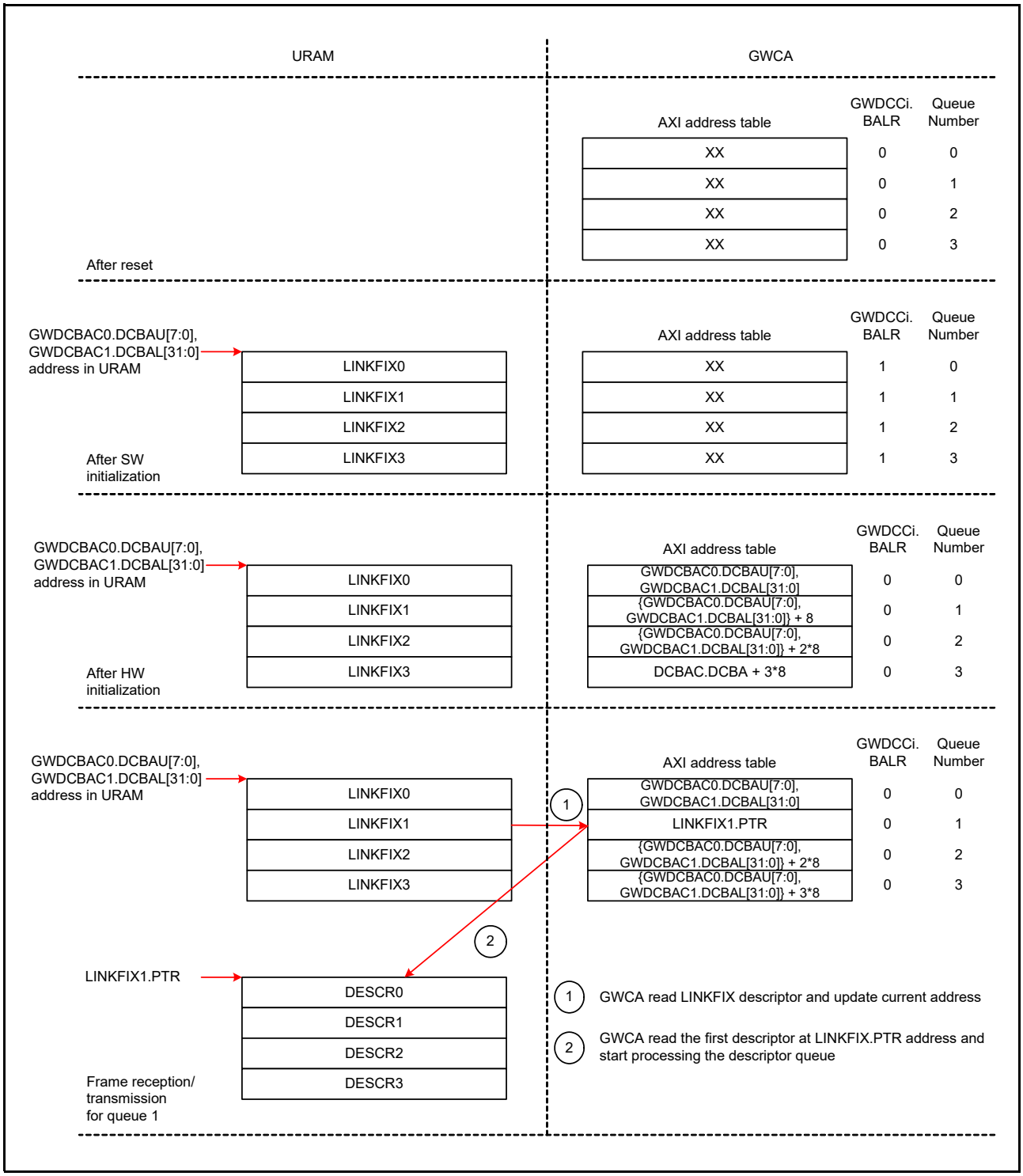


Figure 38.7 LINKFIX Table Setting Example

[Restrictions]

- SW: A LEMPTY descriptor queues must be written for preventing them from starting.
- HW: Add descriptors in LINKFIX table are 8 byte-descriptors independently from GWDCCi.EDE and GWDCCi.ETS bits settings.

Note: LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

(2) LINKFIX Table Descriptors

For the LINKFIX table, a LINKFIX descriptor is used to start a descriptor queue, a LEMPTY descriptor is used to disable a TX queues (GWDCCi.DQT = 1) and a LEMPTY descriptor is used to disable a RX queues (GWDCCi.DQT = 0). In the LINKFIX table, only basic descriptors can be used (refer to (1) General Formats). The descriptor formats for LINKFIX table are describes in Table 38.5 and Table 38.6. If a reception disabled queue is started despite being disabled in the LINKFIX table, the descriptor queue be considered as full (refer to register GWEIS2i explanation).

Table 38.5 LINKFIX Table Descriptor Formats

LINKFIX for LINKFIX Table								
Byte	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
0	00000000b							
1	0000b				0000b			
2	0				0b	0b	0b	0b
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
LEMPTY for LINKFIX Table								
Byte	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
0	00000000b							
1	0000b				0000b			
2	12				0b	0b	0b	0b
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

Table 38.6 LINKFIX Table Descriptor Field Description (1/2)

Field name	LINKFIX	LEMPTY
DS	000h	000h
INFO0	0000b	0000b
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	0	0
DT	0	12

Table 38.6 LINKFIX Table Descriptor Field Description (2/2)

Field name	LINKFIX	LEMPTY
PTR	PTR Pointer pointing to the first descriptor of the descriptor chain	0000000000h
INFO1 (GWDCCi.EDE = 1)	NA for LINKFIX table	NA for LINKFIX table
TS (GWDCCi.ETS = 1)	NA for LINKFIX table	NA for LINKFIX table

38.3.1.4 AXI Common Descriptor Utilization

AXI common descriptors are the descriptors which are not reserved to a special use-case and can be inserted in any descriptor chain at any time. It includes LINKFIX and LINK descriptors.

(1) LINKFIX Descriptor Utilization

A LINKFIX descriptor is used to change a descriptor queue pointer in the GWCA to move it to another place and is described in Figure 38.8. The LINKFIX descriptor is never written back to decrease the AXI load.
The LINKFIX descriptor format is described in Table 38.7 and Table 38.8.

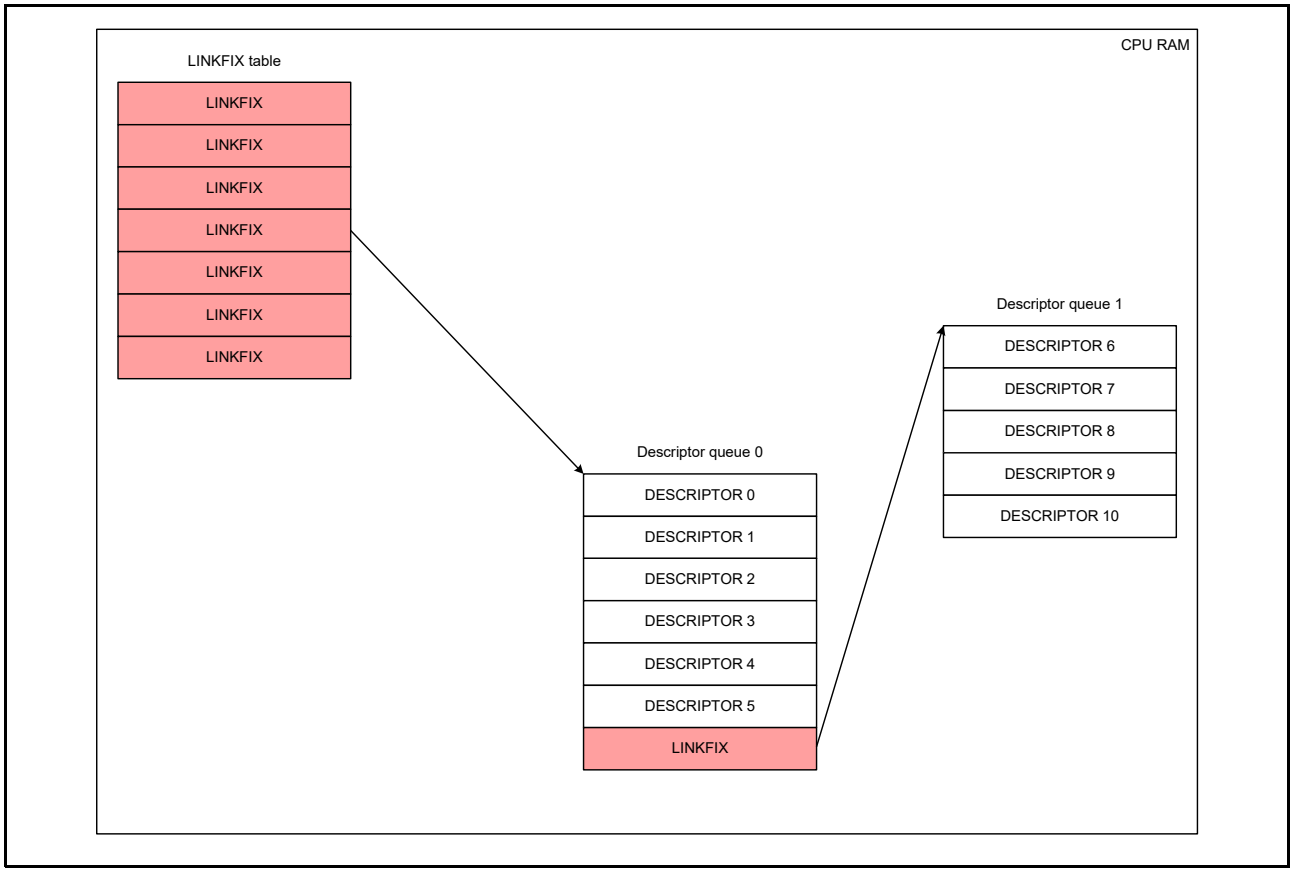


Figure 38.8 LINKFIX Descriptor Utilization

Note: The LINKFIX table is described in section 38.3.1.3, LINKFIX Table.
Note: A LINKFIX descriptor can be placed anywhere in a descriptor queue.
Note: LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

Table 38.7 LINKFIX Utilization Format (for Basic Descriptor, refer to (1) General Formats)

LINKFIX Written by SW								
Byte	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
0	00000000b							
1	0000b				0000b			
2	0				0b	0b	0b	0b
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

Table 38.8 LINKFIX Utilization Field Description

Field name	Field description written by SW	Field description after HW write-back
DS	000h	NA: LINKFIX descriptors are never written back
INFO0	0000b	NA: LINKFIX descriptors are never written back
ERR	0	NA: LINKFIX descriptors are never written back
DSE	0	NA: LINKFIX descriptors are never written back
AXIE	0	NA: LINKFIX descriptors are never written back
DIE	0	NA: LINKFIX descriptors are never written back
DT	0	NA: LINKFIX descriptors are never written back
PTR	PTR Pointer pointing to the new descriptor queue	NA: LINKFIX descriptors are never written back
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	NA: LINKFIX descriptors are never written back
TS (GWDCCi.ETS = 1)	0000000000000000h	NA: LINKFIX descriptors are never written back

(2) LINK Descriptor Utilization

A LINK descriptor has the same usage as a LINKFIX descriptor (refer to (1) LINKFIX Descriptor Utilization). Contrary to the LINKFIX descriptor the LINK descriptor can be written back by HW. The LINK descriptor format is described in Figure 38.9 and Table 38.9.

Note: LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

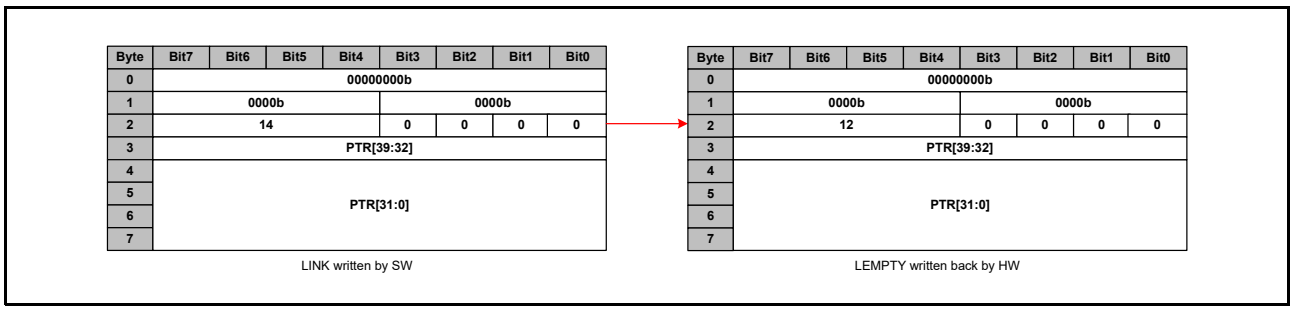


Figure 38.9 LINK Utilization Format (for Basic Descriptor, refer to (1) General Formats)

Table 38.9 LINK Utilization Field Description

Field name	Field description written by SW	Field description after HW write-back
DS	000h	000h (Value is copied from read descriptor)
INFO0	0000b	0000b (Value is copied from read descriptor)
ERR	0	0 (Value is copied from read descriptor)
DSE	0	0 (Value is copied from read descriptor)
AXIE	0	0 (Value is copied from read descriptor)
DIE	0	0 (Value is copied from read descriptor)
DT	14	12 (Value is copied from read descriptor)
PTR	PTR (user setting) Pointer pointing to the new descriptor queue	PTR (Value is copied from read descriptor)
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	NA: INFO1 not written back for LINK descriptors
TS (GWDCCi.ETS = 1)	0000000000000000h	NA: TS not written back for LINK descriptors

38.3.1.5 AXI Descriptor Queue Format

The AXI master accepts two kinds of descriptor queues: linear descriptor queues and cyclic descriptor queues. A linear descriptor queue is a queue which is ended by a terminate descriptor. A linear descriptor queue example is described in Figure 38.10. A cyclic descriptor queue is a queue which is looping on itself. A cyclic descriptor queue example is described in Figure 38.11.

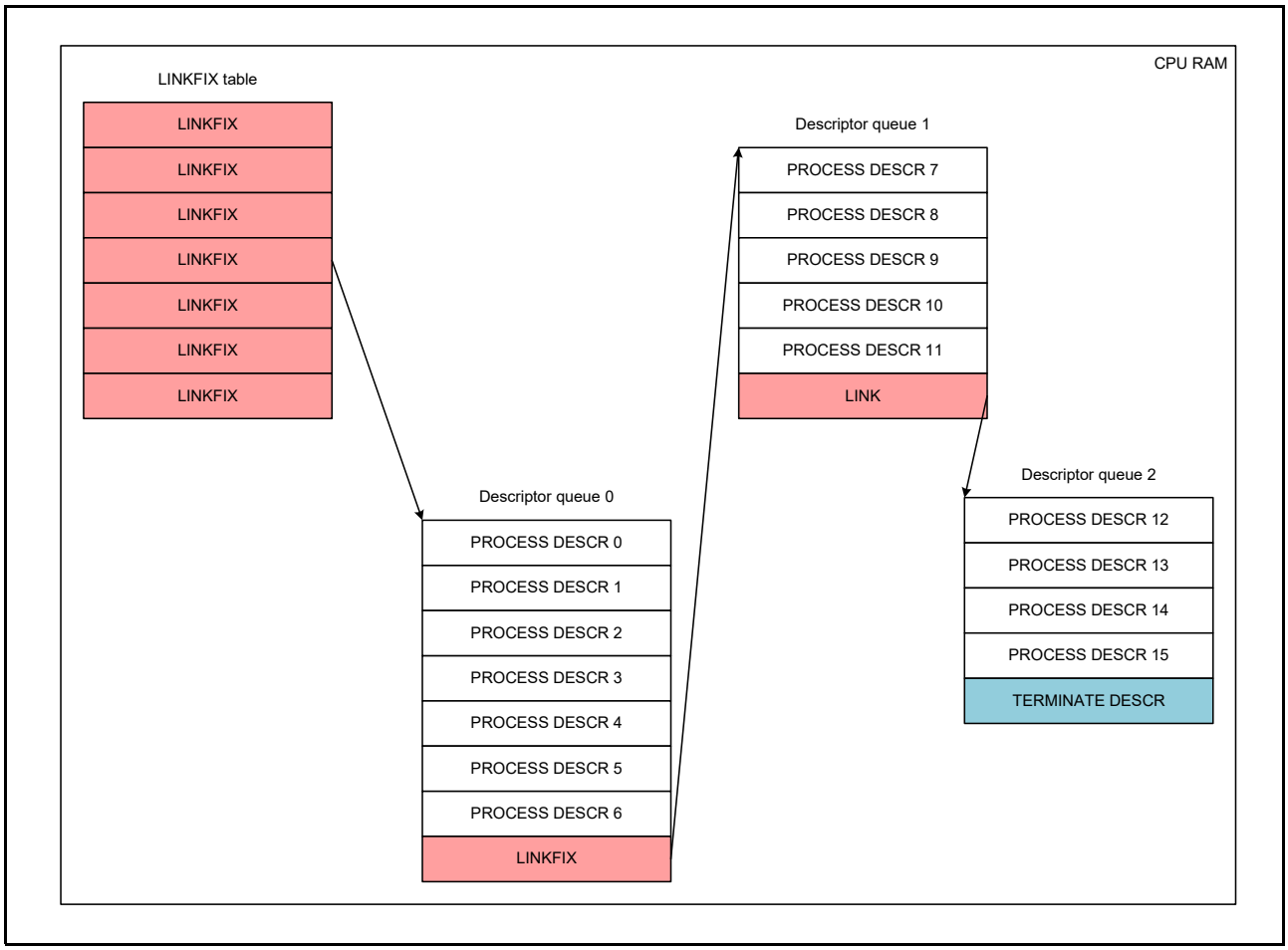


Figure 38.10 Linear Descriptor Queue Example

[Restriction]

- SW: To add data to a linear queue, the SW should write the data and process/terminate descriptor and overwrite the previous terminate descriptor with a LINK/LINKFIX descriptor pointing to the new descriptor chain.

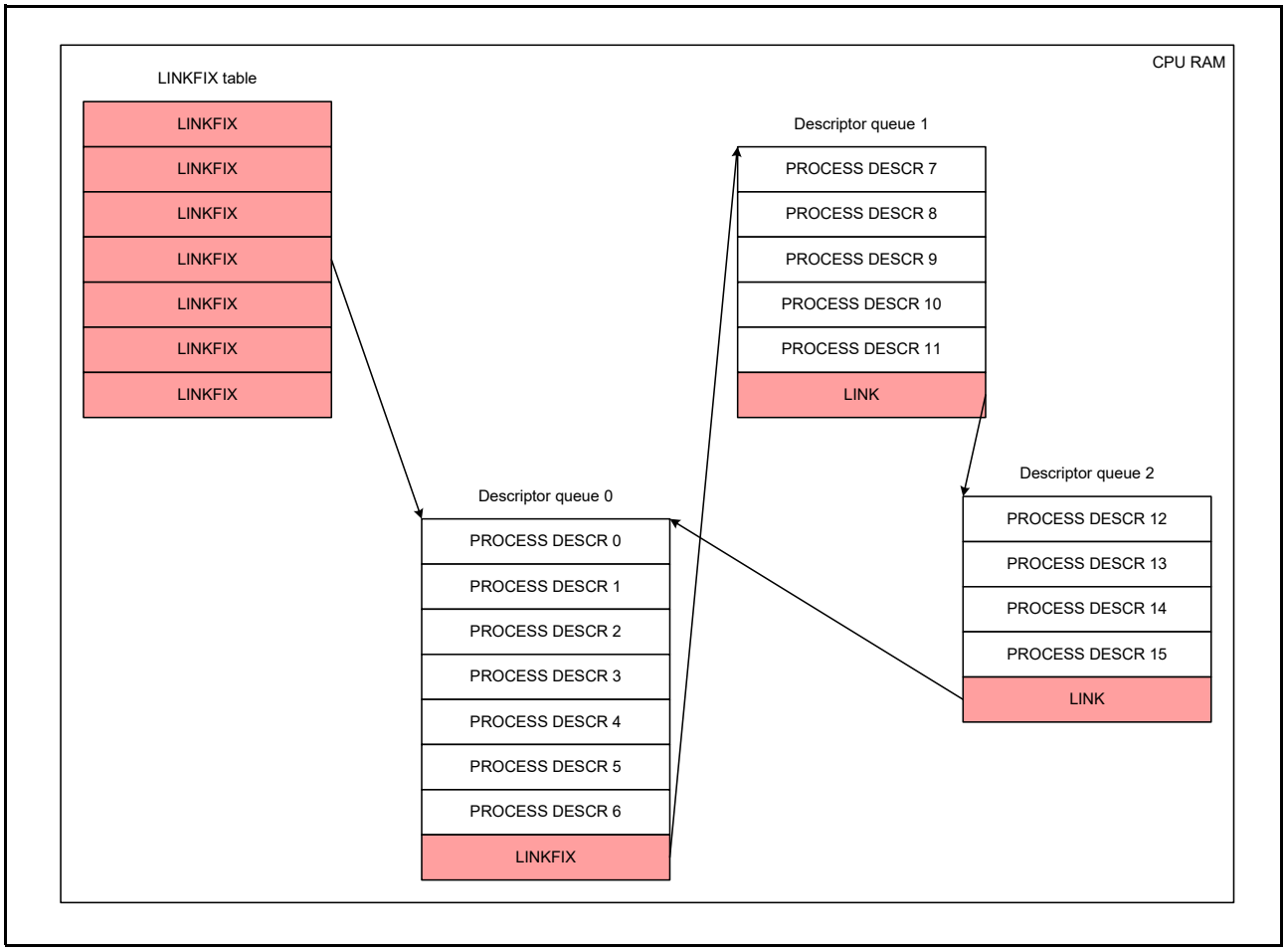


Figure 38.11 Cyclic Descriptor Queue Example

Note: The LINKFIX table is described in section 38.3.1.3, LINKFIX Table.

Note: The LINK, LINKFIX utilization is described in section 38.3.1.4, AXI Common Descriptor Utilization.

Note: A terminate descriptor (TERMINATE DESCR in Figure 38.10) correspond to a FSINGLE or a LEMPTY descriptor for an RX queue, a FEMPTY or a LEMPTY descriptor for a TX queue and a FSINGLE or a LEMPTY descriptor for a TS queue.

Note: A process descriptor (PROCESS DESCR in Figure 38.10 and Figure 38.11) correspond to a FEMPTY_IS, FEMPTY_IC, FEMPTY_ND, FEMPTY, FEMPTY_START, FEMPTY_MID or FEMPTY_END descriptor for an RX queue (refer to section 38.3.3.4, AXI Master Interface), to a FSINGLE, FSTART, FMID, FEND or EOS descriptor for a TX queue (refer to section 38.3.2.2, AXI Master Interface) and to a FEMPTY_ND descriptor for a TS queue (refer to section 38.3.4.2, AXI Master Interface).

38.3.1.6 AXI Address RAM Searching

Searching functions is used to read entries in the AXI address RAM. Table 38.10 describes register used to read an entry in the TAS RAM. Table 38.11 describes the read results.

Table 38.10 AXI Address Read Registers

Register name	Field name/corresponding field in MAC table	Field explanation
GWAARSS.AARA	Entry address Set the descriptor queue number from which AXU address should be read	Not present in AXI address RAM, entry will be read from this address

Table 38.11 AXI Address Read Result

Register name	Field name/corresponding field in MAC table	Field explanation
{GWAARSR0.ACARU[7:0], GWAARSR1.ACARL[31:0]}	AXI address value	Address of the next descriptor to be processed for AXI descriptor queue number GWAARSS.AARA[5:0]

[Caution]

- {GWAARSR0.ACARU[7:0], GWAARSR1.ACARL[31:0]} indicates the address of the next descriptor to be processed for AXI descriptor queue number but it does not mean that the previous descriptor process is completed. That's why this register only has debugging purpose and not HW/SW synchronization purposes.

38.3.2 Data Transmission

GWCA allows data transmission through the GWCA TX data path, The TX data path is described in Figure 38.12.

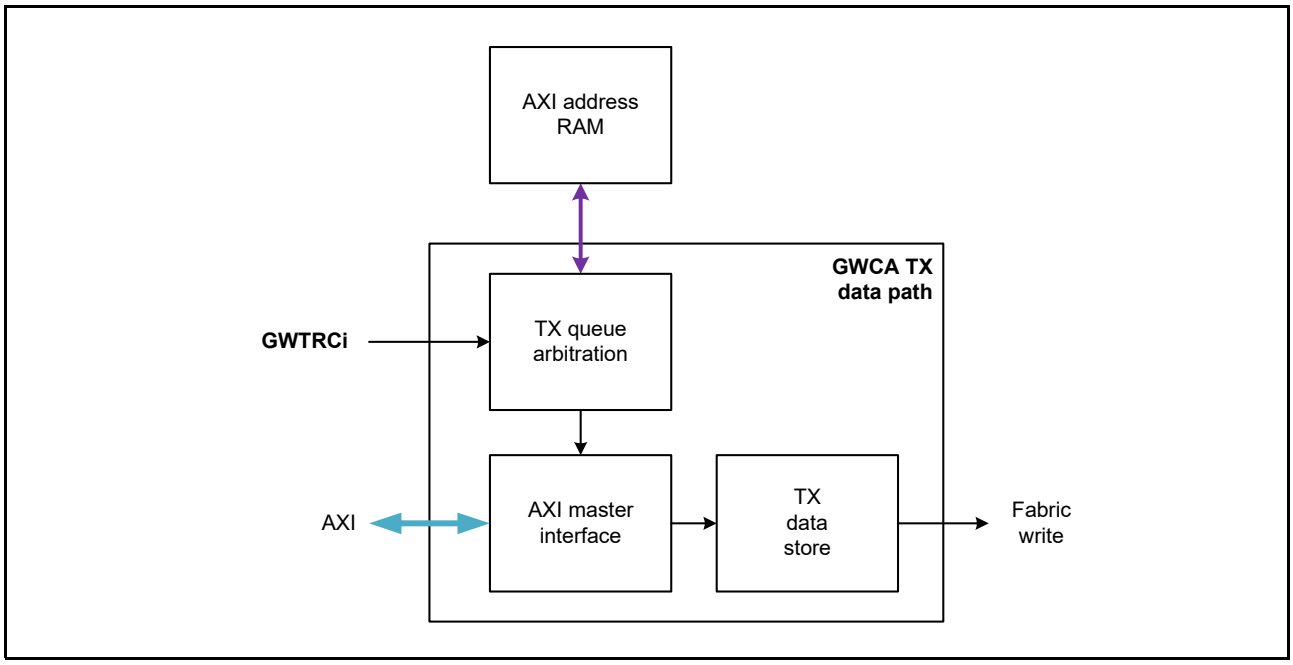


Figure 38.12 GWCA TX Data Path Block Diagram

The TX data path is separated in three blocks:

- TX queue arbitration: This block arbitrates between TX queues when several queues are requesting for transmission.
- AXI master interface: This block handles the data/AXI descriptor exchange with the CPU.
- TX data store: This block extracts TAG information for frames, applies VLAN tagging and save the frames in the local RAM. This block can also release pointers while going out of OPERATION mode. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.

38.3.2.1 TX Queue Arbitration

TX queue arbitration has four functions. Pausing certain queues when switch is about to overflow, limiting TX queue data rates, arbitrating between TX descriptor queue with a strict priority algorithm and arbitrating between same priority queues with a round robin algorithm.

(1) Per Priority Pause

Per priority pause aims at pausing descriptor queue transmission depending on their priority using PAUSE[2 for GWCA0] values [COMA] and GWTPCp register. It avoids the AXI master to inject uncritical data in the switch when it is about to overflow.

[Function]

- GWTPCp sets the priorities that should be paused when PAUSE[2 for GWCA0][p] is set.

(2) Rate Limiters

Rate limiters are controlling the input flow of the switch. Using these limiters could prevent the switch from overflowing. GWCA has two types of rate limiters, a global rate limiter and 8 per-queue rate limiters. By setting the rate limiter configurations correctly, expected bandwidth with a tolerance value of less than ± 2% can be achieved.

(a) Per-Queue Rate-Limiter

A per-queue rate-limiter aims at limiter the data throughput per one descriptor queue using GWRLCi.RLE, GWRLCi.RLIV[11:0] and GWRLULCi.RLUL[23:0] bits. For per-port rate limiter/descriptor queue mapping information refer to section 38.3.1.1, AXI Descriptor Queue Mapping.

[Function]

- GWRLCi.RLE bit enables rate-limiter number i.
- GWRLCi.RLIV[11:0] bits sets the maximum throughput accepted by rate limiter number i and should be set respecting equation (1).
- GWRLULCi.RLUL[23:0] bits sets the maximum collision time accepted by rate limiter number i and should be set respecting equation (2).

(1) GWRLCi.RLIV[11:0] = 256 × ACLK_period[ns] × maxBandwidth[Gbps]
(2) GWRLULCi.RLUL[23:0] = maximumBurst[bit]

[Where]

- ACLK_period is ACLK clock period.
- maxBandwidth is the expected maximum throughput accepted by rate limiter number i.
- maximumBurst is the maximum burst in bit accepted for corresponding queue.

(b) Global Rate Limiter

The global rate limiter aims at limiting all AXI data transmission throughput using GWGRLC.GRLE, GWGRLC.GRLIV[15:0] and GWGRLULC.GRLUL[23:0] bits and can be monitored using GWGRLC.GRLULRS.

[Functions]

- GWGRLC.GRLE bit enables the global rate limiter.
- GWGRLC.GRLIV[15:0] bits set the maximum throughput accepted by AXI data transmission and should be set respecting equation (1).
- GWGRLULC.GRLUL[23:0] bits set the maximum number of credits that the global rate limiter can stack and should be set respecting equation (2).
- GWGRLC.GRLULRS flag shows if some credit has been lost because of a too high AXI latency or because of a too slow AXI.

(1) GWGRLC.GRLIV[15:0] = 256 × ACLK_period[ns] × max_bandwidth[Gbps]
(2) GWGRLULC.GRLUL[23:0] = ACLK_period[ns] × max_bandwidth[Gbps] × 2*maximum_axi_latency[cycle]

[Where]

- ACLK_period is ACLK clock period in ns.
- max_bandwidth is the expected maximum throughput accepted by GWCA.
- maximum_axi_latency is the AXI maximum latency in ACLK clock number.

[Restriction]

- SW: The sum of all GWRLCi.RLIV[11:0] bits should always be smaller or equal to GWGRLC.GRLIV[15:0].

(3) Strict Priority

Strict priority arbitrates between TX queue which are authorized by rate limiters to transmit using GWDCCi.DCP[2:0]. The strict priority algorithm operates as the one in the data reception descriptor store. Refer to (a) Strict priority (SP) in (2) Descriptor Arbitration for more information.

[Functions]

- GWDCCi.DCP[2:0] bits set descriptor queue i priority. The highest priority is the biggest value.

(4) Round Robin

Round robin arbitrates between the highest priority queues selected by the strict arbiter. The round robin operates as the one in the data reception descriptor store. Refer to (b) Round Robin (RR) in (2) Descriptor Arbitration for more information.

38.3.2.2 AXI Master Interface

The AXI master interface handles the data/AXI descriptor exchange with the CPU. It receives the queue number of the descriptor queue number authorised to transmit by the TX queue arbitration, read the corresponding data from the CPU RAM and send the data to TS data store. To do so, it uses GWTRCi.TSRt, GWDCCi.EDE, GWDCCi.ETS, GWDCCi.SM[1:0] bits and can be monitored using GWAARSS, GWAARSR0, and GWAARSR1 registers.

[Functions]

- GWTRCi.TSRt bit is used to start transmission when a descriptor queue is ready.
- GWDCCi.EDE and GWDCCi.ETS bits set the descriptor format that will be used for descriptor queue number i refer to (1) General Formats.
- GWDCCi.SM[1:0] bits sets how GWCA will write back descriptors. In this section all explanation will use GWDCCi.SM[1:0] equal to 00b. For GWDCCi.SM[1:0] equal to 01b or 10b refer to the register explanation.
- For GWAARSS, GWAARSR0 and GWAARSR1 registers refer to section 38.3.1.6, AXI Address RAM Searching.

[Restriction]

- SW: Only basic descriptors and extended descriptors are authorized for data transmission, refer to (1) General Formats.

Data transmission can be done using linear or cyclic descriptor queues (refer to section 38.3.1.5, AXI Descriptor Queue Format). Data transmission process descriptors can be set two different ways, through the basic data transmission descriptor chain (refer to (1) Basic Data Transmission) and the time-controlled data transmission descriptor chain (refer to (2) Time-Controlled Data Transmission).

(1) Basic Data Transmission

Basic data transmission process descriptor utilization is described in Figure 38.13. Basic data transmission process descriptor format is described in Figure 38.14 and Table 38.12.

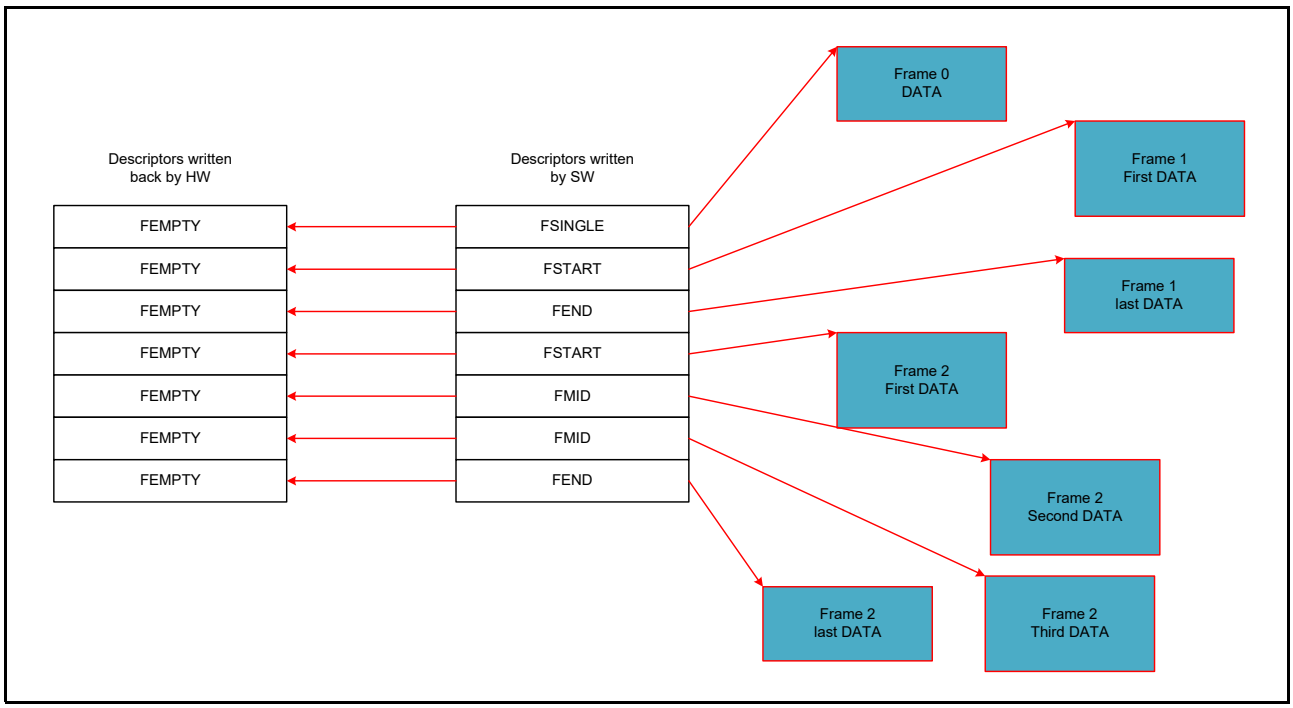


Figure 38.13 Basic Data Transmission Process Descriptor Utilization

Note: If needed, the SW can split a frame between several descriptors. The first descriptor should be a FSTART descriptor, the last descriptor should be a FEND descriptor and all other descriptors should be FMID descriptors.

Note: When data is ready, CPU should set the corresponding GWTRCi.TSRt bit to start transfer. Refer to GWTRCi.TSRt description.

Note: GWTRCi.TSRt can be set anytime even if data is not available. The GWCA will in this case read the terminate descriptor and clear GWTRCi.TSRt bit.

[Restriction]

- SW: When a frame is split, the FSTART descriptor should be written after FMIDs and FEND descriptor for a given frame.

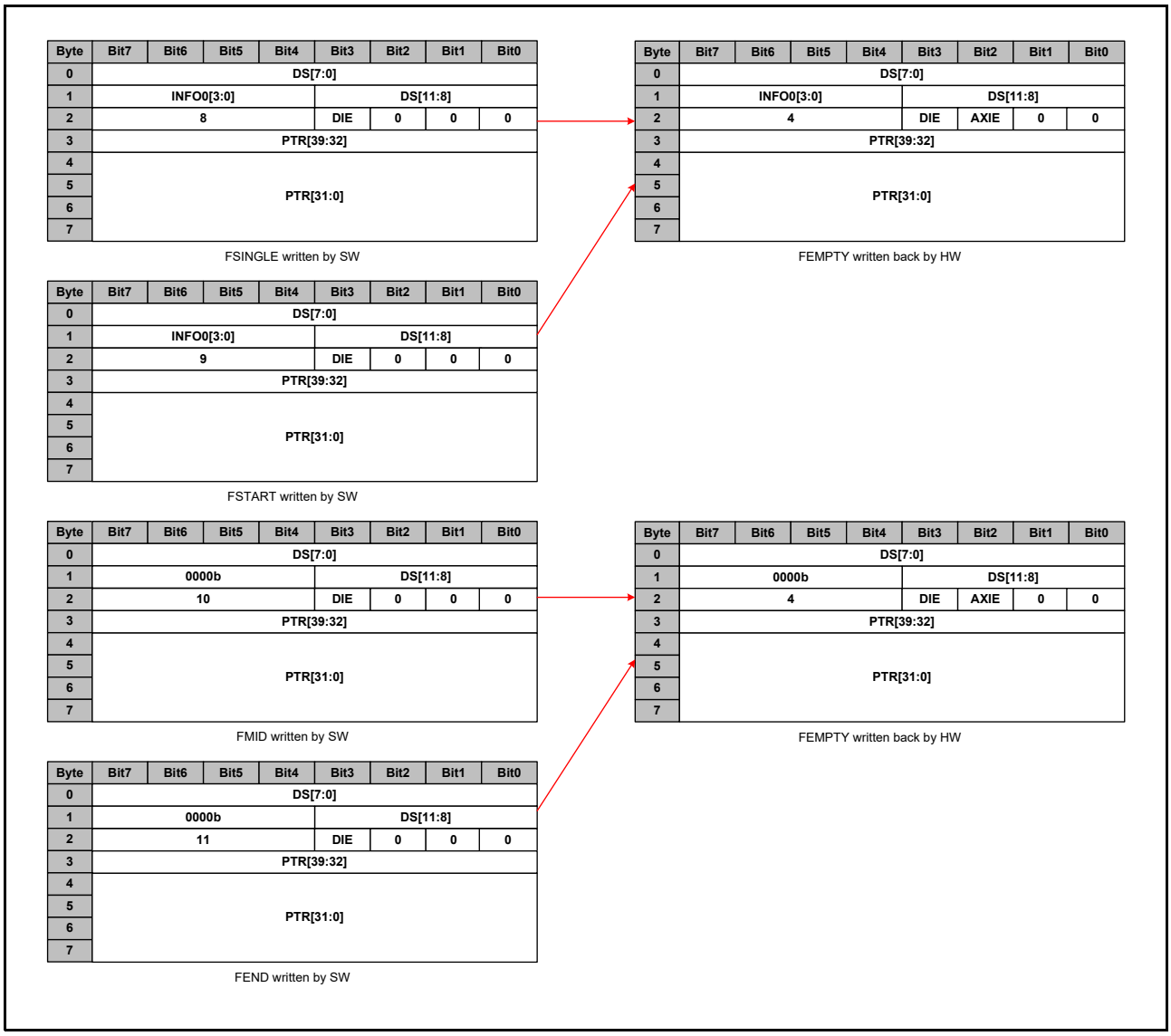


Figure 38.14 Basic Data Transmission Process Descriptor Utilization Format (for Basic Descriptor, Refer to (1) General Formats)

Table 38.12 Basic Data transmission Process Descriptor Utilization Field Description

Field name	Field description written by SW	Field description after HW write-back
DS	Data what has been written to the URAM by SW	DS (Value is copied from read descriptor)
INFO0	For FSINGLE and FSTART refer to (3) Data Transmission Common Descriptor Format 0000b for FMID and FEND.	INFO0 (Value is copied from read descriptor)
ERR	0	0
DSE	0	0
AXIE	0	[Setting conditions] HW: An AXI error happened while reading a data (RRESP ≠ 00b).
DIE	DIE	DIE (Value is copied from read descriptor)
DT	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND	4 for FEMPTY
PTR	Address where data is written in the URAM	PTR (Value is copied from read descriptor)
INFO1 (GWDCCi.EDE = 1)	For FSINGLE and FSTART refer to (3) Data Transmission Common Descriptor Format 0000b for FMID and FEND.	NA: INFO1 not written back for transmission
TS (GWDCCi.ETS = 1)	NA: No timestamp descriptor for transmission	NA: No timestamp descriptor for transmission

(2) Time-Controlled Data Transmission

Time-controlled data transmission process descriptor utilization is described in Figure 38.15. Time-controlled data transmission process descriptor format is described in Figure 38.16 and Table 38.13. FSINGLE, FSTART, FMID, FEND and FEMPTY formats are not described in this section, for their description refer to Figure 38.14 and Table 38.12.

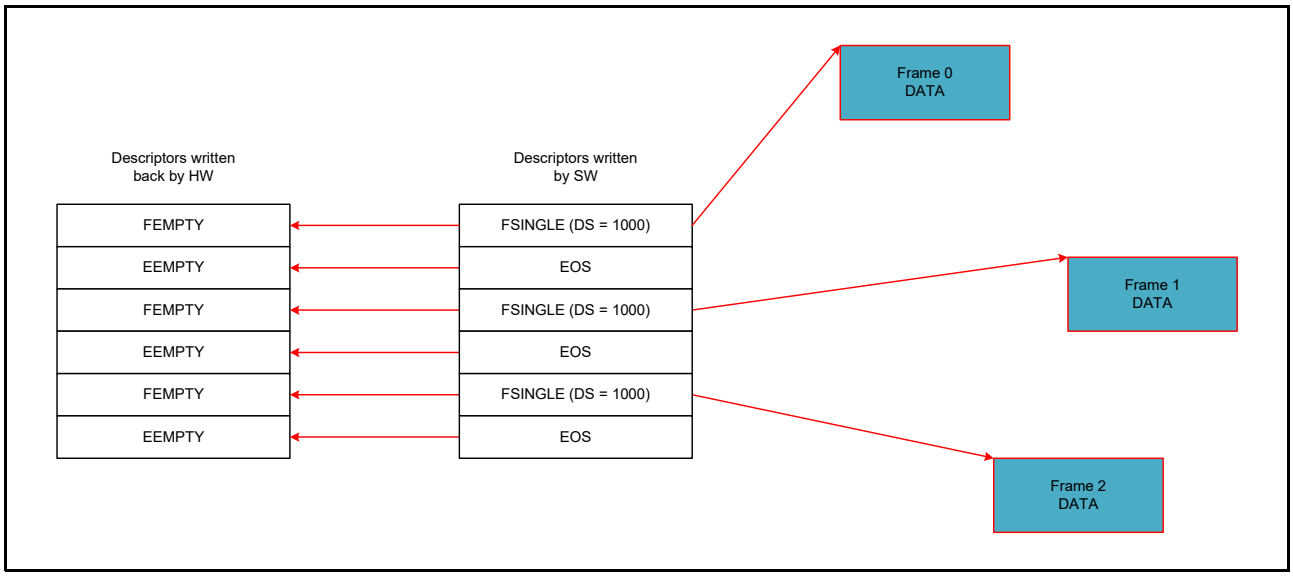


Figure 38.15 Time-Controlled Data Transmission Process Descriptor Utilization

Note: If needed, the SW can split a frame between several descriptors. The first descriptor should be a FSTART descriptor, the last descriptor should be a FEND descriptor and all other descriptors should be FMID descriptors.

Note: When data is ready, CPU should set the corresponding GWTRCi.TSRt bit to start transfer. Refer to GWTRCi.TSRt description.

Note: Reading an EOS descriptor will have per effect to clear GWTRCi.TSRt bit. So, if the CPU set GWTRCi.TSRt bit at a cyclic timing, CPU can control the data throughput (If TSR is already set when the CPU tries to set it the next time, only one frame will be sent so a decrease in throughput can happen).

Note: GWTRCi.TSRt bit can be set automatically using gPTP timer [GPTP]. Refer to section 38.3.5, gPTP Triggered Transmission.

Note: GWTRCi.TSRt can be set anytime even if data is not available. The GWCA will in this case read the terminate descriptor and clear GWTRCi.TSRt bit.

[Restriction]

- SW: When a frame is split, the FSTART descriptor should be written after FMIDs and FEND descriptor for a given frame.

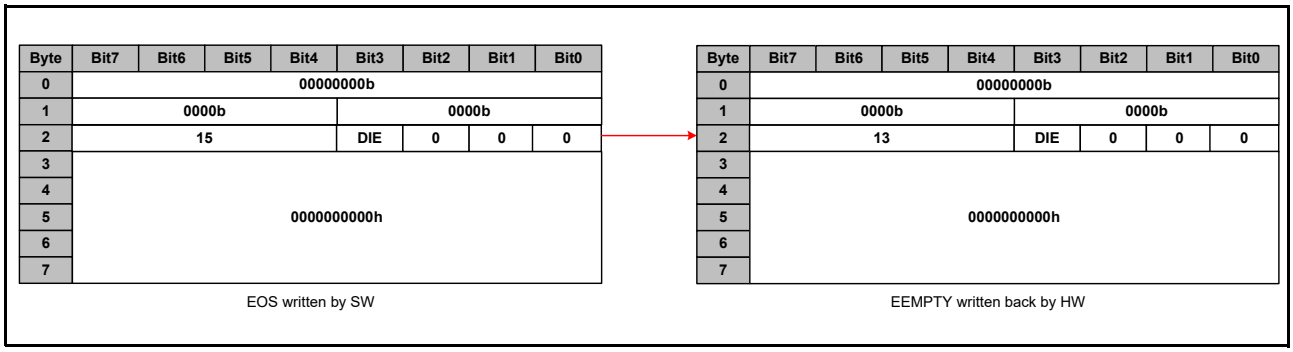


Figure 38.16 Time-Controlled Data Transmission Process Descriptor Utilization Format (for Basic Descriptor, Refer to (1) General Formats)

Table 38.13 Time-Controlled Data Transmission Process Descriptor Utilization Field Description

Field name	Field description written by SW	Field description after HW write-back
DS	000h	DS (Value is copied from read descriptor)
INFO0	0000b	INFO0 (Value is copied from read descriptor)
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	DIE	DIE (Value is copied from read descriptor)
DT	15	13
PTR	0000000000h	PTR (Value is copied from read descriptor)
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	NA: INFO1 not written back for EOS descriptors
TS (GWDCCi.ETS = 1)	NA: No timestamp descriptor for transmission	NA: No timestamp descriptor for transmission

(3) Data Transmission Common Descriptor Format

In this section are described the fields which are common for any data transmission (INFO0 and INFO1). There are two types of formats, the transmission direct descriptor and the transmission ethernet descriptor formats. the transmission direct descriptor could be used, for example, for non-routable protocol forwarding (network control) or for CPU-CPU communication and the transmission ethernet descriptor could be used for all normal ethernet transmissions. All the fields contained in AXI descriptors, if quoted, will be written ADESCR.{Field name}.

(a) Transmission Direct Descriptor

Transmission direct descriptor is used by a CPU to send frames by deciding their target directly (Forwarding engine processing will be by-passed [MFWD]). Transmission direct descriptor INFO formats are described in Figure 38.17 and Table 38.14.

[Restriction]

- SW: The Transmission direct descriptor can only be and extended descriptor, refer to (1) General Formats.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	RSV		SEC	FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	RN[7:0]							
3	FW	IPV[2:0]			RV	RSV		
4	RSV	CSD0[6:0]						
5	RSV							
6	RSV	DV[6:0]						
7	RSV							

Figure 38.17 Transmission Direct Descriptor INFO Formats

Table 38.14 Transmission Direct Descriptor INFO Field Descriptions

Field name	Bit width	Description
FI	1	FCS in.
SEC	1	Secure descriptor
FMT	1	Descriptor format Restrictions: • SW: Fixed to 1 for Direct descriptor.
TXC	1	TX Timestamp capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
IET	1	Timestamp insertion request [RMAC]. Used to set corresponding bit in RMAC TX interface.
CRT	1	Residence time calculation request [RMAC]. Used to set corresponding bit in RMAC TX interface.
TN	1	Timer utilized for capture/insertion [RMAC]. Used to set corresponding bit in RMAC TX interface.
TSUN	8	Timestamp unique number [RMAC]. Used to set corresponding bit in RMAC TX interface.
RV	1	
RN	8	
IPV	3	Internal priority value Target priority of the frame.
FW	1	The FCS contained in the frame is wrong (It will be corrected in the switch)
CSD0	6	CPU sub destination for GWCA0
DV	4	Destination vector [MFWD]
RSV	—	Reserved area Restrictions: • SW: these fields should be set to 0.

(b) Transmission Ethernet Descriptor

Transmission ethernet descriptor is used by a CPU to send frames that will be forwarded by the Forwarding engine [MFWD]. Transmission ethernet descriptor INFO formats are described in Figure 38.18 and Table 38.15.

[Restriction]

- SW: Because FW field (refer to (a) Transmission Direct Descriptor) is not present in an Ethernet descriptor. Any transmit frame should have either a correct FCS or no FCS.

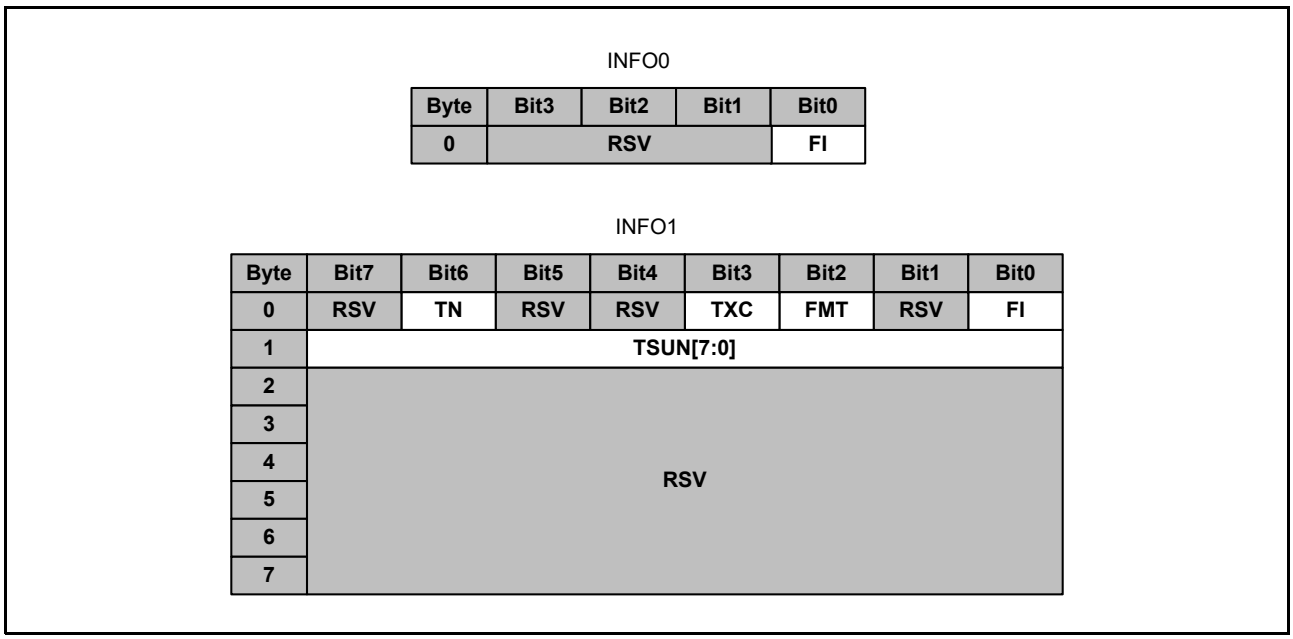


Figure 38.18 Transmission Ethernet Descriptor INFO Formats

Table 38.15 Transmission Ethernet Descriptor INFO Field Descriptions

Field name	Bit width	Description
FI	1	FCS in.
FMT	1	Descriptor format Restrictions: • SW: Fixed to 0 for Ethernet descriptors.
TXC	1	TX Timestamp capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
TN	1	Timer utilized for capture/insertion [RMAC]. Used to set corresponding bit in RMAC TX interface.
TSUN	8	Timestamp unique number [RMAC]. Used to set corresponding bit in RMAC TX interface.
RSV	—	Reserved Restrictions: • SW: these fields should be set to 0.

38.3.2.3 TX Data Store

TX data store has follow functions, it extracts TAG information for frames, applies VLAN tagging, and save the frames in the local RAM. These functions are applied to all frames independently from their descriptor format.

(1) TAG Information Extraction

TAG information extraction is done in HW without setting intervention. The extracted format can be filtered using GWTTFC register. TAG information extraction is described in Figure 38.19 along with GWTTFC register bit explanation for TAG filtering. The extracted fields are SN, CoS-TCI, C-TCI and S-TCI.

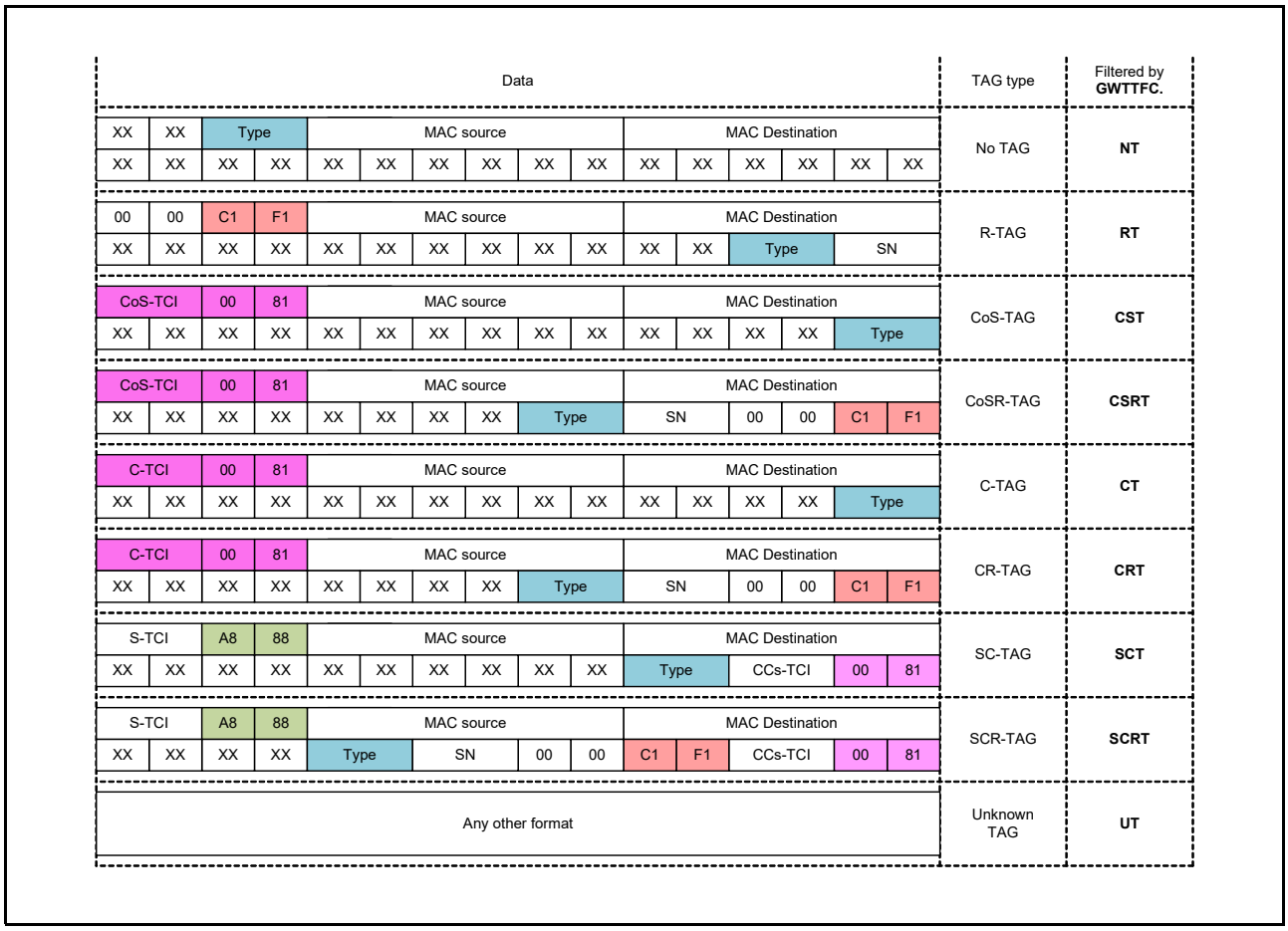


Figure 38.19 TAG Information Extraction

Note: Only colored part in Figure 38.19 are considered for TAG information extraction.
Note: HF1C1 corresponds to R-TAG TPID and SN correspond to R-TAG sequence number [802.1CB].
Note: H8100 corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].
Note: H88A8 corresponds to S-TAG TPID and S-TCI corresponds to a S-TAG TCI [802.1Q].
Note: Type corresponds to any Ethernet type value which is not HF1C1, H88A8 or H8100.
Note: TPID values can be configured in Forwarding Engine [MFWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

[Restriction]

- HW: Because 32 bytes are needed for TAG extraction, GWCA will reject transmit frames smaller than 32 bytes and set GWEIS0.USMFSES bit.
- SW: In R-TAG, the reserved part should always be set to 0. If the reserved part is not set to 0 and corresponding

frame contains an FCS, the frame could be forwarded with an error.

(2) VLAN Tagging

VLAN tagging replaces or overwrites present TAGs in a frame using the format given by TAG information extraction (refer to (1) TAG Information Extraction), switch VLAN mode (FWGC.SVM[1:0] bits in forwarding engine [MFWD]) and, GWVCC and GWVTC registers. After VLAN tagging, any frame has three TAGs, one R-TAG, one C-TAG and one S-TAG which will be directly saved in the TAG RAM [MFWD].

Functions:

- GWVCC.VIM bit sets the TX data path in Port based VLAN (all incoming frames are sent to the forwarding engine with the same VLAN ID). GWVCC.VIM setting is ignored when the switch in is No VLAN mode (FWGC.SVM bit in forwarding engine set to 00b) [MFWD].
- GWVTC register is used to set the HW TAGs. There are three possible HW TAGs. The HW C-TAG made from GWVTC.CTV[11:0] as VID, GWVTC.CTP[2:0] as PCP and GWVTC.CTD as DEI. The HW S-TAG made from GWVTC.STV[11:0] as VID, GWVTC.STP[2:0] as PCP and GWVTC.STD as DEI. The HW CoS-TAG made from GWVTC.CTV[11:0] as VID, the incoming frame PCP as PCP and the incoming frame DEI as DEI.

TAGs after VLAN tagging are described in Figure 38.20, Figure 38.21 and Figure 38.22 depending on Switch VLAN mode (FWGC.SVM bit in forwarding engine [MFWD]) and the type of frame given by TAG information extraction (refer to (1) TAG Information Extraction). In the figures, the data extracted from the incoming frames are in blue and the added HW TAGs are in red.

TAG type	R-TAG	C-TAG	S-TAG
No TAG	0	0	0
R-TAG	SN	0	0
CoS-TAG	0	CoS-TCI	0
CoSR-TAG	SN	CoS-TCI	0
C-TAG	0	C-TCI	0
CR-TAG	SN	C-TCI	0
SC-TAG	0	C-TCI	S-TCI
SCR-TAG	SN	C-TCI	S-TCI
Unknown TAG	0	0	0

Figure 38.20 TAGs after VLAN Tagging: Switch in No-VLAN Mode [MFWD]

	Incoming VLAN mode (**GWVCC.VIM** = 0)			Port based VLAN mode (**GWVCC.VIM** = 1)		
TAG type	R-TAG	C-TAG	S-TAG	R-TAG	C-TAG	S-TAG
No TAG	0	HW C-TAG	0	0	HW C-TAG	0
R-TAG	SN	HW C-TAG	0	SN	HW C-TAG	0
CoS-TAG	0	HW CoS-TAG	0	0	HW CoS-TAG	0
CoSR-TAG	SN	HW CoS-TAG	0	SN	HW CoS-TAG	0
C-TAG	0	C-TCI	0	0	HW C-TAG	0
CR-TAG	SN	C-TCI	0	SN	HW C-TAG	0
SC-TAG	0	C-TCI	S-TCI	0	HW C-TAG	0
SCR-TAG	SN	C-TCI	S-TCI	SN	HW C-TAG	0
Unknown TAG	0	HW C-TAG	0	0	HW C-TAG	0

Figure 38.21 TAGs after VLAN Tagging: Switch in C-TAG Mode [MFWD]

	Incoming VLAN mode (**GWVCC.VIM** = 0)			Port based VLAN mode (**GWVCC.VIM** = 1)		
TAG type	R-TAG	C-TAG	S-TAG	R-TAG	C-TAG	S-TAG
No TAG	0	HW C-TAG	HW S-TAG	0	HW C-TAG	HW S-TAG
R-TAG	SN	HW C-TAG	HW S-TAG	SN	HW C-TAG	HW S-TAG
CoS-TAG	0	HW CoS-TAG	HW S-TAG	0	HW CoS-TAG	HW S-TAG
CoSR-TAG	SN	HW CoS-TAG	HW S-TAG	SN	HW CoS-TAG	HW S-TAG
C-TAG	0	C-TCI	HW S-TAG	0	C-TCI	HW S-TAG
CR-TAG	SN	C-TCI	HW S-TAG	SN	C-TCI	HW S-TAG
SC-TAG	0	C-TCI	S-TCI	0	C-TCI	HW S-TAG
SCR-TAG	SN	C-TCI	S-TCI	SN	C-TCI	HW S-TAG
Unknown TAG	0	HW C-TAG	HW S-TAG	0	HW C-TAG	HW S-TAG

Figure 38.22 TAGs after VLAN Tagging: Switch in SC-TAG Mode [MFWD]

(3) Frame Saving in Local RAM

For frame saving in the local RAM, refer to Fabric specification document [MFWD], section “Data representation in RAMs”. Only the frame descriptor mentioned in Fabric specification document [MFWD] will be described in this section.

There are two types of Local RAM descriptors, direct local descriptors and the ethernet local descriptors. They are respectively created from direct descriptors and ethernet descriptors, refer to (3) Data Transmission Common Descriptor Format.

All the fields in these descriptors, if quoted, will be written LDESCR.{Field name}.

(a) Direct Local Descriptor

Direct local descriptor format is described in Figure 38.23 and Table 38.16.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	RSV				RTGI	VCTRL[2:0]		
4	RN[7:0]							
5	FW	IPV[2:0]			RV	RSV		
6	TPL[15:0]							
7								
8	RSV	CSD0[6:0]						
9	RSV	CSD1[6:0]						
10	RSV	DV[6:0]						
11	RSV							
12								
13								
14								
15								

Figure 38.23 Direct Local Descriptor Format

Table 38.16 Direct Local Descriptor Field Description (1/2)

Field name	Bit width	Description	Value for GWCA
FI	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
SEC	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
FMT	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
TXC	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
IET	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
CRT	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
TN	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.

Table 38.16 Direct Local Descriptor Field Description (2/2)

Field name	Bit width	Description	Value for GWCA
TSUN	8	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
SAEF	8	Source agent error flags	Refer to (c) SAEF Format.
VCTRL	3	VLAN control	VCTRL[2] values: Set to GWVCC.VIM. VCTRL[1:0] values 00b: For No TAG, R-TAG and Unknown TAG frames, refer to (1) TAG Information Extraction. 01b: For C-TAG and CR-TAG frames, refer to (1) TAG Information Extraction. 10b: For SC-TAG and SCR-TAG frames, refer to (1) TAG Information Extraction. 11b: For CoS-TAG and CoSR-TAG frames, refer to (1) TAG Information Extraction.
RTGI	1	R-TAG in	0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames, refer to (1) TAG Information Extraction. 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames, refer to (1) TAG Information Extraction.
RV	1	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
RN	8	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
IPV	3	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
FW	1	Refer to Table 38.14.	Copied from transmission direct descriptor except when set by checksum handling function. Refer to (a) Transmission Direct Descriptor for transmission direct descriptor and refer to 8.2.3.4.
TPL	16	Total payload length	Payload length of frame. It only includes the data saved in the data RAM [MFWD].
CSD0	6	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
DV	3	Refer to Table 38.14.	Copied from transmission direct descriptor. Refer to (a) Transmission Direct Descriptor.
RSV	—	Reserved area	Set to 0

(b) Ethernet Local Descriptor

Ethernet local descriptor format is described in Figure 38.24 and Table 38.17.

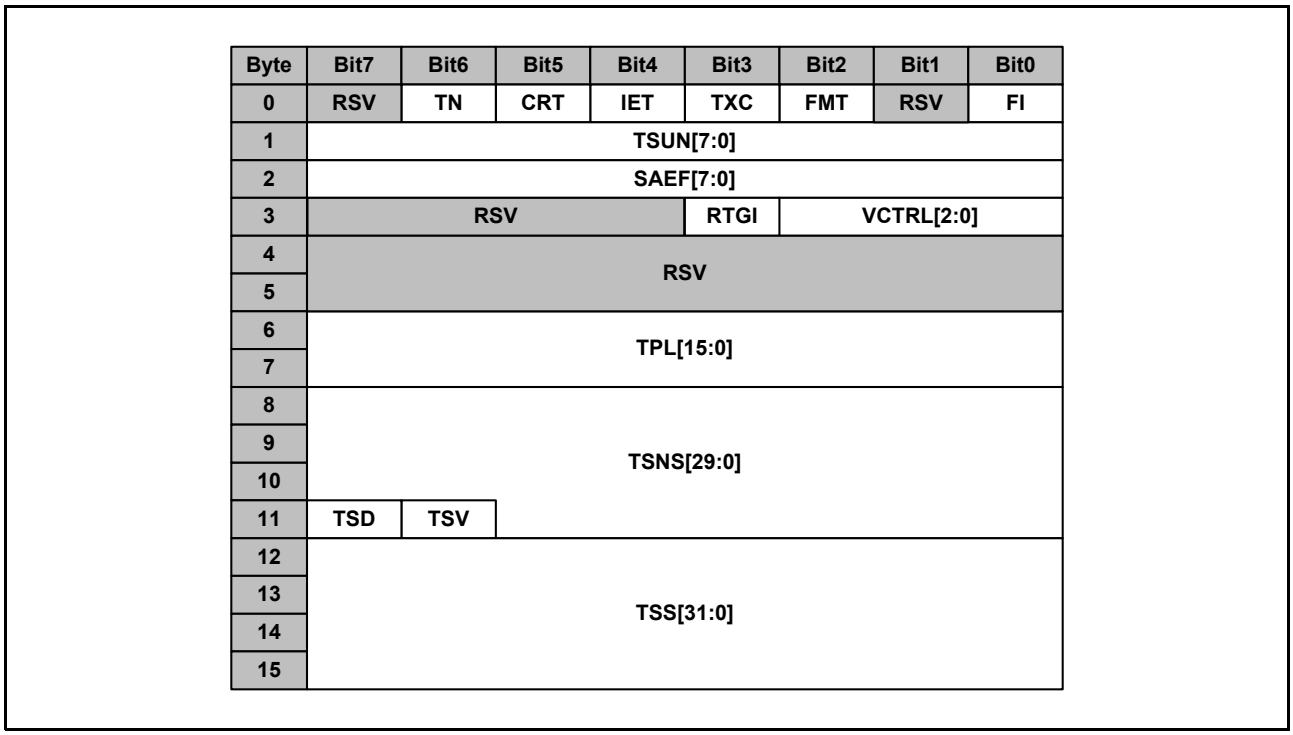


Figure 38.24 Ethernet Local Descriptor Format

Table 38.17 Ethernet Local Descriptor Field Description (1/2)

Field name	Bit width	Description	Value for GWCA
FI	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
FMT	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
TXC	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
IET	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
CRT	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
TN	1	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
TSUN	8	Refer to Table 38.15.	Copied from transmission ethernet descriptor. Refer to (b) Transmission Ethernet Descriptor.
SAEF	8	Source agent error flags	Refer to (c) SAEF Format.
VCTRL	3	VLAN control	VCTRL[2] values: Set to GWVCC.VIM. VCTRL[1:0] values 00b: For No TAG, R-TAG and Unknown TAG frames, refer to (1) TAG Information Extraction. 01b: For C-TAG and CR-TAG frames, refer to (1) TAG Information Extraction. 10b: For SC-TAG and SCR-TAG frames, refer to (1) TAG Information Extraction. 11b: For CoS-TAG and CoSR-TAG frames, refer to (1) TAG Information Extraction.

Table 38.17 Ethernet Local Descriptor Field Description (2/2)

Field name	Bit width	Description	Value for GWCA
RTGI	1	R-TAG in	0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames, refer to (1) TAG Information Extraction. 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames, refer to (1) TAG Information Extraction.
TPL	16	Total payload length	Payload length of frame. It only includes the data saved in the data RAM [MFWD].
TSV	1	Timestamp valid	Set to 0. Unused in GWCA.
TSD	1	Timestamp default	Set to 0. Unused in GWCA.
TSNS	30	Timestamp nanosecond [GPTP]	Set to 00000000h. Unused in GWCA.
TSS	32	Timestamp second [GPTP] Restrictions: • HW: The 16-upper bits of the gPTP [GPTP] second part are truncated.	Set to 00000000h. Unused in GWCA.
RSV	—	Reserved area	Set to 0

(c) SAEF Format

Source agent error flag format is described in Figure 38.25 and Table 38.18.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	DNE	RSV	SEQE	AXE	RSV	RSV	TFE

Figure 38.25 Source Agent Error Flag Format

Table 38.18 Source Agent Error Flag Field Description

Field name	Bit width	Description
TFE	1	TAG Filtering Error [Setting conditions] • HW: When a frame does not fit the required TAG ingress format. Refer to GWTTFC register explanation.
AXE	1	AXI Error [Setting conditions] • HW: When an AXIE error happened during a split frame, the frame is sent incomplete.
SEQE	1	Sequence Error [Setting conditions] • HW: When a FSINGLE, FSTART, EOS or unknown descriptor is received during a split frame, the frame is sent incomplete.
DNE	1	Descriptor Number Error • HW: When a number GWMDNC.TXDMN[4:0]+1 of descriptor has been already read to fetch a frame during a spit frame and the frame is not finished. The frame is sent incomplete.
RSV	—	Reserved Set to 0

38.3.3 Data Reception

GWCA allows data reception through the GWCA RX data path, The RX data path is described in Figure 38.26.

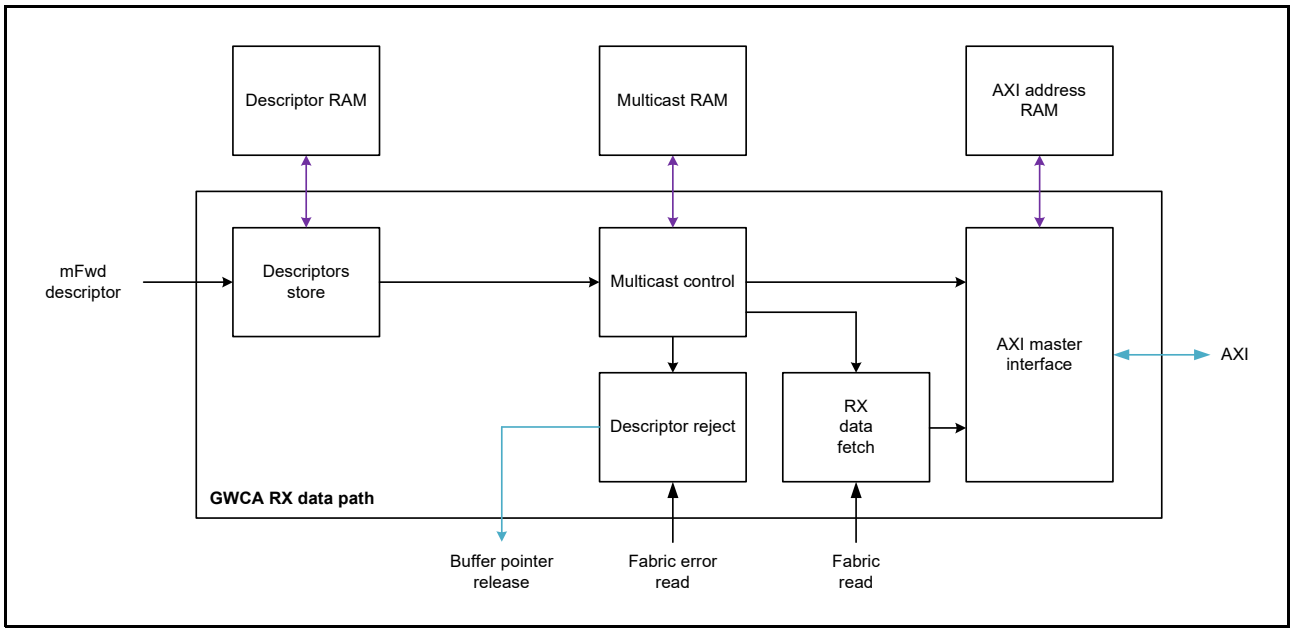


Figure 38.26 GWCA RX Data Path Block Diagram

The RX data path is separated in five blocks:

- Descriptor store: This block aims at storing the descriptor received from the Forwarding Engine [MFWD] in the Descriptor RAM and controls the arbitration between descriptors.
- Multicast control: This block aims at checking the frame size and security level and decide to forward or discard frames and at copying the descriptors to be able to send the data at several CPU sub-destinations.
- Descriptor reject: This block aims at releasing the pointer of the frames rejected by Frame size control block. This block is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- RX data fetch: This block fetches frames data from the local RAM, update frames depending on the information obtained by L2/L3 update module, on VLAN control information and the R-TAG information, can remove frame FCSs, and release frame pointers. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- AXI master interface: This block handles the data/AXI descriptor exchange with the CPU.

38.3.3.1 Descriptor Store

Descriptor store has two functions. Storing the descriptors coming from the descriptor bus [MFWD] in the descriptor RAM and arbitrating descriptor read from the descriptor RAM to decide in which order the descriptors will be sent to the CPU.

(1) Descriptor Storage

Descriptor storage controls the descriptor storage using GWIRC.IPVRi[2:0], GWRDQSC.RDQSLq, GWRDQC.RDQDq, GWRDQC.RDQPq and GWRDQDCq.DQD[9:0] bits and can be monitored using GWRDQMq.DNQ[9:0] and GWRDQMLMq.DMLQ[9:0] bits and GWEIS1.DQOESq and GWEIS1.DQSESq flags. Figure 38.27 describes an example for Descriptor storage setting.

[Functions]

- GWIRC.IPVRi[2:0] bits are used to map descriptor received from forwarding engine with a given FDESCR.IPV [MFWD] value to a determined descriptor queue. For example, by setting GWIRC.IPVR6[2:0] to 1 all the descriptors received with FDESCR.IPV equal to 6 will be stored in descriptor queue number 1.
- GWRDQSC.RDQSLq is used to set a queues security level. If a queue is secure (GWRDQSC.RDQSLq is set) and a non-secure descriptor is received (FDESCR.SEC [MFWD] is not set) for this queue, the descriptor will not be stored and GWEIS1.DQSESq flag will be set.
- GWRDQC.RDQPq bit is used to pause descriptor queues. For paused descriptor queues, the descriptor transmission to CPU will stop but the descriptor storage in the descriptor RAM will continue.
- GWRDQC.RDQDq bit is used to disable descriptor queues. For disabled descriptor queues and no descriptor will be stored in the Descriptor RAM.
- GWRDQDCq.DQD[9:0] bits are used to set the maximum number of descriptors that can be stored in each descriptor queue. If a descriptor is received for descriptor q while queue q is full (GWRDQDCq.DQD[9:0] = GWRDQMq.DNQ[9:0]), the descriptor will not be stored and GWEIS1.DQOESq flag will be set.
- GWRDQMq.DNQ[9:0] bits are used to monitor the current number of descriptors in each descriptor queue.
- GWRDQMLMq.DMLQ[9:0] bits are used to monitor the maximum number of descriptors that has been held in each queue since the previous reset or the previous time corresponding register has been read.
- GWEIS1.DQOESq flag signals that descriptors have been discarded because of a descriptor queue overflow.
- GWEIS1.DQSESq flag signals that descriptors have been discarded because of a descriptor queue security error.

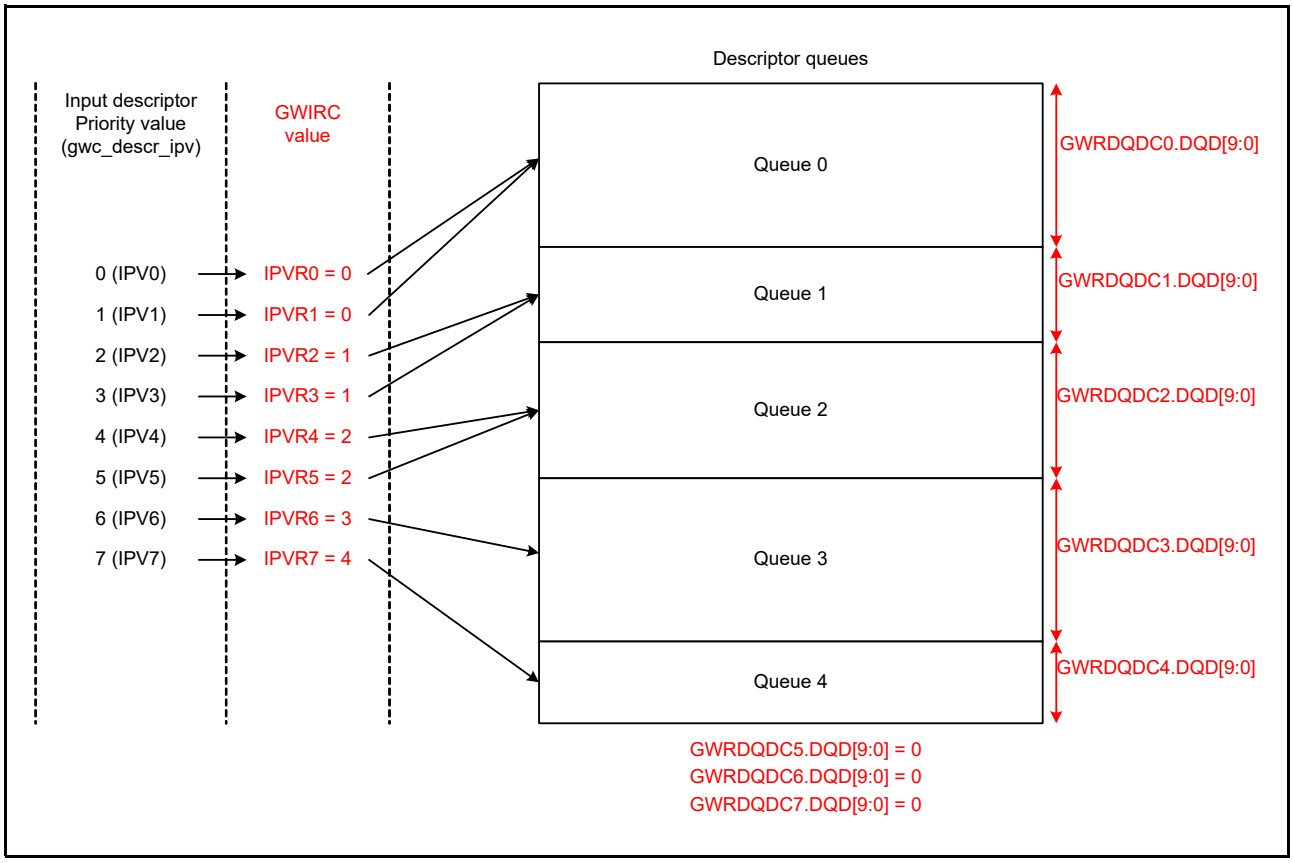


Figure 38.27 Descriptor Storage Setting Example8 == 8

(2) Descriptor Arbitration

Descriptor arbitration controls the descriptor reading order for reception using GWRDQAC register. It has 4 modes. The Strict Priority, the Round Robin, the Weight Round Robin and the Hybrid mode. This function is used to ensure QoS when the AXI access allows a throughput which is too small to allow all the data to be stored in the user RAM.

(a) Strict priority (SP)

To set the descriptor arbitration in strict priority mode, all the GWRDQAC.RDQAi[3:0] (i = 0 to 7) bits should be set to 0000b. In this case a strict priority algorithm will be applied for descriptor arbitration. The strict priority algorithm is described through an example in Figure 38.28.

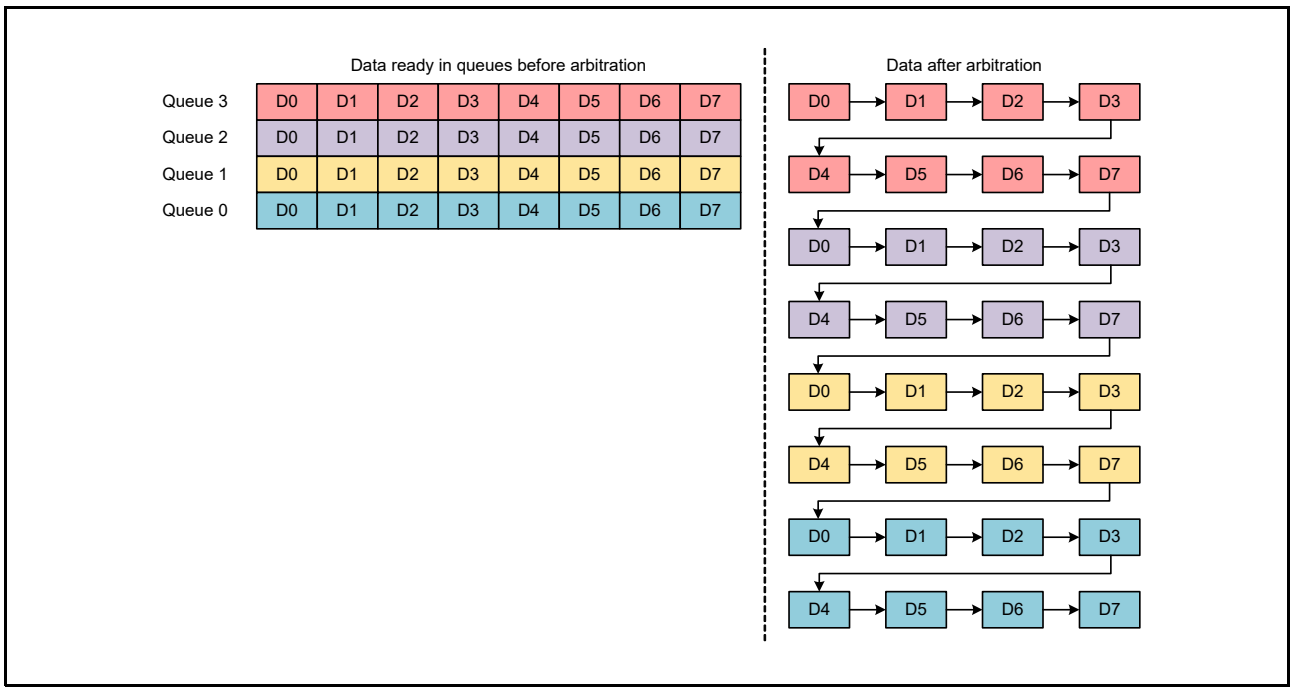


Figure 38.28 Strict Priority Arbitration Example

(b) Round Robin (RR)

To set the descriptor arbitration in round robin mode, all the GWRDQAC.RDQAi[3:0] (i = 0 to 7) bits should be set to 1111b. In this case a round robin algorithm will be applied for descriptor arbitration. The round robin algorithm is described through an example in Figure 38.29.

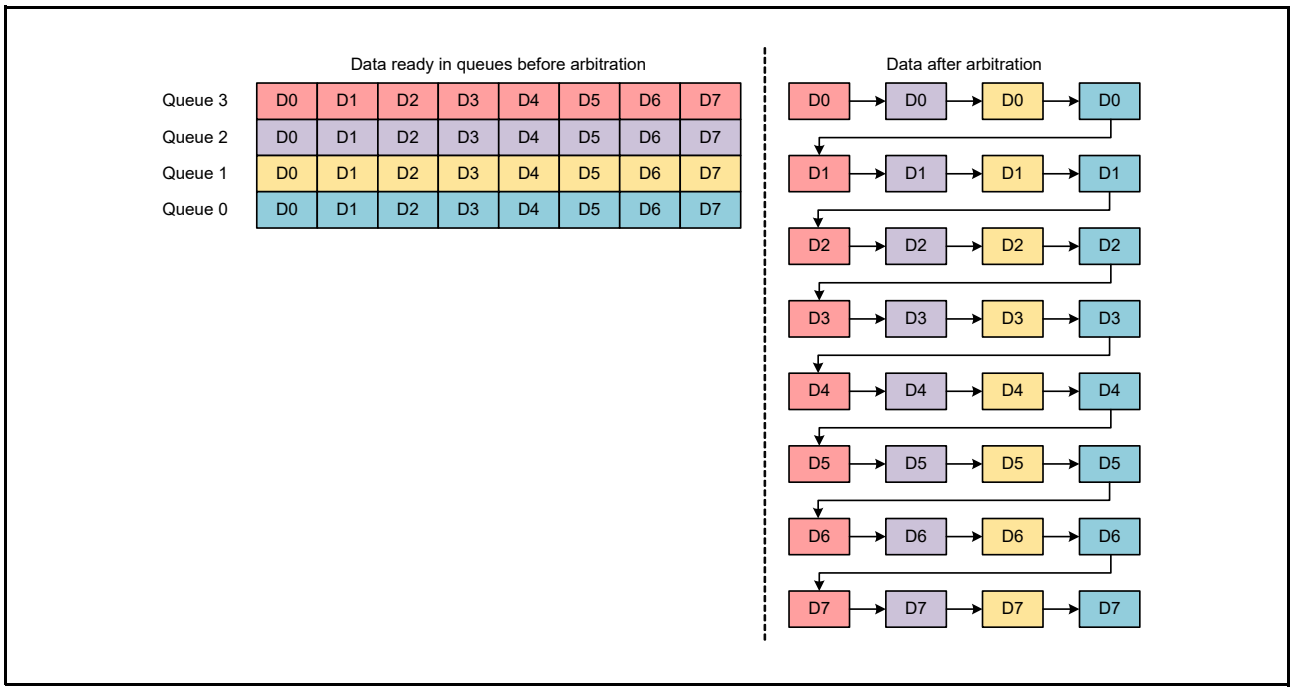


Figure 38.29 Round Robin Arbitration Example

(c) Weight Round Robin (WRR)

To set the descriptor arbitration in weight round robin mode, all the GWRDQAC.RDQAi[3:0] (i = 0 to 7) bits should be set to a value different than 0 with at least one queue set to a value different than 1. In this case a weight round robin algorithm will be applied for descriptor arbitration. The weight round robin algorithm is described through an example in Figure 38.30 (RDQA0 = 2, RDQA1 = 1, RDQA2 = 3 and RDQA3 = 4). In this arbitration GWRDQAC.RDQAi[3:0] bits represents the weight of queue i. The heavier a queue is the more chance it will have to transmit data.

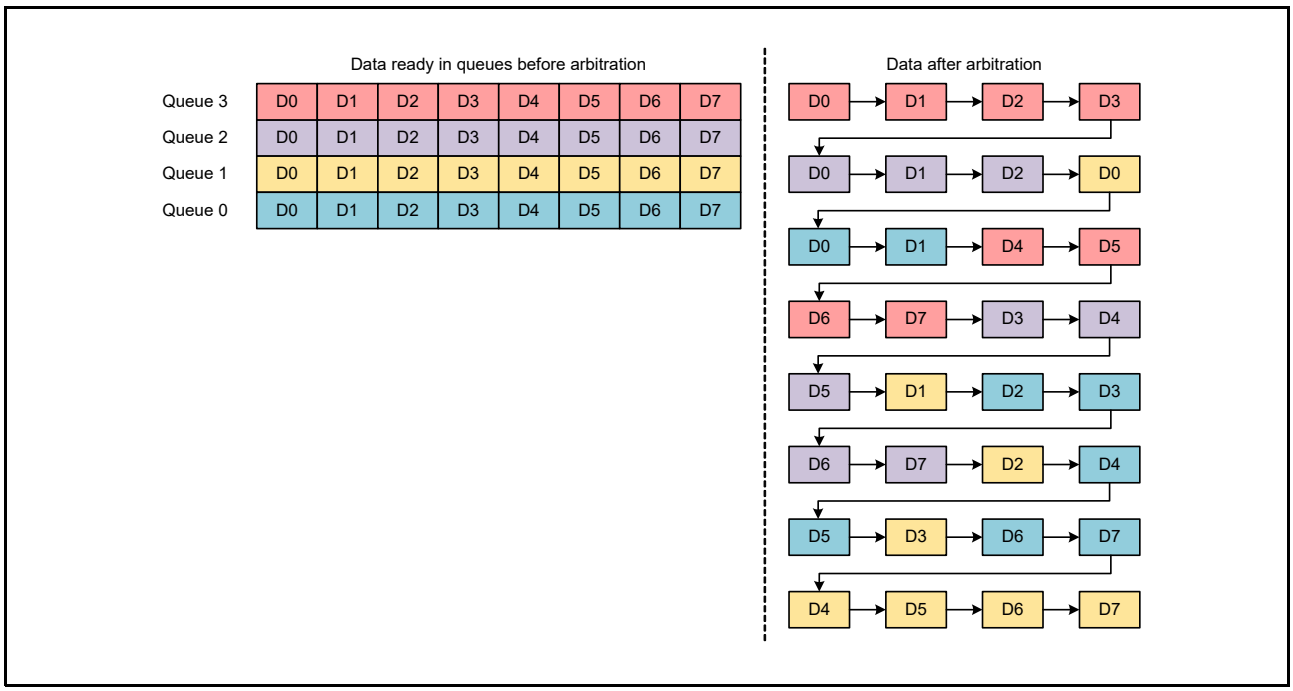


Figure 38.30 Weight Round Robin Arbitration Example

(d) Hybrid arbitration

It is allowed to use at the same time SP and RR or SP and WRR. This mode is called the hybrid arbitration mode. To set the descriptor arbitration in hybrid arbitration mode, some of the GWRDQAC.RDQAi[3:0] (i = 0 to 7) bits should be set to 0 and some others to a value different than 0 (For restrictions refer to GWRDQAC.RDQAi[3:0] bits explanation). The hybrid arbitration algorithm is described through an example in Figure 38.31 (RDQA0[3:0] = 0, RDQA1[3:0] = 1, RDQA2[3:0] = 2 and RDQA3[3:0] = 0).

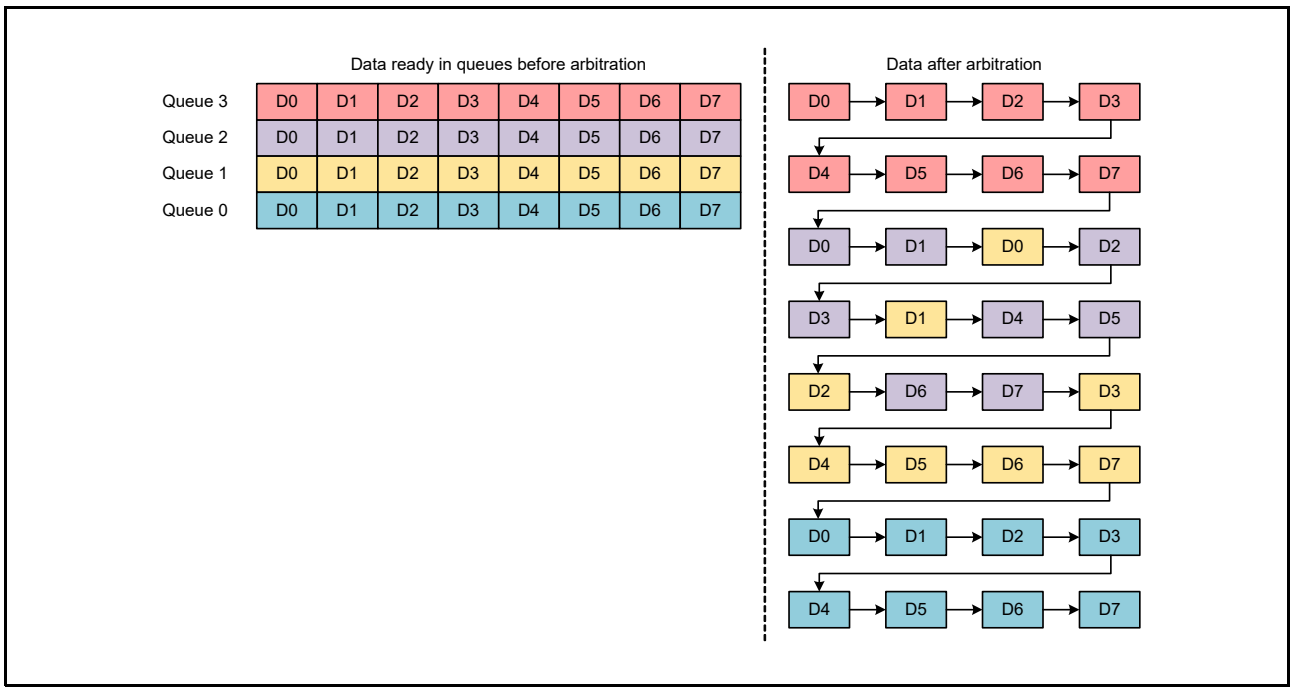


Figure 38.31 Hybrid Arbitration Example

38.3.3.2 Multicast Control

Multicast function aims at copying the same frame to several AXI descriptor queue. It allows different CPU to process same frames at the same time without interfering with each other. If nothing is set, all the descriptor queues are initialized as unicast queues.
Multicast control has three functions. Checking the frame size, security level and queue type (TX or RX), modifying the multicast table using multicast learning and reading the multicast table using multicast searching.

(1) Frame Check

Frame check aims at checking if a frame has the right properties to be received by CPU using GWRMFSCq.MFS[15:0], GWDCCi.SL, GWDCCi.DQT bits and can be monitored using GWEIS0.FSESq, GWEIS4.DSSES, GWEIS4.DSSEIOS, GWEIS4.DSSECN[5:0], GWEIS5.DCTES, GWEIS5.DCTEIOS, GWEIS5.DCTECN[5:0] flags.
[Functions]

- GWRMFSCq.MFS[15:0] bits are used to set the maximum frame size accepted for each descriptor queue. Any frame received for queue q with a size bigger than GWRMFSCq.MFS[15:0] will be discarded and GWEIS0.FSESq flag will be set. The frame size considered is the size after VLAN update and with FCS. In this case multicast would not happen and the frame will totally be discarded.
- GWDCCi.SL bit is used to set the security level of a CPU sub destination. Any frame received unsecure (FDESCR.SEC [MFWD] is not set) for a secure CPU sub destination (GWDCCi.SL is set) discarded and GWEIS4.DSSES flag will be set (during multicast, for all CPU sub destination for which GWDCCi.SL is set frame will be discarded and for all CPU sub destination for which GWDCCi.SL is not set, frame will be received).
- GWDCCi.DQT is used to control is the received frame is going to a reception descriptor queue. Any frame received for a transmission descriptor queue will be discarded and GWEIS5.DCTES flag will be set. In case of multicast, only the targeted transmission queue will see their data lost.
- GWEIS0.FSESq flag signals that a descriptor has been rejected because the corresponding frame where oversized.
- GWEIS4.DSSES, GWEIS4.DSSEIOS and GWEIS4.DSSECN[5:0] flag signal that a descriptor has been rejected because of an AXI descriptor queue security error.
- GWEIS5.DCTES, GWEIS5.DCTEIOS, GWEIS5.DCTECN[5:0] flag signal that a descriptor has been rejected because of a descriptor chain type error.

(2) Descriptor Copy

Descriptor copy aims at multicasting descriptor to send them to the AXI master interface and the RX data fetch using the multicast table.

The multicast table is divided in two fields, MUL.MN[2:0] which is the multicast number and MUL.MNRCN[5:0] which is the next descriptor chain and is used to link AXI descriptor chains together. AXI descriptor chains linked together form a multicast chain to which a frame will be multicast.
Because the multicast table is based on links all the multicast patterns are not possible.

The multicast table utilization is described in Figure 38.32 through an example.

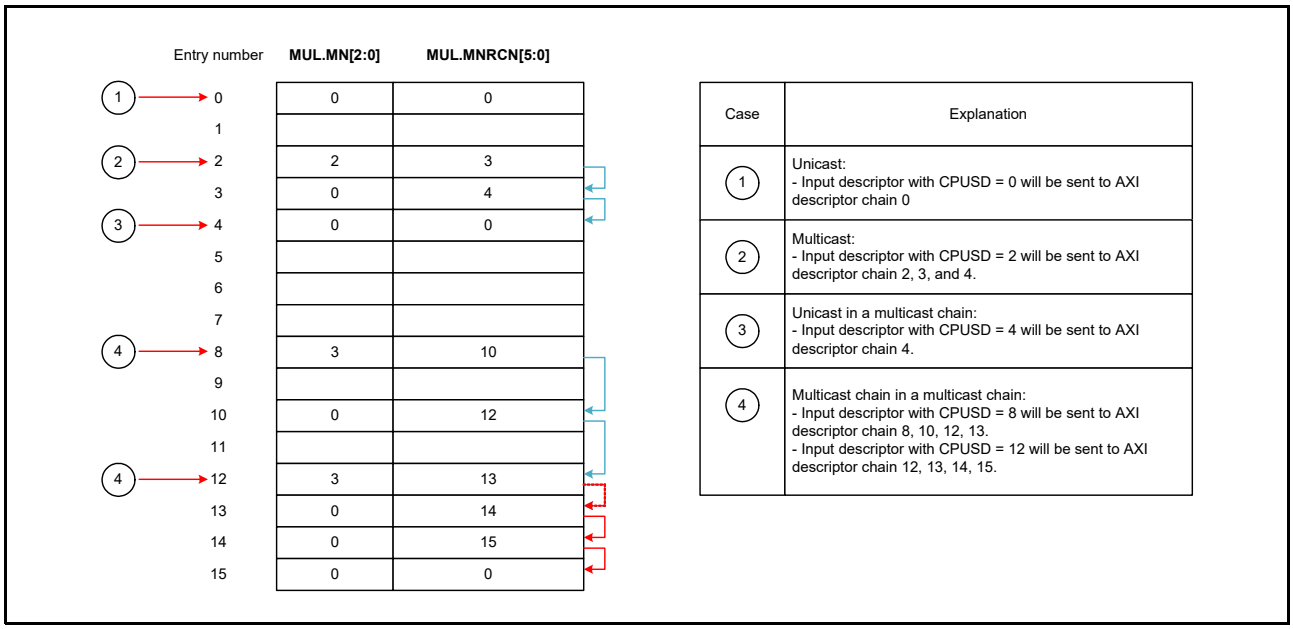


Figure 38.32 Multicast Table Utilization Example

(3) Multicast Learning

Multicast learning aims at updating entries in the multicast table using GWMSTLS and GWMSTLR registers. Table 38.19 describes mapping between learning registers and fields in multicast table. Table 38.20 describes the learn results, their mapping with the multicast table and how to handle them. It is possible to multicast a frame to 8 AXI descriptor chains.

[Functions]

- GWMSTLS register is used to set the information needed to overwrite an entry.
- GWMSTLR register is used to check if the previously set entry has been overwritten.
- To set entries in the multicast table, refer to software flow in section 38.3.8.6, Multicast Table Setting Flow.
- If GWAC.AMP = 1 (the AXI transaction is pending), learning will not complete. Release the AXI master pause.

Table 38.19 Learn Registers/Multicast Table Mapping Registers

Register name	Field name in L3 table
GWMSTLS.MSENL[5:0]	Not present in multicast table, used to indicate which entry should be overwritten.
GWMSTLS.MNL[2:0]	MUL.MN, refer to (2) Descriptor Copy
GWMSTLS.MNRCNL[5:0]	MUL.MNRCN, refer to (2) Descriptor Copy

Table 38.20 Learn Result

Register name	Field name	Field explanation
GWMSTLR.MTLF	Learning fail	Learning fail will happen in one of the following conditions: When an APB tries to learn a multicast entry and the multicast table is not ready (GWMTIRM.MTR is not set)

(a) Static Learning

When a system aims at using the multicast table in a static way (set in CONFIG mode only) there is no restriction regarding the entry format, the entry link pointers and the learning order of entries.

(b) Dynamic Learning

When a system aims at using the multicast table in a dynamic way (set in OPERATION mode), the restrictions described in Figure 38.33 should be respected.

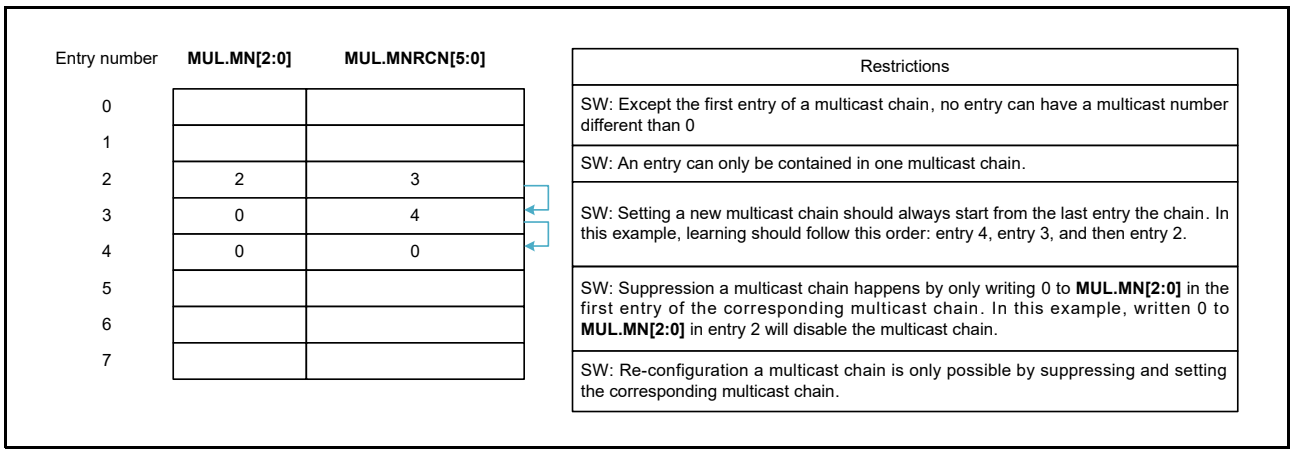


Figure 38.33 Multicast Table Dynamic Utilization Restrictions

(4) Multicast Searching

Multicast searching aims at reading entries in the multicast table using GWMSTSS and GWMSTSR registers. Table 38.21 describes mapping between learning registers and fields in multicast table.

[Functions]

- GWMSTSS register is used to set the entry that should be read.
- GWMSTSR register is used to read the read result.
- To read an entry in the multicast table, refer to software flow in section 38.3.8.7, Multicast Table Read Flow.

Table 38.21 Search Registers/Multicast Table Mapping Registers

Register name	Field name in L3 table
GWMSTSS.MSENS[5:0]	Not present in multicast table, used to indicate which entry should be read.

Table 38.22 Search Result

Register name	Field name/ corresponding field in multicast table	Field explanation
GWMSTSR.MNR[2:0]	MUL.MN	Multicast table read value
GWMSTSR.MNRCNR[5:0]	MUL.MNRCN	Multicast table read value

38.3.3.3 RX Data Fetch

RX data fetch has follow functions. Fetching the data from the Local RAM, un-tagging/VLANs depending on VLAN settings, handling FCS, sending the data the AXI master interface and releasing the buffer pointer of the read frames. The data fetching from Local RAM, the data sending to the AXI master interface and the pointer release are here for switch hardware purpose and its description is not required for switch utilization so it will not be described.

(1) VLAN(SC-TAG)

Fetching the data from the Local RAM, updating the data depending on “L23 update” and “TAG function registers”. Updating VLANs data path is described in Figure 38.34, VLAN Data Path.

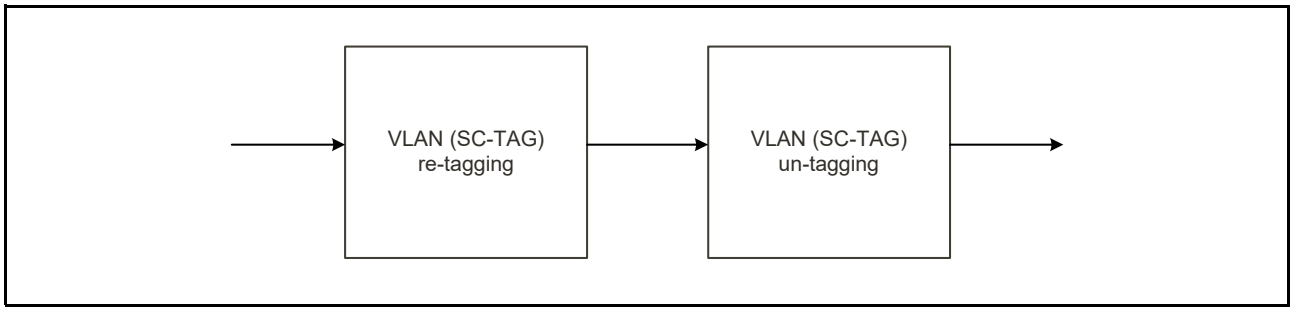


Figure 38.34 VLAN Data Path

(a) VLAN(SC-TAG) Re-Tagging

Layer 2 update can happen for the following fields:

- Destination MAC address updated to L23U.MDA when L23U.MDAU set.
- Source MAC address is updated to {GWMAC0.MAU[15:0], GWMAC1.MAL[31:0]} when L23U.MSAU set.
- C-TAG VID is updated to L23U.CVID when L23U.CVIDU set.
- C-TAG PCP is updated to L23U.CPCP when L23U.CPCPU set.
- C-TAG VID is updated to L23U.CDEI when L23U.CDEIU set.
- S-TAG VID is updated to L23U.SVID when L23U.SVIDU set.
- S-TAG PCP is updated to L23U.SPCP when L23U.SPCPU set.
- S-TAG VID is updated to L23U.SDEI when L23U.SDEIU set.

[Restriction]

- HW: To update the C-TAG/S-TAG fields, the C-TAG/S-TAG should be present in the frame after VLAN un-tagging (refer to (1) VLAN(SC-TAG)). If not preset, the update will not happen (e.g. L2 update does not change the frame format and does not permit TAG insertion).

(b) VLAN(SC-TAG) Un-Tagging

VLAN update aims at modifying VLAN frame format depending on switch VLAN mode (FWGC.SVM[1:0] bits in forwarding engine [MFWD]), the VLAN control information given by the forwarding engine descriptor (FDESCR.VCTRL [MFWD]) and GWVCC.VEM[2:0] bits.

[Function]

- GWVCC.VEM[2:0] bits set the VLAN reception mode.

The frames format after VLAN un-tagging are described in Table 38.23, Table 38.24 and Table 38.25 per Switch VLAN mode (FWGC.SVM[1:0] bits in forwarding engine [MFWD]) and VLAN control information. The conversion between the frame stored in the Local RAM and the frame format after VLAN un-tagging is described in Figure 38.35.

Table 38.23 Frame Format After VLAN Un-Tagging: Switch in No-VLAN Mode [MFWD]

VCTRL	GWVCC.VEM[2:0]				
	000b	001b	010b	011b	100b
0	No TAG	No TAG	No TAG	No TAG	No TAG
1	No TAG	C-TAG	C-TAG	C-TAG	C-TAG
2	No TAG	C-TAG	C-TAG	SC-TAG	SC-TAG
3	No TAG	CoS-TAG	CoS-TAG	CoS-TAG	CoS-TAG
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA
7	NA	NA	NA	NA	NA

Table 38.24 Frame Format After VLAN Un-Tagging: Switch in C-VLAN Mode [MFWD]

VCTRL	GWVCC.VEM[2:0]				
	000b	001b	010b	011b	100b
0	No TAG	No TAG	C-TAG	No TAG	C-TAG
1	No TAG	C-TAG	C-TAG	C-TAG	C-TAG
2	No TAG	C-TAG	C-TAG	SC-TAG	SC-TAG
3	No TAG	CoS-TAG	C-TAG	CoS-TAG	C-TAG
4	No TAG	No TAG	C-TAG	No TAG	C-TAG
5	No TAG	No TAG	C-TAG	No TAG	C-TAG
6	No TAG	No TAG	C-TAG	No TAG	C-TAG
7	No TAG	CoS-TAG	C-TAG	CoS-TAG	C-TAG

Table 38.25 Frame Format After VLAN Un-Tagging: Switch in SC-VLAN Mode [MFWD]

VCTRL	GWVCC.VEM[2:0]				
	000b	001b	010b	011b	100b
0	No TAG	No TAG	C-TAG	No TAG	SC-TAG
1	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
2	No TAG	C-TAG	C-TAG	SC-TAG	SC-TAG
3	No TAG	CoS-TAG	C-TAG	CoS-TAG	SC-TAG
4	No TAG	No TAG	C-TAG	No TAG	SC-TAG
5	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
6	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
7	No TAG	CoS-TAG	C-TAG	CoS-TAG	SC-TAG

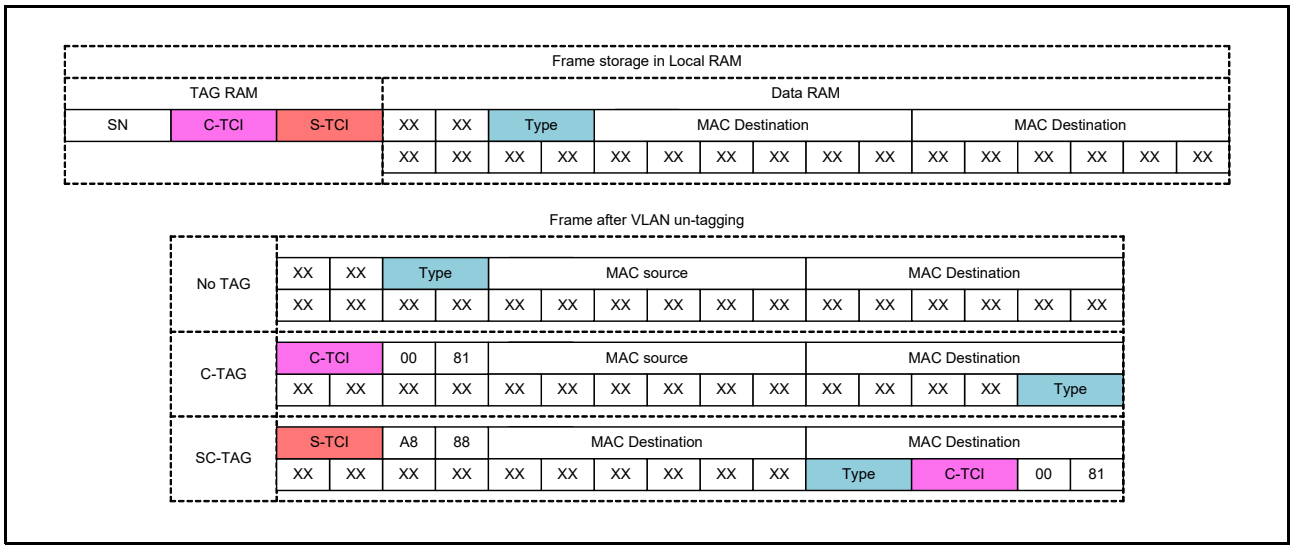


Figure 38.35 Local RAM Frame to Frame Format after VLAN Un-Tagging Conversion

Note: The Local RAM frame to frame format after VLAN un-tagging conversion for CoS-TAG is not described because it is the same than for C-TAG except that the VID in the C-TCI will be fixed to 0.

Note: 8100h corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].

Note: 88A8h corresponds to S-TAG TPID and S-TCI corresponds to a S-TAG TCI [802.1Q].

Note: Type corresponds to any Ethernet type value which is not F1C1h, 88A8h or 8100h.

Note: TPID values can be changed in Forwarding Engine [MFWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

(2) R-TAG Insertion

The R-TAG insertion happens for the following cases:

- For frames received with an R-TAG (LDESCR.RTGI was set during reception, refer to (3) Frame Saving in Local RAM, so, in descriptor received for forwarding, FDESCR.RTGI was also set) and for which routing is not requested.

The R-TAG sequence number (SN in Figure 38.36) will be set to the sequence number FDESCR.SEQN [MFWD] received in received descriptor.

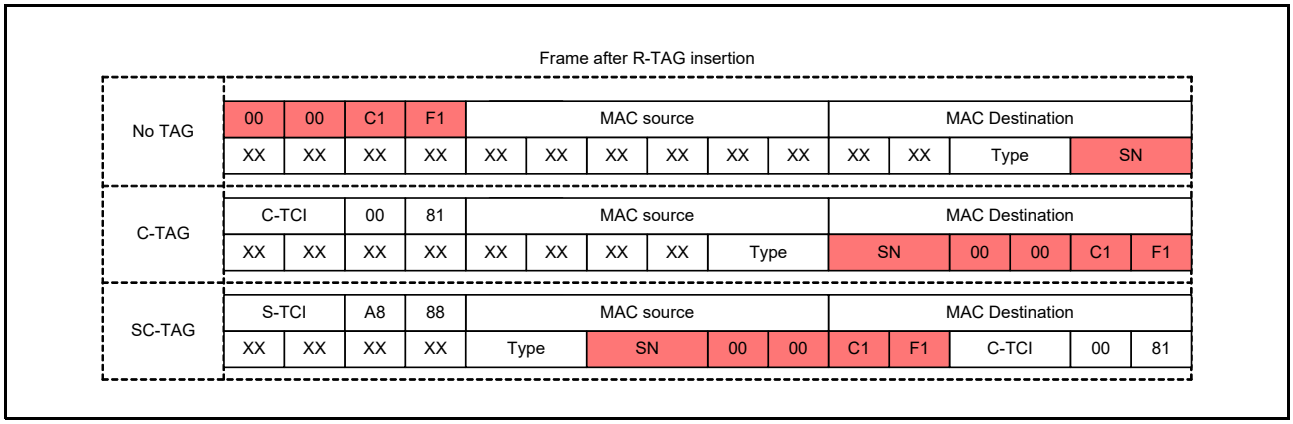


Figure 38.36 R-TAG inserted frame format

Note: F1C1h corresponds to R-TAG TPID and SN correspond to R-TAG sequence number[802.1CB].
Note: 8100h corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].
Note: 88A8h corresponds to S-TAG TPID and S-TCI corresponds to a S-TAG TCI [802.1Q].
Note: Type corresponds to any Ethernet type value which is not F1C1h, 88A8h or 8100h.
Note: TPID values can be changed in Forwarding Engine [MFWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

(3) FCS Handling

Ethernet frame FCS can be forwarded to the CPU using GWRGC.RCPT bit.

[Function]

- GWRGC.RCPT is used to pass frame FCSs to the CPU.

However, following conditions should be met for GWCA to forward a frame FCS to the CPU:

- For Ethernet agent [ETHA], RMAC should not have removed the FCS [RMAC] at the input port.
- For GWCA, the FCS should be present in the ingress frame and valid (FI set and FW not set in input descriptor, refer to (3) Data Transmission Common Descriptor Format).
- The VLANs should be the same at the egress than at the ingress port.
- The frame should not be a routed frame (received descriptor FDESCR.RV field not set [MFWD])

Note: If one of the above conditions is not met by a frame, the FCS will be removed from it even if GWRGC.RCPT bit is set to 1.

Note: If a frame is received by CPU with an FCS, FI flag will be set in the reception descriptor. If not, it will not be set.

38.3.3.4 AXI Master Interface

The AXI master interface handles the data/AXI descriptor exchange with the CPU. It will receive the updated data and the information related to the data for RX data fetch block and store them in the CPU RAM. To do so, it uses GWDCCi.EDE, GWDCCi.ETS, GWDCCi.SM[1:0] bits, and can be monitored using GWAARSS, GWAARSR0 and GWAARSR1, GWIDAUASi, GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS2i.DFESt, GWEIS3.IAOESi, GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN[5:0] flags.

[Functions]

- GWDCCi.EDE and GWDCCi.ETS bits set the descriptor format that will be used for descriptor queue number i refer to (1) General Formats.
- GWDCCi.SM[1:0] bits set how GWCA will write back descriptors. In this section all explanation will use GWDCCi.SM[1:0] equal to 00b. For GWDCCi.SM[1:0] equal to 01b or 10b refer to the register explanation.
- For GWAARSS, GWAARSR0 and GWAARSR1 registers refer to section 38.3.1.6, AXI Address RAM Searching.
- GWIDAUASi, GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS3.IAOESi interrupt register are used for incremental area monitoring, refer to (3) Single Page Incremental Data Reception.
- GWEIS2i.DFESt flag signals that frames have been lost because a full descriptor queue.
- GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN[5:0] flags signal that frames have been received with an unexpected size, refer to(2) Size-Controlled Data Reception.

Data reception can be done using linear or cyclic descriptor queues (refer to section 38.3.1.5, AXI Descriptor Queue Format). Data reception process descriptors can be set six different ways, through the basic data reception descriptor chain (refer to (1) Basic Data Reception), the Size-controlled data reception descriptor chain (refer to (2) Size-Controlled Data Reception), the Single page incremental data reception descriptor chain (refer to (3) Single Page Incremental Data Reception), the Interrupt based multi-page incremental data reception descriptor chain (refer to (4) Interrupt Based Multi-Page Incremental Data Reception), the Header removal incremental data reception descriptor chain (refer to (5) Header Removal Incremental Data Reception).

(1) Basic Data Reception

Basic data reception process descriptor utilization is described in Figure 38.37. Basic data reception process descriptor format is described in Figure 38.38 and Table 38.26.

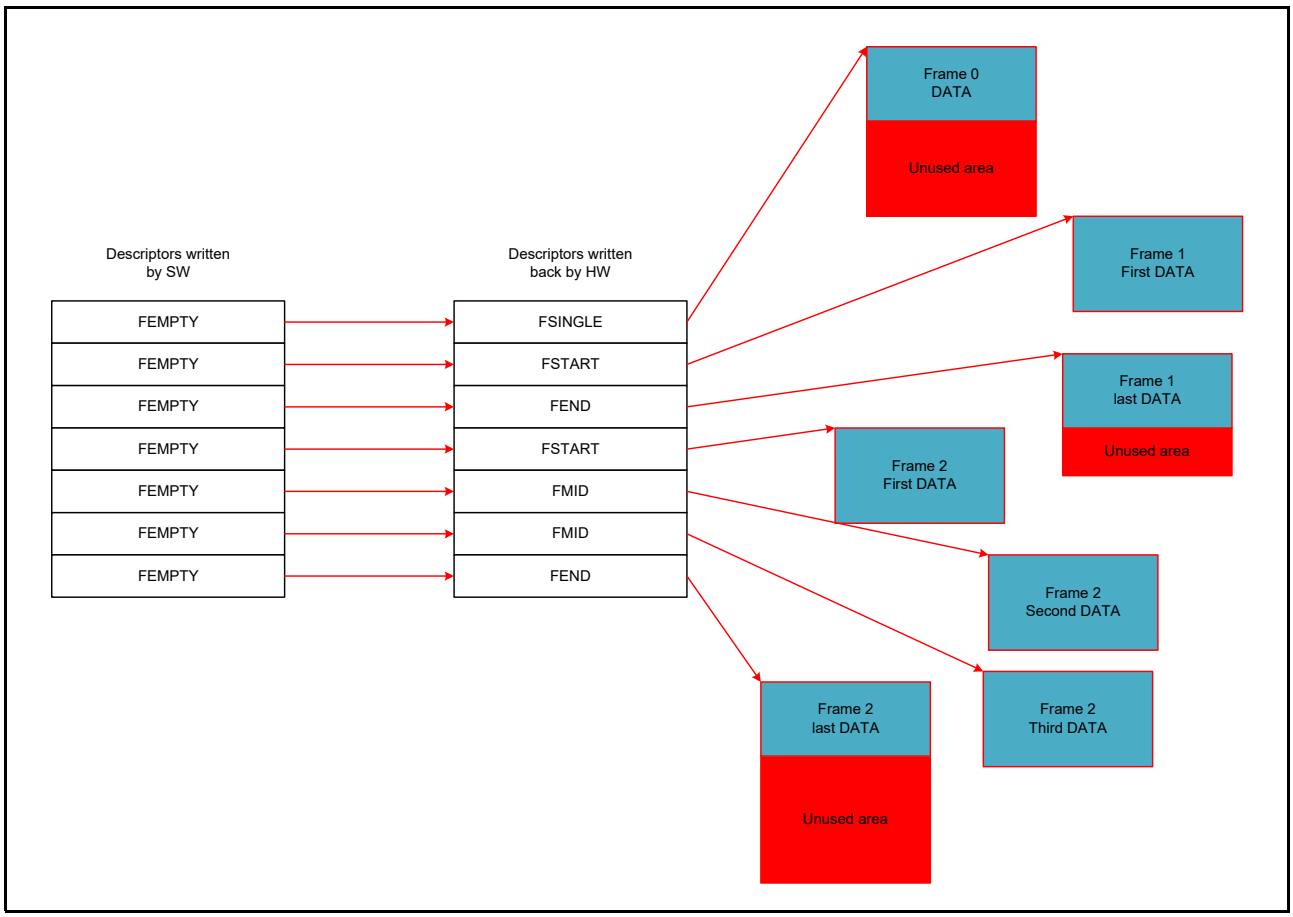


Figure 38.37 Basic Data Reception Process Descriptor Utilization

Note: If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND and all other descriptors will be FMID.

Note: When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.

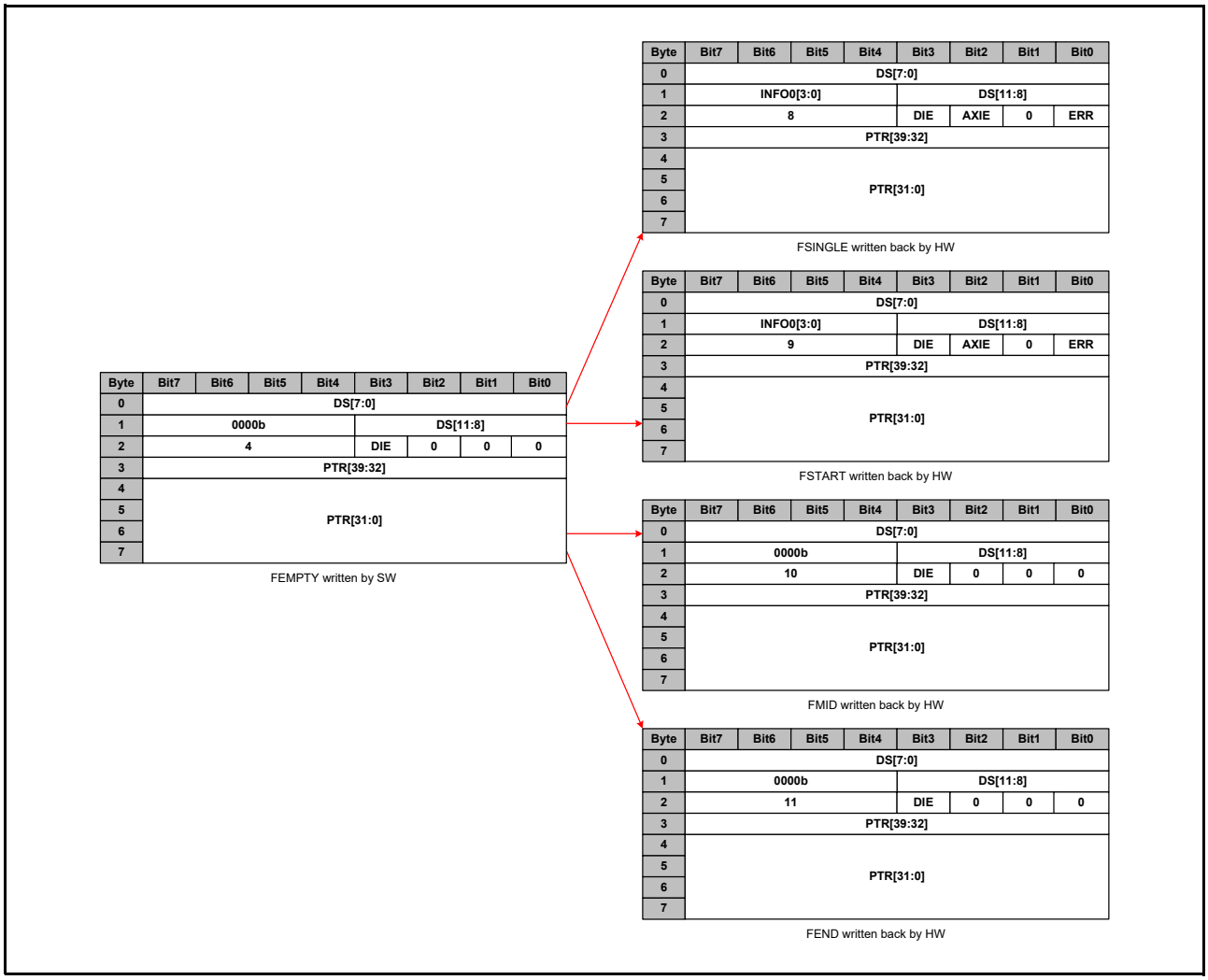


Figure 38.38 Basic Data Reception Process Descriptor Utilization Format (for Basic Descriptor, Refer to (1) General Formats)

Table 38.26 Basic data reception process descriptor utilization field description

Field name	Field description before write-back	Field description after write-back
DS	Size of available URAM memory associated to the pointer for data storage	Actual size that has been written by the AXIBMI in the URAM.
INFO0	0000b	0000b for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
ERR	0	[Setting conditions] • HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. • HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0
AXIE	0	[Setting conditions] • HW: An AXI error happened while writing a data (BRESP ≠ 00b) and a FSINGLE or a FSTART descriptor written back. • HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	4 for FEMPTY	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	Address where data should be written in the URAM	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
TS (GWDCCi.ETS = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.

(2) Size-Controlled Data Reception

Size-controlled data reception aims at controlling the frame size at reception while purposely splitting the header and the payload of a frame. By size-controlled data reception the SW ensure that descriptor order is maintained even if a too short or too long frame is received. Size-controlled data reception process descriptor utilization is described in Figure 38.39. Size-controlled data reception process descriptor format is described in Figure 38.40 and Table 38.27. Size-controlled data reception can be monitored using GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN[5:0] flags.

[Function]

- GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN[5:0] flags signal a frame received with un unexpected size.

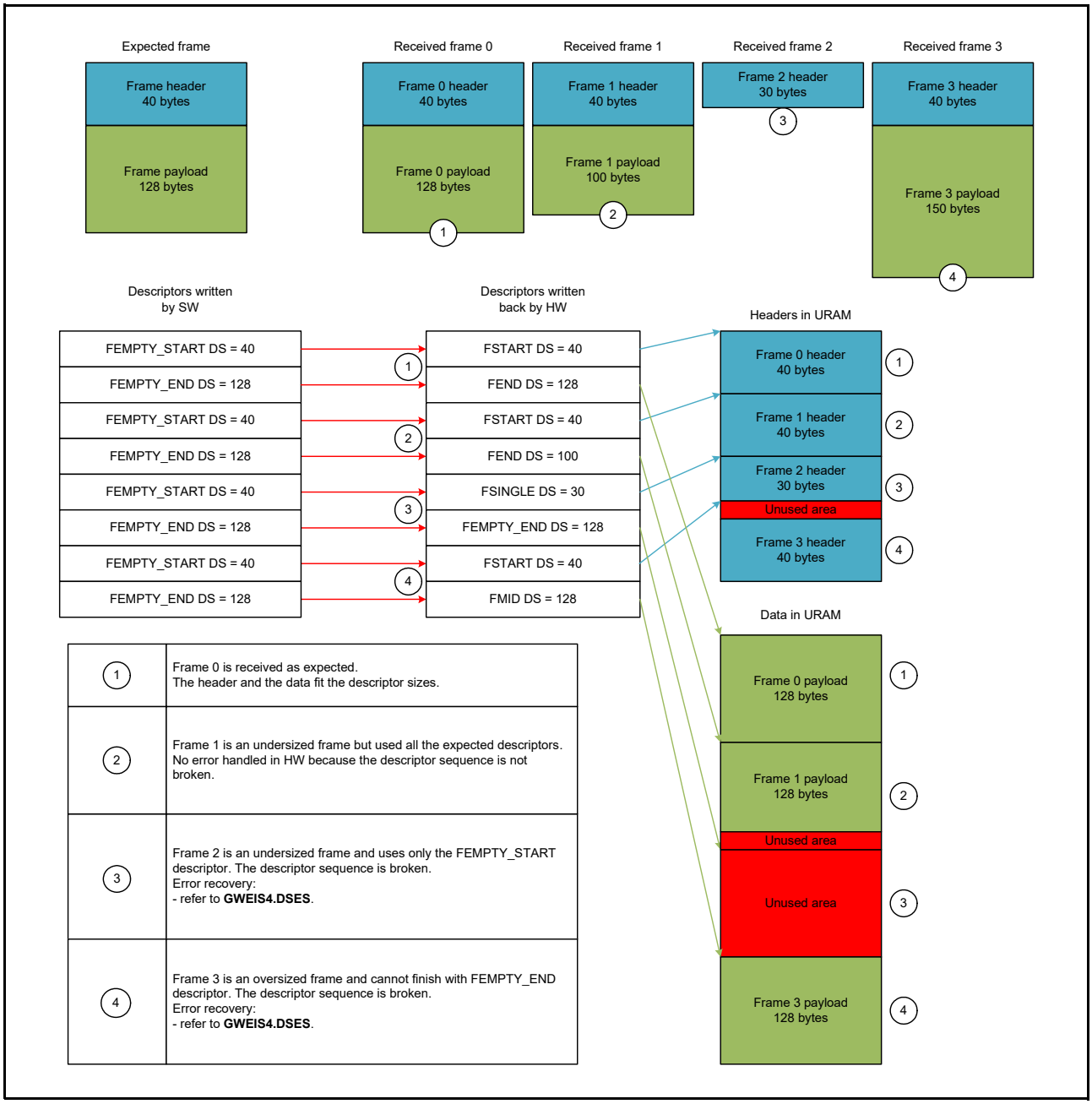


Figure 38.39 Size-Controlled Data Reception Process Descriptor Utilization

Note: When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.

Note: A frame can be split in more than two descriptors by adding FEMPTY_MID descriptors between

FEMPTY_START and FEMPTY_END descriptors.

[Restriction]

- SW: The process descriptor queue should always repeat the pattern of the same descriptors from the first descriptor to the descriptor before the termination descriptor. This pattern should start with an FEMPTY_START descriptor, finish by a FEMPTY_END descriptor and in between be only composed by FEMPTY_MID descriptors. If this is violated (for example, FEMPTY_START is detected in the middle), the frame will be discarded/rejected without notification.

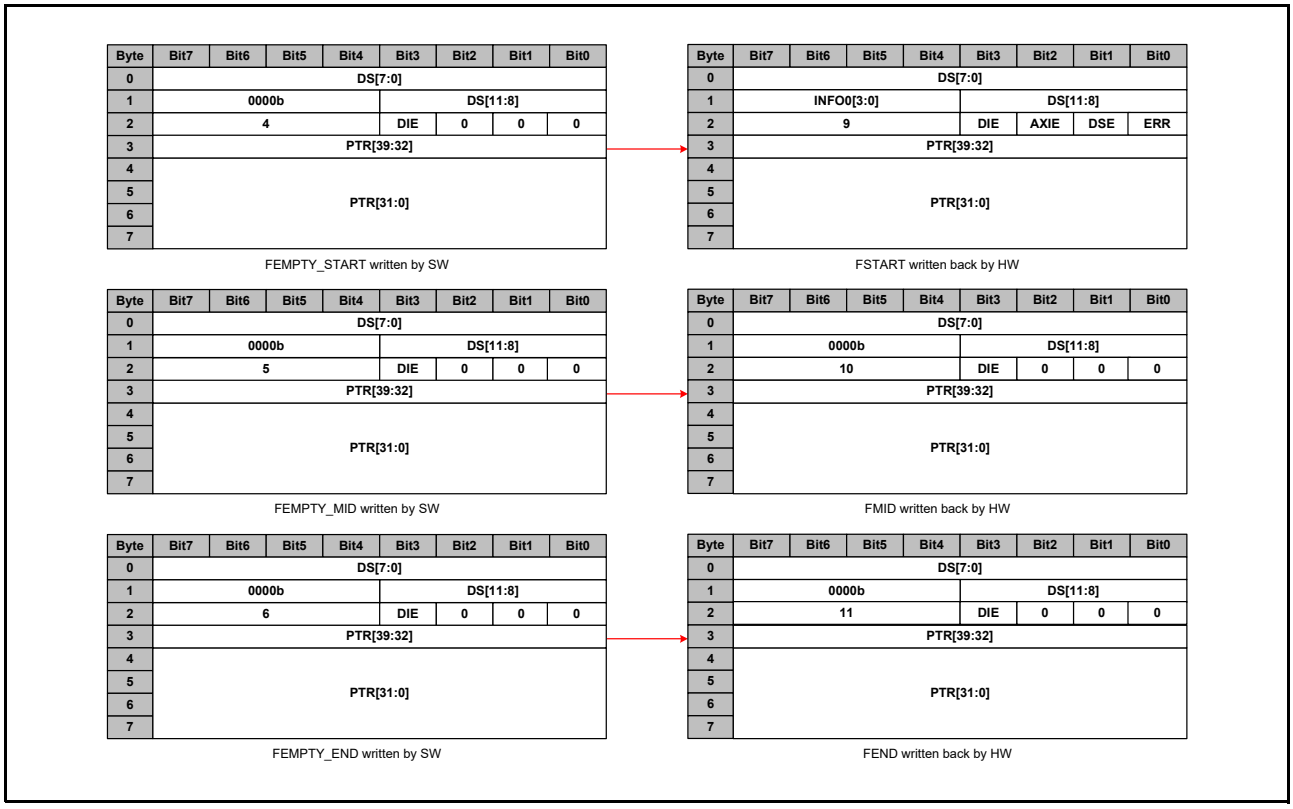


Figure 38.40 Size-controlled Data Reception Process Descriptor Utilization Format (for Basic Descriptor, Refer to (1) General Formats)

Table 38.27 Size-controlled data reception process descriptor utilization field description

Field name	Field description before write-back	Field description after write-back
DS	Size of available URAM memory associated to the pointer for data storage	Actual size that has been written by the AXIBMI in the URAM.
INFO0	0000b	0000b for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
ERR	0	[Setting conditions] • HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. (This case should not happen due to the process descriptor order) • HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	[Setting conditions] • HW: The frame has been finished with a FEMPTY_START or a FEMPTY_MID descriptor (undersized frame) • HW: The frame is too big to finish with a FEMPTY_END descriptor (oversized frame)
AXIE	0	[Setting conditions] • HW: An AXI error happened while writing a data (BRESP ≠ 00b) and a FSINGLE or a FSTART descriptor written back. • HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	5 for FEMPTY_START 6 for FEMPTY_MID 7 for FEMPTY_END	9 for FSTART 10 for FMID 11 for FEND
PTR	Address where data should be written in the URAM	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
TS (GWDCCi.ETS = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.

(3) Single Page Incremental Data Reception

Single page incremental data reception aims at saving the received frames back to back in a given memory area. By single page incremental data reception, the SW avoid wasting CPU power to stack data together. Single page incremental data reception process descriptor utilization is described in Figure 38.41. Single page incremental data reception process descriptor format is described Figure 38.42, Table 38.28, Figure 38.43 and Table 38.29. Single page incremental data reception HW/SW synchronization should be done by GWIDAUASi register and can be monitored by GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS3.IAOESi flag.

[Functions]

- GWIDAUASi register shows the number of bytes that has been written in the incremental area. After reading the data from the CPU RAM, SW should write to this register the quantity of data that has been read. The CPU data handling can be done, either based on the written back descriptor in the corresponding descriptor queue, or on the byte value contained in the increment area that can be read in GWIDAUASi register.
- GWIDASMi register is used to monitor the total size of the incremental area.
- {GWIDASAMi0.IDASAU[7:0], GWIDASAMi1.IDASAL[31:0]} bit array is used to monitor the start address of the incremental area.
- {GWIDACAMi0.IDASAU[7:0], GWIDACAMi1.IDASAL[31:0]} bit array is used to monitor the next used address in the incremental area.
- GWEIS3.IAOESi flag signals data loss due to incremental area overflow.

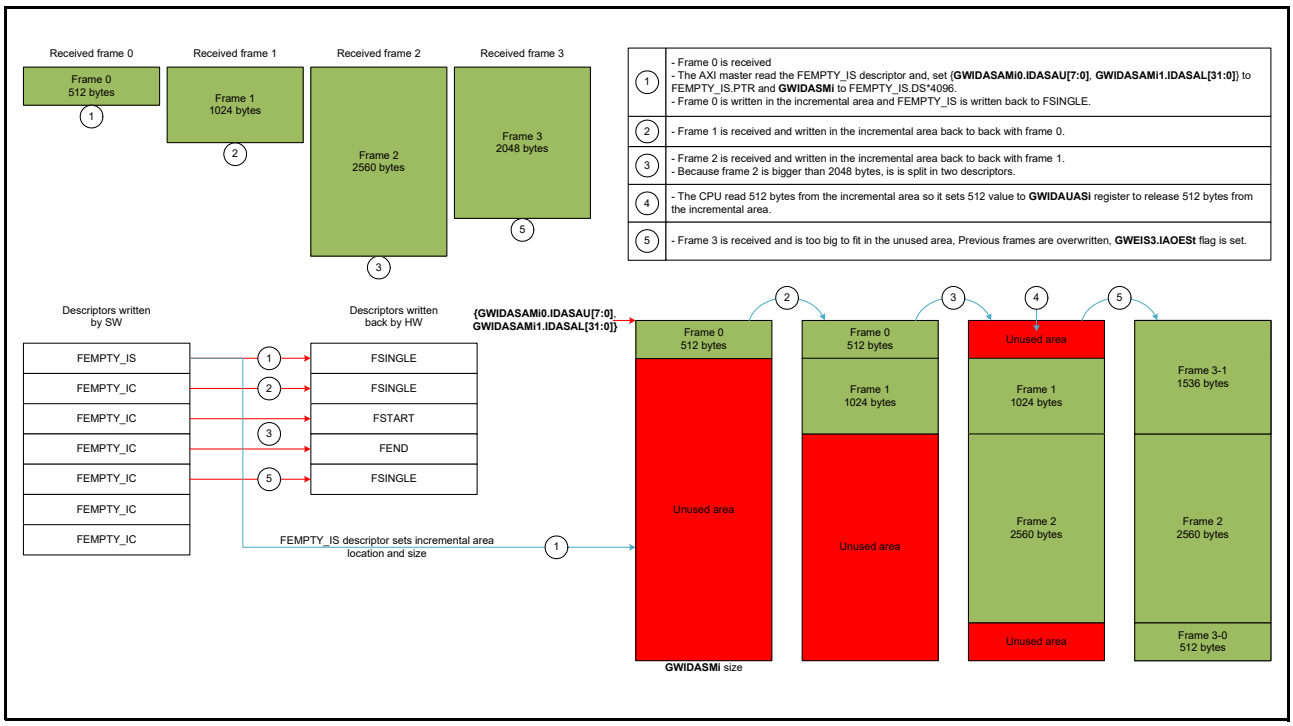


Figure 38.41 Single Page Incremental Data Reception Process Descriptor Utilization (for Basic Descriptor, Refer to (1) General Formats)

[Restriction]

- SW: This reception type can only be used for the first 4 RX chains, refer to section 38.3.1.1, AXI Descriptor Queue Mapping.

Note: If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND and all other descriptors will be FMID.

Note: When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep

DT mode, this order is kept.

Note: When GWCA reset has been applied (GWMS.OPS[1:0] set to 00b), all the incremental registers (i.e incremental page start address, current address, page size and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS[1:0] set to 11b) and a frame has been received by GWCA incremental chain, it's not be able to write frame using FEMPTY_IC descriptor. Descriptor of AXI current address have to be(set up) a FEMPTY_IS.

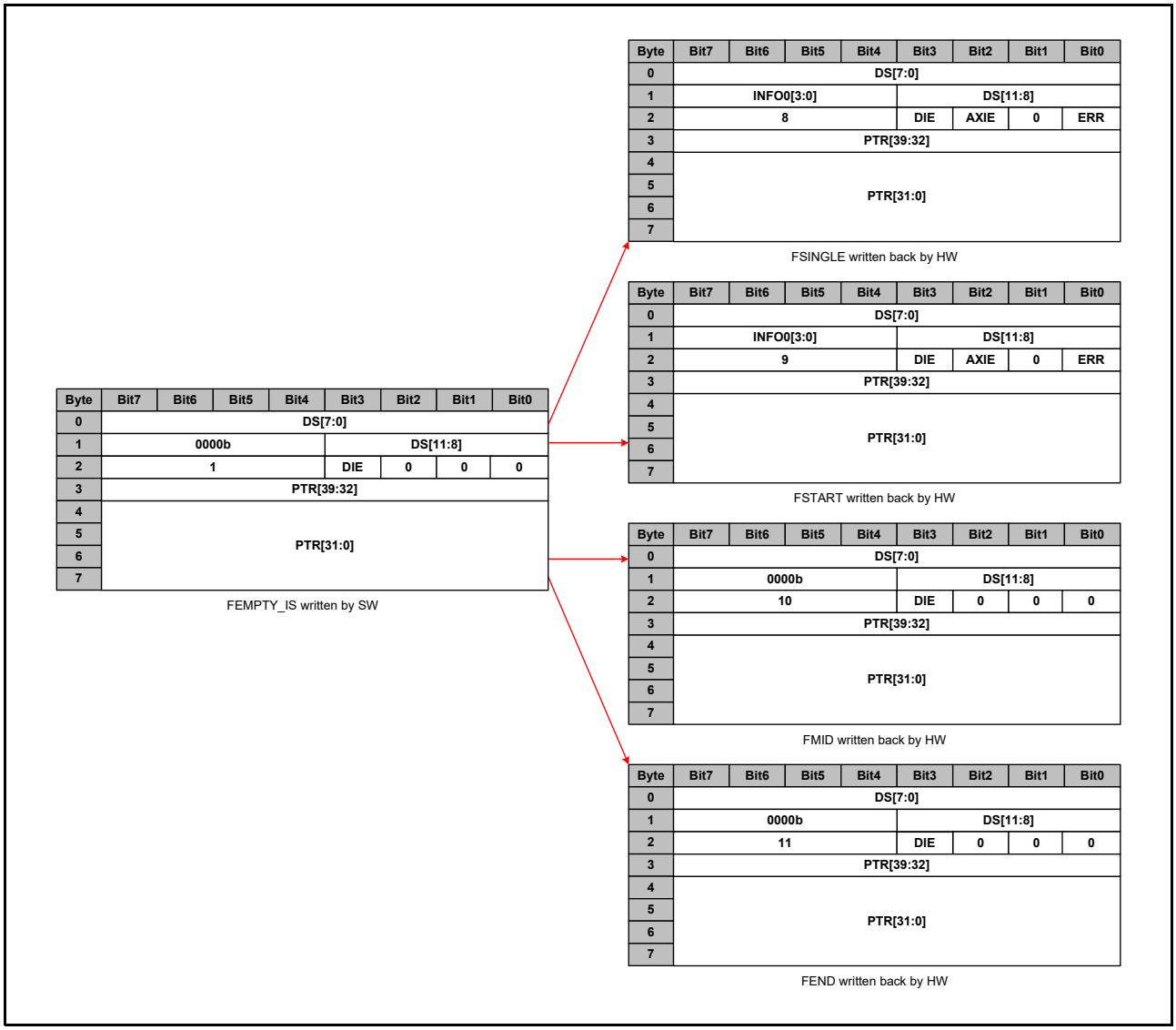


Figure 38.42 Single Page Incremental Data Reception FEMPTY_IS Process Descriptor Utilization Format

Table 38.28 Single Page Incremental Data Reception FEMPTY_IS Process Descriptor Utilization Field Description

Field name	Field description before write-back	Field description after write-back
DS	Size of the incremental area in 4 Kbytes unity.	Size of the data stored in the URAM associated to the descriptor.
INFO0	0000b	0000b for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
ERR	0	[Setting conditions] • HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. • HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0
AXIE	0	[Setting conditions] • HW: An AXI error happened while writing a data (BRESP ≠ 00b) and a FSINGLE or a FSTART descriptor written back. • HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	1 for FEMPTY_IS	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	Address where the incremental area starts in the URAM.	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
TS (GWDCCi.ETS = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.

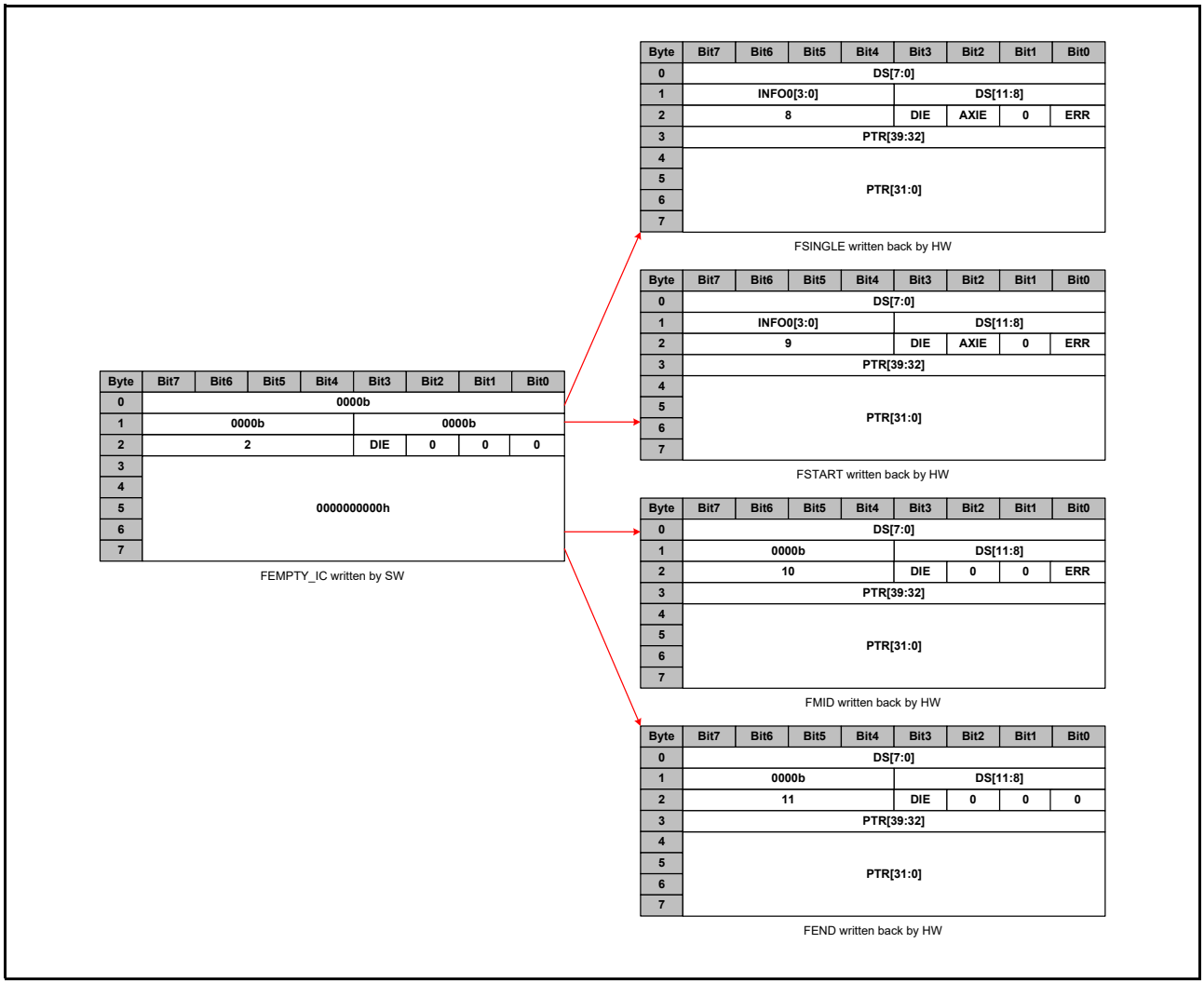


Figure 38.43 Single Page Incremental Data Reception FEMPTY_IC Process Descriptor Utilization Format

Table 38.29 Single Page Incremental Data Reception FEMPTY_IC Process Descriptor Utilization Field Description

Field name	Field description before write-back	Field description after write-back
DS	000h	Size of the data stored in the URAM associated to the descriptor.
INFO0	0000b	0000b for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
DSE	0	0
AXIE	0	[Setting conditions] • HW: An AXI error happened while writing a data (BRESP ≠ 00b) and a FSINGLE or a FSTART descriptor written back. • HW: An AXI error happened while reading a descriptor for a split frame and the descriptor was not the first.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	2 for FEMPTY_IC	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	0000000000h	Address where the data corresponding to the descriptor has been written in the URAM
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
TS (GWDCCi.ETS = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.

(4) Interrupt Based Multi-Page Incremental Data Reception

Interrupt based multi-page incremental data reception aims at saving the received frames back to back in a given memory area. By Interrupt based multi-page incremental data reception, the SW avoid wasting CPU power to stack data together. Interrupt based multi-page incremental data reception process descriptor utilization is described in Figure 38.44. Interrupt based multi-page incremental data reception process descriptor format are the same than for single page incremental data reception, refer to (3) Single Page Incremental Data Reception.

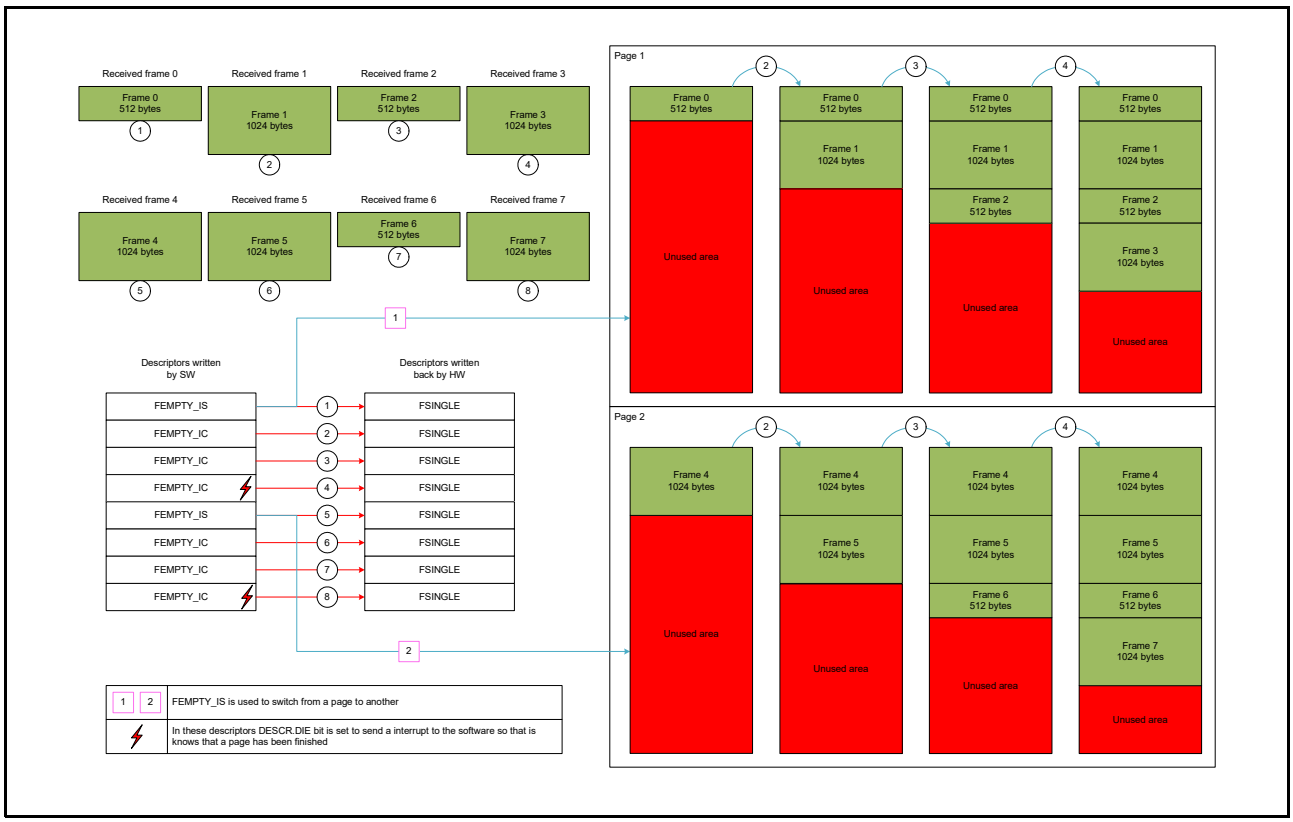


Figure 38.44 Interrupt Based Multi-page Incremental Data Reception Process Descriptor Utilization

[Restrictions]

- SW: It should be ensured that, knowing the expected data and/or controlling page sizes and the number of descriptors for a page, a page cannot overflow.
- SW: This reception type can only be used for the first 4 RX chains.

Note: If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND and all other descriptors will be FMID.

Note: When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.

Note: Interrupt based multi-page incremental data reception allows to use any number of pages.

Note: When GWCA reset has been applied (GWMS.OPS[1:0] set to 00b), all the incremental registers (i.e incremental page start address, current address, page size and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS[1:0] set to 11b) and a frame has been received by GWCA incremental chain, it is not be able to write frame using FEMPTY_IC descriptor. Descriptor of AXI current address have to be(set up) a FEMPTY_IS.

(5) Header Removal Incremental Data Reception

Header removal incremental data reception aims at suppressing header from incremental data. It is an update of single page incremental data reception (refer to (3) Single Page Incremental Data Reception) and interrupt based multi-page incremental data reception (refer to (4) Interrupt Based Multi-Page Incremental Data Reception) by adding a FEMPTY_ND between each incremental descriptor. Header removal incremental data reception process descriptor utilization is described in Figure 38.45. Header removal process descriptor format is described in Figure 38.46 and Table 38.30.

[Restriction]

- SW: A FEMPTY_IS/FEMPTY_IC descriptor can only contain 2048 bytes. It should be taken in account when setting the descriptor in the descriptor queue because the HW does not ensure that the descriptor sequence is correct.

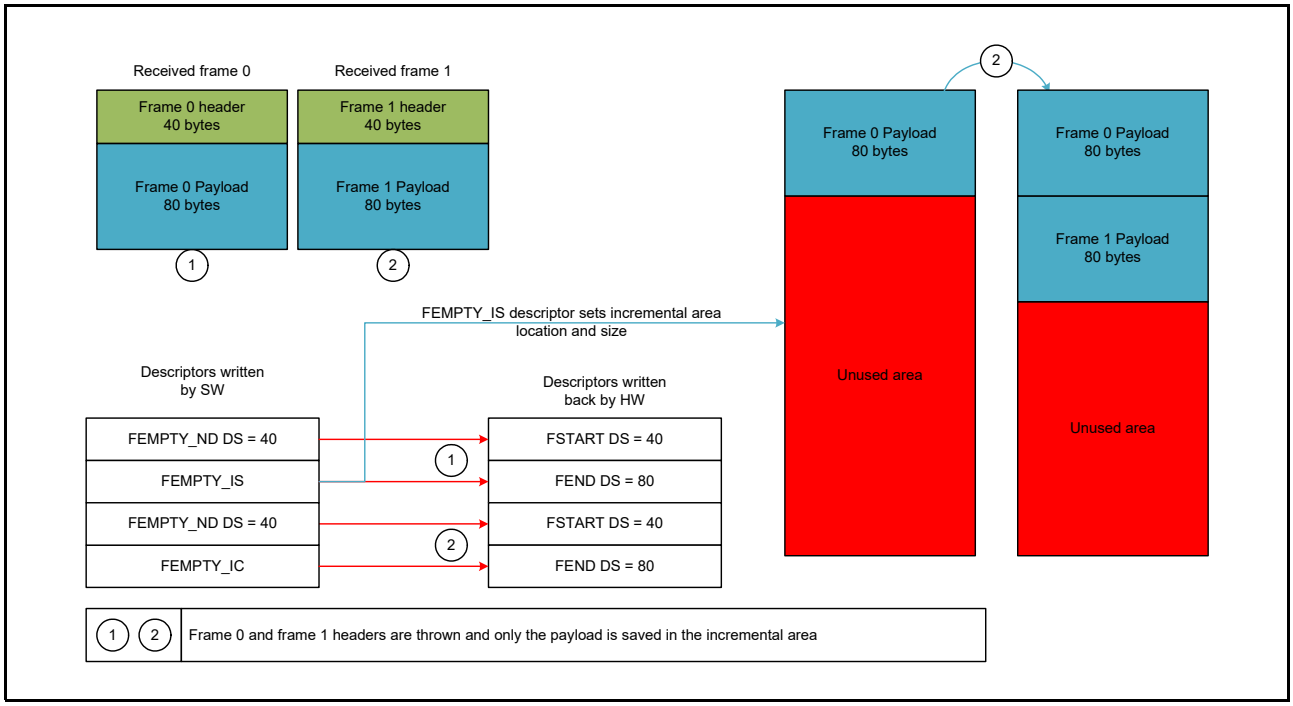


Figure 38.45 Header Removal Incremental Data Reception Process Descriptor Utilization

Note: When using Size controlled data reception, to get rid of the header part of a frame and still keep the descriptor order, it is possible to start every frame with a FEMPTY_START descriptor always pointing toward the same memory address (by setting always the same pointer).

Note: When GWCA reset has been applied (GWMS.OPS[1:0] set to 00b), all the incremental registers (i.e incremental page start address, current address, page size and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS[1:0] set to 11b) and a frame has been received by GWCA incremental chain, it is not be able to write frame using FEMPTY_IC descriptor. Descriptor of AXI current address have to be(set up) a FEMPTY_IS.

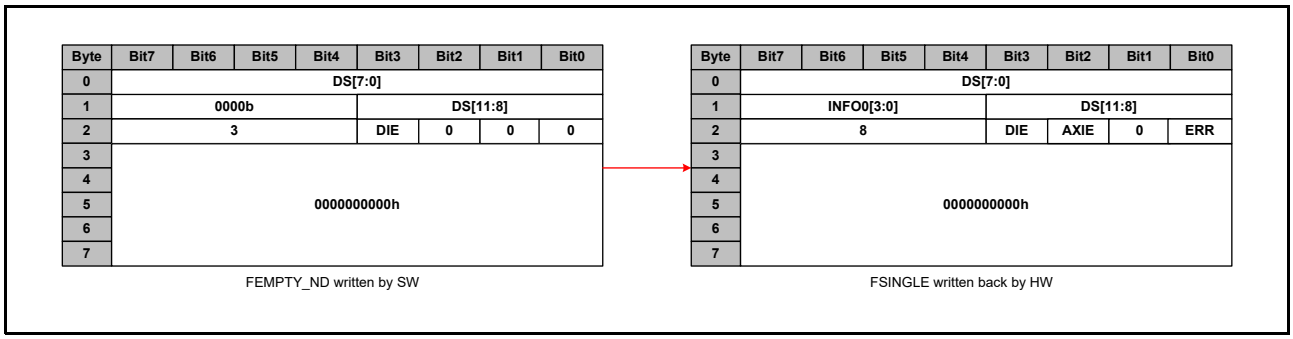


Figure 38.46 Header Removal Utilization Format (for Basic Descriptor, refer to (1) General Formats)

Table 38.30 Header Removal Process Descriptor Utilization Field Description

Field name	Field description before write-back	Field description after write-back
DS	Size to be rejected	Actual size that has been rejected (it should be the same as DS before write-back).
INFO0	0000b	0000b for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
ERR	0	[Setting conditions] • HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. • HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0
AXIE	0	[Setting conditions] • HW: An AXI error happened while writing a data (BRESP ≠ 00b) and a FSINGLE or a FSTART descriptor written back. • HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	3 for FEMPTY_ND	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND Restrictions: • SW: Because the HW turns a FEMPTY_ND descriptor in a FSINGLE, FSTART, FMID or FEND descriptor, by reading the descriptor it is impossible to know that is contains no data. Consequently, SW should remember where the FEMPTY_ND descriptors have been written.
PTR	00000000h	00000000h
INFO1 (GWDCCi.EDE = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE, refer to (6) Data Reception Common Descriptor Format.
TS (GWDCCi.ETS = 1)	0000000000000000h	0000000000000000h for FEND and FMID. For FSTART and FSINGLE refer to (6) Data Reception Common Descriptor Format.

(6) Data Reception Common Descriptor Format

In this section are described the fields which are common for any data transmission (INFO0, INFO1 and TS). There are two types of formats, the reception direct descriptor and the reception ethernet descriptor formats. They are respectively created from direct local descriptors and ethernet local descriptors, refer to (3) Frame Saving in Local RAM. All the fields contained in AXI descriptors, if quoted, will be written ADESCR.{Field name}.

(a) Reception Direct Descriptor

Reception direct descriptor corresponds to frames sent by a CPU and by deciding their target directly (Forwarding engine processing has been by-passed [MFWD]). Reception direct descriptor INFO and TS formats are described in Figure 38.47 and Table 38.31.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	RSV		SEC	FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	RN[7:0]							
4	RSV	SPN[2:0]			RV	RSV		
5	FESF[23:0]							
6								
7								

TS

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	CSD0[6:0]						
1	RSV	CSD1[6:0]						
2	RSV	DV[6:0]						
3	RSV							
4								
5								
6								
7								

Figure 38.47 Reception Direct Descriptor INFO and TS Formats

Table 38.31 Reception Direct Descriptor INFO and TS Field Descriptions

Field name	Bit width	Description	Value for GWCA
FI	1	Refer to Table 38.14.	Refer to (3) FCS Handling.
SEC	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
FMT	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
TXC	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
IET	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
CRT	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
TN	1	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
TSUN	8	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
SAEF	8	Refer to Table 38.16.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
RV	1	Refer to Table 38.14.	Copied from received descriptor FDESCR.RV field [MFWD]
RN	8	Refer to Table 38.14.	Copied from received descriptor FDESCR.RN field [MFWD]
SPN	2	Source port number	Port number from which the frame entered the switch [MFWD] Copied from received descriptor FDESCR.SPN field [MFWD]
FESF	24	Forwarding engine status flags	Copied from received descriptor FDESCR.MINFO field [MFWD]
CSD0	6	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
DV	3	Refer to Table 38.14.	Copied from direct local descriptor. Refer to(a) Direct Local Descriptor.
RSV	—	Reserved area	Set to 0

(b) Reception Ethernet Descriptor

Reception ethernet descriptor corresponds to frames that has been by the Forwarding engine [MFWD]. Reception ethernet descriptor INFO and TS formats are described in Figure 38.48 and Table 38.32.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	RSV		SEC	FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	RSV	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	RN[7:0]							
4	RSV	SPN[2:0]			RV	RSV		
5	FESF[23:0]							
6								
7								

TS

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	TSNS[29:0]							
1								
2								
3	TSD	TSV						
4	TSS[31:0]							
5								
6								
7								

Figure 38.48 Reception Ethernet Descriptor INFO and TS Formats

Table 38.32 Reception Ethernet Descriptor INFO and TS Field Descriptions

Field name	Bit width	Description	Value for GWCA
FI	1	Refer to Table 38.15.	Refer to (3) FCS Handling.
SEC	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
FMT	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TXC	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
IET	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
CRT	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TN	1	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSUN	8	Refer to Table 38.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
SAEF	8	Refer to Table 38.17.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
RV	1	Refer to Table 38.15.	Copied from received descriptor FDESCR.RV field [MFWD]
RN	8	Refer to Table 38.15.	Copied from received descriptor FDESCR.RN field [MFWD]
SPN	2	Source port number	Port number from which the frame entered the switch [MFWD] Copied from received descriptor FDESCR.SPN field [MFWD]
FESF	24	Forwarding engine status flags	Copied from received descriptor FDESCR.MINFO field [MFWD]
TSV	1	Timestamp valid	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSD	1	Timestamp default	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSNS	30	Timestamp nanosecond [GPTP]	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSS	32	Timestamp second [GPTP] Restrictions: • HW: The 16-upper bits of the gPTP [GPTP] second part are truncated.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
RSV	—	Reserved area	Set to 0

38.3.4 Timestamp Reception [GPTP]

GWCA allows TimeStamp reception through the GWCA TS path, The TS path is described in Figure 38.49. Timestamps received through this function are only Ethernet agent TX timestamps. Ethernet agent RX timestamps are stored in the local RAM along with their corresponding data and can only be received by CPU through RX data path with the corresponding data.

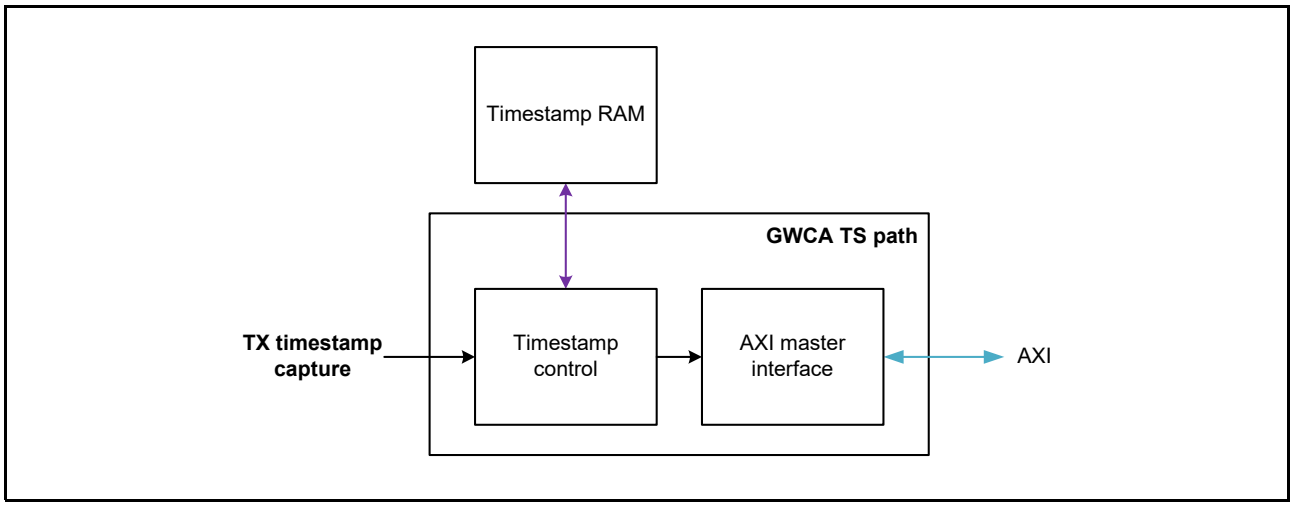


Figure 38.49 GWCA TS Path Block Diagram

The TS path is separated in two blocks:

- Timestamp control: This block aims at receiving the timestamps from TX Timestamp Capture interface [RMAC], storing them in the Timestamp RAM and sending them to the AXI master interface.
- AXI master interface: This block handles the timestamp/AXI descriptor exchange with the CPU.

38.3.4.1 Timestamp Control

Timestamp control controls the descriptor storage/fetching to/from the Timestamp RAM using GWTSDCCs.TE bit and can be monitored using GWTSNM and GWTSMNM registers and GWEIS0.TSOVFES flags.

[Functions]

- GWTSDCCs.TE is used to enable timestamp reception for timer s. If enabled, all timestamps received from TX Timestamp Capture interfaces [RMAC] for timer s will be stored in the timestamp RAM and later given to the AXI master interface.
- GWTSNM register is used to monitor the current number of timestamps present in the timestamp RAM.
- GWTSMNM register is used to monitor the maximum number of timestamps that has been held in the timestamp RAM since the previous reset or the previous time corresponding register has been read.
- GWEIS0.TSOVFES flag is used to flag a timestamp loss because of Timestamp RAM overflow.

38.3.4.2 AXI Master Interface

The AXI master interface handles the timestamp/AXI descriptor exchange with the CPU. It will receive the timestamps and the information related to the timestamps and store them in the CPU RAM. To do so, it uses GWTDCACs0, GWTDCACs1 registers and GWTSDCCs.DCS bit and it can be monitored using GWTSDIS.TSDISs and GWEIS0.TDFESs flags.

[Functions]

- GWTDCACs0 and GWTDCACs1 registers form the AXI address {GWTDCACs0.TSCCAU[7:0], GWTDCACs1.TSCCAL[31:0]} with correspond to the current address being processed by the GWCA for TS descriptor queue s. It can be used to set the first address of the descriptor queue during GWCA initialization, move an under-use TS descriptor queue to another location in the CPU RAM or monitor the currently processed descriptor address.
- GWTSDCCs.DCS bit is used to map timer s timestamps to a TS descriptor queue number.
- GWTSDIS.TSDISs flag is used to flag some timestamps are ready in the CPU RAM.
- GWEIS0.TDFES flag is used to flag a timestamp loss because the timestamp descriptor queue is full.

Timestamp reception can be done using linear or cyclic descriptor queues (refer to section 38.3.1.5, AXI Descriptor Queue Format). Timestamp reception process descriptor utilization is described in Figure 38.50. Timestamp reception process descriptor format is described in Figure 38.51 and Table 38.33.

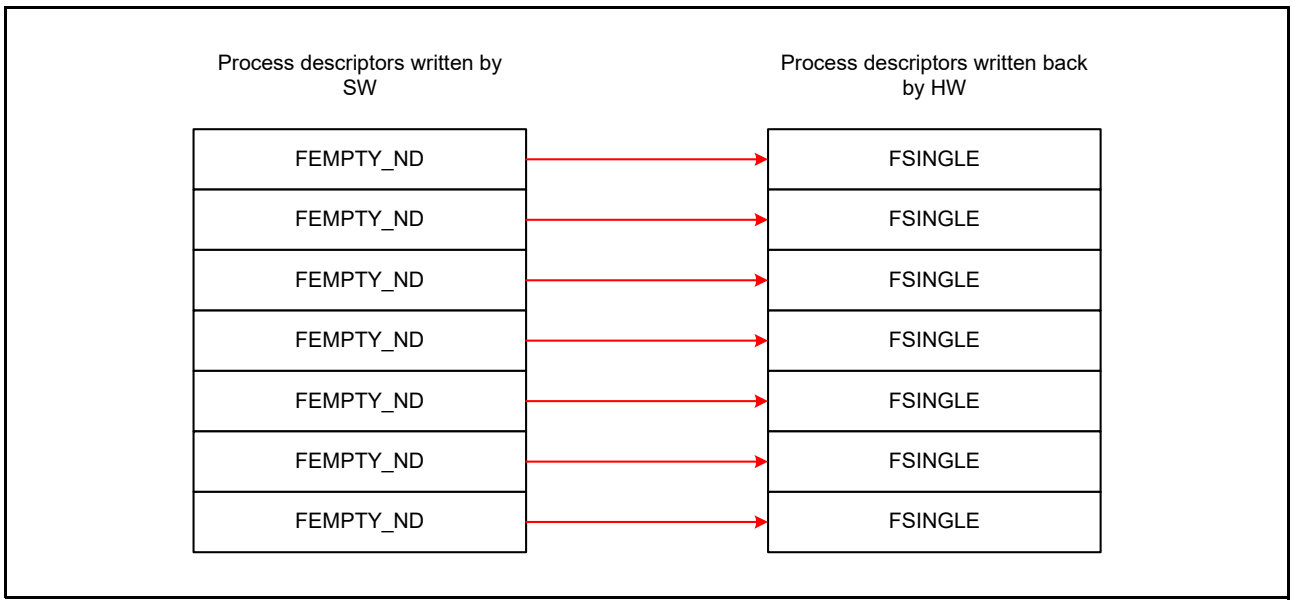


Figure 38.50 TS Reception Process Descriptor Utilization

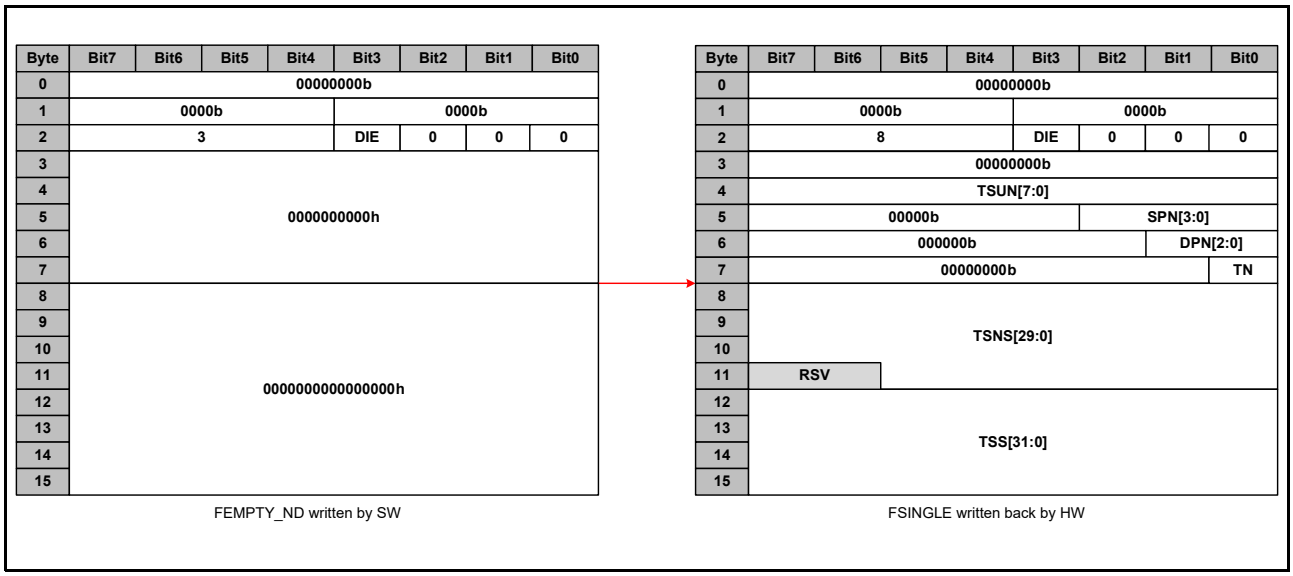


Figure 38.51 TS Reception Process Descriptor Utilization Format (only Timestamp Descriptor, Refer to (1) General Formats)

Table 38.33 TS Reception Process Descriptor Utilization Field Description

Field name	Field description written by SW	Field description after HW write-back
DS	000h	000h
INFO0	0000b	0000b
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	DIE	DIE (Value is copied from read descriptor).
DT	3 for FEMPTY_ND	8 for FSINGLE
PTR	0000000000h	Used as timestamp related information field Refer to Table 38.34
INFO1	NA: Info1 is never present for TS reception	NA: Info1 is never present for TS reception
TS	0000000000000000h	Refer to Table 38.35

Table 38.34 TS Reception Process Descriptor PTR Field Explanation

Field name	Bit width	Description
TSUN	8	Timestamp unique number [RMAC]
SPN	2	Port number from which the timestamp corresponding frame entered the switch [MFWD]
DPN	1	Port number by which the timestamp has been taken
TN	1	Timer Number

Table 38.35 TS Reception Process Descriptor TS Field Explanation

Field name	Bit width	Description
TSNS	30	Timestamp nanosecond [GPTP]
TSS	32	Timestamp second [GPTP] Restrictions: • HW: The 16-upper bits of the gPTP [GPTP] second part are truncated.

38.3.5 gPTP Triggered Transmission

It is possible to use gPTP timer to trigger a periodic transmission start using time-controlled data transmission (refer to Table 38.13). To do so, CYCLIC_COMP[i][c] functional interface is linked to INTER_TX[8+c] (refer to Figure 38.2).

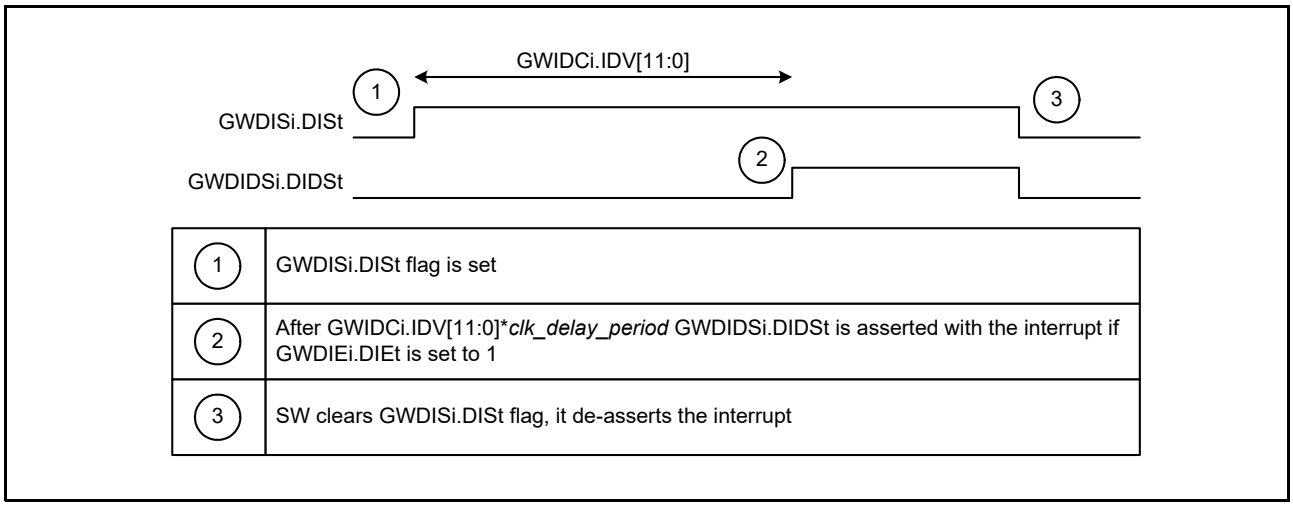
For CYC_COMP_N parameter explanation, refer to gPTP specification [GPTP].

38.3.6 Interrupt Delay Function

Interrupt delay function aims at delaying data interrupts (GWDISi.DISt) to accumulate data during a certain among of time before requesting software to process the data. As a result, software load is diminished, and data latency is still guaranteed. To operate, interrupt delay function uses GWIDPC and GWIDCi registers and can be monitored using GWDIDSi register. Interrupt delay function is described in Figure 38.52.

[Functions]

- GWIDPC register sets an internal clock of period clk_delay_period used as base clock for interrupt delay function.
- GWIDCi register sets the delay time for descriptor queue i in clk_delay_period.
- GWDIDSi.DIDSt flag is the delayed GWDISi.DISt flag and triggers the interrupt gwc_gwdis_int[t].
- For interrupt handling, software can decide to read GWDIDSi registers and to process only interrupts that has set until their corresponding delay time is archived (allows to be sure a certain amount of data came before processing) or can decide to read GWDISi to process all the interrupts even the ones which did not reached their delay time (allows to be sure a certain amount of data came before processing at least for one queue can diminish the interrupt load).



1	GWDISi.DISt flag is set
2	After GWIDCi.IDV[11:0]**clk_delay_period* GWDIDSi.DIDSt is asserted with the interrupt if GWDIEi.DIEt is set to 1
3	SW clears GWDISi.DISt flag, it de-asserts the interrupt

Figure 38.52 Interrupt Delay Function

[Restrictions]

- HW: The maximum delay achievable by the interrupt delay function is clk_period*4194304. For example, it will be around 10ms for a clock clk with a frequency of 400MHz.

38.3.7 Operation Modes

Table 38.36 describes GWCA operation modes.

Table 38.36 GWCA Operation Modes

Operation mode	GWMS.OPS value	Description
DISABLE	1	• No transaction is ongoing. But there can be tolerance (left behind) of transactions*from GWCA when the agent has moved from OPERATION to DISABLE. (*: max 8 read transactions or max 8 write transactions) • Only status registers are accessible for writing when the agent clock is enabled.
RESET	0	• No transaction is ongoing. • An internal Reset is asserted to reset GWCA logic with status registers (When a register is reset in RESET mode, it is mentioned in its description). • No register is accessible for writing. • RAM values are held.
CONFIG	2	• No transaction is ongoing. • Static and some dynamic registers are accessible for writing.
OPERATION	3	• Transactions are ongoing. • Dynamic registers are accessible for writing.

38.3.7.1 Operation Mode Transitions

Figure 38.53 shows the operating mode transitions.
A mode transition can be trigger by:

- Hardware reset.
- Software reset.
- Configuration of GWMC.OPC[1:0]. In that case, mode transition will be confirmed by GWMS.OPS[1:0].

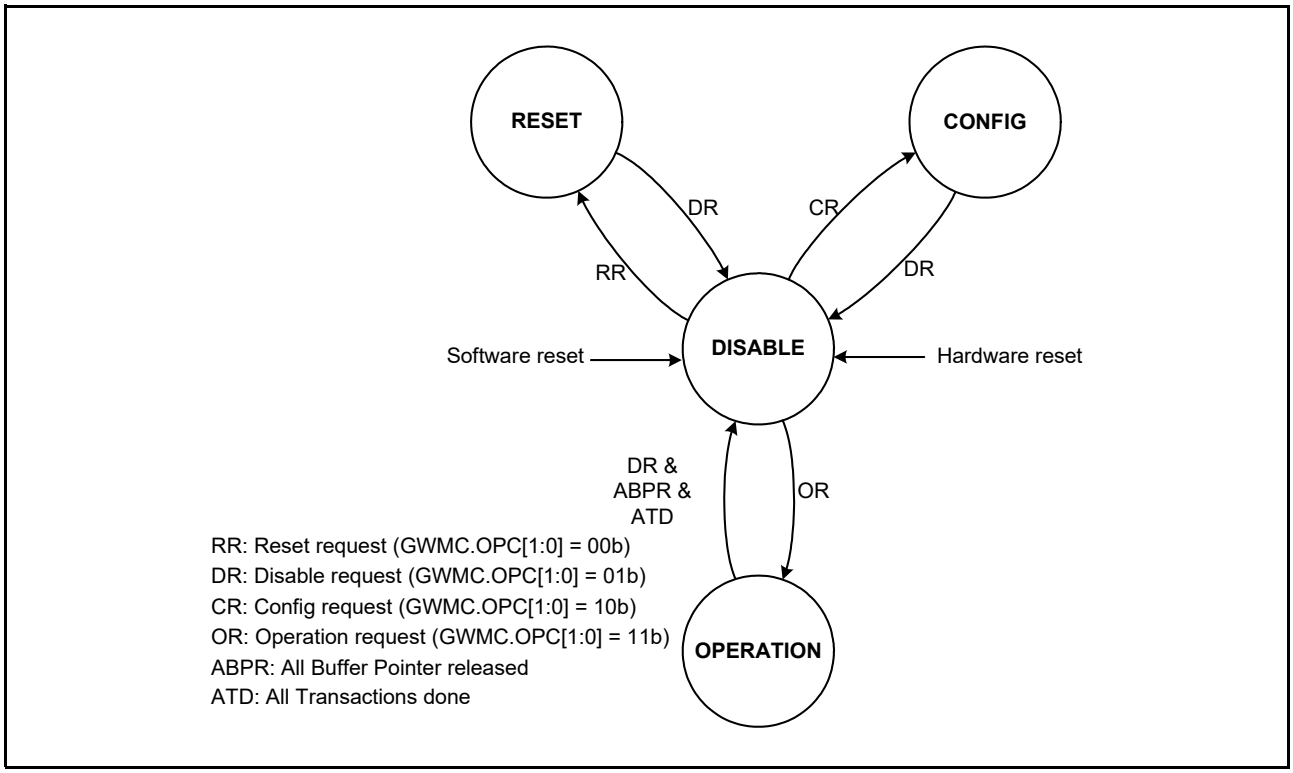


Figure 38.53 Operating Mode Transitions

Complementary information on transition conditions
ABPR:

- All the buffer pointers contained in the GWCA are release to the forwarding engine [MFWD].

ATD:

- All the descriptors already received are processed and corresponding frames are sent to CPU and AXI descriptors are written back.

All ongoing transmit frames are fully sent to the forwarding engine [MFWD] and AXI descriptors are written back.

[Restriction]

- Software shall only trigger transitions shown to Figure 38.53.
  Exp: Do not directory transition from OPERATION to CONFIG.

38.3.8 Software Flows

38.3.8.1 Mode Transition Flow

The mode transition flow is described in Figure 38.54.

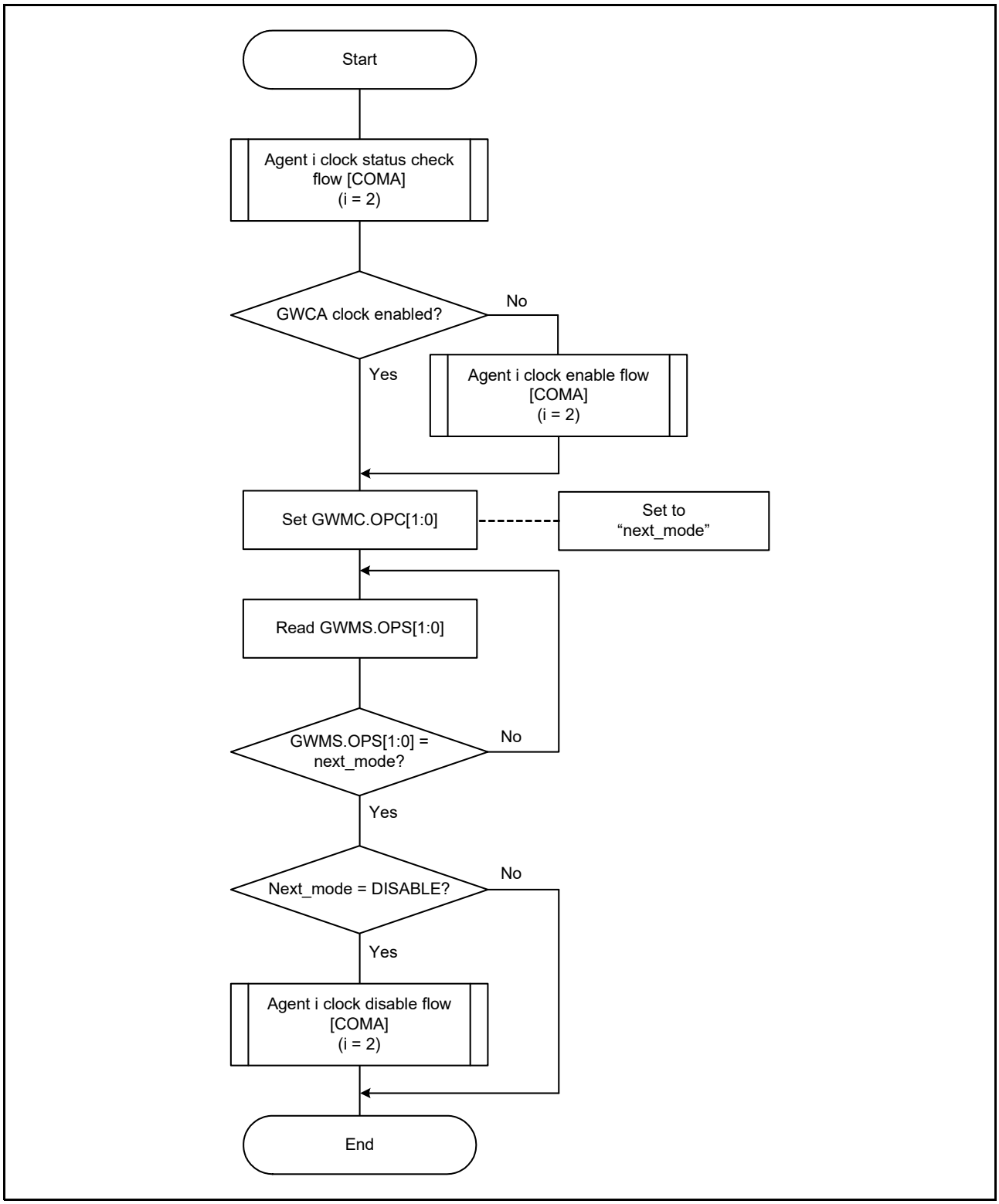


Figure 38.54 Mode Transition Flow

Note: If SW already know that the clock is enabled, clock status check step can be skipped.

Note: If SW does not need to disable the agent clock in DISABLE mode (because it will go directly to another mode or because the clock never needs to be disabled), clock disable step can be skipped.

38.3.8.2 Reset Flow

The reset flow is described in Figure 38.55.

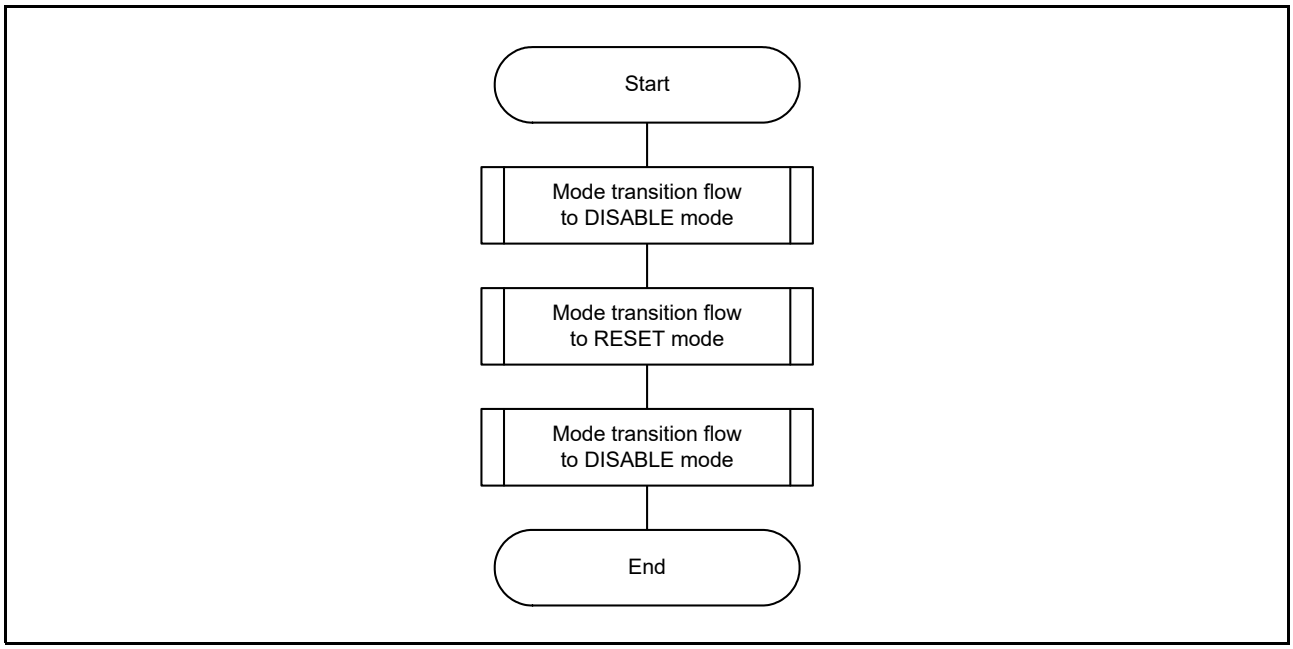


Figure 38.55 Reset Flow

38.3.8.3 Initialization Flow

The initialization flow is described in Figure 38.56.

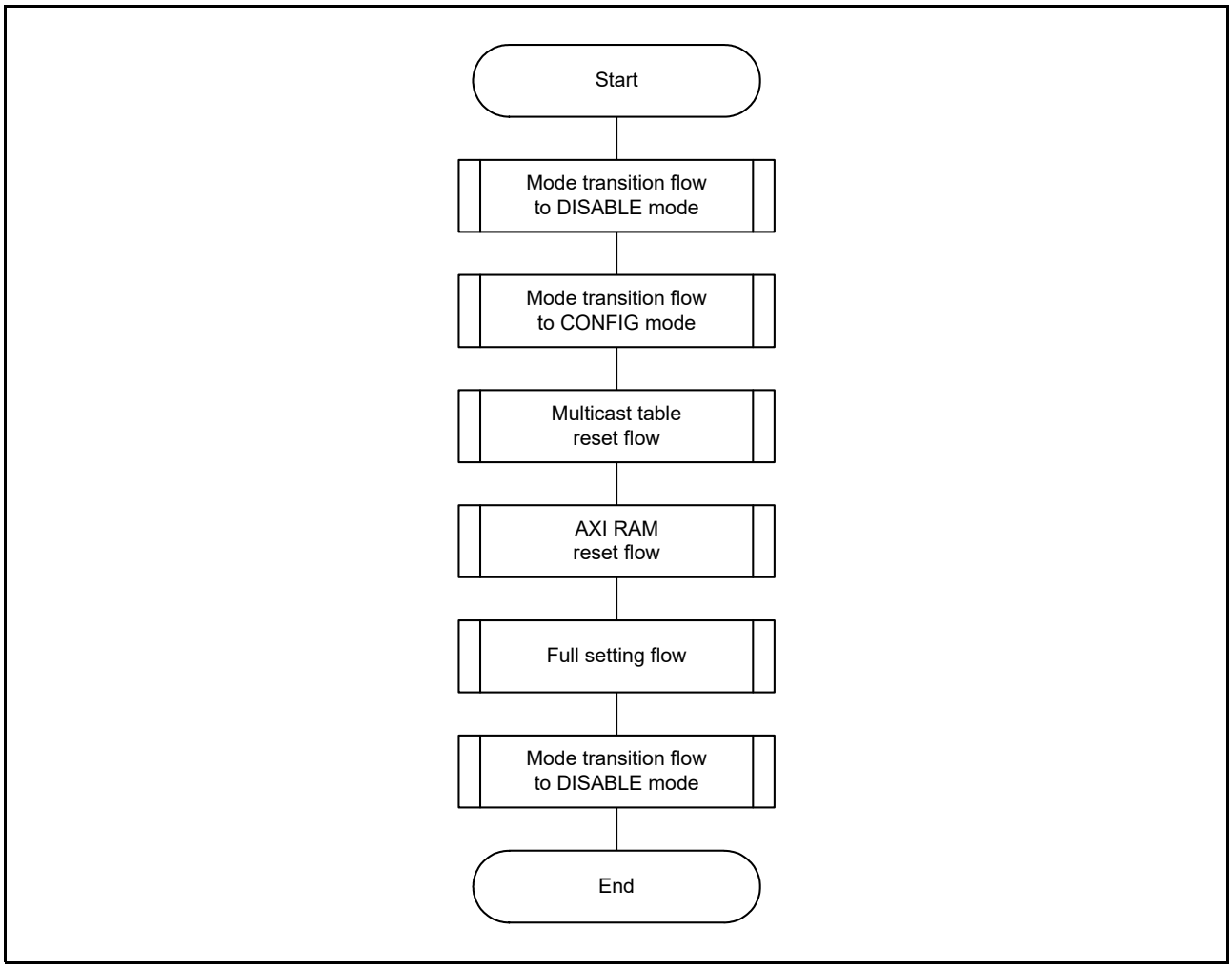


Figure 38.56 Initialization Flow

38.3.8.4 Reinitialization Flow

The reinitialization flow is described in Figure 38.57.

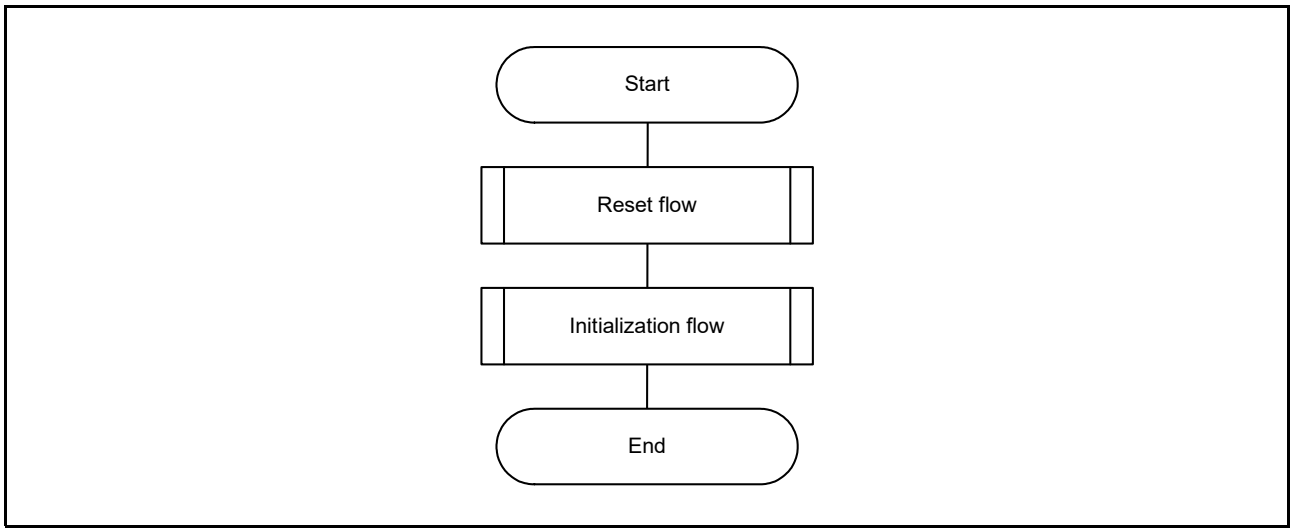


Figure 38.57 Reinitialization Flow

38.3.8.5 Multicast Table Reset Flow

The multicast table reset flow is described in Figure 38.58.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

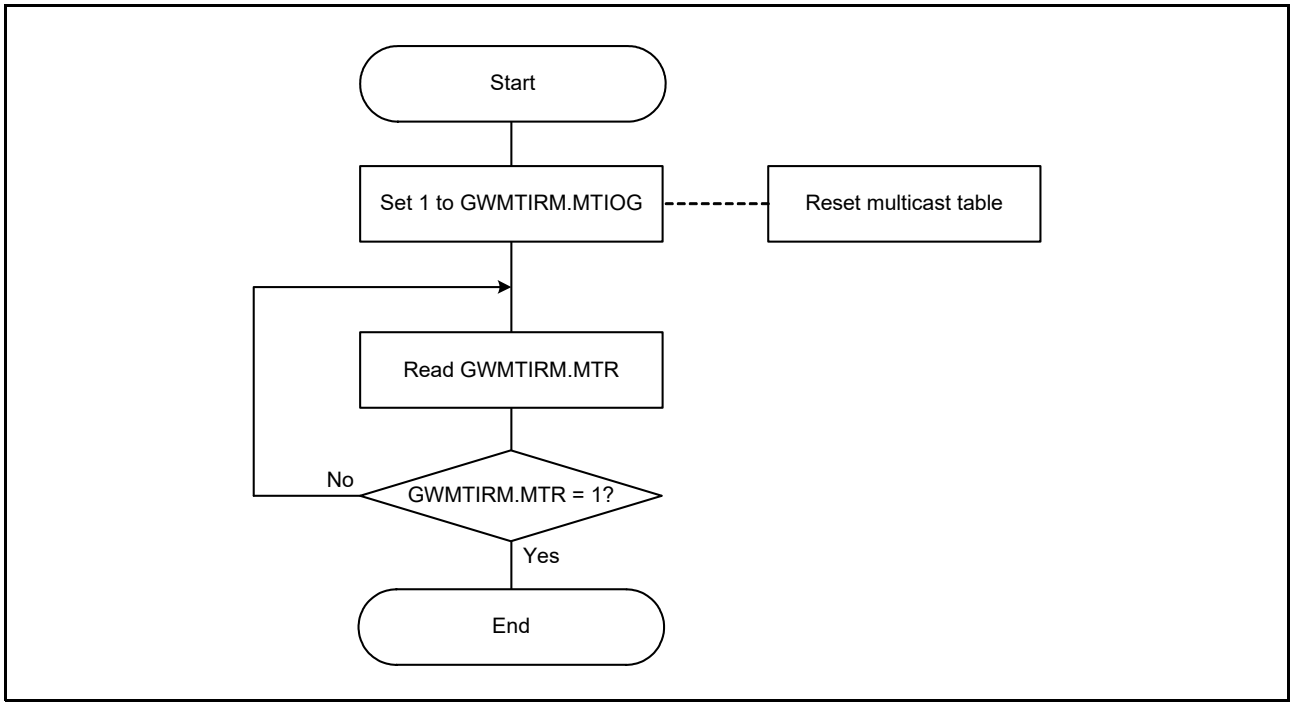


Figure 38.58 Multicast table Reset Flow

38.3.8.6 Multicast Table Setting Flow

The multicast table setting flow is described in Figure 38.59.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

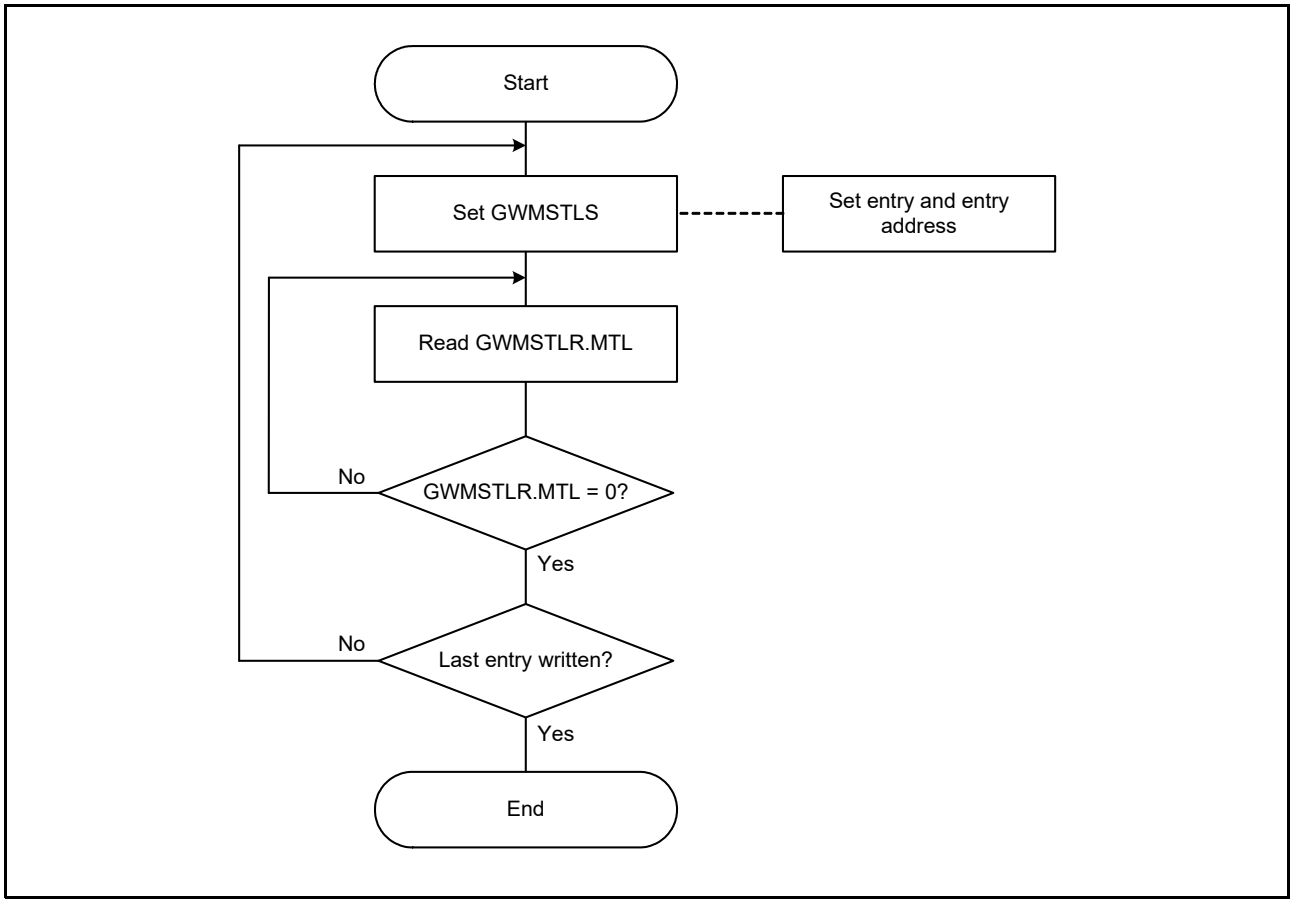


Figure 38.59 Multicast Table Setting Flow

38.3.8.7 Multicast Table Read Flow

The multicast table read flow is described in Figure 38.60.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

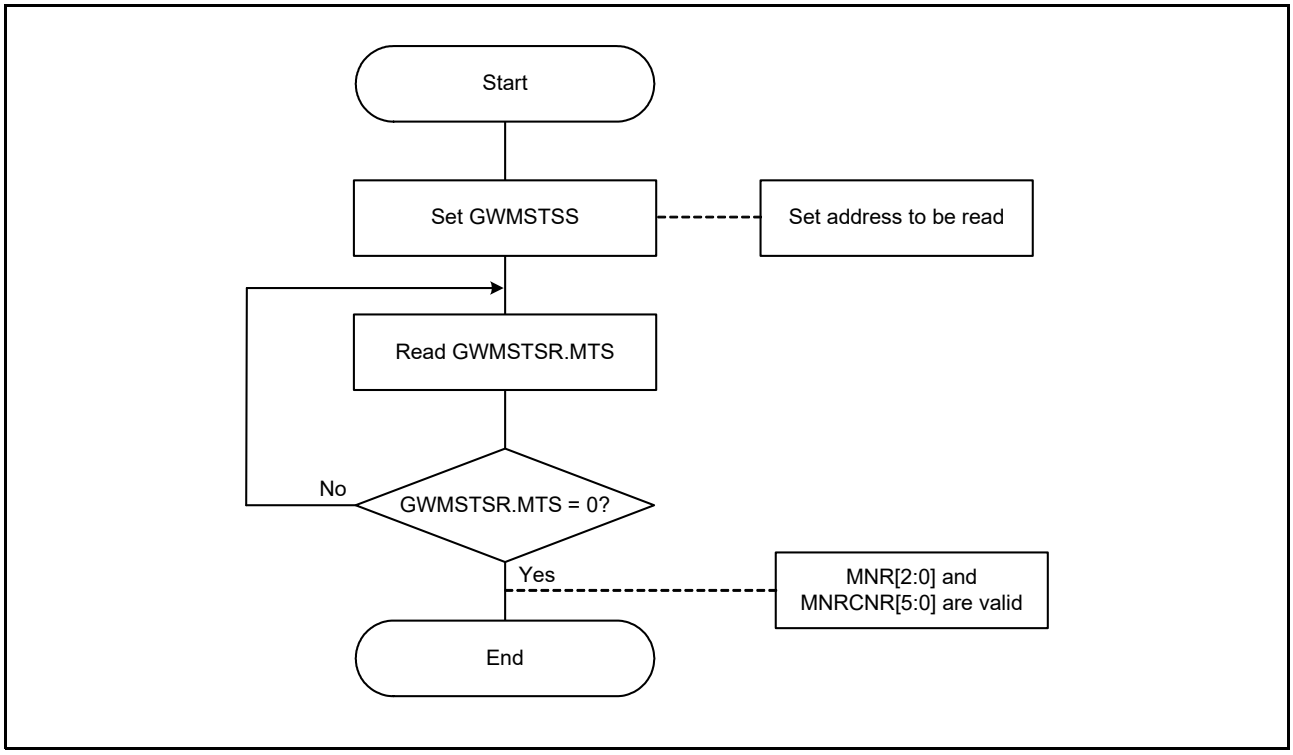


Figure 38.60 Multicast Table Read Flow

38.3.8.8 AXI RAM Reset Flow

The AXI RAM reset flow is described in Figure 38.61.
[Restriction]

- This flow is not usable in RESET, DISABLE and OPERATION modes.

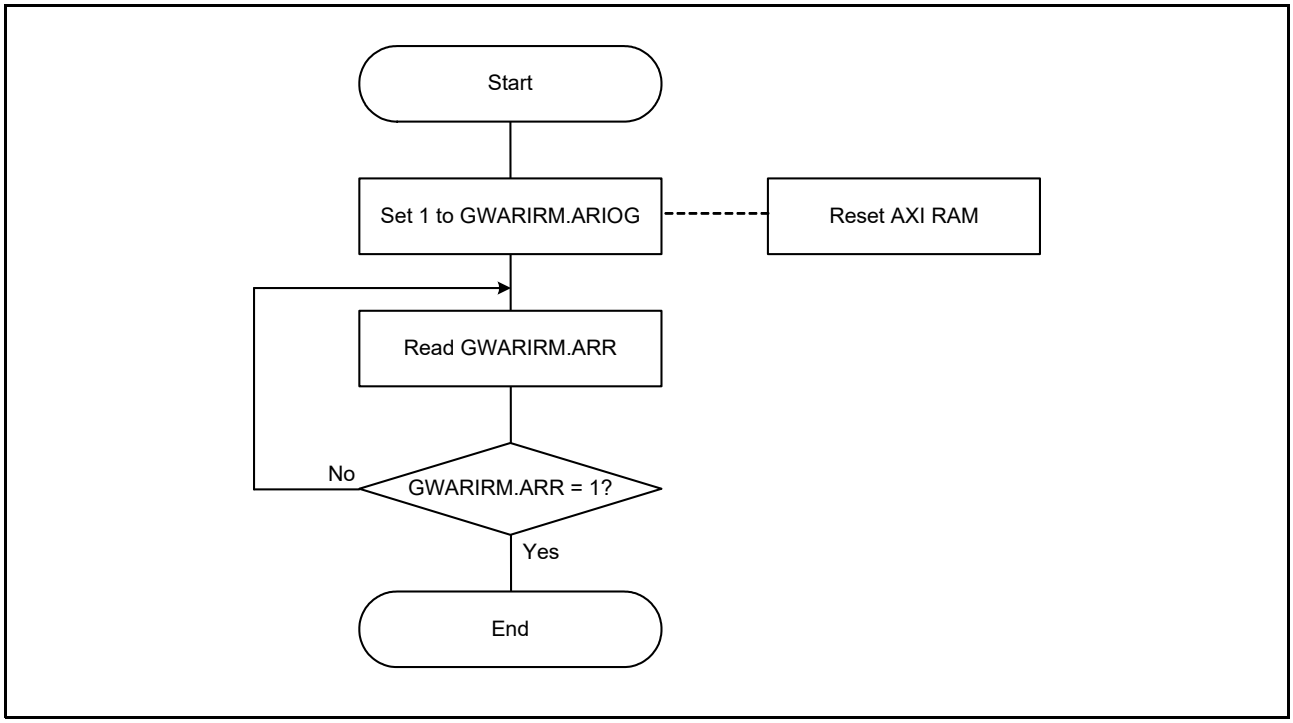


Figure 38.61 AXI RAM Reset Flow

38.3.8.9 AXI RAM Read Flow

The AXI RAM read flow is described in Figure 38.62.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

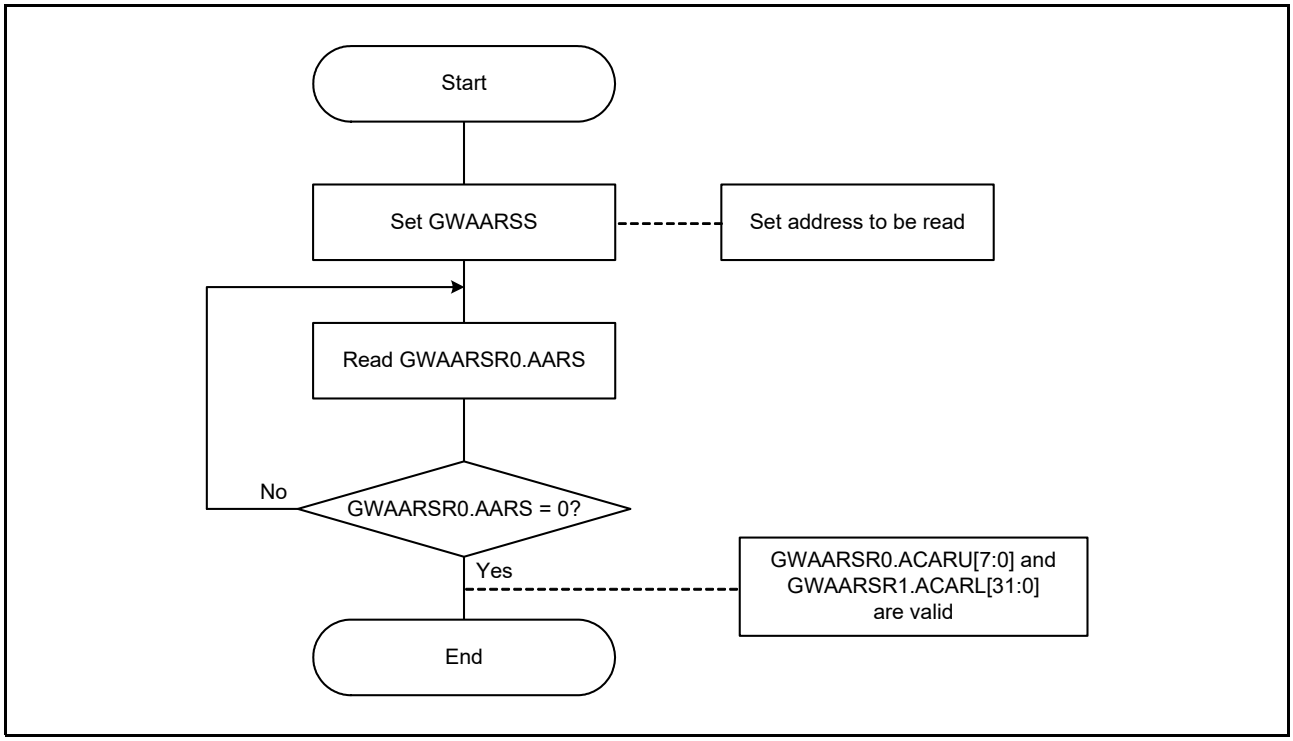


Figure 38.62 AXI RAM Read Flow

38.3.8.10 Global Rate Limiter Setting Flow

The global rate limiter setting flow is described in Figure 38.63.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

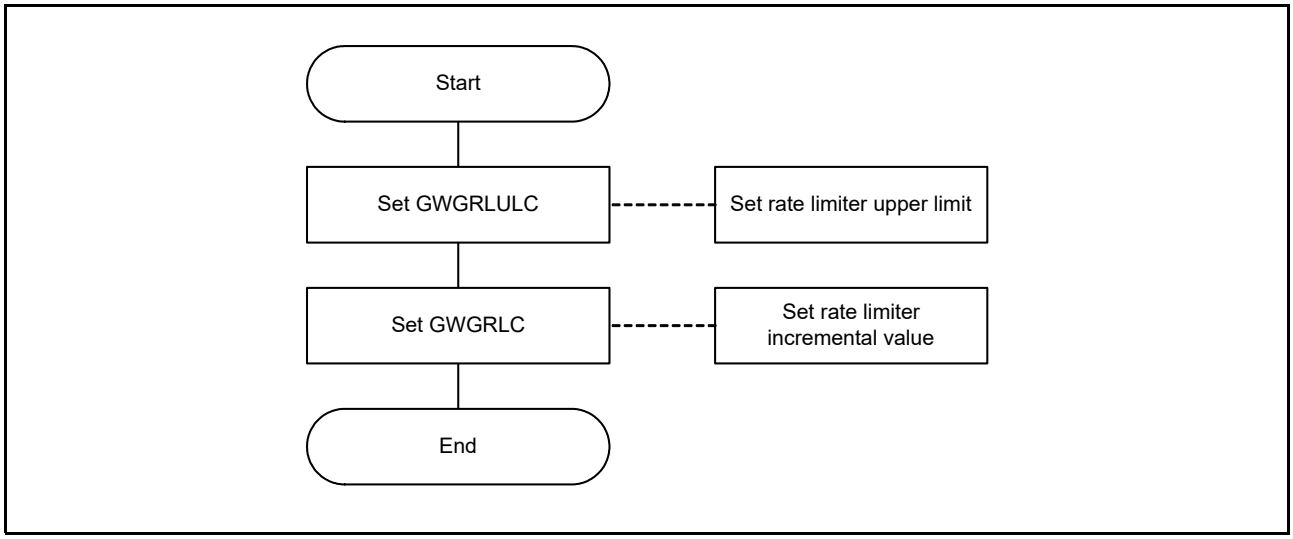


Figure 38.63 Global Rate Limiter Setting Flow

38.3.8.11 Global Rate Limiter Disabling Flow

The global rate limiter disabling flow is described in Figure 38.64.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

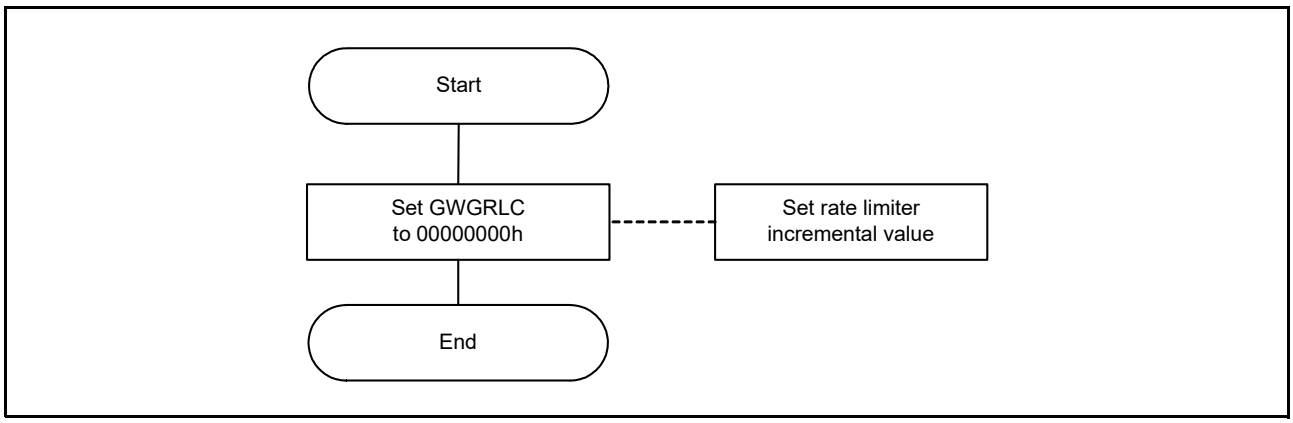


Figure 38.64 Global Rate Limiter Disabling Flow

38.3.8.12 Rate Limiter i Setting Flow

The rate limiter i setting flow is described in Figure 38.65.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

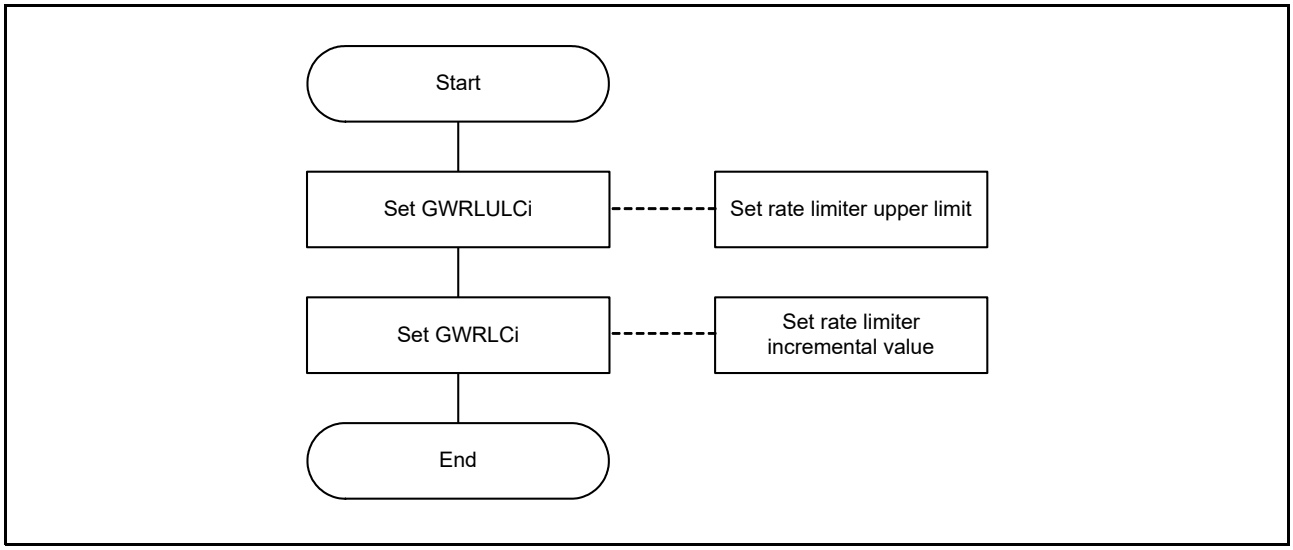


Figure 38.65 Rate Limiter i Setting Flow

38.3.8.13 Rate Limiter i Disabling Flow

The rate limiter i disabling flow is described in Figure 38.66.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

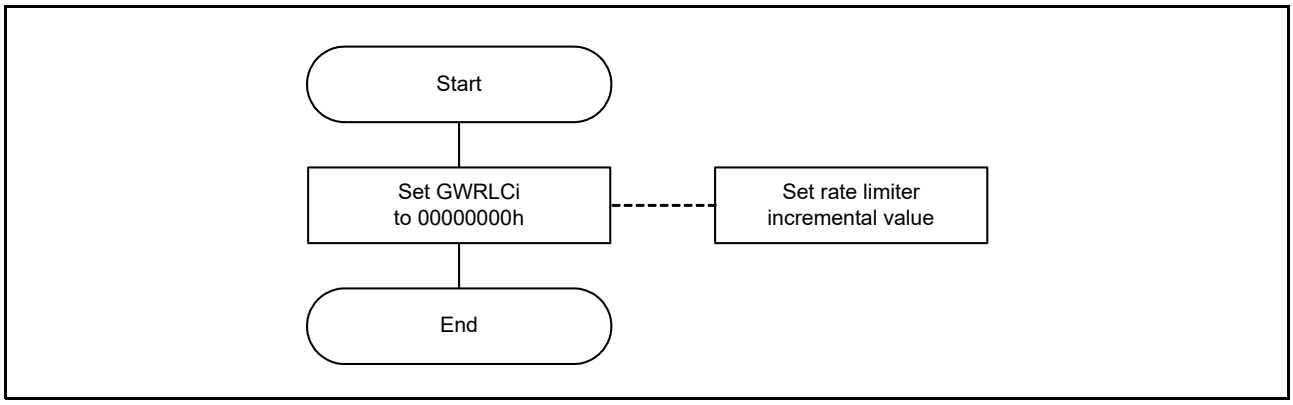


Figure 38.66 Rate Limiter i Disabling Flow

38.3.8.14 AXI Table Read Flow

The AXI table read flow is described in Figure 38.67.
[Restriction]

- This flow is not usable in RESET and DISABLE modes.

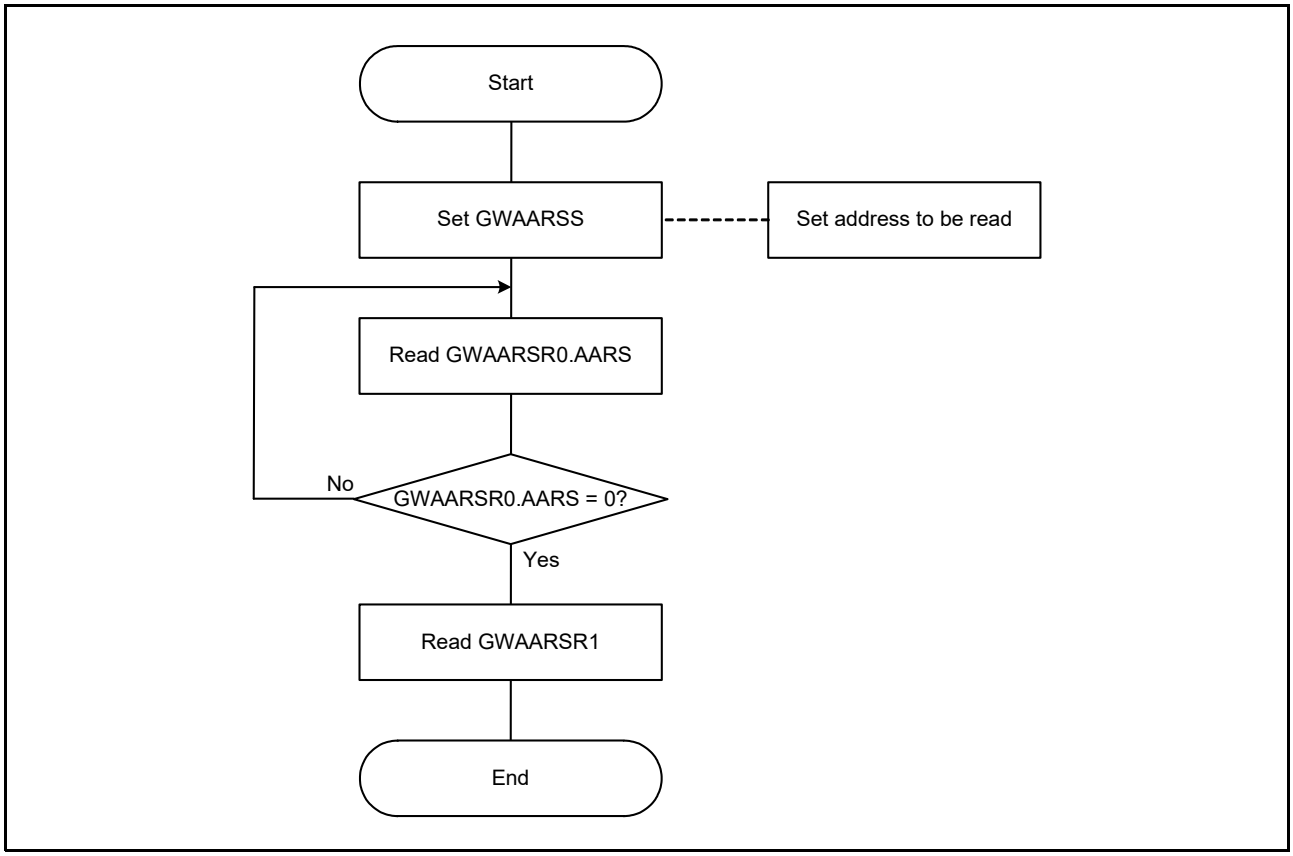


Figure 38.67 AXI Table Read Flow

38.3.8.15 Interrupt Handling Flow

The interrupt handling flow is described in Figure 38.68.
[Restriction]

- This flow is not usable in RESET mode.

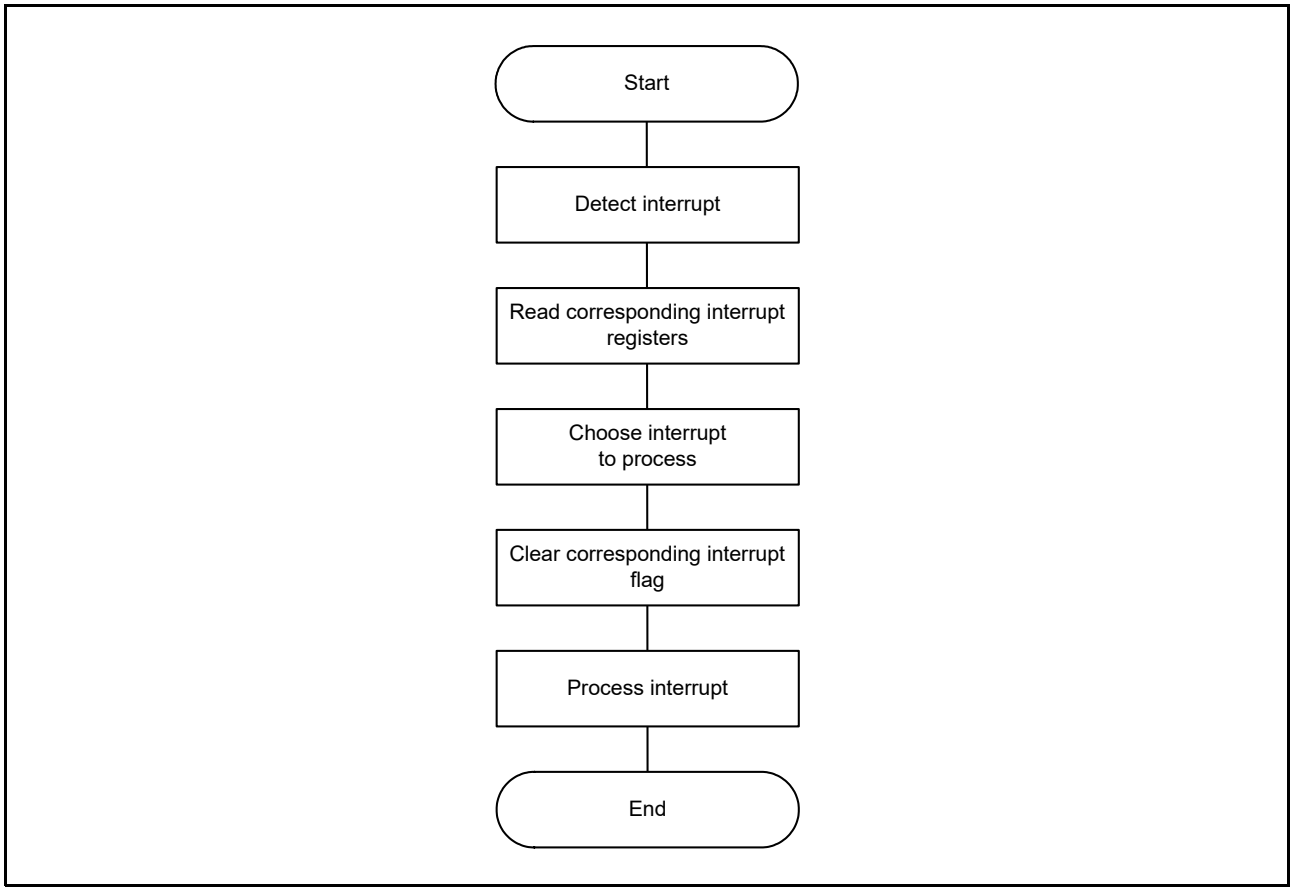


Figure 38.68 Interrupt Handling Flow

38.3.8.16 Called Software Flows

The flows described in this section can only be called from other flows thanks to a “flow link” box (Figure 38.xx) and cannot be used alone.

(1) Full Setting Flow

The full setting flow is described in Figure 38.69.
For LINKFIX table setting refer to (1) LINKFIX Table Initialization.

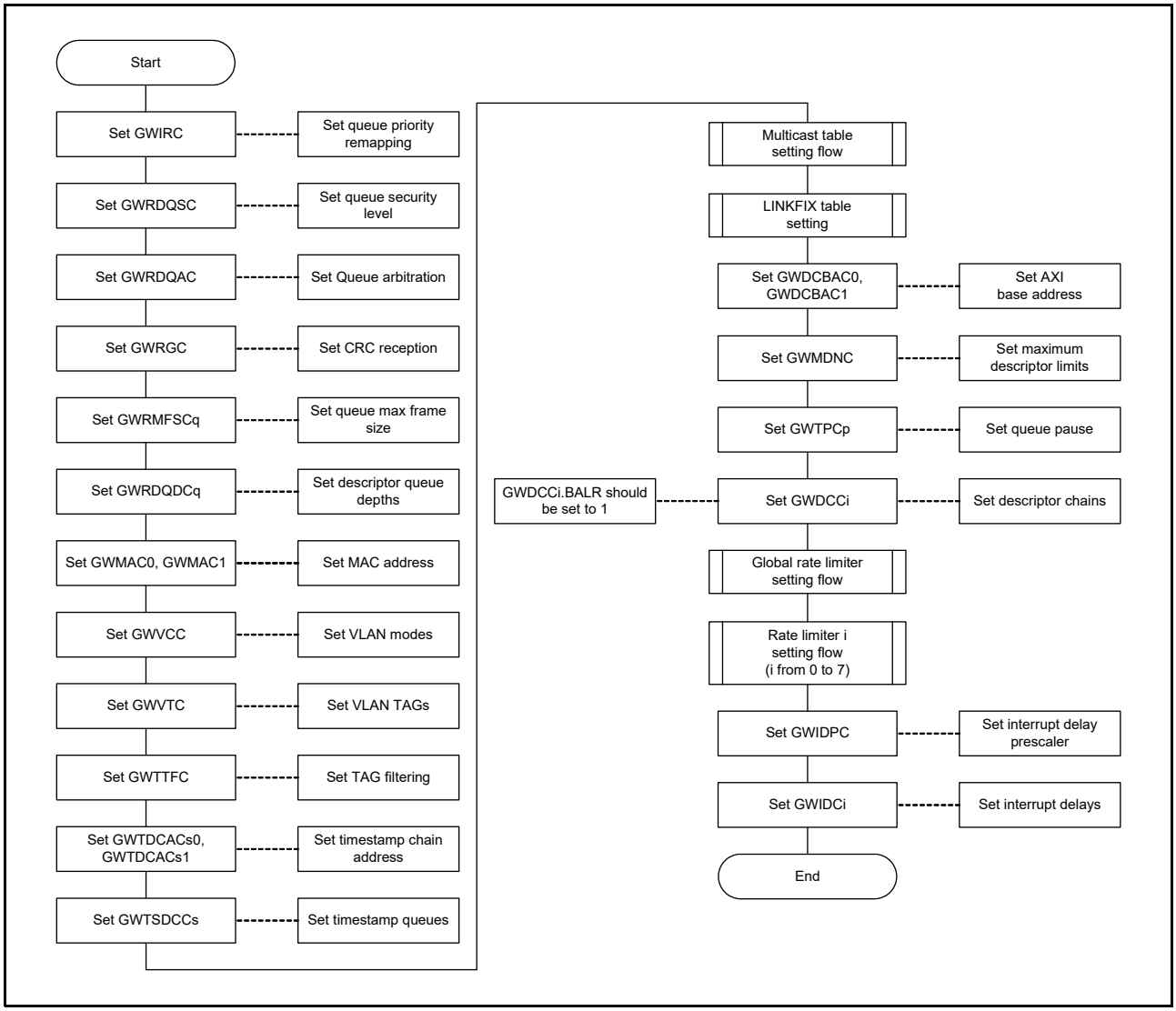


Figure 38.69 Full Setting Flow

(2) AXI Emergency Stop Flow

The AXI emergency stop flow is described in Figure 38.70.

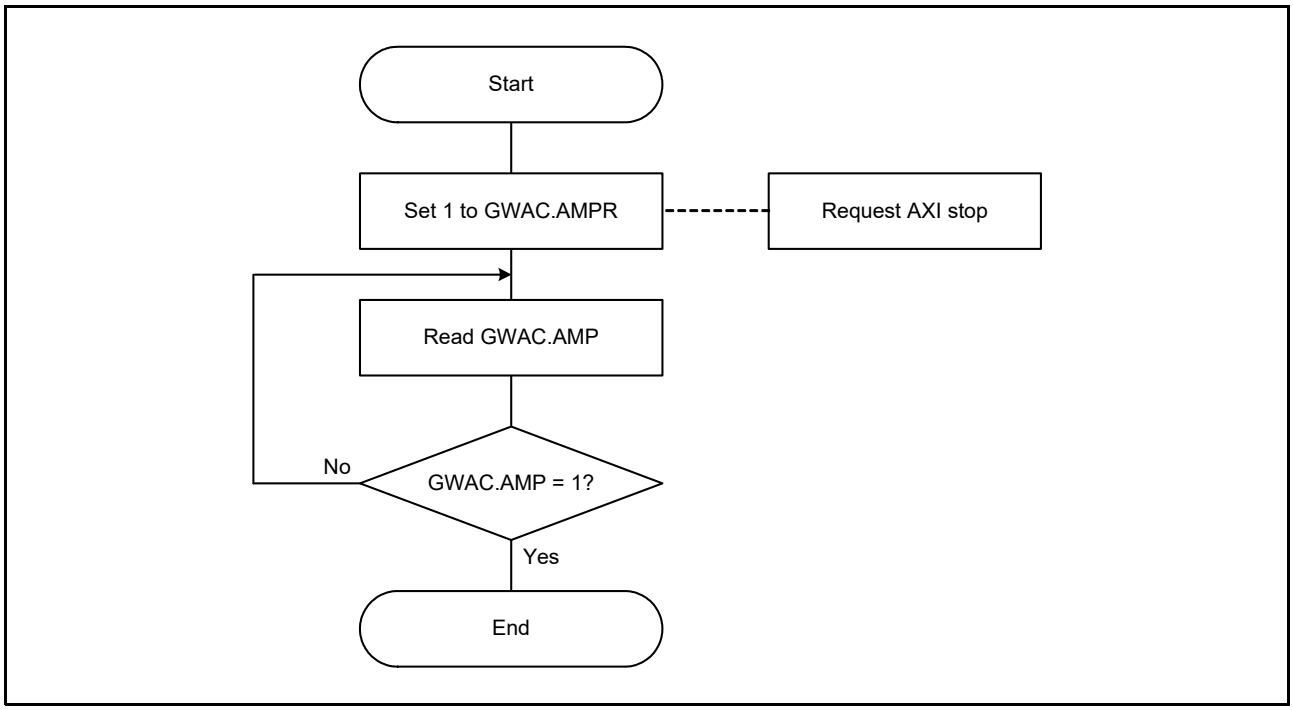


Figure 38.70 AXI Emergency Stop Flow

38.3.8.17 Register Writable Without Software Flow

This section describes registers that have not been described so far. These registers can be changed dynamically. (However, it is necessary that the initial settings such as the clock enabling have been completed.)

Note: When a frame has reached GWCA and before the frame has been written into URAM by AXI BMI if GWDCCi register configurations of target chain is reconfigured by the user then the frame will be handled using the GWDCCi register settings before reconfiguration and not using the reconfigured GWDCCi register settings.

39. Ethernet Generic PTP Timer (GPTP)

39.1 Overview

The gPTP timer enables accurate synchronization of the clock in the control system. The pulse output timer can generate arbitrary pulses based on the gPTP timer.

39.1.1 Features

The gPTP features are described in Table 39.1. The pulse output timer features are described in Table 39.2.

Table 39.1 gPTP Feature List

Function		Details
Overall function	Timer	• gPTP and AVTP timer
Data provision	—	• gPTP timer Data width: 78bit ([77:30] = second, [29:0] = nanosecond) • AVTP timer Data width: 64bit/32bit
Functions	Timer control	• gPTP timer increment value control • gPTP timer offset control • Generation of gPTP and AVTP timer
	PPS	• PPS signal generation
	Media clock	• Media clock capture • Media clock recovery
	Cyclic compare	• cyclic comparison to emit pulses

Table 39.2 Pulse Output Timer Feature List

Function		Details
Overall function	Pulse Generator	• Pulse generation timer, pulsed output and interrupt output.
Data provision	—	• Pulsed output channel: 2ch The pulse period: 80bit ([79:32] = second and [31:0] = ns) Pulse width: 20bit ([19:0] = Counter) • Interrupt output channel: 2ch
Functions	PTP pulse generation	• Uses the reference time of the gPTP timer to generate pulse outputs of any period and any pulse width.
	Interrupt pulse output	• The rising edge of the pulse output is detected and the interrupt pulse is output.

39.1.2 Block Diagram

Figure 39.1 shows gPTP timer the block diagram.

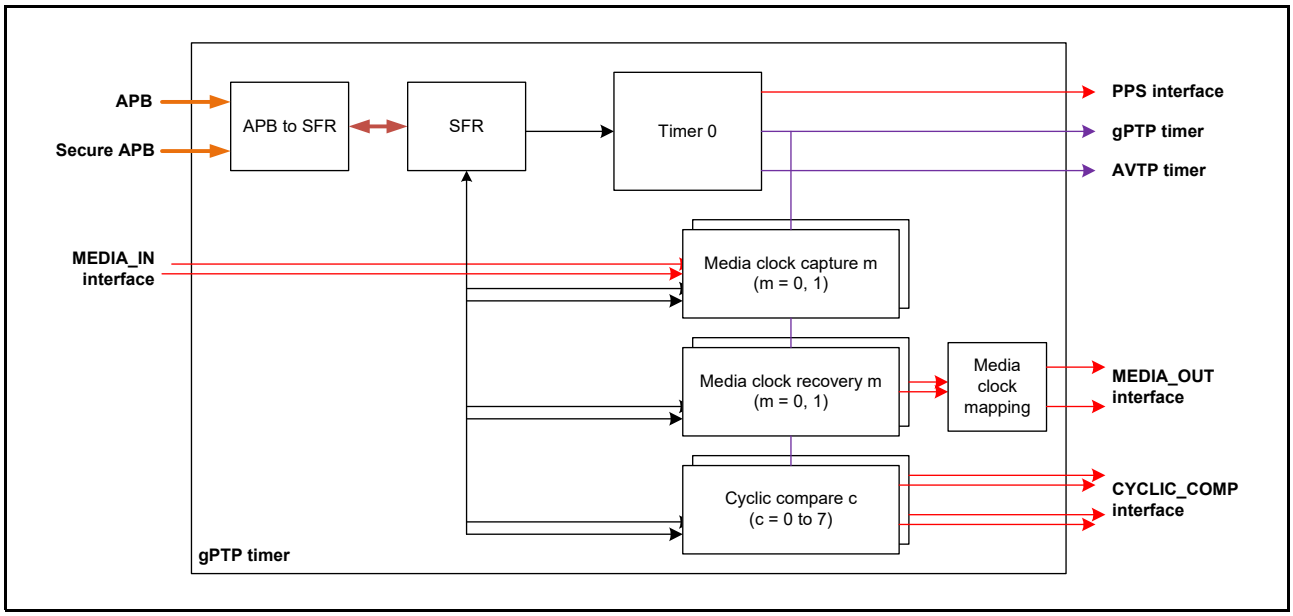


Figure 39.1 gPTP Timer Block Diagram

Table 39.3 gPTP Timer Functional Blocks

Block name	Function
APB to SFR	Handle SFR accesses
SFR	gPTP SFRs
Timer	Generate gPTP an AVTP timers
Media clock capture	Capture input media clocks
Media clock recovery	Recover output media clocks
Cyclic compare	Generate a cyclic pulse

Figure 39.2 shows a block diagram of the pulse output timer.

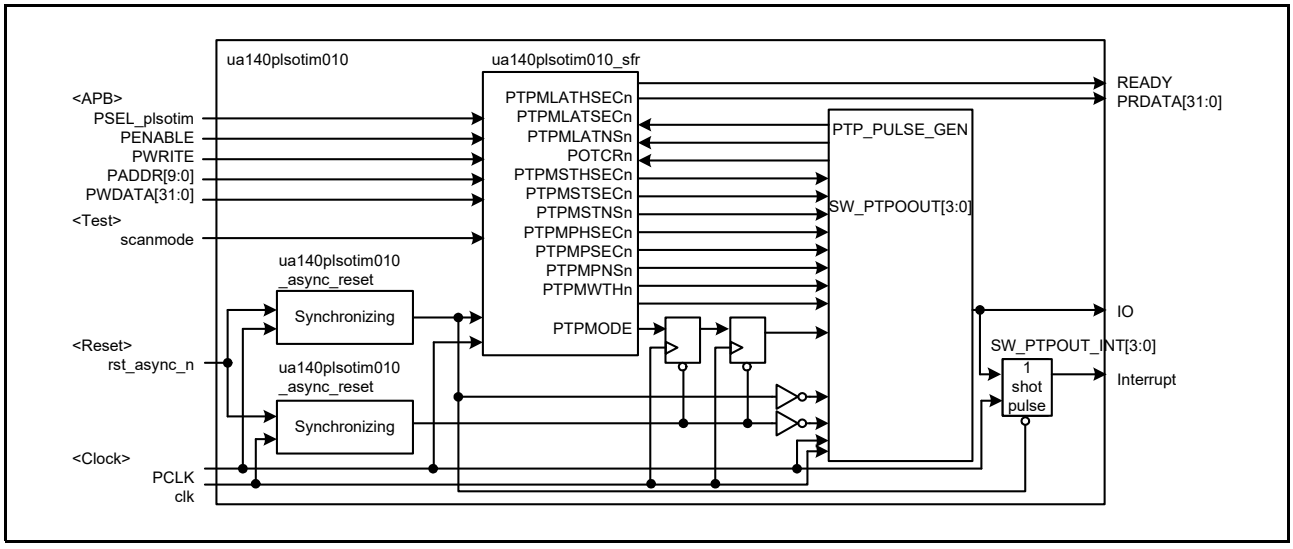


Figure 39.2 Pulse Output Timer Block Diagram (n = 0, 1)

Table 39.4 Pulse Output Timer Functional Blocks

Block name	Function
Reset rise async to ESWCLK	Synchronize the rising edge of an asynchronous reset to ESWCLK.
Reset rise async to PCLKA	Synchronize the rising edge of an asynchronous reset to PCLKA.
SFR	SFR access processing from the pulse output timer registers and APB.
PTP_PULSE_GEN	Pulse generation and pulsed output.
Interrupt output	Interruption generation and interrupt output.

39.2 Register Descriptions

39.2.1 Version Register

39.2.1.1 IP Version Register (PTPIPV)

Address(es): GPTP.PTPIPV 874E 0000h, GPTP_NS.PTPIPV 074E 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IPV[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IPV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	IPV[31:0]	IP Version		R

39.2.2 Timer Configuration Registers

39.2.2.1 Timer Enable Configuration Register (PTPTMEC)

Address(es): GPTP.PTPTMEC 874E 0010h, GPTP_NS.PTPTMEC 074E 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TE0	Timer 0 Enable	0: Timer 0 disabled 1: Timer 0 enabled	R/W
b31 to b1	—	Reserved	These bits are read as 0.	R

TE0 Bit (Timer 0 Enable)

When timer 0 is disabled, all counters are fixed to 0.

[Setting Condition]

- SW: Writing 1 to the TE0 bit will set it.

[Clearing Conditions]

- SW: Writing 1 to the PTPTMDC.TDn bit will clear the TE0 bit.

39.2.2.2 Timer Disable Configuration Register (PTPTMDC)

Address(es): GPTP.PTPTMDC 874E 0014h, GPTP_NS.PTPTMDC 074E 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TD0	Timer 0 Disable	Writing 1 to this bit will clear the PTPTMEC.TE0 bit.	R/W
b31 to b1	—	Reserved	These bits are read as 0.	R

39.2.2.3 Timer 0 Increment Value Configuration Register (PTPTIVC0)

Address(es): GPTP.PTPTIVC0 874E 0020h, GPTP_NS.PTPTIVC0 074E 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TIV[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TIV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TIV[31:0]	Timer Increment Value	Timer increment value in ns per the ESWCLK.	R/W

TIV[31:0] Bits (Timer Increment Value)

These bits represent a fixed-point number with the decimal point positioned between b27 and b26. The TIV[31:27] bits indicate nanoseconds, while the TIV[26:0] bits indicates sub-nanoseconds.

[Example]

- When the ESWCLK frequency is 50 MHz, 100 MHz, 200 MHz, 320 MHz, or 400 MHz, set this register to a value around A000 0000h (20.000 ns), 5000 0000h (10.000 ns), 2800 0000h (5.000 ns), 1900 0000h (3.125 ns), or 1400 0000h (2.500 ns), respectively.

39.2.2.4 Timer 0 Offset Value Configuration Register L (PTPTOVC0L)

Address(es): GPTP.PTPTOVC0L 874E 0030h, GPTP_NS.PTPTOVC0L 074E 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TOVL[29:16]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOVL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	TOVL[29:0]	Timer Offset Value Lower Part	Timer offset value in gPTP format.	R/W
b31, b30	—	Reserved	These bits are read as 0.	R

TOVL[29:0] Bits (Timer Offset Value Lower Part)

The PTPTOVC0L.TOVL[29:0] bits represent a part of Nanoseconds.

Writing in this register will load the new offset to timer 0.

{The PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, 00b, the PTPTOVC0L.TOVL[29:0] bits} is timer 0 offset.

[Restriction]

- SW: Only value from 0000 0000h to 3B9A C9FFh can be set to these bits.

39.2.2.5 Timer 0 Offset Value Configuration Register M (PTPTOVC0M)

Address(es): GPTP.PTPTOVC0M 874E 0034h, GPTP_NS.PTPTOVC0M 074E 0034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOVM[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOVM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TOVM[31:0]	Timer Offset Value Middle Part	Timer offset value in gPTP format.	R/W

TOVM[31:0] Bits (Timer Offset Value Middle Part)

The PTPTOVC0M.TOVM[31:0] bits represent lower 32 bits of Seconds.

{The PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, 00b, the PTPTOVC0L.TOVL[29:0] bits} is timer 0 offset.

[Update condition]

- SW: Writing the PTPTOVC0L register will update these bits to the latest value that has been written to it.

39.2.2.6 Timer 0 Offset Value Configuration Register U (PTPTOVC0U)

Address(es): GPTP.PTPTOVC0U 874E 0038h, GPTP_NS.PTPTOVC0U 074E 0038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TOVU[15:0]	Timer Offset Value Upper Part	Timer offset value in gPTP format.	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

TOVU[15:0] Bits (Timer Offset Value Upper Part)

The PTPTOVC0U.TOVU[15:0] bits represent upper 16 bits of Seconds.

{The PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, 00b, the PTPTOVC0L.TOVL[29:0] bits} is timer 0 offset.

[Update condition]

- SW: Writing the PTPTOVC0L register will update these bits to the latest value that has been written to it.

39.2.3 Timer Monitoring Registers

39.2.3.1 AVTP Timer 0 Monitoring Register L (PTPAVTPTM0L)

Address(es): GPTP.PTPAVTPTM0L 874E 0040h, GPTP_NS.PTPAVTPTM0L 074E 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AVTPL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AVTPL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	AVTPL[31:0]	AVTP Timer Value Lower Part	AVTP time is in nanoseconds and derive from gPTP timer 0.	R

AVTPL[31:0] Bits (AVTP Timer Value Lower Part)

The PTPAVTPTM0L.AVTPL[31:0] bits represent either 32-bit AVTP timer or lower 32 bits of 64-bit AVTP timer. These bits are always equal to 32-bit AVTP timer or lower 32 bits of 64-bit AVTP timer.

39.2.3.2 AVTP Timer 0 Monitoring Register U (PTPAVTPTM0U)

Address(es): GPTP.PTPAVTPTM0U 874E 0044h, GPTP_NS.PTPAVTPTM0U 074E 0044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AVTPU[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AVTPU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	AVTPU[31:0]	AVTP Timer Value Upper Part	AVTP time is in nanoseconds and derive from gPTP timer 0.	R

AVTPU[31:0] Bits (AVTP Timer Value Upper Part)

The PTPAVTPTM0U.AVTPU[31:0] bits represent upper 32 bits of 64-bit AVTP timer.
[Update condition]

- HW: Reading the PTPAVTPTM0L register update these bits to upper 32 bits of 64-bit AVTP timer.

39.2.3.3 gPTP Timer 0 Monitoring Register L (PTPGPTPTM0L)

Address(es): GPTP.PTPGPTPTM0L 874E 0050h, GPTP_NS.PTPGPTPTM0L 074E 0050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	GPTPL[29:16]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GPTPL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	GPTPL[29:0]	gPTP Timer Value Lower Part	These bits represent a part of Nanoseconds of gPTP timer.	R
b31, b30	—	Reserved	These bits are read as 0.	R

GPTPL[29:0] Bits (gPTP Timer Value Lower Part)

This register is always equal to a part of Nanoseconds of gPTP timer.

39.2.3.4 gPTP Timer 0 Monitoring Register M (PTPGPTPTM0M)

Address(es): GPTP.PTPGPTPTM0M 874E 0054h, GPTP_NS.PTPGPTPTM0M 074E 0054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	GPTPM[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GPTPM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	GPTPM[31:0]	gPTP Timer Value Middle Part	These bits represent lower 32 bits of Seconds of gPTP timer.	R

GPTPM[31:0] Bits (gPTP Timer Value Middle Part)

[Update condition]

- HW: Reading the PTPGPTPTM0L register update these bits to lower 32 bits of Seconds of gPTP timer.

39.2.3.5 gPTP Timer 0 Monitoring Register U (PTPGPTPTM0U)

Address(es): GPTP.PTPGPTPTM0U 874E 0058h, GPTP_NS.PTPGPTPTM0U 074E 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GPTPU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	GPTPU[15:0]	gPTP Timer Value Upper Part	These bits represent upper 16 bits of Seconds of gPTP timer.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

GPTPU[15:0] Bits (gPTP Timer Value Upper Part)

[Update condition]

- HW: Reading the PTPGPTPTM0L register update these bits to upper 16 bits of Seconds of gPTP timer.

39.2.4 Media Clock Capture Registers

39.2.4.1 Media Clock m Capture Configuration Register (PTPMCCCm) (m = 0, 1)

Address(es): GPTP.PTPMCCC0 874E 0200h, GPTP.PTPMCCC1 874E 0210h, GPTP_NS.PTPMCCC0 074E 0200h, GPTP_NS.PTPMCCC1 074E 0210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MCTNS	MCTTS	MCNEE	MCPEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCPEE	Media Clock Capture Positive Edge Enable	0: Timer value will not be captured on positive edge of GPTP_CAPTUREm signal. 1: Timer value will be captured on positive edge of GPTP_CAPTUREm signal.	R/W
b1	MCNEE	Media Clock Capture Negative Edge Enable	0: Timer value will not be captured on negative edge of GPTP_CAPTUREm signal. 1: Timer value will be captured on negative edge of GPTP_CAPTUREm signal.	R/W
b2	MCTTS	Media Clock Capture Timer Type Select	0: Timer value will be captured on gPTP timer. 1: Timer values will be captures on AVTP timer.	R/W
b3	MCTNS	Media Clock Capture Timer Number Select	This register selects from which timer the timer values will be captured. Set this bit to 0 for this product.	R/W
b15 to b4	—	Reserved	These bits are read as 0.	R
b16	MCCR	Media Clock Capture Request	Writing 1 to this register will trig timer capture.	R/W
b31 to b17	—	Reserved	These bits are read as 0.	R

39.2.4.2 Media Clock m Capture Monitoring Register L (PTPMCCMmL) (m = 0, 1)

Address(es): GPTP.PTPMCCM0L 874E 0204h, GPTP.PTPMCCM1L 874E 0214h,
GPTP_NS.PTPMCCM0L 074E 0204h, GPTP_NS.PTPMCCM1L 074E 0214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MCCTVL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MCCTVL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MCCTVL[31:0]	Media Clock Captured Timer Value Lower Part	• If a gPTP timer is captured, these bits represent a part of Nanoseconds of gPTP timer. • If an AVTP timer is captured, these bits represent either the 32bit-AVTP timer or lower 32 bits of the 64bit-AVTP timer.	R

MCCTVL[31:0] Bits (Media Clock Captured Timer Value Lower Part)

[Update conditions]

- Reading the PTPMCCMmL register will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

39.2.4.3 Media Clock m Capture Monitoring Register M (PTPMCCMmM) (m = 0, 1)

Address(es): GPTP.PTPMCCM0M 874E 0208h, GPTP.PTPMCCM1M 874E 0218h,
GPTP_NS.PTPMCCM0M 074E 0208h, GPTP_NS.PTPMCCM1M 074E 0218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MCCTVM[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MCCTVM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MCCTVM[31:0]	Media Clock Captured Timer Value Middle Part	• If a gPTP timer is captured, these bits represent lower 32 bits of Seconds of gPTP timer. • If an AVTP timer is captured, these bits represent upper 32 bits of 64bit-AVTP timer.	R

MCCTVM[31:0] Bits (Media Clock Captured Timer Value Middle Part)

[Update conditions]

- Reading the PTPMCCMmM register will update the PTPMCCMmL register, the PTPMCCMmM register, and PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

39.2.4.4 Media Clock m Capture Monitoring Register U (PTPMCCMmU) (m = 0, 1)

Address(es): GPTP.PTPMCCM0U 874E 020Ch, GPTP.PTPMCCM1U 874E 021Ch, GPTP_NS.PTPMCCM0U 074E 020Ch, GPTP_NS.PTPMCCM1U 074E 021Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	MCCN[1:0]		—	—	—	—	—	MCSWC	MCNEC	MCPEC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MCCTVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MCCTVU[15:0]	Media Clock Captured Timer Value Upper Part	• If a gPTP timer is captured, these bits represent upper 16 bits of Seconds of gPTP timer. • If an AVTP timer is captured, these bits are not used.	R
b16	MCPEC	Media Clock Positive Edge Captured		R
b17	MCNEC	Media Clock Negative Edge Captured		R
b18	MCSWC	Media Clock Software Captured		R
b23 to b19	—	Reserved	These bits are read as 0.	R
b25, b24	MCCN[1:0]	Media Clock Capture Number	These bits indicate the number of captures for the media clock.	R
b31 to b26	—	Reserved	These bits are read as 0.	R

MCCTVU[15:0] Bits (Media Clock Captured Timer Value Upper Part)

[Update conditions]

- Reading the PTPMCCMmU register will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

MCPEC Bit (Media Clock Positive Edge Captured)

[Update conditions]

- Reading the PTPMCCMmU register will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

MCNEC Bit (Media Clock Negative Edge Captured)

[Update conditions]

- Reading the PTPMCCMmU register will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

MCSWC Bit (Media Clock Software Captured)

[Update conditions]

- Reading the PTPMCCMmL register will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the next captures value.
- Capturing a timer while the capture buffer is empty will update the PTPMCCMmL register, the PTPMCCMmM register, and the PTPMCCMmU register to the captured value.

MCCN[1:0] Bits (Media Clock Capture Number)

[Increment conditions]

- HW: Incremented by one when different than 10b and media capture unit m captures a timer.

[Decrement conditions]

- SW: Decremented by one when different than 0 and SW reads.

39.2.5 Media Clock Recovery Registers

39.2.5.1 Media Clock m Recovery Configuration Register (PTPMCRCm) (m = 0, 1)

Address(es): GPTP.PTPMCRC0 874E 0300h, GPTP.PTPMCRC1 874E 0310h, GPTP_NS.PTPMCRC0 074E 0300h, GPTP_NS.PTPMCRC1 074E 0310h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRPL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MRTNS	MRAMS	MRTTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRTTS	Media Clock Recovery Timer Type Select	0: Timer value will be compared to gPTP timer. 1: Timer values will be compared to AVTP timer.	R/W
b1	MRAMS	Media Clock Recovery AVTP Mode Select	0: 64bit-AVTP will be used for comparison. 1: 32bit-AVTP will be used for comparison.	R/W
b2	MRTNS	Media Clock Recovery Timer Number Select	This bit selects from which timer the timer values will be taken. Set this bit to 0 for this product.	R/W
b15 to b3	—	Reserved	These bits are read as 0.	R
b31 to b16	MRPL[15:0]	Media Clock Recovery Pulse Length	Pulse length for clock recovery in ESWCLK clocks – 1	R/W

MRTTS Bit (Media Clock Recovery Timer Type Select)

[Restriction]

- HW: For gPTP timer, overflow is not supported. If a time less than current gPTP timer is set, the comparison will match instantaneously.

MRAMS Bit (Media Clock Recovery AVTP Mode Select)

[Restriction]

- HW: For 64-AVTP timer, overflow is not supported. If a time less than current AVTP timer is set, the comparison will match instantaneously.

39.2.5.2 Media Clock m Recovery Time Configuration Register L (PTPMCRTCmL) (m = 0, 1)

Address(es): GPTP.PTPMCRTC0L 874E 0304h, GPTP.PTPMCRTC1L 874E 0314h, GPTP_NS.PTPMCRTC0L 074E 0304h, GPTP_NS.PTPMCRTC1L 074E 0314h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRTVL[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRTVL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MRTVL[31:0]	Media Clock Recovery Timer Value Lower Part	• If the compared timer is a gPTP timer, these bits represent a part of Nanoseconds of gPTP timer. • If the compared timer is a 32-bit AVTP timer, these bits represent the 32bit-AVTP timer. • If the compared timer is a 64-bit AVTP timer, these bits represent lower 32 bits of 64-bit AVTP timer.	R/W

MRTVL[31:0] Bits (Media Clock Recovery Timer Value Lower Part)

Writing to this register will load the PTPMCCMmU.MCCTVU[15:0] bits, the PTPMCCMmM.MCCTVM[31:0] bits, and the PTPMCCMmL.MCCTVL[31:0] bits for comparison in the recovery buffer.

39.2.5.3 Media Clock m Recovery Time Configuration Register M (PTPMCRTCmM) (m = 0, 1)

Address(es): GPTP.PTPMCRTC0M 874E 0308h, GPTP.PTPMCRTC1M 874E 0318h, GPTP_NS.PTPMCRTC0M 074E 0308h, GPTP_NS.PTPMCRTC1M 074E 0318h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRTVM[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRTVM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MRTVM[31:0]	Media Clock Recovery Timer Value Middle Part	• If the compared timer is a gPTP timer, these bits represent lower 32 bits of Seconds of gPTP timer. • If the compared timer is a 32-bit AVTP timer, these bits represent nothing relevant. • If the compared timer is a 64-bit AVTP timer, these bits represent upper 32 bits of 64bit-AVTP timer.	R/W

39.2.5.4 Media Clock m Recovery Time Configuration Register U (PTPMCRTCmU) (m = 0, 1)

Address(es): GPTP.PTPMCRTC0U 874E 030Ch, GPTP.PTPMCRTC1U 874E 031Ch, GPTP_NS.PTPMCRTC0U 074E 030Ch, GPTP_NS.PTPMCRTC1U 074E 031Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRBCR	—	—	—	—	—	—	—	—	—	—	MCRN[2:0]			MRTT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRTVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MRTVU[15:0]	Media Clock Recovery Timer Value Upper Part	• If the compared timer is a gPTP timer, these bits represent upper 16 bits of Seconds of gPTP timer. • If the compared timer is a 32-bit AVTP timer, these bits represents nothing relevant. • If the compared timer is a 64-bit AVTP timer, these bits represent nothing relevant.	R/W
b17, b16	MRTT[1:0]	Media Clock Recovery Trigger Type	b17 b16 0 0: When timer match, time recovery pin is set to 1. 0 1: When timer match, time recovery pin is set to 0. 1 0: When timer match, time recovery pin is inverted. 1 1: When timer match, a positive pulse of length the PTPMCRCm.MRPL[15:0] bits is output on time recovery pin.	R/W
b20 to b18	MCRN[2:0]	Media Clock Recovery Number	Displays the number on entry in the recovery buffer.	R
b30 to b21	—	Reserved	These bits are read as 0.	R
b31	MRBCR	Media Clock Recovery Buffer Clear Request	Writing 1 to this bit will clear the internal timer buffer along with the PTPMCRTCmU.MCRN[2:0] bits.	R/W

39.2.5.5 Media Clock m Pin Configuration Register (PTPMCPCm) (m = 0, 1)

Address(es): GPTP.PTPMCPC0 874E 0400h, GPTP.PTPMCPC1 874E 0404h,
GPTP_NS.PTPMCPC0 074E 0400h, GPTP_NS.PTPMCPC1 074E 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRS	PE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PE	Pin Enable	0: The GPTP_MATCHm pin is fixed to low 1: The GPTP_MATCHm pin outputs the comparison result of the PTPMCPCm.MRS bit recovery logic	R/W
b1	MRS	Media Clock Recovery Select	Select recovery logic for GPTP_MATCHm	R/W
b31 to b2	—	Reserved	These bits are read as 0.	R

PE Bit (Pin Enable)

[Restriction]

- Do not change this setting on any timer enabled (the PTPTMEC.TE0 bit = 1).

MRS Bit (Media Clock Recovery Select)

[Restriction]

- Do not change this setting on any timer enabled (the PTPTMEC.TE0 bit = 1).

39.2.6 Cyclic Compare Registers

39.2.6.1 Cyclic Compare c Configuration Register 0 (PTPCCCc0) (c = 0 to 7)

Address(es): GPTP.PTPCCC00 874E 0500h, GPTP.PTPCCC10 874E 0508h, GPTP.PTPCCC20 874E 0510h, GPTP.PTPCCC30 874E 0518h, GPTP.PTPCCC40 874E 0520h, GPTP.PTPCCC50 874E 0528h, GPTP.PTPCCC60 874E 0530h, GPTP.PTPCCC70 874E 0538h, GPTP_NS.PTPCCC00 074E 0500h, GPTP_NS.PTPCCC10 074E 0508h, GPTP_NS.PTPCCC20 074E 0510h, GPTP_NS.PTPCCC30 074E 0518h, GPTP_NS.PTPCCC40 074E 0520h, GPTP_NS.PTPCCC50 074E 0528h, GPTP_NS.PTPCCC60 074E 0530h, GPTP_NS.PTPCCC70 074E 0538h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CCOPS	—	—	—	CCTNS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCTNS	Cyclic Compare Timer Number Select	This bit selects from which timer the cyclic compare will be based on.	R/W
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	CCOPS	Cyclic Compare Output Pin Select	CYCLIC_COMP[the PTPCCCc0.CCOPS bit][c] is selected as output for cyclic compare c.	R/W
b31 to b5	—	Reserved	These bits are read as 0.	R

39.2.6.2 Cyclic Compare c Configuration Register 1 (PTPCCCc1) (c = 0 to 7)

Address(es): GPTP.PTPCCC01 874E 0504h, GPTP.PTPCCC11 874E 050Ch, GPTP.PTPCCC21 874E 0514h, GPTP.PTPCCC31 874E 051Ch, GPTP.PTPCCC41 874E 0524h, GPTP.PTPCCC51 874E 052Ch, GPTP.PTPCCC61 874E 0534h, GPTP.PTPCCC71 874E 053Ch, GPTP_NS.PTPCCC01 074E 0504h, GPTP_NS.PTPCCC11 074E 050Ch, GPTP_NS.PTPCCC21 074E 0514h, GPTP_NS.PTPCCC31 074E 051Ch, GPTP_NS.PTPCCC41 074E 0524h, GPTP_NS.PTPCCC51 074E 052Ch, GPTP_NS.PTPCCC61 074E 0534h, GPTP_NS.PTPCCC71 074E 053Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCV[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	CCV[31:0]	Cyclic Compare Value	Setting these bits to a value less than 32 will disable compare function. Setting these bits to a value greater or equal to 32 will enable compare function.	R/W

39.2.7 Interrupt Registers

39.2.7.1 Interrupt Status Register 0 (PTPIS0)

Address(es): GPTP.PTPIS0 874E 0700h, GPTP_NS.PTPIS0 074E 0700h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOES1	MCCOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCS1	MCCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCCS0	Media Clock 0 Capture Status Flag	0: No Media Clock 0 Capture 1: Media Clock 0 Capture occurred.	R/W
b1	MCCS1	Media Clock 1 Capture Status Flag	0: No Media Clock 1 Capture 1: Media Clock 1 Capture occurred.	R/W
b15 to b2	—	Reserved	These bits are read as 0.	R
b16	MCCOES0	Media Clock 0 Capture Overflow Error Status Flag	0: No Media Clock 0 Capture Overflow Error 1: Media Clock 0 Capture Overflow Error occurred.	R/W
b17	MCCOES1	Media Clock 1 Capture Overflow Error Status Flag	0: No Media Clock 1 Capture Overflow Error 1: Media Clock 1 Capture Overflow Error occurred.	R/W
b31 to b18	—	Reserved	These bits are read as 0.	R

MCCSm Flag (Media Clock m Capture Status Flag) (m = 0, 1)

[Setting condition]

- HW: Media clock capture unit m captured a timer.

[Clearing condition]

- SW: Writing 1 to this flag will clear it.

MCCOESm Flag (Media Clock m Capture Overflow Error Status Flag) (m = 0, 1)

[Setting condition]

- HW: This flag is set to 1 when media clock capture unit m captured a timer but it could not be stored because the timer FIFO is full (the PTPMCCMmU.MCCN[1:0] bits equal to 10b).

[Clearing condition]

- SW: Writing 1 to this flag will clear it.

[Error recovery]

- HW: The captured timer is lost. Next capture will continue normally.
- SW: The timer reading is too slow, SW should be reviewed.

39.2.7.2 Interrupt Enable Register 0 (PTPIE0)

Address(es): GPTP.PTPIE0 874E 0704h, GPTP_NS.PTPIE0 074E 0704h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOEE1	MCCOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCE1	MCCE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCCE0	Media Clock 0 Capture Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	MCCE1	Media Clock 1 Capture Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b15 to b2	—	Reserved	These bits are read as 0.	R
b16	MCCOEE0	Media Clock 0 Capture Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b17	MCCOEE1	Media Clock 1 Capture Overflow Error Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b18	—	Reserved	These bits are read as 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the PTPID0 register.

MCCEm Bit (Media Clock m Capture Enable) (m = 0, 1)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- SW: Writing 1 to the PTPID0.MCCDm bit will clear this bit.

MCCOEEm Bit (Media Clock m Capture Overflow Error Enable) (m = 0, 1)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- SW: Writing 1 to the PTPID0.MCCOEDm bit will clear this bit.

39.2.7.3 Interrupt Disable Register 0 (PTPID0)

Address(es): GPTP.PTPID0 874E 0708h, GPTP_NS.PTPID0 074E 0708h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOED1	MCCOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCD1	MCCD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCCD0	Media Clock 0 Capture Disable	0: Has no effects. 1: Sets PTPIE0.MCCE0 bit to 0.	R/W
b1	MCCD1	Media Clock 1 Capture Disable	0: Has no effects. 1: Sets PTPIE0.MCCE1 bit to 0.	R/W
b15 to b2	—	Reserved	These bits are read as 0.	R
b16	MCCOED0	Media Clock 0 Capture Overflow Error Disable	0: Has no effects. 1: Sets PTPIE0.MCCOEE0 bit to 0.	R/W
b17	MCCOED1	Media Clock 1 Capture Overflow Error Disable	0: Has no effects. 1: Sets PTPIE0.MCCOEE1 bit to 0.	R/W
b31 to b18	—	Reserved	These bits are read as 0.	R

39.2.7.4 Interrupt Status Register 1 (PTPIS1)

Address(es): GPTP.PTPIS1 874E 0710h, GPTP_NS.PTPIS1 074E 0710h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRMS1	MCRMS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCRMS0	Media Clock 0 Recovery Match Status Flag	0: No Media Clock 0 Recovery Match 1: Media Clock 0 Recovery Match occurred.	R/W
b1	MCRMS1	Media Clock 1 Recovery Match Status Flag	0: No Media Clock 1 Recovery Match 1: Media Clock 1 Recovery Match occurred.	R/W
b31 to b2	—	Reserved	These bits are read as 0.	R

MCRMSn Flag (Media Clock n Recovery Match Status Flag) (n = 0, 1)

[Setting condition]

- HW: Media clock capture unit m oldest timer matched.

[Clearing condition]

- SW: Writing 1 to this flag will clear it.

39.2.7.5 Interrupt Enable Register 1 (PTPIE1)

Address(es): GPTP.PTPIE1 874E 0714h, GPTP_NS.PTPIE1 074E 0714h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRME1	MCRME0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCRME0	Media Clock 0 Recovery Match Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b1	MCRME1	Media Clock 1 Recovery Match Enable	0: Interrupt disabled 1: Interrupt enabled	R/(W) *1
b31 to b2	—	Reserved	These bits are read as 0.	R

Note 1. Only 1 can be written. To set this bit to 0, write 1 to the corresponding bit in the PTPID1 register.

MCRME0 Bit (Media Clock 0 Recovery Match Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- SW: Writing 1 to the PTPID1.MCRMD0 bit will clear this bit.

MCRME1 Bit (Media Clock 1 Recovery Match Enable)

[Setting condition]

- SW: Writing 1 to this bit will set it.

[Clearing condition]

- SW: Writing 1 to the PTPID1.MCRMD1 bit will clear this bit.

39.2.7.6 Interrupt Disable Register 1 (PTPID1)

Address(es): GPTP.PTPID1 874E 0718h, GPTP_NS.PTPID1 074E 0718h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRMD1	MCRMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MCRMD0	Media Clock 0 Recovery Match Disable	0: Has no effects. 1: Sets PTPIE1.MCRME0 bit to 0.	R/W
b1	MCRMD1	Media Clock 1 Recovery Match Disable	0: Has no effects. 1: Sets PTPIE1.MCRME1 bit to 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0.	R

39.2.8 Pulse Output Timer Configuration Registers

39.2.8.1 Pulse Output Timer Configuration Register (POTCFGR)

Address(es): GPTP.POTCFGR 874E 1000h, GPTP_NS.POTCFGR 074E 1000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REFSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REFSEL	Reference Timer Select	0: Select and use the reference time of gPTP timer 0. 1: Setting prohibited	R/W
b31 to b1	—	Reserved	These bits are read as 0.	R

39.2.8.2 Pulse Output Timer Control Register n (POTCRn) (n = 0, 1)

Address(es): GPTP.POTCR0 874E 1004h, GPTP.POTCR1 874E 1034h,
GPTP_NS.POTCR0 074E 1004h, GPTP_NS.POTCR1 074E 1034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	START	Timer Start	0: Pulse output stop 1: Pulse output enable • Setting 1 enables pulse output generation and allows pulse output. • If set to 0, no pulse will be output. By writing 0 during pulse output, pulse output stops (0 output). • Pulse generation is performed by taking in the set values of the pulse generation registers when this bit is set to 1.	R/W
b31 to b1	—	Reserved	These bits are read as 0.	R

39.2.8.3 Pulse Output Start Time Setting Register nU (POTSTRnU) (n = 0, 1)

Address(es): GPTP.POTSTR0U 874E 1008h, GPTP.POTSTR1U 874E 1038h,
GPTP_NS.POTSTR0U 074E 1008h, GPTP_NS.POTSTR1U 074E 1038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	Seconds (upper 16 bits) of pulse output start time. • Set the upper 16-bit seconds of the time to start pulse output. • For comparison with the reference time of the gPTP timer. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read.	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

39.2.8.4 Pulse Output Start Time Setting Register nM (POTSTRnM) (n = 0, 1)

Address(es): GPTP.POTSTR0M 874E 100Ch, GPTP.POTSTR1M 874E 103Ch, GPTP_NS.POTSTR0M 074E 100Ch, GPTP_NS.POTSTR1M 074E 103Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	Seconds (lower 32 bits) of pulse output start time. • Sets the lower 32 bits second of the time to start pulse output. • For comparison with the reference time of the gPTP timer. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read.	R/W

39.2.8.5 Pulse Output Start Time Setting Register nL (POTSTRnL) (n = 0, 1)

Address(es): GPTP.POTSTR0L 874E 1010h, GPTP.POTSTR1L 874E 1040h,
GPTP_NS.POTSTR0L 074E 1010h, GPTP_NS.POTSTR1L 074E 1040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	—	—	Nano seconds (30 bits) of pulse output start time. • Sets the nano second of the time to start pulse output. • For comparison with the reference time of the gPTP timer. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read. Restrictions • Set the setting value between 0 and 3B9AC9FFh. Note: 3B9ACA00h is 1 second. Please set with the register of second.	R/W
b31, b30	—	Reserved	These bits are read as 0.	R

39.2.8.6 Period Setting Register nU (POTPERnU) (n = 0, 1)

Address(es): GPTP.POTPER0U 874E 1014h, GPTP.POTPER1U 874E 1044h, GPTP_NS.POTPER0U 074E 1014h, GPTP_NS.POTPER1U 074E 1044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	Seconds (upper 16 bits) of pulse output Period. • Set the upper 16-bit seconds of the time to start pulse output. • For calculating the reference time of the start time to be compared next. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read.	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

39.2.8.7 Period Setting Register nM (POTPERnM) (n = 0, 1)

Address(es): GPTP.POTPER0M 874E 1018h, GPTP.POTPER1M 874E 1048h,
GPTP_NS.POTPER0M 074E 1018h, GPTP_NS.POTPER1M 074E 1048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	Seconds (lower 32 bits) of pulse output period. • Set the lower 32-bit seconds period of the pulse output. For calculating the reference time of the start time to be compared next. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read.	R/W

39.2.8.8 Period Setting Register nL (POTPERnL) (n = 0, 1)

Address(es): GPTP.POTPER0L 874E 101Ch, GPTP.POTPER1L 874E 104Ch, GPTP_NS.POTPER0L 074E 101Ch, GPTP_NS.POTPER1L 074E 104Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	—	—	Nano seconds (30 bits) of pulse output period. • Sets the nano second of the time to period pulse output. • For calculating the reference time of the start time to be compared next. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read. Restrictions • Set the setting value between 0 and 3B9AC9FFh. Note: 3B9ACA00h is 1 second. Please set with the register of second.	R/W
b31, b30	—	Reserved	These bits are read as 0.	R

39.2.8.9 Pulse Width Setting Register n (POTPWRn) (n = 0, 1)

Address(es): GPTP.POTPWR0 874E 1020h, GPTP.POTPWR1 874E 1050h,
GPTP_NS.POTPWR0 074E 1020h, GPTP_NS.POTPWR1 074E 1050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	High Pulse Width Count number for pulse output. • Sets the number of counts for the high pulse width. • Counting starts from the start of pulse output. • The high pulse width time varies depending on the clock period. Valid value • The register value used to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1. • Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Note: When reading, the written value can be read.	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

39.2.9 Pulse Output Timer Monitoring Registers

39.2.9.1 Time Capture Register nU (POTCPRnU) (n = 0, 1)

Address(es): GPTP.POTCPR0U 874E 1028h, GPTP.POTCPR1U 874E 1058h, GPTP_NS.POTCPR0U 074E 1028h, GPTP_NS.POTCPR1U 074E 1058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits hold the upper 16 bits of the Seconds of the pulse output start time. • Can read the Upper 16-bit Seconds of the time when the pulse output started.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

39.2.9.2 Time Capture Register nM (POTCPRnM) (n = 0, 1)

Address(es): GPTP.POTCPR0M 874E 102Ch, GPTP.POTCPR1M 874E 105Ch, GPTP_NS.POTCPR0M 074E 102Ch, GPTP_NS.POTCPR1M 074E 105Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the lower 32 bits of the Seconds of the pulse output start time. • Can read the Lower 32-bit Seconds of the time when the pulse output started.	R

39.2.9.3 Time Capture Register nL (POTCPRnL) (n = 0, 1)

Address(es): GPTP.POTCPR0L 874E 1030h, GPTP.POTCPR1L 874E 1060h,
GPTP_NS.POTCPR0L 074E 1030h, GPTP_NS.POTCPR1L 074E 1060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	—	—	These bits hold the nano seconds of the pulse output start time. • Can read the nano seconds of the time when the pulse output started.	R
b31, b30	—	Reserved	These bits are read as 0.	R

39.3 Operations

39.3.1 Timer 0

Each timer instantiation possesses three timers. A gPTP timer, a 64-bit AVTP timer, and a 32-bit AVTP timer.

39.3.1.1 gPTP Timer 0

The gPTP timer 0 GPTP[0] can be enabled by setting the PTPTMEC.TE0 bit to 1. When enabled, gPTP timer 0 will automatically calculate its value by using equation (1).

(1) *gptpTimer*[0] = *TimerOffset*[0] + *FreeRunningTimerValue*[0]

Where:

- *gptpTimer*[0] is the calculated timer value GPTP[0] and is divided in two parts. The second part *gptpSecond*[0] which is GPTP[0][77:30] and the nanosecond part *gptpNano*[0] which is GPTP[0][29:0].
- *TimerOffset*[0] is {the PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, the PTPTOVC0L.TOVL[29:0] bits}
- *FreeRunningTimerValue*[0] is the value of a free running timer starting from 0 (when timer is disabled) and being incremented by the PTPTIVC0.TIV[31:0] bits every cycle (The free running timer has a 27-bit sub-nanosecond part not described here so it should be set to *clk_period*[ns]*2^{27} where *clk_period* is the ESWCLK period).

Functions:

- When gPTP timer 0 is a master, the PTPTIVC0.TIV[31:0] bits and {the PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, the PTPTOVC0L.TOVL[29:0] bits} can be kept to fixed values.
- When gPTP timer 0 is a slave, the PTPTIVC0.TIV[31:0] bits can be used in priority for time re-synchronization. If gPTP timer 0 is too far from the corresponding master value, {the PTPTOVC0U.TOVU[15:0] bits, the PTPTOVC0M.TOVM[31:0] bits, the PTPTOVC0L.TOVL[29:0] bits} can be used for time re-synchronization.

39.3.1.2 64-bit AVTP Timer 0

64-bit AVTP timer 0 AVTP64[0] is used as a reference to synchronize any necessary media clocks and to determine when the payload of a stream is to be presented to the time-sensitive application. Media clocks vary with audio/video types therefore the exact usage of the AVTP presentation time is media format dependent [1722].
64-bit AVTP timer 0 is generated based on the gPTP timer 0 and can be calculated by using equation (2).

(2) *avtpTimer64*[0] = (*gptpSecond*[0] * 10^9 + *gptpNano*[0]) mod 2^{64}

Where:

- *avtpTimer64*[0] is the calculated timer value AVTP64[0].
- *gptpSecond*[0] is *gptpTimer*[0] second part GPTP[0][77:30].
- *gptpNano*[0] is *gptpTimer*[0] nanosecond part GPTP[0][29:0].

39.3.1.3 32-bit AVTP Timer 0

32-bit AVTP timer 0 AVTP32[0] is the 32-bit version of 64-bit AVTP timer 0 and can be calculated using equation (3).

(3) $avtpTimer32[0] = (gptpSecond[0] * 10^9 + gptpNano[0]) \bmod 2^{32}$

Where:

- *avtpTimer32*[0] is the calculated timer value AVTP32[0] (AVTP64[0][31:0]).
- *gptpSecond*[0] is *gptpTimer*[0] second part GPTP[0][77:30].
- *gptpNano*[0] is *gptpTimer*[0] nanosecond part GPTP[0][29:0].

39.3.1.4 PPS Signal

GPTP0_PPS signal is used to check if gPTP synchronization is success full. As per gPTP standard [GPTP], GPTP0_PPS signal should be output as pins at chip level to allow synchronization checking on oscilloscope. GPTP0_PPS is the PPS signal corresponding to timer number 0. For PPS signal details, refer to gPTP standard [GPTP].

39.3.2 Media Clock Capture m (m = 0, 1)

Media clock capture m aims at capturing timer values on the GPTP_CAPTUREm positive edge and/or negative edge.

Functions:

- The PTPMCCCm.MCPEE bit is used to enable timer value capture on the GPTP_CAPTUREm positive edge.
- The PTPMCCCm.MCNEE bit is used to enable timer value capture on the GPTP_CAPTUREm negative edge.
- The PTPMCCCm.MCTNS bit is used to select timer number for timer value capture.
- The PTPMCCCm.MCTTS bit is used to select between gPTP and AVTP timer for timer value capture.
- The PTPMCCMmL/PTPMCCMmM/PTPMCCMmU register are used to read the captured timer values.
- The PTPIS0.MCCSm flag is used to signal CPU that a timer value has been captured.
- The PTPIS0.MCCOESm flag is used to signal CPU that a captured timer value has been lost because the timer capture value FIFO were full.

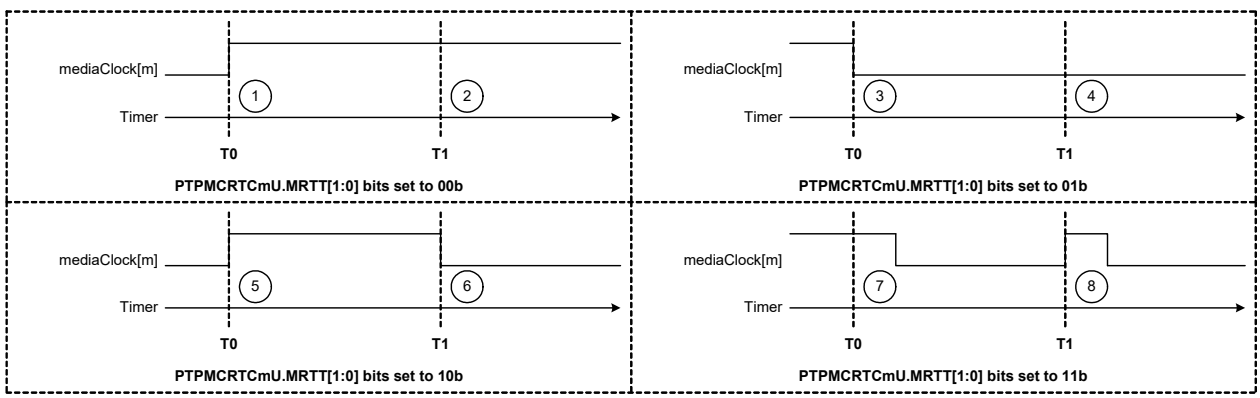
39.3.3 Media Clock Recovery m (m = 0, 1)

Media clock recovery m aims at creating an internal clock mediaClock[m] which can be mapped to an output pin (section 39.3.4, mediaClock[m] Mapping to Output Pins).

Functions:

- The PTPMCRCm.MRTNS bit is used to select timer number for clock recovery.
- The PTPMCRCm.MRTTS bit is used to select between gPTP and AVTP timer for clock recovery.
- The PTPMCRCm.MRAMS bit is used to select between 32-bit AVTP timer and 64-bit AVTP timer for clock recovery when the PTPMCRCm.MRTTS bit is set to AVTP timer.
- The PTPMCRTCmL/PTPMCRTCmM/PTPMCRTCmU register are used to add a timer value and command to execute to mediaClock[m].
- The PTPIS1.MCRMS flag is used to signal CPU that the oldest timer value which did not match yet set in PTPMCRTCmL/PTPMCRTCmM/PTPMCRTCmU register matched the selected timer so mediaClock[m] value has been updated.

Figure 39.3 describes mediaClock[m] behavior when a timer value matches depending on the timer value type setting and the initial value of mediaClock[m].



1	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 00b matched at Timer T0 when mediaClock[m] is not set. mediaClock[m] gets set.
2	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 00b matched at Timer T1 when mediaClock[m] is set. mediaClock[m] stays set.
3	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 01b matched at Timer T0 when mediaClock[m] is set. mediaClock[m] gets cleared.
4	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 01b matched at Timer T1 when mediaClock[m] is not set. mediaClock[m] stays cleared.
5	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 10b matched at Timer T0 when mediaClock[m] is not set. mediaClock[m] gets inverted.
6	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 10b matched at Timer T1 when mediaClock[m] is set. mediaClock[m] gets inverted.
7	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 11b matched at Timer T0 when mediaClock[m] is set. mediaClock[m] stays set for **PTPMCRCm.MRPL[15:0]**+1 clock cycles.
8	Compare entry set with **PTPMCRTCmU.MRTT[1:0] bits** set to 11b matched at Timer T1 when mediaClock[m] is not set. mediaClock[m] gets set for **PTPMCRCm.MRPL[15:0]**+1 clock cycles.

Figure 39.3 mediaClock[m] Behavior

Note: When timer matched for an entry with PTPMCRTCmU.MRTT[1:0] set to 11b and the next matched happens before mediaClock[m] de-assertion, mediaClock[m] de-assertion command is overwritten and will not happen.

39.3.4 mediaClock[m] Mapping to Output Pins

The mediaClock[m] signal created by media clock recovery m (section 39.3.3, Media Clock Recovery m (m = 0, 1)) can be output to be used inside the chip or to be output as a chip pin.

Functions:

- By setting the PTPMCPCm.PE bit to 1 and the PTPMCPCm.MRS bit to m, mediaClock[m] signal will be assigned to the GPTP_MATCHm pin.

39.3.5 Cyclic Compare c (c = 0 to 7)

The cyclic compare c aims at creating a pulse cyclically on CYCLIC_COMP[i][c] for i = 0, 1 interface to trigger an event. It can be used, for example, to trigger a periodic data transmission for gPTP synchronization (network control frame transmission).

Functions:

- The PTPCCCc0.CCTNS bit is used to select timer number for cyclic compare. During cyclic compare, timer offset is ignored, and only incremental value is taken in account.
- The PTPCCCc1.CCV[31:0] bits are used to enable cyclic compare. When cyclic compare c is enabled, a pulse is sent every PTPCCCc1.CCV[31:0] bits to CYCLIC_COMP[the PTPCCCc0.CCOPS bit][c].
- When this feature is enabled (when the PTPCCCc1.CCV[31:0] bits configuration is updated to a value greater than 32), the first pulse from the comparator “c” is output immediately (after one system clock). (Regardless of the PTPCCCc1.CCV[31:0] bits configuration)

39.3.6 Pulse Output Timer n (n = 0, 1)

This module uses the reference time of the Ether Switch gPTP timer to generate a pulse output of any period and any pulse width.

39.3.6.1 gPTP Timer Selection

Select the gPTP timer of the reference time to be used by setting the POTCFGR.REFSEL bit.

The REFSEL bit Register Setting:
0: Generates a pulse output using the reference time of gPTP timer 0.

Functions:

- The reference time used for pulse generation can be selected by setting the POTCFGR.REFSEL bit.
- When the REFSEL bit = 0: Generates a pulse using the time input signal of gPTP timer 0.

Note: When changing the POTCFGR.REFSEL bit, stop the pulse output (the POTCRn.START bit = 0) before changing.

39.3.6.2 Pulse Output Generation Registers

By setting the following registers, it is possible to generate a high pulse with an arbitrary start time, an arbitrary period, and an arbitrary pulse width.

- Pulse output start time setting register: POTSTRnU, POTSTRnM, POTSTRnL
- Period setting register: POTPERnU, POTPERnM, POTPERnL
- Pulse width setting register: POTPWRn

If the value of the pulse width setting register is 0000h, no pulse is generated and the output is fixed at low.
If the period set in the period setting register is shorter than or equal to the pulse width, the output is fixed at high.

39.3.6.3 Pulse Output Start

By changing the POTCRn.START bit from 0 to 1, the values of the pulse output generation registers are captured and held. Pulses are generated according to the set value of the held pulse output generation registers.
The output pulse is as follows.

- Rise of the first pulse:
  Pulse width setting register ≠ 0000h: It goes high at the set start time.
  Pulse width setting register = 0000h: Low output (no pulse output)
- Fall of the pulse:
  Period > pulse width: It goes low when the number of ESWCLK is counted for the value of the pulse width setting register after the output goes high.
  Period ≤ pulse width: The output remains high until the POTCRn.START bit is set to 0.
- Rise after the second time:
  When pulse width ≠ 0000h and period > pulse width: It goes high at the time of the previous start time + period.

39.3.6.4 Pulse Output Stop

Pulse output can be stopped by setting the POTCRn.START bit to 0.
When stopped, the output is fixed at low.

When the POTCRn.START bit is set to 1 again, the register setting value is taken in and held, and a new pulse is generated and output.

Note: Even if the pulse output is stopped in the middle, it will not be restarted. A new pulse is generated and output.

39.3.6.5 Timing of Pulse Output Permission

If the POTCRn.START bit is changed from 0 to 1 after the set pulse output start time, no pulse output will be performed.
The pulse is output from the time when the reference time goes around once and the start time is reached again.

39.3.6.6 Interrupt Output

Detects the rising edge of the pulse output and outputs an interrupt signal synchronized with PCLKA.
There are 2 channels of interrupt output, and the interrupt signal corresponding to each pulse output channel is output.
The interrupt signal outputs high with a width of 1 Cycle of PCLKA.

39.3.7 Pulse Output Time Chart

An example of pulse output is shown below.

Case 1: When changing the POTCRn.START bit from 0 to 1 after setting the register and before the output start time
Case 2: When changing the POTCRn.START bit from 0 to 1 after setting the register and after the output start time
Case 3: When changing the POTCRn.START bit from 0 to 1 by setting the pulse width ≥ period
Case 4: When changing the POTCRn.START bit from 0 to 1 with pulse width = 0
Case 5: When stopping pulse output (when setting the POTCRn.START bit to 0)
Case 6: When the register setting is changed when the POTCRn.START bit is 1

(1) Case 1

When changing the POTCRn.START bit from 0 to 1 after setting the register and before the output start time

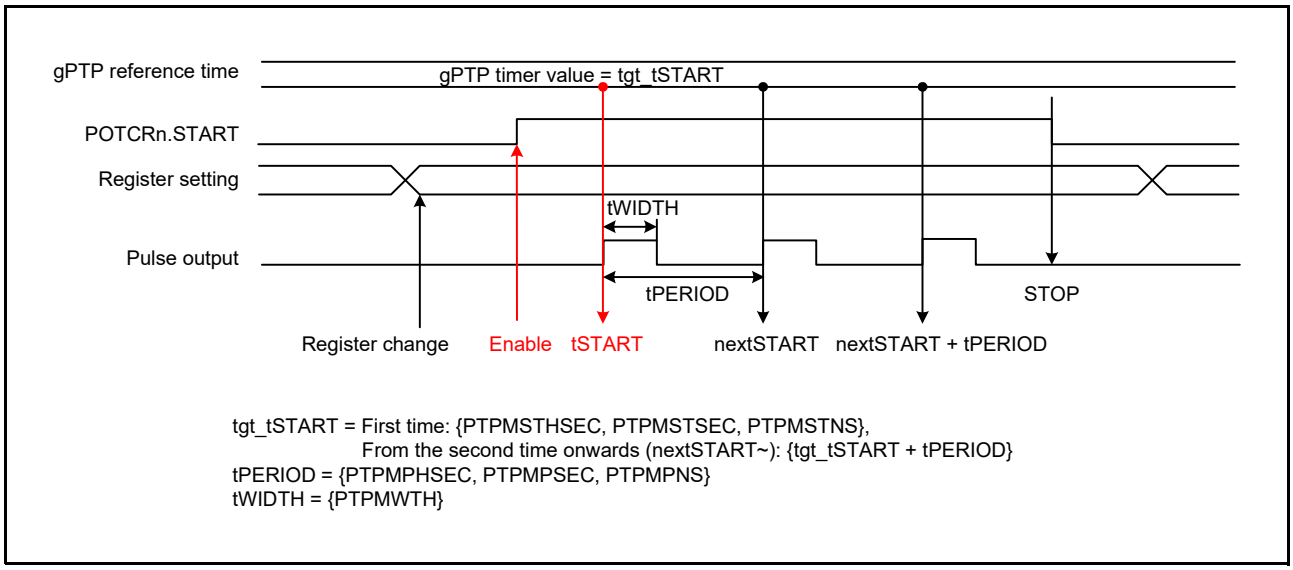


Figure 39.4 Example of Pulse Output Time (1)

(2) Case 2

When changing the POTCRn.START bit from 0 to 1 after setting the register and after the output start time

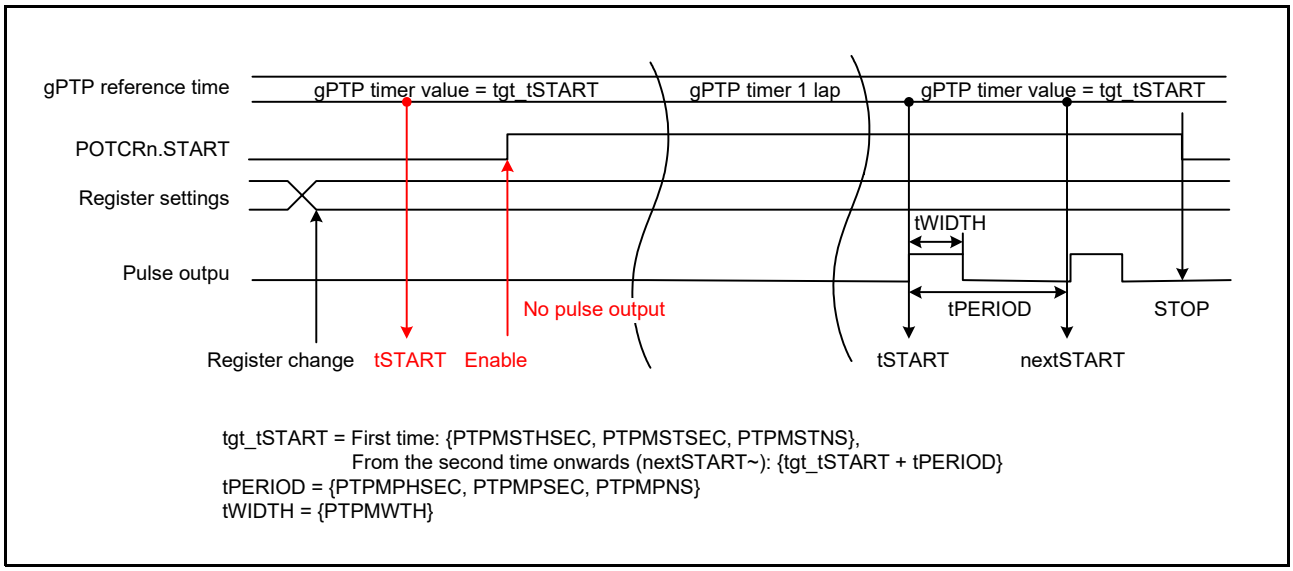


Figure 39.5 Example of Pulse Output Time (2)

(3) Case 3

When the POTCRn.START bit is changed from 0 to 1 by setting the pulse width ≥ the period

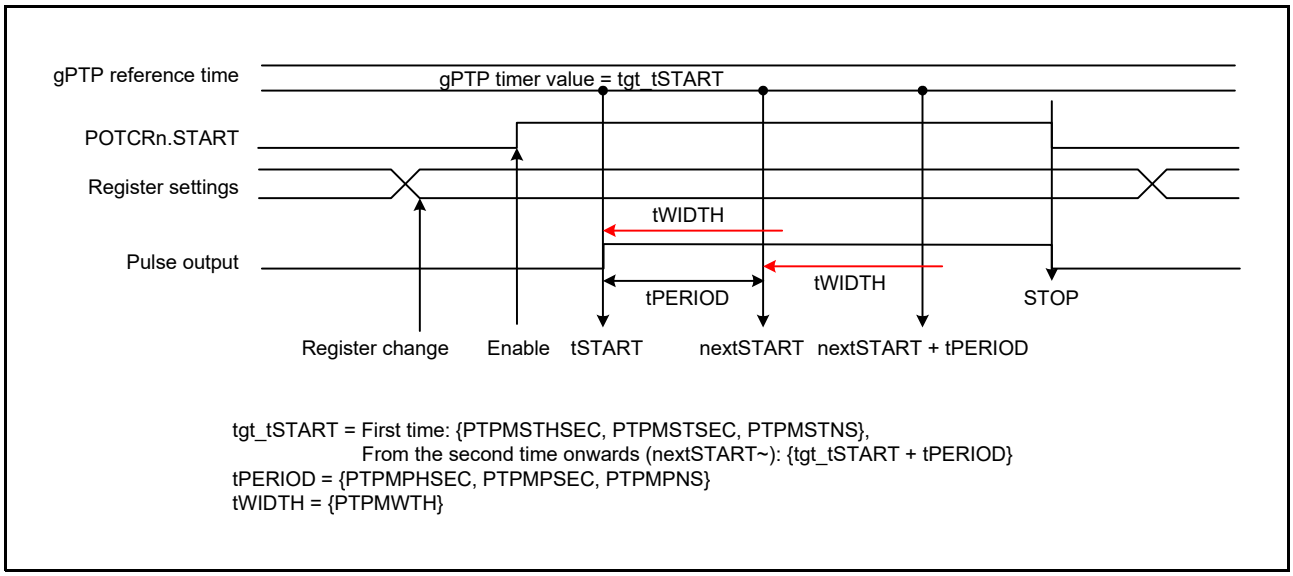


Figure 39.6 Example of Pulse Output Time (3)

(4) Case 4

When changing the POTCRn.START bit from 0 to 1 with pulse width = 0

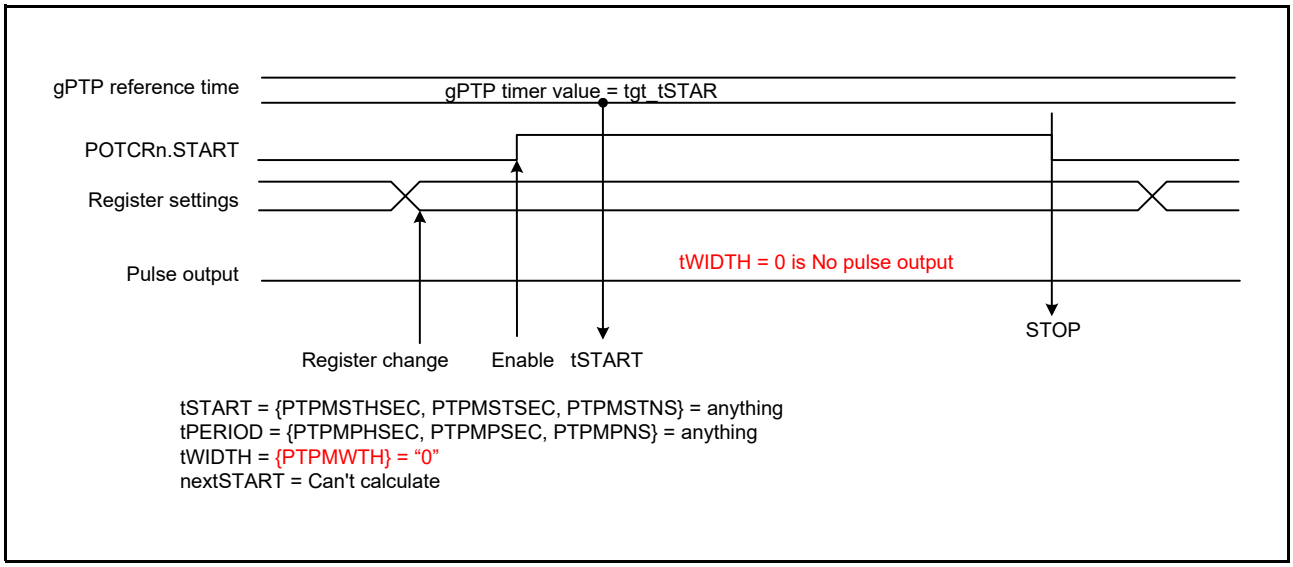


Figure 39.7 Example of Pulse Output Time (4)

(5) Case 5

When the pulse output is stopped (when the POTCRn.START bit is set to 0)

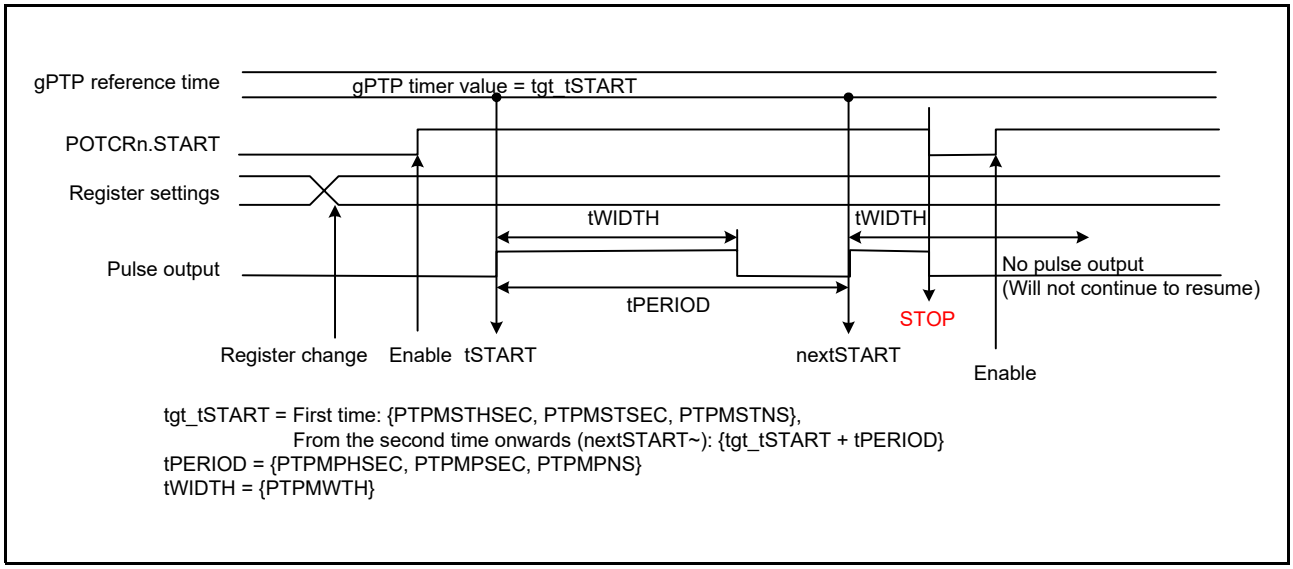


Figure 39.8 Example of Pulse Output Time (5)

(6) Case 6

When the register setting is changed when the POTCRn.START bit is 1

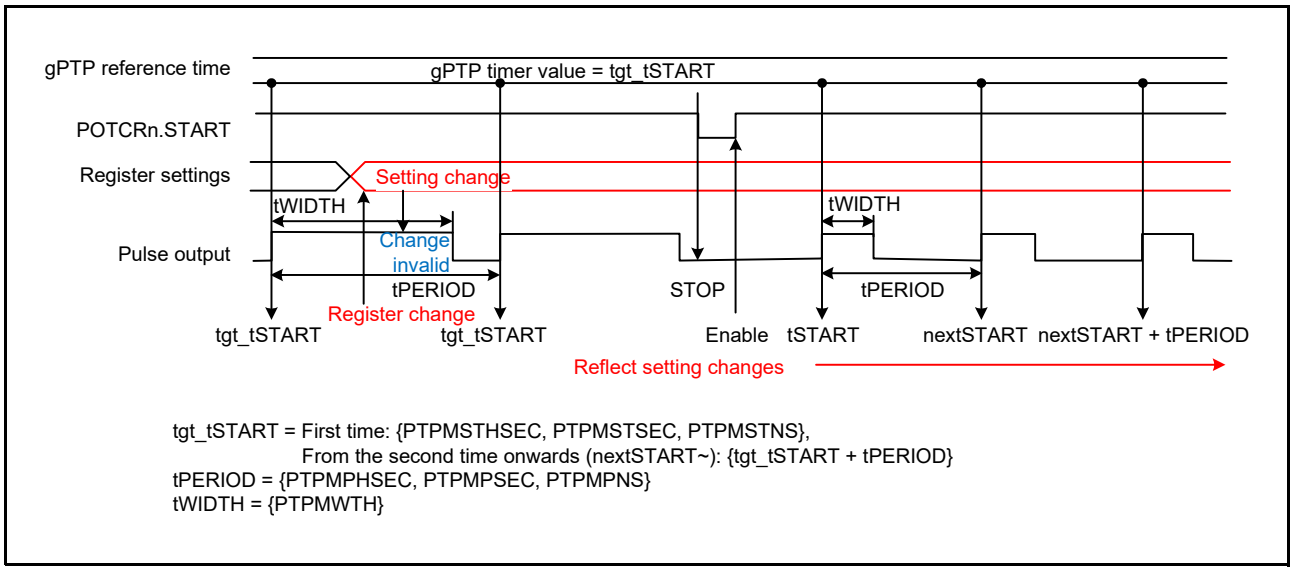


Figure 39.9 Example of Pulse Output Time (6)

39.3.8 Interrupt Output Time Chart

The time chart of interrupt output is shown below.
The interrupt output occurs only when the rising edge of the pulse output is detected.

(1) Interrupt Output

Detects the rising edge of the pulse output and outputs one pulse synchronized with PCLKA.

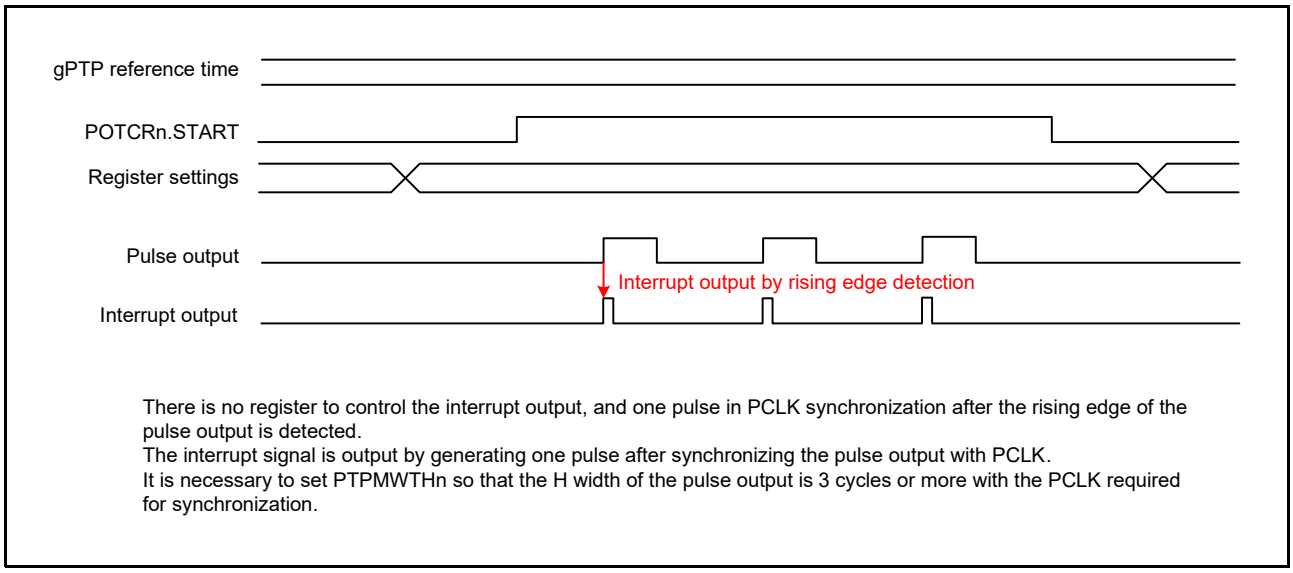


Figure 39.10 Example of Interrupt Output

39.3.9 Software Flows

39.3.9.1 Interrupt Handling Flow

The interrupt handling flow is described in Figure 39.11.

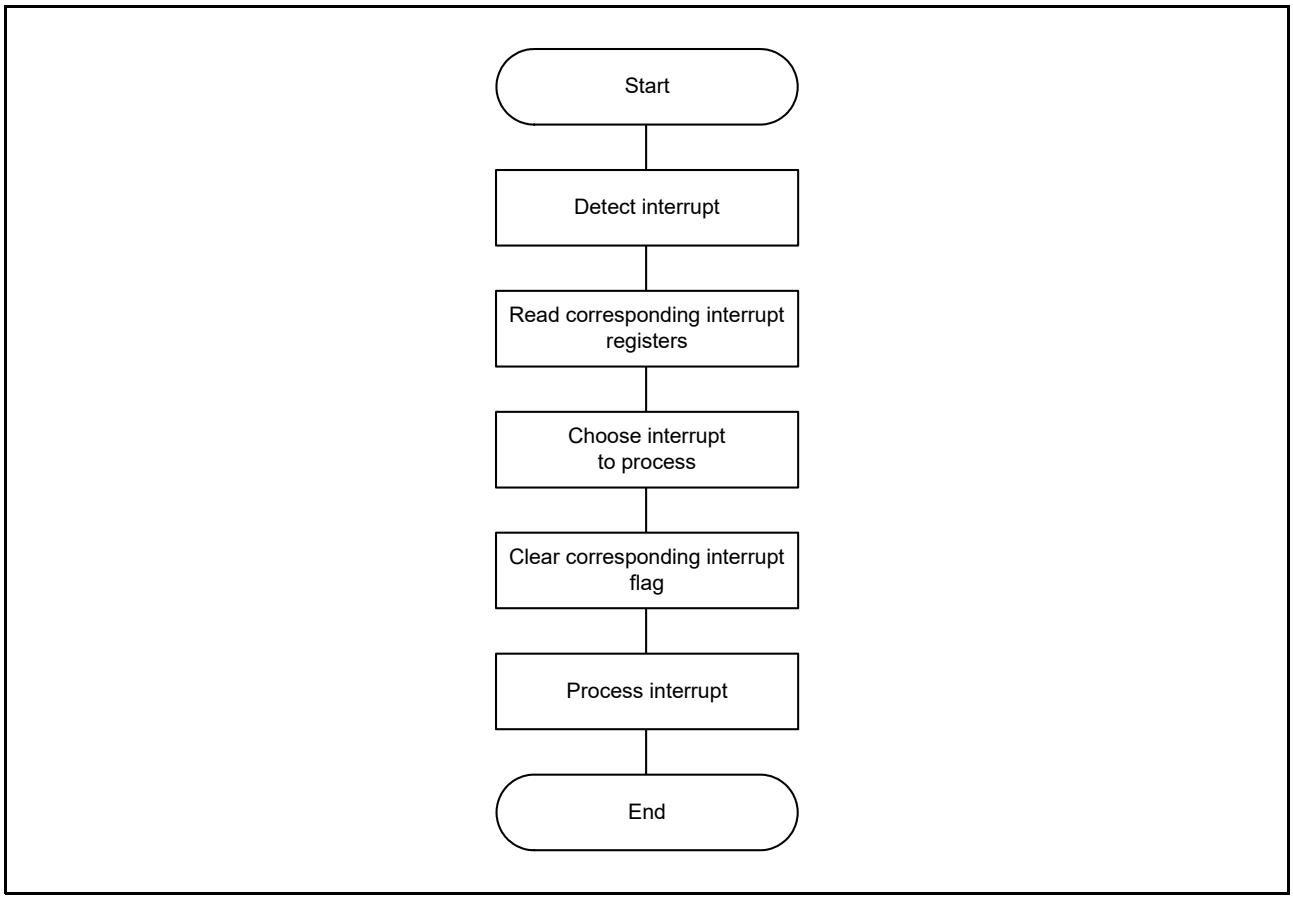


Figure 39.11 Interrupt Handling Flow

39.3.9.2 Timer 0 Flows

In this section are described the flows to set and monitor timer 0.

(1) Timer 0 Offset Setting Flow

The timer 0 offset setting flow is described in Figure 39.12.

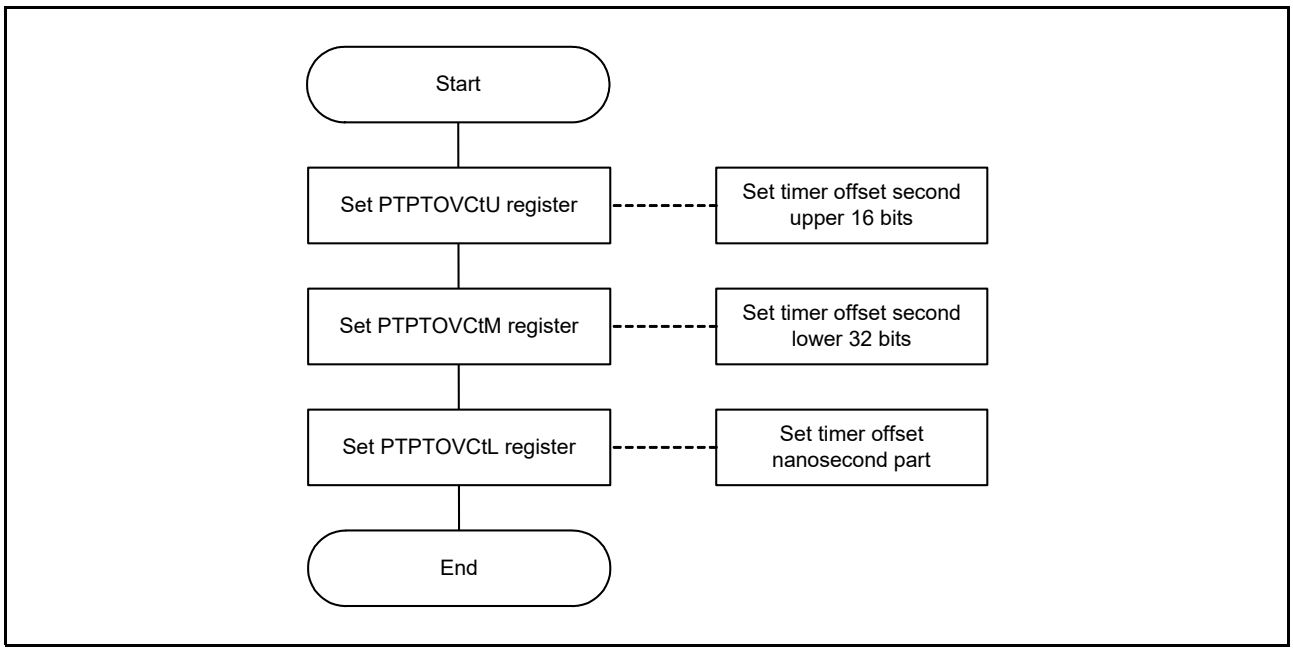


Figure 39.12 Timer 0 Offset Setting Flow

(2) 32-bit AVTP Timer 0 Reading Flow

The 32-bit AVTP timer 0 reading flow is described in Figure 39.13.

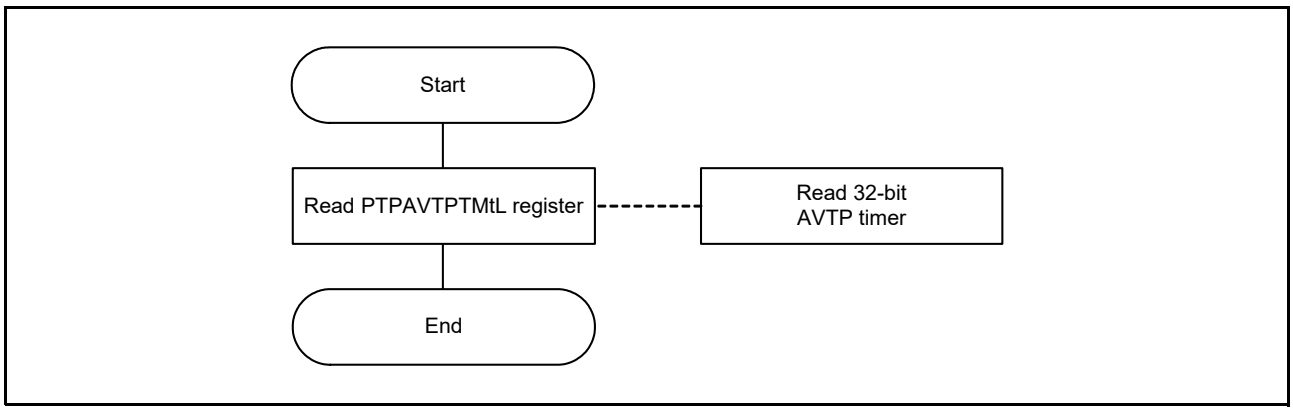


Figure 39.13 32-bit AVTP Timer 0 Reading Flow

(3) 64-bit AVTP Timer 0 Reading Flow

The 64-bit AVTP timer 0 reading flow is described in Figure 39.14.

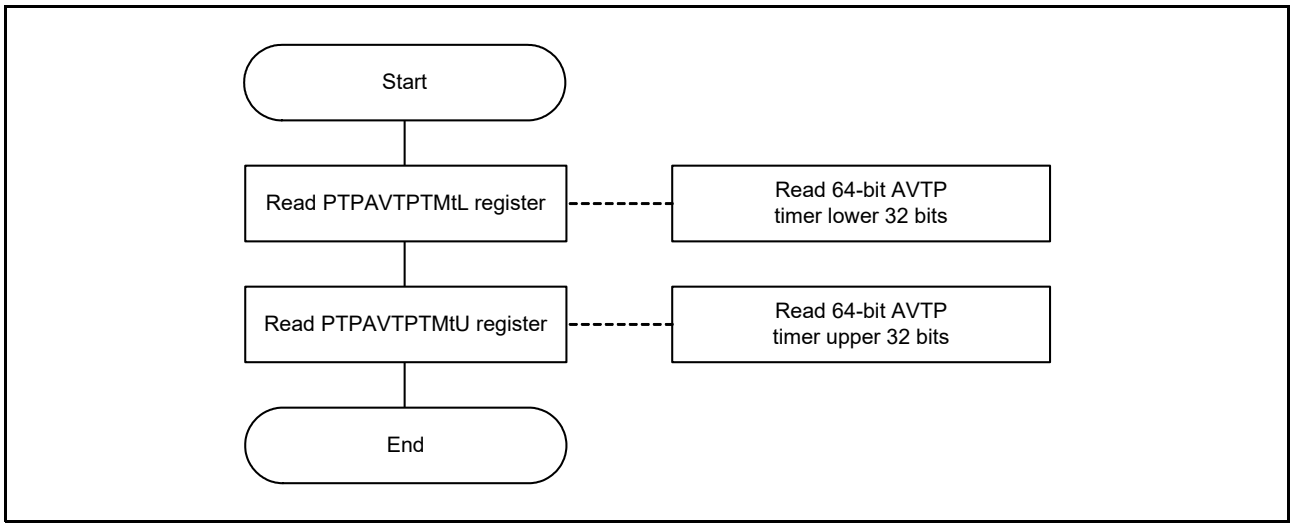


Figure 39.14 64-bit AVTP Timer 0 Reading Flow

(4) gPTP Timer 0 Reading Flow

The gPTP timer 0 reading flow is described in Figure 39.15.

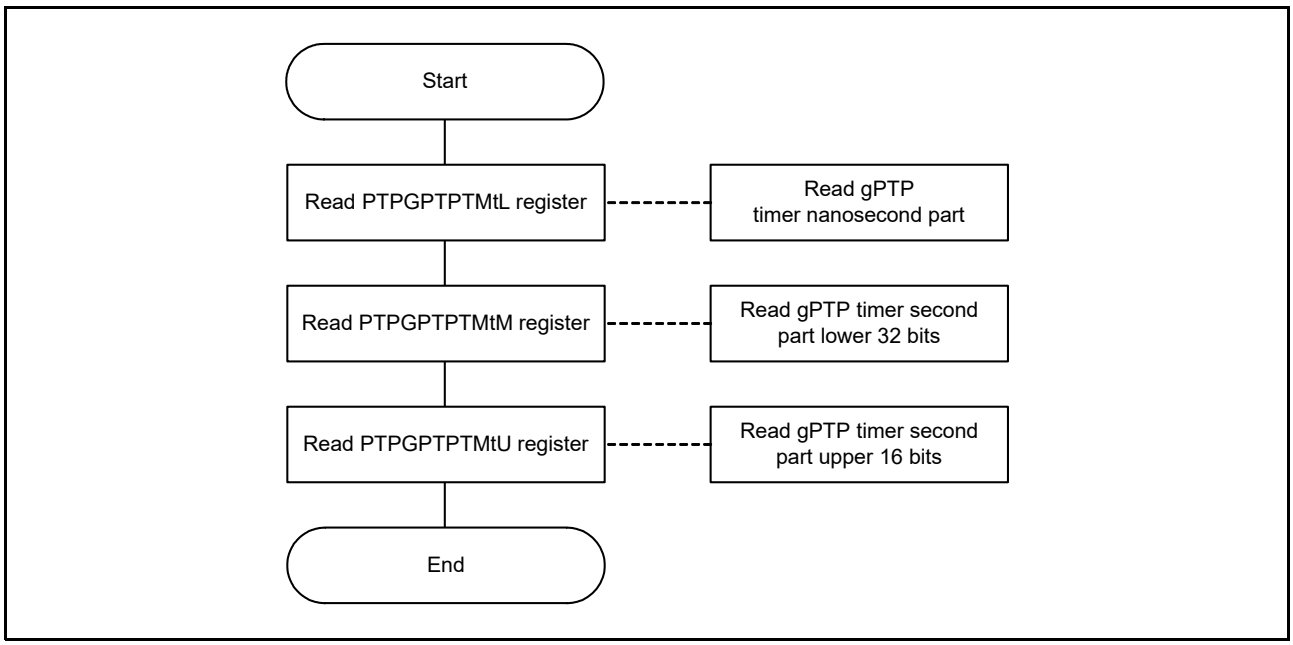


Figure 39.15 gPTP Timer 0 Reading Flow

(5) Register Writable Without Software Flow for Timer 0

Table 39.5 describes the registers writable without software flow and in which mode they can be written.

Table 39.5 Writable without Software Flow Registers

Register name	Writable anytime
PTPTMEC	Writable
PTPTMDC	Writable
PTPTIVC0	Writable

39.3.9.3 Media Clock m Capture Flows (m = 0, 1)

In this section are described the flows to set and monitor media clock m capture.

(1) 32-bit AVTP Timer Captured Value Reading Flow

The 32-bit AVTP timer captured value reading flow is described in Figure 39.16.

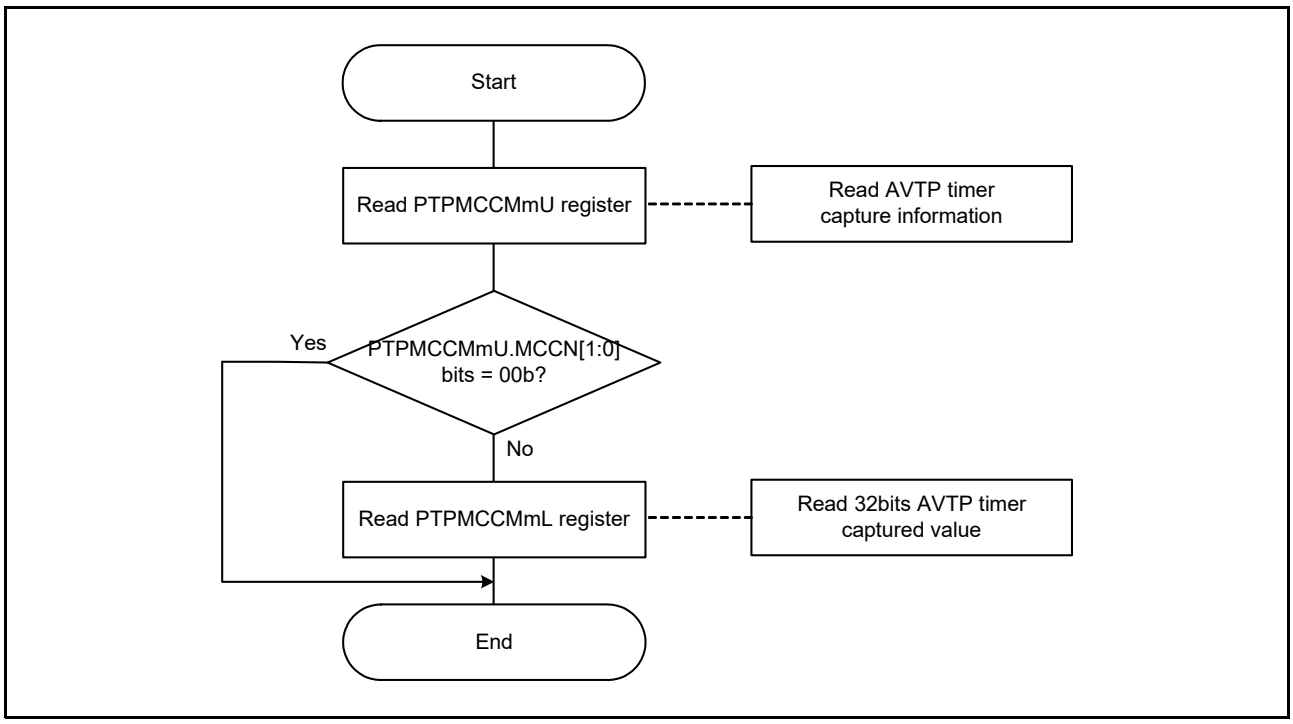


Figure 39.16 32-bit AVTP Timer Captured Value Reading Flow

(2) 64-bit AVTP Timer Captured Value Reading Flow

The 64-bit AVTP timer captured value reading flow is described in Figure 39.17.

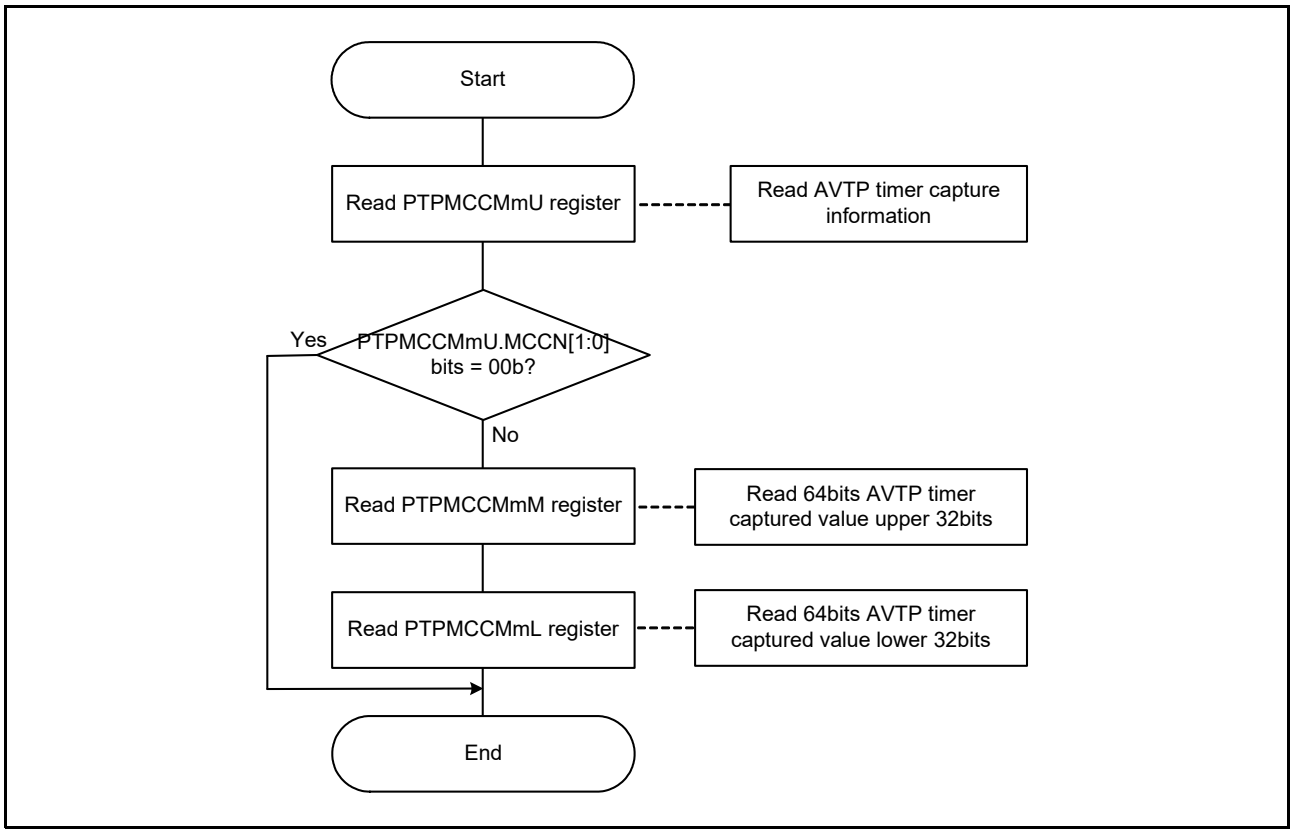


Figure 39.17 64-bit AVTP Timer Captured Value Reading Flow

(3) gPTP Timer Captured Value Reading Flow

The gPTP timer captured value reading flow is described in Figure 39.18.

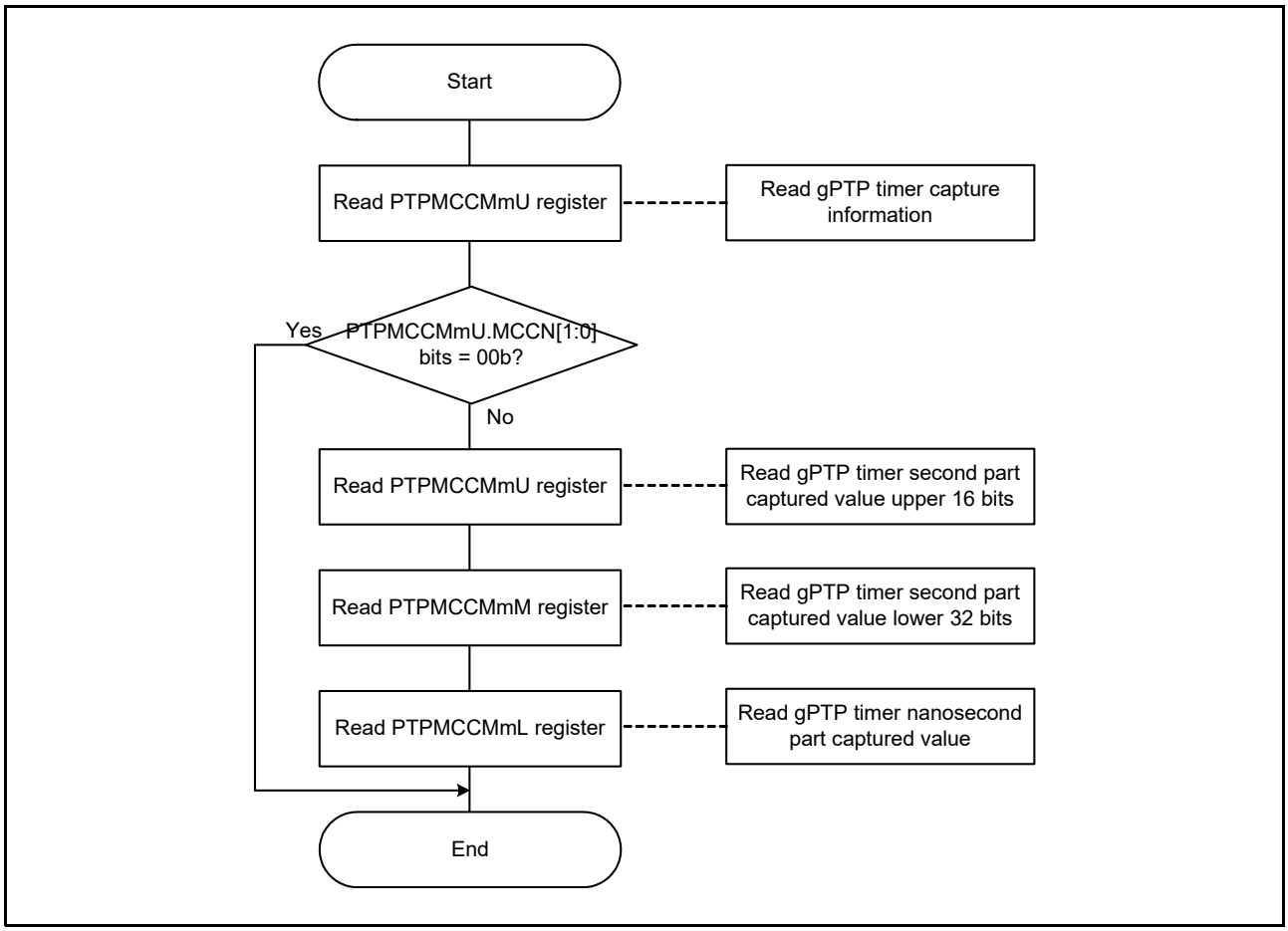


Figure 39.18 gPTP Timer Captured Value Reading Flow

(4) Register Writable Without Software Flow for Media Clock Capture m

Table 39.6 describes the registers writable without software flow and in which mode they can be written.

Table 39.6 Writable without Software Flow Registers

Register name	Writable anytime
PTPMCCCm	Writable

39.3.9.4 Media Clock Recovery m Flows (m = 0, 1)

In this section are described the flows to set and monitor media clock recovery m.

(1) Media Clock Recovery Time Value Setting Flow

The media clock recovery time value setting flow is described in Figure 39.19.

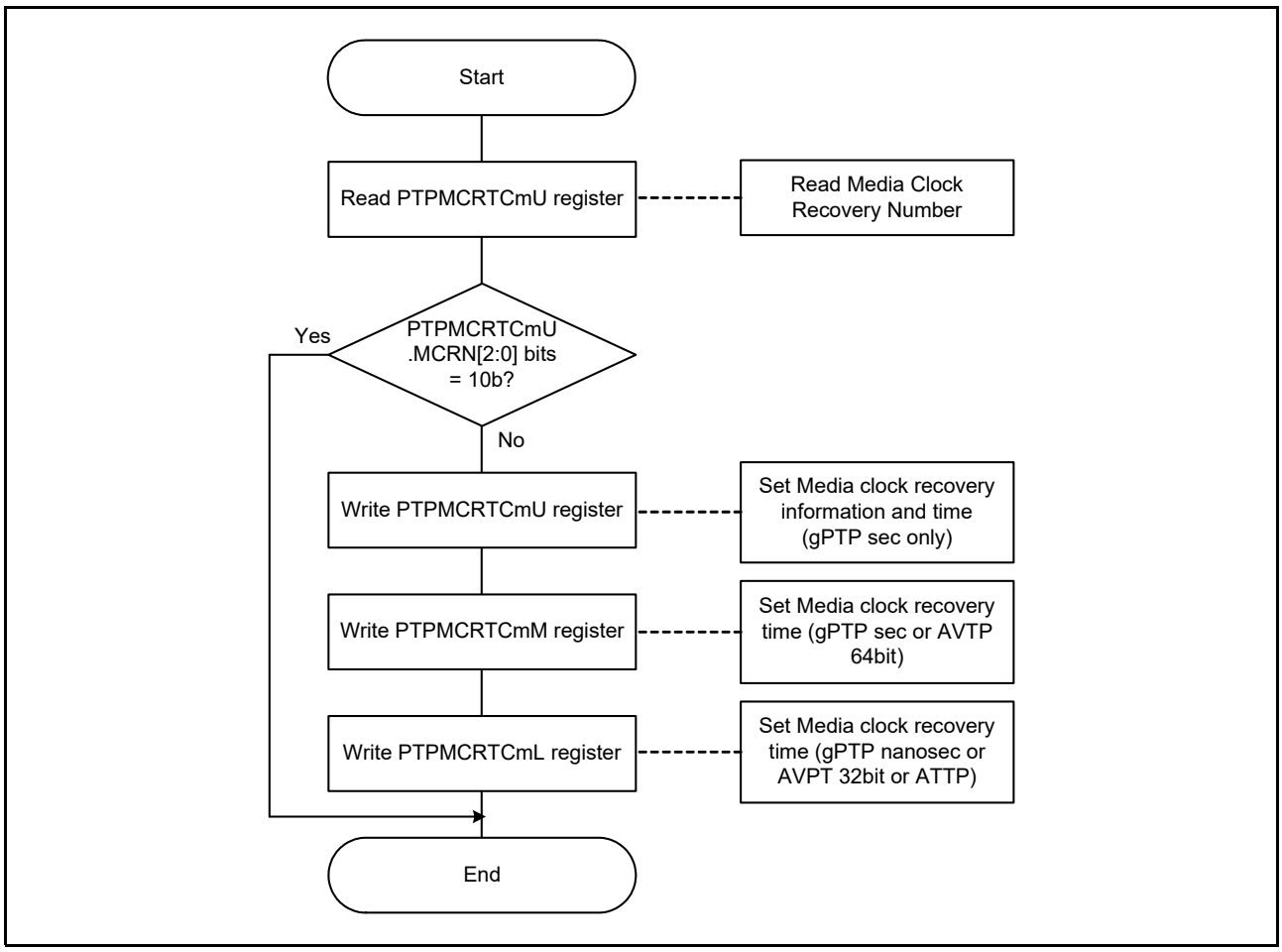


Figure 39.19 Media Clock Recovery Time Value Setting Flow

(2) Media Clock Recovery Buffer Clear Flow

The media clock recovery buffer clear flow is described in Figure 39.20.

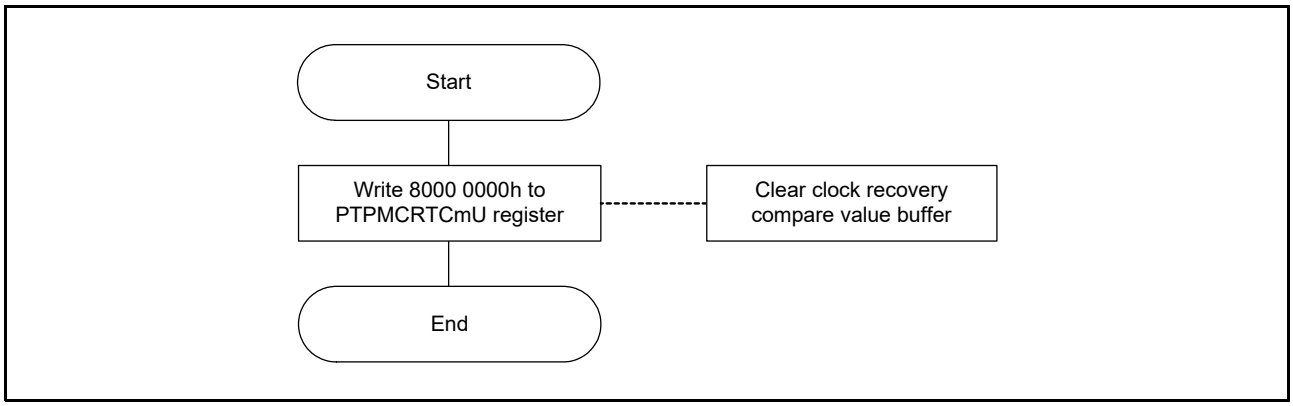


Figure 39.20 Media Clock Recovery Buffer Clear Flow

(3) Register Writable Without Software Flow for Media Clock Recovery m

Table 39.7 describes the registers writable without software flow and in which mode they can be written.

Table 39.7 Writable without Software Flow Registers

Register name	Writable anytime
PTPMCRCm	Writable

39.3.9.5 Cyclic Compare c Flows (c = 0 to 7)

In this section are described the flows to set and monitor cyclic compare c.

(1) Register Writable Without Software Flow for Cyclic Compare c

Table 39.8 describes the registers writable without software flow and in which mode they can be written.

Table 39.8 Writable without Software Flow Registers

Register name	Writable anytime
PTPCCCc0	Writable
PTPCCCc1	Writable

39.3.9.6 Pulse Output Timer n Operation Flow (n = 0, 1)

This section describes the flow of register setting required to operate the pulse output timer n.

(1) Operation Setting Flow

Operation setting flow of the pulse output timer is described in Figure 39.21.

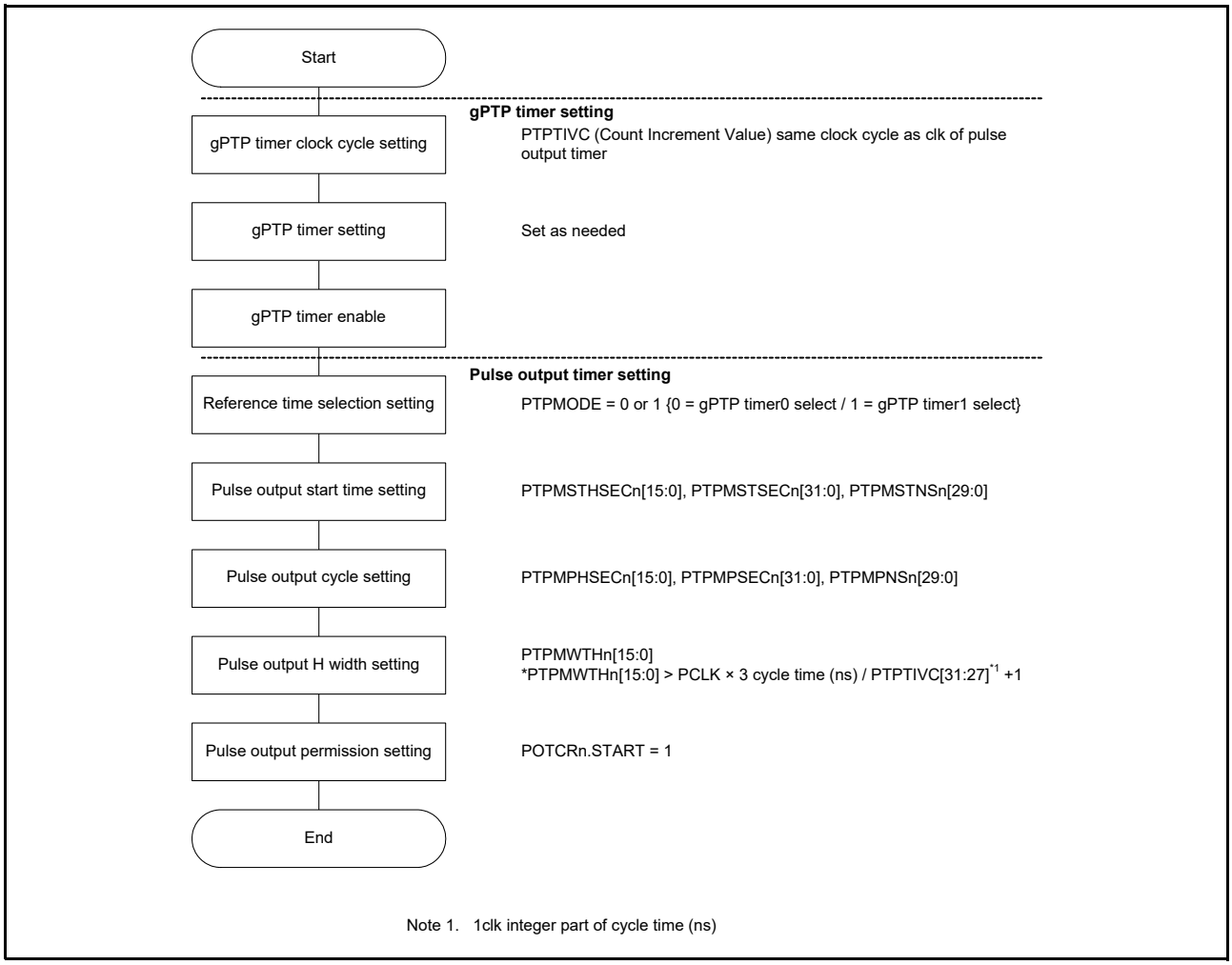


Figure 39.21 Operation Setting Flow

39.4 Usage Notes

39.4.1 Note on Register Setting

The POTCFGR.REFSEL bit switches the gPTP timer selection immediately when the setting is changed regardless of whether it is stopped or operating.
Be careful when changing during operation.

39.4.2 Restrictions on Register Setting

The following registers should be set within the limit range of the set value.

- POTSTRnL, POTPERnL: Set the value in the range of 0 to 3B9AC9FFh.

Note: 3B9ACA00h is 1 second, so if it is 1 second or longer, reflect it in the register in seconds and set it.

- POTPWRn: For settings other than 0, set for a time of 3 or more cycles of the PCLKA cycle.

Note: The time required to generate the interrupt output signal.

40. USB 2.0 Full-Speed Host/Function Module (USBf)

40.1 Overview

This MCU incorporates a USB 2.0 full-speed host/function module (USB0).
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed (only for the host controller) transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.
Table 40.1 shows the specifications of the USB.

Table 40.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected.
	When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Multiple peripheral devices can be connected for communication via a one-stage hub.
	When the function controller is selected: • Full-speed transfer (12 Mbps) is supported*1 • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • Pipes 1 to 9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer, 64-byte single buffer • Pipes 1 and 2: 64-byte double buffer can be specified for bulk transfer and 256-byte double buffer for isochronous transfer • Pipes 3 to 5: Bulk transfer, 64-byte double buffer • Pipes 6 to 9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D–
Low power consumption function	Module-stop state can be set.
Protect Zone filter	Security attribute can be set.*2

Note 1. When the function controller is selected, low-speed transfer (1.5 Mbps) is not supported.
Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 40.1 shows a block diagram of the USB.

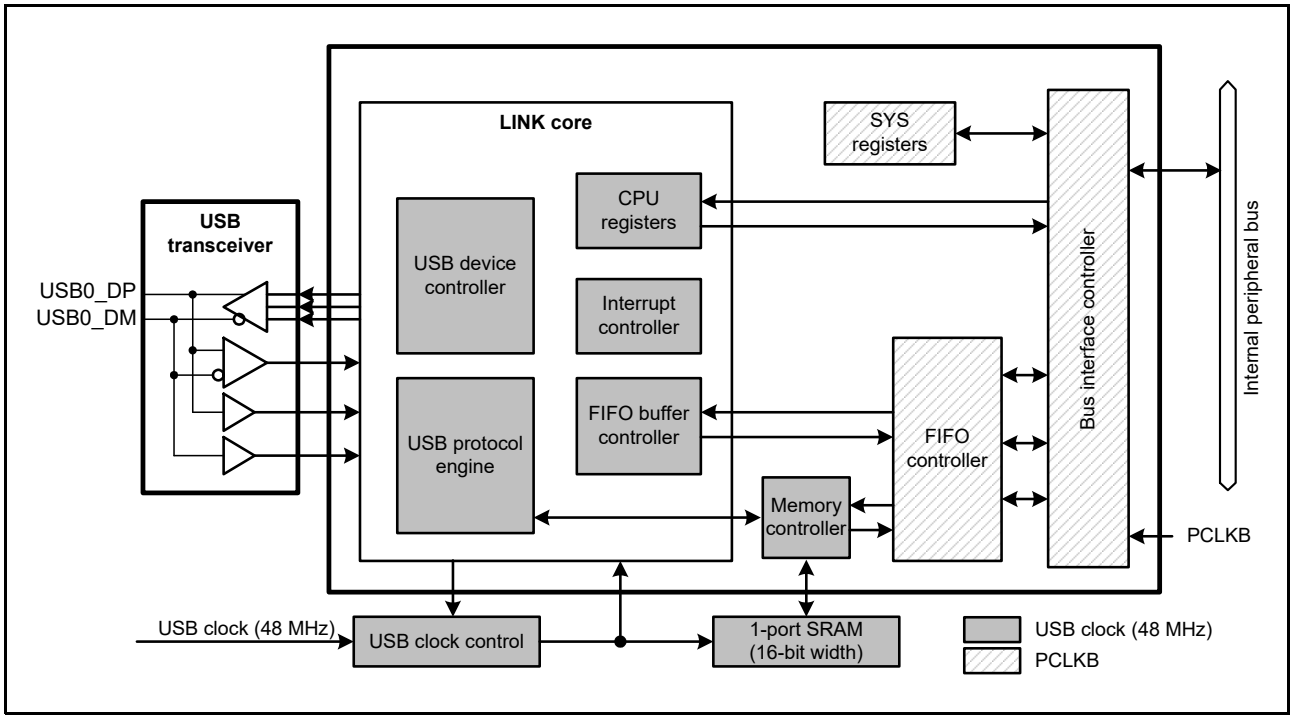


Figure 40.1 USB Block Diagram

Table 40.2 lists the I/O pins of the USB.

Table 40.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

40.2 Register Descriptions

40.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 8785 0000h, USB0_NS.SYSCFG 0785 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*1	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 40.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 40.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBADDR	USBADDR[6:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	—	The value is retained when the host controller is selected.
	USBINDX	—	The value is retained when the host controller is selected.
	USBLENG	—	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DPRPU and DRPD bits are all 0.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

40.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 8785 0004h, USB0_NS.SYSSTS0 0785 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 40.4.	R
b2	IDMON	ID Input Pin Monitor Flag	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b4, b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b5	SOFEA	SOF Active Monitor Flag When the Host Controller is Selected	Indicates the SOF output status. 0: SOF output is stopped. 1: SOF output is operating.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	OVRCURA/OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USB0_OVRCURA pin. The OVCMON[0] flag indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pins.

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 40.4.
The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

SOFEA Flag (SOF Active Monitor Flag When the Host Controller is Selected)

When the USB module is suspended while host mode is used, this bit can be used to confirm whether the last SOF has been output after DVSTCTR0.UACT is set to 0.
Confirm that both the HTACT and SOFEA flags are 0 before stopping the USB module while SYSCFG.USBE is 0 and setting the SYSCFG.SCKE bit to 0 to stop the clock during host mode communication.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.
Confirm that the HTACT flag is 0 before setting this controller to the USB suspended state while DVSTCTR0.UACT bit is 0 and setting the SCKE bit to 0 to stop the clock during host mode communication.

OVCMON[1:0] Flags (OVRCURA/OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 40.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

40.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 8785 0008h, USB0_NS.DVSTCTR0 0785 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection • When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	VBUSEN Output Pin Control	0: USB0_VBUSEN pin outputs low. 1: USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	EXICEN Output Pin Control	0: USB0_EXICEN pin outputs low. 1: USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don’t care

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.
With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state).
With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.
While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).
This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.
The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.
According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.
Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).
This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.
After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

40.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): USB0.CFIFO 8785 0014h, USB0.D0FIFO 8785 0018h, USB0.D1FIFO 8785 001Ch, USB0_NS.CFIFO 0785 0014h, USB0_NS.D0FIFO 0785 0018h, USB0_NS.D1FIFO 0785 001Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): USB0.CFIFO.L 8785 0014h, USB0.D0FIFO.L 8785 0018h, USB0.D1FIFO.L 8785 001Ch, USB0_NS.CFIFO.L 0785 0014h, USB0_NS.D0FIFO.L 0785 0018h, USB0_NS.D1FIFO.L 0785 001Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 40.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the lower byte in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the lower byte in bytes.

Table 40.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

40.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 8785 0020h, USB0_NS.CFIFOSEL 0785 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The CFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The CFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): USB0.D0FIFOSEL 8785 0028h, USB0.D1FIFOSEL 8785 002Ch, USB0_NS.D0FIFOSEL 0785 0028h, USB0_NS.D1FIFOSEL 0785 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DnFIFOCTR.DTLN[8:0] flags are set to 000h when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN[8:0] flag value is set to 000h when all the data has been read from only a single plane.) 1: The DnFIFOCTR.DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data in the selected pipe has been read.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the DnFIFOCTR.DTLN[8:0] flags.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

40.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 8785 0022h, USB0.D0FIFOCTR 8785 002Ah, USB0.D1FIFOCTR 8785 002Eh, USB0_NS.CFIFOCTR 0785 0022h, USB0_NS.D0FIFOCTR 0785 002Ah, USB0_NS.D1FIFOCTR 0785 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.
While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0
  The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- RCNT = 1
  The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)
  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only the FIFO buffer on the CPU side even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

40.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 8785 0030h, USB0_NS.INTENB0 0785 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected. Set these bits to 0 when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

40.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 8785 0032h, USB0_NS.INTENB1 0785 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected. Set these bits to 0 when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

40.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 8785 0036h, USB0_NS.BRDYENB 0785 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

40.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 8785 0038h, USB0_NS.NRDYENB 0785 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPEnNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPEnNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

40.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 8785 003Ah, USB0_NS.BEMPENB 0785 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

40.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 8785 003Ch, USB0_NS.SOFCFG 0785 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

40.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 8785 0040h, USB0_NS.INTSTS0 0785 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag *5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.
Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.
The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag (n = 0 to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 40.3.3.1, BRDY Interrupt.
The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.
The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag (n = 0 to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 40.3.3.2, NRDY Interrupt.
The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.
The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag (n = 0 to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 40.3.3.3, BEMP Interrupt.
The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.
The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected
The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS flag to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

40.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 8785 0042h, USB0_NS.INTSTS1 0785 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status Flag*2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specification 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

OVRCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

40.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 8785 0046h, USB0_NS.BRDYSTS 0785 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status Flag for Pipe 0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for Pipe 1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for Pipe 2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for Pipe 3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for Pipe 4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for Pipe 5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for Pipe 6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for Pipe 7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for Pipe 8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for Pipe 9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

40.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 8785 0048h, USB0_NS.NRDYSTS 0785 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

40.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 8785 004Ah, USB0_NS.BEMPSTS 0785 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

40.2.18 Frame Number Register (FRMNUM)

Address(es): USB0.FRMNUM 8785 004Ch, USB0_NS.FRMNUM 0785 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

40.2.19 Device State Change Register (DVCHGR)

Address(es): USB0.DVCHGR 8785 004Eh, USB0_NS.DVCHGR 0785 004Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DVCHG	Device State Change	0: Disables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits. 1: Enables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits.	R/W

For details, refer to section 40.3.1.5, Release from Deep Software Standby Mode 1 Due to USB Suspend/ Resume Interrupts.

40.2.20 USB Address Register (USBADDR)

Address(es): USB0.USBADDR 8785 0050h, USB0_NS.USBADDR 0785 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	When the function controller is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	• Recovery when the function controller is selected b11 b8 1 0 0 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (Default state) 1 0 1 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (Address state) 1 0 1 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (Configured state) Settings other than above are prohibited. • Recovery when the host controller is selected b11 b8 1 0 0 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

USBADDR[6:0] Bits (USB Address)

On detecting the USB bus reset, the USB sets the USBADDR[6:0] bits to 00h.

The writing to these bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.

When the host controller is selected, the USBADDR[6:0] bits are invalid.

The USBADDR[6:0] bits are initialized when a USB bus reset is detected.

STSRECOV[3:0] Bits (Status Recovery)

These bits are used to resume the state of the internal sequencer on returning from the USB power shut-off. For details, refer to section 40.3.1.5, Release from Deep Software Standby Mode 1 Due to USB Suspend/Resume Interrupts.

The writing to the STSRECOV[3:0] bits is enabled while the DVCHGR.DVCHG bit is set to 1.

40.2.21 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 8785 0054h, USB0_NS.USBREQ 0785 0054h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

40.2.22 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 8785 0056h, USB0_NS.USBVAL 0785 0056h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register.
When the host controller is selected, the value of wValue to be transmitted is set.
The USBVAL register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not modify the value of the USBVAL register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the USBVAL register has no effect.

40.2.23 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 8785 0058h, USB0_NS.USBINDX 0785 0058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The USBINDX register stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
The USBINDX register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not modify the value of the USBINDX register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the USBINDX register has no effect.

40.2.24 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 8785 005Ah, USB0_NS.USBLENG 0785 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The USBLENG register stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
The USBLENG register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not modify the value of the USBLENG register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the USBLENG register has no effect.

40.2.25 DCP Configuration Register (DCPCFG)

Address(es): USB0.DCPCFG 8785 005Ch, USB0_NS.DCPCFG 0785 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

40.2.26 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 8785 005Eh, USB0_NS.DCPMAXP 0785 005Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

40.2.27 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 8785 0060h, USB0_NS.DCPCTR 0785 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 40.3.4.1, Pipe Control Register Switching Procedures.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.
The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

40.2.28 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 8785 0064h, USB0_NS.PIPESEL 0785 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Pipes 1 to 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

40.2.29 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 8785 0068h, USB0_NS.PIPECFG 0785 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means an unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when the selected pipe is pipe 1 to 5.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 40.3.3.1, BRDY Interrupt.

TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

40.2.30 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 8785 006Ch, USB0_NS.PIPEMAXP 0785 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) • Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of pipes 1 and 2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

40.2.31 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 8785 006Eh, USB0_NS.PIPEPERI 0785 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for pipes 3 to 5; set the IITV[2:0] bits to 000b for pipes 3 to 5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

40.2.32 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 8785 0070h, USB0.PIPE2CTR 8785 0072h, USB0.PIPE3CTR 8785 0074h, USB0.PIPE4CTR 8785 0076h, USB0.PIPE5CTR 8785 0078h,
USB0_NS.PIPE1CTR 0785 0070h, USB0_NS.PIPE2CTR 0785 0072h, USB0_NS.PIPE3CTR 0785 0074h,
USB0_NS.PIPE4CTR 0785 0076h, USB0_NS.PIPE5CTR 0785 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Write disabled 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Write disabled 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 40.6 and Table 40.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 0 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 40.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 40.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.
The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 40.9.

Table 40.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 40.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Table 40.8 Information Cleared by USB by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 40.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

- PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 8785 007Ah, USB0.PIPE7CTR 8785 007Ch, USB0.PIPE8CTR 8785 007Eh, USB0.PIPE9CTR 8785 0080h, USB0_NS.PIPE6CTR 0785 007Ah, USB0_NS.PIPE7CTR 0785 007Ch, USB0_NS.PIPE8CTR 0785 007Eh, USB0_NS.PIPE9CTR 0785 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 40.6 and Table 40.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 0 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 40.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 40.9.

Table 40.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

40.2.33 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 8785 0090h, USB0.PIPE2TRE 8785 0094h, USB0.PIPE3TRE 8785 0098h, USB0.PIPE4TRE 8785 009Ch, USB0.PIPE5TRE 8785 00A0h,
USB0_NS.PIPE1TRE 0785 0090h, USB0_NS.PIPE2TRE 0785 0094h, USB0_NS.PIPE3TRE 0785 0098h, USB0_NS.PIPE4TRE 0785 009Ch, USB0_NS.PIPE5TRE 0785 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN register through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN register.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN register.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the PIPEnTRN register and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the PIPEnTRN register before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

40.2.34 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 8785 0092h, USB0.PIPE2TRN 8785 0096h, USB0.PIPE3TRN 8785 009Ah, USB0.PIPE4TRN 8785 009Eh, USB0.PIPE5TRN 8785 00A2h,
USB0_NS.PIPE1TRN 0785 0092h, USB0_NS.PIPE2TRN 0785 0096h, USB0_NS.PIPE3TRN 0785 009Ah,
USB0_NS.PIPE4TRN 0785 009Eh, USB0_NS.PIPE5TRN 0785 00A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The PIPEnTRN register retains the setting by a USB bus reset.

- When written to:
  This register specifies the total of packets (number of transactions) to be received in corresponding pipe.
- When read from:
  This register indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.
  This register indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.

The USB increments the value of the PIPEnTRN register by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the PIPEnTRN register to 0000h when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the PIPEnTRN register to 0000h.
When the transaction counter is not used, set the PIPEnTRN register to 0000h.
Setting the number of transactions to be transferred to the PIPEnTRN register is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

40.2.35 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USB0.DEVADD0 8785 00D0h, USB0.DEVADD1 8785 00D2h, USB0.DEVADD2 8785 00D4h, USB0.DEVADD3 8785 00D6h, USB0.DEVADD4 8785 00D8h, USB0.DEVADD5 8785 00DAh, USB0_NS.DEVADD0 0785 00D0h, USB0_NS.DEVADD1 0785 00D2h, USB0_NS.DEVADD2 0785 00D4h, USB0_NS.DEVADD3 0785 00D6h, USB0_NS.DEVADD4 0785 00D8h, USB0_NS.DEVADD5 0785 00DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for pipes 0 to 9.
When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.
The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.
Set these bits to 10b when a full-speed device is connected via the HUB.
When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.
When the function controller is selected, set these bits to 00b.

40.2.36 Deep Standby USB Transceiver Control/Pin Monitoring Register (DPUSR0R)

Address(es): USB0.DPUSR0R 8785 0400h, USB0_NS.DPUSR0R 0785 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SRPC0	Single End Receiver Control	0: Input through the D+ and D– inputs is disabled. 1: Input through the D+ and D– inputs is enabled.	R/W
b1	RPUE0	D+ Pull-Up Resistor Control*1	0: Disables D+ pull-up resistor. 1: Enables D+ pull-up resistor.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DRPD0	D+/D– Pull-Down Resistor Control *1	0: Disables D+/D– pull-down resistor. 1: Enables D+/D– pull-down resistor.	R/W
b4	FIXPHY0	Transceiver Output Fix	0: The outputs are fixed in normal mode and on return from deep software standby mode 1. 1: The outputs are fixed on transitions to deep software standby mode 1.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DP0	D+ Input Flag	Indicates the D+ input signal.	R
b17	DM0	D– Input Flag	Indicates the D– input signal.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVCA0	OVRCURA Input Flag	Indicates the OVRCURA input signal.	R
b21	DOVCB0	OVRCURB Input Flag	Indicates the OVRCURB input signal.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTS0	VBUS Input Flag	Indicates the VBUS input signal.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use this bit during operation in deep software standby mode 1. For details, refer to section 40.3.1.5, Release from Deep Software Standby Mode 1 Due to USB Suspend/Resume Interrupts.

SRPC0 Bit (Single End Receiver Control)

The SRPC0 bit controls the D+ and D– inputs of the USB transceiver.

This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 Bit (Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

40.2.37 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): USB0.DPUSR1R 8785 0404h, USB0_NS.DPUSR1R 0785 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPINTE0	D+ Interrupt Enable/Clear	0: Recovery from deep software standby mode 1 is disabled. 1: Recovery from deep software standby mode 1 is enabled.	R/W
b1	DMINTE0	D– Interrupt Enable/Clear	0: Recovery from deep software standby mode 1 is disabled. 1: Recovery from deep software standby mode 1 is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DOVRCRAE0	OVRCURA Interrupt Enable/Clear	0: Recovery from deep software standby mode 1 is disabled. 1: Recovery from deep software standby mode 1 is enabled.	R/W
b5	DOVRCRBE0	OVRCURB Interrupt Enable/Clear	0: Recovery from deep software standby mode 1 is disabled. 1: Recovery from deep software standby mode 1 is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	DVBSE0	VBUS Interrupt Enable/ Clear	0: Recovery from deep software standby mode 1 is disabled. 1: Recovery from deep software standby mode 1 is enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DPINT0	D+ Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode 1. 1: The system has returned from deep software standby mode 1.	R
b17	DMINT0	D– Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode 1. 1: The system has returned from deep software standby mode 1.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVRCRA0	OVRCURA Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode 1. 1: The system has returned from deep software standby mode 1.	R
b21	DOVRCRB0	OVRCURB Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode 1. 1: The system has returned from deep software standby mode 1.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBINT0	VBUS Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode 1. 1: The system has returned from deep software standby mode 1.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPINTE0 Bit (D+ Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode 1 by the D+ input.
Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINTE0 Bit (D– Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode 1 by the D– input. Writing 0 to the

DMINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRAE0 Bit (OVRCURA Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode 1 by the OVRCURA input. Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRBE0 Bit (OVRCURB Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode 1 by the OVRCURB input. Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBSE0 Bit (VBUS Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode 1 by the VBUS input. Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

DPINT0 Flag (D+ Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode 1 due to the D+ input. Recovery from deep software standby mode 1 due to D+ input is only enabled when the DPINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINT0 Flag (D– Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode 1 due to the D– input. Recovery from deep software standby mode 1 due to D– input is only enabled when the DMINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRA0 Flag (OVRCURA Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode 1 due to the OVRCURA input. Recovery from deep software standby mode 1 due to OVRCURA input is only enabled when the DOVRCRAE0 bit is 1.
Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRB0 Flag (OVRCURB Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode 1 due to the OVRCURB input. Recovery from deep software standby mode 1 due to OVRCURB input is only enabled when the DOVRCRBE0 bit is 1.
Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBINT0 Flag (VBUS Interrupt Source Recovery Flag)

This indicates that the system has returned from deep software standby mode 1 due to the VBUS input. Recovery from deep software standby mode 1 due to VBUS input is only enabled when the DVBSE0 bit is 1.
Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

40.3 Operation

40.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

40.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

40.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pull-up resistor of the D+line and pull-down resistors of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0 and DRPD bit = 0).

40.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the DPRPU bit to 1 and pull up the D+ line (during full-speed).
When the DPRPU bit is set to 0 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the DRPD bit and pull down the D+ and D– lines.

Table 40.11 USB Data Bus Resistor Control

SYSCFG register				
DRPD bit	**DPRPU bit**	**D–**	**D+**	**Function**
0	0	Open	Open	Not in use
0	1	Open	Pull-up	When operating as the function controller (in full-speed)
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Setting prohibited

40.3.1.4 Example of USB External Connection Circuit

Figure 40.2 shows an example of connection between OTG device and USB connector in the self-powered state.
The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected and the DPRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

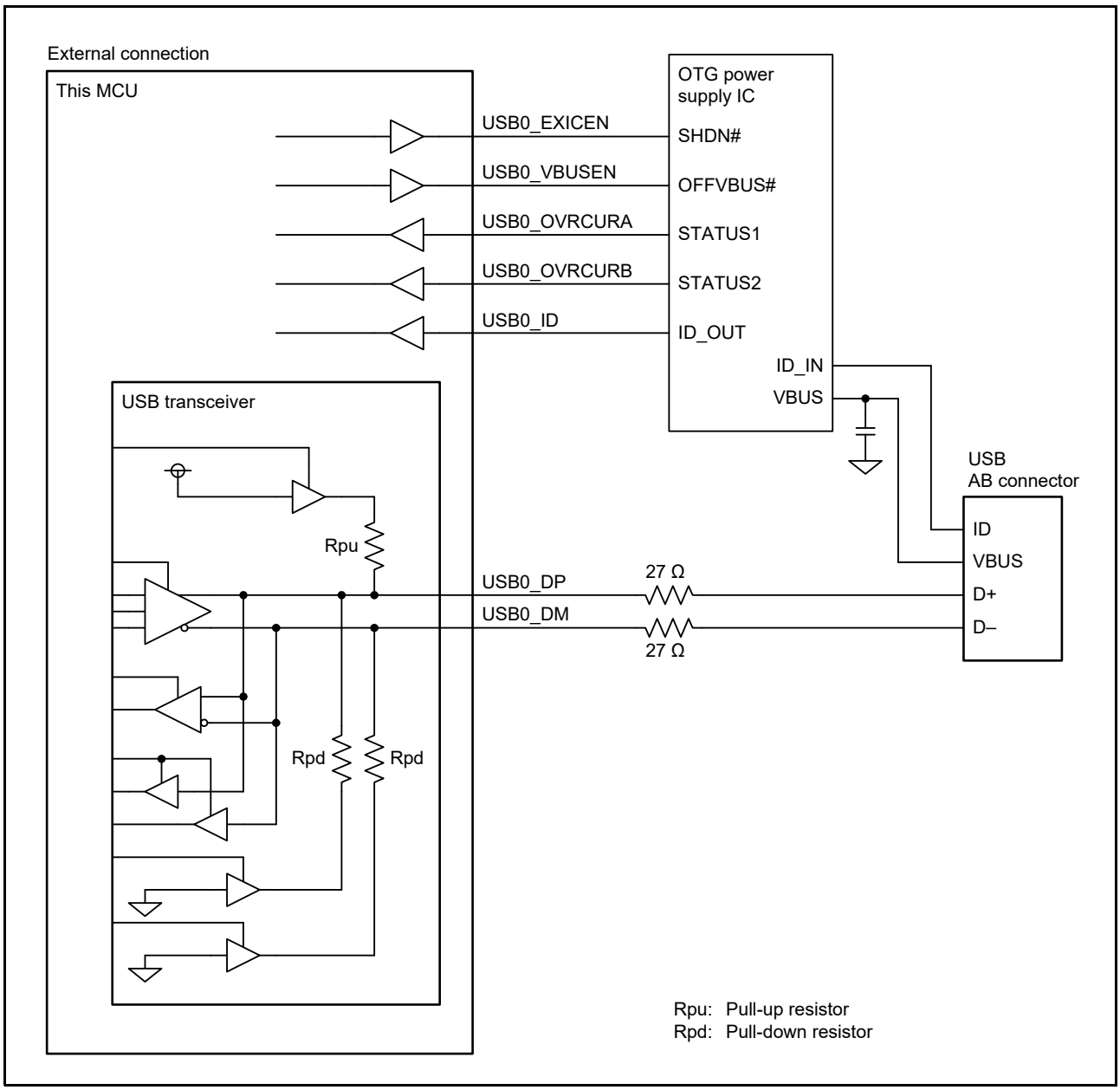


Figure 40.2 Example of Connection between Self-Powered OTG Device and USB Connector

Figure 40.3 shows an example of connection between function controller and USB connector in self-powered state.

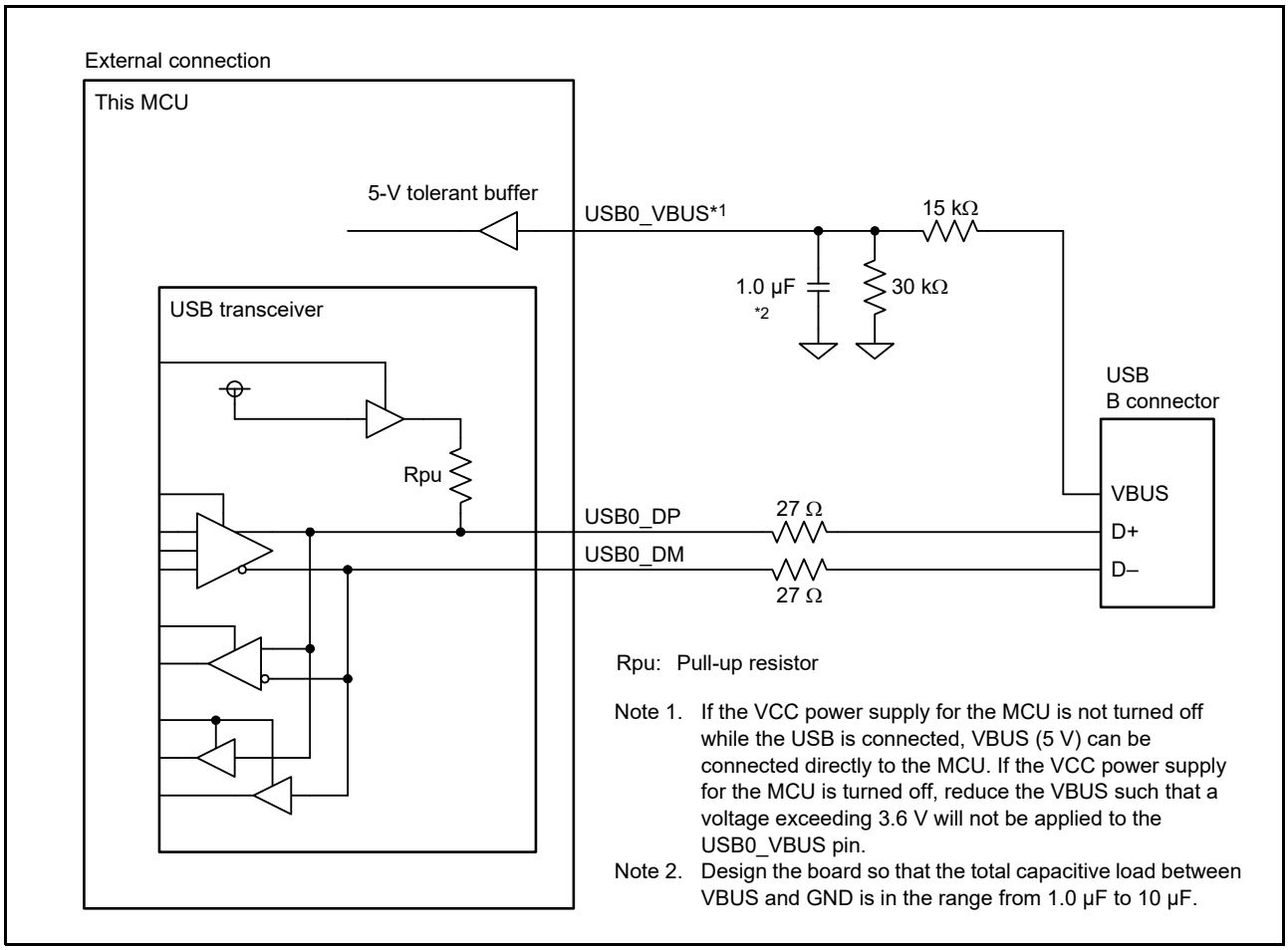


Figure 40.3 Example of Connection between Self-Powered Function and USB Connector

Figure 40.4 shows an example of connection between host controller and USB connector.

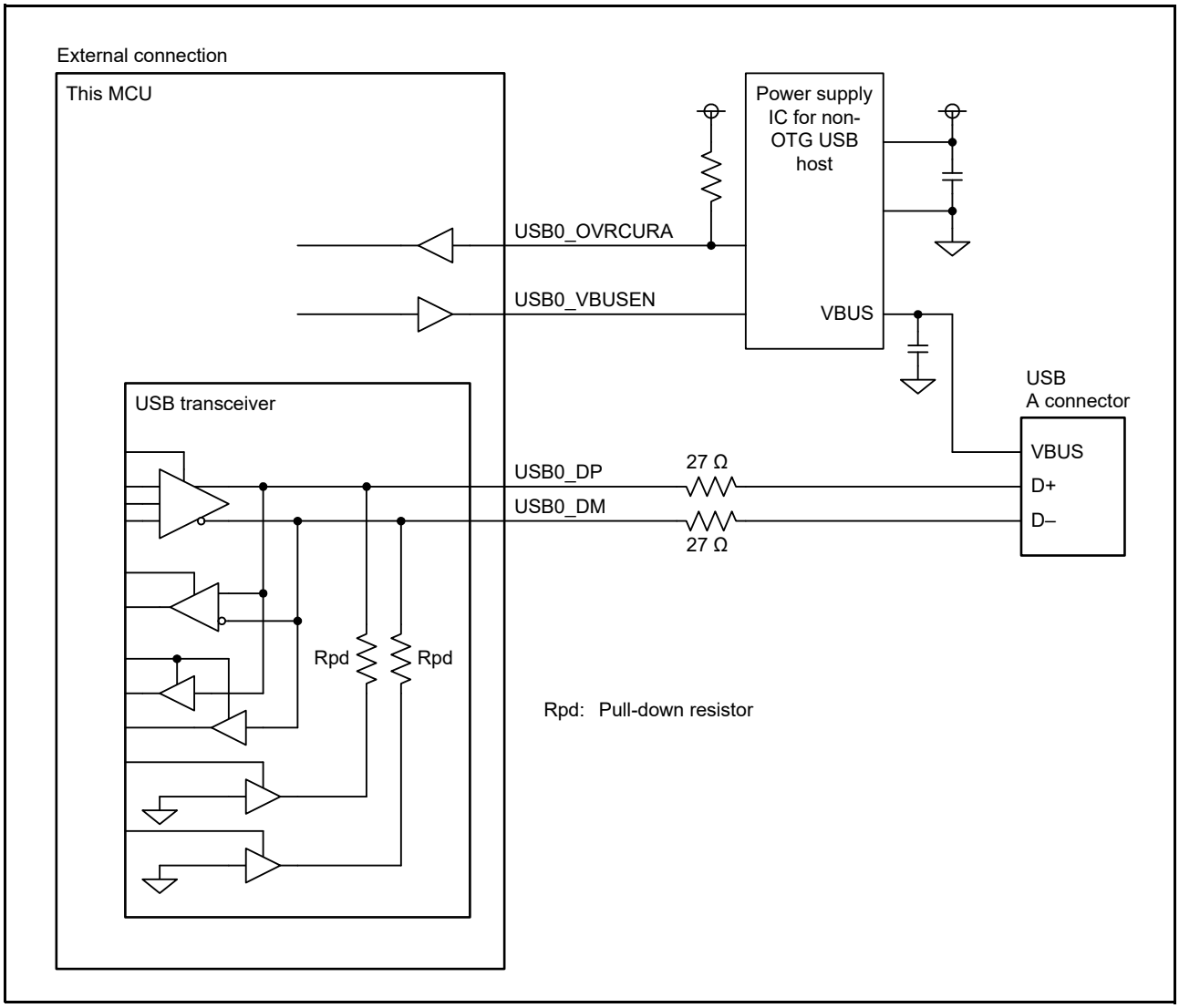


Figure 40.4 Example of Connection between Host and USB Connector

Figure 40.5 shows an example of connection between function controller and USB connector in bus powered state.

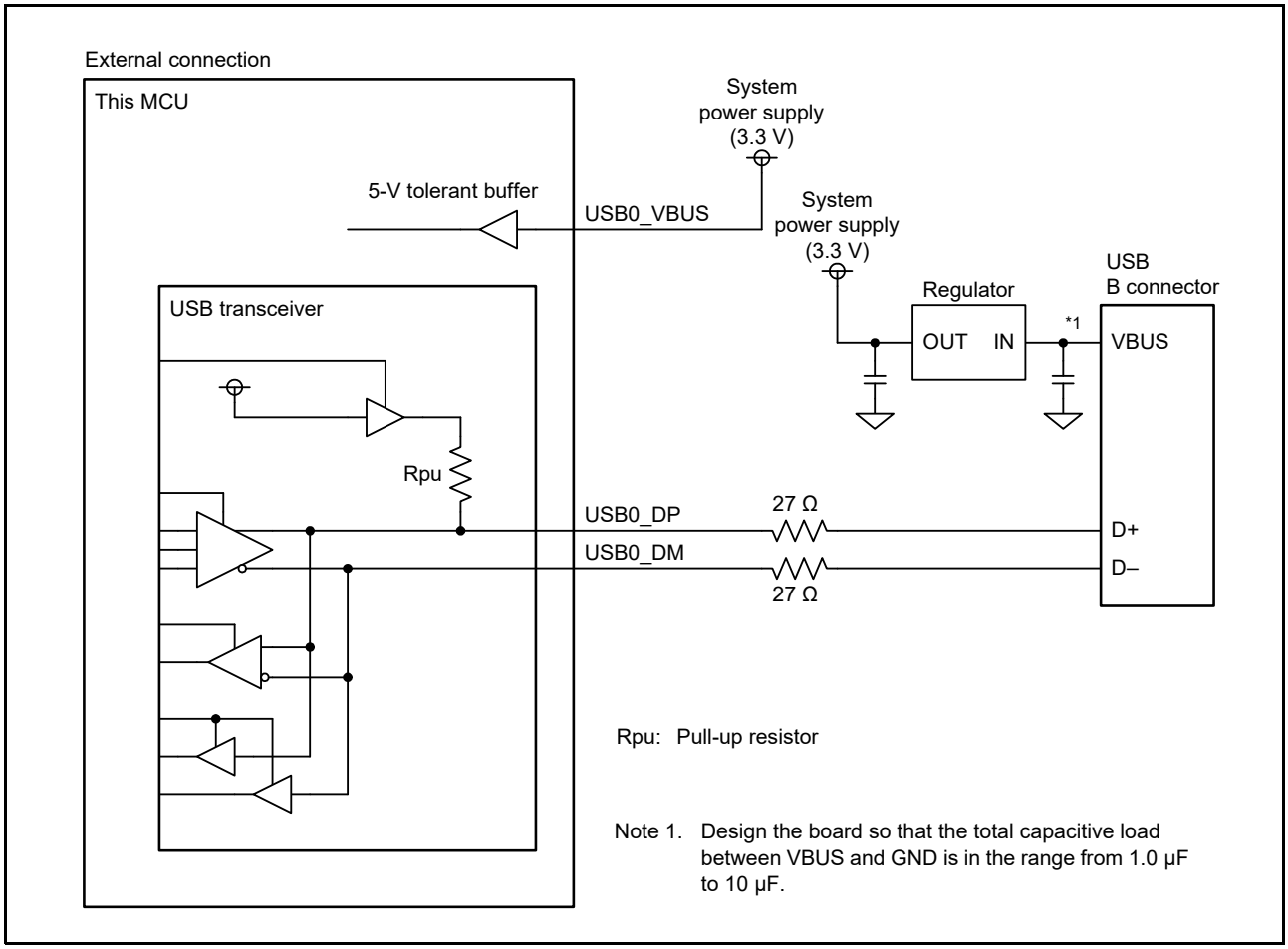


Figure 40.5 Example of Connection between Bus-Powered Function and USB Connector

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

40.3.1.5 Release from Deep Software Standby Mode 1 Due to USB Suspend/Resume Interrupts

Deep software standby mode 1 can be canceled by a USB suspend/resume interrupt.
A USB suspend/resume interrupt is detected at the USB resume detecting unit. The USB resume detecting unit controls and monitors the I/O pins for USB to detect USB suspend/resume interrupts.
Figure 40.6 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB0.

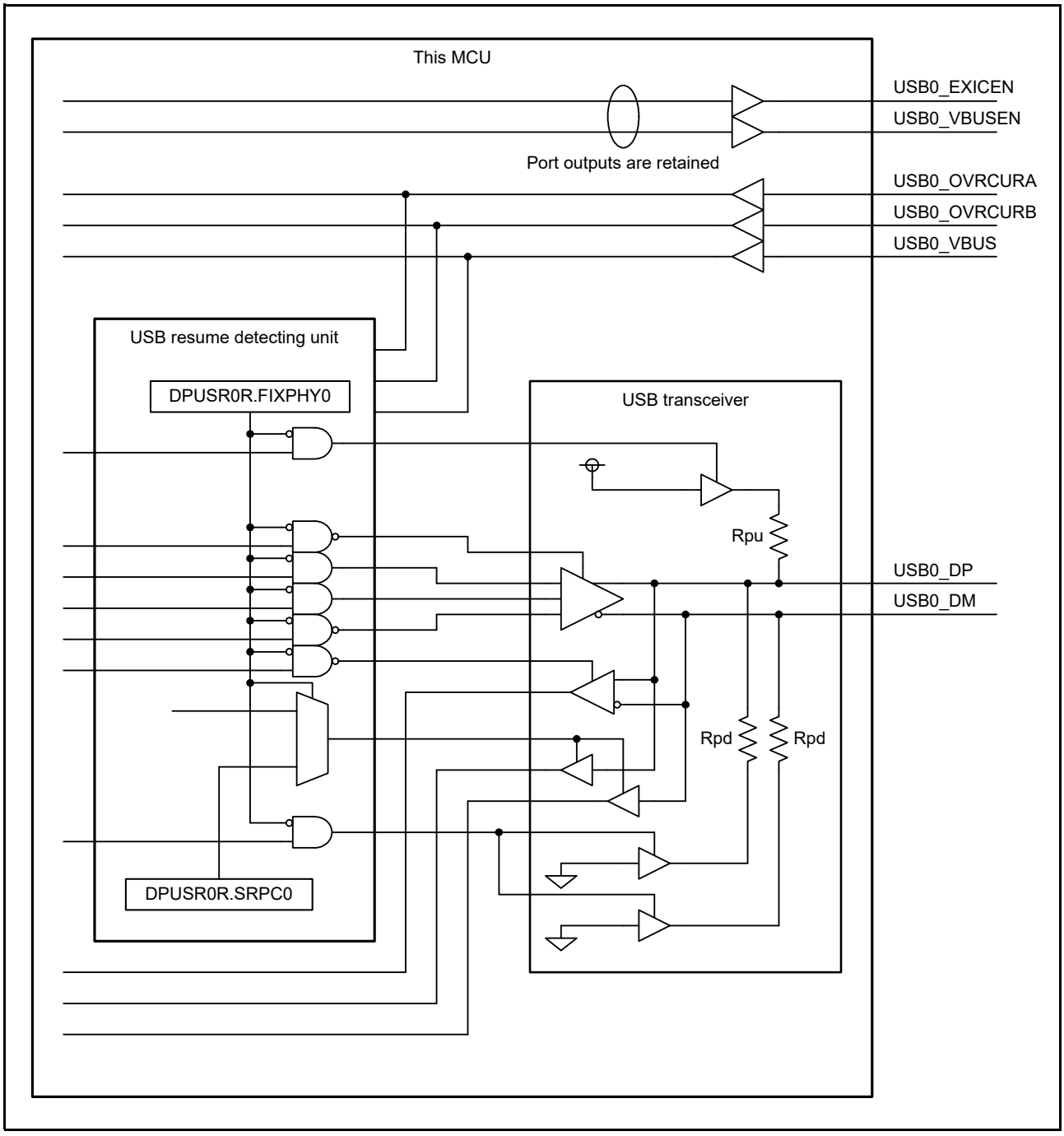


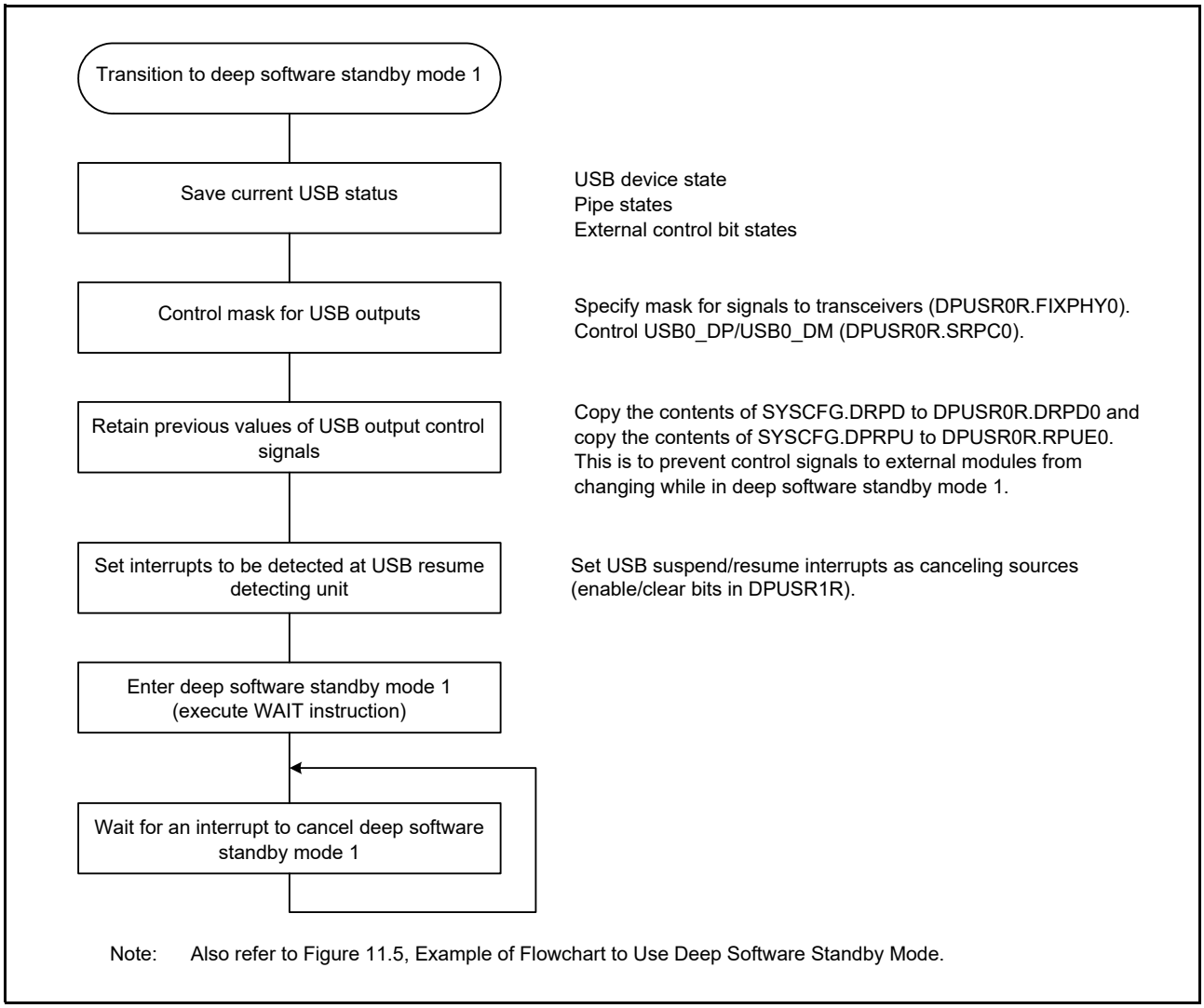
Figure 40.6 Overview of Connection between USB Resume Detecting Unit and USB I/O Pins

Table 40.12 shows the USB suspend/resume interrupt sources and their corresponding I/O pins.

Table 40.12 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins

USB Operating Mode	Source	Pin Name
Function/OTG	Resume	USB0_DP
Host/OTG	Connection/Disconnection	USB0_DP/USB0_DM
Function	Connection/Disconnection	USB0_VBUS
Host	Overcurrent detection	USB0_OVRCURA
OTG	Overcurrent detection	USB0_OVRCURA/USB0_OVRCURB

Figure 40.7 shows a flowchart for setting the USB when entering deep software standby mode 1 while the host controller or function controller is selected. Figure 40.8 shows a flowchart for setting the USB when canceling deep software standby mode 1 while the host controller is selected. Figure 40.9 shows a flowchart for setting the USB when canceling deep software standby mode 1 while the function controller is selected.



Note: Also refer to Figure 11.5, Example of Flowchart to Use Deep Software Standby Mode.

Figure 40.7 USB Setting Flowchart for Transition to Deep Software Standby Mode 1 as Host/Function Controller

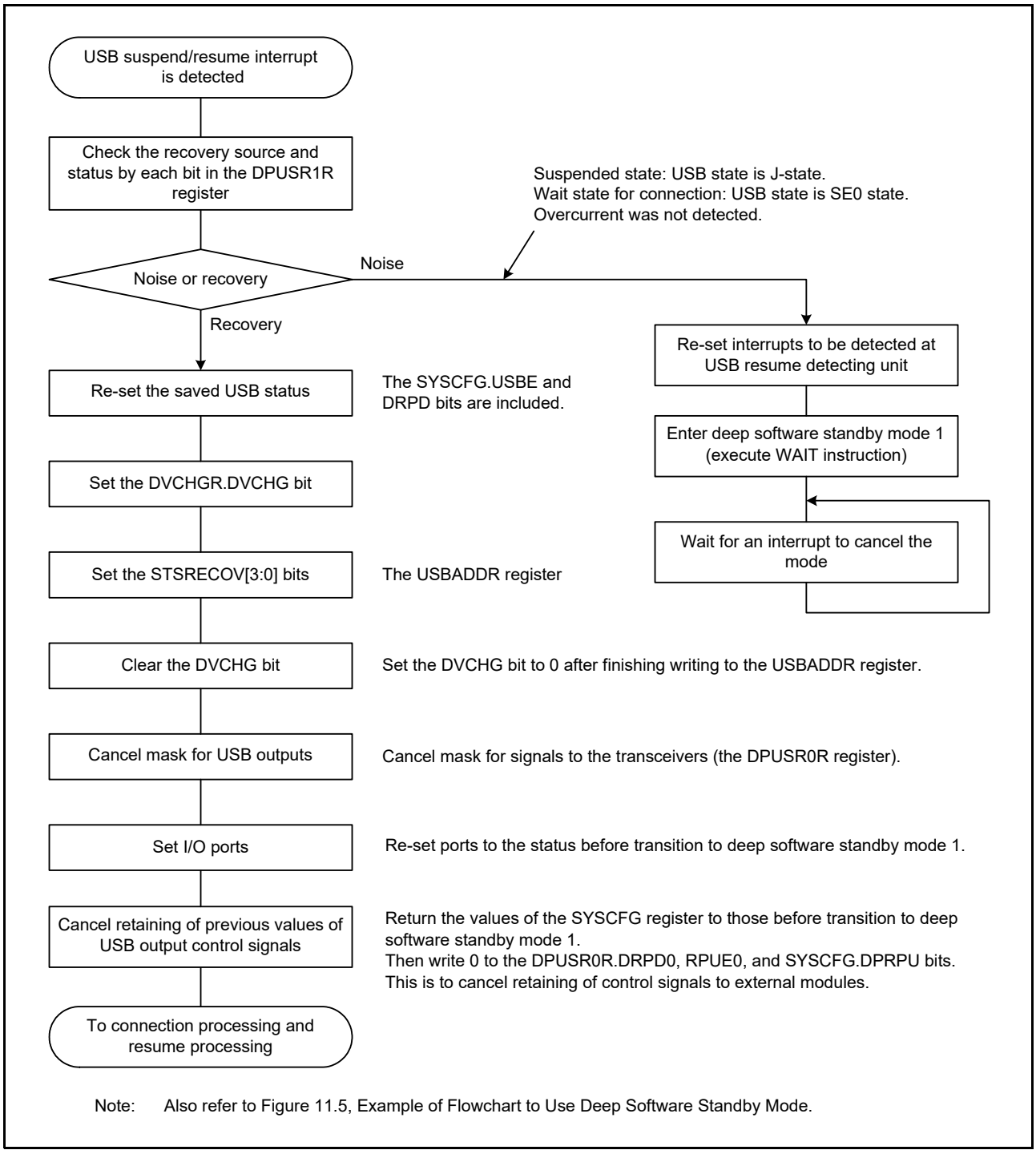


Figure 40.8 USB Setting Flowchart for Canceling Deep Software Standby Mode 1 as Host Controller

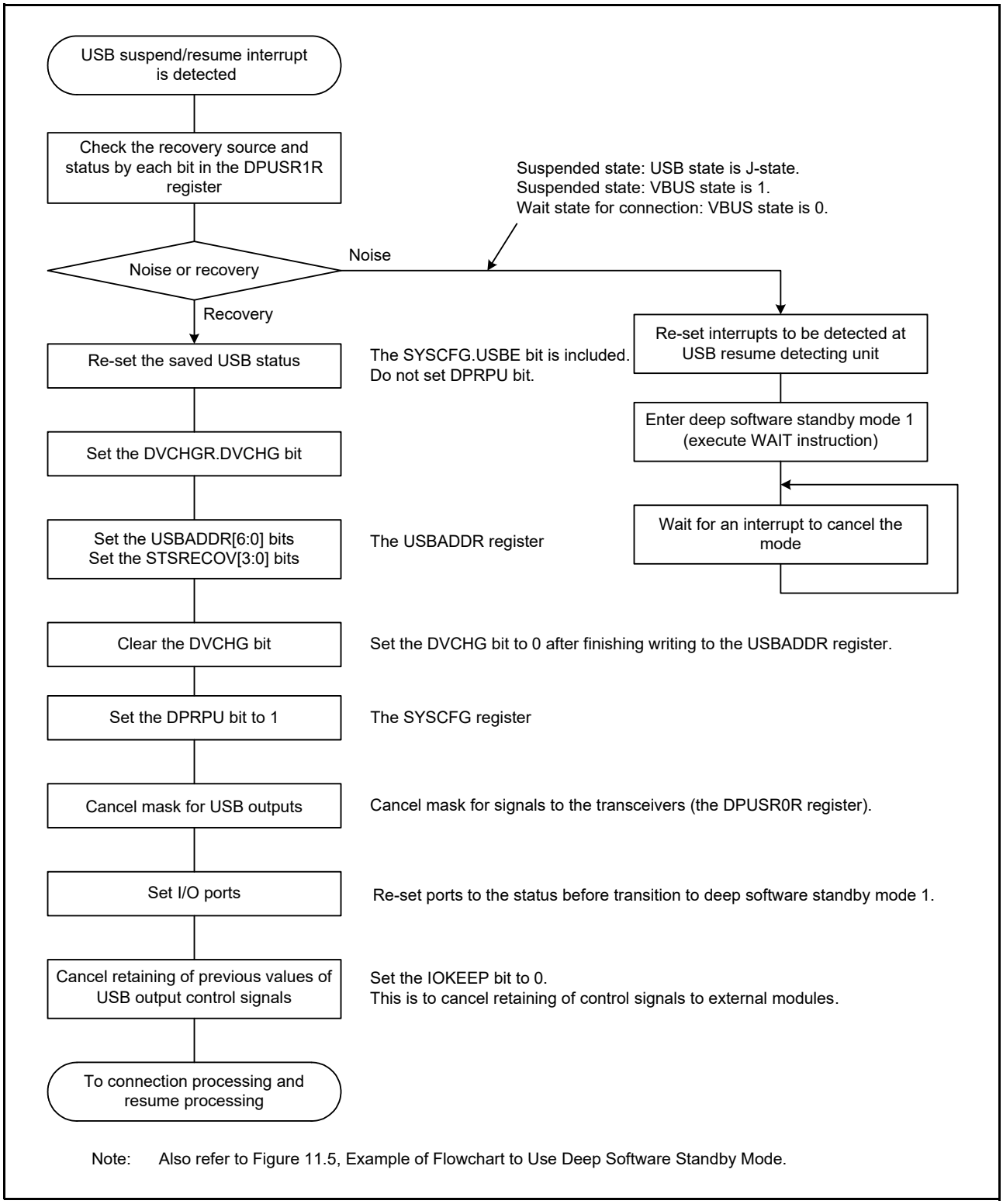


Figure 40.9 USB Setting Flowchart for Canceling Deep Software Standby Mode 1 as Function Controller

40.3.2 Interrupt Sources

Table 40.13 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 40.13 Interrupt Sources

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	• When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)	Host/function*1	INTSTS0. VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller is selected] • When an SOF packet with a different frame number has been transmitted [Function controller is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspended state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) • When an overrun/underrun occurred during isochronous transfer [Function controller is selected] • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS. PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1. OVRCR
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 40.10 shows the circuits related to the interrupts in the USB.

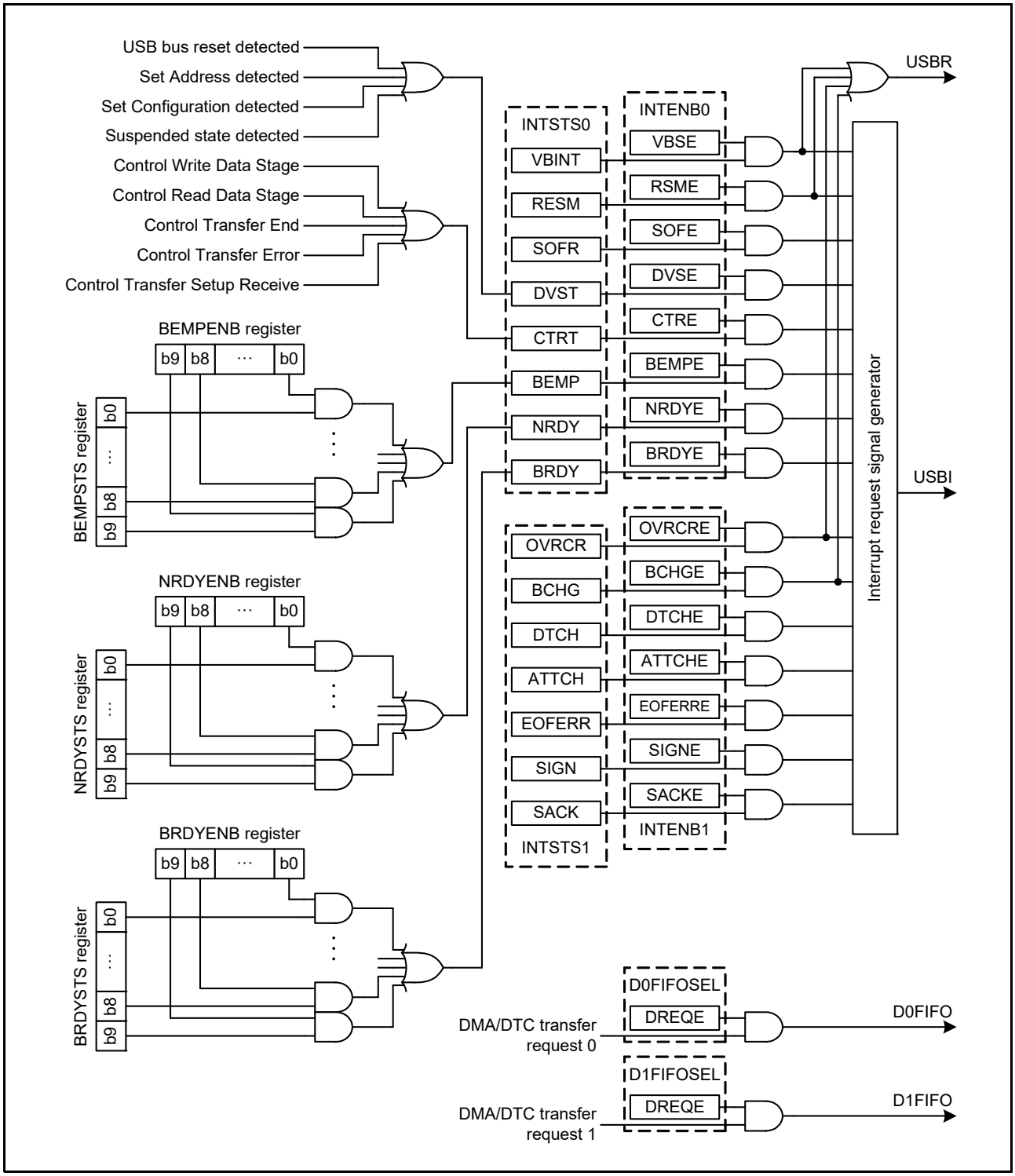


Figure 40.10 Circuits Related to Interrupts in USB

Table 40.14 shows the interrupts generated in the USB0.

Table 40.14 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation
D0FIFO	DMA/DTC transfer request 0	Possible	Possible
D1FIFO	DMA/DTC transfer request 1	Possible	Possible
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, and setup error	Not possible	Not possible
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible

40.3.3 Interrupt Descriptions

40.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
  No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the pipe n transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN register are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY flag cannot be set to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 40.11 shows the timing of BRDY interrupt generation.

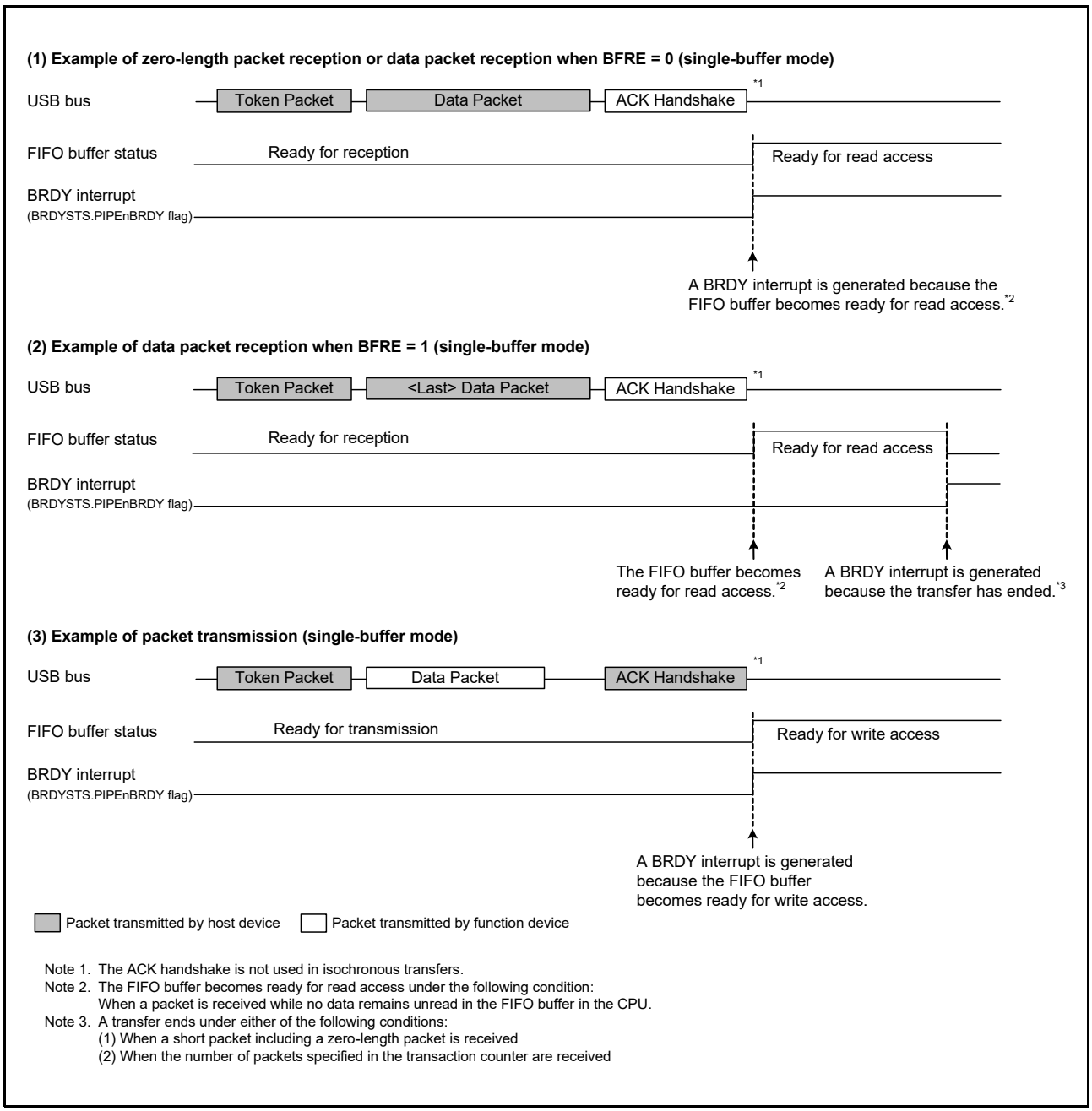


Figure 40.11 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 40.15 shows the condition for clearing the BRDY flag.

Table 40.15 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

40.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPEnNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY flag corresponding to the pipe and the OVRN flag to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.
  However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Figure 40.12 shows the timing of NRDY interrupt generation when the function controller is selected.

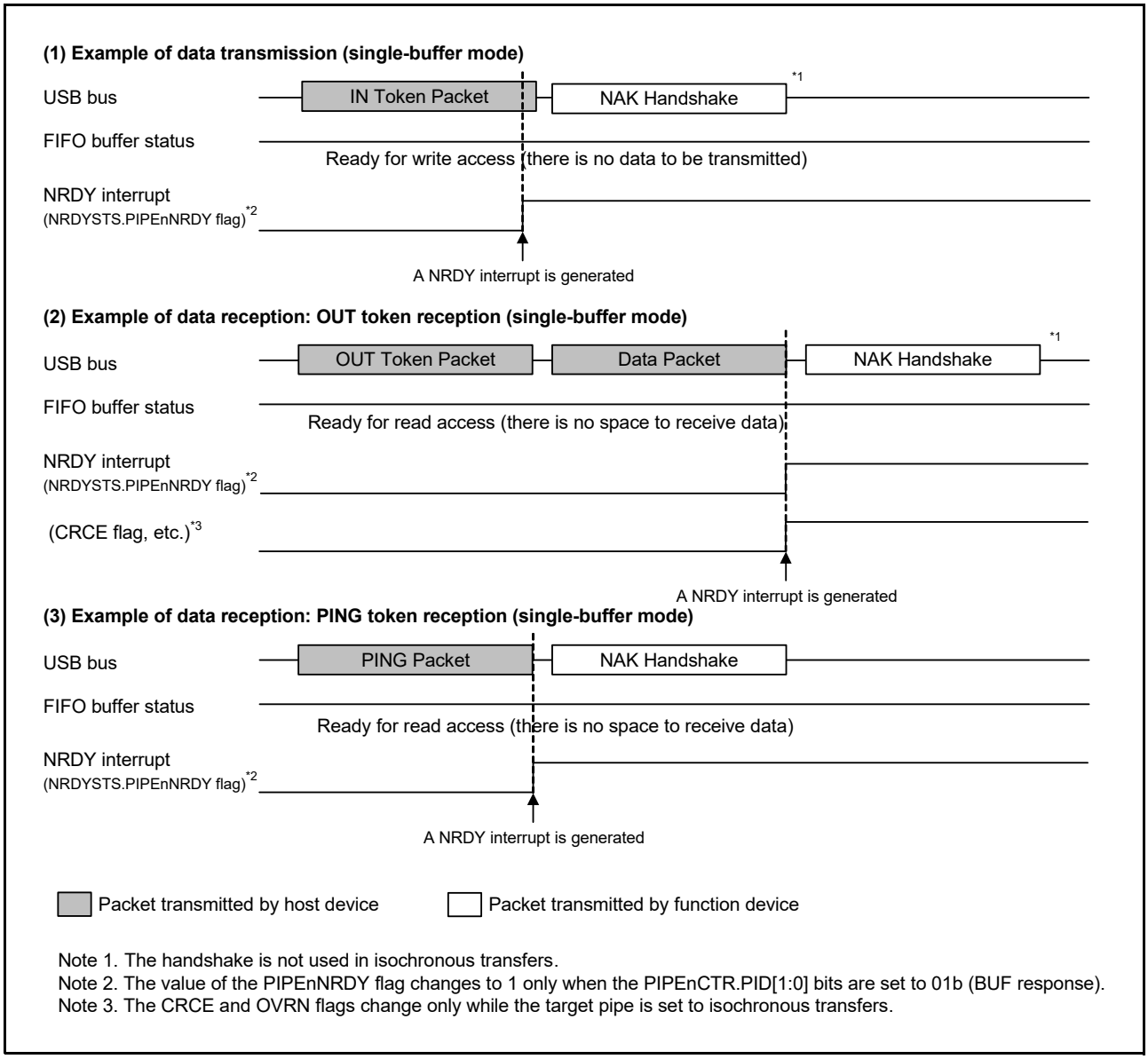


Figure 40.12 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

40.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 40.13 shows the timing of BEMP interrupt generation when the function controller is selected.

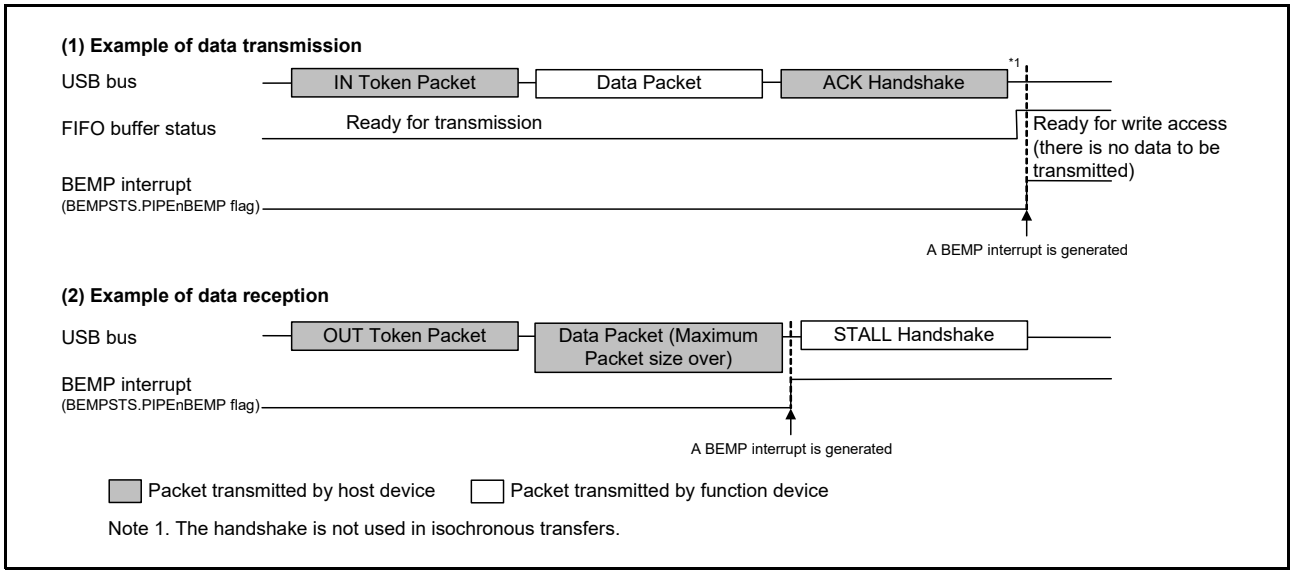


Figure 40.13 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

40.3.3.4 Device State Transition Interrupt

Figure 40.14 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.

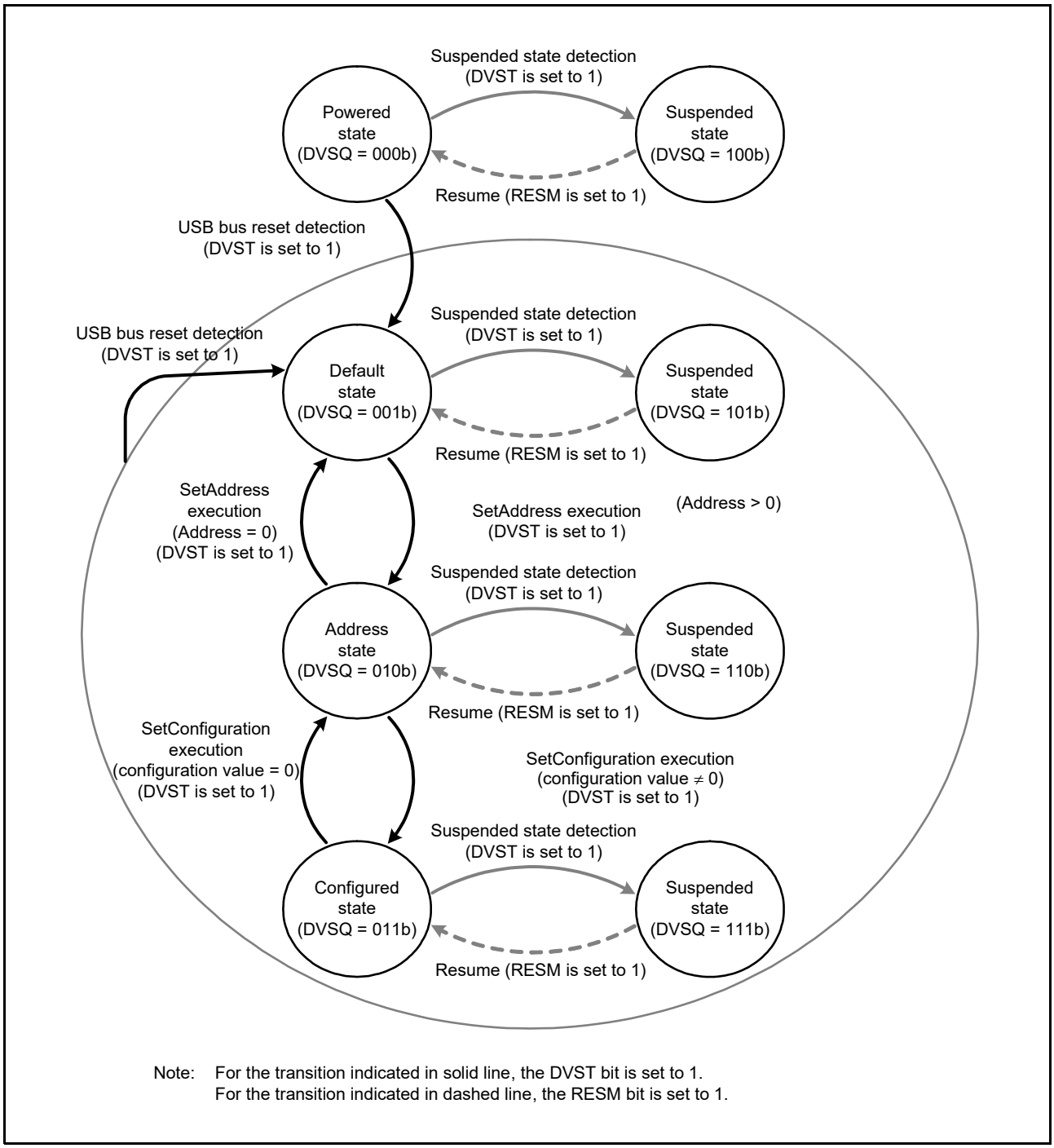


Figure 40.14 Device State Transitions

40.3.3.5 Control Transfer Stage Transition Interrupt

Figure 40.15 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

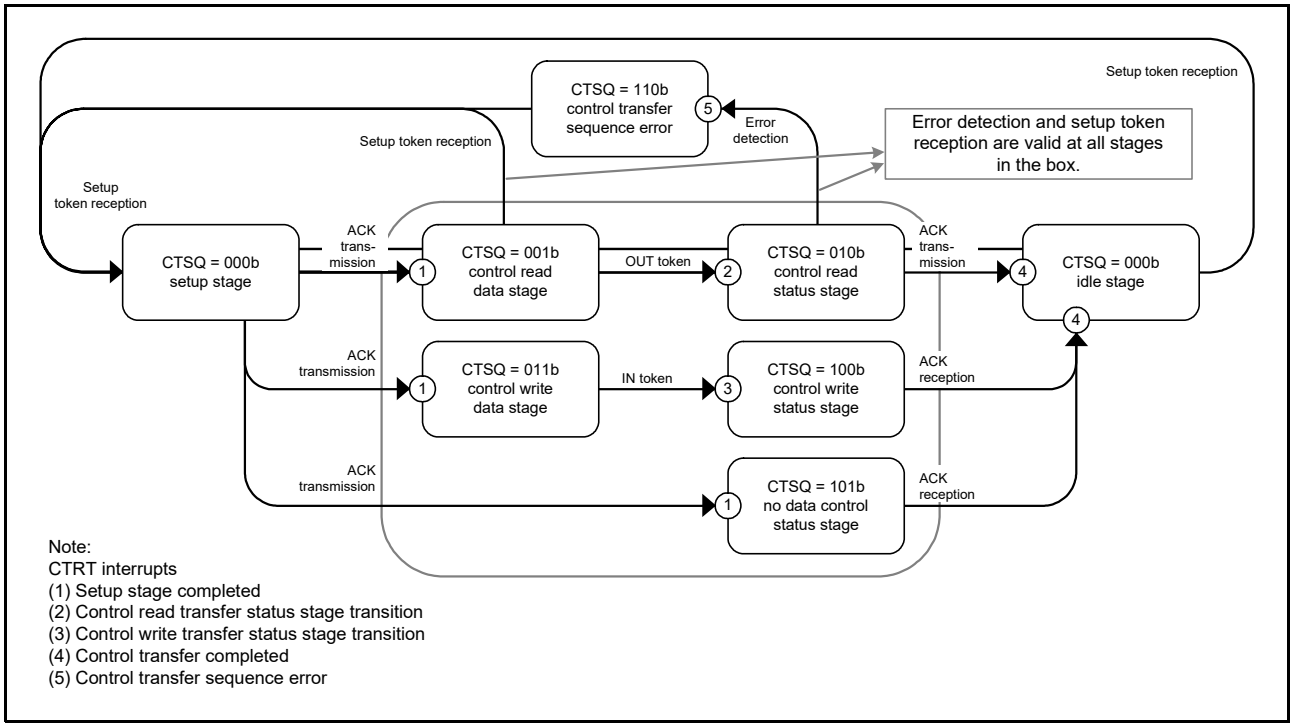


Figure 40.15 Control Transfer Stage Transitions

40.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

40.3.3.7 VBUS Interrupt

When the USB0_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0_VBUS pin level.

40.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

40.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

40.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

40.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

40.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

40.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

40.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

40.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

40.3.4 Pipe Control

Table 40.16 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 40.16 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	Pipes 1 to 9: Can be set
	BFRE	Selects the BRDY interrupt mode	Pipes 1 to 5: Can be set
	DBLB	Selects double buffer mode	Pipes 1 to 5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Can be set (only when bulk transfer has been selected) Pipes 3 to 5: Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 9: Cannot be set
	IITV[2:0]	Interval counter	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 5: Cannot be set Pipes 6 to 9: Can be set (only when the host controller has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	Pipes 1 to 5: Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	Pipes 1 to 9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
	PID[1:0]	Response PID	Refer to section 40.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Can be set
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Can be set
PIPEnTRN	—	Transaction counter	Pipes 1 to 5: Can be set

40.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

40.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

40.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

40.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- Pipes 1 to 5: Set 8, 16, 32, or 64 when using bulk transfer.
- Pipes 1 and 2: Set a value between 1 and 256 when using isochronous transfer.
- Pipes 6 to 9: Set a value between 1 and 64.

40.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

40.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
  (For details, refer to section 40.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

40.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

40.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

40.3.4.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

40.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

40.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

40.3.5 FIFO Buffer Memory

40.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

(1) Buffer Status

Table 40.17 and Table 40.18 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for pipe 1 to pipe 5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 40.17 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 40.18 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

40.3.5.2 FIFO Buffer Clearing

Table 40.19 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for pipes 1 to 5, using the PIPECFG.DBLB bit.

Table 40.19 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

40.3.5.3 FIFO Port Functions

Table 40.20 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 40.20 FIFO Port Function Settings

Register	Bit	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[8:0] read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE[3:0]	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN[8:0]	Checks the length of receive data.

(1) FIFO Port Selection

Table 40.21 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 40.21 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.
If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.
If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.
To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

40.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.
The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.
Table 40.22 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 40.22, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.
The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 40.22 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

40.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

40.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

40.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.
Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 40.15.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).
After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

40.3.7 Bulk Transfers (Pipes 1 to 5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 40.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1)
- Transaction count function
  (PIPEnTRE.TRENB and TRCLR bits and PIPEnTRN register: refer to section 40.3.4.5, Transaction Counter (For Pipes 1 to 5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 40.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 40.3.4.9, Auto Response Mode)

40.3.8 Interrupt Transfers (Pipes 6 to 9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

40.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

40.3.9 Isochronous Transfers (Pipes 1 and 2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

40.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 40.23 and Table 40.24 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 40.23 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Table 40.24 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

40.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

40.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 40.25 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 40.25 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV[2:0]}$ frames.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

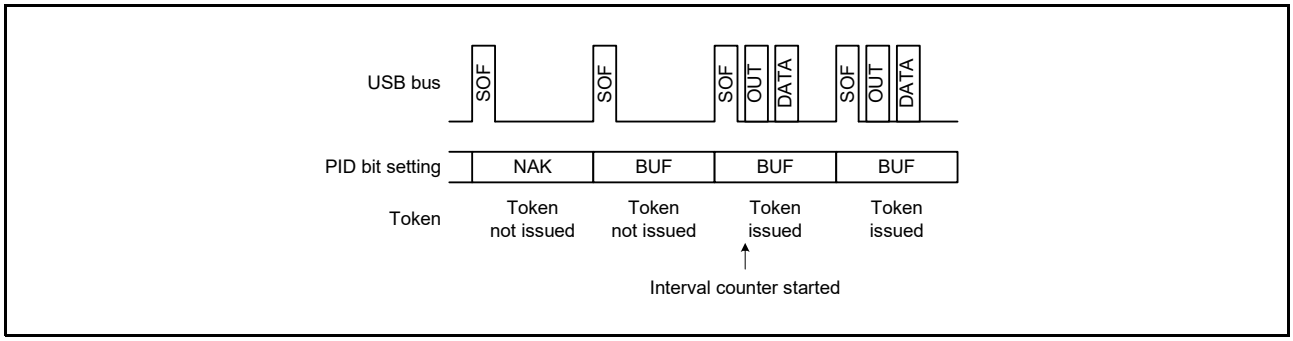


Figure 40.16 Token Issuance When IITV[2:0] = 000b

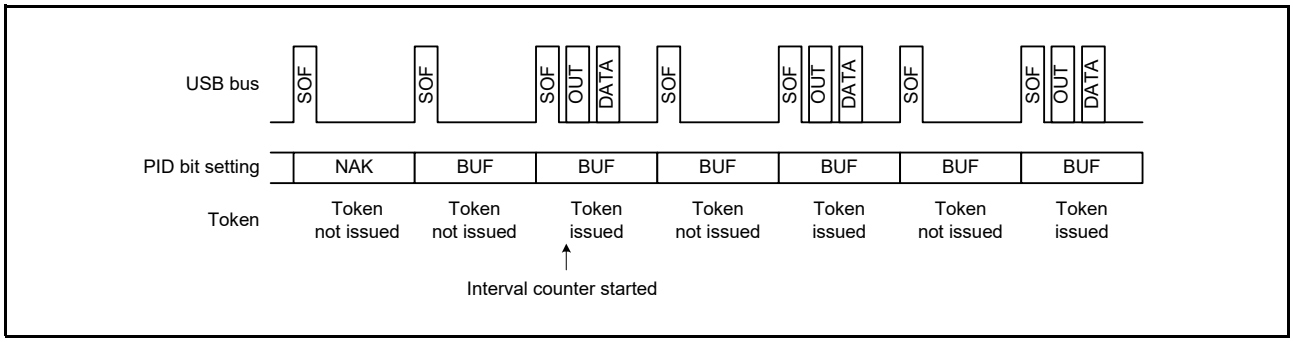


Figure 40.17 Token Issuance When IITV[2:0] = 001b

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

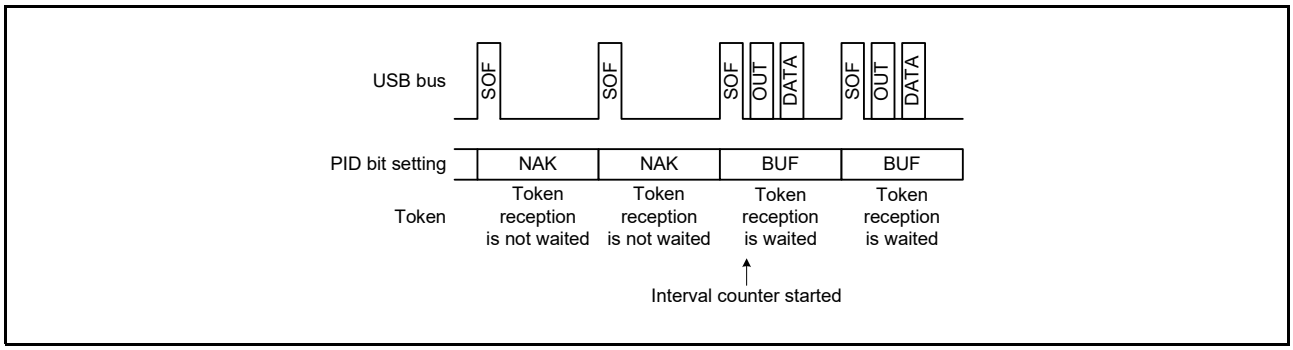


Figure 40.18 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

- When the IITV[2:0] ≠ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

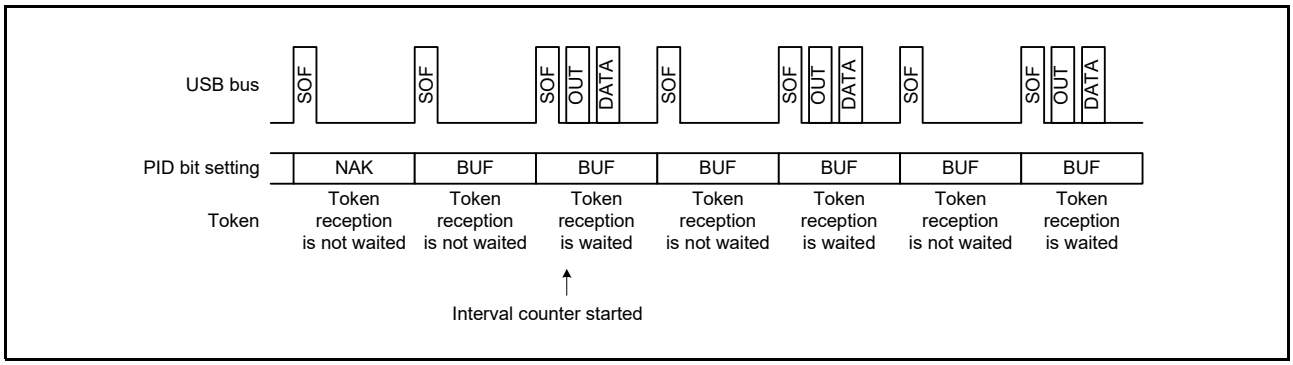


Figure 40.19 Relationship between Frames and Expected Token Reception When IITV[2:0] ≠ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 40.20 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

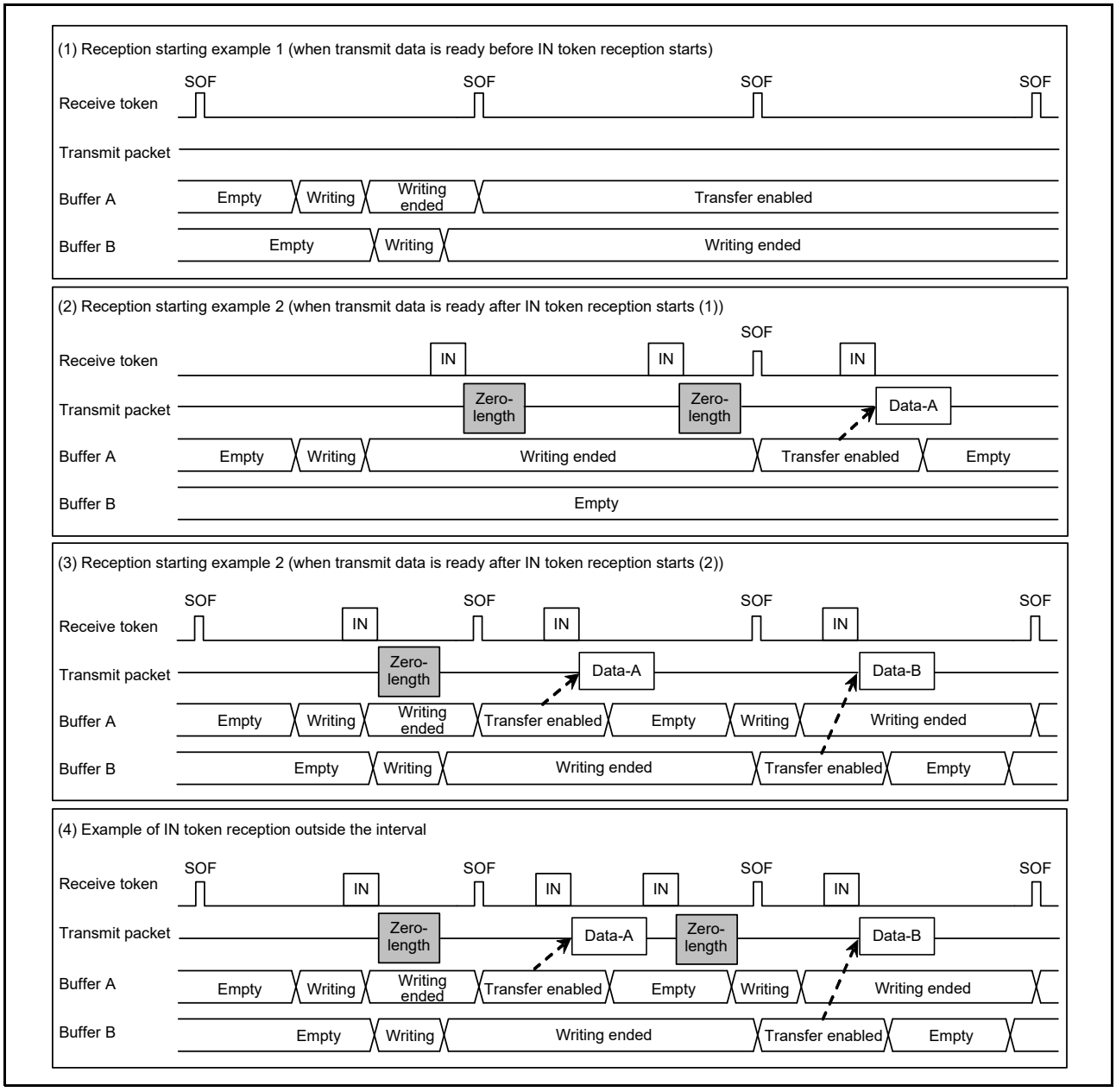


Figure 40.20 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
  The buffer flush operation is carried out after the first successful transaction.

Figure 40.21 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

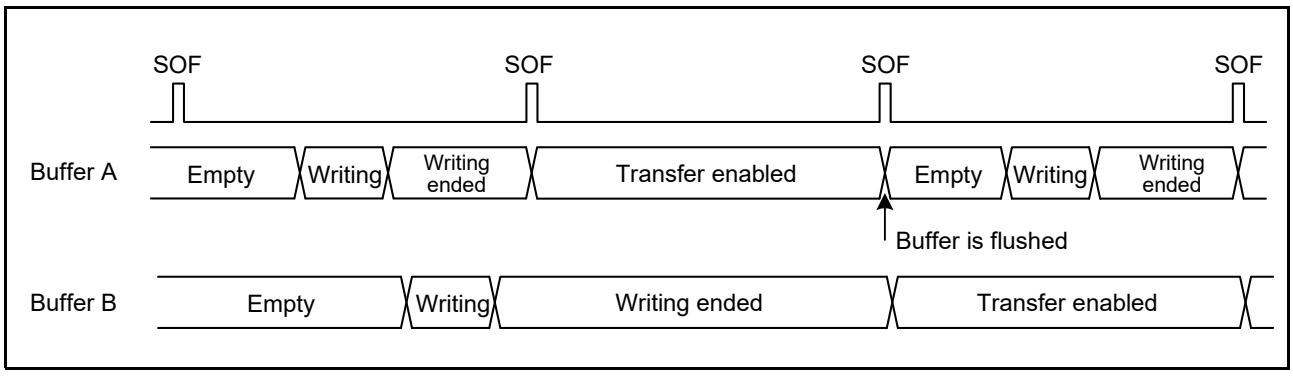


Figure 40.21 Example of Buffer Flush Operation

Figure 40.22 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN flag should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

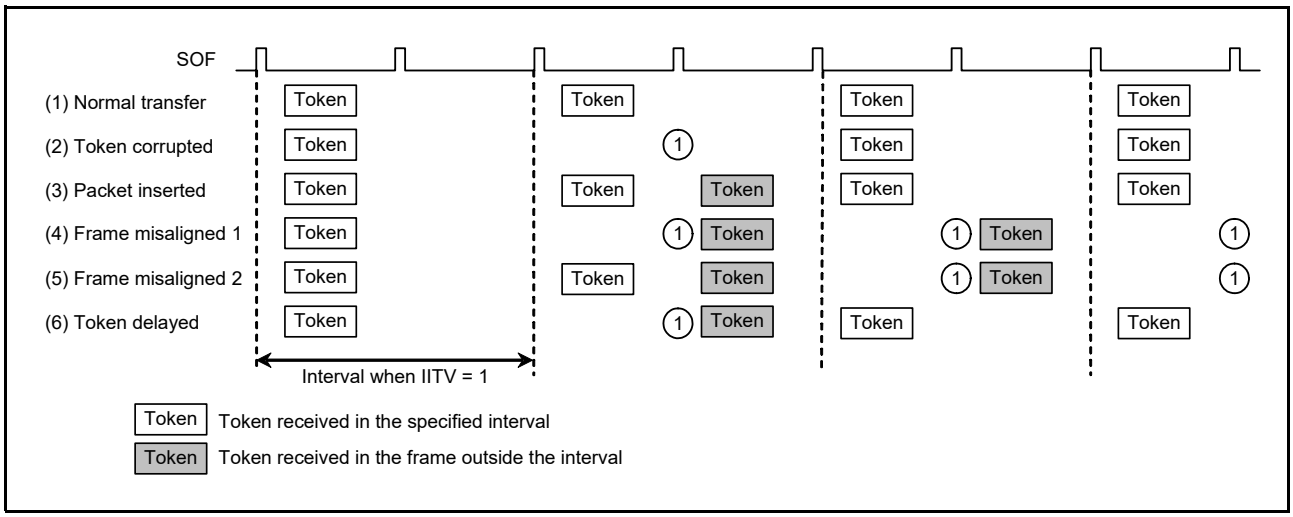


Figure 40.22 Example of Interval Error Occurrence When IITV[2:0] = 001b

40.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] flags are not updated.

40.3.11 Pipe Schedule

40.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 40.26.

Table 40.26 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID[1:0]**	**IITV[0]**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

40.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers
   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

40.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.
Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

40.4 Usage Notes

40.4.1 Setting the Module-Stop Function

Operation of the USB module can be disabled or enabled by setting a bit in the module stop control register B (MSTPCRB). The USB is initially disabled after a reset. Registers in the USB only become accessible after it has been released from the module-stop state. For details, refer to section 11, Low Power Consumption.

41. USB 2.0 High-Speed Host/Function Module (USBHS)

41.1 Overview

This MCU incorporates a USB 2.0 high-speed host/function module (USBHS). The USBHS is a USB controller that is equipped to operate as a host controller or a function controller. As a host controller, the USBHS supports high-speed transfer, full-speed transfer, and low-speed transfer as defined in the Universal Serial Bus Specification Revision 2.0 (hereinafter referred to as USB 2.0). As a function controller, the USBHS supports high-speed transfer and full-speed transfer as defined in USB 2.0. The USBHS has an internal USB transceiver and supports all of the transfer types defined in USB 2.0. The USBHS has FIFO buffers for data transfer, providing a maximum of 10 pipes. Any endpoint numbers can be assigned to pipes 1 to 9, based on the peripheral devices or user system for communication.

Table 41.1 USBHS Specifications

Item	Specifications
Features	• Incorporates a USB device controller (UDC) and transceiver for USB 2.0 The host controller and function controller operations are provided and the On-The-Go (OTG) function is supported. • Host controller and function controller operations are switched by software.
	In host controller operation: • Supports high-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) • Automatic scheduling for Start-of-Frame (SOF) packets and other packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected via a single hub
	In function controller operation: • Supports high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) • Control transfer stage control function • Device state control function • Auto response function for SetAddress() request • SOF recovery function
Communication data transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• FIFO buffer of up to 8.5 Kbytes for USB communications is available. • Up to 10 pipes can be selected (including the default control pipe). • Programmable pipe configurations • Endpoint numbers can be assigned flexibly to pipes 1 to 9.
	• Transfer conditions that can be set for each pipe: Pipe 0 (default control pipe): Control transfer, 64-byte fixed single buffer Pipes 1 and 2: Bulk transfer/isochronous transfer in continuous transfer mode. Programmable buffer size up to 2 Kbytes. Selectable single buffer or double buffer mode. Pipes 3 to 5: Bulk transfer in continuous transfer mode. Programmable buffer size up to 2 Kbytes. Selectable single buffer or double buffer mode. Pipes 6 to 9: Interrupt transfer, 64-byte fixed single buffers
Other features	• Transfer ending function using transaction count • Function that changes the BRDY interrupt event notification timing • Function that automatically clears the FIFO buffer after the data for the pipe specified at the D0FIFO port or D1FIFO port has been read • NAK setting function for response packet ID (PID) generated by end of transfer • Internal pull-up and pull-down resistors for D+ and D– • Supports the USB 2.0 Engineering Change Notice for Link Power Management (LPM). A new sleep state (referred to as the L1 state) is available. • Supports Battery Charging Specification Revision 1.2 • To reduce power consumption, a classic-only mode where operation is only in accord with the USB 1.1 standard is selectable.
Protect Zone filter	Security attribute can be set.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 41.1 shows a block diagram of the USBHS.

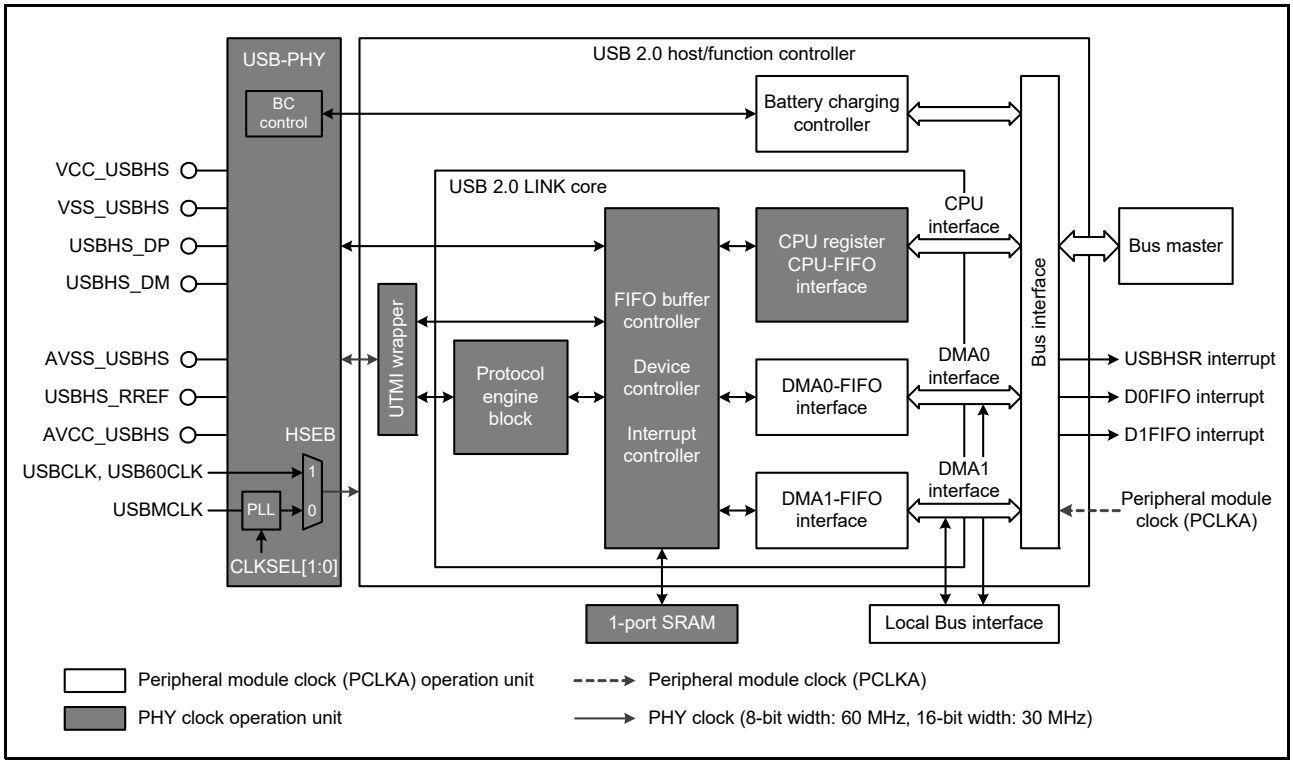


Figure 41.1 USBHS Block Diagram

Table 41.2 lists the I/O pins of the USBHS.

Table 41.2 Pin Configuration

Pin Symbol	I/O	Function
VCC_USBHS	Input	Power supply pin for the USBHS
VSS_USBHS	Input	Ground pin for the USBHS
AVCC_USBHS	Input	Analog power supply pin for the USBHS
AVSS_USBHS	Input	Analog ground pin for the USBHS.
USBHS_RREF	I/O	Reference current source pin for the USBHS. This pin should be connected to the AVSS_USBHS pin through a resistor of 2.2 kΩ (±1%).
USBHS_DP	I/O	I/O pin for the D+ data line of the USB bus
USBHS_DM	I/O	I/O pin for the D– data line of the USB bus
USBHS_EXICEN	Output	This pin should be connected to the OTG power supply IC.
USBHS_ID	Input	This pin should be connected to the OTG power supply IC.
USBHS_VBUSEN	Output	VBUS power enable pin for USB
USBHS_OVRCURA USBHS_OVRCURB	Input	Overcurrent pin for USB
USBHS_VBUS	Input	USB cable connection monitor input pin

41.2 Register Descriptions

41.2.1 System Configuration Control Register (SYSCFG)

Address(es): USBHS.SYSCFG 8745 1000h, USBHS_NS.SYSCFG 0745 1000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CNEN	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	x	x	x	x	x	x	x	0	0	0	1	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the line is disabled. 1: Pulling down the line is enabled.	R/W
b6	DCFM	Controller Operation Select	0: Function controller operation is selected. 1: Host controller operation is selected.	R/W
b7	HSE	High-Speed Operation Enable	0: High-speed operation is disabled. (In function controller operation: Full-speed. In host controller operation: Full-/low-speed) 1: High-speed operation is enabled (the controller detects the communication speed).	R/W
b8	CNEN	Single End Receiver Enable	0: Single end receiver operation is disabled. 1: Single end receiver operation is enabled.	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

The SYSCFG register can be write accessed even when the LPSTS.SUSPENDM bit is 0 (PHY clock is stopped). Note, however, that written values are only reflected in the SYSCFG register after the LPSTS.SUSPENDM bit is set to 1 (PHY clock is supplied to the LINK).

USBE Bit (USB Operation Enable)

The USBE bit enables and disables USBHS operation. Use the PHYSET.CLKSEL[1:0] bits to specify the input clock and verify that the PLLSTA.PLLLOCK flag is 1 before setting the USBE bit to 1. When operating in classic-only mode, set the PHYSET.HSEB bit to 1 before setting the USBE bit to 1. At that time, the USBCLK must be set to 48 MHz and USB60CLK must be set to 60 MHz. Refer to section 41.3.3 for details on setting clocks.

In host controller operation, set the DRPD bit to 1, eliminate chattering of the SYSSTS0.LNST[1:0] flags, and confirm that the USB bus state is stable before setting the USBE bit to 1.

When modifying the USBE bit from 1 to 0, some bits are initialized. Refer to Table 41.3 for a list of the bits initialized from setting the USBE bit to 0.

Table 41.3 Bits Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Bits	Remarks
Function controller operation (when the DCFM bit is 0)	SYSSTS0.LNST[1:0]	The value is retained in host controller operation.
	DVSTCTR0.RHST[2:0]	
	INTSTS0.DVSQ[2:0]	The value is retained in host controller operation.
	USBADDR.USBADDR[6:0]	The value is retained in host controller operation.
	USBREQ.BREQUEST[7:0] and BMREQUESTTYPE[7:0]	The value is retained in host controller operation.
	USBVAL.WVALUE[15:0]	The value is retained in host controller operation.
	USBINDX.WINDEX[15:0]	The value is retained in host controller operation.
	USBLENG.WLENGTH[15:0]	The value is retained in host controller operation.
Host controller operation (When the DCFM bit is 1)	DVSTCTR0.RHST[2:0]	
	FRMNUM.FRNM[10:0]	The value is retained in function controller operation.
	UFRMNUM.UFRNM[2:0]	The value is retained in function controller operation.

DPRPU Bit (D+ Line Resistor Control)

In function controller operation, the DPRPU bit enables and disables pulling up the D+ line.

When the DPRPU bit is set to 1 in function controller operation, the USBHS pulls up the D+ line and notifies the USB host of the attached state. Modifying the DPRPU bit from 1 to 0 stops the USBHS from pulling up the D+ line, thus it appears that the USBHS is detached from the USB host.

Set the DPRPU bit to 1 in function controller operation, and set it to 0 in host controller operation.

DRPD Bit (D+/D– Line Resistor Control)

In host controller operation, the DRPD bit enables and disables pulling down the D+ and D– lines.

Set the DRPD bit to 1 in host controller operation. Set the DRPD bit to 0 when OTG is not used in function controller operation.

DCFM Bit (Controller Operation Select)

The DCFM bit selects the function of the USBHS. Modify this bit when both the DPRPU and DRPD bits are 0.

HSE Bit (High-Speed Operation Enable)

This bit enables or disables high-speed operation.

When the HSE bit is 1, the USBHS performs high-speed or full-speed operation based on the results of reset handshake.

1. In host controller operation
   Setting the HSE bit to 0 lets the USBHS perform low-speed operation or full-speed operation. If the DVSTCTR0.RHST[2:0] bits indicate the attach state of a peripheral device in low-speed operation, set the HSE bit to 0. Setting the HSE bit to 1 lets the USBHS perform high-speed or full-speed operation according to the results of the reset handshake protocol. Modify the HSE bit after detection of the attach state (detection of an ATTCH interrupt) and before USB bus reset (before setting the DVSTCTR0.USBRST bit to 1), or after detection of the detach state (DTCH interrupt detected).
2. In function controller operation
   When the HSE bit is set to 0, the USBHS performs full-speed operation. When the HSE bit is set to 1, the USBHS executes the reset handshake protocol, and then performs high-speed or full-speed operation according to the results. Modify the HSE bit when the DPRPU bit is 0.

CNEN Bit (Single End Receiver Enable)

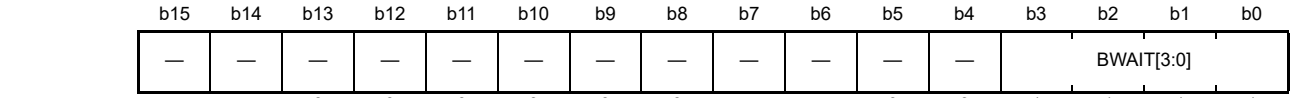
Setting the CNEN bit to 1 enables single-ended receiver operation and enables the SYSSTS0.LNST[1:0] flags to monitor the states of the D+ and D– lines.

When the USBHS is detached, the CNEN bit is used to prevent through-current damage which may occur from a single-ended receiver floating.

In host controller operation, verify that the PHY clock is being supplied before setting this bit to 1. In function controller operation, set this bit to 1 when the VBUS is detected due to a VBUS interrupt, and set the bit to 0 when the VBUS is removed.

41.2.2 CPU Bus Wait Register (BUSWAIT)

Address(es): USBHS.BUSWAIT 8745 1002h, USBHS_NS.BUSWAIT 0745 1002h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BWAIT[3:0]	CPU Bus Access Wait Select	b3 b0 0 0 0 0: 0 waits (2 access cycles) 0 0 0 1: 1 wait (3 access cycles) 0 0 1 0: 2 waits (4 access cycles) 0 0 1 1: 3 waits (5 access cycles) 0 1 0 0: 4 waits (6 access cycles) 0 1 0 1: 5 waits (7 access cycles) 0 1 1 0: 6 waits (8 access cycles) 0 1 1 1: 7 waits (9 access cycles) 1 0 0 0: 8 waits (10 access cycles) 1 0 0 1: 9 waits (11 access cycles) 1 0 1 0: 10 waits (12 access cycles) 1 0 1 1: 11 waits (13 access cycles) 1 1 0 0: 12 waits (14 access cycles) 1 1 0 1: 13 waits (15 access cycles) 1 1 1 0: 14 waits (16 access cycles) 1 1 1 1: 15 waits (17 access cycles) (initial value)	R/W
b5, b4	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b13 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b15, b14	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

BWAIT[3:0] Bits (CPU Bus Access Wait Select)

The BWAIT[3:0] bits specify the wait period to access registers associated with the USBHS.

Among the registers associated with the USBHS, when consecutively accessing registers other than the SYSCFG and BUSWAIT registers, the access cycle time must be at least 41 ns. Though the initial value of the BWAIT[3:0] bits is 1111b (17 access cycles), satisfy this condition by setting the best value for wait period in accordance with the frequency of the PCLKA. This setting is the same as the waits in accesses to the FIFO port register. The maximum speed for accessing the FIFO port registers is as follows:

MBW[1:0] bits are 00b (8-bit width): 15 Mbytes/s

MBW[1:0] bits are 01b (16-bit width): 30 Mbytes/s

MBW[1:0] bits are 10b (32-bit width): 60 Mbytes/s

41.2.3 System Configuration Status Register (SYSSTS0)

Address(es): USBHS.SYSSTS0 8745 1004h, USBHS_NS.SYSSTS0 0745 1004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flags	These flags indicate the status of the USB data line. See Table 41.4 for details.	R
b2	IDMON	USBHS_ID Pin Monitor Flag	0: Low is input to the USBHS_ID pin. 1: High is input to the USBHS_ID pin.	R
b4, b3	—	Reserved	These bits are undefined when read.	R
b5	SOFEA	SOF Active Monitor Flag While in Host Controller Operation	0: (μ)SOF output is stopped. 1: (μ)SOF output is continued.	R
b6	HTACT	Host Sequencer Status Monitor Flag	0: Host sequencer is stopped. 1: Host sequencer is operating.	R
b13 to b7	—	Reserved	These bits are undefined when read.	R
b15, b14	OVCMON[1:0]	USBHS_OVRCURA/ USBHS_OVRCURB Pin Monitor Flags	The OVCMON[1] flag indicates the status of the USBHS_OVRCURA pin. The OVCMON[0] flag indicates the status of the USBHS_OVRCURB pin.	R

LNST[1:0] Flags (USB Data Line Status Monitor Flags)

These flags indicate the state of the USB data lines (D+ line and D– line). See Table 41.4 for details. In function controller operation, set the SYSCFG.CNEN and USBE bits to 1 before reading the LNST[1:0] flags. In host controller operation, set the SYSCFG.DRPD bit to 1 before reading the LNST[1:0] flags.

However, for hardware-based contact checking using the battery charging function in function controller operation, set the SYSCFG.DRPD, SYSCFG.CNEN, and BCCTRL.IDPSRCE bits to 1 before reading the LNST[1:0] flags. Refer to section 41.3.15, Battery Charging Detection Processing for details.

Table 41.4 Status of USB Data Bus Lines

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation	During High-Speed Operation	During Chirp Operation
00b	SE0	SE0	Squelch	Squelch
01b	K state	J state	Unsquelch	Chirp J
10b	J state	K state	Invalid	Chirp K
11b	SE1	SE1	Invalid	Invalid

Chirp: The reset handshake protocol is being executed when high-speed operation is enabled (HSE bit is set to 1).
Squelch: SE0 or Idle state
Unsquelch: High-speed J state or high-speed K state
Chirp J: Chirp J state
Chirp K: Chirp K state

SOFEA Flag (SOF Active Monitor Flag While in Host Controller Operation)

When the USB bus enters the Suspend state by setting the DVSTCTR0.UACT bit to 0 in host controller operation, the SOFEA flag is used to check whether the output of the last (μ)SOF is completed. If the SYSCFG.USBE bit is set to 0 to stop the USBHS module and the LPSTS.SUSPENDM bit is set to 0 to stop supply of the clock signal while

communications are in progress in host controller operation, make sure the HTACT and SOFEA flags are set to 0.

HTACT Flag (Host Sequencer Status Monitor Flag)

The HTACT flag is set to 0 when the host sequencer of the USBHS is completely stopped. In host controller operation, make sure the HTACT flag is set to 0 when setting the DVSTCTR0.UACT bit to 0 to place the USBHS interface in the USB bus Suspend state or setting the LPSTS.SUSPENDM bit to 0 to stop the clock. In addition, when communicating in host controller operation, make sure the HTACT flag and SOFEA flag are both 0 when setting the SYSCFG.USBE bit to 0 to stop the USBHS, or when setting the LPSTS.SUSPENDM bit to 0 to stop the clock.

OVCMON[1:0] Flags (USBHS_OVRCURA/USBHS_OVRCURB Pin Monitor Flags)

The OVCMON[1] flag indicates the status of the USBHS_OVRCURA pin, and the OVCMON[0] flag indicates the status of the USBHS_OVRCURB pin.

41.2.4 PLL Status Register (PLLSTA)

Address(es): USBHS.PLLSTA 8745 1006h, USBHS_NS.PLLSTA 0745 1006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLLOCK
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	PLLLOCK	PLL Lock Flag	0: PLL is not locked. 1: PLL is locked.	R
b15 to b1	—	Reserved	These bits are undefined when read.	R

PLLLOCK Flag (PLL Lock Flag)

Indicates whether the USB-PHY internal PLL is locked or not. When classic-only mode is not used, make sure the PLL is locked before starting USB communication.

41.2.5 Device State Control Register 0 (DVSTCTR0)

Address(es): USBHS.DVSTCTR0 8745 1008h, USBHS_NS.DVSTCTR0 0745 1008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Reset Handshake Status	• In host controller operation b2 b0 0 0 0: Communication speed not determined (powered state or disconnected) 0 0 1: Low-speed connection 0 1 0: Full-speed connection 0 1 1: High-speed connection 1 0 0: USB reset handshake in progress 1 0 1: USB reset handshake in progress 1 1 0: USB reset handshake in progress 1 1 1: USB reset handshake in progress • In function controller operation b2 b0 0 0 0: Communication speed not determined (powered state or disconnected) 0 0 1: Reserved 0 1 0: Full-speed connection 0 1 1: High-speed connection 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b3	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b4	UACT	USB Bus Operation Enable for Host Controller Operation	0: Downstream port is disabled (SOF or µSOF transmission is disabled). 1: Downstream port is enabled (SOF or µSOF transmission is enabled).	R/W
b5	RESUME	Resume Signal Output for Host Controller Operation	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output for Host Controller Operation	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Remote Wakeup Detection Enable for Host Controller Operation	0: Downstream port remote wakeup detection is disabled 1: Downstream port remote wakeup detection is enabled	R/W
b8	WKUP	Remote Wakeup Output for Function Controller Operation	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	USBHS_VBUSEN Output Pin Control	0: The external USBHS_VBUSEN pin outputs the low level. 1: The external USBHS_VBUSEN pin outputs the high level.	R/W
b10	EXICEN	USBHS_EXICEN Output Pin Control	0: The external USBHS_EXICEN pin outputs the low level. 1: The external USBHS_EXICEN pin outputs the high level.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	0: HNP not enabled 1: HNP is enabled	R/W
b15 to b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

RHST[2:0] Bits (USB Reset Handshake Status)

These flags indicate that a reset handshake is in progress.

1. In host controller operation
   When the USBRST bit is set to 1, the RHST[2:0] bits become 100b. If the SYSCFG.HSE bit is 1, the RHST[2:0] bits become 111b as soon as the USBHS detects Chirp K from the function. When the USBRST bit is set to 0, the USBHS stops the USB bus reset signal, and the USBHS fixes the RHST[2:0] bit value. When setting the TESTMODE.UTST[3:0] bits to “1xxxb” (when the USB test signal is output in host controller operation), the RHST[2:0] bits become 011b.
2. In function controller operation
   If the SYSCFG.HSE bit is 1, when the USBHS detects a USB bus reset, the RHST[2:0] bits become 100b. Then, the USBHS outputs Chirp K, and when the USB host detects Chirp J and/or Chirp K three times, the RHST[2:0] bits become 011b. If high-speed is not confirmed within 2.5 ms of Chirp K being output, the RHST[2:0] bits become 010b. If the SYSCFG.HSE bit is 0, when the USBHS detects a USB bus reset, the RHST[2:0] bits become 010b.After the USBHS detects the USB bus reset, when the RHST[2:0] bits are confirmed to be 010b or 011b, the device state transition interrupt is generated.

UACT Bit (USB Bus Operation Enable for Host Controller Operation)

When the UACT bit is set to 1 in host controller operation, the USBHS enables the USB bus for the USB port.
The USBHS starts the output of an (µ)SOF packet within one (micro)frame period after the UACT bit is set to 1.
If the UACT bit is set to 0, the USBHS enters the Idle state after an (µ)SOF packet is output.
The USBHS sets the UACT bit to 0 on either of the following conditions.

- A DTCH interrupt is detected
- An EOFERR interrupt is detected

Set the UACT bit to 1 either when the USB bus reset processing is complete (USBRST bit is set to 0), or when the USBHS has exited the Suspend state (RESUME bit is set to 0).
When the HL1CTRL1.L1REQ bit is set to 1 to issue an LPM token and an ACK response is received, the USBHS sets the UACT bit to 0. In addition, after exiting the L1 state, the USBHS sets the UACT bit to 1.
In function controller operation, set the UACT bit to 0.

RESUME Bit (Resume Signal Output for Host Controller Operation)

This bit controls the output of the resume signal in host controller operation.
If the RESUME bit set to 1 while in the Suspend state, the USBHS outputs the resume signal to the USB port. In addition, if the RWUPE bit is 1 and a remote wakeup signal is detected while in the Suspend state, the USBHS sets the RESUME bit to 1, and USBHS outputs the resume signal to the USB port.
The USBHS continues outputting the resume signal until the RESUME bit is set to 0 by software. Ensure that the period the RESUME bit is 1 (resume period) complies with the time defined in USB 2.0. In addition, set the UACT bit to 1 at the same time the resume processing is complete (RESUME bit is set to 0).
If the RESUME bit to 1 during the L1 state, the USBHS outputs the resume signal to the USB port. The resume period is the period indicated by the HL1CTRL2.HIRD[3:0] bits. The USBHS sets the RESUME bit to 0 at the end of the resume period.
Only set the RESUME bit is 1 while in the Suspend state or L1 state. In addition, set the RESUME bit to 0 in function controller operation.

USBRST Bit (USB Bus Reset Output for Host Controller Operation)

In host controller operation, this bit controls the output of the USB bus reset signal.
In host controller operation, if the USBRST bit is set to 1, the USBHS performs a USB bus reset on the USB port. If the SYSCFG.HSE bit is 1, the reset handshake protocol is executed.
The USBHS continues outputting the bus reset signal until the USBRST bit is set to 0 by software. Ensure that the period the USBRST bit is 1 (USB bus reset period) complies with the time defined in USB 2.0.
When setting the USBRST bit to 1 either during communication (UACT bit is 1), or during the resume processing (RESUME bit is 1), the USBHS prevents the USB but reset processing from starting until both the UACT and RESUME bits become 0. Set the UACT bit to 1 at the same time the USB bus reset processing is complete (USBRST bit is set to 0).
In function controller operation, set the USBRST bit to 0.

RWUPE Bit (Remote Wakeup Detection Enable for Host Controller Operation)

In host controller operation, this bit enables and disables remote wakeup signals (resume signals) from peripheral devices connected to downstream ports.
While the RWUPE bit set to 1, the USBHS performs resume processing (resume signal is output) when a remote wakeup signal (K state for 2.5 µs) is detected from a peripheral device connected to a downstream port.
While the RWUPE bit is 0, even if the USBHS detects a remote wakeup signal from a peripheral device connected to the USB port, the remote wakeup signal is ignored.
When the RWUPE bit is set to 1, do not stop the PHY clock even in the Suspend state (keep the LPSTS.SUSPENDM bit to 1). Also, executing a USB bus reset (setting the USBRST bit to 1) while in the Suspend state is prohibited in USB 2.0.
In addition, the RWUPE bit is used to enable and disable detection of the remote wakeup signal while in the L1 state.
In function controller operation, set the RWUPE bit to 0.

WKUP Bit (Remote Wakeup Output for Function Controller Operation)

In function controller operation, this bit is used to output the remote wakeup signal (resume signal) to the USB port.
The USBHS controls the output time of the remote wakeup signal. When the WKUP bit is set to 1, the USBHS sets this bit to 0 after outputting a 10-ms remote wakeup signal.
According to USB 2.0, the USBHS must be in the USB bus Idle state for at least 5 ms before the remote wakeup signal is transmitted. For this reason, even if the WKUP bit is set to 1 immediately after the USB bus is detected as being in the suspended state, the USBHS waits 2 ms before outputting the remote wakeup signal.
Only set the WKUP bit to 1 when the device state is in the suspended state (INTSTS0.DVSQ[2:0] bits are 1xxb) and the remote wakeup signaling is enabled by the USB host.
Also, when the WKUP bit is set to 1, do not stop the PHY clock even if the USBHS is in the suspended state (keep the LPSTS.SUSPENDM bit to 1).
If the WKUP bit is set to 1 while the USBHS is in the L1 state, the USBHS outputs the remote wakeup signal for 50 µs and then sets the WKUP bit to 0.
In host controller operation, set the WKUP bit to 0.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

When the USBHS is used as an OTG device B, set this bit to switch from function controller operation to host controller operation.
If the HNPBTOA bit is 1, the USBHS remains in the suspended state until HNP processing is complete even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. At this time, a resume interrupt is not generated even if a falling edge on the D+ line is detected.
After setting the HNPBTOA bit to 1, when attachment of a host is detected (due to pulling up by the other party) or if a timeout occurs during HNP processing, set the HNPBTOA bit to 0 to end HNP processing.

41.2.6 USB Test Mode Register (TESTMODE)

Address(es): USBHS.TESTMODE 8745 100Ch, USBHS_NS.TESTMODE 0745 100Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	UTST[3:0]	Test Mode	These bits output the USB test signals. See Table 41.5 for details on the output signals.	R/W
b15 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

UTST[3:0] Bits (Test Mode)

Writing values to these bits allows the USBHS to output a USB test signal in high-speed operation mode.
Table 41.5 lists the Test Mode Operation of the USBHS.

Table 41.5 Test Mode Operation

Test Mode	UTST[3:0] Bit Settings	
	In function controller operation	In host controller operation
Normal operation	0000b	0000b
Test_J	0001b	1001b
Test_K	0010b	1010b
Test_SE0_NAK	0011b	1011b
Test_Packet	0100b	1100b
Test_Force_Enable	—	1101b
Reserved	0101b to 0111b	1110b to 1111b

- In host controller operation:
  The UTST[3:0] bits can be set after setting the SYSCFG.DRPD bit to 1. After setting the UTST[3:0] bits, the USBHS outputs waveforms by setting the DVSTCTR0.UACT bit to 1. When the UTST[3:0] bits are set while in host controller operation, even if the USBHS is not connected to a peripheral device, the USBHS performs a test as if it were connected to a high-speed terminating resistor.
  Use the following procedure to set the UTST[3:0] bits.
  (1) Reset the MCU.
  (2) Start supplying the PHY clock and then set the LPSTS.SUSPENDM bit to 1.
  (3) Set the SYSCFG.DCFM and DRPD bits to 1 (the SYSCFG.HSE bit does not need to be set to 1).
  (4) Set the SYSCFG.USBE bit to 1.
  (5) Set the UTST[3:0] bits according to the test content.
  (6) Set the DVSTCTR0.UACT bit to 1.

  Use the following procedure to change the UTST[3:0] bits. Steps (1) to (6) described above are assumed to have been executed in advance.
  (1) Set the DVSTCTR0.UACT and SYSCFG.USBE bits to 0.
  (2) Set the SYSCFG.USBE bit to 1.
  (3) Set the UTST[3:0] bits according to the test content.
  (4) Set the DVSTCTR0.UACT bit to 1.

When the UTST[3:0] bits are set to 1011b (Test_SE0_NAK), the USBHS does not output (µ)SOF packets even if the DVSTCTR0.UACT bit is set to 1.
When the UTST[3:0] bits are set to 1101b (Test_Force_Enable), the USBHS outputs (µ)SOF packets by setting the DVSTCTR0.UACT bit is set to 1. Also, in this test mode, even if the USBHS detects a high-speed disconnection (DTCH interrupt detected), the USBHS does not control the hardware corresponding to the detection.
When setting the UTST[3:0] bits, set the PID[1:0] bits of all pipe control registers to 00b (NAK response).
When performing normal USB communication after setting test mode, reset the MCU.

- In function controller operation:
  Set the UTST[3:0] bits according to the SetFeature request from the USB host during high-speed communication.
  The USBHS does not enter the suspended state when the UTST[3:0] bits are set to a value from 0001b to 0100b.
  When performing normal USB communication after setting test mode, reset the MCU.

41.2.7 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) FIFO port register values when the MBW[1:0] bits are 10b (32-bit width)

Address(es): USBHS.CFIFO 8745 1014h, USBHS.D0FIFO 8745 1018h, USBHS.D1FIFO 8745 101Ch, USBHS_NS.CFIFO 0745 1014h, USBHS_NS.D0FIFO 0745 1018h, USBHS_NS.D1FIFO 0745 101Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(2) FIFO port register values when the MBW[1:0] bits are 01b (16-bit width) and the BIGEND bit is 0 (Little endian)

Address(es): USBHS.CFIFO.H 8745 1016h, USBHS.D0FIFO.H 8745 101Ah, USBHS.D1FIFO.H 8745 101Eh, USBHS_NS.CFIFO.H 0745 1016h, USBHS_NS.D0FIFO.H 0745 101Ah, USBHS_NS.D1FIFO.H 0745 101Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(3) FIFO port register values when the MBW[1:0] bits are 01b (16-bit width) and the BIGEND bit is 1 (Big endian)

Address(es): USBHS.CFIFO.L 8745 1014h, USBHS.D0FIFO.L 8745 1018h, USBHS.D1FIFO.L 8745 101Ch, USBHS_NS.CFIFO.L 0745 1014h, USBHS_NS.D0FIFO.L 0745 1018h, USBHS_NS.D1FIFO.L 0745 101Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(4) FIFO port register values when the MBW[1:0] bits are 00b (8-bit width) and the BIGEND bit is 0 (Little endian)

Address(es): USBHS.CFIFO.HH 8745 1017h, USBHS.D0FIFO.HH 8745 101Bh, USBHS.D1FIFO.HH 8745 101Fh, USBHS_NS.CFIFO.HH 0745 1017h, USBHS_NS.D0FIFO.HH 0745 101Bh, USBHS_NS.D1FIFO.HH 0745 101Fh

Value after reset:

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(5) FIFO port register values when the MBW[1:0] bits are 00b (8-bit width) and the BIGEND bit is 1 (Big endian)

Address(es): USBHS.CFIFO.LL 8745 1014h, USBHS.D0FIFO.LL 8745 1018h, USBHS.D1FIFO.LL 8745 101Ch, USBHS_NS.CFIFO.LL 0745 1014h, USBHS_NS.D0FIFO.LL 0745 1018h, USBHS_NS.D1FIFO.LL 0745 101Ch

Value after reset:

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a FIFO port register, FIFO port select register, and a FIFO port control register. The FIFO port registers (CFIFO, D0FIFO, and D1FIFO) read data from and write data to the FIFO buffers. The FIFO port select registers (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) are used to select the pipe assigned to the FIFO port. The FIFO port control registers are CFIFOCTR, D0FIFOCTR, and D1FIFOCTR.

The following are features and precautions pertaining to the FIFO ports:

- Access the FIFO buffer for the default control pipe (control transfer) through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- Registers configuring one FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- Access rights for a FIFO buffer can be on the CPU side or on the serial interface engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bits

Using the FIFO port bits, the received data can be read from the FIFO buffer and the transmit data can be written to the FIFO buffer. The FIFO port registers can only be accessed when the FRDY flag in the corresponding FIFO port control register is 1. Valid bits in the FIFO port registers depend on the MBW[1:0] and BIGEND bit values in the corresponding FIFO port select register. When the MBW[1:0] bits are 01b or 10b, data allocation changes in relation to the MDE.MDE[2:0] bit value and the CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND bits. Table 41.6 and Table 41.7 show the various data allocation.

Table 41.6 Data Allocation in a 32-Bit Access (MBW[1:0] Bits are 10b)

MDE[2:0] Bits	BIGEND Bit	b31 to b24	b23 to b16	b15 to b8	b7 to b0	Remarks
000b (big endian)	0 (little endian)	Address N+3	Address N+2	Address N+1	Address N+0	Data is transmitted sequentially from address N+3. Data received is stored sequentially from address N+3.
	1 (big endian)	Address N+0	Address N+1	Address N+2	Address N+3	Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
111b (little endian)	0 (little endian)	Address N+3	Address N+2	Address N+1	Address N+0	Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
	1 (big endian)	Address N+0	Address N+1	Address N+2	Address N+3	Data is transmitted sequentially from address N+3. Data received is stored sequentially from address N+3.

Table 41.7 Data Allocation in a 16-Bit Access (MBW[1:0] Bits are 01b)

MDE[2:0] Bits	BIGEND Bit	b15 to b8	b7 to b0	Remarks
000b (big endian)	0 (little endian)	Address N+1	Address N+0	Access the upper word (.H). Data is transmitted sequentially from address N+1. Data received is stored sequentially from address N+1.
	1 (big endian)	Address N+0	Address N+1	Access the lower word (.L). Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
111b (little endian)	0 (little endian)	Address N+1	Address N+0	Access the upper word (.H). Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
	1 (big endian)	Address N+0	Address N+1	Access the lower word (.L). Data is transmitted sequentially from address N+1. Data received is stored sequentially from address N+1.

41.2.8 CFIFO Port Select Register (CFIFOSEL)

Address(es): USBHS.CFIFOSEL 8745 1020h, USBHS_NS.CFIFOSEL 0745 1020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	x	x	0	0	x	0	x	x	0	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: Default control pipe (DCP) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b4	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b5	ISEL	FIFO Port Access Direction When Default Control Pipe is Selected	0: Reading from the FIFO buffer is selected 1: Writing to the FIFO buffer is selected	R/W
b7, b6	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11, b10	MBW[1:0]	CFIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Do not set this value.	R/W
b13, b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	REW	Buffer Pointer Rewind	Set this bit to 1 to rewind the buffer pointer. Writing 0 has no effect.	R/W
b15	RCNT	Read Count Mode	0: The CFIFOCTR.DTLN[11:0] bits are set to 000h when all of the receive data has been read from the CFIFO register. 1: The CFIFOCTR.DTLN[11:0] bits are decremented each time the receive data is read from the CFIFO register.	R/W

The same pipe number should not be set to the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the current pipe number for which data is read or written through the CFIFO port. When the CURPIPE[3:0] bits are modified, read the bits to confirm they have been successfully modified, and then proceed to the next processing.

Do not set the same pipe number in the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

If the CURPIPE[3:0] bits are modified while accessing the FIFO buffer, access until that point is retained, and after the CURPIPE[3:0] bits are modified back to the original setting, access can be resumed.

ISEL Bit (FIFO Port Access Direction When Default Control Pipe is Selected)

If the ISEL bit is modified while the default control pipe is the current pipe, read the bit to confirm it has been successfully modified. Set the ISEL bit and the CURPIPE[3:0] bits simultaneously.

BIGEND Bit (FIFO Port Endian Control)

Set this bit to specify the byte endian of the CFIFO port.

MBW[1:0] Bits (CFIFO Port Access Bit Width)

Set this bit to specify the bit width for accessing the CFIFO port.
When the transfer direction of the current pipe is receiving, if the MBW[1:0] bits are set to start reading data from the FIFO buffer, do not modify the MBW[1:0] bits until all the data has been read. When modifying the MBW[1:0] bits, set the CURPIPE[3:0] bits to a different value first, and set the CURPIPE[3:0] bits and MBW[1:0] bits simultaneously.
When the transfer direction of the current pipe is transmitting, while data is being written to the FIFO buffer, the bit width cannot be changed from 8-bit width to 16- or 32-bit width, and 16-bit width cannot be changed to 32-bit width. Also, even when 16-bit or 32-bit width is selected, an odd number of bytes can be written by byte access.

REW Bit (Buffer Pointer Rewind)

When the transfer direction of the current pipe is receiving, if the REW bit is set to 1 while reading data from the FIFO buffer, the FIFO buffer can be reread from the first data. In double buffer mode, the buffer that was being read can be reread from the first data.
Do not modify the CURPIPE[3:0] bits when setting the REW bit to 1. Also, confirm that the FRDY flag is 1 before setting the REW bit to 1.
For pipes in the receiving transfer direction, to rewrite to the FIFO buffer from the first data, set the BCLR bit.

RCNT Bit (Read Count Mode)

When the RCNT bit is set to 0, when all data has been read from the FIFO buffer (in double buffer mode, when a single plane is done being read) for the pipe specified by the CURPIPE[3:0] bits, the USBHS sets the CFIFOCTR.DTLN[11:0] bits to 000h.
When the RCNT bit set to 1, each time data is read from the FIFO buffer for the pipe specified by the CURPIPE[3:0] bits, the USBHS decrements the value of the CFIFOCTR.DTLN[11:0] bits.

41.2.9 D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL)

Address(es): USBHS.D0FIFOSEL 8745 1028h, USBHS.D1FIFOSEL 8745 102Ch, USBHS_NS.D0FIFOSEL 0745 1028h, USBHS_NS.D1FIFOSEL 0745 102Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	x	0	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11, b10	MBW[1:0]	FIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Do not set this value.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Clear Mode after Current Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	Set this bit to 1 to rewind the buffer pointer. Writing 0 has no effect.	R/W
b15	RCNT	Read Count Mode	0: The DTLN[11:0] bits in the D0FIFOCTR/D1FIFOCTR register are set to 000h when all of the receive data has been read from D0FIFO/D1FIFO (after reading a single plane in double buffer mode). 1: The DTLN[11:0] bits in the D0FIFOCTR/D1FIFOCTR register are decremented each time the receive data is read from D0FIFO/D1FIFO.	R/W

The same pipe number should not be set to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. In addition, a pipe is not assigned if the setting of the D0FIFOSEL/D1FIFOSEL.CURPIPE[3:0] bits is 0000b.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the current pipe number for which data is read or written through the D0FIFO or D1FIFO port.

When the CURPIPE[3:0] bits are modified, read the bits to confirm they have been successfully modified, and then proceed to the next processing.

Do not set the same pipe number in the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.
If the CURPIPE[3:0] bits are modified while accessing the FIFO buffer, access until that point is retained, and after the CURPIPE[3:0] bits are modified back to the original setting, access can be resumed.

BIGEND Bit (FIFO Port Endian Control)

Set this bit to specify the byte endian of the D0FIFO and D1FIFO port.

MBW[1:0] Bits (FIFO Port Access Bit Width)

This bit specifies the bit width for accessing the D0FIFO or D1FIFO port.
When the current pipe is in the receiving direction, once reading data from the FIFO buffer is started after setting the MBW[1:0] bits, these bits should not be modified until all the data has been read. When modifying the MBW[1:0] bits, set the CURPIPE[3:0] bits to 0000b first, and set the CURPIPE[3:0] bits and the MBW[1:0] bits simultaneously.
When the current pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit/32-bit width or 16-bit width to 32-bit width while data is being written to the FIFO buffer. An odd number of bytes can also be written through byte-access control even when 16-bit or 32-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

This bit enables or disables the DMA/DTC transfer request.
Modify the bit when the CURPIPE[3:0] bits are 0000b.
To enable the DMA/DTC transfer request, set the CURPIPE[3:0] bits to the desired pipe number after setting the bit to 1.

DCLRM Bit (Automatic FIFO Buffer Clear Mode after Current Pipe is Read)

This bit enables or disables the automatic FIFO buffer clear after data in the current pipe is read. When the DCLRM bit is set to 1 on receiving a zero-length packet while the FIFO buffer assigned to the current pipe is empty, or when reading the received short packet is completed while the PIPECFG.BFRE bit is 1, the USBHS sets the BCLR bit in a FIFO port control register to 1. Set the DCLRM bit to 0 when the SOFCFG.BRDYM bit 1.

REW Bit (Buffer Pointer Rewind)

When the current pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data. When double-buffering is in use and reading is already in progress, reading either FIFO buffer from the first entry is possible. Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1. To rewrite to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

With the RCNT bit set to 0, the USBHS sets the DTLN[11:0] bits in the D0FIFOCTR/D1FIFOCTR register to 000h upon finishing reading all the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits (after reading a single plane in double buffer mode). With the RCNT bit set to 1, the USBHS decrements the value indicated by the DTLN[11:0] bits each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits. Set the RCNT bit to 0 when accessing D0FIFO/D1FIFO with the PIPECFG.BFRE bit set to 1.

41.2.10 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USBHS.CFIFOCTR 8745 1022h, USBHS.D0FIFOCTR 8745 102Ah, USBHS.D1FIFOCTR 8745 102Eh, USBHS_NS.CFIFOCTR 0745 1022h, USBHS_NS.D0FIFOCTR 0745 102Ah, USBHS_NS.D1FIFOCTR 0745 102Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	DTLN[11:0]	Receive Data Length	These bits indicate the length of the receive data. The meaning of the value of these bits differs with the setting of the RCNT bit in the FIFO port select register.	R
b12	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled 1: FIFO port access is enabled	R
b14	BCLR	CPU Buffer Clear	This bit should be set to 1 to clear the FIFO buffer on the CPU side for the pipe specified by the CURPIPE[3:0] bits. Writing 0 has no effect. This bit is 0 when read.	R/W
b15	BVAL	FIFO Buffer Valid	This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe specified by the CURPIPE[3:0] bits. Writing 0 has no effect.	R/W

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers respectively correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[11:0] Bits (Receive Data Length)

These bits indicate the length of the receive data. While data is being read from the FIFO buffer, the value of the DTLN[11:0] bits differs according to the setting of the RCNT bit in the FIFO port select registers as described below.

- When the RCNT bit is 0
  The USBHS sets the DTLN[11:0] bits to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data in the FIFO buffer (until a single plane has been read in double buffer mode).
  When the PIPECFG.BFRE bit for the current pipe is set to 1, the USBHS retains the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- When the RCNT bit is 1
  The USBHS decrements the value indicated by the DTLN[11:0] bits each time the CPU or DMAC/DTC reads the receive data from the FIFO buffer. (The value is decremented by one when the MBW[1:0] bits are 00b, by two when the MBW[1:0] bits are 01b, and by four when the MBW[1:0] bits are 10b.)
  The DTLN[11:0] bits become 000h when all the data has been read from the FIFO buffer. However, in double buffer mode, if data reception is completed in one FIFO buffer plane before all the data has been read from the other plane, the DTLN[11:0] bits indicate the length of the receive data in the latter plane when all the data has been read from the former plane.
  In addition, updating the DTLN[11:0] bits can take up to 150 ns. After reading the FIFO buffer, wait at least 150 ns before reading the DTLN[11:0] bits.

FRDY Flag (FIFO Port Ready Flag)

This flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the cases below, the FRDY flag also becomes 1, but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the current pipe is empty.
- When the PIPECFG.BFRE bit for the current pipe is 1, a short packet is received and the data read operation is complete.

BCLR Bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side of the current pipe.
When double buffer mode is set for the FIFO buffer assigned to the current pipe, the USBHS only clears the FIFO buffer on the CPU side even when both planes are readable.
When the current pipe is the default control pipe, setting the BCLR bit to 1 clears the FIFO buffer regardless of whether the FIFO buffer access rights are on the CPU side or the SIE side. To clear the FIFO buffer when the SIE has the access right, set the DCPCTR.PID[1:0] bits to 00b (NAK response) and then set the BCLR bit to 1.
When the current pipe is a pipe other than the default control pipe, set the BCLR bit to 1 when the FRDY flag in a FIFO port control register is 1.

BVAL Bit (FIFO Buffer Valid)

When the transfer direction of the pipe specified by the CURPIPE[3:0] bits is transmitting, set the BVAL bit to 1 in the cases listed below. When the BVAL bit is set to 1, the USBHS switches the FIFO buffer access right from the CPU side to the SIE side, enabling transmission.

- To transmit a short packet, set this bit to 1 after transmit data has been written.
- To transmit a zero-length packet, set this bit to 1 before transmit data is written.
- For pipes in continuous transfer mode, set this bit to 1 after writing data that is an integral multiple of the maximum packet size and less than the buffer size.

When the BVAL bit and BCLR bit are set to 1 simultaneously, the USBHS clears the data already written and enables the transmission of a zero-length packet. For pipes in discontinuous transfer mode, when the maximum packet size of data is written, the USBHS sets the BVAL bit to 1, and switches the FIFO buffer access right from the CPU side to the SIE side, enabling transmission.
Set the BVAL bit to 1 while the FRDY flag is 1.
When the transfer direction of the current pipe is receiving, do not set the BVAL bit to 1.

41.2.11 Interrupt Enable Register 0 (INTENB0)

Address(es): USBHS.INTENB0 8745 1030h, USBHS_NS.INTENB0 0745 1030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BRDYE	Buffer Ready Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b9	NRDYE	Buffer Not Ready Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b12	DVSE	Device State Transition Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b13	SOFE	Frame Number Refresh Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b14	RSME	Resume Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b15	VBSE	VBUS Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W

Note 1. The CTRE bit, DVSE bit, and RSME bit can only be set to 1 in function controller operation. Set these bits to 0 in host controller operation.

When a status flag in the INTSTS0 register becomes 1, if the corresponding interrupt request enable bit in the INTENB0 register is 1, the USBHS requests the USBHSR interrupt.
Regardless of the INTENB0 register setting, if the conditions for each status are satisfied, the status flags corresponding to the INTSTS0 register become 1.
When the status flags in the INTSTS0 register are 1, if a corresponding interrupt request enable bit in the INTENB0 register is rewritten from 0 to 1, the USBHSR interrupt request is generated.

41.2.12 Interrupt Enable Register 1 (INTENB1)

Address(es): USBHS.INTENB1 8745 1032h, USBHS_NS.INTENB1 0745 1032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	L1RSMENDE	LPMENDE	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINTE	Portable Device Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b5	SIGNE	Setup Transaction Error Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b6	EOFERRE	End-of-Frame Error Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	LPMENDE	LPM Transaction End Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b9	L1RSMENDE	L1 Resume End Interrupt Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	ATTCHE	Device Connection Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b12	DTCHE	Device Disconnection Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	BCHGE	Bus Change Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b15	OVRCRE	Overcurrent Change Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W

Note 1. This bit can only be set to 1 in host controller operation. Set this bit to 0 in function controller operation.

When a status flag in the INTSTS1 register becomes 1, if the corresponding interrupt request enable bit in the INTENB1 register is 1, the USBHS requests the USBHSR interrupt.
Regardless of the INTENB1 register setting, if the conditions for each status are satisfied, the status flags corresponding to the INTSTS1 register become 1.
When the status flags in the INTSTS1 register are 1, if a corresponding interrupt request enable bit in the INTENB1 register is rewritten from 0 to 1, the USBHSR interrupt request is generated.

41.2.13 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USBHS.BRDYENB 8745 1036h, USBHS_NS.BRDYENB 0745 1036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBRDYE[9:0]	Pipe 9 to Pipe 0 BRDY Interrupt Request Enable *1	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBRDYE[9] bit, pipe 8 corresponds to the PIPEBRDYE[8] bit, etc.).

When a flag in the BRDYSTS register becomes 1, if the corresponding BRDYENB.PIPEBRDYE[9:0] bit is 1, the INTSTS0.BRDY flag becomes 1.
In this case, if the INTENB0.BRDYE bit is 1, the USBHS requests the USBHSR interrupt.
When any of the BRDYSTS.PIPEBRDY[9:0] flags is 1, if the corresponding interrupt request enable bit in the BRDYENB register is rewritten from 0 to 1, the INTSTS0.BRDY flag becomes 1.

41.2.14 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USBHS.NRDYENB 8745 1038h, USBHS_NS.NRDYENB 0745 1038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPENRDYE[9:0]	Pipe 9 to Pipe 0 NRDY Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPENRDYE[9] bit, pipe 8 corresponds to the PIPENRDYE[8] bit, etc.).

When a flag in the NRDYSTS register becomes 1, if the corresponding NRDYENB.PIPENRDYE[9:0] bit is 1, the INTSTS0.NRDY flag becomes 1.
In this case, if the INTENB0.NRDYE bit is 1, the USBHS requests the USBHSR interrupt.
When any of the NRDYSTS.PIPENRDY[9:0] flags is 1, if the corresponding interrupt request enable bit in the NRDYENB register is rewritten from 0 to 1, the INTSTS0.NRDY flag becomes 1.

41.2.15 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USBHS.BEMPENB 8745 103Ah, USBHS_NS.BEMPENB 0745 103Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBEMPE[9:0]	Pipe 9 to Pipe 0 BEMP Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBEMPE[9] bit, pipe 8 corresponds to the PIPEBEMPE[8] bit, etc.).

When a flag in the BEMPSTS register becomes 1, if the corresponding BEMPENB.PIPEBEMPE[9:0] bit is 1, the INTSTS0.BEMP flag becomes 1.
In this case, if the INTENB0.BEMPE bit is 1, the USBHS requests the USBHSR interrupt.
When any of the BEMPSTS.PIPEBEMP[9:0] flags is 1, if the corresponding interrupt request enable bit in the BEMPENB register is rewritten from 0 to 1, the INTSTS0.BEMP flag becomes 1.

41.2.16 SOF Output Configuration Register (SOFCFG)

Address(es): USBHS.SOFCFG 8745 103Ch, USBHS_NS.SOFCFG 0745 103Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	x	0	0	0	0	0	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	EDGESTS	Interrupt Edge Processing Status Flag*1	This flag is 1 when the USBHS outputs an edge interrupt request signal.	R
b5	INTL	Interrupt Signal Output Method Select *2	0: Do not set this value (edge output) 1: Level output	R/W
b6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*3	0: The PIPEBRDY[9:0] flags are set to 0 by software. 1: The PIPEBRDY[9:0] flags are set to 0 by reading the FIFO buffer or writing to the FIFO buffer.	R/W
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*4	0: Not low-speed compatible 1: Low-speed compatible	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supplied to the USBHS.
Note 2. Set this bit to 1 before operating the USBHS.
Note 3. When setting the BRDYM bit to 1, set the INTL bit to 1.
Note 4. The TRNENSEL bit setting is only valid in host controller operation. Even in host controller operation, the setting of this bit has no effect on the transaction-enabled time during high-speed communication.

EDGESTS Flag (Interrupt Edge Processing Status Flag)

This flag is enabled when the INTL bit is 0 (edge output).

When multiple interrupt requests are generated, if one of the interrupt requests is cleared, the USBHS temporarily negates the interrupt request signal, after waiting 666.6 µs, the interrupt request signal is reasserted. While the interrupt request signal is negated, the EDGESTS flag becomes 1.

Confirm the EDGESTS flag is 0 when stopping the PHY clock.

BRDYM Bit (PIPEBRDY Interrupt Status Clear Timing)

This bit specifies the timing to clear the BRDY interrupt status (BRDYSTS.PIPEBRDY[9:0] flags) for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

When the USB port is in use for full-speed or low-speed communications, this bit specifies the timing with which the USBHS issues tokens in a frame (transaction-enabled time).

Set the TRNENSEL bit to 1 when a low-speed device is connected.

The TRNENSEL bit is only valid in host controller operation.

Set this bit to 0 in function controller operation.

41.2.17 PHY Setting Register (PHYSET)

Address(es): USBHS.PHYSET 8745 103Eh, USBHS_NS.PHYSET 0745 103Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HSEB	—	—	—	—	—	REPSEL[1:0]		—	—	CLKSEL[1:0]		CDPEN	—	PLLRESET	DIRPD
Value after reset:	0	x	x	x	0	x	0	0	x	x	1	1	0	x	1	1

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	DIRPD	Power-Down Control	0: Normal operation 1: Power-down mode	R/W
b1	PLLRESET	PLL Reset Control*1	0: Disable PLL reset control for USB-PHY 1: Enable PLL reset control for USB-PHY	R/W
b2	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b3	CDPEN	Charging Downstream Port Enable	0: Disable the charging downstream port 1: Enable the charging downstream port	R/W
b5, b4	CLKSEL[1:0]	Input System Clock Frequency	b5 b4 0 0: 12 MHz 0 1: 48 MHz 1 0: 20 MHz 1 1: 24 MHz	R/W
b7, b6	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b9, b8	REPSEL[1:0]	Terminating Resistance Adjustment	b9 b8 0 0: Disable the terminating resistance adjustment 0 1: Enable the terminating resistance adjustment 1 0: Do not set this value. 1 1: Do not set this value.	R/W
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b14 to b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15	HSEB	Classic-Only Mode*2	0: Classic-only mode is disabled. 1: Classic-only mode is enabled.	R/W

Note 1. Since the PLLRESET bit is 1 after a reset, changing the setting after release from the reset state is not required because of PLL reset control. Setting the PLLRESET bit to 1 after setting it to 0 may cause the USBHS operate erroneously.
Note 2. Do not set the HSEB bit to 0 after setting it to 1. Doing so may cause the USBHS operate erroneously.

CLKSEL[1:0] Bits (Input System Clock Frequency)

These bits select the frequency of the PLL clock source (USBMCLK) necessary for generating the USBHS PHY clock. The USBHS PHY clock is generated by the PLL embedded in the USB-PHY. The CLKSEL[1:0] bits set the frequency of the clock source (USBMCLK) supplied to the PLL. The USBMCLK is input from the EXTAL pin, but the clock specification must follow the clock standard in the USB 2.0. In classic-only mode, the PLL in the USB-PHY is stopped, so writing to the CLKSEL[1:0] bits is invalid (refer to the HSEB bit description for details).
Refer to section 41.3.3, Supplying Clocks for details on setting clocks.

REPSEL[1:0] Bits (Terminating Resistance Adjustment)

These bits automatically adjust the terminating resistance. Before operating the USBHS, set the REPSEL[1:0] bits to 01b to enable the automatic adjustment function for the terminating resistance. Note that the automatic adjustment function for the terminating resistance is stopped when the HSEB bit is 1 (classic-only mode is enabled).

HSEB Bit (Classic-Only Mode)

This bit selects whether the USBHS module operates in classic-only mode.

High-speed transfer by the USBHS requires the use of internal high-speed analog circuits including the PLL and clock and data recovery (CDR) circuit in the USB-PHY block. Classic-only mode limits transfer to the USB 1.1 specification (full-speed and low-speed transfer), so power consumption can be reduced by stopping the PHY module's internal PLL and other high-speed analog circuits.

In classic-only mode, USBCLK (48 MHz) and USB60CLK (60 MHz) clocks generated in the clock generation circuit must be supplied to the USBHS.

Refer to section 9, Clock Generation Circuit for details on supplying the clock. Refer to section 41.3.3 for details on setting clocks.

41.2.18 Interrupt Status Register 0 (INTSTS0)

Address(es): USBHS.INTSTS0 8745 1040h, USBHS_NS.INTSTS0 0745 1040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	x	0	0	0/1*1	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage*2	b2 b0 0 0 0: Idle state or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Reserved	R
b3	VALID	USB Request Reception Flag*2	0: Setup packet is not received 1: Setup packet is received	R/(W)
b6 to b4	DVSQ[2:0]	Device State*2	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 0 0: Suspended state 1 0 1: Suspended state 1 1 0: Suspended state 1 1 1: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: The USBHS_VBUS pin is low 1: The USBHS_VBUS pin is high	R
b8	BRDY	BRDY Interrupt Status Flag	0: BRDY interrupt not requested 1: BRDY interrupt requested	R
b9	NRDY	NRDY Interrupt Status Flag	0: NRDY interrupt not requested 1: NRDY interrupt requested	R
b10	BEMP	BEMP Interrupt Status Flag	0: BEMP interrupt not requested 1: BEMP interrupt requested	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*3	0: Control transfer stage transition interrupt not requested 1: Control transfer stage transition interrupt requested	R/(W) *4
b12	DVST	Device State Transition Interrupt Status Flag*3	0: Device state transition interrupt not requested 1: Device state transition interrupt requested	R/(W) *4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: Frame number refresh interrupt not requested 1: Frame number refresh interrupt requested	R/(W) *4
b14	RESM	Resume Interrupt Status Flag*3	0: Resume interrupt not requested 1: Resume interrupt requested	R/(W) *4
b15	VBINT	VBUS Interrupt Status Flag*5	0: VBUS interrupt is not requested on detecting a change in the USBHS_VBUS pin. 1: VBUS interrupt is requested on detecting a change in the USBHS_VBUS pin.	R/(W) *4

Note 1. The value is 0 when the MCU is reset, and 1 when a USB bus reset occurs.
Note 2. The CTSQ[2:0] bits, VALID flag, and DVSQ[2:0] bits are only valid in function controller operation.
Note 3. The CTRT, DVST, and RESM flags are only changed in function controller operation. Set the corresponding interrupt enable bits to 0 (disabled) in host controller operation.
Note 4. To clear the CTRT, DVST, SOFR, RESM, or VBINT flags, write 0 only to the flags to be cleared, and write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The USBHS detects a change in the status indicated by the VBINT flag even while the clock supply is stopped (while the LPSTS.SUSPENDM bit is 0), and requests the USBHSR interrupt when the corresponding interrupt enable bit is 1. Clear these flags after supplying the PHY clock.

BRDY Flag (BRDY Interrupt Status Flag)

This flag indicates the state of the BRDY interrupt. Refer to section 41.2.13 for conditions of the BRDY flag becoming 1. The BRDY flag becomes 0 when 0 is written to the BRDYSTS.PIPEBRDY[9:0] flags for all pipes for which the BRDYENB.PIPEBRDYE[9:0] bits are 1 (interrupt request output enabled). The BRDY flag does not become 0 even when set to 0 by software.

NRDY Flag (NRDY Interrupt Status Flag)

This flag indicates the state of the NRDY interrupt. Refer to section 41.2.14 for conditions of the NRDY flag becoming 1. The NRDY flag becomes 0 when 0 is written to the NRDYSTS.PIPENRDY[9:0] flags for all pipes for which the NRDYENB.PIPENRDYE[9:0] bits are 1 (interrupt request output enabled). The NRDY flag does not become 0 even when set to 0 by software.

BEMP Flag (BEMP Interrupt Status Flag)

This flag indicates the state of the BEMP interrupt. Refer to section 41.2.15 for conditions of the BEMP flag becoming 1. The BEMP flag becomes 0 when 0 is written to the BEMPSTS.PIPEBEMP[9:0] flags for all pipes for which the BEMPENB.PIPEBEMPE[9:0] bits are 1 (interrupt request output enabled). The BEMP flag does not become 0 even when set to 0 by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

In function controller operation, the USBHS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt is generated, set the CTRT flag to 0 before the USBHS detects the next control transfer stage transition. Values read from the CTRT flag are invalid in host controller operation.

DVST Flag (Device State Transition Interrupt Status Flag)

In function controller operation, the USBHS updates the value of the DVSQ[2:0] bits and sets the DVST flag to 1 when a change in the device state is detected. When a device state transition interrupt is generated, set the DVST flag to 0 before the USBHS detects the next device state transition. Values read from the DVST flag are invalid in host controller operation.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) In host controller operation
When the DVSTCTR0.UACT bit is 1, the SOFR flag becomes 1 (a frame number refresh interrupt is detected every millisecond) when the frame number is updated.

(2) In function controller operation
The USBHS sets the SOFR flag to 1 (a frame number refresh interrupt is detected every millisecond) when the frame number is updated. The USBHS can detect a frame number refresh interrupt using the SOF recovery function even when a corrupted SOF packet is received from the USB host. Refer to section 41.3.13 for details on the SOF recovery function.

RESM Flag (Resume Interrupt Status Flag)

In function controller operation, the RESM flag becomes 1 when the USBHS detects the resume signal in the suspended state (DVSQ[2:0] bits are 1xxb). Values read from the RESM flag are invalid in host controller operation.

VBINT Flag (VBUS Interrupt Status Flag)

The VBINT flag becomes 1 when a level change (high to low or low to high) is detected in the USBHS_VBUS pin input signal. The USBHS sets the VBSTS flag to indicate the USBHS_VBUS pin input level. When the VBUS interrupt request is generated, eliminate chattering by reading the VBSTS flag several times by software and checking that the values read are the same.

41.2.19 Interrupt Status Register 1 (INTSTS1)

Address(es): USBHS.INTSTS1 8745 1042h, USBHS_NS.INTSTS1 0745 1042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	L1RSMEND	LPMEND	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINT	Portable Device Detection Interrupt Status Flag*1	0: Portable Device detection interrupt not requested 1: Portable Device detection interrupt requested	R/(W) *2
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag *3	0: SACK interrupt not requested 1: SACK interrupt requested	R/(W) *2
b5	SIGN	Setup Transaction Error Interrupt Status Flag*3	0: SIGN interrupt not requested 1: SIGN interrupt requested	R/(W) *2
b6	EOFERR	End-of-Frame Error Detection Interrupt Status Flag*3	0: EOFERR interrupt not requested 1: EOFERR interrupt requested	R/(W) *2
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	LPMEND	LPM Transaction End Interrupt Status Flag*3	0: LPMEND interrupt not requested 1: LPMEND interrupt requested	R/(W) *2
b9	L1RSMEND	L1 Resume End Interrupt Status Flag*3	0: L1RSMEND interrupt not requested 1: L1RSMEND interrupt requested	R/(W) *2
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	ATTCH	Device Attach Detection Interrupt Status Flag*3	0: ATTCH interrupt not requested 1: ATTCH interrupt requested	R/(W) *2
b12	DTCH	Device Detach Detection Interrupt Status Flag*3	0: DTCH interrupt not requested 1: DTCH interrupt requested	R/(W) *2
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	BCHG	Bus Change Interrupt Status Flag*3	0: BCHG interrupt not requested 1: BCHG interrupt requested	R/(W) *2
b15	OVRCR	Overcurrent Change Interrupt Status Flag*1, *3	0: OVRCR interrupt not requested 1: OVRCR interrupt requested	R/(W) *2

Note 1. If a change in the status indicated by the PDDETINT or OVRCR flag is detected even while the clock is stopped (when the LPSTS.SUSPENDM bit is 0), and if the corresponding interrupt enable bit is 1, the USBHSR interrupt is requested. Supply the PHY clock before setting these flags to 0. Interrupts indicated by flags other than the PDDETINT and OVRCR flags are not detected while the PHY clock is stopped (when the LPSTS.SUSPENDM bit is 0).
Note 2. To clear the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.
Note 3. Only enable interrupts other than the Portable Device detection interrupt in host controller operation.

PDDETINT Flag (Portable Device Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the Portable Device detection interrupt.
The PDDETINT flag becomes 1 when the USBHS detects a Portable Device. The USBHS indicates the D+ line status in the BCCTRL.PDDETSTS flag. When the Portable Device detection interrupt is generated, eliminate chattering by reading the BCCTRL.PDDETSTS flag several times by software and checking that the values read are the same.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

In host controller operation, this flag indicates the status of the setup transaction normal response interrupt (SACK interrupt). When the USBHS issues a setup transaction, if an ACK response is received from a peripheral device, the SACK flag becomes 1. At this time, if the INTENB1.SACKE bit is 1, the USBHS requests the SACK interrupt.
In function controller operation, the SACK flag read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

In host controller operation, this flag indicates the status of the setup transaction error interrupt (SIGN interrupt). When the USBHS issues a setup transaction, if an ACK response is not returned from a peripheral device three consecutive times, the SIGN flag becomes 1. At this time, if the INTENB1.SIGNE bit is 1, the USBHS requests the SIGN interrupt.
If any of the following conditions occurs three times consecutively, the USBHS detects the setup transaction error

- A timeout is detected by the USBHS when there is no response from a peripheral device
- An ACK packet is corrupted.
- A handshake other than ACK (i.e. NAK, NYET, or STALL) is received

In function controller operation, the SIGN flag read value is invalid.

EOFERR Flag (End-of-Frame Error Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the EOF error detection interrupt (EOFERR interrupt). When communication is detected as not being complete at the EOF2 timing prescribed in USB 2.0, the EOFERR flag becomes 1. At this time, if the INTENB1.EOFERRE bit is 1, the USBHS requests the EOFERR interrupt. After detecting the EOF error, regardless of the INTENB1.EOFERRE bit setting, the USBHS sets the DVSTCTR0.UACT bit to 0 and enters the Idle state. controls hardware as described below. Use software to end all communication currently being performed on pipes, and repeat enumeration.
In function controller operation, the EOFERR flag read value is invalid.

LPMEND Flag (LPM Transaction End Interrupt Status Flag)

When the HL1CTRL1.L1REQ bit is set to 1, the USBHS transmits an LPM token. When the LPM transaction is ended because a response from the peripheral device or a timeout is detected, the LPMEND flag becomes 1.
In function controller operation, the LPMEND flag read value is invalid.

L1RSMEND Flag (L1 Resume End Interrupt Status Flag)

When performing processing to exit the L1 state, the L1RSMEND flag becomes 1 when the processing is completed.
In function controller operation, the L1RSMEND flag read value is invalid.

ATTCH Flag (Device Attach Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the device attach detection interrupt (ATTCH interrupt). When the USBHS detects J state or K state of the full-speed or low-speed signal level for 2.5 µs on the port, the USBHS detects that a device has been attached, and sets the ATTCH flag to 1. At this time, if the INTENB1.ATTCHE bit is 1, the USBHS requests the USBHSR interrupt.
The conditions for the USBHS to detect a device being attached are as follows:

- The K state, SE0, or SE1 changes to the J state, and the J state continues for 2.5 µs
- The J state, SE0, or SE1 changes to the K state, and the K state continues for 2.5 µs

In function controller operation, the ATTCH flag read value is invalid.

DTCH Flag (Device Detach Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the device detach detection interrupt (DTCH interrupt).
When a USB bus device disconnect is detected, the USBHS detects device disconnection, and the DTCH flag becomes 1. At this time, if the INTENB1.DTCHE bit is set to 1, the USBHS requests the USBHSR interrupt.
The USBHS detects bus disconnection based on USB 2.0.
After device disconnect is detected, regardless of the INTENB1.DTCHE bit setting, the USBHS sets the DVSTCTR0.UACT bit to 0 and enters the Idle state. Use software to end all communication currently being performed on pipes, and transition to the wait for attach (ATTCH interrupt generated) state.
In function controller operation, the DTCH flag read value is invalid.

BCHG Flag (Bus Change Interrupt Status Flag)

This flag indicates the status of the bus change interrupt (BCHG interrupt).
When a status change to full-speed or low-speed occurs on the USB port (when the J state, K state, or SE0 changes to the J state, K state, or SE0), the USBHS detects a change in the bus status, and the BCHG flag becomes 1. At this time, if the INTENB1.BCHGE bit is 1, the USBHS requests the USBHSR interrupt.
The SYSSTS0.LNST[1:0] flags indicate the current state of the USB port. When the BCHG interrupt is generated, in order to eliminate chattering, read the LNST[1:0] flags several times by software and confirm that the values read are the same.
In function controller operation, the BCHG flag read value is invalid.

OVRCR Flag (Overcurrent Change Interrupt Status Flag)

This flag indicates the status change interrupt (OVRCR interrupt) for the USBHS_OVRCURA pin or USBHS_OVRCURB pin.
If there is a change in the input of either the USBHS_OVRCURA pin or USBHS_OVRCURB pin (from high to low or low to high), the USBHS detects a change in the overcurrent pin, and the OVRCR flag becomes 1. At this time, if the INTENB1.OVRCRE bit is 1, the USBHS requests the USBHSR interrupt.
The SYSSTS0.OVCMON[1:0] flags to indicate the current status of the USBHS_OVRCURA and USBHS_OVRCURB pins.
In host controller operation, if an overcurrent notification signal (from the power supply IC which supplies VBUS to the peripheral devices) is connected to the USBHS_OVRCURA or USBHS_OVRCURB pin, software can detect an occurrence of overcurrent.

41.2.20 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USBHS.BRDYSTS 8745 1046h, USBHS_NS.BRDYSTS 0745 1046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBRDY[9:0]	Pipe 9 to Pipe 0 BRDY Interrupt Status Flags*1	0: Interrupt not generated 1: Interrupt generated	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBRDY[9] bit, pipe 8 corresponds to the PIPEBRDY[8] bit, etc.).
Note 2. When the SOFCFG.BRDYM bit is 0, to clear the status indicated by the PIPEBRDY[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.
When the SOFCFG.BRDYM bit is set to 0, clear the BRDY interrupt before accessing the FIFO buffer.

PIPEBRDY[9:0] Flags (Pipe 9 to Pipe 0 BRDY Interrupt Status Flags)

When the BRDY interrupt is detected, the USBHS sets the bits corresponding to the PIPEBRDY[9:0] flags to 1. Refer to section 41.3.6.1 for details on the BRDY interrupt.

41.2.21 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USBHS.NRDYSTS 8745 1048h, USBHS_NS.NRDYSTS 0745 1048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPENRDY[9:0]	Pipe 9 to Pipe 0 NRDY Interrupt Status Flags*1	0: Interrupt not generated 1: Interrupt generated	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. Pipe 9 corresponds to the PIPENRDY[9] bit, pipe 8 corresponds to the PIPENRDY[8] bit, etc.).
Note 2. To clear the status indicated by the PIPENRDY[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.

PIPENRDY[9:0] Flags (Pipe 9 to Pipe 0 NRDY Interrupt Status Flags)

If an NRDY interrupt source is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS module sets the bits corresponding to the PIPENRDY[9:0] flags to 1. Refer to section 41.3.6.2 for details on the NRDY interrupt.

41.2.22 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USBHS.BEMPSTS 8745 104Ah, USBHS_NS.BEMPSTS 0745 104Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBEMP[9:0]	Pipe 9 to Pipe 0 BEMP Interrupt Status Flags*1	0: Interrupt not generated 1: Interrupt generated (n = 0 to 9)	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBEMP[9] bit, pipe 8 corresponds to the PIPEBEMP[8] bit, etc.).
Note 2. To clear the status indicated by the PIPEBEMP[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.

PIPEBEMP[9:0] Flags (Pipe 9 to Pipe 0 BEMP Interrupt Status Flags)

If a BEMP interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS module sets the bits corresponding to the PIPEBEMP[9:0] flags to 1. Refer to section 41.3.6.3 for details on the BEMP interrupt.

41.2.23 Frame Number Register (FRMNUM)

Address(es): USBHS.FRMNUM 8745 104Ch, USBHS_NS.FRMNUM 0745 104Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	Latest frame number	R
b13 to b11	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	CRCE	CRC Error Detection Status Flag	0: No error occurred 1: Error occurred	R/(W) *2
b15	OVRN	Overrun/Underrun Detection Status Flag*1	0: No error occurred 1: Error occurred	R/(W) *2

Note 1. The OVRN flag is used for debugging. Design the timing so that no overrun or underrun occurs in the system.
Note 2. When setting this flag to 0, write 0 only to the flags to be cleared; write 1 to the other flags.

FRNM[10:0] Bits (Frame Number)

An SOF packet is issued or received every millisecond. Each time an SOF packet is issued or received, the USBHS rewrites these flags to indicate the latest frame number.

CRCE Flag (CRC Error Detection Status Flag)

This flag becomes 1 when a CRC error or bit stuffing error occurs during isochronous transfer. Write 0 to the CRCE flag to clear it. When writing 0 to the CRCE flag, write 1 to the other flags in the FRMNUM register. When a CRC error is detected, the USBHS generates an internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

This flag becomes 1 when an overrun or underrun error occurs during isochronous transfer. Write 0 to the OVRN flag to clear it. When writing 0 to the OVRN flag, write 1 to the other flags in the FRMNUM register.

(1) In host controller operation
The USBHS sets the OVRN flag to 1 when either of the following conditions is satisfied.

- For pipes set to isochronous transfer for transmitting, the timing to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For pipes set to isochronous transfer for receiving, the timing to issue an IN token comes while no FIFO buffer plane is empty.

(2) In function controller operation
The USBHS sets the OVRN flag to 1 when any of the following conditions is satisfied.

- For pipes set to isochronous transfer for transmitting, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For pipes set to isochronous transfer for receiving, the OUT token is received while no FIFO buffer plane is empty.

41.2.24 Microframe Number Register (UFRMNUM)

Address(es): USBHS.UFRMNUM 8745 104Eh, USBHS_NS.UFRMNUM 0745 104Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	UFRNM[2:0]	Microframe Number	Microframe number	R
b14 to b3	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15	DVCHG	Device State Change	0: Writing to the USBADDR register is disabled. 1: Writing to the USBADDR register is enabled.	R/W

UFRNM[2:0] Bits (Microframe Number)

The USBHS sets these bits to indicate the microframe number during high-speed operation. During operation other than high-speed operation, the USBHS sets these bits to 000b. When reading these bits by software, confirm that the same value is read twice.

DVCHG Bit (Device State Change)

This bit enables or disables the writing to the USBADDR register. Set this bit to 1 when writing back the device state and device address after release from deep software standby mode 1. For details, refer to section 41.3.18, Release from Deep Software Standby Mode 1 in Response to the USBHS Resume Interrupt.

41.2.25 USB Address Register (USBADDR)

Address(es): USBHS.USBADDR 8745 1050h, USBHS_NS.USBADDR 0745 1050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	STSRECOV0[2:0]			—	USBADDR[6:0]						
Value after reset:	x	x	x	x	x	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	In function controller operation, these flags indicate the USB address assigned by the host when the USBHS processed the SetAddress() request successfully.	R/(W) *1
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b10 to b8	STSRECOV0[2:0]	Status Recovery	• In host controller operation These bits are used to rewrite the value of the DVSCTR0.RHST[2:0] bits. b10 b8 0 1 0: Set the RHST[2:0] bits to 001b (low-speed) 1 0 0: Set the RHST[2:0] bits to 010b (full-speed) 1 1 0: Set the RHST[2:0] bits to 011b (high-speed) Settings other than above are prohibited. • In function controller operation These bits are used to rewrite the value of the DVSCTR0.RHST[2:0] and INTSTS0.DVSQ[2:0] bits. b10 b8 0 0 0: Retained 0 0 1: Set the RHST[2:0] bits to 010b (full-speed) and the DVSQ[2:0] bits to 001b (Default state) 0 1 0: Set the RHST[2:0] bits to 010b (full-speed) and the DVSQ[2:0] bits to 010b (Address state) 0 1 1: Set the RHST[2:0] bits to 010b (full-speed) and the DVSQ[2:0] bits to 011b (Configured state) 1 0 0: Set the DVSQ[2:0] bits to 100b (Suspended state) 1 0 1: Set the RHST[2:0] bits to 011b (high-speed) and the DVSQ[2:0] bits to 001b (Default state) 1 1 0: Set the RHST[2:0] bits to 011b (high-speed) and the DVSQ[2:0] bits to 010b (Address state) 1 1 1: Set the RHST[2:0] bits to 011b (high-speed) and the DVSQ[2:0] bits to 011b (Configured state)	R/(W) *1
b15 to b11	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. These bits can only be written when the UFRMNUM.DVCHG is 1.

USBADDR[6:0] Bits (USB Address)

In function controller operation, these bits indicate the USB address received in the USBADDR[6:0] bits when the USBHS processed the SetAddress() request successfully. The USBADDR[6:0] bits indicate 00h when a USB bus reset is detected.

After releasing from the deep software standby mode 1, set the UFRMNUM.DVCHG bit to 1 and write back the previously saved address.

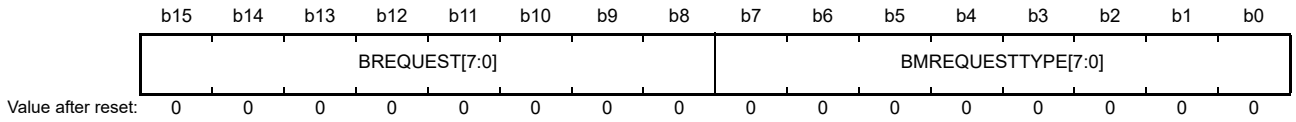
In host controller operation, these bits are not used.

STSRECOV0[2:0] Bits (Status Recovery)

These bits are used to write back the device state after release from deep software standby mode 1. For details, refer to section 41.3.18, Release from Deep Software Standby Mode 1 in Response to the USBHS Resume Interrupt.

41.2.26 USB Request Type Register (USBREQ)

Address(es): USBHS.USBREQ 8745 1054h, USBHS_NS.USBREQ 0745 1054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	USB request *bmRequestType* value	R/W*1
b15 to b8	BREQUEST[7:0]	Request	USB request *bRequest* value	R/W*1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits store the *bmRequestType* value of USB requests.

- In host controller operation
  Set these bits to the value of the USB request data in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the value of the USB request data in setup transactions to be received. Writing to these bits is invalid.

BREQUEST[7:0] Bits (Request)

These bits store the *bRequest* value of USB requests.

- In host controller operation
  Set these bits to the value of the USB request data in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the value of the USB request data in setup transactions to be received. Writing to these bits is invalid.

41.2.27 USB Request Value Register (USBVAL)

Address(es): USBHS.USBVAL 8745 1056h, USBHS_NS.USBVAL 0745 1056h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	Value of the USB request *wValue*	R/W*1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

WVALUE[15:0] Bits (Value)

These bits store the *wValue* value of USB requests.

- In host controller operation
  Set the *wValue* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wValue* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

41.2.28 USB Request Index Register (USBINDX)

Address(es): USBHS.USBINDX 8745 1058h, USBHS_NS.USBINDX 0745 1058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	Value of the USB request *wIndex*	R/W*1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

WINDEX[15:0] Bits (Index)

These bits store the *wIndex* value of USB requests.

- In host controller operation
  Set the *wIndex* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wIndex* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

41.2.29 USB Request Length Register (USBLENG)

Address(es): USBHS.USBLENG 8745 105Ah, USBHS_NS.USBLENG 0745 105Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WLENGTH[15:0]	Length	Value of the USB request *wLength*	R/W*1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

WLENGTH[15:0] Bits (Length)

These bits store the *wLength* value of USB requests.

- In host controller operation
  Set the *wLength* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wLength* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

41.2.30 Default Control Pipe Configuration Register (DCPCFG)

Address(es): USBHS.DCPCFG 8745 105Ch, USBHS_NS.DCPCFG 0745 105Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	DIR	Transfer Direction*1	0: Receive data 1: Transmit data	R/W
b6, b5	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: The pipe is continued at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
b8	CNTMD	Continuous Transfer Mode*1	0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Only change the settings of bits in the DCPCFG register when the DCPCTR.PID[1:0] bits are 00b (NAK response). Only change the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) after confirming that the value of the DCPCTR.PBUSY flag is 0. However, the PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

DIR Bit (Transfer Direction)

In host controller operation, this bit sets the transfer direction of the data stage and status stage for control transfer.
In function controller operation, set this bit to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

When the control transfer direction is receiving, this bit specifies whether the DCPCTR.PID[1:0] bits are changed from 00b (NAK response) when transfer is complete.
The SHTNAK bit is valid when the default control pipe transfer direction is receiving.
When the SHTNAK bit is 1, the DCPCTR.PID[1:0] bits become 00b (NAK response) when the USBHS transfer is complete. The USBHS determines that transfer is complete when short packet data (including the zero-length packet) is received successfully.

CNTMD Bit (Continuous Transfer Mode)

This bit specifies whether the default control pipe communicates in continuous transfer mode or not.

41.2.31 Default Control Pipe Maximum Packet Size Register (DCPMAXP)

Address(es): USBHS.DCPMAXP 8745 105Eh, USBHS_NS.DCPMAXP 0745 105Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	x	x	x	x	x	1	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size *1	These bits specify the maximum data payload (maximum packet size) for the default control pipe.	R/W
b11 to b7	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 0 1: Address 0101b 0 1 1 0: Address 0110b 0 1 1 1: Address 0111b 1 0 0 0: Address 1000b 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b Only set the values listed above.	R/W

Note 1. Only change the MXPS[6:0] bit setting when the DCPCTR.CSSTS flag is 0, the DCPCTR.PID[1:0] bits are 00b (NAK response), and the CFIFOSEL.CURPIPE[3:0] bits are 0000b. When changing the setting after changing the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response), confirm the DCPCTR.CSSTS flag and DCPCTR.PBUSY flag are 0 before modifying these bits. However, when the USBHS changes the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response), the CSSTS flag and PBUSY flag do not need to be checked by software. In addition, after the MXPS[6:0] bit setting is changed, set the CFIFOCTR.BCLR bit to 1 to perform buffer clear processing.

Note 2. Only change the DEVSEL[3:0] bit setting when the DCPCTR.CSSTS flag is 0, the DCPCTR.PID[1:0] bits are 00b (NAK response), and the DCPCTR.SUREQ bit is 0. Only change the DCPCTR.PID[1:0] bits of the default control pipe from 01b (BUF response) to 00b (NAK response) after confirming the values of the DCPCTR.CSSTS flag and the DCPCTR.PBUSY flag are 0. However, the PBUSY flag does not need to be checked by software if the USBHS changes the DCPCTR.PID[1:0] bits to 00b (NAK response).

MXPS[6:0] Bits (Maximum Packet Size)

Set the default control pipe's maximum data payload (maximum packet size) to the MXPS[6:0] bits. The initial value is 40h (64 bytes).

The value set to the MXPS[6:0] bits must be compliant with USB 2.0.

When the MXPS[6:0] bits are set to 00h, do not write to the FIFO buffer and do not set the DCPCTR.PID[1:0] bits to 01b (BUF response).

DEVSEL[3:0] Bits (Device Select)

In host controller operation, these bits specify the peripheral device address of the communication target for control transfer.

Set the DEVADDm register corresponding to the value set in the DEVSEL[3:0] bits, and then set the DEVSEL[3:0] bits (m = 0 to A). For example, to set the DEVSEL[3:0] bits to 0010b, set the address in the DEVADD2 register.

In function controller operation, set the DEVSEL[3:0] bits to 0000b.

41.2.32 Default Control Pipe Control Register (DCPCTR)

Address(es): USBHS.DCPCTR 8745 1060h, USBHS_NS.DCPCTR 0745 1060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	x	x	x	0	0	1	0	0	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the FIFO buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Completion of control transfer is disabled. 1: Completion of control transfer is enabled.	R/W
b3	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b4	PINGE	PING Token Issue Enable	0: Issuing a PING token is disabled. 1: Normal PING operation	R/W
b5	PBUSY	Pipe Busy Flag	0: The default control pipe is not used for the USB bus. 1: The default control pipe is in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*1	This bit allows setting for the expected value of the sequence toggle bit of the next transaction for data 1 in default control pipe transfer. Writing 0 has no effect. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear*1	This bit allows setting for the expected value of the sequence toggle bit of the next transaction for data 0 in default control pipe transfer. Writing 0 has no effect. This bit is read as 0.	R/W
b10, b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11	SUREQCLR	SUREQ Bit Clear	In host controller operation, the SUREQ bit becomes 0 when this bit is set to 1. Writing 0 has no effect. This bit is read as 0.	R/W
b12	CSSTS	CSSTS Status Flag	0: Start-split (SSPLIT) transaction being processed, or split transaction unused device being processed 1: Complete-split (CSPLIT) transaction being processing	R
b13	CSCLR	CSSTS Status Flag Clear	In host controller operation, for transfers that use the SSPLIT transaction, the CSSTS flag becomes 0 by setting this bit to 1. Subsequently, the next default control pipe transfer restarts from the SSPLIT. Writing 0 has no effect. This bit is read as 1.	R/W
b14	SUREQ	SETUP Token Transmission	In host controller operation, setting this bit to 1 transmits the setup packet. Writing 0 has no effect.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only write 1 to bits SQSET, SQCLR, and PINGE while the DCPCTR.CSSTS flag is 0 and the DCPCTR.PID[1:0] bits are 00b (NAK response). Only change the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) after confirming that the value of the DCPCTR.CSSTS flag and DCPCTR.PBUSY flag are 0. However, the PBUSY flag does not need to be checked by software if the USBHS changes the DCPCTR.PID[1:0] bits to 00b (NAK response).

PID[1:0] Bits (Response PID)

The DCPCTR.PID[1:0] bits control the USB response for control transfer.

(1) In host controller operation

Perform the following to modify the DCPCTR.PID[1:0] bits from 00b (NAK response) to 01b (BUF response).

- When transmitting
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and the DCPCTR.PID[1:0] bits are 00b (NAK response), and then set the DCPCTR.PID[1:0] bits to 01b (BUF response). After the DCPCTR.PID[1:0] bits have been set to 01b (BUF response), the USBHS executes the OUT transaction (or PING transaction).
- When receiving
  Confirm that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and the DCPCTR.PID[1:0] bits are 00b (NAK response), and then set the DCPCTR.PID[1:0] bits to 01b (BUF response). After the DCPCTR.PID[1:0] bits have been set to 01b (BUF response), the USBHS executes the IN transaction.

The USBHS modifies the value of the DCPCTR.PID[1:0] bits in the following cases.

- When the DCPCTR.PID[1:0] bits are set to 01b (BUF response) and the USBHS has received data exceeding the maximum packet size, the USBHS sets the DCPCTR.PID[1:0] bits to 11b (STALL response).
- When a reception error such as CRC error is detected three times in succession, the USBHS sets the DCPCTR.PID[1:0] bits to 00b (NAK response).
- When a STALL handshake is received, the USBHS sets the DCPCTR.PID[1:0] bits to 11b (STALL response).

(2) In function controller operation

The USBHS modifies the value of the DCPCTR.PID[1:0] bits in the following cases.

- When a setup packet is received, the USBHS modifies the DCPCTR.PID[1:0] bits to 00b (NAK response). At this time, the USBHS sets the INTSTS0.VALID flag to 1, and the DCPCTR.PID[1:0] bits cannot be modified by software until the VALID flag is set to 0 by software.
- When the DCPCTR.PID[1:0] bits are set to 01b (BUF response) and the USBHS receives data exceeding the maximum packet size, the USBHS sets the DCPCTR.PID[1:0] bits to 11b (STALL response).
- When the USBHS detects a control transfer sequence error, the DCPCTR.PID[1:0] bits are set to 10b or 11b (STALL response).
- When the USBHS detects a USB bus reset, the DCPCTR.PID[1:0] bits are set to 00b (NAK response).

The USBHS does not check the setting of the DCPCTR.PID[1:0] bits during the SetAddress() request processing (automatic processing).

CCPL Bit (Control Transfer End Enable)

In function controller operation, setting this bit to 1 enables completion of the status stage of the control transfer.
If the CCPL bit is set to 1 while the corresponding DCPCTR.PID[1:0] bits are 01b (BUF response), the USBHS completes the status stage of the control transfer.
During a control read transfer, the USBHS transmits an ACK handshake for the OUT transaction from the USB host.
During a control write or no-data control transfer, the USBHS transmits a zero-length packet for the IN transaction from the USB host. However, when the SetAddress() request is detected, the USBHS performs automatic responses from the SETUP stage until the status stage is completed regardless of the setting of the CCPL bit.
When a new setup packet is received, the USBHS modifies the CCPL bit from 1 to 0.
When the INTSTS0.VALID flag is 1, the CCPL bit cannot be set to 1.
The CCPL bit is initialized by the USB bus reset.
In host controller operation, set the CCPL bit to 0.

PINGE Bit (PING Token Issue Enable)

In host controller operation, when the PINGE bit is set to 1 by software, the USBHS issues the PING token based on transfer in the transmit direction. Transfer in the transmit direction starts from the PING transaction.
If an ACK handshake is detected in the PING transaction, the next transaction issued is an OUT transaction.
If a NAK handshake or NYET handshake is detected in the OUT transaction, the next transaction issued is a PING transaction.
In host controller operation, when the PINGE bit is set to 0 by software, the USBHS does not issue the PING token based on transfer in the transmit direction. All transfers in the transmit direction are issued by the OUT transaction.

PBUSY Flag (Pipe Busy Flag)

When the USBHS starts a USB transaction using the default control pipe, the USBHS changes the PBUSY flag from 0 to 1. The USBHS switches the PBUSY flag back to 0 at the end of the transaction.
After the DCPCTR.PID[1:0] bits are set to 00b (NAK response), the read value of the PBUSY flag indicates whether or not changes to pipe settings can proceed. Refer to section 41.3.7.1 for details.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

This flag indicates the value of the sequence toggling bit for the next transaction in default control pipe transfer. When a transaction is processed successfully, the USBHS toggles the SQMON flag. However, the SQMON flag is not toggled when data PID mismatch occurs during data reception. In function controller operation, the USBHS sets the SQMON flag to 1 (setting the expected value for DATA1) when a setup packet is received successfully. In addition, in function controller operation, the USBHS does not check the SQMON flag during IN or OUT transactions in the status stage, and the flag is not toggled even at the end of a successful processing.

SQSET Bit (Sequence Toggle Bit Set)

This bit sets the value of the sequence toggling bit to DATA1 for the next transaction in default control pipe transfer. Do not set bits SQCLR and SQSET to 1 at the same time.

SQCLR Bit (Sequence Toggle Bit Clear)

This bit sets the value of the sequence toggling bit to DATA0 for the next transaction in default control pipe transfer. Do not set bits SQCLR and SQSET to 1 at the same time.

SUREQCLR Bit (SUREQ Bit Clear)

In host controller operation, the SUREQ bit can be cleared by setting the SUREQCLR bit to 1. If transfer stops while the SUREQ bit is set to 1 in a SETUP transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a successful setup transaction because the USBHS will automatically set the SUREQ bit to 0. Use the SUREQCLR bit to control the SUREQ bit when transfer stops while the DVSTCTR0.UACT bit is set to 0 or transfer is not proceeding because detaching of the port has been detected. Set the SUREQCLR bit to 0 in function controller operation.

CSSTS Flag (CSSTS Status Flag)

In host controller operation, the CSSTS flag indicates the transaction status of the complete-split transaction for non-isochronous transfers. When a CSPLIT transaction starts, the USBHS sets the CSSTS flag to 1, and when the CSPLIT transaction is detected as being complete, the CSSTS flag is set to 0.
In function controller operation, the value read from the CSSTS flag is not valid.

CSCLR Bit (CSSTS Status Flag Clear)

In host controller mode, when the CSCLR bit is set to 1, the CSSTS flag becomes 0. For transfers that use the split transaction, when the next transfer is forced to restart from the start-split, set the CSCLR bit to 1. In a normal split transaction, when the CSPLIT transaction is complete, the USBHS automatically sets the CSSTS flag to 0, so the flag does not need to be set to 0 by software. Use the CSCLR bit to set the CSSTS flag to 0, when the DVSTCTR0.UACT bit is 0, or when it is certain that transfer is not performed when a detach is detected. When the CSSTS flag is 0, it remains set to 0 even if the CSCLR bit is set to 1.
In function controller operation, set the CSCLR bit to 0.

SUREQ Bit (SETUP Token Transmission)

In host controller operation, a setup packet is transmitted when the SUREQ bit is set to 1. After the SETUP transaction processing is completed, the USBHS generates the SACK interrupt or SIGN interrupt and sets the SUREQ bit to 0. When the SUREQCLR bit is set to 1, the SUREQ bit becomes 0.
Set the SUREQ bit to 1 after a USB request to be transmitted in the setup transaction is set in the DCPMAXP.DEVSEL[3:0] bits and registers USBREQ, USBVAL, USBINDX, and USBLENG. Before setting the SUREQ bit to 1, confirm that the DCPCTR.PID[1:0] bits are set to 00b (NAK response). In addition, after the SUREQ bit is set to 1, until the setup transaction ends (SUREQ bit is 1), do not modify the DCPMAXP.DEVSEL[3:0] bits or values of registers USBREQ, USBVAL, USBINDX, and USBLENG. Set the SUREQ bit to 1 only when issuing a SETUP token. In other cases, be sure to write 0 to this bit. Set the SUREQ bit to 0 in function controller operation.

BSTS Flag (Buffer Status Flag)

This flag indicates whether the default control pipe’s FIFO buffer can be accessed.
The meaning of the BSTS flag varies depending on the setting of the CFIFOSEL.ISEL bit.

- When the ISEL bit is 0, the BSTS flag indicates whether receive data can be read.
- When the ISEL bit is 1, the BSTS flag indicates whether transmit data can be written.

41.2.33 Pipe Window Select Register (PIPESEL)

Address(es): USBHS.PIPESEL 8745 1064h, USBHS_NS.PIPESEL 0745 1064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b15 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Set pipes 1 to 9 using registers PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPEnCTR (n = 1 to 9), PIPEnTRE (n = 1 to 5), and PIPEnTRN (n = 1 to 5). After selecting the pipe to be set by the PIPESEL register, each pipe setting is performed by the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. In addition, the PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection by the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

When the pipe number is specified in the PIPESEL[3:0] bits, the pipe information is indicated in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. Perform the pipe setting by modifying these registers.
When the PIPESEL[3:0] bits are set to 0000b, reading bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers returns 0. Writing has no effect.

41.2.34 Pipe Configuration Register (PIPECFG)

Address(es): USBHS.PIPECFG 8745 1068h, USBHS_NS.PIPECFG 0745 1068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	x	x	x	0	0	0	0	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. 0000b indicates the pipe is not used.	R/W
b4	DIR	Transfer Direction*2, *3	0: Receive data 1: Transmit data	R/W
b6, b5	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: The pipe is continued at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
b8	CNTMD	Continuous Transfer Mode*2, *3	0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
b9	DBLB	Double Buffer Mode*2, *3	0: Single buffer mode 1: Double buffer mode	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2, *3	0: BRDY interrupt when data is transmitted or received 1: BRDY interrupt when all data has been read	R/W
b13 to b11	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• For pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Do not set this value. 1 1: Isochronous transfer • For pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Do not set this value. 1 1: Do not set this value. • For pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Do not set this value. 1 0: Interrupt transfer 1 1: Do not set this value.	R/W

Note 1. Only change the settings of the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.CSSTS flag is 0, PIPEnCTR.PBUSY flag is 0, and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

Note 2. Only change the setting of the BFRE, DBLB, CNTMD, and DIR bits while the PIPEnCTR.CSSTS flag is 0, PIPEnCTR.PBUSY flag is 0, and the PIPEnCTR.PID[1:0] bits are 00b (NAK response) with the pipe not specified in the CURPIPE[3:0] bits of a FIFO port select register. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.CSSTS flag and PIPEnCTR.PBUSY flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

Note 3. When the setting of the BFRE, DBLB, CNTMD, and DIR bits are modified after USB communications have been carried with the selected pipe, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously by software and clear the FIFO buffer assigned to the selected pipe, in addition to the note described in Note 2.

EPNUM[3:0] Bits (Endpoint Number)

These bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used. Set these bits so that the combination of the settings of the DIR and EPNUM[3:0] bits is different from the settings of other pipes. (The EPNUM[3:0] bits being set to 0000b can be set for all pipes.)

DIR Bit (Transfer Direction)

This bit specifies the transfer direction for the selected pipe. When the DIR bit is set to 0, the USBHS uses the selected pipe to receive data. When the DIR bit is set to 1, the USBHS uses the selected pipe to transmit data.

SHTNAK Bit (Pipe Disabled at End of Transfer)

This bit selects whether to modify the PIPEnCTR.PID[1:0] bits to 00b (NAK response) at the end of transfer when receiving direction is set for the selected pipe. This bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction. When software processing sets this bit to 1 for a pipe in the receiving direction, the USBHS modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK response) on determining the end of the transfer. The USBHS determines that the transfer has ended from the following conditions.

- Short packet data (including a zero-length packet) is received successfully.
- The transaction counter is used and packets of the number specified by the transaction counter are received successfully.

CNTMD Bit (Continuous Transfer Mode)

This bit is valid when the selected pipe is pipe 1 to 5 of the bulk transfer type. According to the setting of this bit, the USBHS determines the completion of transmission or reception for the FIFO buffer allocated to the selected pipe as shown in Table 41.8.

Table 41.8 Relationship Between the CNTMD Bit Setting and Methods for Determining Completion of FIFO Buffer Transmission/Reception

CNTMD Bit Setting	Method for Determining Readability and Transmittability
0	Conditions for FIFO buffer readable state when receiving direction is set (DIR bit is 0): The USBHS has received one packet.
	Conditions for FIFO buffer transmittable state when transmitting direction is set (DIR bit is 1): When (1) or (2) is satisfied. (1) CPU (or DMAC/DTC) has written data of the maximum packet size to the FIFO buffer. (2) CPU (or DMAC/DTC) has written data of the short packet size (including 0 bytes) to the FIFO buffer and has set the BVAL bit in the FIFO port control register to 1.
1	Conditions for FIFO buffer readable state when receiving direction is set (DIR bit is 0): (1) The byte count of data received in the FIFO buffer allocated to the selected pipe is equal to the allocated byte count ((PIPEBUF.BUFSIZE[4:0] + 1) × 64). (2) While there is no data stored in the FIFO buffer allocated to the selected pipe, the USBHS has received a short packet except for a zero-length packet. (3) The USBHS has received a zero-length packet when data is already contained in the FIFO buffer allocated to the selected pipe. (4) The number of packets specified by the transaction counter set for the selected pipe have been received.
	Conditions for FIFO buffer transmittable state when transmitting direction is set (DIR bit is 1): When (1) or (2) is satisfied. (1) CPU (or DMAC/DTC) has written data that is equal to the size of the FIFO buffer allocated to the selected pipe. (2) CPU (or DMAC/DTC) has written data of size smaller than the size of the FIFO buffer allocated to the selected pipe (including 0 bytes) and has set the BVAL bit in the FIFO port control register to 1.

DBLB Bit (Double Buffer Mode)

This bit specifies single buffer or double buffer for the FIFO buffer used by the selected pipe. This bit is valid when the selected pipe is pipe 1 to 5. When this bit is set to 1, the USBHS allocates twice the FIFO buffer size specified by the PIPEBUF.BUFSIZE[4:0] bits for the selected pipe. The FIFO buffer size that the USBHS allocates to the selected pipe is (PIPEBUF.BUFSIZE[4:0] + 1) × 64 × (DBLB + 1) [bytes].

BFRE Bit (BRDY Interrupt Operation Specification)

This bit specifies the timing for transmitting the BRDY interrupt. When this bit is set to 1 and the selected pipe is used in the receiving direction, the USBHS detects the end of the transfer and issues a BRDY interrupt upon reading the packet. When a BRDY interrupt is generated with this setting, set the BCLR bit in a FIFO port control register to 1. Until the BCLR bit is set to 1, the FIFO buffer allocated to the selected pipe does not become the receivable state. When software processing sets this bit to 1 and the selected pipe is used in the transmitting direction, the USBHS generates no BRDY interrupt. Refer to section 41.3.6.1 for details.

TYPE[1:0] Bits (Transfer Type)

These bits specify the transfer type of the pipe specified by the PIPESEL.PIPESEL[3:0] bits. Before modifying the PIPEnCTR.PID[1:0] bits of the selected pipe to 01b (BUF response) (before starting the USB communication with the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

41.2.35 Pipe Buffer Register (PIPEBUF)

Address(es): USBHS.PIPEBUF 8745 106Ah, USBHS_NS.PIPEBUF 0745 106Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	x	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description			R/W
b7 to b0	BUFNMB[7:0]	Buffer Number*1	These bits specify the start block of the FIFO buffer allocated to the selected pipe (04h to 87h).			R/W
b9, b8	—	Reserved	These bits are undefined when read. Set them to 0 when writing.			R/W
b14 to b10	BUFSIZE[4:0]	Buffer Size*1	00h: 64 bytes 01h: 128 bytes 02h: 192 bytes 03h: 256 bytes 04h: 320 bytes 05h: 384 bytes 06h: 448 bytes 07h: 512 bytes 08h: 576 bytes 09h: 640 bytes 0Ah: 704 bytes	0Bh: 768 bytes 0Ch: 832 bytes 0Dh: 896 bytes 0Eh: 960 bytes 0Fh: 1,024 bytes 10h: 1,088 bytes 11h: 1,152 bytes 12h: 1,216 bytes 13h: 1,280 bytes 14h: 1,344 bytes 15h: 1,408 bytes	16h: 1,472 bytes 17h: 1,536 bytes 18h: 1,600 bytes 19h: 1,664 bytes 1Ah: 1,728 bytes 1Bh: 1,792 bytes 1Ch: 1,856 bytes 1Dh: 1,920 bytes 1Eh: 1,984 bytes 1Fh: 2,048 bytes	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.			R/W

Note 1. Only change the settings of bits in the PIPEBUF register while the PIPEnCTR.CSSTS flag is 0, PIPEnCTR.PID[1:0] bits are 00b (NAK response), and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

BUFNMB[7:0] Bits (Buffer Number)

Specify the first block number of the FIFO buffer to be allocated to the selected pipe. Blocks of the FIFO buffer that the USBHS allocates to the selected pipe are shown below.
Block number: BUFNMB[7:0] to block number: BUFNMB[7:0] + (PIPEBUF.BUFSIZE[4:0] + 1) × (DBLB + 1) -1

Set the BUFNMB[7:0] bits to a value that observes the conditions below, and has a allocated block number between 04h and 87h (8.5 Kbytes of the on-chip memory).

- 00h is for default control pipe only.
- 04h is for pipe 6 only, but is available for other pipes when pipe 6 is not used. When pipe 6 is selected, writing a value to these bits is disabled. The USBHS automatically allocates 04h to the BUFNMB[7:0] bits for pipe 6.
- 05h is for pipe 7 only, but is available for other pipes when pipe 7 is not used. When pipe 7 is selected, writing a value to these bits is disabled. The USBHS automatically allocates 05h to the BUFNMB[7:0] bits for pipe 7.
- 06h is for pipe 8 only, but is available for other pipes when pipe 8 is not used. When pipe 8 is selected, writing a value to these bits is disabled. The USBHS automatically allocates 06h to the BUFNMB[7:0] bits for pipe 8.
- 07h is for pipe 9 only, but is available for other pipes when pipe 9 is not used. When pipe 9 is selected, writing a value to these bits is disabled. The USBHS automatically allocates 07h to the BUFNMB[7:0] bits for pipe 9.

BUFSIZE[4:0] Bits (Buffer Size)

Specify the FIFO buffer size (number of blocks) to be allocated to the selected pipe. One block is 64 bytes.
When the DBLB bit is set to 1, the USBHS allocates twice the FIFO buffer size specified by these bits for the selected pipe. Setting the DBLB bit to 1 is effective only when the selected pipe is pipe 1 to 5. The FIFO buffer size that the USBHS allocates to the selected pipe is shown below.

(BUFSIZE[4:0] + 1) × 64 × (DBLB + 1) [bytes]

Set a value within the following range for the BUFSIZE[4:0] bits.

(1) When the selected pipe is pipe 1 to 5, a value of 00h to 1Fh can be set (up to 2 Kbytes).
(2) When the selected pipe is pipe 6 to 9, only a value of 00h (64 bytes) can be set.

41.2.36 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USBHS.PIPEMAXP 8745 106Ch, USBHS_NS.PIPEMAXP 0745 106Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	MXPS[10:0]										
Value after reset:	0	0	0	0	x	0	0	0	0	0*1	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	MXPS[10:0]	Maximum Packet Size *2	• For pipes 1 and 2 1 byte (001h) to 1,024 bytes (400h) • For pipes 3 to 5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h), 512 bytes (200h) • For pipes 6 to 9 1 byte (001h) to 64 bytes (040h)	R/W
b11	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b15 to b12	DEVSEL[3:0]	Device Select *3	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 1 0: Address 0110b 0 1 1 1: Address 0111b 1 0 0 0: Address 1000b 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b 1011 to 1111: Reserved	R/W

Note 1. The initial value of the MXPS[10:0] bits for each pipe is 040h.

Note 2. Only change the settings of the MXPS[10:0] bits while the PIPEnCTR.CSSTS flag is 0, PIPEnCTR.PID[1:0] bits are 00b (NAK response), and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

Note 3. Only change the settings of the DEVSEL[3:0] bits while the PIPEnCTR.CSSTS flag is 0 and PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

MXPS[10:0] Bits (Maximum Packet Size)

These bits set the maximum data payload (maximum packet size) for the selected pipe. Set the value of the MXPS[10:0] bits compliant with USB 2.0 for each transfer type. While the MXPS[10:0] bits are 000h, do not write data to the FIFO buffer or set the PIPEnCTR.PID[1:0] bits to 01b (BUF response). When the transfer type is an isochronous pipe, and when performing split transaction transfer, set the MXPS[10:0] bits to a value that is 188 bytes or less.

DEVSEL[3:0] Bits (Device Select)

In host controller operation, set the address of the communication target USB device for these bits. Set these bits after the settings for the DEVADDm register corresponding to the set value of the DEVSEL[3:0] bits has been made (m = 0 to A). When setting the DEVSEL[3:0] bits to 0010b, for example, set the DEVADD2 register. In function controller operation, set the DEVSEL[3:0] bits to 0000b.

41.2.37 Pipe Cycle Control Register (PIPEPERI)

Address(es): USBHS.PIPEPERI 8745 106Eh, USBHS_NS.PIPEPERI 0745 106Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	These bits specify the interval error detection interval for the selected pipe as 2^n of the frame timing.	R/W
b11 to b3	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Only change the settings of the IITV[2:0] bits while the PIPEnCTR.CSSTS flag is 0 and PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

IITV[2:0] Bits (Interval Error Detection Interval)

To modify these bits to another value after these bits have been set and USB communication has been performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK response) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer. The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits corresponding to pipes 3 to 5.

The IITV[2:0] bit function differs in host controller operation and function control operation. Refer to section 41.3.11 and section 41.3.12 for details.

IFIS Bit (Isochronous IN Buffer Flush)

This bit specifies whether to flush the buffer when the pipe specified by the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfer.

In function controller operation, this bit provides a buffer flush function for the USBHS to automatically clear the FIFO buffer if the USBHS does not receive IN token from the USB host in (micro)frames at each interval set by the IITV[2:0] bits for the selected pipe of isochronous transfer type in the IN transfer direction.

When double buffer is specified (PIPECFG.DBLB bit is 1), the USBHS only clears the data in the plane used earlier.

The USBHS clears the FIFO buffer on receiving the (μ)SOF packet immediately after the (micro)frame in which the USBHS has expected to receive IN token. Even if the (μ)SOF packet is corrupted, the FIFO buffer is cleared by the recovery function at the timing when the (μ)SOF packet should be received.

In host controller operation, set the IITV[2:0] bits to 000b.

Set the IITV[2:0] bits to 000b when the selected pipe is not for isochronous transfer.

41.2.38 Pipe n Control Register (PIPEnCTR) (n = 1 to 9)

Address(es): USBHS.PIPE1CTR 8745 1070h, USBHS.PIPE2CTR 8745 1072h, USBHS.PIPE3CTR 8745 1074h, USBHS.PIPE4CTR 8745 1076h, USBHS.PIPE5CTR 8745 1078h, USBHS.PIPE6CTR 8745 107Ah, USBHS.PIPE7CTR 8745 107Ch, USBHS.PIPE8CTR 8745 107Eh, USBHS.PIPE9CTR 8745 1080h,
USBHS_NS.PIPE1CTR 0745 1070h, USBHS_NS.PIPE2CTR 0745 1072h, USBHS_NS.PIPE3CTR 0745 1074h, USBHS_NS.PIPE4CTR 0745 1076h, USBHS_NS.PIPE5CTR 0745 1078h, USBHS_NS.PIPE6CTR 0745 107Ah, USBHS_NS.PIPE7CTR 0745 107Ch, USBHS_NS.PIPE8CTR 0745 107Eh, USBHS_NS.PIPE9CTR 0745 1080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	x	x	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b5	PBUSY	Pipe Busy Flag	0: Pipe n is not used for the USB bus. 1: Pipe n is in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*1	This bit is set to 1 when the expected value of the sequence toggle bit for the next transaction of pipe n is set for DATA1. Writing 0 has no effect. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear*1	This bit is set to 1 when the expected value of the sequence toggle bit for the next transaction of pipe n is returned to DATA0. Writing 0 has no effect. This bit is read as 0.	R/W
b9	ACLRM	Auto Buffer Clear Mode*2	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*1, *3	0: Auto response mode is disabled. 1: Auto response mode is enabled	R/W
b11	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b12	CSSTS	CSSTS Status Flag	0: SSPLIT transaction being processing, or split transaction unused device being processed 1: CSPLIT transaction being processed	R
b13	CSCLR	CSPLIT Status Clear	Set this bit to 1 to set the CSSTS flag for pipe n to 0. Writing 0 has no effect.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag*3	0: No transmittable data is present in the FIFO buffer. 1: Transmittable data is present in the FIFO buffer.	R
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only change the settings of the ATREPM bit and write 1 to the SQCLR or SQSET bit while the PIPEnCTR.CSSTS flag is 0 and PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PIPEnCTR.PID[1:0] bits to 00b (NAK response).

Note 2. Only change the settings of the ACLRM bit while the PIPEnCTR.CSSTS flag is 0, PIPEnCTR.PID[1:0] bits are 00b (NAK response), and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

Note 3. The ATREPM bit and the INBUFM flag in registers PIPE6CTR to PIPE9CTR are reserved. This bit is undefined when read. Set it to 0 when writing.

PID[1:0] Bits (Response PID)

These bits specify the response type for the next transaction of pipe n.
The initial setting of the PIPEnCTR.PID[1:0] bits is 00b (NAK response). Modify these bits to 01b (BUF response) to use pipe n for USB transfer. The basic operation (with no error in communication packets) of the USBHS according to the setting of the PIPEnCTR.PID[1:0] bits is shown in Table 41.9 and Table 41.10.
After modifying the setting of the PIPEnCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) during communication using pipe n, check that the PBUSY flag is 0 to confirm that the USB transfer using pipe n has actually entered the NAK state.

The USBHS modifies the PIPEnCTR.PID[1:0] bits in the following cases.

- If pipe n is in the receiving direction, and the PIPECFG.SHTNAK bit for the selected pipe is set to 1, when the USBHS sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on recognizing the completion of transfer.
- The USBHS sets the PIPEnCTR.PID[1:0] bits to 11b (STALL response) on receiving a data packet with a payload exceeding the maximum packet size of pipe n.
- The USBHS sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on detecting a USB bus reset in function controller operation.
- The USBHS sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on detecting a reception error, such as a CRC error, three consecutive times in host controller operation.
- The USBHS sets the PIPEnCTR.PID[1:0] bits to 11b (STALL response) on receiving the STALL handshake in host controller operation.

The PIPEnCTR.PID[1:0] bits should be set according to the procedure below.

- To make a transition from 00b (NAK response) to 10b (STALL response), write 10b.
- To make a transition from 01b (BUF response) to 11b (STALL response), write 11b.
- To make a transition from 11b (STALL response) to 00b (NAK response), write 10b and then 00b.
- To make a transition from 11b (STALL response) to 01b (BUF response), modify the bits to 00b (NAK response) and then to 01b (BUF response).

Table 41.9 USBHS Operation According to the PIPEnCTR.PID[1:0] Bit Setting (In Host Controller Operation)

PID[1:0] Bit Value	Transfer Type (TYPE[1:0] Bit Value)	Transfer Direction (DIR Bit Value)	USBHS Operation
00b (NAK response)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.
01b (BUF response)	Bulk or Interrupt	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1, regardless of the state of the FIFO buffer corresponding to pipe n. Does not issue tokens when the UACT bit is 0.
10b or 11b (STALL response)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.

Table 41.10 USBHS Operation According to the PIPEnCTR.PID[1:0] Bit Setting (In Function Controller Operation)

PID[1:0] Bit Value	Transfer Type (TYPE[1:0] Bit Value)	Transfer Direction (DIR Bit Value)	USBHS Operation
00b (NAK response)	Bulk or Interrupt	Does not depend on the setting.	Returns a NAK response to the token from the USB host.
	Isochronous	Receiving direction (DIR bit is 0)	Returns no response to the token from the USB host.
		Transmitting direction (DIR bit is 1)	Transmits a zero-length packet in response to the token from the USB host.
01b (BUF response)	Bulk	Receiving direction (DIR bit is 0)	Receives data and returns an ACK or NYET response in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Otherwise, returns a NAK response. When there is a PING token from the USB host, if the FIFO buffer for the selected pipe is receive enabled, ACK is returned. If the FIFO buffer is not receive enabled, then NAK is returned.
	Interrupt	Receiving direction (DIR bit is 0)	Receives data and returns an ACK response in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Otherwise, returns a NAK response.
	Bulk or Interrupt	Transmitting direction (DIR bit is 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Otherwise, returns a NAK response.
	Isochronous	Receiving direction (DIR bit is 0)	Receives data in response to the OUT token from the USB host if the corresponding FIFO buffer is ready for reception. Otherwise, discards data.
		Transmitting direction (DIR bit is 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b or 11b (STALL response)	Bulk or Interrupt	Does not depend on the setting.	Returns a STALL response to the token from the USB host.
	Isochronous	Does not depend on the setting.	Returns no response to the token from the USB host.

PBUSY Flag (Pipe Busy Flag)

This flag indicates whether pipe n is being used for the current transaction. The USBHS modifies the PBUSY flag from 0 to 1 at the beginning of a USB transaction for pipe n, and modifies the PBUSY flag from 1 to 0 upon successful completion of one transaction. After setting the PIPEnCTR.PID[1:0] bits to 00b (NAK response), reading the PBUSY flag allows checking whether modification of the pipe setting is possible. Refer to section 41.3.7.1 for details.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

This flag indicates the value of the sequence toggle bit for the next transaction of pipe n. When pipe n is not the isochronous transfer type, the USBHS toggles the SQMON flag upon successful completion of the transaction. However, the USBHS does not toggle the SQMON flag when a data PID inconsistency occurs during transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

This bit is set to 1 when the value of the sequence toggle bit for the next transaction of pipe n is set for DATA1. When the SQSET bit is set to 1, the USBHS sets the expected value of the sequence toggle bit of pipe n for DATA1.

SQCLR Bit (Sequence Toggle Bit Clear)

This bit is set to 1 when the value of the sequence toggle bit for the next transaction of pipe n is returned to DATA0. When the SQCLR bit is set to 1, the USBHS sets the expected value of the sequence toggle bit of pipe n for DATA0.

ACLRM Bit (Auto Buffer Clear Mode)

This bit enables or disables auto buffer clear mode for pipe n. To completely clear the information in the FIFO buffer allocated to pipe n, write 1 and then 0 to the ACLRM bit continuously. Also, allow 100 ns between modifying the ACLRM bit from 1 to 0 as sequence execution time for the internal hardware. The following lists the information cleared by the USBHS by writing 1 and 0 continuously to the ACLRM bit.

(a) All data in the FIFO buffer allocated to pipe n (both planes in double buffer mode).
(b) The interval counter value when the transfer type for pipe n is isochronous transfer or interrupt transfer.

Perform this processing when the following conditions are satisfied:

(a) When all data in the FIFO buffer allocated to pipe n needs to be cleared.
(b) When the interval counter value needs to be reset.
(c) When modifying the PIPECFG.DIR, DBLB, or BFRE bit.
(d) When the transaction count function is forcibly ended.

ATREPM Bit (Auto Response Mode)

This bit enables or disables the auto response of pipe n.
When the transfer type of pipe n is set to bulk IN function controller operation, this bit can be set to 1.
When this bit is set to 1, the USBHS responds as follows to the token from the USB host.

(1) When pipe n is bulk IN transfer type (PIPECFG.TYPE[1:0] bits are 01b and the PIPECFG.DIR bit is 1)
When the ATREPM bit is 1 and the PIPEnCTR.PID[1:0] bits are 01b (BUF response), the USBHS transmits a zero-length packet in response to IN token. The USBHS updates (toggles) the sequence toggle bit (data PID) each time the USBHS receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received). In this case, the USBHS generates no BRDY interrupt or BEMP interrupt.

(2) When pipe n is bulk OUT transfer type (PIPECFG.TYPE[1:0] bits are 01b and the PIPECFG.DIR bit is 0)
When the ATREPM bit is 1 and the PIPEnCTR.PID[1:0] bits are 01b (BUF response), the USBHS returns a NAK response to OUT token or PING token and generates an NRDY interrupt. When performing USB communication with this bit set to 1, set this bit while the FIFO buffer is empty. Do not write data to the FIFO buffer during USB communication with this bit set to 1. Be sure to set this bit to 0 when pipe n is used for isochronous transfer. In host controller operation, be sure to set this bit to 0.

CSSTS Flag (CSSTS Status Flag)

In host controller operation, the CSSTS flag indicates the transaction status of the complete-split transaction for non-isochronous transfers. When a CSPLIT transaction starts, the USBHS sets the CSSTS flag to 1, and when the CSPLIT transaction is detected as being complete, the CSSTS flag is set to 0. When a detach occurs during the CSPLIT transaction, the CSSTS flag may remain set to 1. In this case, set the CSCLR bit to 1 and set the CSSTS flag to 0.
In function controller operation, the value read from the CSSTS flag is invalid.

CSCLR Bit (CSPLIT Status Clear)

In host controller mode, when the CSCLR bit is set to 1, the CSSTS flag becomes 0. For transfers using a split transaction, when the next transfer is forced to restart from the start-split, set the CSCLR bit to 1. In a normal split transaction, when the CSPLIT transaction is complete, the USBHS automatically sets the CSSTS flag to 0, so the flag does not need to be set to 0 by software. Use the CSCLR bit to set the CSSTS flag to 0 when the DVSTCTR0.UACT bit is 0, or when it is certain that transfer is not performed when a detach is detected. When the CSSTS flag is 0, it remains set to 0 even if the CSCLR bit is set to 1. In function controller operation, set the CSCLR bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

This flag indicates the FIFO buffer status for pipe n in the transmitting direction. With pipe n set in the transmitting direction (PIPECFG.DIR bit is 1), the USBHS sets the INBUFM flag to 1 when the CPU or DMAC/DTC has completed writing data to at least one FIFO buffer plane. The USBHS sets the INBUFM flag to 0 when the USBHS has completely transmitted the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (PIPECFG.DBLB bit is 1), the USBHS sets the INBUFM flag to 0 when the USBHS has completely transmitted data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane. The INBUFM flag indicates the same value as the BSTS flag when pipe n is in the receiving direction (PIPECFG.DIR bit is 0).

BSTS Flag (Buffer Status Flag)

This flag indicates the FIFO buffer status for pipe n. The meaning of the BSTS flag varies as shown in Table 41.11 depending on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DCLRM bit in the D0FIFOSEL/ D1FIFOSEL register.

Table 41.11 Operation of the BSTS Flag

DIR Bit Value	BFRE Bit Value	DCLRM Bit Value	Meaning of the BSTS Flag
0	0	0	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 upon completion of data read.
		1	Do not set this combination.
	1	0	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 when the BCLR bit is set in a FIFO port control register to 1 after the data read has been completed.
		1	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 upon completion of data read.
1	0	0	Becomes 1 when transmit data can be written to the FIFO buffer, and becomes 0 upon completion of data write.
		1	Do not set this combination.
	1	0	Do not set this combination.
		1	Do not set this combination.

41.2.39 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USBHS.PIPE1TRE 8745 1090h, USBHS.PIPE2TRE 8745 1094h, USBHS.PIPE3TRE 8745 1098h, USBHS.PIPE4TRE 8745 109Ch, USBHS.PIPE5TRE 8745 10A0h,
USBHS_NS.PIPE1TRE 0745 1090h, USBHS_NS.PIPE2TRE 0745 1094h, USBHS_NS.PIPE3TRE 0745 1098h, USBHS_NS.PIPE4TRE 0745 109Ch, USBHS_NS.PIPE5TRE 0745 10A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	TRCLR	Transaction Counter Clear	This bit is set to 1 when the transaction counter is reset. Writing 0 has no effect. This bit is read as 0.	R/W
b9	TRENB	Transaction Counter Enable	0: The transaction count function is disabled. 1: The transaction count function is enabled.	R/W
b15 to b10	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note: Only change the settings of the PIPEnTRE register while the PIPEnCTR.CSSTS flag is 0 and PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

TRCLR Bit (Transaction Counter Clear)

The USBHS clears the current count value of the transaction counter corresponding to pipe n and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

This bit enables or disables the transaction counter.

When the total number of packets is set in the PIPEnTRN.TRNCNT[15:0] bits and then the TRENB bit is set to 1 for the pipe in the receiving direction, the USBHS performs the following control upon completion of receiving packets as many as the value of the TRNCNT[15:0] bits.

- In continuous transfer mode (CNTMD bit is 1), the USBHS toggles the TRENB bit to the CPU side even if the FIFO buffer is not full at the end of reception.
- When the PIPECFG.SHTNAK bit is 1, the USBHS modifies the PIPEnCTR.PID[1:0] bits of the corresponding pipe to 00b (NAK response) upon completion of receiving packets as many as the value of the TRNCNT[15:0] bits.
- When the PIPECFG.BFRE bit is 1, the USBHS asserts the BRDY interrupt upon completion of reading the final data after receiving packets as many as the value of the TRNCNT[15:0] bits.

For pipes in the transmitting direction, set the TRENB bit to 0.

When the transaction count function is not used, set the TRENB bit to 0.

When the transaction count function is used, set the TRNCNT[15:0] bits and then set the TRENB bit to 1. Furthermore, set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

41.2.40 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USBHS.PIPE1TRN 8745 1092h, USBHS.PIPE2TRN 8745 1096h, USBHS.PIPE3TRN 8745 109Ah, USBHS.PIPE4TRN 8745 109Eh, USBHS.PIPE5TRN 8745 10A2h,
USBHS_NS.PIPE1TRN 0745 1092h, USBHS_NS.PIPE2TRN 0745 1096h, USBHS_NS.PIPE3TRN 0745 109Ah, USBHS_NS.PIPE4TRN 0745 109Eh, USBHS_NS.PIPE5TRN 0745 10A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter*1	• When writing to the register These bits specify the number of total packets (number of transactions) to be received by the relevant pipe. • When reading the register When the PIPEnTRE.TRENB bit is 0, these bits indicate the specified number of transactions. When the PIPEnTRE.TRENB bit is 1, these bits indicate the number of transaction in the current count (current count value).	R/W

Note 1. Only change the settings of the TRNCNT[15:0] bits while the PIPEnCTR.CSSTS flag and PIPEnTRE.TRENB bit are 0 and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag and PIPEnCTR.CSSTS flag are 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBHS changes the PID[1:0] bits to 00b (NAK response).

The value set to the PIPEnTRN register is retained even if a USB bus reset occurs.

TRNCNT[15:0] Bits (Transaction Counter)

The USBHS increments the value of the TRNCNT[15:0] bits (current counter value) when all of the following conditions for reception state are satisfied.

- The PIPEnTRE.TRENB bit is 1.
- When receiving a packet, the value of the TRNCNT[15:0] bits is different from the current counter value + 1.
- The payload of the received packet matches the value set to the PIPEMAXP.MXPS[10:0] bits.

The USBHS sets the value of the TRNCNT[15:0] bits (current counter value) to 0 when any of the following conditions is satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit is 1.
- When receiving a packet, the value of the TRNCNT[15:0] bits is the same as the current counted value + 1.
- The payload of the received packet matches the value set to the PIPEMAXP.MXPS[10:0] bits.

(2) Both of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit is 1.
- The USBHS has received a short packet.

(3) The following condition is satisfied.
- The PIPEnTRE.TRCLR bit is set to 1.

For pipes in the transmitting direction, set the TRNCNT[15:0] bits to 0. When the transaction count function is not used, set the TRNCNT[15:0] bits to 0. The number of transactions can be set to the TRNCNT[15:0] bits only when the PIPEnTRE.TRENB bit is 0. If the specified number of transactions is modified, set the PIPEnTRE.TRCLR bit to 1 (the current counter is cleared) before setting the PIPEnTRE.TRENB bit to 1 (count starts).

41.2.41 Device Address m Configuration Register (DEVADDm) (m = 0 to A)

Address(es): USBHS.DEVADD0 8745 10D0h, USBHS.DEVADD1 8745 10D2h, USBHS.DEVADD2 8745 10D4h, USBHS.DEVADD3 8745 10D6h, USBHS.DEVADD4 8745 10D8h, USBHS.DEVADD5 8745 10DAh, USBHS.DEVADD6 8745 10DCh, USBHS.DEVADD7 8745 10DEh, USBHS.DEVADD8 8745 10E0h, USBHS.DEVADD9 8745 10E2h, USBHS.DEVADDA 8745 10E4h, USBHS_NS.DEVADD0 0745 10D0h, USBHS_NS.DEVADD1 0745 10D2h, USBHS_NS.DEVADD2 0745 10D4h, USBHS_NS.DEVADD3 0745 10D6h, USBHS_NS.DEVADD4 0745 10D8h, USBHS_NS.DEVADD5 0745 10DAh, USBHS_NS.DEVADD6 0745 10DCh, USBHS_NS.DEVADD7 0745 10DEh, USBHS_NS.DEVADD8 0745 10E0h, USBHS_NS.DEVADD9 0745 10E2h, USBHS_NS.DEVADDA 0745 10E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	x	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDm register is not used. 0 1: Low-speed 1 0: Full-speed 1 1: High-speed	R/W
b10 to b8	HUBPORT[2:0]	HUB Port Connected for Communication	b10 b8 0 0 0: Connected directly to the USBHS port 001 to 111: HUB port number	R/W
b14 to b11	UPPHUB[3:0]	HUB Register Connected for Communication	b14 b11 0 0 0 0: Connected directly to the USBHS port 0001 to 1010: USB address of the HUB 1011 to 1111: Reserved	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W

The DEVADDm register specifies the communication speed of the communication target peripheral device for pipes 0 to 9, and specifies the port number and address of the connected upstream facing hub. In host controller operation, set each bit in the DEVADDm register before communications start to pipes.

Modify each bit in the DEVADDm register while there is no valid pipe specified by such a bit. The valid pipe is defined as the pipe satisfying both the conditions listed below.

- The DEVSEL[3:0] bits specify the DEVADDm register.
- The PIPEnCTR.PID[1:0] bits of the selected pipe are 01b (BUF response), or the default control pipe is the selected pipe and the DCPCTR.SUREQ bit is set to 1.

In function controller operation, set each bit in this register to 0.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

In host controller operation, the USBHS generates a packet according to the value of these bits.

HUBPORT[2:0] Bits (HUB Port Connected for Communication)

In host controller operation, the USBHS generates a packet according to the value of these bits when a split transaction is executed.

UPPHUB[3:0] Bits (HUB Register Connected for Communication)

In host controller operation, the USBHS generates a packet according to the value of these bits when a split transaction is executed.

41.2.42 Low Power Control Register (LPCTRL)

Address(es): USBHS.LPCTRL 8745 1100h, USBHS_NS.LPCTRL 0745 1100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	HWUPM	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7	HWUPM	Return Mode Setting	0: Hardware return is not made while the CPU clock is stopped. 1: Hardware return is made while the CPU clock is stopped.	R/W
b8	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

HWUPM Bit (Return Mode Setting)

This bit specifies whether to enable the hardware processing for return from a low power state of the LPM L1 state even while the CPU clock is stopped. In function controller operation, the processing for return from low power mode upon detecting resume signal is enabled even while the CPU clock is stopped. The HWUPM bit specifies whether to detect resume signal while the CPU clock is stopped. The PL1CTRL1.L1EXTMD bit controls whether to make hardware return. To make hardware return from a low power state of the LPM L1 state while the CPU clock is stopped, set the HWUPM bit and the PL1CTRL1.L1EXTMD bit to 1.

41.2.43 Low Power Status Register (LPSTS)

Address(es): USBHS.LPSTS 8745 1102h, USBHS_NS.LPSTS 0745 1102h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	SUSPENDM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	x	0	x	x	x	0	x	x	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b2	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b3	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b11 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b12	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	SUSPENDM	SuspendM Control	0: Suspend mode 1: Normal mode	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W

SUSPENDM Bit (SuspendM Control)

This bit controls the SuspendM signal output to the PHY. The initial value is 0 and the PHY is in suspend mode. To supply the PHY clock to operate the USB 2.0 host/function controller, set this bit to 1. Refer to section 41.3.3 for details on setting clocks.

When the SUSPENDM bit is 0, the registers in the USBHS are not writable, but are readable. However, registers listed in Table 41.12 below are write accessible even when the SUSPENDM bit is 0.

Table 41.12 Registers That Can be Written by Software when the SUSPENDM Bit is 0

Address	Register/Bit
8745 1000h or 0745 1000h	SYSCFG register
8745 1002h or 0745 1002h	BUSWAIT register
8745 1032h or 0745 1032h	INTENB1.PDDETINTE bit
8745 103Eh or 0745 103Eh	PHYSET register
8745 1100h or 0745 1100h	LPCTRL register
8745 1102h or 0745 1102h	LPSTS register
8745 1140h or 0745 1140h	BCCTRL register
8745 1164h or 0745 1164h	DPUSR1R register
8745 1168h or 0745 1168h	DPUSR2R register
8745 116Ah or 0745 116Ah	DPUSRCR register

However, the value written to the SYSCFG register while the PHY clock is stopped is updated after the PHY clock is oscillated. The PHY clock is oscillated in the following cases.

(1) The SUSPENDM bit is set to 1 and then the PLLSTA.PLLLOCK flag is set to 1 after the predetermined time has passed.
(2) USBCLK (48 MHz) and USB60CLK (60 MHz) are supplied to the USBHS from the main clock with the SUSPENDM bit set to 0
(Refer to section 41.2.17 for details on classic-only mode.)

While the PL1CTRL1.L1EXTMD bit is 0, the SUSPENDM bit must be controlled by software. When the PL1CTRL1.L1EXTMD bit is 1, when transitioning to the L1 or L2 state (suspended state), the SUSPENDM bit must be set to 0 by software, but when exiting the L1 or L2 state, the USBHS sets the SUSPENDM bit to 1.

41.2.44 Battery Charging Control Register (BCCTRL)

Address(es): USBHS.BCCTRL 8745 1140h, USBHS_NS.BCCTRL 0745 1140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDDETSTS	CHGDETSTS	—	—	—	VDMSRCE	IDPSINKE	VDPSRCE	IDMSINKE	IDPSRCE
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IDPSRCE	D+ Line IDPSRC Output Control	0: Stopped 1: 10 µA output	R/W
b1	IDMSINKE	D– Line 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b2	VDPSRCE	D+ Line VDPSRC (0.6 V) Output Control	0: Stopped 1: 0.6 V output	R/W
b3	IDPSINKE	D+ Line 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b4	VDMSRCE	D– Line VDMSRC (0.6 V) Output Control	0: Stopped 1: 0.6 V output	R/W
b7 to b5	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	CHGDETSTS	D– Line 0.6 V Input Detection Status Flag*1	0: Not detected 1: Detected	R
b9	PDDETSTS	D+ Line 0.6 V Input Detection Status Flag*2	0: Not detected 1: Detected	R
b15 to b10	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Valid when the IDMSINKE bit is 1.
Note 2. Valid when the IDPSINKE bit is 1.

IDPSRCE Bit (D+ Line IDPSRC Output Control)

In function controller operation, set this bit to 1 to perform Data Contact Detect. In the Battery Charging Specification Revision 1.2, there are two methods to handle Data Contact Detect; the method realized by software wait and the method to contact the data line by hardware. The IDPSRCE bit adopts the latter method. When the IDPSRCE bit is set to 1, the USBHS enables the IDP_SRC circuit and, at the same time, controls D– line pull-down.

IDMSINKE Bit (D– Line 0.6 V Input Detection (Comparator and Sink) Control)

In function controller operation, the following can be detected when the IDMSINKE bit is set to 1:

- During Primary Detection, VDMSRC (0.6 V) is input to the USBHS_DM pin from the host
- During Primary Detection, VDPSRC (0.6 V) output from the USBHS to the USBHS_DP pin is input to the USBHS_DM pin via the USB host

VDPSRCE Bit (D+ Line VDPSRC (0.6 V) Output Control)

In function controller operation, if Primary Detection is executed, this bit controls VDPSRC (0.6 V) output from the USBHS_DP pin.

IDPSINKE Bit (D+ Line 0.6 V Input Detection (Comparator and Sink) Control)

When the IDPSINKE bit is set to 1, the following states can be detected:

- In host controller operation
  During Primary Detection, VDPSRC (0.6 V) is input to the USBHS_DP pin from a peripheral device.
- In function controller operation
  During Secondary Detection, VDMSRC (0.6 V) output from the USBHS to the USBHS_DM pin is input to the USBHS_DP pin via the host.

VDMSRCE Bit (D– Line VDMSRC (0.6 V) Output Control)

- In host controller operation
  During Primary Detection, this bit controls VDMSRC (0.6 V) output from the USBHS_DM pin.
- In function controller operation
  During Secondary Detection, this bit controls VDMSRC (0.6 V) output from the USBHS_DM pin.

CHGDETSTS Flag (D– Line 0.6 V Input Detection Status Flag)

In function controller operation, this flag is enabled when the IDMSINKE bit is 1.
The CHGDETSTS flag becomes 1 under the following conditions:

- During Primary Detection, VDMSRC (0.6 V) is input to the USBHS_DM pin from the USB host.
- During Primary Detection, the VDPSRC (0.6 V) output from the USBHS to the USBHS_DP pin is input to the USBHS_DM pin via the USB host.

PDDETSTS Flag (D+ Line 0.6 V Input Detection Status Flag)

The PDDETSTS flag is enabled when the IDPSINKE bit is 1.
The PDDETSTS flag becomes 1 under the following conditions:

- In host controller operation
  During Primary Detection, VDPSRC (0.6 V) is input to the USBHS_DP pin from a peripheral device.
- In function controller operation
  During Secondary Detection, the VDMSRC (0.6 V) output from the USBHS to the USBHS_DM pin is input to the USBHS_DP pin via the host.

41.2.45 Function L1 Control Register 1 (PL1CTRL1)

Address(es): USBHS.PL1CTRL1 8745 1144h, USBHS_NS.PL1CTRL1 0745 1144h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	L1EXTMD	—	—	HIRDTHR[3:0]				DVSQ[3]	DVSQ[2:0]			L1NEGOMD	L1RESPMD[1:0]		L1RESPEN
Value after reset:	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	L1RESPEN	L1 Response Enable	0: LPM is not supported. 1: LPM is supported.	R/W
b2, b1	L1RESPMD[1:0]	L1 Response Mode	b2 b1 0 0: NYET response 0 1: ACK response 1 0: STALL response 1 1: Comply with the L1NEGOMD bit	R/W
b3	L1NEGOMD	L1 Response Negotiation Control*1	0: When the received HIRD field is larger than the HIRDTHR[3:0] bit value, an ACK response is returned. When the received HIRD field is smaller than or equal to the value set to the HIRDTHR[3:0] bits, an NYET response is returned. 1: When the received HIRD field is smaller than the HIRDTHR[3:0] bit value, an ACK response is returned. When the received HIRD field is greater than or equal to the value set to the HIRDTHR[3:0] bits, an NYET response is returned.	R/W
b6 to b4	DVSQ[2:0]	Extension Device State	These flags indicate the same value with the INTSTS0.DVSQ[2:0] bits.	R
b7	DVSQ[3]		This flag indicates the L1 state together with the INTSTS0.DVSQ[2:0] bits. 0000: Powered state 0001: Default state 0010: Address state 0011: Configured state 0100: Suspended state 0101: Suspended state 0110: Suspended state 0111: Suspended state 1000 to 1011: L1 state	R
b11 to b8	HIRDTHR[3:0]	L1 Response Negotiation Threshold Value*1	HIRD threshold value used when the L1RESPMD[1:0] bits are 11b. The format is the same as the HIRD field in the HL1CTRL2 register.	R/W
b13, b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	L1EXTMD	PHY Control Mode at L1 Return	0: The LPSTS.SUSPENDM bit is not set by hardware when exiting the L1 state. 1: The LPSTS.SUSPENDM bit is set by hardware when exiting the L1 state.	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W

Note 1. The L1NEGOMD bit is only valid when the L1RESPMD[1:0] bits are 11b.

L1RESPEN Bit (L1 Response Enable)

When the L1RESPEN bit is 0, even if the LPM token is received, the USBHS returns no response in regards to the LPM token. When the L1RESPEN bit is 1, if the LPM token is received, the USBHS returns a response in accordance with the L1RESPMD[1:0] bit setting.

L1RESPMD[1:0] Bits (L1 Response Mode)

When the L1RESPEN bit is set to 1, the USBHS returns a response to the LPM token in accordance with the L1RESPMD[1:0] bit setting.

L1NEGOMD Bit (L1 Response Negotiation Control)

The L1NEGOMD bit sets the response conditions for the HIRD[3:0] bit value. The L1NEGOMD bit is valid when the L1RESPMD[1:0] bits are 11b.

DVSQ[2:0] Bits (Extension Device State)

These flags mirror the INTSTS0.DVSQ[2:0] bits.

DVSQ[3] Bit (Extension Device State)

This flag is the fourth bit of the device state bits (DVSQ[2:0] bits).

HIRDTHR[3:0] Bits (L1 Response Negotiation Threshold Value)

The HIRDTHR[3:0] bits specify the HIRD threshold value used by the HIRD[3:0] bit value in the negotiation. The HIRDTHR[3:0] bits are valid when the L1RESPMD[1:0] bits are 11b.
The format of the set value is the same as the HL1CTRL2.HIRD[3:0] bits.

L1EXTMD Bit (PHY Control Mode at L1 Return)

In the L1 state, with the LPSTS.SUSPENDM bit set to 0 to stop the PHY clock, this bit specifies the LPSTS.SUSPENDM bit control method when receiving the resume signal.
When supporting the L1 state, set the L1EXTMD bit to 1 during the initialization process.
When entering the L1 state, the LPSTS.SUSPENDM bit must be set to 0 by software regardless of the L1EXTMD bit value.
When the L1EXTMD bit is set to 1, the USBHS also sets the LPSTS.SUSPENDM bit to 1 when exiting the L2 state (suspended state).

41.2.46 Function L1 Control Register 2 (PL1CTRL2)

Address(es): USBHS.PL1CTRL2 8745 1146h, USBHS_NS.PL1CTRL2 0745 1146h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RWEMON	HIRDMON[3:0]				—	—	—	—	—	—	—	—
Value after reset:	x	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11 to b8	HIRDMON[3:0]	HIRD Value Monitor Flag	These bits reflect the HIRD field value of the recently received LPM token.	R/W
b12	RWEMON	bRemoteWake Value Monitor Flag	This bit reflects the bRemoteWake bit value of the recently received LPM token.	R/W
b15 to b13	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

HIRDMON[3:0] Flags (HIRD Value Monitor Flag)

These flags monitor the HIRD field value of the received LPM token. These bits reflect the HIRD field value of the recently received LPM token.

RWEMON Flag (bRemoteWake Value Monitor Flag)

This flag monitors the bRemoteWake field value of the received LPM token. This flag reflects the bRemoteWake field value of the recently received LPM token.

41.2.47 Host L1 Control Register 1 (HL1CTRL1)

Address(es): USBHS.HL1CTRL1 8745 1148h, USBHS_NS.HL1CTRL1 0745 1148h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L1STATUS[1:0]		L1REQ
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	L1REQ	L1 Transition Request	Set this bit to 1 when making a request that a peripheral device transition to the L1 state. This bit is automatically set to 0 when the LPM transaction is completed.	R/W
b2, b1	L1STATUS[1:0]	L1 Request Completion Status	b2 b1 0 0: ACK response received 0 1: NYET response received 1 0: STALL response received 1 1: Transaction error occur ed	R/W
b15 to b3	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

L1REQ Bit (L1 Transition Request)

Set this bit to 1 to transition a peripheral device to the L1 state. When the L1REQ bit is set to 1, the USBHS starts the LPM transaction, and when the transaction is complete, the L1REQ bit is set to 0.

L1STATUS[1:0] Bits (L1 Request Completion Status)

These bits reflect the result of the LPM transaction executed by the L1REQ bit.

41.2.48 Host L1 Control Register 2 (HL1CTRL2)

Address(es): USBHS.HL1CTRL2 8745 114Ah, USBHS_NS.HL1CTRL2 0745 114Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BESL	—	—	L1RWE	HIRD[3:0]				—	—	—	—	L1ADDR[3:0]			
Value after reset:	0	x	x	0	0	0	0	0	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	L1ADDR[3:0]	LPM Token Device Address	These bits specify the value to be set in the ADDR field of the LPM token.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11 to b8	HIRD[3:0]	LPM Token HIRD	These bits specify the value to be set in the BESL field of the LPM token.	R/W
b12	L1RWE	LPM Token L1 Remote Wakeup Enable	These bits specify the value to be set in the bRemoteWake field of the LPM token.	R/W
b14, b13	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15	BESL	BESL & Alternate HIRD	This bit selects the resume signal output period when exiting the L1 state.	R/W

L1ADDR[3:0] Bits (LPM Token Device Address)

When the HL1CTRL1.L1REQ bit is set to 1, these bits specify the value to be set in the ADDR field of the LPM token transmitted by the USBHS.

HIRD[3:0] Bits (LPM Token HIRD)

When the HL1CTRL1.L1REQ bit is set to 1, these bits specify the value to be set in the BESL field of the LPM token transmitted by the USBHS. Table 41.13 indicates the Relationship Between the Values Set to the HIRD[3:0] and BESL Bits With the HIRD and BESL Values.

Table 41.13 Relationship Between the Values Set to the HIRD[3:0] and BESL Bits With the HIRD and BESL Values

	When the BESL Bit is 0	When the BESL Bit is 1	
Value Set to the HIRD[3:0] Bits	HIRD value	HIRD value	BESL value
0000b	50 µs (do not set this value)	75 µs	125 µs
0001b	125 µs	100 µs	150 µs
0010b	200 µs	150 µs	200 µs
0011b	275 µs	250 µs	300 µs
0100b	350 µs	350 µs	400 µs
0101b	425 µs	450 µs	500 µs
0110b	500 µs	950 µs	1000 µs
0111b	575 µs	1,950 µs	2,000 µs
1000b	650 µs	2,950 µs	3,000 µs
1001b	725 µs	3,950 µs	4,000 µs
1010b	800 µs	4,950 µs	5,000 µs
1011b	875 µs	5,950 µs	6,000 µs
1100b	950 µs	6,950 µs	7,000 µs
1101b	1,025 µs (do not set this value)	7,950 µs	8,000 µs
1110b	1,100 µs (do not set this value)	8,950 µs	9,000 µs
1111b	1,175 µs (do not set this value)	9,950 µs	10,000 µs

Note: In host controller operation, the HIRD[3:0] bit setting value is used for the resume signal output period when exiting the L1 state.

L1RWE Bit (LPM Token L1 Remote Wakeup Enable)

When the HL1CTRL1.L1REQ bit is set to 1, this bit specifies the value to be set in the bRemoteWake field of the LPM token transmitted by the USBHS. Depending on the L1RWE bit setting, the USBHS does not control detection of the remote wakeup signal in the L1 state. The remote wakeup signal is controlled by the DVSTCTR0.RWUPE bit as is the case when the USBHS is in the L2 state (suspended state).

BESL Bit (BESL & Alternate HIRD)

In host controller operation, this bit selects the resume signal output period when exiting the L1 state. Refer to the description of the HIRD[3:0] bits for details.

41.2.49 Deep Standby USB Transceiver Pin Monitor Register (DPUSR0R)

Address(es): USBHS.DPUSR0R 8745 1160h, USBHS_NS.DPUSR0R 0745 1160h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTSHM	—	DOVCBHM	DOVCAHM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b20	DOVCAHM	OVRCURA Input Flag	This flag indicates the level of the input signal to the USBHS_OVRCURA pin.	R
b21	DOVCBHM	OVRCURB Input Flag	This flag indicates the level of the input signal to the USBHS_OVRCURB pin.	R
b22	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b23	DVBSTSHM	VBUS Input Flag	This flag indicates the level of the input signal to the USBHS_VBUS pin.	R
b31 to b24	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

41.2.50 Deep Standby USB Resume Interrupt Register (DPUSR1R)

Address(es): USBHS.DPUSR1R 8745 1164h, USBHS_NS.DPUSR1R 0745 1164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTSH	—	DOVCBH	DOVCAH	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSTSHE	—	DOVCBHE	DOVCAHE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b4	DOVCAHE	OVRCURA Interrupt Enable/Clear	0: Interrupt is disabled and the DOVCAH flag is cleared. 1: Interrupt is enabled.	R/W
b5	DOVCBHE	OVRCURB Interrupt Enable/Clear	0: Interrupt is disabled and the DOVCBH flag is cleared. 1: Interrupt is enabled.	R/W
b6	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b7	DVBSTSHE	VBUS Interrupt Enable/Clear	0: Interrupt is disabled and the DVBSTSH flag is cleared. 1: Interrupt is enabled.	R/W
b19 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b20	DOVCAH	OVRCURA Interrupt Source Resume Status Flag	0: OVRCURA interrupt is not detected. 1: OVRCURA interrupt is detected.	R
b21	DOVCBH	OVRCURB Interrupt Source Resume Status Flag	0: OVRCURB interrupt is not detected. 1: OVRCURB interrupt is detected.	R
b22	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b23	DVBSTSH	VBUS Interrupt Source Resume Status Flag	0: VBUS interrupt is not detected. 1: VBUS interrupt is detected.	R
b31 to b24	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note: The value of this register is retained even in deep software standby mode 1. It is also not initialized by deep software standby reset 1.

This register is used to select the source to return from deep software standby mode 1.

DOVCAHE Bit (OVRCURA Interrupt Enable/Clear)

This bit is used to specify whether a change on the USBHS_OVRCURA pin is used as a source to return from deep software standby mode 1.

When this bit is set to 0 while the DOVCAH flag is 1, the DOVCAH is set to 0.

DOVCBHE Bit (OVRCURB Interrupt Enable/Clear)

This bit is used to specify whether a change on the USBHS_OVRCURB pin is used as a source to return from deep software standby mode 1.

When this bit is set to 0 while the DOVCBH flag is 1, the DOVCBH is set to 0.

DVBSTSHE Bit (VBUS Interrupt Enable/Clear)

This bit is used to specify whether a change on the USBHS_VBUS pin is used as a source to return from deep software standby mode 1.

When this bit is set to 0 while the DVBSTSH flag is 1, the DVBSTSH is set to 0.

DOVCAH Flag (OVRCURA Interrupt Source Resume Status Flag)

This flag indicates that the MCU has returned from deep software standby mode 1 due to a change on the USBHS_OVRCURA pin.

This flag is valid only when the DOVCAHE bit is set to 1. Writing 0 to the DOVCAHE bit sets this flag to 0.

DOVCBH Flag (OVRCURB Interrupt Source Resume Status Flag)

This flag indicates that the MCU has returned from deep software standby mode 1 due to a change on the USBHS_OVRCURB pin.

This flag is valid only when the DOVCBHE bit is set to 1. Writing 0 to the DOVCBHE bit sets this flag to 0.

DVBSTSH Flag (VBUS Interrupt Source Resume Status Flag)

This flag indicates that the MCU has returned from deep software standby mode 1 due to a change on the USBHS_VBUS pin.

This flag is valid only when the DVBSTSHE bit is set to 1. Writing 0 to the DVBSTSHE bit sets this flag to 0.

41.2.51 Deep Standby USB Resume Interrupt Register 2 (DPUSR2R)

Address(es): USBHS.DPUSR2R 8745 1168h, USBHS_NS.DPUSR2R 0745 1168h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DPINTE	DMINTE	—	—	DMVAL	DPVAL	—	—	DPINT	DMINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMINT	D– Interrupt Source Resume Status Flag	0: D– interrupt is not detected. 1: D– interrupt is detected.	R
b1	DPINT	D+ Interrupt Source Resume Status Flag	0: D+ interrupt is not detected. 1: D+ interrupt is detected.	R
b3, b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b4	DPVAL	D+ Input Flag	This flag indicates the level of the input signal to the USBHS_DP pin.	R
b5	DMVAL	D– Input Flag	This flag indicates the level of the input signal to the USBHS_DM pin.	R
b7, b6	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b8	DMINTE	D– Interrupt Enable/Clear	0: Interrupt is disabled and the DMINT flag is cleared. 1: Interrupt is enabled.	R/W
b9	DPINTE	D+ Interrupt Enable/Clear	0: Interrupt is disabled and the DPINT flag is cleared. 1: Interrupt is enabled.	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note: The value of this register is retained even in deep software standby mode 1. It is also not initialized by deep software standby reset 1.

This register is used to select the source to return from deep software standby mode 1.

DMINT Flag (D– Interrupt Source Resume Status Flag)

This flag indicates that the MCU has returned from deep software standby mode 1 in response to a change on the USBHS_DM pin. This flag is only valid when the setting of the DMINTE bit is 1. Writing 0 to the DMINTE bit sets this flag to 0.

DPINT Flag (D+ Interrupt Source Resume Status Flag)

This flag indicates that the MCU has returned from deep software standby mode 1 in response to a change on the USBHS_DP pin. This flag is only valid when the setting of the DPINTE bit is 1. Writing 0 to the DPINTE bit sets this flag to 0.

DMINTE Bit (D– Interrupt Enable/Clear)

This bit determines whether or not a signal transition on the USBHS_DM pin is used as a trigger for return from deep software standby mode 1. Setting this bit to 0 while the DMINT flag is 1 sets the DMINT flag to 0.

DPINTE Bit (D+ Interrupt Enable/Clear)

This bit determines whether or not a signal transition on the USBHS_DP pin is used as a trigger for return from deep software standby mode 1. Setting this bit to 0 while the DPINT flag is 1 sets the DPINT flag to 0.

41.2.52 Deep Standby USB Transceiver Control Register (DPUSRCR)

Address(es): USBHS.DPUSRCR 8745 116Ah, USBHS_NS.DPUSRCR 0745 116Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIXPHYPD	FIXPHY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FIXPHY	USB-PHY Output Fix	0: Normal operation 1: USBHS_DP and USBHS_DM pin inputs are enabled during deep software standby mode 1.	R/W
b1	FIXPHYPD	USB-PHY Power-Down Signal Mask	0: Normal operation 1: USB-PHY power-down signal is masked.	R/W
b15 to b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note: The value of this register is retained even in deep software standby mode 1. It is also not initialized by deep software standby reset 1.

FIXPHY Bit (USB-PHY Output Fix)

This bit is used to prevent the output of undefined values from the USBHS_DP and USBHS_DM pins during deep software standby mode 1. Set this bit to 1 when the resume interrupt or the BCHG interrupt is to be used as a trigger for return from deep software standby mode 1. Restore the setting of this bit to 0 after the MCU has returned from deep software standby mode 1.

FIXPHYPD Bit (USB-PHY Power-Down Signal Mask)

This bit is used to mask the USB-PHY power-down signal (PHYSET.DIRPD bit) to keep the USB-PHY block powered during deep software standby mode 1. Set this bit to 1 when the resume interrupt or the BCHG interrupt is to be used as a trigger for return from deep software standby mode 1. Restore the setting of this bit to 0 after the MCU has returned from deep software standby mode 1.

41.3 Operation

41.3.1 System Control

This section describes register settings necessary to initialize the USBHS and registers necessary for power consumption control.

41.3.1.1 Setting Registers Associated with the USBHS

After supplying the PHY clock, set the SYSCFG.USBE bit to 1 to enable USBHS operation.
Refer to section 41.3.3 for details on supplying the PHY clock.

41.3.1.2 Selecting the Controller Function

The USBHS can be used in host controller operation or function controller operation. Set the SYSCFG.DCFM bit to select a controller operation. However, modify the DCFM bit in the initial settings immediately after a reset, or when pulling up the D+ line is disabled, and pulling down the D+ and D– lines is disabled (both SYSCFG.DPRPU bit and SYSCFG.DRPD bit are 0).

41.3.2 Controlling the USB Data Bus Resistors

The USBHS incorporates pull-up and pull-down resistors for the D+ and D– lines. Set the SYSCFG.DPRPU bit and SYSCFG.DRPD bit to select pull up or pull down.
In function controller operation, confirm there is a connection to the USB host, then set the SYSCFG.DPRPU bit to 1 to pull up the D+ line. If the SYSCFG.DPRPU bit is set to 0 while communicating with the USB host, the pull-up resistor for the USB data line is disabled, and the USB host can be notified that the device has been disconnected.
In host controller operation, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D– lines.
Table 41.14 lists the settings for the USB data bus resistors. Set the DRPD and DPRPU bits to select the USB data bus resistors.

Table 41.14 Controlling the USB Data Bus Resistors (Excluding OTG Operation)

Settings		USB Data Bus Resistor Control		
DRPD bit	DPRPU bit	D– line	D+ line	Remarks
0	0	Open	Open	When the resistor is not used
0	1	Open	Pull-up	When operating as the function controller
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Do not set this value except for OTG operation

41.3.3 Supplying Clocks

The two input clocks required for the USBHS are shown in Table 41.15.

Table 41.15 Input Clocks

Input Clock Name	Description
Peripheral module clock (PCLKA)	PCLKA is used to access USBHS registers from the CPU, DMAC, and DTC. The clock frequency has no restrictions on the USBHS side.
PHY clock	Generating the PHY clock • When the PHYSET.HSEB bit is 0 Supply a clock to the EXTAL pin from a crystal oscillator with an oscillation frequency of 12, 20, 24, or 48 MHz and a frequency accuracy of ±100 ppm or less, and use a PLL in the USB-PHY to generate the PHY clock. • When the PHYSET.HSEB bit is 1 (CL only mode) The PHY clock is supplied when USBCLK (48 MHz) and USB60CLK (60 MHz) are supplied to the USBHS from the clock generation circuit and the HSEB bit is set to 1. In this case, high-speed operation is not supported.

Figure 41.2 shows the PHY Clock Settings in Classic-Only Mode.

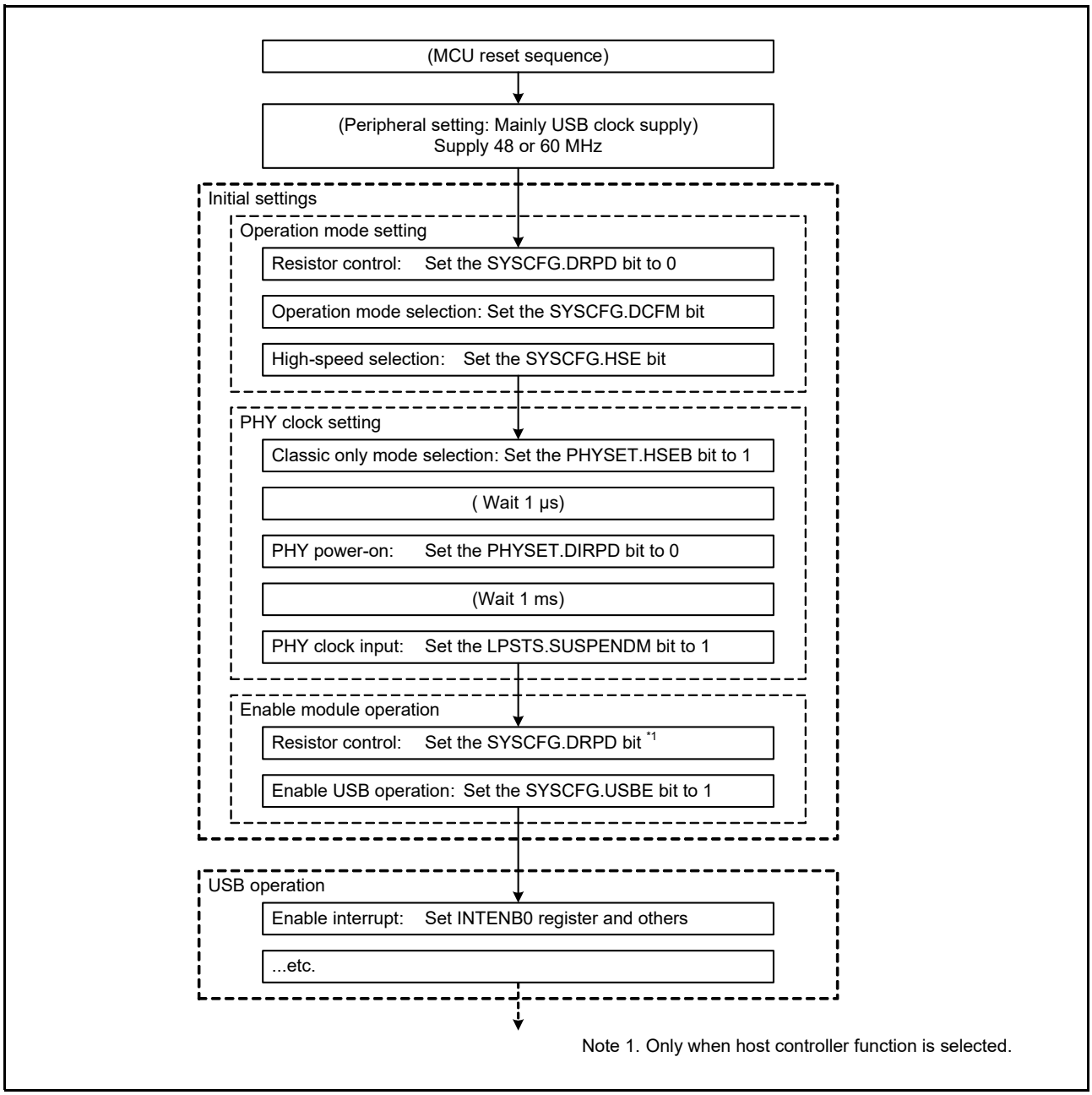


Figure 41.2 PHY Clock Settings in Classic-Only Mode

Figure 41.3 shows the PHY Clock Settings When Not Using Classic-Only Mode.

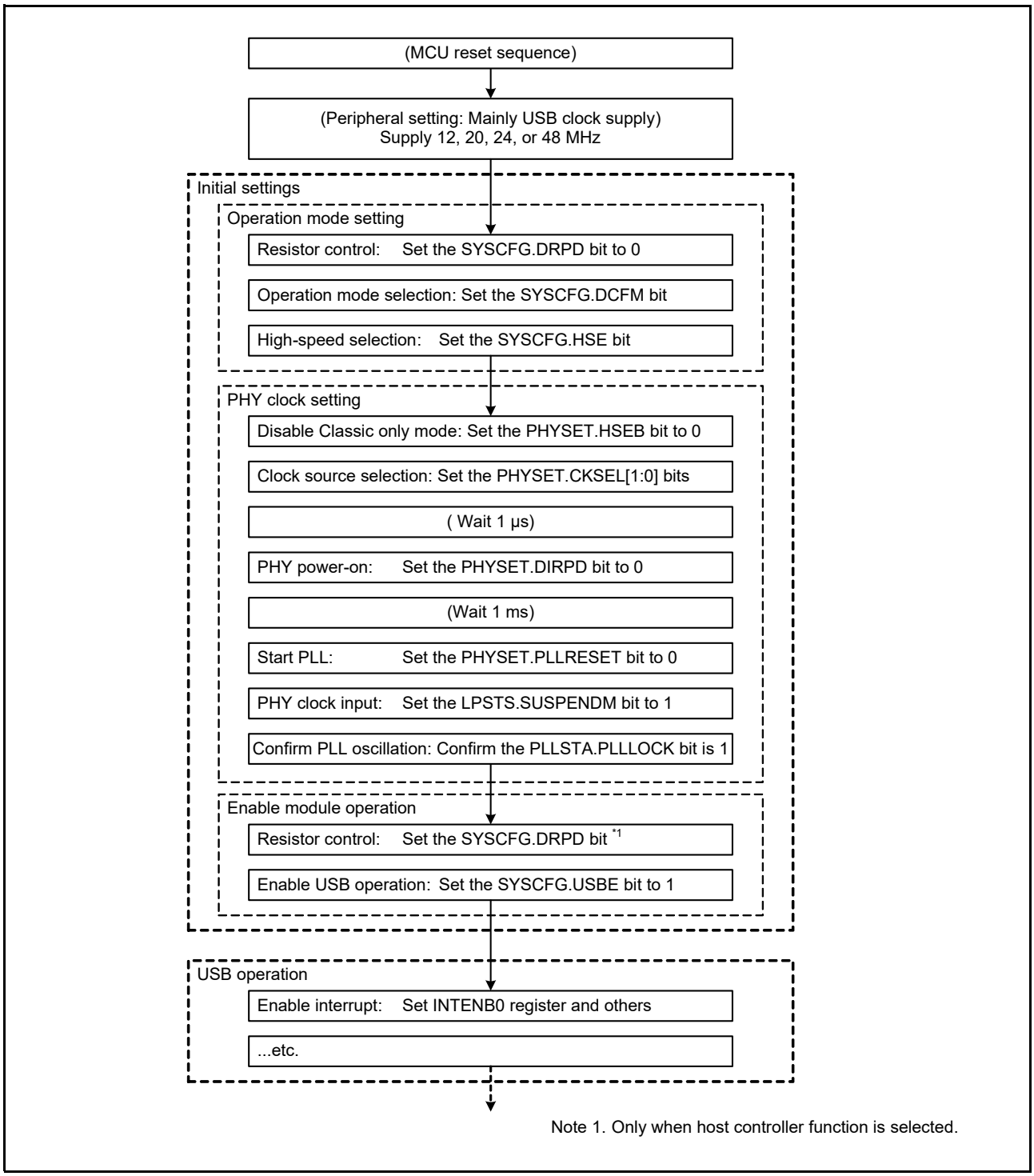


Figure 41.3 PHY Clock Settings When Not Using Classic-Only Mode

41.3.4 Notes on Stopping Clock

The CPU clock and PHY clock can be stopped when a device is disconnected or the USBHS is in the suspend state. However, in function controller operation, if either of these clocks are stopped while the USBHS is in the suspended state, they must be resupplied when exiting the suspended state. The PHY clock must be resupplied within 5.5 ms of generating the resume interrupt.

41.3.5 Interrupts

Table 41.16 lists the interrupt functions of the USBHS. When a condition to generate an interrupt is satisfied, and the interrupt output is enabled using the corresponding interrupt enable register, the USBHS issues a USBHSR interrupt request to the interrupt controller (ICU).

Table 41.16 List of Interrupt Functions (1/2)

Flag That Becomes 1	Interrupt	Interrupt Source	Function Generated	Status Flag
VBINT	VBUS interrupt	• USBHS_VBUS input pin state change was detected (from low to high and from high to low).	Host *1, function	INTSTS0. VBSTS
RESM	Resume interrupt	• USB bus state change was detected in the suspended state. (J state to K state or J state to SE0)	Function	—
SOFR	Frame number refresh interrupt	In host controller operation • An SOF packet with a different frame number was transmitted In function controller operation • An SOF packet with a different frame number was received. • Failed to receive an SOF packet when the microframe number is 0 due to packet corruption.	Host, function	—
DVST	Device state transition interrupt	• When device state transition was detected USB bus reset detection Suspension detection SetAddress() request received SetConfiguration request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When control transfer stage transition was detected Completed setup stage Transition to control write transfer status stage Transition to control read transfer status stage Completed control transfer A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt (BEMP interrupt)	• The buffer was empty after all FIFO buffer data was transmitted. • A packet exceeding the maximum packet size was received.	Host, function	BEMPSTS. PIPEBEMP[9:0]
NRDY	Buffer not ready interrupt (NRDY interrupt)	In host controller operation • The STALL response from the function was received in response to the token issued. • The response from the function in response to the issued token was not received successfully (no response three times in succession or packet receive error three times in succession). • An overrun or underrun error occurred during isochronous transfer. In function controller operation • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are set to 01b (BUF response) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer. • An interval error occurred during data reception in isochronous transfer.	Host, function	NRDYSTS. PIPENRDY[9:0]
BRDY	Buffer ready interrupt (BRDY interrupt)	• The buffer is ready (readable or writable state).	Host, function	BRDYSTS. PIPEBRDY[9:0]
OVRCR	Overcurrent change interrupt (OVRCR interrupt)	• USBHS_OVRCURA pin or USBHS_OVRCURB pin state change was detected (both low to high and high to low).	Host, function	SYSSTS0. OVCMON[1:0]
BCHG	Bus change interrupt (BCHG interrupt)	• USB bus state change was detected.	Host, function	SYSSTS0. LNST[1:0]
DTCH	Device disconnection detect interrupt (DTCH interrupt)	• Disconnection of peripheral device was detected.	Host	DVSTCTR0. RHST[2:0]

Table 41.16 List of Interrupt Functions (2/2)

Flag That Becomes 1	Interrupt	Interrupt Source	Function Generated	Status Flag
ATTCH	Device connection detect interrupt (ATTCH interrupt)	• The J state (continuous USB bus state for 2.5 µs) or K state (continuous USB bus state for 2.5 µs) was detected. This interrupt can be used for detecting connection of peripheral device.	Host	—
EOFERR	EOF error detect interrupt (EOFERR interrupt)	• EOF error of peripheral device was detected.	Host	—
SACK	SETUP transaction normal response interrupt (SACK interrupt)	• The SETUP transaction normal response (ACK) was received.	Host	—
SIGN	SETUP transaction error interrupt (SIGN interrupt)	• A SETUP transaction error (no response or corrupted ACK packet) was detected three times in succession.	Host	—
PDDETINT	Portable Device detection interrupt	• Portable Device connection was detected.	Host	BCCTRL. PDDETSTS
LPMEND	LPM transaction end interrupt (LPMEND interrupt)	• LPM transaction has been completed.	Host	HL1CTRL1. L1STATUS[1:0]
L1RSMEND	L1 resume end interrupt (L1RSMEND interrupt)	• Resume processing from L1 state has been completed.	Host	—

Note 1. Though this interrupt can be generated in host controller operation, it is not usually used in host controller operation.

41.3.5.1 Setting the USBHSR Interrupt Signal Output Method

The USBHSR interrupt signal output method can be selected by setting the SOFCFG.INTL bit. However, since the detection method for detecting USBHSR interrupts in this MCU is level sense, set the INTL bit to 1 (level output).

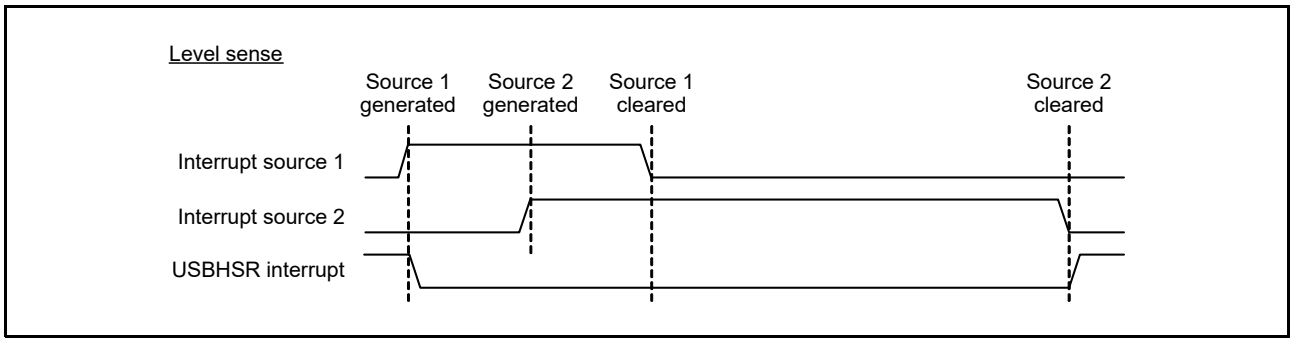


Figure 41.4 USBHSR Interrupt Operation

Figure 41.5 shows the Interrupt Association Diagram.

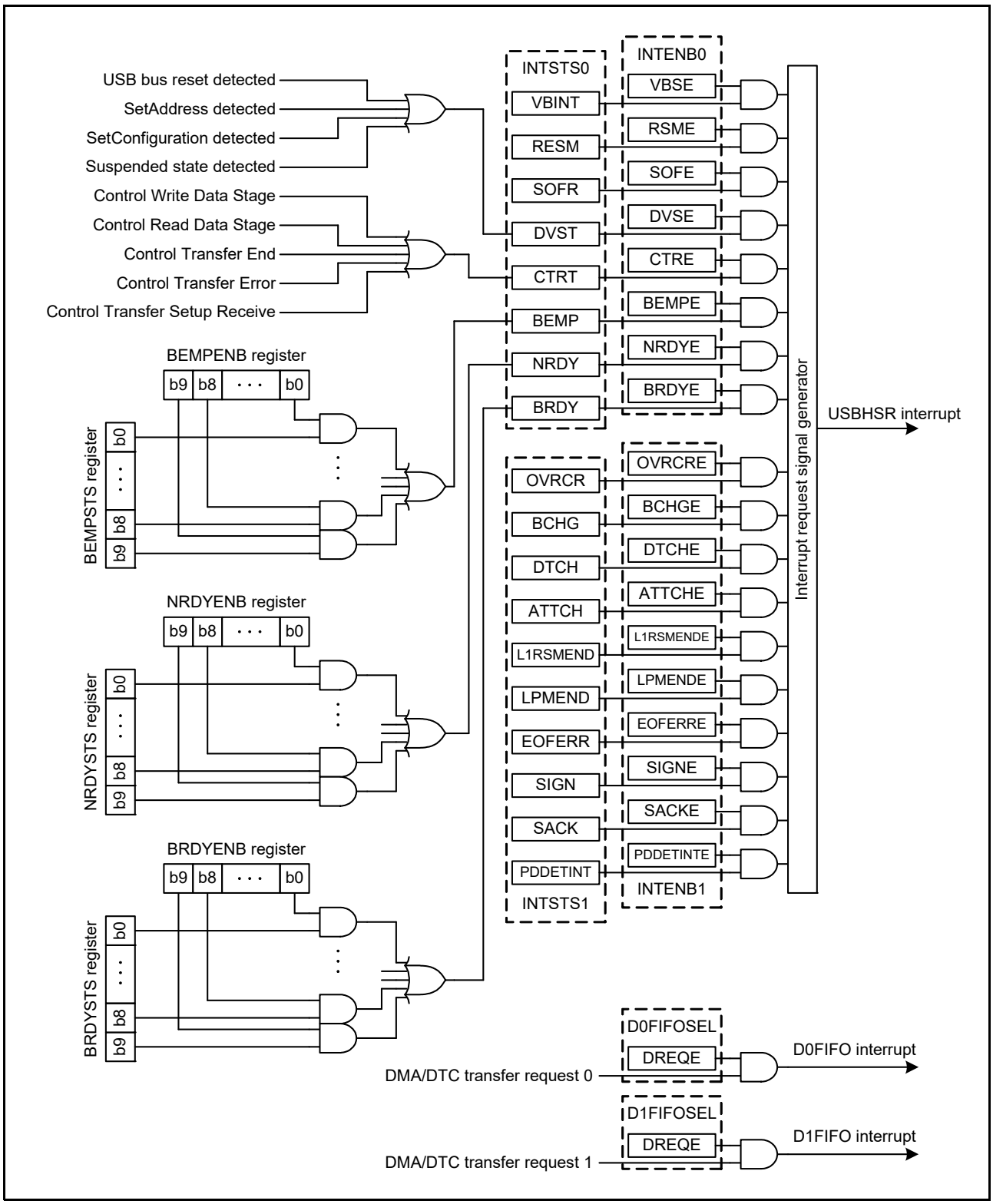


Figure 41.5 Interrupt Association Diagram

Table 41.17 lists the USBHS Interrupt Sources.

Table 41.17 USBHS Interrupt Sources

Interrupt	Interrupt Status Flag	DTC/DMAC Triggerable	Priority
D0FIFO	DMA/DTC transfer request 0	Yes	High
D1FIFO	DMA/DTC transfer request 1	Yes	↑
USBHSR	VBUS interrupt, resume interrupt, frame number refresh interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent change interrupt, bus change interrupt, device disconnection detect interrupt, device connection detect interrupt, EOF error detect interrupt, SETUP transaction normal response interrupt, SETUP transaction error interrupt, portable device detection interrupt, LPM transaction end interrupt, and L1 resume end interrupt	No	Low

41.3.6 Interrupt Descriptions

41.3.6.1 BRDY Interrupt

The BRDY interrupt is generated in both host controller operation and function controller operation. When each pipe satisfies the conditions listed below, the USBHS sets the corresponding bit in the BRDYSTS register to 1. Under these conditions, if a bit in the BRDYENB register that corresponds to a given pipe and the INTENB0.BRDYE bit are 1, the USBHS generates the BRDY interrupt request.
The conditions for generating and method for clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit, and the PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit is 0 and the PIPECFG.BFRE Bit is 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible. On any of the following conditions, the USBHS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEBRDY[9:0] flag corresponding to the pertinent pipe to 1.

(a) For pipes set to transmit:

- When the DIR bit is changed from 0 to 1.
- When writing by the CPU to the FIFO buffer for a given pipe is disabled (when the BSTS flag is read as 0) and the USBHS module has completed packet transmission.
  In continuous transfer mode, the BRDY interrupt is generated when data from one FIFO buffer is done being transmitted.
- If a FIFO buffer is specified as a double buffer, when data is done being written to one plane of the FIFO buffer, the other plane becomes empty.
  While data is being written to one plane of a FIFO buffer, even if the other plane is done transmitting, a request trigger is not generated until the data being written to the current plane is complete.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- Setting the PIPEnCTR.ACLRM bit to 1 changes the state of the FIFO buffer from write disabled to write enabled.

A request trigger is not generated for the default control pipe (during data transmission for control transfers).

(b) For pipes set to receive:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is 0).
  However, no request trigger is generated for the transaction in which data PID mismatch has occurred.
  In continuous transfer mode, when the maximum packet size of data is received and there is still space in the FIFO buffer, the request trigger is not generated. When a short packet is received, the request trigger is generated even if the FIFO buffer has available space. When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

In function controller operation, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEBRDY[9:0] flags. In this case, 1's should be written to the corresponding bits for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit is 0 and the PIPECFG.BFRE Bit is 1

With these settings, the USBHS determines that a BRDY interrupt is generated on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the flag in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USBHS determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the PIPEnTRN register is used and the number of packets specified by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USBHS determines that all data for a single transfer has been completely read. When a zero-length packet is received while the FIFO buffer is empty, the USBHS determines that all data for a single transfer has been completely read when the FRDY flag in a FIFO port control register is 1 and the DTLN[11:0] bits are 000h. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFO port control register. With these settings, the USBHS does not detect a BRDY interrupt for the pipe in the transmitting direction. The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEBRDY[9:0] flag. In this case, 1's should be written to the corresponding bits for the other pipes. In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit is 1 and the PIPECFG.BFRE Bit is 0

With these settings, the BRDYSTS.PIPEBRDY[9:0] flag values are linked to the BSTS flag setting for each pipe. The BRDY interrupt status bits are set to 1 or 0 by the USBHS depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO port is ready for write access, and are set to 0 when it is not ready. However, the BRDY interrupt is not generated even if the default control pipe in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access). When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until 1 is written to BCLR bit.

With this setting, the PIPEBRDY[9:0] flags cannot be set to 0 by software. When the SOFCFG.BRDYM bit is set to 1, the PIPECFG.BFRE bit (for all pipes) should be set to 0, and the SOFCFG.INTL bit should be set to 1 (level control).

Figure 41.6 shows the BRDY Interrupt Generation Timing.

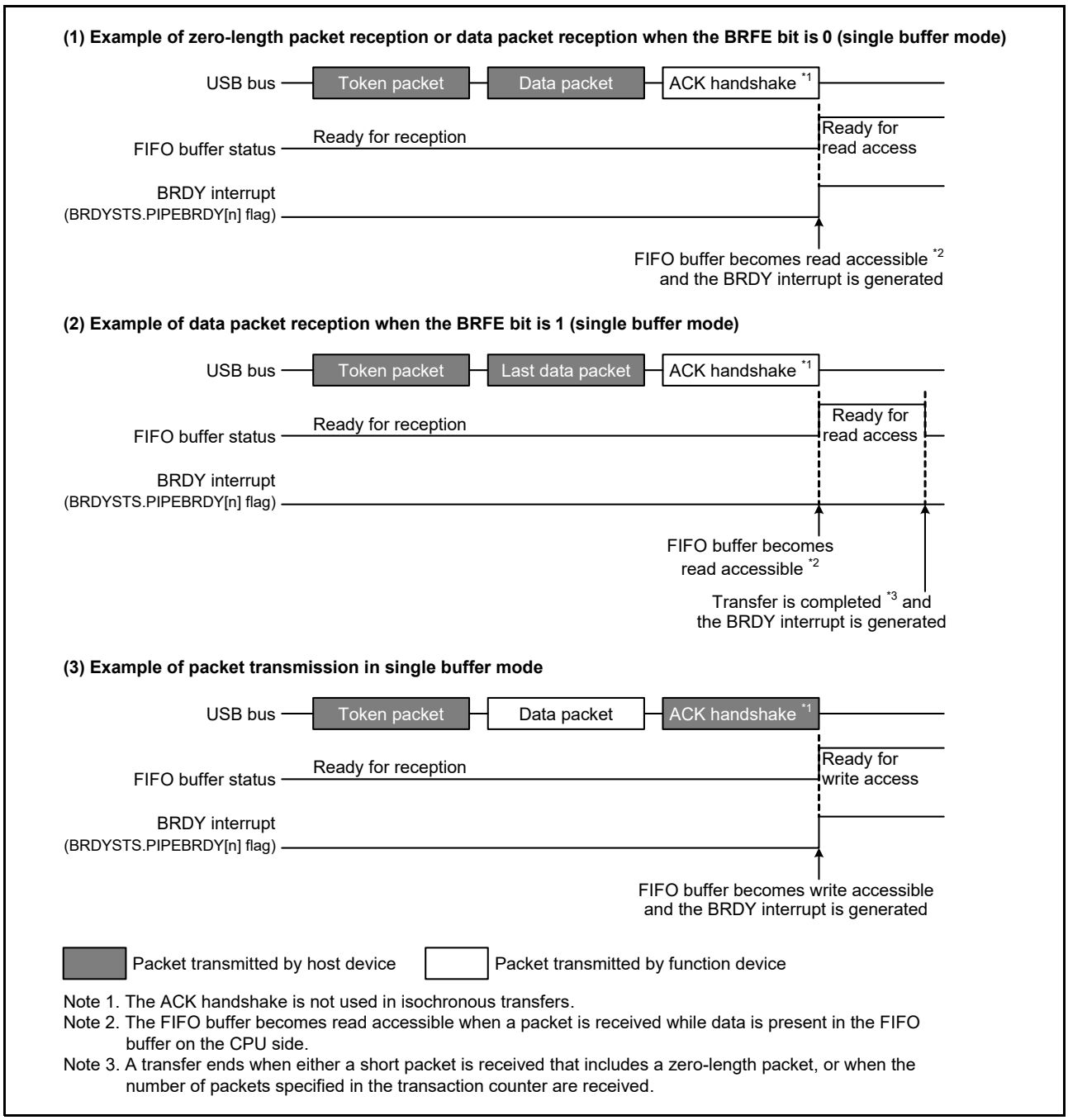


Figure 41.6 BRDY Interrupt Generation Timing

The condition that the USBHS clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting value. Table 41.18 shows the condition for clearing the BRDY flag.

Table 41.18 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing the BRDY Flag
0	The USBHS sets the BRDY flag to 0 when all flags in the BRDYSTS register have been set to 0 by software.
1	The USBHS sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

41.3.6.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF response), the USBHS sets the bit corresponding to the NRDYSTS.PIPENRDY[9:0] flag to 1. If the corresponding bit in the NRDYENB register has been set to 1, the USBHS sets the INTSTS0.NRDY flag to 1 and generates a USBHSR interrupt request.
The following describes the conditions in which the USBHS generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution in host controller operation. During setup transactions in host controller operation, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer in function controller operation.

(1) In Host Controller Operation, and Connecting When a SPLIT Transaction is Not Performed

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USBHS detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USBHS transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPENRDY[9:0] flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USBHS sets the bit corresponding to the PIPENRDY[9:0] flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK response).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device (this corresponds not only to the STALL handshake for the OUT token, but also the STALL handshake for the PING token).
  In this case, the USBHS sets the bit corresponding to the PIPENRDY[9:0] flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL response).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USBHS discards the received data for the IN token and sets the PIPENRDY[9:0] flag corresponding to the pipe and the OVRN flag to 1. When a packet error is detected in the received data for the IN token, the USBHS also sets the FRMNUM.CRCE flag to 1.
- For the pipe for transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USBHS (when timeout is detected before detection of the data packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USBHS sets the PIPENRDY[9:0] flag corresponding to each pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK response).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the data packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL.

(2) In Host Controller Operation, and Connecting When a SPLIT Transaction is Performed

On any of the following conditions, the USBHS detects an NRDY interrupt.

(a) For the pipe in the transmitting direction:

- For the isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer
  In this case, when the start-split transaction is issued, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and the FRMNUM.OVRN flag to 1. In addition, a zero-length packet is transmitted following the OUT token.
- If a start-split or complete-split transaction is used for a transfer other than the isochronous transfer, the NRDY interrupt is issued when any combination of the following conditions occur three times subsequently: 1) The hub returns no response (a timeout is detected without the handshake packet from the hub being detected); and 2) An error is detected in the packet from the hub.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and modifies the PID[1:0] bits of the corresponding pipe to 00b (NAK response). When the complete-split is issued, if the NRDY interrupt is detected, the USBHS sets the CSSTS flag to 0.
- When the STALL handshake is received in response to the complete-split token.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1, modifies the PID[1:0] bits of the corresponding pipe to 11b (STALL response), and sets the CSSTS flag to 0. However, the NRDY interrupt is not detected for the SETUP transaction.
- If a complete-split token is issued for an interrupt transfer, and the NYET handshake is received when the microframe is 4.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and the CSSTS flag to 0 (the PID[1:0] bits of the corresponding pipe are not modified).

(b) For the pipe in the receiving direction:

- For isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer
  In this case, when the start-split token is issued, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and the FRMNUM.OVRN flag to 1. In addition, the received data for the IN token is discarded.
- If a start-split token or complete-split token is issued for a bulk transfer or a transfer other than a SETUP transaction for the default control pipe, the NRDY interrupt is issued when any combination of the following conditions occur three times subsequently: 1) The hub returns no response (a timeout is detected without the data packet from the hub being detected) with respect to the IN token issued by the USBHS; and 2) An error is detected in the packet from the hub.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and modifies the PID[1:0] bits of the corresponding pipe to 00b (NAK response). If these conditions are fulfilled when the complete-split is issued, the USBHS sets the CSSTS flag to 0.
- If a complete-split token is issued for an isochronous transfer or interrupt transfer, the NRDY interrupt is issued when any combination of the following conditions occur three times subsequently: 1) The hub returns no response

(a timeout is detected without the data packet from the hub being detected) with respect to the IN token issued by the USBHS; and 2) An error is detected in the packet from the hub.
When the transfer type is an interrupt pipe, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1 and modifies the PID[1:0] bits of the corresponding pipe to 00b (NAK response). If these conditions are fulfilled when the complete-split is issued, the USBHS sets the CSSTS flag to 0.
When the transfer type is the isochronous type, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1, the CRCE flag to 1, and the CSSTS flag to 0 (the PID[1:0] bits of the corresponding pipe are not modified).

- If a complete-split token is issued for a transfer other than an isochronous transfer, and a STALL handshake is received.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1, modifies the PID[1:0] bits of the corresponding pipe to 11b (STALL response), and sets the CSSTS flag to 0.
- If a complete-split token is issued for an isochronous transfer or interrupt transfer, and the NYET handshake is received when the microframe is 4.
  In this case, the USBHS sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to the specified pipe to 1, the CRCE flag to 1, and the CSSTS flag to 0 (the PID[1:0] bits of the corresponding pipe are not modified).

(3) In Function Controller Operation

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USBHS generates an NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPENRDY[9:0] flag to 1. For the pipe for isochronous transfers in which an interrupt is generated, the USBHS transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for isochronous transfers in which an interrupt is generated, the USBHS generates an NRDY interrupt request at the reception of the OUT token and sets the NRDYSTS.PIPENRDY[9:0] flag to 1 and FRMNUM.OVRN flag to 1. For the pipe for transfers other than isochronous transfers in which an interrupt is generated, the USBHS generates an NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY[9:0] flag to 1. However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the data packet, the NRDY interrupt request is not generated.
- A PING token is received when the FIFO buffer is not empty.
  When a PING token is received, the USBHS generates the NRDY interrupt request, and sets the NRDYSTS.PIPENRDY[9:0] flags to 1.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USBHS generates an NRDY interrupt request when (μ)SOF is received, and sets the NRDYSTS.PIPENRDY[9:0] flags to 1.

Figure 41.7 shows the NRDY Interrupt Generation Timing (In Function Controller Operation).

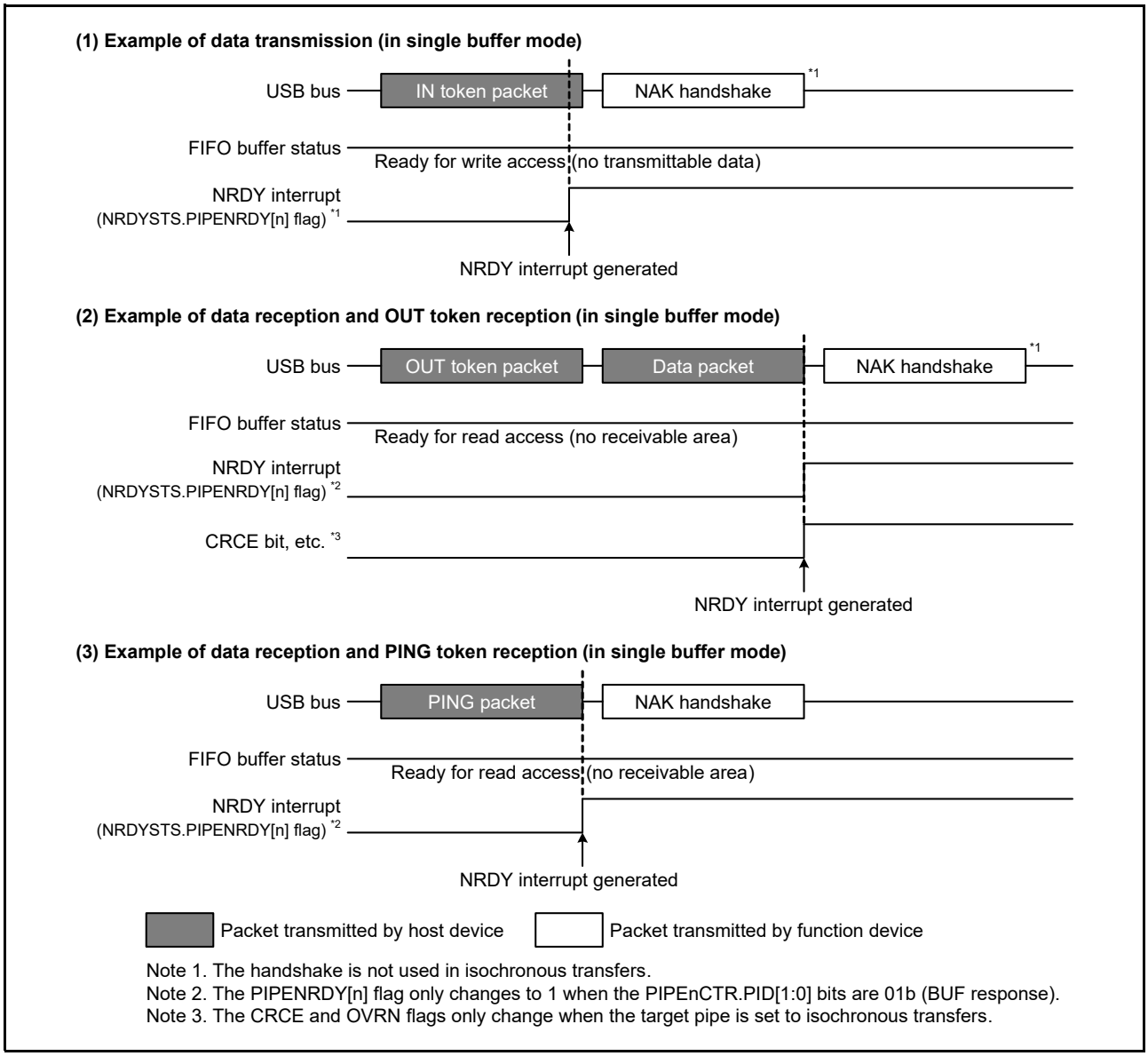


Figure 41.7 NRDY Interrupt Generation Timing (In Function Controller Operation)

41.3.6.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits in the pipe control register are set to 01b (BUF response), the USBHS sets the bit corresponding to the BEMPSTS.PIPEBEMP[9:0] flag to 1. If the corresponding bit in the BEMPENB register is set to 1, the USBHS sets the INTSTS0.BEMP flag to 1 and generates a USBHSR interrupt request.
The following describes the conditions in which the USBHS generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission), and during single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than default control pipe. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- In double buffer mode, data starts to be written to the FIFO buffer on the CPU side when data transfer from one plane is complete.
- When the buffer is cleared by setting the PIPEnCTR.ACLRM or the BCLR bit in a FIFO port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage during function controller operation.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBHS generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEBEMP[9:0] flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL response). Here, the USBHS returns no response in host controller operation, and returns a STALL response in function controller operation.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed

Figure 41.8 shows the BEMP Interrupt Generation Timing (In Function Controller Operation).

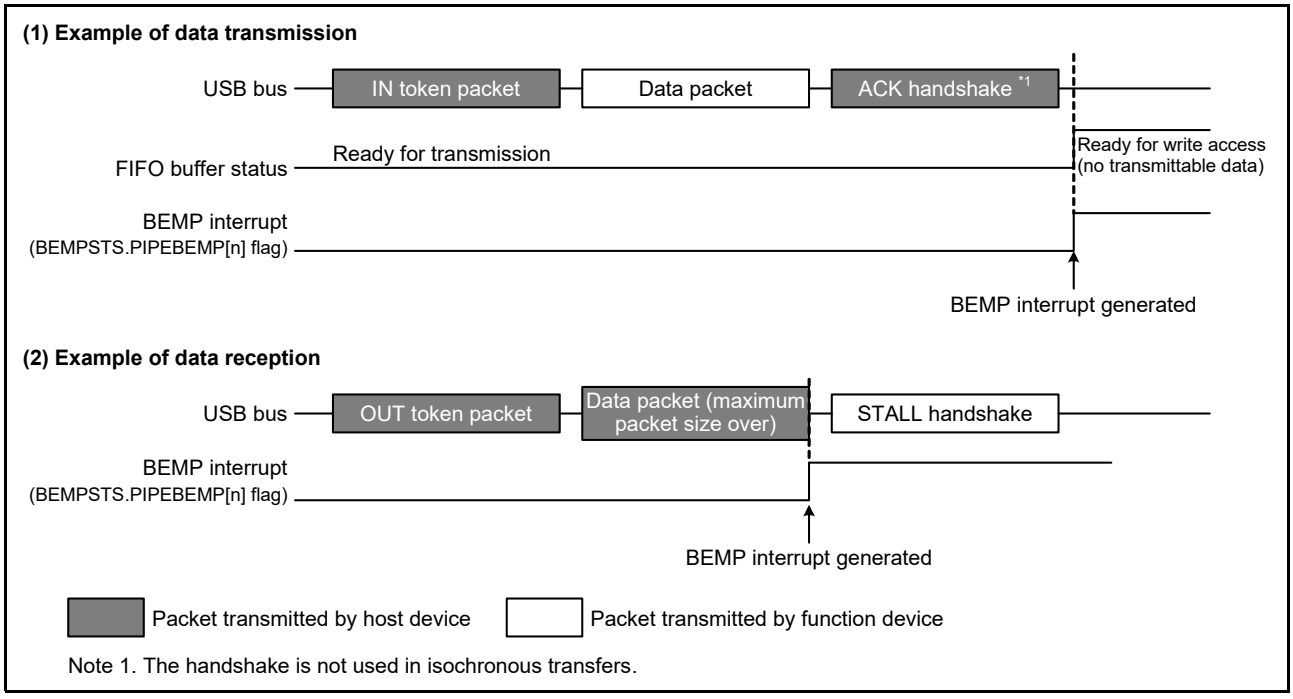


Figure 41.8 BEMP Interrupt Generation Timing (In Function Controller Operation)

41.3.6.4 Device State Transition Interrupt in Function Controller Operation

Figure 41.9 shows the Device State Transition Chart for the USBHS. The USBHS controls the device states and generates device state transition interrupts. However, the USBHS detects a return from the suspended state (by detecting the resume signal) by the resume interrupt. Device state transition interrupts can be enabled or disabled by the INTENB0 register. The transitioned device state can be checked by the INTSTS0.DVSQ[2:0] bits. To make a transition to the default state, a device state transition interrupt is generated after a USB bus reset is detected. The USBHS controls device states only in function controller operation. Device state transition interrupts are also generated only in function controller operation.

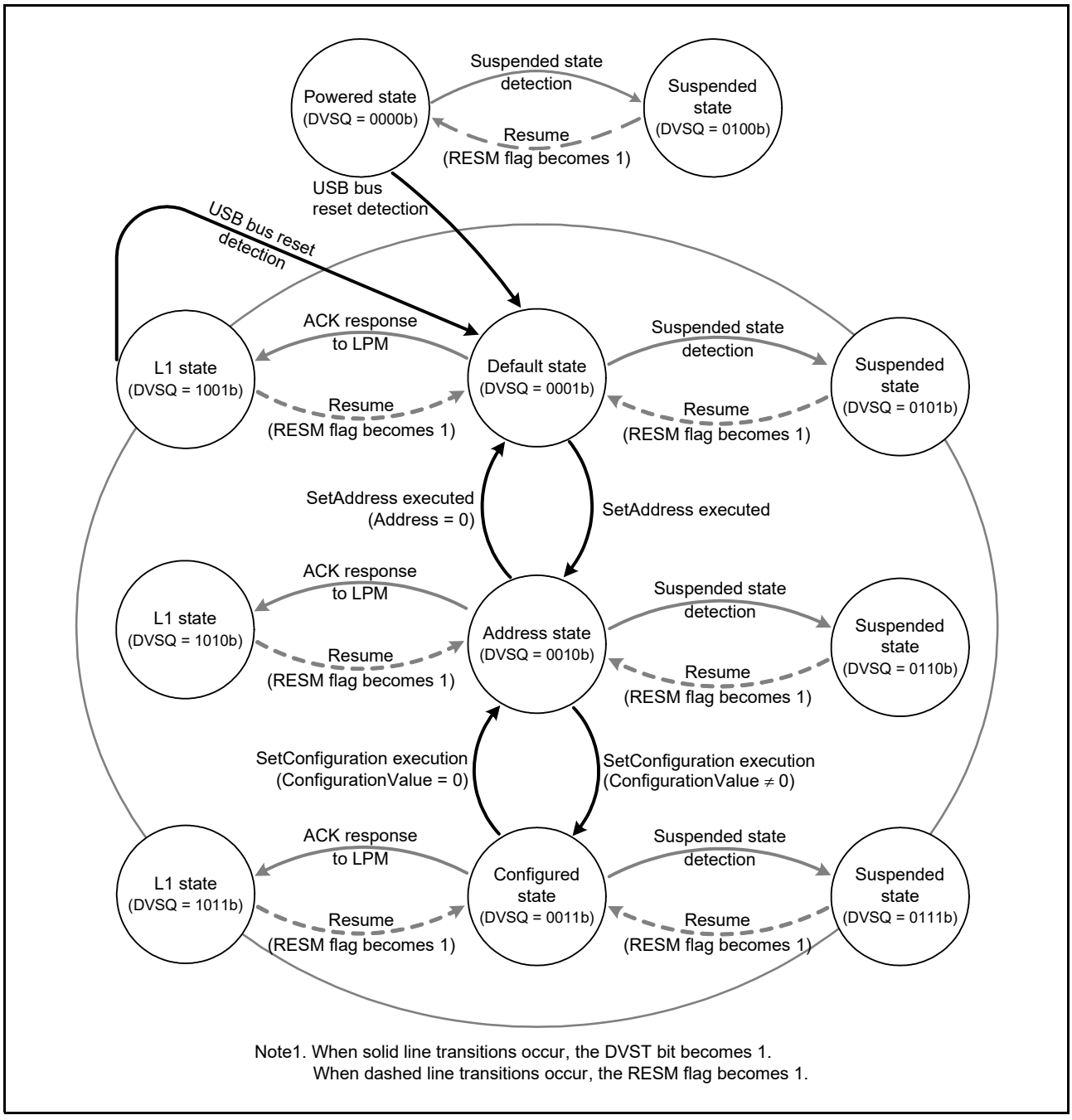


Figure 41.9 Device State Transition Chart

41.3.6.5 Control Transfer Stage Transition Interrupt in Function Controller Operation

Control transfer stage transitions of the USBHS are illustrated in Figure 41.10. The USBHS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled by the INTENB0 register. The transitioned transfer stage can be checked by the INTSTS0.CTSQ[2:0] bis. Control transfer stage transition interrupts are generated only in function controller operation. Control transfer sequence errors are shown below. When an error occurs, the DCPCTR.PID[1:0] bits are set to 10b or 11b (STALL response).

(1) Control read transfer
  (a) The OUT token is received with no data transferred in response to the IN or PING token in the data stage.
  (b) The IN token is received in the status stage.
  (c) A data packet of data PID = DATA0 is received in the status stage.
(2) Control write transfer
  (a) The IN token is received with no ACK response returned in response to the OUT token in the data stage.
  (b) The first data packet of data PID = DATA0 is received in the data stage.
  (c) The OUT or PING token is received in the status stage.
(3) Control write no data transfer
  (a) The OUT or PING token is received in the status stage.

If the number of pieces of receive data has exceeded the *wLength* value in the USB request in the control write transfer data stage, it is not recognized as a control transfer sequence error. When the USBHS receives a packet other than the zero-length packet in the control read transfer status stage, the USBHS returns an ACK response and completes the processing successfully.
When a control transfer stage transition interrupt is generated (INTSTS0.CTRT flag is 1) due to a sequence error, the value of CTSQ[2:0] bits are 110b is retained until the CTRT flag is set to 0 (interrupt status clear). Therefore, while CTSQ[2:0] bits are 110b is retained, the control transfer stage transition interrupt of setup stage completion is not generated even if a new USB request is received. (Completion of setup stage is retained by the USBHS, and a control transfer stage transition interrupt is generated after the interrupt status is cleared by software.)

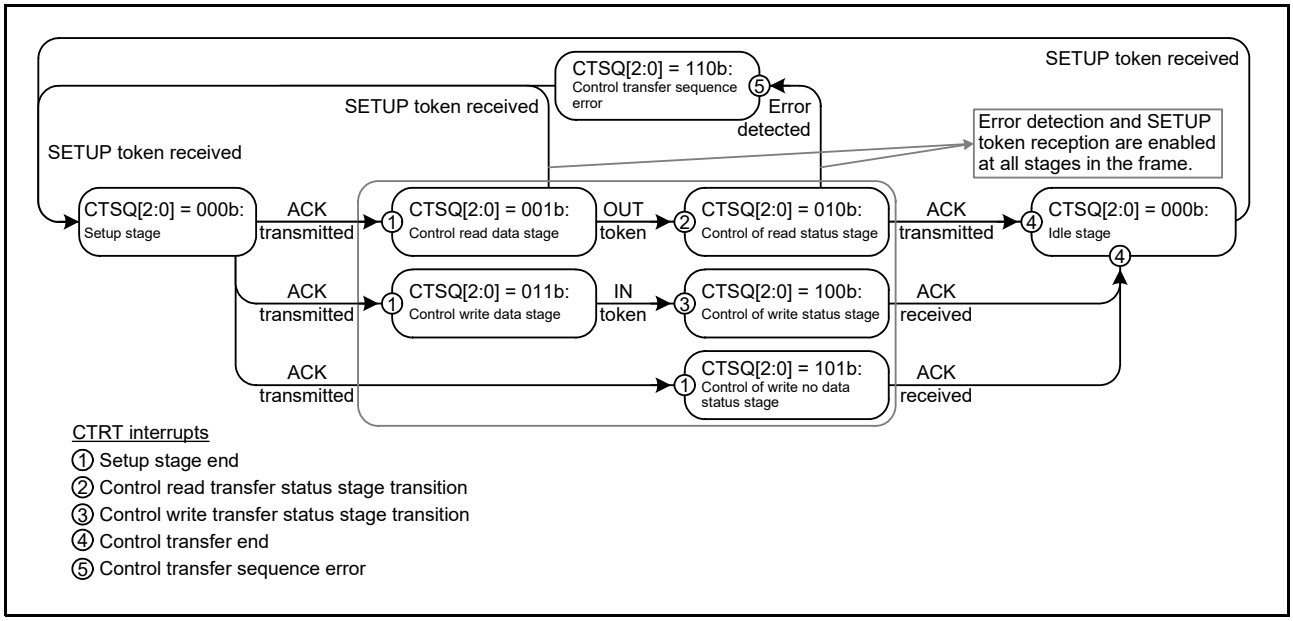


Figure 41.10 Control Transfer Stage Transition Chart

41.3.6.6 Frame Number Refresh Interrupt

In host controller operation, an interrupt is generated at the timing when the frame number is updated.
In function controller operation, a frame number refresh interrupt is generated when the frame number is updated. If the SOF cannot be received due to something like packet corruption, the frame number refresh interrupt is generated by the SOF recovery function. However, in this case the FRMNUM.FRNM[10:0] bits will not be updated.

41.3.6.7 VBUS Interrupt

When the USBHS_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBHS_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USBHS_VBUS pin level.

41.3.6.8 Resume Interrupt

In function controller operation, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J state to K state, or from J state to SE0). Recovery from the suspended state is detected by means of the resume interrupt. When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

41.3.6.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USBHS_OVRCURA or USBHS_OVRCURB pin level has changed. The levels of the USBHS_OVRCURA and USBHS_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the external power supply IC VBUS comparator can be checked using the OVRCR interrupt.

41.3.6.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup in host controller operation. The BCHG interrupt is generated regardless during host controller operation or function controller operation.

41.3.6.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected in host controller operation. The USBHS detects bus disconnection based on USB 2.0.
After detecting a DTCH interrupt, the USBHS controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies and indicates the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the Idle state.

41.3.6.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device in host controller operation. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

41.3.6.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times in host controller operation. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

41.3.6.14 ATTCH Interrupt

An ATTCH interrupt is generated when J state or K state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller operation. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K state, SE0, or SE1 changes to J state, and J state continues 2.5 µs.
- When J state, SE0, or SE1 changes to K state, and K state continues 2.5 µs.

41.3.6.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing defined in USB 2.0.
After detecting an EOFERR interrupt, the USBHS controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies and indicates the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the Idle state.

41.3.6.16 Portable Device Detection Interrupt

The USBHS sets the INTSTS1.PDDETINT flag to 1 on detecting a portable device and generates the portable device detection interrupt. When the portable device detection interrupt is generated, use software to repeat reading the BCCTRL.PDDETSTS flag until the same value is read three or more times, and perform debounce processing.

41.3.6.17 LPMEND Interrupt

When the LPM transaction is ended because a response from the peripheral device or a timeout is detected, the INTSTS1.LPMEND flag is set to 1 and the LPMEND interrupt is generated.

41.3.6.18 L1RSMEND Interrupt

When performing processing to exit the L1 state, when the exit processing is complete, the INTSTS1.L1RSMEND flag becomes 1, and the L1RSMEND interrupt is generated.

41.3.7 Pipe Control

Pipe setting items of the USBHS are listed in Table 41.19. USB data transfer is performed through logical pipes called endpoints. The USBHS is provided with 10 pipes for data transfer. Set each pipe according to the user system specifications.

Table 41.19 Pipe Setting Items

Register	Bit	Setting	Note
DCPCFG PIPECFG	TYPE[1:0]	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Selection of double buffer	Pipes 1 to 5: Settable
	CNTMD	Select continuous transfer mode or discontinuous transfer mode	Pipes 1, 2: Settable only when bulk transfer is selected Pipes 3 to 5: Settable
	DIR	Selection of transfer direction	IN or OUT settable
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Settable Set this number to a value other than 0000b when one or more pipes are used.
	SHTNAK	Selection of pipe disable at the end of transfer	Pipes 1, 2: Settable only when bulk transfer is selected Pipes 3 to 5: Settable
PIPEBUF	BUFSIZE[4:0]	FIFO buffer size	Default control pipe: Setting disabled (fixed to 64 bytes) Pipes 1 to 5: Settable (up to 2 Kbytes specifiable) Pipes 6 to 9: Setting disabled (fixed to 64 bytes)
	BUFNMB[7:0]	FIFO buffer number	default control pipe: Setting disabled (fixed to area 00h) Pipes 1 to 5: Setting disabled (area 08h to 87h specifiable) Pipes 6 to 9: Setting disabled (fixed to area 04h to 07h respectively)
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selection of device	Viewable only in host controller operation
	MXPS[6:0] MXPS[10:0]	Maximum packet size	Setting compliant with USB 2.0
PIPEPERI	IFIS	Buffer flush	Pipes 1, 2: Settable only when isochronous transfer is selected Pipes 3 to 5: Setting disabled Pipes 6 to 9: Setting disabled
	IITV[2:0]	Interval counter	Pipes 1, 2: Settable only when isochronous transfer is selected Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller operation
DCPCTR PIPEnCTR	BSTS	Buffer status	States of receive/transmit buffers of default control pipe are switched by the ISEL bit
	INBUFM	IN buffer monitor	Provided only for pipes 1 to 5
	SUREQ	SETUP request	Settable only for the default control pipe Controllable only in host controller operation
	SUREQCLR	SUREQ bit clear	Settable only for the default control pipe Controllable only in host controller operation
	CSCLR	CSSTS clear	Controllable only in host controller operation
	CSSTS	Check SPLIT status	Controllable only in host controller operation
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in function controller operation
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clearing the data toggle bit
	SQSET	Sequence set	Setting the data toggle bit
	SQMON	Sequence check	Checking the data toggle bit
	PBUSY	PIPE busy check	
	PID[1:0]	Response PID	
PIPEnTRE	TRENB	Transaction count enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT[15:0]	Transaction counter	Pipes 1 to 5: Settable

41.3.7.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] bits are 00b (NAK response)). Figure 41.11 shows pipe control register switching procedures when USB communication is enabled (PID[1:0] bits are 01b (BUF response)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] bits are 01b (BUF response)).

- Bits in registers DCPCFG and DCPMAXP
- DCPCTR.SQCLR and SQSET bits
- Bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers
- PIPEnCTR.ATREPM, ACLRM, SQCLR, and SQSET bits
- Bits in registers PIPEnTRE and PIPEnTRN
- Bits in the DEVADDm register (m = 0 to A)

For details on setting the CSCLR bit or setting the bits in the DEVADDm register, follow the procedures of each bit described in section 41.2, Register Descriptions.

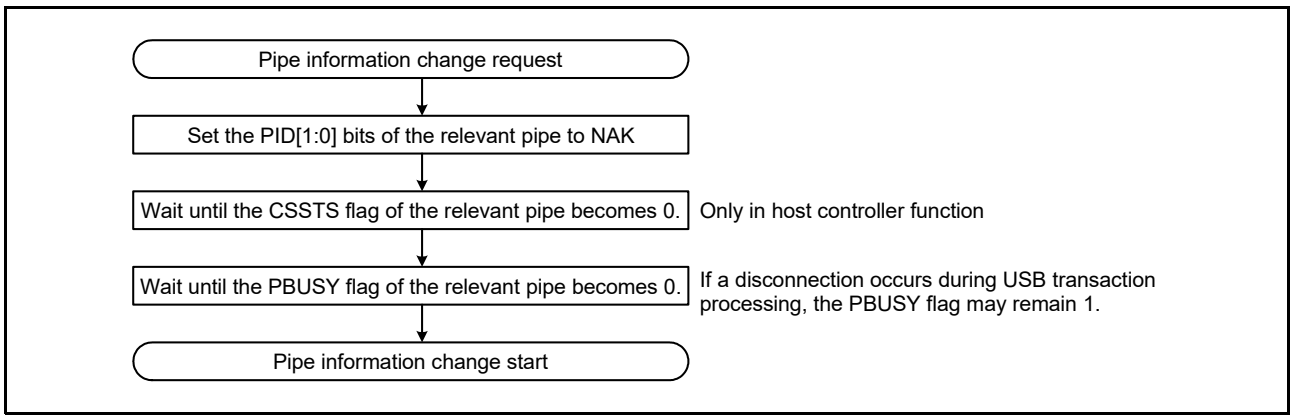


Figure 41.11 Procedure for Changing Pipe Information from the USB Communication Enabled State (PID[1:0] Bits are 01b (BUF Response))

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set while the CURPIPE[3:0] bits are set:

- Bits in registers DCPCFG and DCPMAXP
- Bits in registers PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- PIPEnCTR.ACLRM bit

In order to modify pipe information, CURPIPE[3:0] bits should be set to a pipe other than the pipe to be modified. For the default control pipe, the buffer should be cleared using the BCLR bit after the pipe information is modified.

41.3.7.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- Default control pipe: No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

41.3.7.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The default control pipe is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- Default control pipe: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits do not overlap.

41.3.7.4 Setting the Maximum Packet Size

Set the maximum packet size of each pipe with the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[10:0] bits. For the default control pipe and pipes 1 to 5, all maximum packet sizes defined in USB 2.0 can be set. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size before starting a transfer (PID[1:0] bits are set to 01b (BUF response)).

- Default control pipe: Set 64 for high-speed operation.
  Select 8, 16, 32, or 64 for full-speed operation.
- Pipes 1 to 5: Set 512 for high-speed bulk transfer.
  Select 8, 16, 32, or 64 for full-speed bulk transfer.
- Pipes 1, 2: Set a value from 1 to 1024 for high-speed isochronous transfer.
  Set a value from 1 to 1023 for full-speed isochronous transfer.
- Pipes 6 to 9: Set a value of 1 to 64.

High bandwidth interrupt transfers and isochronous transfers are not supported.

41.3.7.5 Transaction Counter (For Pipes 1 to 5 in the Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USBHS recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK response) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from the PIPEnTRN register differs depending on the setting of the PIPEnTRE.TRENB bit.

- TRENB bit is 0: The specified transaction counter value can be read.
- TRENB bit is 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and the PIPEnCTR.PID[1:0] bits are set to 01b (BUF response), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

41.3.7.6 Response PID

Set the response PID of each pipe by the PIPEnCTR.PID[1:0] bits DCPCTR.PID[1:0]. The following shows the USB operation with various response PID settings:

(1) Setting the response PID in host controller operation

Specify the execution of transaction for the response PID.

(a) NAK setting: Pipe prohibited state where no transaction is made.

(b) BUF setting: Transaction is made according to the FIFO buffer state.
OUT direction: The OUT token is issued when the FIFO buffer contains transmit data.
IN direction: The IN token is issued when the FIFO buffer is not full and can receive data.

(c) STALL setting: Pipe prohibited state where no transaction is made

Note: Make the default control pipe's setup transaction using the DCPCTR.SUREQ bit.

(2) Setting the response PID in function controller operation

For the response PID, specify the response to transactions from the host.

(a) NAK setting: A NAK response is always returned to generated transactions.

(b) BUF setting: A response is returned to transactions according to the FIFO buffer.

(c) STALL setting: A STALL response is always returned to generated transactions.

Note: An ACK response is always returned to the setup transaction regardless of the value of the PID[1:0] bits and the USB request is stored in the register.

The USBHS may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(1) USBHS sets the response PID in host controller operation

(a) NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, stopping issuance of tokens automatically.

- In the response to the transmitted token during a transfer other than the isochronous transfer, an NRDY interrupt occurs (refer to section 41.3.6.2 for details).
- A short packet was received during bulk transfer with the PIPECFG.SHTNAK bit set to 1.
- The transaction counter ended during bulk transfer with the PIPECFG.SHTNAK bit set to 1.

(b) BUF setting: BUF is not written by the USBHS.

(c) STALL setting: The PID[1:0] bits are set to STALL in the following cases, stopping issuance of tokens automatically.

- A STALL response was received in response to the transmitted token.
- A data packet exceeding the maximum packet size was received.

(2) USBHS sets the response PID in function controller operation

(a) NAK setting:

- The SETUP token was successfully received (only default control pipe).
- The PIPECFG.SHTNAK bit was set to 1 during the bulk transfer and the transaction counter ended, or a short packet was received.

(b) BUF setting: BUF is not written by the USBHS.

(c) STALL setting:

- A maximum packet size over error was detected in the receive data packet.
- A control transfer sequence error was detected.

41.3.7.7 Data PID Sequence Bit

When data is successfully transferred in the data stage, bulk transfer, or interrupt transfer of the control transfer, the sequence bit of the data PID is automatically toggled by the USBHS. The sequence bit of the data PID transmitted next can be checked by the SQMON flag in DCPCTR and PIPEnCTR. The sequence bit changes upon receiving the ACK handshake during data transmission, or upon transmitting the ACK handshake during data reception. Furthermore, the data PID sequence bit can be modified by the SQCLR and SQSET bits in DCPCTR and PIPEnCTR.
In the control transfer during function controller operation, the USBHS automatically sets the sequence bit at the stage transition. The data PID becomes DATA1 at the end of the setup stage. The sequence bit is not accessed in the status stage, and a response is returned with PID = DATA1. Therefore, sequence bit setting by software is not necessary. In the control transfer during host controller operation, the sequence bit must be set by software at the stage transition.
Note that the data PID sequence bit must be set by software when transmitting or receiving the ClearFeature() request regardless during the host or function controller operation.

41.3.7.8 Response PID = NAK Function

The USB has a function that disables pipe operation (PID[1:0] bits are set to 00b (NAK response)) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1. When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, the pipe should be set to the enabled state again (PID[1:0] bits are set to 01b (BUF response)) by software. The response PID = NAK function can be used only when bulk transfers are used.

41.3.7.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR bit is 0), OUT-NAK mode is entered, and during an IN transfer (DIR bit is 1), null auto response mode is entered.

41.3.7.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set 01b (BUF response)). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host. To transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)). In normal mode, reception of OUT data is enabled.

41.3.7.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, confirm that the PIPEnCTR.INBUFM flag is 0 because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port. To make a transition from null auto response mode to normal mode, pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)).

41.3.8 FIFO Buffers

41.3.8.1 FIFO Buffers

The USBHS incorporates FIFO buffers for data transfer. The area used for each pipe is managed by the USBHS. Access rights to FIFO buffers can be assigned to the system (CPU side) or the USBHS (SIE side).

(1) Status of the FIFO Buffers

Table 41.20 and Table 41.21 list the states of the FIFO buffers in the USBHS. The state of the FIFO buffers can be confirmed by reading the DCPCTR.BSTS flag and PIPEnCTR.INBUFM flag. The transfer direction for the FIFO buffers can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when default control pipe is selected). The INBUFM flag is valid for pipe 1 to pipe 5 when transmitting. When a transfer pipe on the transmitting side is set to double buffer mode, the BSTS flag is used to monitor the buffer status on the CPU side, and the INBUFM flag is used to monitor the buffer status on the SIE side. When the CPU or DMAC/DTC write access to the FIFO port is slow, and the buffer empty status cannot be determined by the BEMP interrupt, the end of transmission can be confirmed by the INBUFM flag.

Table 41.20 FIFO Buffer Status Indicated by the BSTS Flag

ISEL or DIR Bit	BSTS Flag	FIFO Buffer Status
0 (data reception)	0	There is no received data, or data is being received. Data cannot be read from the FIFO port.
0 (data reception)	1	There is received data, or a zero-length packet has been received. Data can be reading from the FIFO port. Note that when a zero-length packet is received, data cannot be read and the FIFO buffer must be cleared.
1 (data transmission)	0	Data transmission is not complete. Data cannot be written to the FIFO port.
1 (data transmission)	1	Data transmission has been completed. Data can be written to the FIFO buffer.

Table 41.21 Buffer Status Indicated by the INBUFM Flag

DIR Bit	INBUFM Flag	FIFO Buffer Status
0 (data reception)	Invalid	Invalid
1 (data transmission)	0	Data transmission has been completed. There is no data to be transmitted, or transmit data is being written to the FIFO port.
1 (data transmission)	1	Transmittable data is present.

41.3.8.2 Clearing the FIFO Buffers

Table 41.22 lists information pertaining to clearing FIFO buffers by the USBHS. FIFO buffers can be cleared using the BCLR bit in a FIFO port control register, DCLRM bit in the D0FIFOSEL/D1FIFOSEL register, and PIPEnCTR.ACLRM bit. Pipes 1 to 5 can be set to single buffer mode or double buffer mode using the PIPECFG.DBLB bit.

Table 41.22 Methods of Clearing the FIFO Buffers

Cleared	FIFO Buffers on the CPU Side	FIFO Buffer That Has Had All Transmit Data Read	All Data in the FIFO Buffer Allocated to Pipe n
Bit used	CFIFOCTR.BCLR bit D0FIFOCTR.BCLR bit D1FIFOCTR.BCLR bit	D0FIFOSEL.DCLRM bit D1FIFOSEL.DCLRM bit	PIPEnCTR.ACLRM bit
Operation	When the BCLR bit is set to 1, the FIFO buffers on the CPU side of the pipes specified by the CFIFOSEL.CURPIPE[3:0] bits and CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are cleared.	If the DCLRM bit is set to 1, after reading the received data for the pipe specified by the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register, the FIFO buffer is automatically cleared.	If the ACLRM bit is set to 1 and then set to 0, all data in the FIFO buffer (both planes in double buffer mode) allocated to pipe n is cleared. Also, If the PID[1:0] bits are set to 01b (BUF response) while the ACLRM bit is still set to 1, all packets received are discarded.

(1) Received Packet Read-and-Discard Function in Auto Buffer Clear Mode

If the PIPEnCTR.ACLRM bit is set to 1 and the PID[1:0] bits are set to 01b (BUF response), the USBHS discards all received data packets. However, if a normal data packet is received, the ACK response is returned to the host controller. The received packet read-and-discard function can only be set to FIFO buffers that are reading data. In addition, do not use pipes that are specified as isochronous transfer or interrupt transfer.

41.3.8.3 FIFO Port Functions

Table 41.23 shows the settings for the functions of the FIFO ports in the USBHS. When writing data to a FIFO buffer, data is written until the maximum packet size is reached, and the FIFO buffer automatically becomes enabled to transmit data. To enable transmission before the maximum packet size is reached, set the BVAL bit in a FIFO port control register to end the write access. To transmit a zero-length packet, after using the BCLR bit to clear the FIFO buffer, the BVAL bit must be set to end the write access.

When reading data from a FIFO buffer, after all data has been read from the FIFO buffer, it automatically becomes ready to receive new packets. However, when a zero-length packet is received (DTLN[11:0] bits are 000h), data cannot be read from the FIFO buffer, and the BCLR bit must be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[11:0] bits in a FIFO port control register.

Table 41.23 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, D0FIFOSEL, D1FIFOSEL	RCNT	Selects DTLN[11:0] read mode.
	REW	FIFO buffer rewind (reread, rewrite).
	DCLRM	Automatically clears receive data for a current pipe after the data has been read (only for D0FIFO and D1FIFO).
	DREQE	Enables DMA/DTC transfers (only for D0FIFO and D1FIFO).
	MBW[1:0]	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for the default control pipe).
	CURPIPE[3:0]	Selects the current pipe.
CFIFOCTR, D0FIFOCTR, D1FIFOCTR	BVAL	Ends writing to the FIFO buffer.
	BCLR	Clears FIFO buffers on the CPU side.
	DTLN[11:0]	Checks the length of the receive data.

(1) FIFO Port Selection

Table 41.24 shows the pipes that can be selected with the various FIFO ports. Use the CURPIPE[3:0] bits in a FIFO port select register to select the pipe to be accessed. After the pipe is selected, check to see that the value written to the CURPIPE[3:0] bits can be read correctly, (if the previous pipe number is read, it indicates the USBHS is modifying the pipe). Then, check to see if the FRDY flag in a FIFO port control register is 1, and access the FIFO buffers. In addition, use the MBW[1:0] bits in a FIFO port select register to select the bus width to be accessed. The PIPECFG.DIR bit is used to select the FIFO buffer access direction, and the ISEL bit in a FIFO port select register determines the direction of the default control pipe.

Table 41.24 FIFO Port Access by Pipe

Pipe	Access Method	Port that can be Used
Default control pipe	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

The pipe currently being accessed can be halted, another pipe can be accessed, and then it is possible to continue processing on the current pipe. Set the REW bit in a FIFO port select register to perform this. If a pipe is selected by setting the CURPIPE[3:0] bits in a FIFO port select register and the REW bit to 1 simultaneously, the pointer used to read from and write to the FIFO buffer is reset, and the read or write access can be carried out from the first byte. If a pipe is selected by setting the REW bit to 0, the pointer used to read from and write to the FIFO buffer is not reset, and the data access previously selected can be continued. To access a FIFO port, after selecting a pipe, check to see if an FRDY flag in a FIFO port control register is 1.

41.3.8.4 DMA/DTC Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA/DTC Transfers

The FIFO port can be accessed using the DMAC/DTC for pipes 1 to 9. When access to the buffer for the pipe targeted for DMA/DTC transfer is enabled, a DMA/DTC transfer request is issued.
Set the MBW[1:0] bits in the D0FIFOSEL/D1FIFOSEL register to select the transfer width to the FIFO port, set the CURPIPE[3:0] bits to select the pipe targeted for DMA/DTC transfer. Do not change the selected pipe during DMA/DTC transfer.

(2) D0FIFO/D1FIFO Auto Clear Mode (Reading Ports D0FIFO and D1FIFO)

When the DCLRM bit in the D0FIFOSEL/D1FIFOSEL register is set to 1, if the FIFO buffer read access is completed, the USBHS automatically clears the FIFO buffer of the selected pipe.
In Table 41.25, the relationship between packet reception and FIFO buffer clearing processing by software is shown for various bit settings. As shown in Table 41.25, the conditions for clearing a FIFO buffer depends on the PIPECFG.BFRE bit setting. Using the DCLRM bit eliminates the need for a FIFO buffer to be cleared by software in any situation that requires FIFO buffer clearing. This enables DMA/DTC transfers without involving software.
D0FIFO/D1FIFO auto clear mode can be used when reading a FIFO buffer.

Table 41.25 Relationship Between Packet Reception and FIFO Buffer Clearing Processing by Software

Buffer Status When a Packet is Received	Bit Setting			
	DCLRM bit is 0		DCLRM bit is 1	
	BFRE bit is 0	BFRE bit is 1	BFRE bit is 0	BFRE bit is 1
Buffer full	Clearing not necessary	Clearing not necessary	Clearing not necessary	Clearing not necessary
Zero-length packet reception	Clearing necessary	Clearing necessary	Clearing not necessary	Clearing not necessary
Normal short packet reception	Clearing not necessary	Clearing necessary	Clearing not necessary	Clearing not necessary
Transaction count end	Clearing not necessary	Clearing necessary	Clearing not necessary	Clearing not necessary

41.3.8.5 Allocating the FIFO Buffers

Figure 41.12 is an example of the memory map for a USBHS FIFO buffer. The FIFO buffer is an area shared by the CPU and the USBHS. Access rights to the FIFO buffers can be given to the CPU side or USBHS side (SIE side).

An independent area is set for the FIFO buffer of each pipe. A memory area is set by the first block number and the number of blocks (specified by the PIPEBUF.BUFSIZE[4:0] and BUFNMB[7:0] bits), regarding 64 bytes as one block. When the PIPECFG.CNTMD bit is 1 (continuous transfer mode), set the BUFSIZE[4:0] bits to an integral multiple of the maximum packet size. In addition, when the PIPECFG.DBLB bit is set to 1 (double buffer mode), twice the memory area specified by the PIPEBUF.BUFSIZE[4:0] bits is allocated to the same pipe.

Three FIFO ports are used to access (read data from and write data to) a FIFO buffer. Set the CFIFOSEL/D0FIFOSEL/D1FIFOSEL.CURPIPE[3:0] bits to specify the pipe the FIFO port is allocated to.

The BSTS flags in registers DCPCTR and PIPEnCTR, and the PIPEnCTR.INBUFM flag can be used to check the status of a FIFO buffer for each pipe. Use the CFIFOCTR/D0FIFOCTR/D1FIFOCTR.FRDY flag to check access rights for a FIFO port.

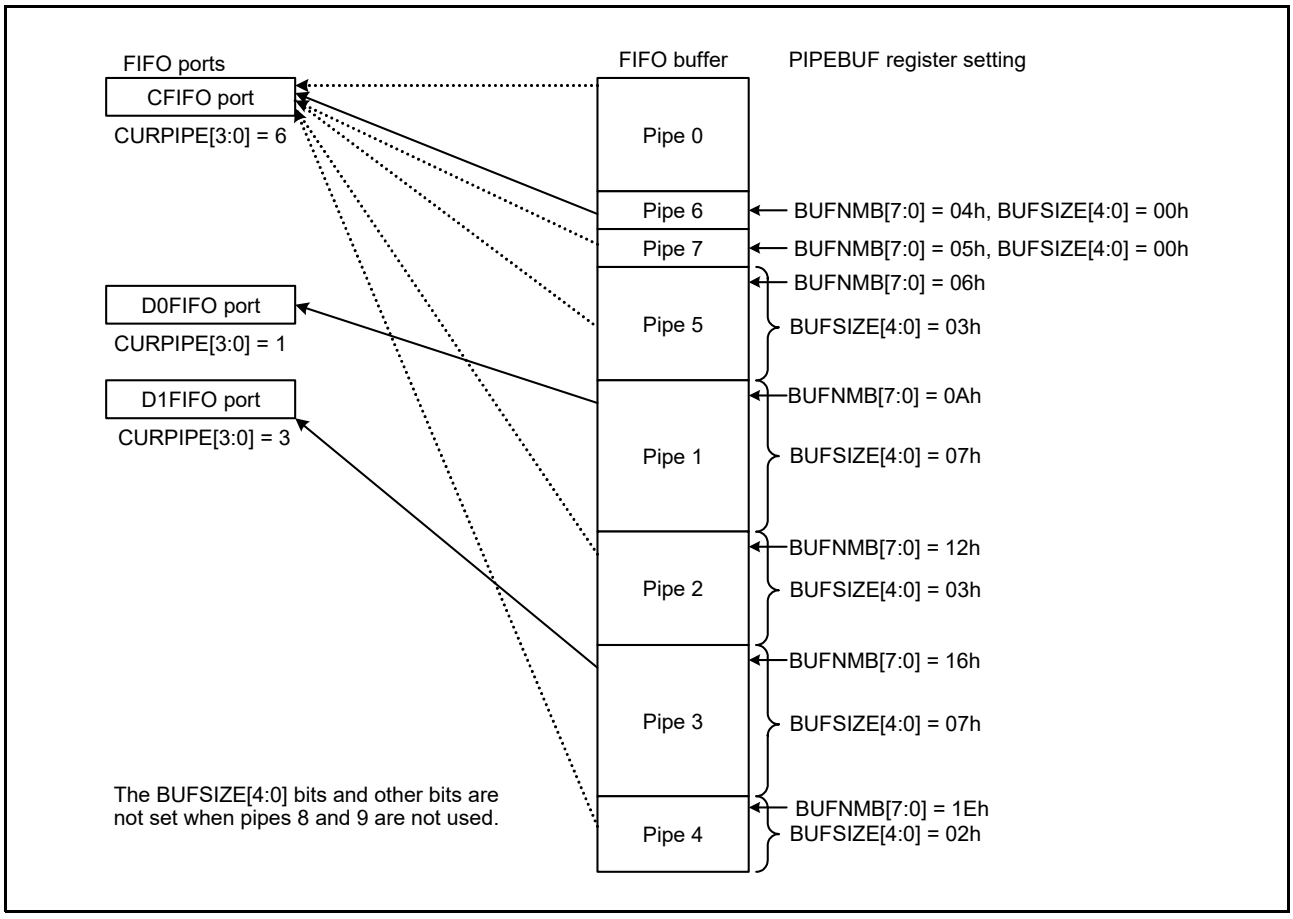


Figure 41.12 Example of a FIFO Buffer Memory Map

41.3.9 Control Transfer Using the Default Control Pipe

The default control pipe is used for data transfer in the control transfer data stage. The default control pipe's FIFO buffer is a 64-byte single buffer of the fixed area for both control read and control write. This FIFO buffer can only be accessed through the CFIFO port.

41.3.9.1 Control Transfer in Host Controller Operation

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are used to transmit the USB request of the SETUP transaction. Writing the setup packet data to the register and then setting the DCPCTR.SUREQ bit to 1 transmits the set data as the SETUP transaction. The SUREQ bit becomes 0 when the transaction is completed. Do not modify the above USB request registers while the SUREQ bit is 1. After the USBHS detects that the peripheral device is connected, issue the first SETUP transaction for the device using the sequence described above by setting the DCPMAXP.DEVSEL[3:0] bits to 0 and setting the DEVADD0.USBSPD[1:0] bits. After the connected peripheral device enters the Address state, issue SETUP transactions using the sequence described above with the assigned USBAddress set in the DEVSEL[3:0] bits and the bits in the DEVADDm register corresponding to the specified USBAddress set appropriately (m = 0 to A). For example, set the DEVADD2 register when the PIPEMAXP.DEVSEL[3:0] bits are 0010b, and set the DEVADD5 register when the PIPEMAXP.DEVSEL[3:0] bits are 0101b. When a transaction is transmitted, an interrupt request is generated by the response from the peripheral device (INTSTS1.SIGN and SACK flags). This interrupt request allows the USBHS to check the SETUP transaction result. The DATA0 data packet (USB request) of the SETUP transaction is transmitted regardless of the DCPCTR.SQMON flag setting.

(2) Data Stage

The data stage is used to transfer data using the default control pipe's FIFO buffer.
Before accessing the default control pipe's FIFO buffer, set the CFIFOSEL.ISEL bit to specify the access direction. In addition, set the DCPCFG.DIR bit to specify the transfer direction. The first data packet in the data stage must be transferred with the data PID set to DATA1. Transactions can be executed by setting the data PID to DATA1 using the DCPCTR.SQSET bit and setting the PID[1:0] bits to 01b (BUF response). Completion of data transfer is detected by the BRDY interrupt or the BEMP interrupt. Data transfer of multiple packets is enabled in continuous transfer mode. If the transmit data size is an integral multiple of the maximum packet size, control the control write transfer by software to transmit a zero-length packet last.

(3) Status Stage

The status stage is used for zero-length packet data transfer in the reverse direction of the data stage. As in the data stage, data transfer using the default control pipe's FIFO buffer is performed in the status stage. Transactions are executed using the same procedure as the data stage. Data packets in the status stage must be transmitted and received with the data PID set to DATA1. Set the data PID to DATA1 by the DCPCTR.SQSET bit. When a zero-length packet is received, check the receive-data length by the CFIFOCTR.DTLN[11:0] bits after the BRDY interrupt is generated, and then clear the FIFO buffer by using the BCLR bit.

41.3.9.2 Control Transfer in Function Controller Operation

(1) Setup Stage

The USBHS returns an ACK response in response to a normal setup packet for the USBHS. The following describes the USBHS operation in the setup stage.
The USBHS sets the following bits upon receiving a new setup packet.

- Sets the INTSTS0.VALID flag to 1.
- Sets the DCPCTR.PID[1:0] bits to 00b (NAK response).
- Sets the DCPCTR.CCPL bit to 0.

When the USBHS receives a data packet following a setup packet, the USBHS stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for the control transfer, be sure to set VALID flag to 0. While VALID flag is 1, the PID[1:0] bits cannot be set to 01b (BUF response) and the data stage cannot be finished. With the VALID flag function, the USBHS can suspend the ongoing request processing upon receiving a new USB request during the control transfer and return a response to the latest request.

Furthermore, the USBHS automatically checks the direction bit (bit 8 of *bmRequestType*), and the request data length (*wLength*) in the received USB request, and recognizes the control read transfer, control write transfer, and control write no data transfer to control stage transitions. In the case of an incorrect sequence, a sequence error of the control transfer stage transition interrupt occurs, and interrupt is reported to software. For details on stage control by the USBHS, see Figure 41.10, Control Transfer Stage Transition Chart.

(2) Data Stage

Perform data transfer by the default control pipe in response to the received USB request. Before accessing the default control pipe's FIFO buffer, set the CFIFOSEL.ISEL bit to specify the access direction. A transaction is executed when the DCPCTR.PID[1:0] bits are set to 10b (BUF response). Completion of data transfer is detected by the BRDY interrupt or BEMP interrupt. Use the BRDY interrupt for control write transfers, and use the BEMP interrupt for control read transfers. When performing a control write in high-speed operation, the response is the NYET handshake depending on the status of the FIFO buffer.

(3) Status Stage

While the DCPCTR.PID[1:0] bits are set to 01b (BUF response), control transfer can be finished by setting the DCPCTR.CCPL bit to 1.
After the DCPCTR.CCPL bit is set to 1, the USBHS automatically executes the status stage according to the data transfer direction determined in the setup stage. The status stage is executed specifically as follows.

- For control read transfer:

The USBHS receives a zero-length packet from the USB host and transmits an ACK response.

- For control write transfer and no data control transfer:

The USBHS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USBHS automatically responds to a normal SetAddress() request. If the SetAddress() request contains any of the following errors, a response must be returned by software.

- When *bmRequestType* is not 00h: Except control write transfer
- When the *wIndex* value is not 00h: Request error
- When the *wLength* value is not 00h: Except no data control transfer
- When the *wValue* value is larger than 7Fh: Request error
- When INTSTS0.DVSQ[2:0] bits are 011b (configured state): control transfer of device state error

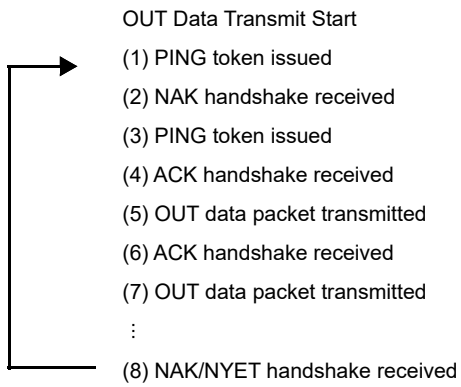
To all requests other than SetAddress(), a response by the corresponding software is required.

41.3.10 Bulk Transfer for Pipes 1 to 5

The FIFO buffer usage (setting of single buffer/double buffer or continuous transfer mode/discontinuous transfer mode) is selectable for the bulk transfer. Up to 2 Kbytes of the FIFO buffer size can be set. The controller manages the FIFO buffer status, and the automatic response is a PING token or NYET handshake.

41.3.10.1 Controlling the PING Token in Host Controller Operation

The USBHS automatically issues the PING token in the OUT direction. As shown below, the USBHS communication in the transmit direction starts with the PING token. When an ACK handshake is received in response to the PING token, the OUT packet is transmitted. If NAK or NYET are received in response to the OUT transaction, the USBHS returns to the PING transmit state.



In addition, sources for the USBHS to return the PING token to the transmit state are a MCU reset, receiving a NYET/NAK handshake, clearing the SQCLR bit, and setting the ACLRM bit to clear the buffer.

41.3.10.2 Controlling the NYET Handshake in Function Controller Operation

Table 41.26 lists the tokens received in response to a bulk transfer and control transfer. When the USBHS receives an OUT token in response to a bulk transfer and control transfer, if there is only enough space in the FIFO buffer for 1 packet of data, the response is NYET. However, if a short packet is received, even if these conditions are met, there is no NYET response, but there is an ACK response.

Table 41.26 Responses for the Token Received

PID[1:0] Bit Setting Value	FIFO Buffer Status	Token Received	Response	Remarks
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY	OUT/PING	ACK	Data packet is received when an OUT token is received *1
	RCV-BRDY	OUT	NYET	Data packet is received *2
	RCV-BRDY	OUT (short)	ACK	Data packet is received *2
	RCV-BRDY	PING	ACK	See Note 2.
	RCV-NRDY	OUT/PING	NAK	
	TRN-BRDY	IN	DATA0/DATA1	Data packet is transmitted
	TRN-NRDY	IN	NAK	

The following are more specific.

Note 1. RCV-BRDY: When an OUT token or PING token is received, the FIFO buffer has at least 2 packets worth of empty space.

Note 2. RCV-BRDY: When an OUT token is received, the FIFO buffer only has 1 packet worth of empty space.
RCV-NRDY: When a PING token is received, there is no empty space in the FIFO buffer.
TRN-BRDY: When an IN token is received, there is transmit data in the FIFO buffer.
TRN-NRDY: When an IN token is received, there is no transmit data in the FIFO buffer.

41.3.11 Interrupt Transfer for Pipes 6 to 9

In function controller operation, the USBHS performs the interrupt transfer according to the cycle managed by the USB host. In the interrupt transfer, the USBHS ignores PING token (no response) and does not transmit the NYET handshake but returns an ACK, NAK, or STALL response.
In host controller operation, the token issuing timing can be set by the interval counter. The USBHS does not issue a PING token but issues an OUT token even for transfer in the OUT direction. The USBHS does not support high-bandwidth interrupt transfers.

41.3.11.1 Interval Counter for Interrupt Transfer in Host Controller Operation

To perform the interrupt transfer, specify the transaction interval for the PIPEPERI.IITV[2:0] bits. The USBHS issues interrupt transfer tokens according to the set interval.

(1) Initializing the Counter

When the MCU is reset, or when the PIPEnCTR.ACLRM bit is set to 1, the interval counter is initialized. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is initialized.

Note that the interval counter is not initialized in the following case.

- USB bus reset, USB suspend
  The IITV[2:0] bits are not initialized. When the DVSTCTR0.UACT bit is set to 1, the count continues from the value before entering the USB bus reset state or the USB suspend state.

(2) Operation when Tokens Cannot Be Transmitted or Received at the Token Generation Timing

No token is generated in the following cases even at the token generation timing. In this case, try to execute a transaction in the next interval.

- The PID[1:0] bits are set to NAK response or STALL response.
- No available area is left in the FIFO buffer at the token transmit timing during transfer in the IN (receiving) direction.
- The FIFO buffer contains no transmit data at the token transmit timing during transfer in the OUT (transmitting) direction.

41.3.12 Isochronous Transfer for Pipes 1 and 2

The USBHS has the following functions for isochronous transfers.

- Notification of isochronous transfer error
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLE function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

The USBHS does not support high-bandwidth isochronous transfers.

41.3.12.1 Error Detection in Isochronous Transfers

The USBHS has a function for detecting the following error information for software to manage isochronous transfer errors. Table 41.27 and Table 41.28 show the priority in which errors are detected by the USBHS and the interrupts generated corresponding to errors.

(1) PID errors
- The PID value of the received packet is invalid.

(2) CRC errors and bit stuffing errors
- A CRC error was found in the received packet or the bit stuffing is incorrect.

(3) Maximum packet size over
- The data size of the received packet exceeded the maximum packet size.

(4) Overrun and underrun errors
- In host controller operation:
  No available area is left in the FIFO buffer at the token transmit timing during transfer in the IN (receiving) direction.
  The FIFO buffer contains no transmit data at the token transmit timing during transfer in the OUT (transmitting) direction.
- In function controller operation:
  The FIFO buffer contains no data when an IN token is received during transfer in the IN (transmitting) direction.
  No available area is left in the FIFO buffer when an OUT token is received during transfer in the OUT (receiving) direction.

(5) Interval error
In function controller operation, the following cases are treated as an interval error.
- Failed to receive an IN token in the interval frame during the isochronous IN transfer.
- Failed to receive an OUT token in the interval frame during the isochronous OUT transfer.

Table 41.27 Detecting an Error When a Token is Issued or Received

Detection Priority	Error Type	Interrupt Generated at Error Detection and Status
1	PID error	No interrupt is generated regardless of the selected controller function. (Ignored as a corrupted packet.)
2	CRC error, bit stuffing error	No interrupt is generated regardless of the selected controller function. (Ignored as a corrupted packet.)
3	Overrun error, underrun error	An NRDY interrupt is generated and the OVRN flag is set to 1 regardless of the selected controller function. In function controller operation, a zero-length packet is transmitted in response to the IN token. No data packet is received in response to the OUT token.
4	Interval error	An NRDY interrupt is generated in function controller operation, but no interrupt is generated in host controller operation.

Table 41.28 Detecting an Error When Receiving a Data Packet

Detection Priority	Error Type	Interrupt Generated at Error Detection and Status
1	PID error	No interrupt is generated (ignored as a corrupted packet.)
2	CRC error, bit stuffing error	An NRDY interrupt is generated and the CRCE flag is set to 1 regardless of the selected controller function.
3	Maximum packet size over error	A BEMP interrupt is generated and the PID[1:0] bits are set to STALL response regardless of the selected controller function.

41.3.12.2 Data PID

The USBHS does not support high-bandwidth transfers. The following describes actions to be taken for the received token in function controller operation.

(1) IN token:
- DATA0: Transmitted as PID of the data packet.
- DATA1: Not transmitted.
- DATA2: Not transmitted.
- mDATA: Not transmitted.

(2) OUT token (full-speed operation):
- DATA0: Successfully received as PID of the data packet.
- DATA1: Successfully received as PID of the data packet.
- DATA2: The packet is ignored.
- mDATA: The packet is ignored.

(3) OUT token (high-speed operation):
- DATA0: Successfully received as PID of the data packet.
- DATA1: Successfully received as PID of the data packet.
- DATA2: Successfully received as PID of the data packet.
- mDATA: Successfully received as PID of the data packet.

41.3.12.3 Interval Counter

The isochronous transfer interval can be set by the PIPEPERI.IITV[2:0] bits. In function controller operation, the interval counter enables functions shown in Table 41.29 below. In host controller operation, the token issuing timing is generated. In host controller operation, the interval counter operates in the same manner as in the interrupt transfer.

Table 41.29 Interval Counter Functions Provided in Function Controller Operation

Transfer Direction	Function	Detecting Condition
IN	Transmit buffer flush function	Failed to receive an IN token successfully in the interval frame during the isochronous IN transfer.
OUT	Notification of reception of no token	Failed to receive an OUT token successfully in the interval frame during the isochronous OUT transfer.

Since the interval counting is performed upon reception of (μ)SOF or by the recovered (μ)SOF, the isochronism can be maintained even if the (μ)SOF packet is corrupted. The settable frame interval is $2^{IITV[2:0]}$ (micro)frame.

(1) Initializing the Interval Counter in Function Controller Operation

When the MCU is reset, a USB bus reset is detected, or when the PIPEnCTR.ACLRM bit is set to 1, the interval counter is initialized. Note that the PIPEPERI.IITV[2:0] bits are initialized when the MCU is reset.

After the interval counter is initialized, the interval count starts under the following conditions after a packet is successfully transferred.

1. (μ)SOF is received after data is transmitted in response to the IN token with PID[1:0] bits set to 01b (BUF response).
2. (μ)SOF is received after data is received in response to the OUT token with PID[1:0] bits set to 01b (BUF response).

The interval counter is not initialized under the following conditions.

- When PID[1:0] bits are set to NAK response or STALL response
  The interval timer does not stop. Try to execute a transaction in the next interval.
- USB suspend
  The IITV[2:0] bits are not initialized. When (μ)SOF is received, the interval counter starts counting from the value before entering the suspended state.

(2) Interval Counting and Transfer Control in Host Controller Operation

The USBHS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBHS issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USBHS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF response).

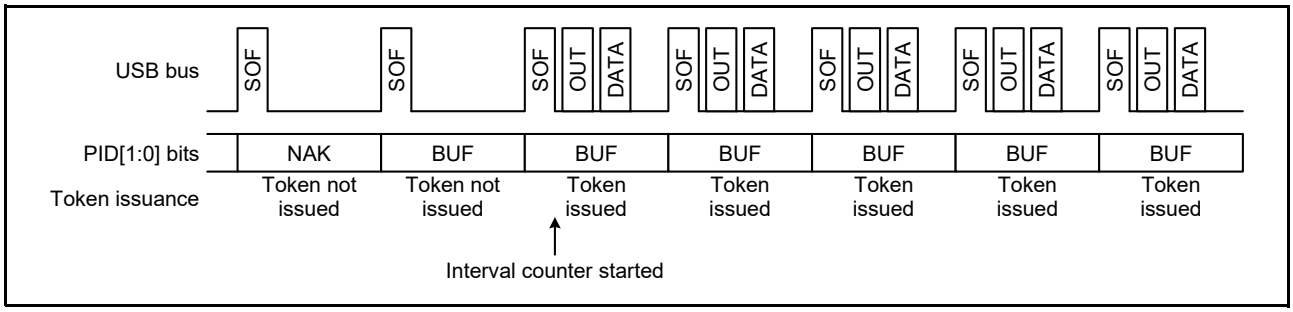


Figure 41.13 Token Issuance When IITV[2:0] Bits are 000b

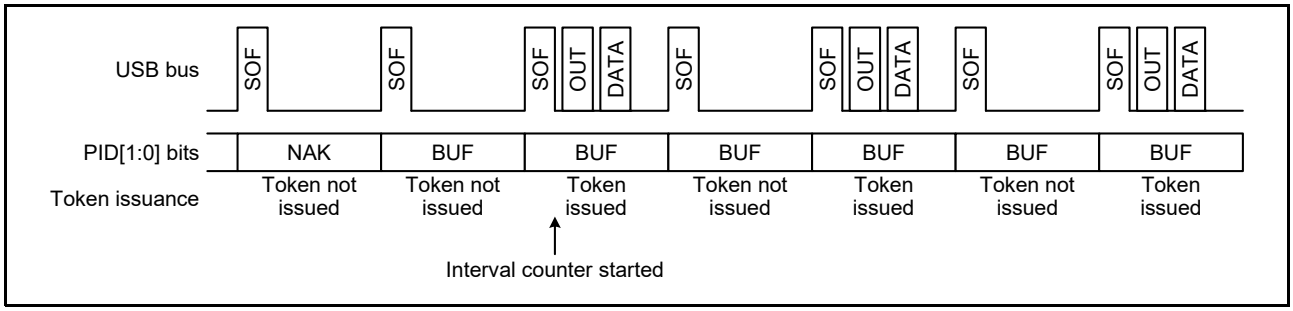


Figure 41.14 Token Issuance When IITV[2:0] Bits are 001b

When the selected pipe is set for isochronous transfers, the USBHS carries out the following operation in addition to controlling the token issuance interval. The USBHS issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USBHS generates an NRDY interrupt when the USBHS issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USBHS sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USBHS cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USBHS sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the MCU is reset
  (The IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit has been set to 1.

(3) Interval Counting and Transfer Control in Function Controller Operation

(a) When the selected pipe is for isochronous OUT transfers

The USBHS generates an NRDY interrupt when the USBHS fails to receive a data packet during the frame over the interval set by the PIPEPERI.IITV[2:0] bits.
The USBHS also generates an NRDY interrupt when the USBHS fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of (µ)SOF packet reception. Even if the (µ)SOF packet is corrupted, the internal recovery function allows the interrupt to be generated at the timing to receive the (µ)SOF packet.
However, when the IITV[2:0] bits are set to a value other than 000b, the USBHS generates an NRDY interrupt on receiving an (µ)SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK response) after starting the interval timer, the USBHS does not generate an NRDY interrupt on receiving an (µ)SOF packet.
The condition to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] bits are 000b:
  The interval counting starts when the PID[1:0] bits of the selected pipe are modified to BUF.

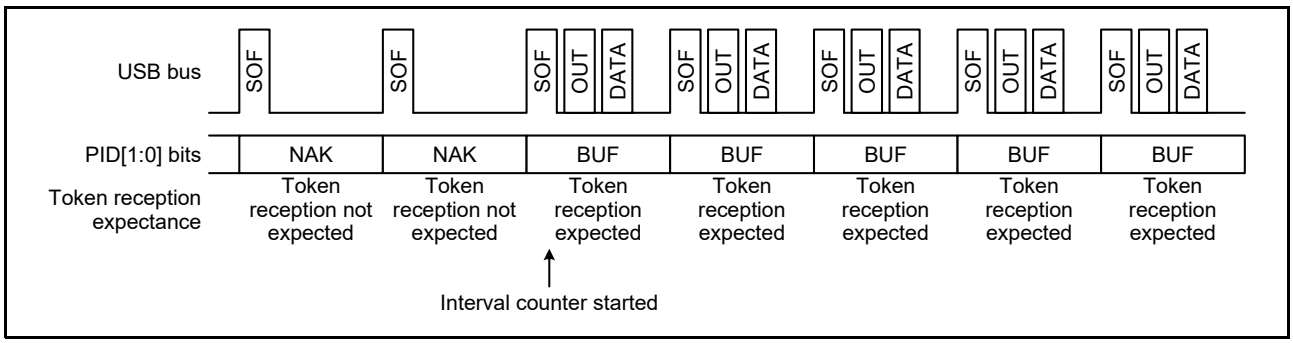


Figure 41.15 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 000b

- When the IITV[2:0] bits are a value other than 000b:
  The interval counter starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF response).

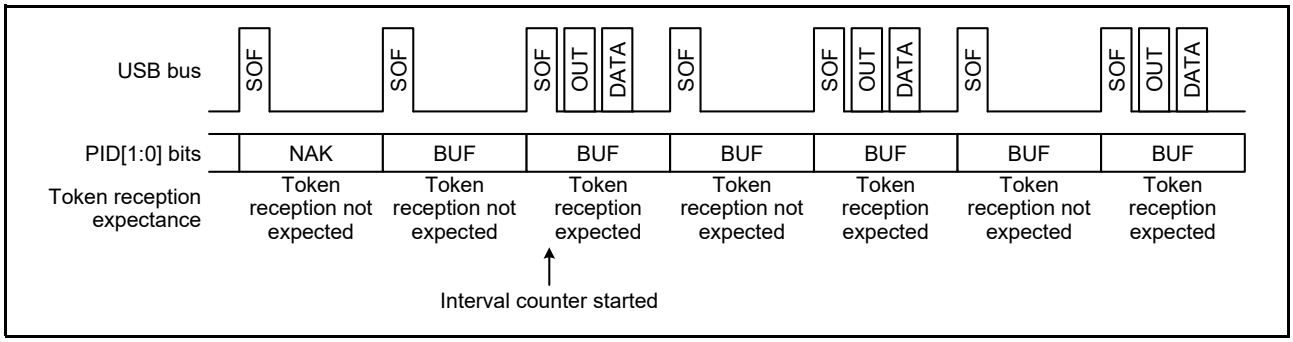


Figure 41.16 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 001b

(b) When the selected pipe is for isochronous IN transfers

When the PIPEPERI.IFIS bit is cleared to 0, the USBHS transmits a data packet in response to a received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When the IFIS bit is set to 1, since the FIFO buffer contains data that can be transmitted, the USBHS clears the FIFO buffer if it fails to receive an IN token during the frame over the interval set by the IITV[2:0] bits.
The USBHS also clears the FIFO buffer when the USBHS fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of (µ)SOF packet reception. Even if the (µ)SOF packet is corrupted, the recovery function allows the FIFO buffer to be cleared at the timing to receive the (µ)SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval counter is cleared when any of the following conditions is fulfilled in function controller operation.

- When the MCU is reset (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1.
- When a USB bus reset is detected.

(4) Isochronous Transfer Transmit Data Setup in Function Controller Operation

In function controller operation, in the isochronous data transmission of the USBHS, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the (μ)SOF packet is detected. This function is called the isochronous transfer transmit data setup function. When double FIFO mode is used, only the one buffer that finished writing earlier can enter the transmittable state even when writing to both buffers has been completed. For this reason, even if two or more IN tokens are received in the same frame, only one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transmitted and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 41.17 shows an example of transmission using the isochronous transfer transmit data setup function when the IITV[2:0] bits are set to 000b (each frame) by the USBHS.

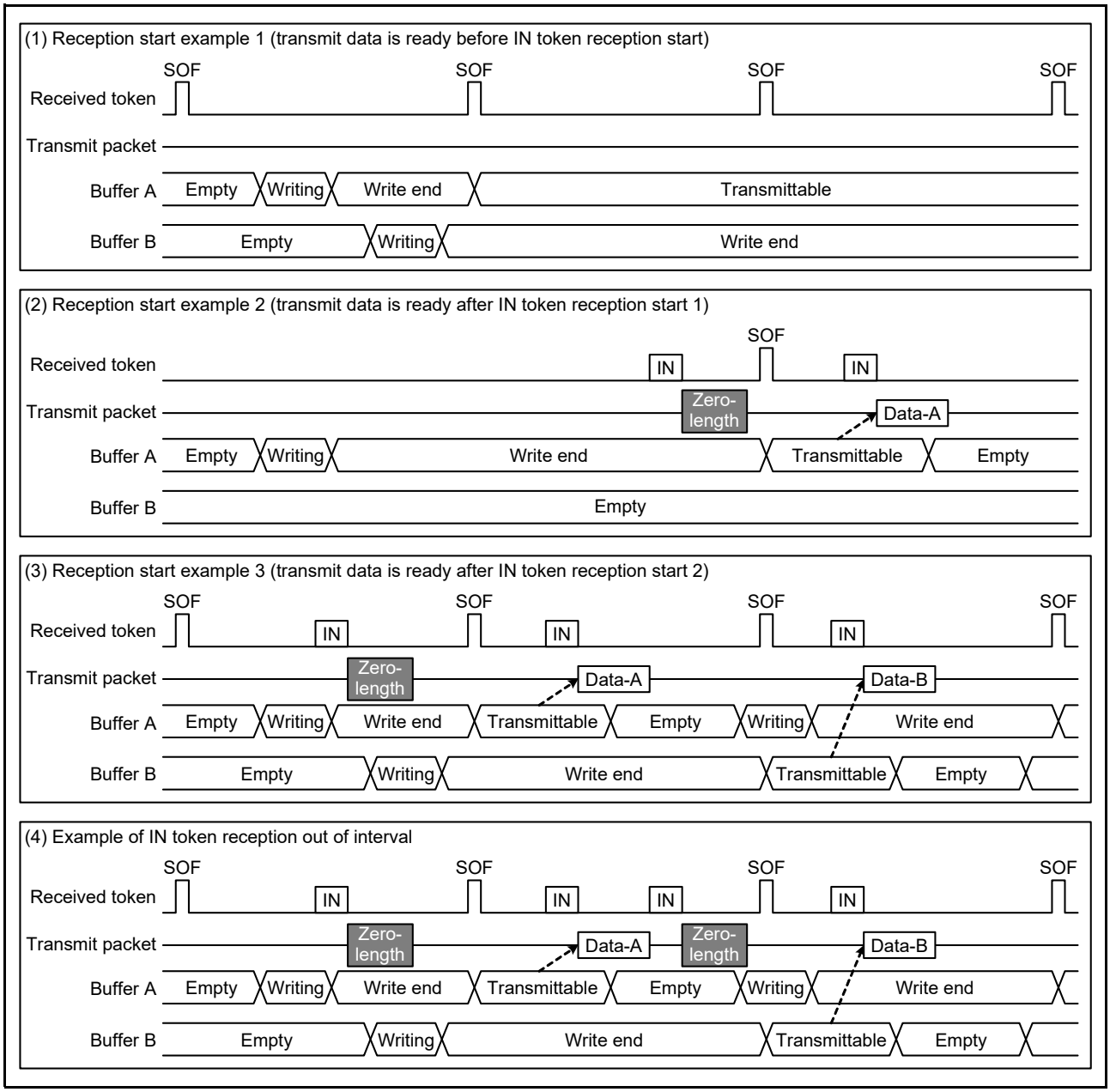


Figure 41.17 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmit Buffer Flush in Function Controller Operation

In function controller operation, the USBHS does not receive an IN token in the interval frame during the isochronous data transmission. When the USBHS receives the (µ)SOF packet of the next frame, the USBHS treats it as a corrupted IN token and clears the buffer that is ready to transmit data to make it writable.

When the double-buffer is used and data has already been written to both of the buffers, the USBHS regards that the discarded FIFO buffer was transmitted in the same interval frame and makes the FIFO buffer (that is not discarded after the next (µ)SOF packet was received) transmittable.

The operation start timing of the buffer flush function varies depending on the value of the PIPEPERI.IITV[2:0] bits.

(1) When the PIPEPERI.IITV[2:0] bits are 000b, the buffer flush operation starts from the first frame after the pipe is enabled.

(2) When the PIPEPERI.IITV[2:0] bits are not 000b, the buffer flush operation starts after the first normal transaction.

Figure 41.18 shows an example of operation of the buffer flush function of the USBHS. However, for tokens prior to the interval frame (out of interval), the written data or a zero-length packet as an underrun error is transmitted according to the data setup status.

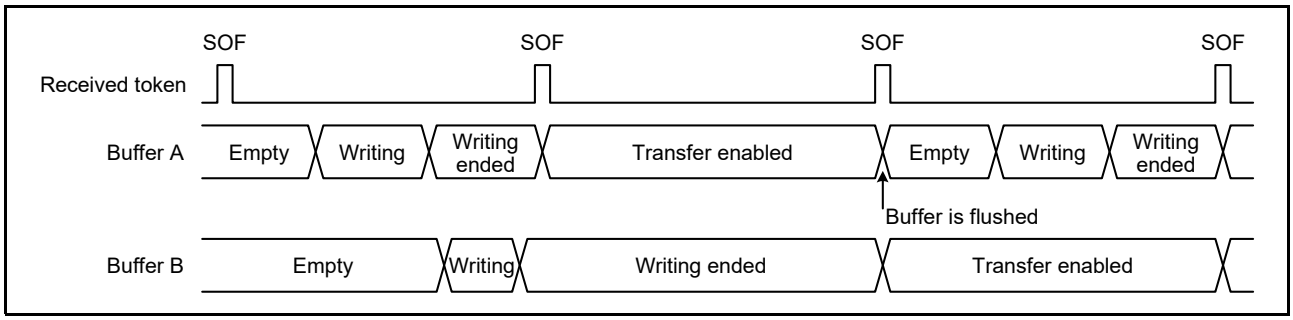


Figure 41.18 Example of Buffer Flush Operation

Figure 41.19 shows an example of occurrence of interval errors of the USBHS. There are five types of interval errors ((2) to (6)) shown in the figure below. An interval error occurs at timing ① and the buffer flush function works. Occurrence of an interval error during the IN transfer activates the buffer flush function, and occurrence of an interval error during the OUT transfer generates an NRDY interrupt.
Distinguish NRDY interrupt (such as receive packet error) from overrun error by the FRMNUM.OVRN flag.
A response according to the FIFO buffer status is returned to shaded tokens.

(1) IN token:

- When the FIFO buffer is ready to transmit data, the data is transmitted and a normal response is returned.
- When the FIFO buffer is not ready to transmit data, a zero-length packet is transmitted and an underrun error occurs.

(2) OUT token:

- When the FIFO buffer is ready to receive data, the data is received and a normal response is returned.
- When the FIFO buffer is not ready to receive data, the received data is discarded and an overrun error occurs.

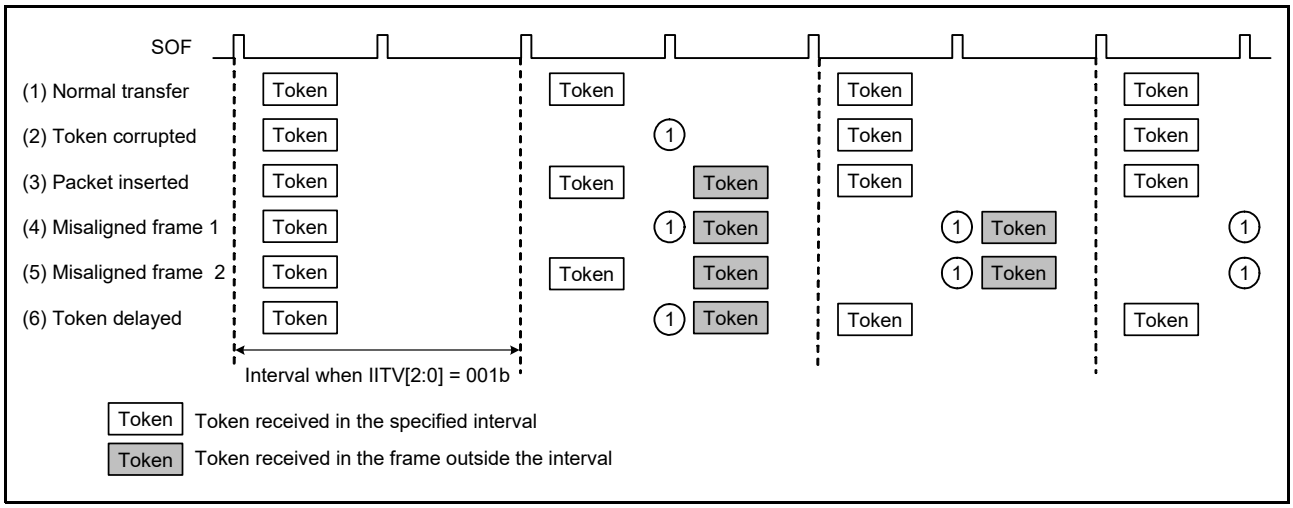


Figure 41.19 Example of Interval Error Occurrence When the PIPEPERI.IITV[2:0] Bits are 001b

41.3.13 SOF Recovery Function

In function controller operation, if an (µ)SOF packet is not received at intervals of 1 ms (full-speed operation) or 125 µs (high-speed operation) due to a corrupted or lost (µ)SOF packet, SOF recovery is made in the controller. The SOF recovery starts working when an (µ)SOF packet is received while the SYSCFG.USBE bit is set to 1 and the LPSTS.SUSPENDM bit is set to 1. The recovery function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Detection of suspension

The SOF recovery function works with the following specifications.

(1) The frame interval (125 µs or 1 ms) follows the reset handshake protocol result.
(2) The recovery function does not work until an (µ)SOF packet is received.
(3) After the first (µ)SOF packet is received, 125 µs or 1 ms is counted by the internal 60-MHz clock to recover (µ)SOF packets.
(4) After the second (or subsequent) (µ)SOF packet is received, (µ)SOF packets are recovered using the previous reception interval.
(5) Recovery is not made during suspension or reception of a USB bus reset.
(In high-speed operation, when transitioning to suspend mode, recovery continues for 3 ms from the last packet.)

The USBHS carries out the following operations when receiving the (µ)SOF packet reception. These operations also operate normally with SOF recovery, if the (µ)SOF packet was missing.

(1) Updating frame number and microframe number
(2) Frame number refresh interrupt and µSOF lock
(3) Isochronous transfer interval count

If an SOF packet is lost during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.
If a µSOF packet is lost during high-speed operation, the UFRMNUM.UFRNM[2:0] bits are updated. However, when the UFRNM[2:0] bits are set to 000b and a µSOF packet is lost, the FRNM[10:0] bits is not updated. In this case, even if the next µSOF packet for a UFRNM[2:0] bit value other than 000b is received successfully, the FRNM[10:0] bits are not updated.

41.3.14 Pipe Schedule

41.3.14.1 Conditions for Generating a Transaction

In host controller operation, after the DVSTCTR0.UACT bit has been set to 1, the USBHS generates a transaction under the conditions shown in Table 41.30.

Table 41.30 Conditions for Generating a Transaction

Transaction	Conditions for Generation				
	DIR	PID[1:0]	IITV[2:0]	Buffer State	SUREQ
Setup	— *1	— *1	— *1	— *1	Set 1
Control transfer data stage, status stage, bulk transfer	IN	BUF	— *1	Receive area exists	— *1
	OUT	BUF	— *1	Transmit data exists	— *1
Interrupt transfer	IN	BUF	Valid *2	Receive area exists	— *1
	OUT	BUF	Valid *2	Transmit data exists	— *1
Isochronous transfer	IN	BUF	Valid *2	See Note 3.	— *1
	OUT	BUF	Valid *2	See Note 4.	— *1

Note 1. The hyphen (—) in the table indicates that the condition is unrelated to the generating of tokens.
Note 2. “Valid” indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter.
Note 3. A transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 4. A transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

41.3.14.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USBHS. After the USBHS transmits an (µ)SOF packet, the transfer is carried out in the sequence described below.

1. Execution of interrupt transfers and isochronous transfers
   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   If a setup transaction is possible, it is transmitted.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of default control pipe → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

41.3.14.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates (µ)SOF packet transmission and transaction generation is enabled. Setting the UACT bit to 0 stops (µ)SOF packet transmission and the USBHS enters the suspend state. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next (µ)SOF packet is transmitted.

41.3.15 Battery Charging Detection Processing

Data Contact Detect processing (checking D+ line contact), Primary Detection processing (charger detection processing), and Secondary Detection processing (charger determination processing) defined by Battery Charging Specification Revision 1.2 can be controlled.
Section 41.3.15.1 and section 41.3.15.2 describe operations in host controller operation and function controller operation.

41.3.15.1 Processing in Function Controller Operation

To operate a USBHS as a battery charging Portable Device, the following processing is required. Figure 41.20 shows the processing flowchart.

(1) VBUS supply is detected by the VBUS interrupt and VBSTS flag.

(2) Start Primary Detection processing after detecting contact with the D+ and D– lines. Battery Charging Specification Revision 1.2 describes two processing methods for Data Contact Detect. The USBHS supports both methods in the ways described below.

- Software wait
  Perform a 300 to 900 ms software wait
- Detecting contact by hardware
  Apply 7 to 13 µA of current to the D+ line to hold the D+ line at the logical high level. This is done to detect the D+ and D– lines going to the logical low level due to pull-down resistors in the host when the D+ and D– lines come in contact with those of the host. The USBHS sets the BCCTRL.IDPSRCE bit to 1 and sets the D+ line high. When the D+ line is detected as becoming low by the SYSSTS0.LNST[1:0] bits, the BCCTRL.IDPSRCE bit is set to 0.

(3) Set the BCCTRL.VDPSRCE bit and BCCTRL.IDMSINKE bit to 1 simultaneously to start the Primary Detection processing. Then, perform a 40 ms software wait, and check the BCCTRL.CHGDETSTS flag. If the CHGDETSTS flag is 1, it is determined that a charger is detected, and the BCCTRL.VDPSRCE and BCCTRL.IDMSINKE bits are set to 0.

(4) Set the BCCTRL.VDMSRCE bit and BCCTRL.IDPSINKE bit to 1 simultaneously to start the Secondary Detection processing. Then, perform a 40 ms software wait, check the BCCTRL.PDDETSTS flag, and determine the result of the Secondary Detection processing.

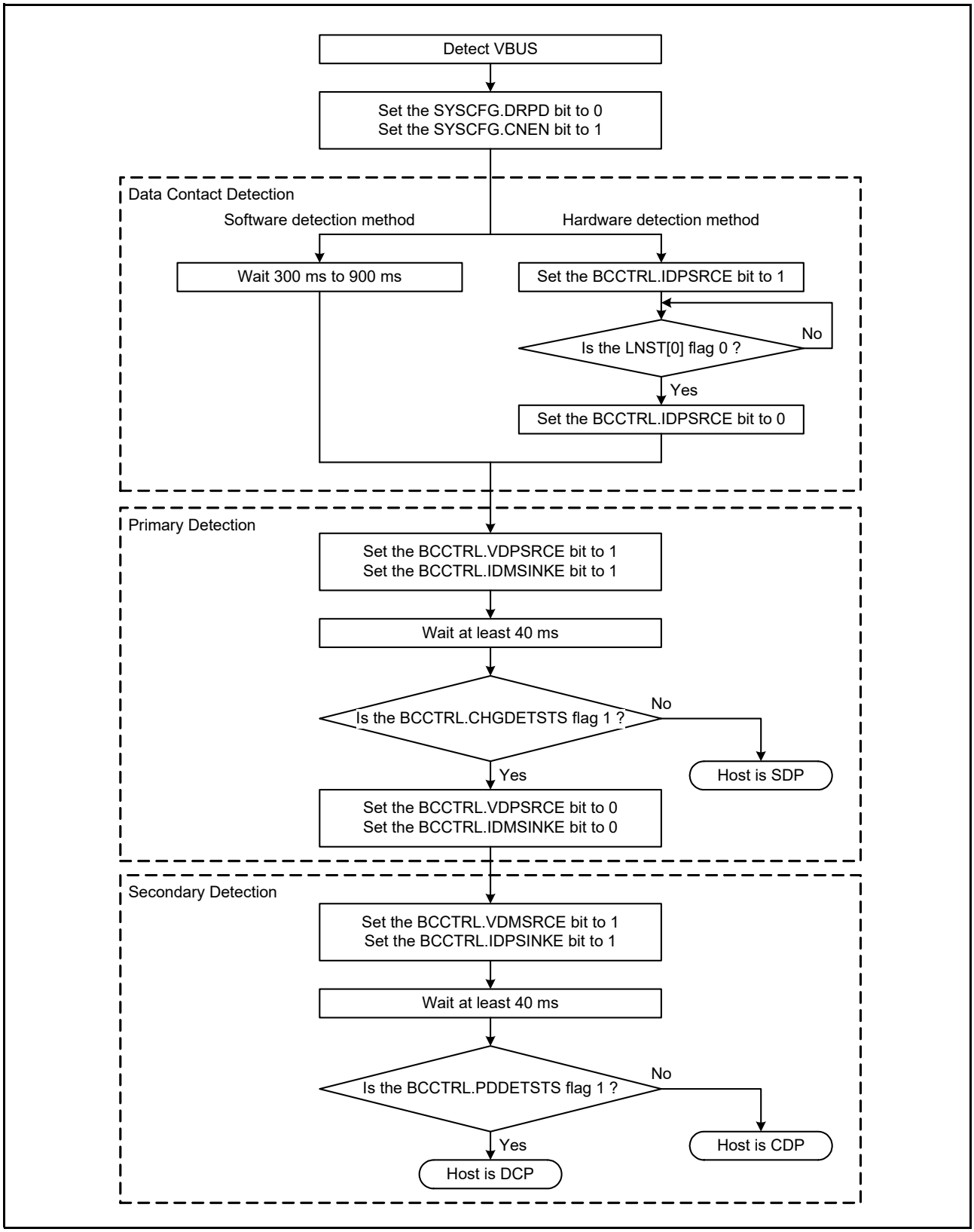


Figure 41.20 Processing as a Portable Device

41.3.15.2 Processing in Host Controller Operation

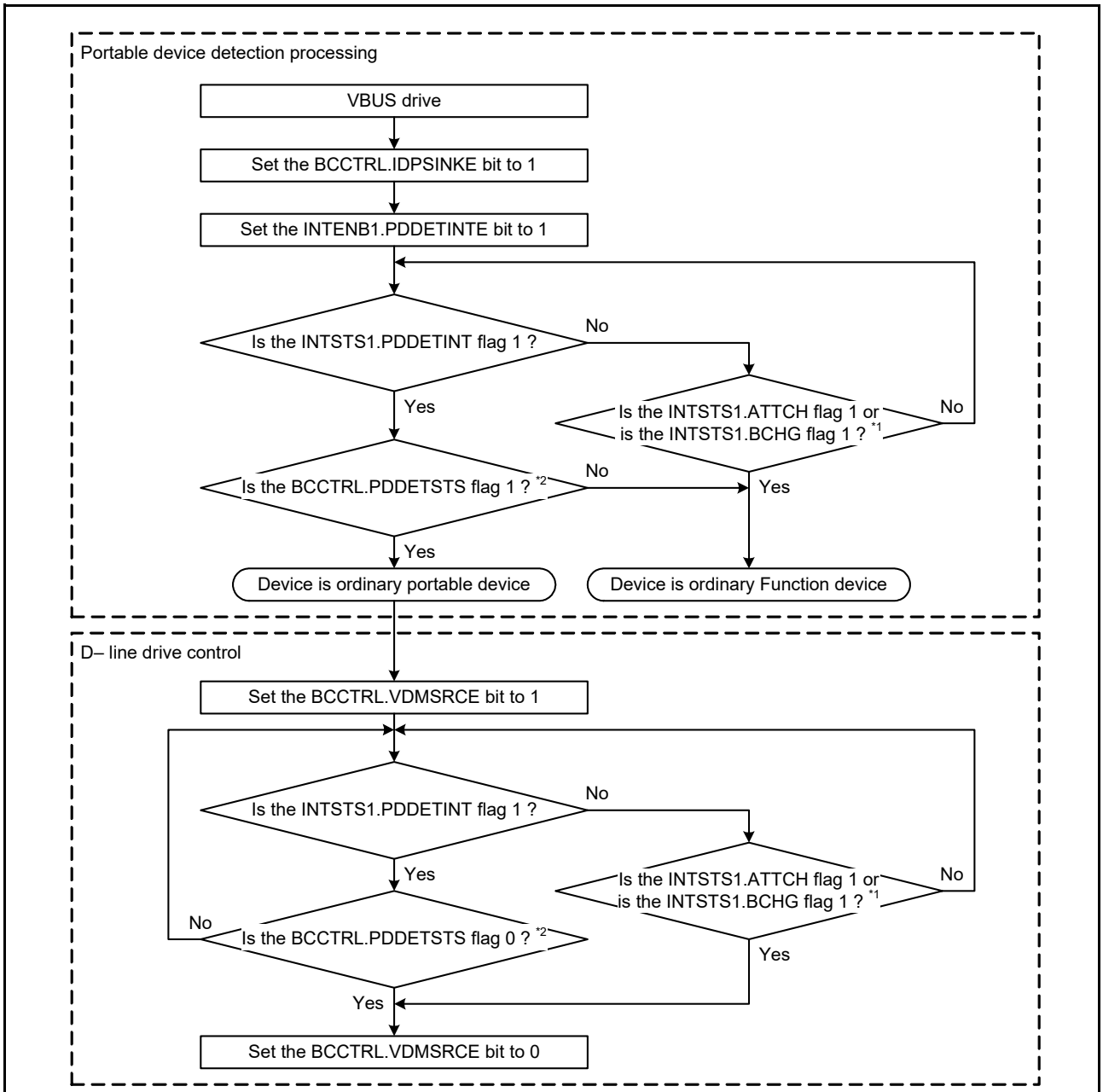
In host controller operation, driving the D– line is required for a Portable Device to implement Primary Detection. Two Primary Detection methods are provided: one for when the hardware has a Portable Device detection function and the other for when the hardware does not have the function or the function is not used even though the hardware has it. Figure 41.21 and Figure 41.22 show processing for the respective methods.

- When using the Portable Device detection function
  The D– line is driven when a Portable Device is detected.

(1) Start driving the VBUS.
(2) Set the BCCTRL.IDMSINKE bit to 1 to enable the Portable Device detection circuit.
(3) Read the BCCTRL.PDDETSTS flag in the Portable Device detection interrupt processing. If the PDDETSTS flag is 1, set the BCCTRL.VDMSRCE bit to 1.
(4) When the BCCTRL.PDDETSTS flag is read in the Portable Device detection interrupt processing after driving the D– line, if the PDDETSTS flag is 0, set the BCCTRL.VDMSRCE bit to 0.

- When not using the Portable Device detection function
  Regardless of whether or not a Portable Device was detected, drive the D– line after device disconnection is detected, and leave the D– line open after connection is detected. Software handles the timing of (1) and (2).

(1) After device disconnection is detected, start driving the D– line within 200 ms.
(2) After device connection is detected, stop driving the D– line within 10 ms.



Note 1. If the PHY clock is being supplied (LPSTS.SUSPENDM bit is 1), the determination is made by the device connection detection interrupt; if the clock is stopped (SUSPENDM bit is 0), the determination is made by the bus change interrupt and the SYSSTS0.LNST[1:0] flags.
Note 2. Read the flag several times to eliminate chattering.

Figure 41.21 Processing Flow When Using the Portable Device Detection Function

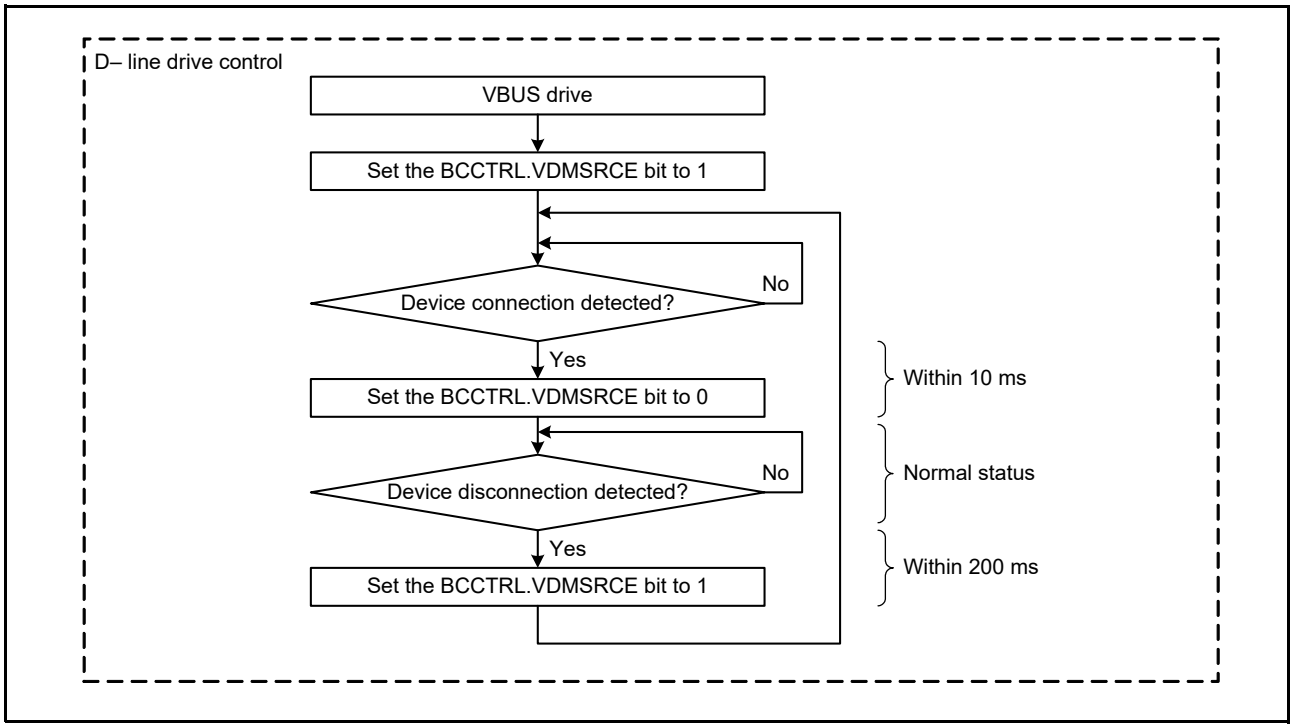


Figure 41.22 Processing Flow When Not Using Portable Device Detection Function

41.3.16 Link Power Management Processing

The Link Power Management standard defines the existing suspended state as the L2 state, and newly defines the L1 state as a state that can be entered into and exited from faster than the L2 state. Table 41.31 is a Comparison of the L1 State and L2 State.

Table 41.31 Comparison of the L1 State and L2 State

Item	L1 State	L2 State
Transition	LPM transaction	Idle for 3 ms
Exit caused by host	(1) Host Drive the K state for a period specified by the HIRD[3:0] bits. Drive period can be specified from 75 µs to 9.96 ms. (2) Function Exits within 10 µs	(1) Host Drive the K state for at least 20 ms (2) Function Exits within 10 ms
Exit caused by function	(1) Function Drive the K state for 50 µs (2) Host Drive the K state for 60 to 990 µs	(1) Function Drive the K state for 1 to 15 ms (2) Host Drive the K state for at least 20 ms
Signaling	Low- and full-speed idle	Low- and full-speed idle
Enabling remote wakeup	Specified by the LPM token	Specified by the SetFeature() request

41.3.16.1 Processing in Function Controller Operation

(1) Contents of the Descriptor

In function controller operation, the USBHS must return its Descriptor when a GetDescriptor() request is received. The content of the Descriptor to be returned must be modified depending on whether the USBHS accepts or denies transition to or exit from the L1 state according to the LPM transaction. Table 41.32 shows the Relationship Between LPM Support and the Descriptor.

Table 41.32 Relationship Between LPM Support and the Descriptor

LPM Support	*bcdUSB* Field	USB 2.0 Extension Descriptor		Response to Received LPM Request	Note
		Provided/ Not provided	LPM bit		
No	0200h	Not provided	—	No response	Normal operation when the LPM is not supported
	0201h	Provided	0	STALL	Setting when it is obvious that LPM is not supported. In this case, a STALL response must be returned.
Yes	0201h	Provided	1	ACK or NYET	Normal operation when the LPM is supported

The LPM bit in USB 2.0 Extension Descriptor declares whether the USBHS accepts or denies transition to or exit from the L1 state. To provide USB 2.0 Extension Descriptor, the *bcdUSB* field of the Standard Device Descriptor must be set to a value of 0201h or larger.

When the LPM is not supported, the USB 2.0 Extension Descriptor is not provided and the *bcdUSB* field value must be 0200h. Even if an LPM token is received in this case, it must be ignored.

When the LPM is not clearly supported, it is also possible to set *bcdUSB* field value to 0201h and the LPM bit in USB 2.0 Extension Descriptor to 0 (no support for the LPM token). In this case, however, the LPM token cannot be ignored and a STALL response must be returned.

When the LPM token is supported, set *bcdUSB* field value to 0201h and set the LPM bit in USB 2.0 Extension Descriptor to 1 (supports the LPM token). This allows acknowledgment of returning a NYET or ACK response to the LPM token.

(2) Processing When Receiving an LPM Token

The following describes the procedure to enter and exit the L1 state in function controller operation.

(1) When the USBHS receives an LPM token from the host, the settings of the PL1CTRL1.L1RESPEN, L1RESPMD[1:0], and L1NEGOMD bits determines whether to return an ACK, NYET, or STALL response, or ignore.

(2) When an LPM token is received and an ACK response is returned, if the LPM token is not retransmitted within 8 µs, the USBHS enters the L1 state. The USBHS processes retransmission of the LPM token and transition to the L1 state. The device state transition interrupt can be used to detect the transition to the L1 state.

(3) There are two types of processing for exiting the L1 state:

- When the host starts the resume:
  When the resume signal is detected, the resume interrupt is generated and the function starts processing to exit the L1 state.
- When the function outputs a remote wakeup signal:
  If the DVSTCTR0.WKUP bit is set to 1, a remote wakeup signal is output to the host. When exiting the L2 state, the DVSTCTR0.WKUP bit must be set to 0 by software, but the USBHS sets the DVSTCTR0.WKUP bit to 0 when exiting from the L1 state.

(3) BESL Field Value Negotiation Function

The BESL field value included in the LPM token is the period that the outputs the resume signal when returning from the L1 state. The BESL field value can be adjusted to accommodate the target system. For example, a low HIRD threshold value must be set for a system that focuses on improving transfer efficiency, while a high HIRD threshold value must be set for a system that focuses on low power consumption.

As per the PL1CTRL1.L1NEGOMD and HIRDTHR[3:0] bit settings, an ACK response or NYET response is returned when the received BESL field value is compared to the HIRD threshold value, and a request can be made to the host to modify the BESL field value.

41.3.16.2 Processing in Host Controller Operation

(1) Processing When Transmitting an LPM Token

The following describes the procedure to enter and exit the L1 state in host controller operation.

(1) When the HL1CTRL1.L1REQ bit is set to 1, an LPM token is transmitted from the host to the function.

(2) If an ACK response is received from the function, a transition to the L1 state starts within 10 µs and is completed within 50 µs. In this case, if a transaction error is detected, the LPM token starts retransmitting within 8 µs. Retransmission can be performed up to two times. The USBHS controls all of this processing.

(3) There are two types of processing for returning from the L1 state.

- When the host starts the resume:
  When the DVSTCTR0.RESUME bit is set to 1, the USBHS starts outputting the resume signal and starts the processing to exit the L1 state.
- When the function outputs a remote wakeup signal:
  When the USBHS detects a remote wakeup signal being output from the function, the USBHS sets the DVSTCTR0.RESUME bit *1 to 1 and starts outputting the resume signal.

Note 1. Unlike exiting from the L2 state, the USBHS sets the DVSTCTR0.RESUME bit to 0. When the RESUME bit becomes 0, the DVSTCTR0.UACT bit becomes 1, and the L1RSMEND interrupt request is generated.

41.3.17 USB External Connection Circuit

Figure 41.23 shows an example of the USB Connector's OTG Connection in Self Powered Mode.
The USBHS controls the pull-up resistor of the D+ line, and the pull-down resistor of the D+ and D– lines. Set the SYSCFG.DPRPU bit and DRPD bit to specify pull-up or pull-down for each line.
In addition, in function control operation, if the DPRPU bit is set to 0 while communicating with the USB host, the pull-up resistor for the USB data line becomes invalid, and the USB host can be notified of a device disconnection.

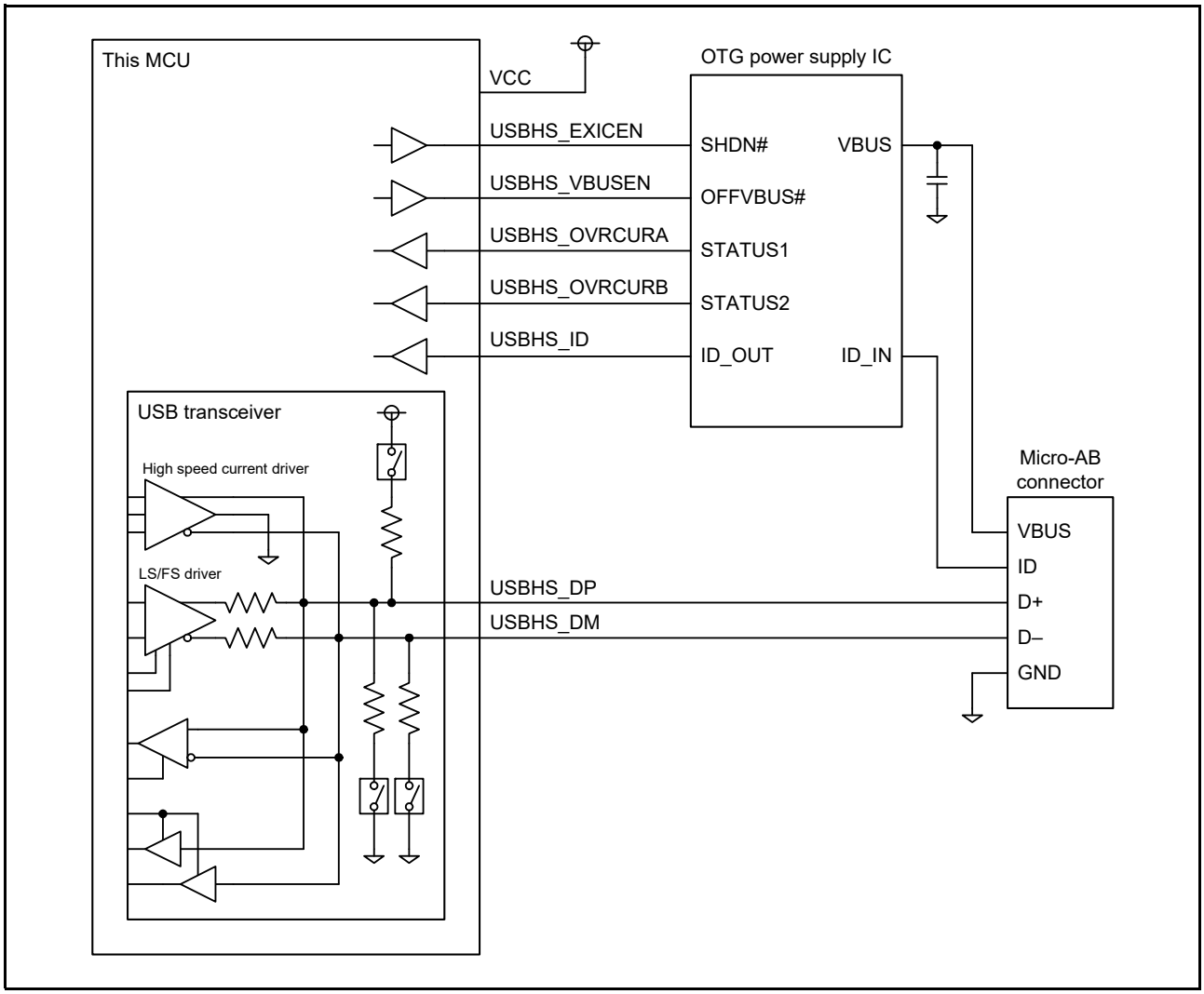


Figure 41.23 USB Connector's OTG Connection in Self Powered Mode

Figure 41.24 shows the USB Connector's Function Connection in Self Powered Mode.

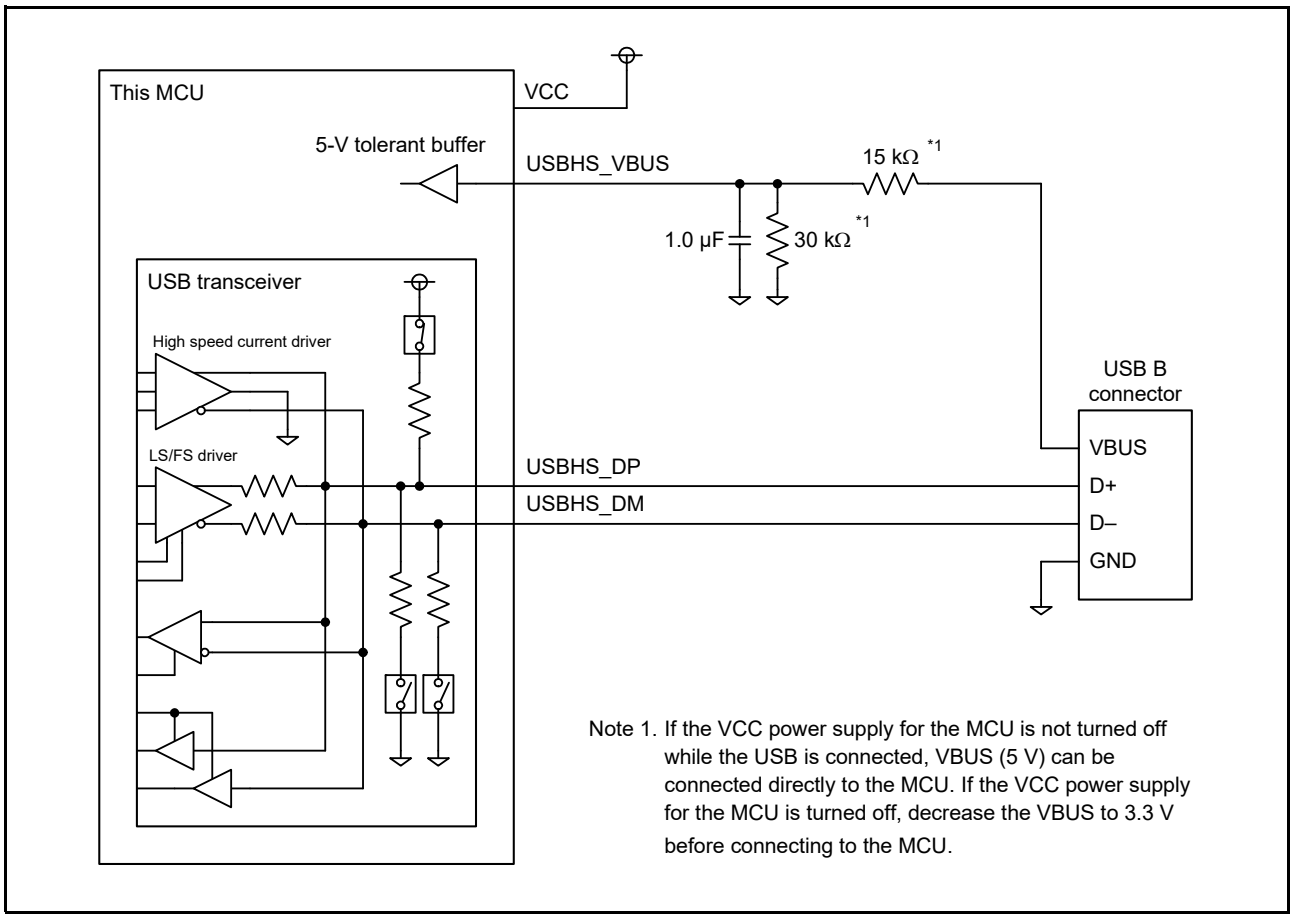


Figure 41.24 USB Connector's Function Connection in Self Powered Mode

Figure 41.25 shows the USB Connector's Function Connection in Bus Powered Mode.

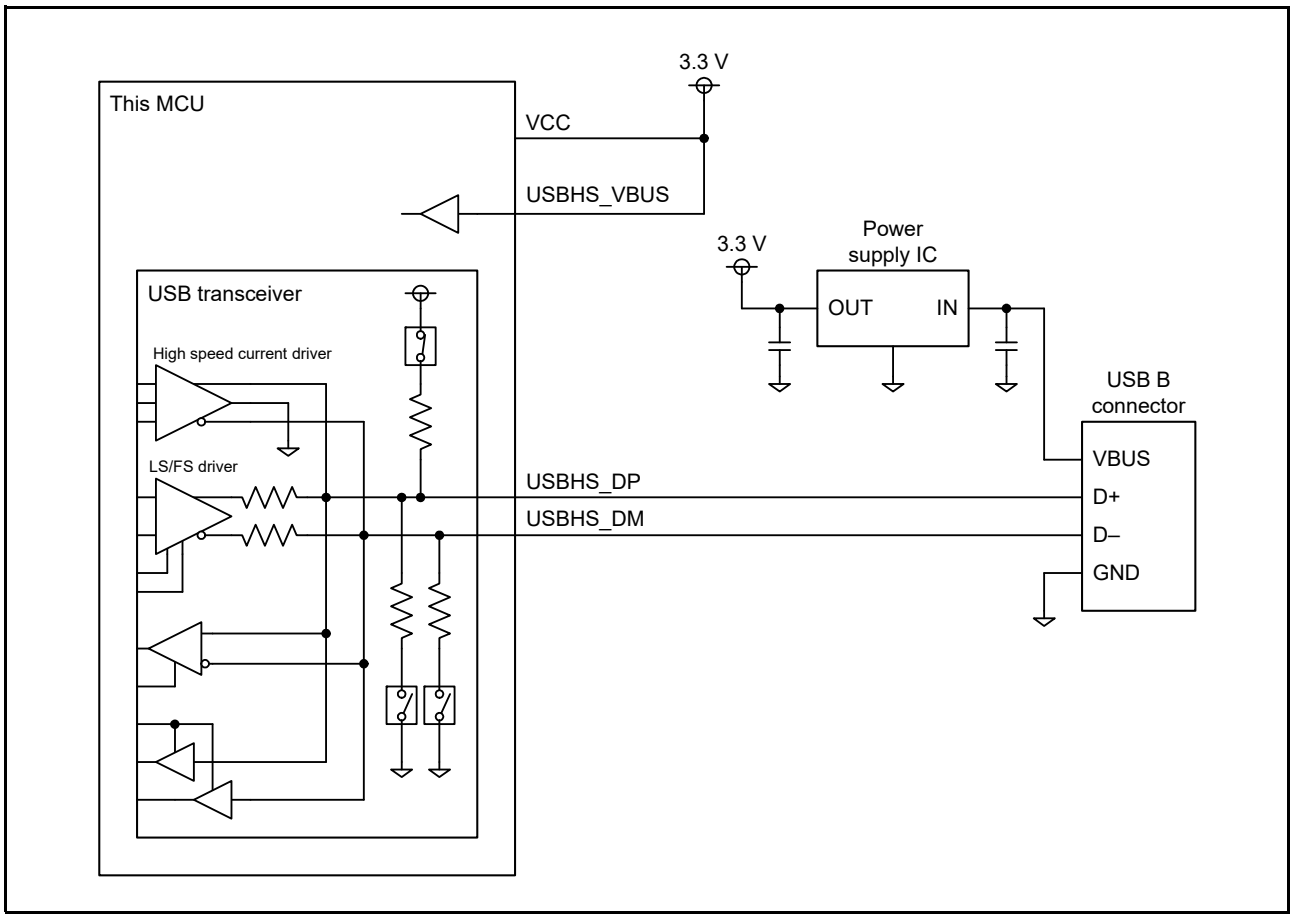


Figure 41.25 USB Connector's Function Connection in Bus Powered Mode

Figure 41.26 shows the USB Connector Host Connection.

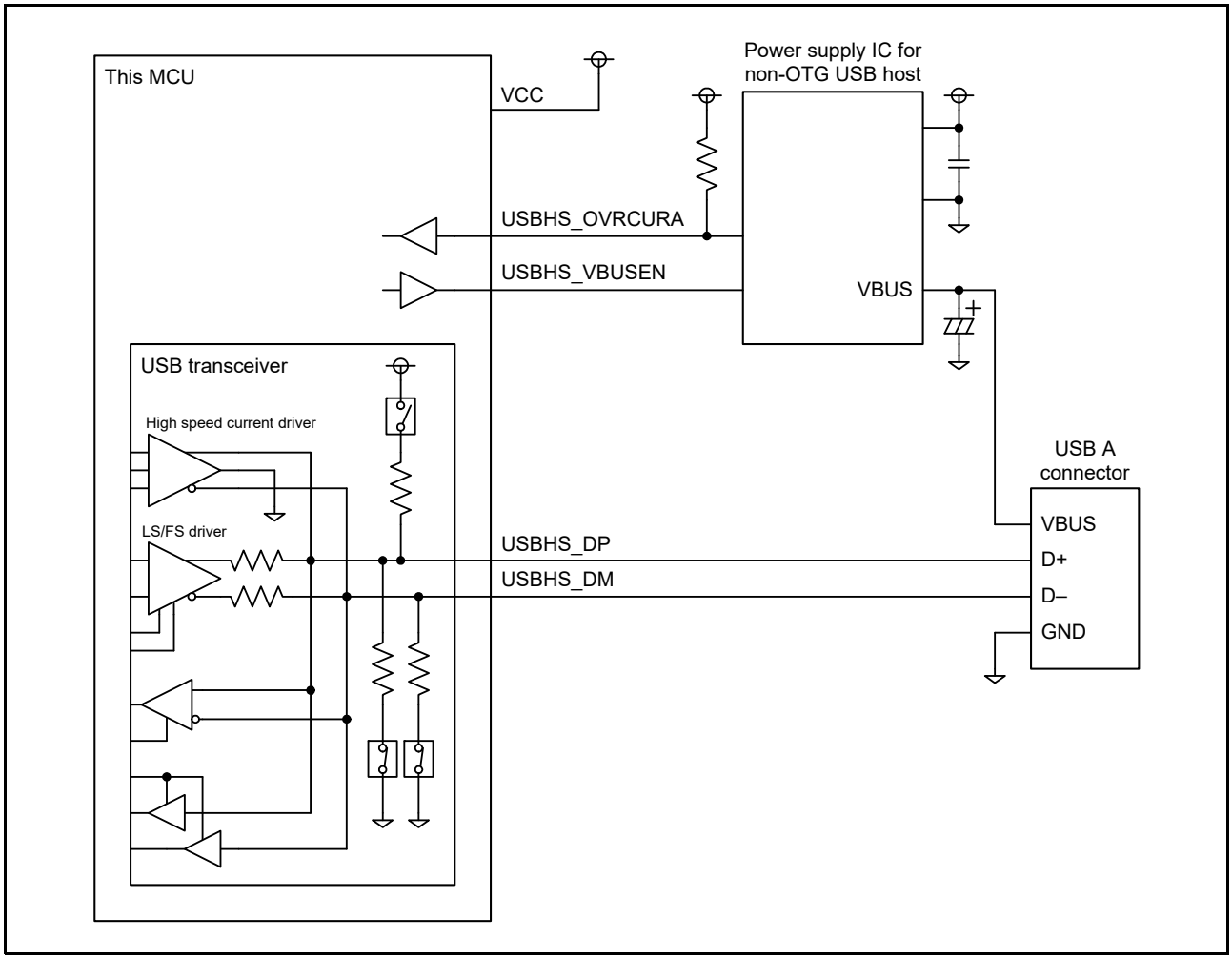


Figure 41.26 USB Connector Host Connection

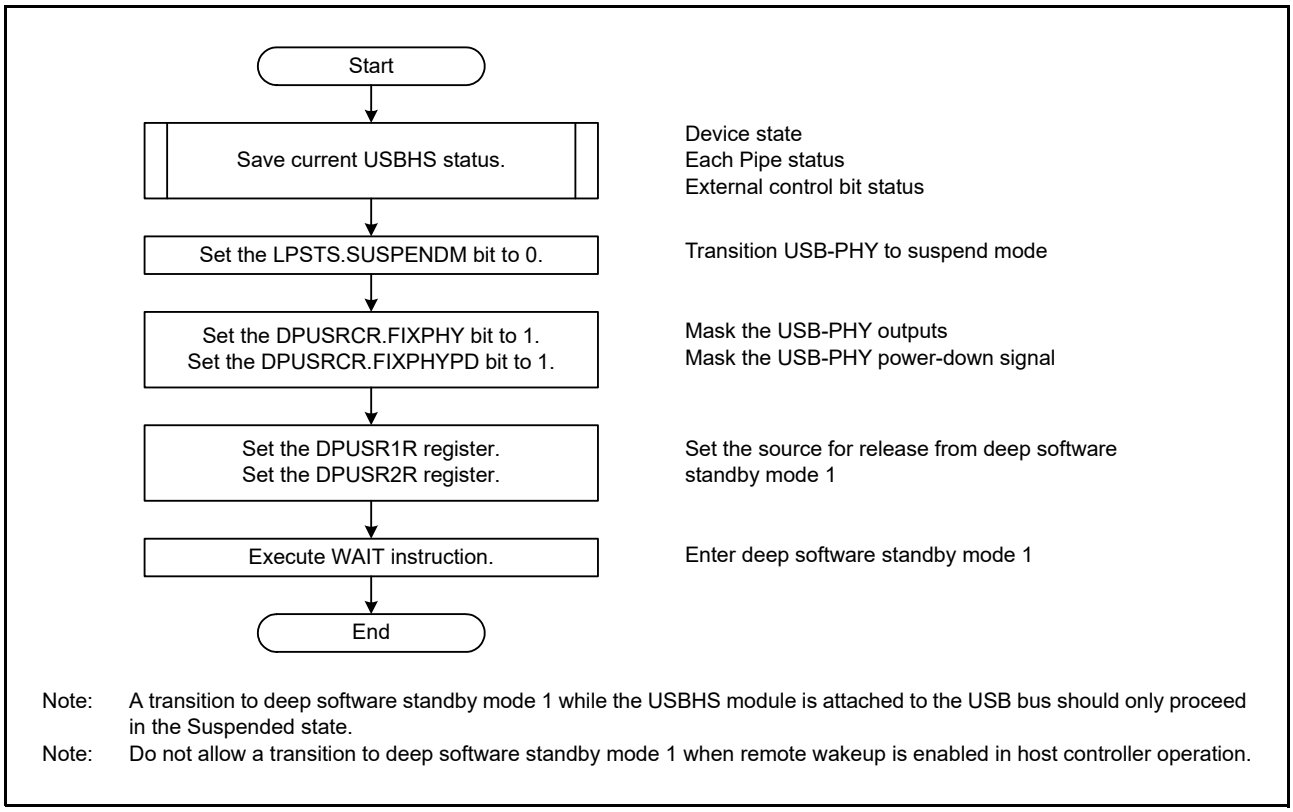
41.3.18 Release from Deep Software Standby Mode 1 in Response to the USBHS Resume Interrupt

The USBHS has blocks to which power is supplied in deep software standby mode 1. The USBHS resume interrupt generated by these blocks can be used as the trigger for release from deep software standby mode 1. Table 41.33 lists the sources for the USBHS resume interrupt.

Table 41.33 Sources of the USBHS Resume Interrupt

Operating Mode	Source	Interrupt to Be Used	Pins to Be Used
Host	Overcurrent detection	OVRCURA interrupt	USBHS_OVRCURA
OTG	Overcurrent detection	OVRCURA interrupt, OVRCURB interrupt	USBHS_OVRCURA, USBHS_OVRCURB
Function	Attach/Detach	VBUS interrupt	USBHS_VBUS
Host/OTG	Attach/Detach	D+ interrupt, D– interrupt	USBHS_DP, USBHS_DM
Function/OTG	Resume	D+ interrupt	USBHS_DP

Figure 41.27 shows a flow of settings for transition to deep software standby mode 1. Figure 41.28 shows a flow of settings after release from deep software standby mode 1 in host controller operation. Figure 41.29 shows a flow of settings after release from deep software standby mode 1 in function controller operation.



Note: A transition to deep software standby mode 1 while the USBHS module is attached to the USB bus should only proceed in the Suspended state.
Note: Do not allow a transition to deep software standby mode 1 when remote wakeup is enabled in host controller operation.

Figure 41.27 Flow for Transition to Deep Software Standby Mode 1

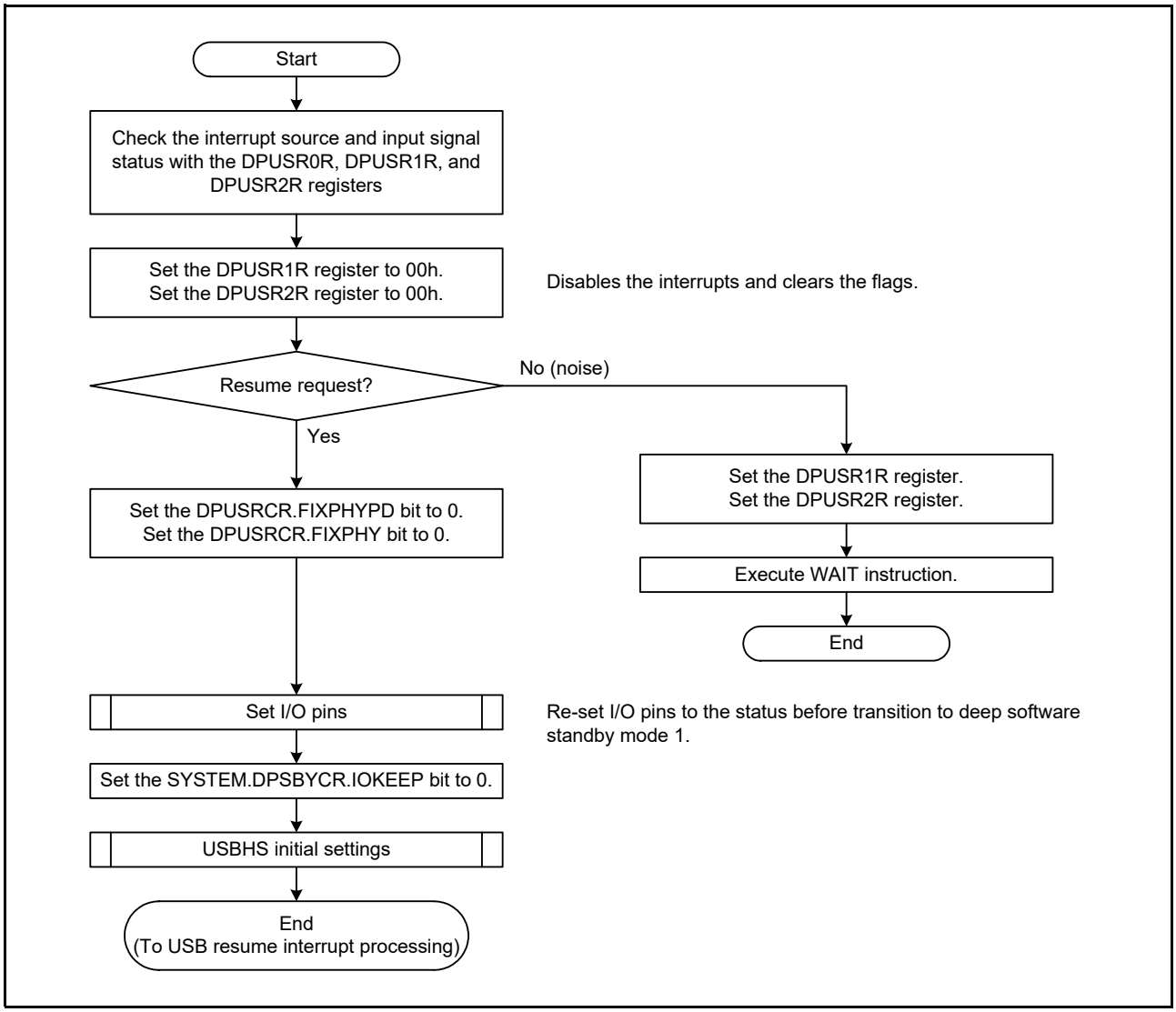


Figure 41.28 Setting Flow after Release from Deep Software Standby Mode 1 in Host Controller Operation

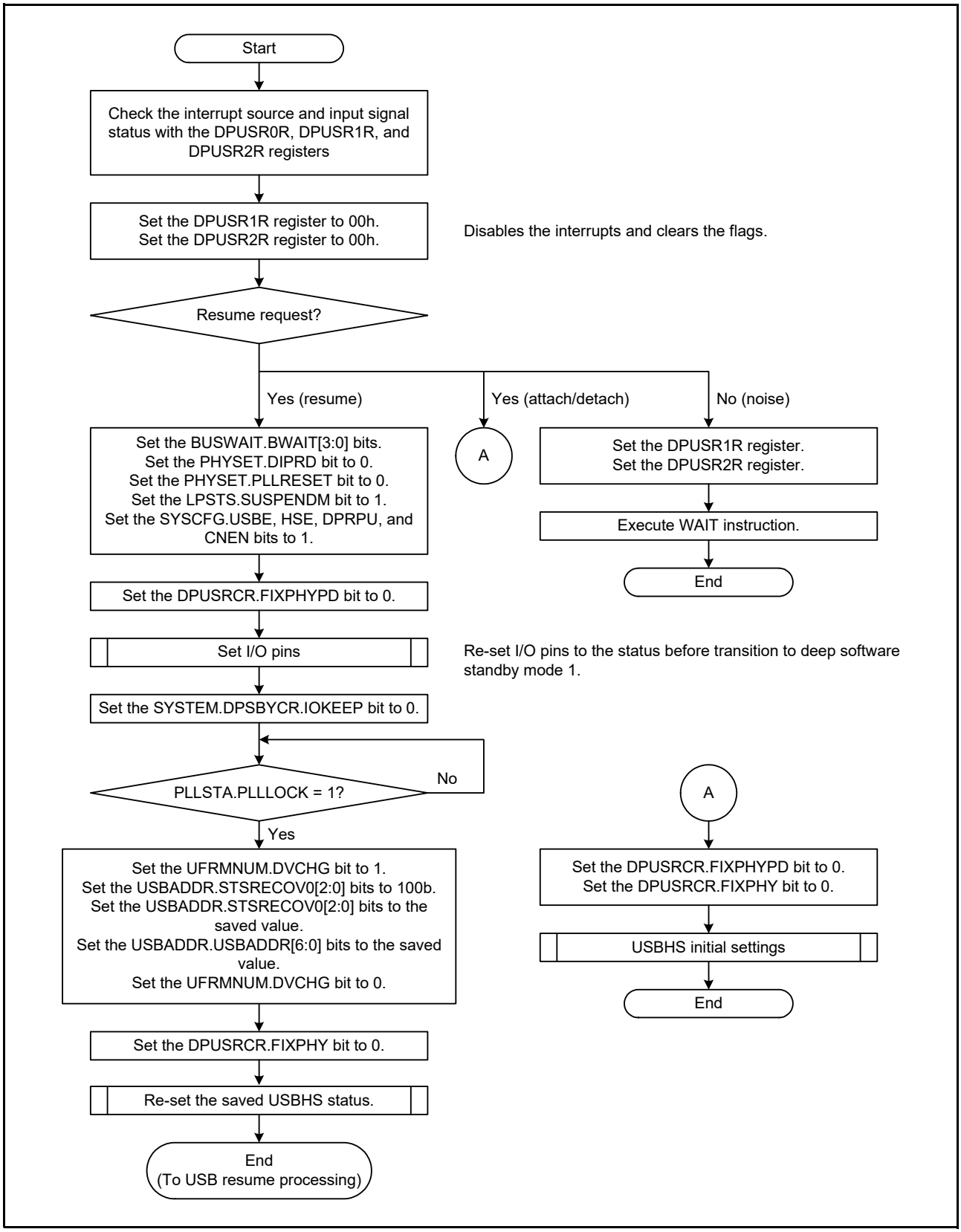


Figure 41.29 Setting Flow after Release from Deep Software Standby Mode 1 in Function Controller Operation

41.4 Notes on Using the USBHS

41.4.1 Setting the USBHS Module-Stop Function

The USBHS operation can be disabled or enabled by setting the module stop control register B (MSTPCRB). The USBHS is stopped after a reset. Registers in the USBHS can be accessed by releasing the module stop state. Refer to section 11, Low Power Consumption for details.
When operating the PHY circuit after releasing the module stop state, configure settings for items such as the frequency of the PLL clock source (USBMCLK), and then set the PHYSET.DIRPD bit to 0.

41.4.2 Setting to Transition to Deep Software Standby Mode

Before transitioning to deep software standby mode, set the DVSTCTR0.VBUSEN bit to 0.

42. Serial Communications Interface (RSCI)

In this section, “PCLK” is used to refer to PCLKA.

42.1 Overview

RSCI can handle both asynchronous and clock synchronous serial communications. RSCI has 16-stage FIFO buffers in transmission/reception blocks, and it can select the FIFO composition, and it can transmit/receive efficiently, and it can also communicate continuously.

Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the RSCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for identification cards).

The RSCI is also supports serial communication using Manchester code (Manchester mode), simple SPI interfaces, simple I2C-bus interfaces (single master), and extended serial communication.

In addition, asynchronous mode has a support function for generating AMI waveform of negative logic coding with 50% duty cycle used in home bus system (HBS) communications.

Table 42.1 lists the RSCI specifications.

Table 42.1 RSCI Specifications (1/3)

Item	Description
Serial communication modes	• Asynchronous • Manchester • Clock synchronous • Smart card interface • Simple I2C-bus • Simple SPI bus • Extended serial
Operating clock (TCLK)	Selectable from peripheral module clock (PCLKA) or SCI clock (SCICLK)
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB first or MSB first transfer
I/O signal level inverting function	Input signal and output signal can be inverted independently.
RXD input path selection	Encoded serial data can be input to RXD2 and RXD3 after having been decoded through GPTW10 and GPTW12, respectively.
TXD output path selection	Serial data output from TXD2 and TXD3 can be encoded through GPTW11 and GPTW13, respectively.
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, and receive data match Break Field detection/transmission, Bus collision detection, Active edge detection Completion of generation of a start condition, restart condition, or stop condition
RS-485 driver control function	Output DE signal to enable external transceiver transmit mode
Loopback function	Self-diagnosis of communication function is possible by connecting TXD and RXD inside the RSCI
Synchronizer bypass function	Synchronizer is bypassed when the peripheral module clock (PCLKA) is selected for the operating clock (TCLK).
Low power consumption function	Module stop state can be set for each channel.
Protect Zone filter	Security attribute can be set for each channel.*1

Table 42.1 RSCI Specifications (2/3)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 16-stage FIFO
	Data match detection	The interrupt request can issue by detecting the match between receive data and comparison data.
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the SSR register.
	Clock source	An internal, external, or TMR clock can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	HBS support mode	Transmission/reception using inverted RZI (Return to Zero, Inverted) code is possible.
Manchester mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Receive error detection	Parity, overrun, framing, Manchester code errors, preface, start bit, and receive Sync
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission.
	Clock source	Only internal clock can be used. (The setting of external clock is prohibited because it is not the object of operation guarantee.)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	Manchester encoding/ decoding function	Function to perform Manchester encoding/decoding of transmission/reception data and communicate using Manchester code
	Preface setting/detection function	Function to detect the beginning of a frame from the preface pattern. Preface pattern can be selected from 4 types. The length can also be changed from 0 to 15 bits.
	Start Bit setting/detection function	The Start Bit length can be set to 1 bit or 3 bits. In the case of 3-bit length, it is possible to judge the type of subsequent data with two types of patterns.
	Reception retiming function	Function to perform timing correction for each bit center edge by using Manchester code having edge at bit center
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

Table 42.1 RSCI Specifications (3/3)

Item		Description
Extended serial mode	Start Frame Transmission	Break Field transmission possible, Break Field transmission complete interrupt output possible Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	Break Field detectable, Break Field detected interrupt output possible Control Field 0/1 data comparison function Control Field 1 can set two types of comparison data of primary and secondary Priority interrupt bit can be set in Control Field 1 Bit rate measurement function
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 42.2.7, Control Register 2 (SCR2) o the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 16-stage FIFO
Simple SPI bus (4-wire serial bus) mode	Data length	8 bits
	Detection of errors	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	SS input pin function	When the SSn# pin is high level, the output pin can be set to high impedance
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
	Transmission/Reception	Selectable either 1 stage register or 16-stage FIFO
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output
		Receive data match event output
		Receive data unmatch event output
		Active edge detection event output

Note 1. When each channel is set to the secure attribute, registers in the channel are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each channel is set to the non-secure attribute, registers in the channel are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 42.1 shows the RSCI Block Diagram.

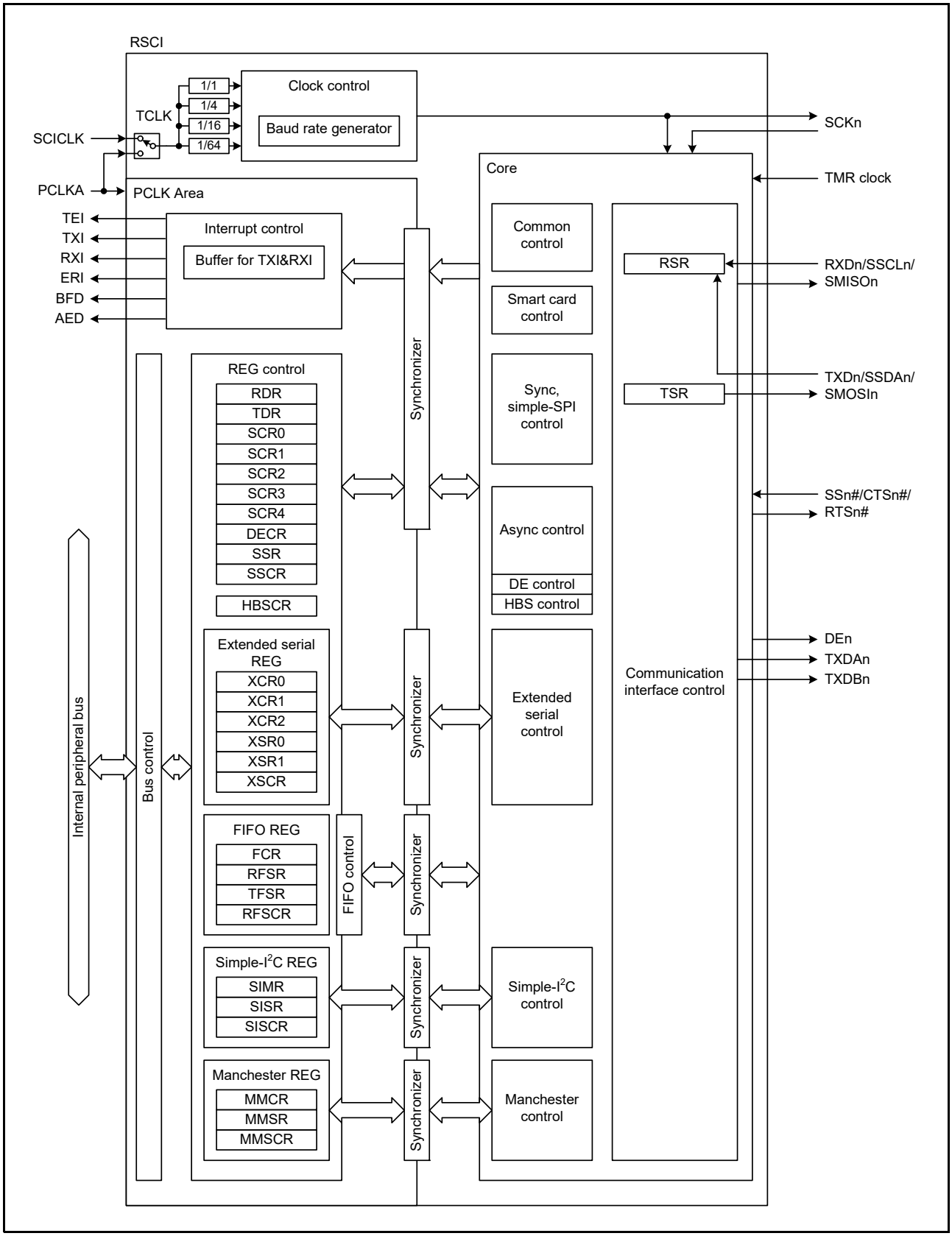


Figure 42.1 RSCI Block Diagram (n = 000 to 012)

Table 42.2 to Table 42.5 list RSCI's input/output pins.

Table 42.2 RSCI Input/Output Pin (Asynchronous Mode/Clock Synchronous Mode/Manchester Mode/ Extended Serial Mode)

Channel	Pin Name	I/O	Function
RSCI0	SCK000	I/O	RSCI0 clock input/output
	RXD000	Input	RSCI0 receive data input
	TXD000	Output	RSCI0 transmit data output
	RTS000#	Output	RSCI0 request-to-send signal output
	CTS000#	Input	RSCI0 transmission start control input
	DE000	Output	RSCI0 RS-485 driver control output
RSCI1	SCK001	I/O	RSCI1 clock input/output
	RXD001	Input	RSCI1 receive data input
	TXD001	Output	RSCI1 transmit data output
	RTS001#	Output	RSCI1 request-to-send signal output
	CTS001#	Input	RSCI1 transmission start control input
	DE001	Output	RSCI1 RS-485 driver control output
RSCI2	SCK002	I/O	RSCI2 clock input/output
	RXD002	Input	RSCI2 receive data input
	TXD002	Output	RSCI2 transmit data output
	RTS002#	Output	RSCI2 request-to-send signal output
	CTS002#	Input	RSCI2 transmission start control input
	DE002	Output	RSCI2 RS-485 driver control output
RSCI3	SCK003	I/O	RSCI3 clock input/output
	RXD003	Input	RSCI3 receive data input
	TXD003	Output	RSCI3 transmit data output
	RTS003#	Output	RSCI3 request-to-send signal output
	CTS003#	Input	RSCI3 transmission start control input
	DE003	Output	RSCI3 RS-485 driver control output
RSCI4	SCK004	I/O	RSCI4 clock input/output
	RXD004	Input	RSCI4 receive data input
	TXD004	Output	RSCI4 transmit data output
	RTS004#	Output	RSCI4 request-to-send signal output
	CTS004#	Input	RSCI4 transmission start control input
	DE004	Output	RSCI4 RS-485 driver control output
RSCI5	SCK005	I/O	RSCI5 clock input/output
	RXD005	Input	RSCI5 receive data input
	TXD005	Output	RSCI5 transmit data output
	RTS005#	Output	RSCI5 request-to-send signal output
	CTS005#	Input	RSCI5 transmission start control input
	DE005	Output	RSCI5 RS-485 driver control output
RSCI6	SCK006	I/O	RSCI6 clock input/output
	RXD006	Input	RSCI6 receive data input
	TXD006	Output	RSCI6 transmit data output
	RTS006#	Output	RSCI6 request-to-send signal output
	CTS006#	Input	RSCI6 transmission start control input
	DE006	Output	RSCI6 RS-485 driver control output

Table 42.2 RSCI Input/Output Pin (Asynchronous Mode/Clock Synchronous Mode/Manchester Mode/ Extended Serial Mode)

Channel	Pin Name	I/O	Function
RSCI7	SCK007	I/O	RSCI7 clock input/output
	RXD007	Input	RSCI7 receive data input
	TXD007	Output	RSCI7 transmit data output
	RTS007#	Output	RSCI7 request-to-send signal output
	CTS007#	Input	RSCI7 transmission start control input
	DE007	Output	RSCI7 RS-485 driver control output
RSCI8	SCK008	I/O	RSCI8 clock input/output
	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	RTS008#	Output	RSCI8 request-to-send signal output
	CTS008#	Input	RSCI8 transmission start control input
	DE008	Output	RSCI8 RS-485 driver control output
RSCI9	SCK009	I/O	RSCI9 clock input/output
	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	RTS009#	Output	RSCI9 request-to-send signal output
	CTS009#	Input	RSCI9 transmission start control input
	DE009	Output	RSCI9 RS-485 driver control output
RSCI10	SCK010	I/O	RSCI10 clock input/output
	RXD010	Input	RSCI10 receive data input
	TXD010	Output	RSCI10 transmit data output
	RTS010#	Output	RSCI10 request-to-send signal output
	CTS010#	Input	RSCI10 transmission start control input
	DE010	Output	RSCI10 RS-485 driver control output
RSCI11	SCK011	I/O	RSCI11 clock input/output
	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	RTS011#	Output	RSCI11 request-to-send signal output
	CTS011#	Input	RSCI11 transmission start control input
	DE011	Output	RSCI11 RS-485 driver control output
RSCI12	SCK012	I/O	RSCI12 clock input/output
	RXD012	Input	RSCI12 receive data input
	TXD012	Output	RSCI12 transmit data output
	RTS012#	Output	RSCI12 request-to-send signal output
	CTS012#	Input	RSCI12 transmission start control input
	DE012	Output	RSCI12 RS-485 driver control output

Table 42.3 RSCI Input/Output Pin (Simple I^2C Mode)

Channel	Pin Name	I/O	Function
RSCI0	SSCL000	I/O	RSCI0 I^2C clock input/output
	SSDA000	I/O	RSCI0 I^2C data input/output
RSCI1	SSCL001	I/O	RSCI1 I^2C clock input/output
	SSDA001	I/O	RSCI1 I^2C data input/output
RSCI2	SSCL002	I/O	RSCI2 I^2C clock input/output
	SSDA002	I/O	RSCI2 I^2C data input/output

Table 42.3 RSCI Input/Output Pin (Simple I²C Mode)

Channel	Pin Name	I/O	Function
RSCI3	SSCL003	I/O	RSCI3 I²C clock input/output
	SSDA003	I/O	RSCI3 I²C data input/output
RSCI4	SSCL004	I/O	RSCI4 I²C clock input/output
	SSDA004	I/O	RSCI4 I²C data input/output
RSCI5	SSCL005	I/O	RSCI5 I²C clock input/output
	SSDA005	I/O	RSCI5 I²C data input/output
RSCI6	SSCL006	I/O	RSCI6 I²C clock input/output
	SSDA006	I/O	RSCI6 I²C data input/output
RSCI7	SSCL007	I/O	RSCI7 I²C clock input/output
	SSDA007	I/O	RSCI7 I²C data input/output
RSCI8	SSCL008	I/O	RSCI8 I²C clock input/output
	SSDA008	I/O	RSCI8 I²C data input/output
RSCI9	SSCL009	I/O	RSCI9 I²C clock input/output
	SSDA009	I/O	RSCI9 I²C data input/output
RSCI10	SSCL010	I/O	RSCI10 I²C clock input/output
	SSDA010	I/O	RSCI10 I²C data input/output
RSCI11	SSCL011	I/O	RSCI11 I²C clock input/output
	SSDA011	I/O	RSCI11 I²C data input/output
RSCI12	SSCL012	I/O	RSCI12 I²C clock input/output
	SSDA012	I/O	RSCI12 I²C data input/output

Table 42.4 RSCI Input/Output Pin (Simple SPI Mode)

Channel	Pin Name	I/O	Function
RSCI0	SCK000	I/O	RSCI0 clock input/output
	SMISO000	I/O	RSCI0 slave transmit data input/output
	SMOSI000	I/O	RSCI0 master transmit data input/output
	SS000#	Input	RSCI0 slave select input
RSCI1	SCK001	I/O	RSCI1 clock input/output
	SMISO001	I/O	RSCI1 slave transmit data input/output
	SMOSI001	I/O	RSCI1 master transmit data input/output
	SS001#	Input	RSCI1 slave select input
RSCI2	SCK002	I/O	RSCI2 clock input/output
	SMISO002	I/O	RSCI2 slave transmit data input/output
	SMOSI002	I/O	RSCI2 master transmit data input/output
	SS002#	Input	RSCI2 slave select input
RSCI3	SCK003	I/O	RSCI3 clock input/output
	SMISO003	I/O	RSCI3 slave transmit data input/output
	SMOSI003	I/O	RSCI3 master transmit data input/output
	SS003#	Input	RSCI3 slave select input
RSCI4	SCK004	I/O	RSCI4 clock input/output
	SMISO004	I/O	RSCI4 slave transmit data input/output
	SMOSI004	I/O	RSCI4 master transmit data input/output
	SS004#	Input	RSCI4 slave select input

Table 42.4 RSCI Input/Output Pin (Simple SPI Mode)

Channel	Pin Name	I/O	Function
RSCI5	SCK005	I/O	RSCI5 clock input/output
	SMISO005	I/O	RSCI5 slave transmit data input/output
	SMOSI005	I/O	RSCI5 master transmit data input/output
	SS005#	Input	RSCI5 slave select input
RSCI6	SCK006	I/O	RSCI6 clock input/output
	SMISO006	I/O	RSCI6 slave transmit data input/output
	SMOSI006	I/O	RSCI6 master transmit data input/output
	SS006#	Input	RSCI6 slave select input
RSCI7	SCK007	I/O	RSCI7 clock input/output
	SMISO007	I/O	RSCI7 slave transmit data input/output
	SMOSI007	I/O	RSCI7 master transmit data input/output
	SS007#	Input	RSCI7 slave select input
RSCI8	SCK008	I/O	RSCI8 clock input/output
	SMISO008	I/O	RSCI8 slave transmit data input/output
	SMOSI008	I/O	RSCI8 master transmit data input/output
	SS008#	Input	RSCI8 slave select input
RSCI9	SCK009	I/O	RSCI9 clock input/output
	SMISO009	I/O	RSCI9 slave transmit data input/output
	SMOSI009	I/O	RSCI9 master transmit data input/output
	SS009#	Input	RSCI9 slave select input
RSCI10	SCK010	I/O	RSCI10 clock input/output
	SMISO010	I/O	RSCI10 slave transmit data input/output
	SMOSI010	I/O	RSCI10 master transmit data input/output
	SS010#	Input	RSCI10 slave select input
RSCI11	SCK011	I/O	RSCI11 clock input/output
	SMISO011	I/O	RSCI11 slave transmit data input/output
	SMOSI011	I/O	RSCI11 master transmit data input/output
	SS011#	Input	RSCI11 slave select input
RSCI12	SCK012	I/O	RSCI12 clock input/output
	SMISO012	I/O	RSCI12 slave transmit data input/output
	SMOSI012	I/O	RSCI12 master transmit data input/output
	SS012#	Input	RSCI12 slave select input

Table 42.5 RSCI Input/Output Pin (HBS Support Mode)

Channel	Pin Name	I/O	Function
RSCI0	RXD000	Input	RSCI0 receive data input
	TXD000	Output	RSCI0 transmit data output
	TXDA000/TXDB000	Output	RSCI0 transmit data output (in alternate output)
RSCI1	RXD001	Input	RSCI1 receive data input
	TXD001	Output	RSCI1 transmit data output
	TXDA001/TXDB001	Output	RSCI1 transmit data output (in alternate output)
RSCI2	RXD002	Input	RSCI2 receive data input
	TXD002	Output	RSCI2 transmit data output
	TXDA002/TXDB002	Output	RSCI2 transmit data output (in alternate output)

Table 42.5 RSCI Input/Output Pin (HBS Support Mode)

Channel	Pin Name	I/O	Function
RSCI3	RXD003	Input	RSCI3 receive data input
	TXD003	Output	RSCI3 transmit data output
	TXDA003/TXDB003	Output	RSCI3 transmit data output (in alternate output)
RSCI4	RXD004	Input	RSCI4 receive data input
	TXD004	Output	RSCI4 transmit data output
	TXDA004/TXDB004	Output	RSCI4 transmit data output (in alternate output)
RSCI5	RXD005	Input	RSCI5 receive data input
	TXD005	Output	RSCI5 transmit data output
	TXDA005/TXDB005	Output	RSCI5 transmit data output (in alternate output)
RSCI0	RXD006	Input	RSCI6 receive data input
	TXD006	Output	RSCI6 transmit data output
	TXDA006/TXDB006	Output	RSCI6 transmit data output (in alternate output)
RSCI7	RXD007	Input	RSCI7 receive data input
	TXD007	Output	RSCI7 transmit data output
	TXDA007/TXDB007	Output	RSCI7 transmit data output (in alternate output)
RSCI8	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	TXDA008/TXDB008	Output	RSCI8 transmit data output (in alternate output)
RSCI9	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	TXDA009/TXDB009	Output	RSCI9 transmit data output (in alternate output)
RSCI10	RXD010	Input	RSCI10 receive data input
	TXD010	Output	RSCI10 transmit data output
	TXDA010/TXDB010	Output	RSCI10 transmit data output (in alternate output)
RSCI11	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	TXDA011/TXDB011	Output	RSCI11 transmit data output (in alternate output)
RSCI12	RXD012	Input	RSCI12 receive data input
	TXD012	Output	RSCI12 transmit data output
	TXDA012/TXDB012	Output	RSCI12 transmit data output (in alternate output)

42.2 Register Descriptions

This chapter describes the RSCI registers, their functional specifications, and their operating specifications.

42.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

42.2.2 Receive Data Register (RDR)

Address(es): RSCI0.RDR 8745 8000h, RSCI1.RDR 8745 8100h, RSCI2.RDR 8745 8200h, RSCI3.RDR 8745 8300h, RSCI4.RDR 8745 8400h, RSCI5.RDR 8745 8500h, RSCI6.RDR 8745 8600h, RSCI7.RDR 8745 8700h, RSCI8.RDR 8745 8800h, RSCI9.RDR 8745 8900h, RSCI10.RDR 8745 8A00h, RSCI11.RDR 8745 8B00h, RSCI12.RDR 8745 8C00h,
RSCI0_NS.RDR 0745 8000h, RSCI1_NS.RDR 0745 8100h, RSCI2_NS.RDR 0745 8200h, RSCI3_NS.RDR 0745 8300h, RSCI4_NS.RDR 0745 8400h, RSCI5_NS.RDR 0745 8500h, RSCI6_NS.RDR 0745 8600h, RSCI7_NS.RDR 0745 8700h, RSCI8_NS.RDR 0745 8800h, RSCI9_NS.RDR 0745 8900h, RSCI10_NS.RDR 0745 8A00h, RSCI11_NS.RDR 0745 8B00h, RSCI12_NS.RDR 0745 8C00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FER	PER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	RDAT[8:0]	Receive Data	RDAT[8:0] bits are a 9-bit field for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
b9	MPB	Multi-Processor Bit Monitor Flag	0: Data transmission cycles 1: ID transmission cycles	R
b10	DR	Receive Data Ready Flag	RFSR.DR flag can be read.	R
b11	PER	Parity Error Flag	(Valid only in asynchronous mode) 0: There is no parity error in the data read from the receive FIFO 1: There is parity error in the data read from the receive FIFO	R
b12	FER	Framing Error Flag	(Valid only in asynchronous mode) 0: There is no framing error in the data read from the receive FIFO 1: There is framing error in the data read from the receive FIFO	R
b23 to b13	—	Reserved	These bits are read as 0.	R
b24	ORER	Overrun Error Flag	SSR.ORER flag can be read.	R
b26, b25	—	Reserved	These bits are read as 0.	R
b27	APER	Aggregate Parity Error Flag	SSR.APER flag can be read.	R
b28	AFER	Aggregate Framing Error Flag	SSR.AFER flag can be read.	R
b31 to b29	—	Reserved	These bits are read as 0.	R

In FIFO mode (SCR3.FM bit = 1), this register is 16-stage FIFO buffer configuration.

RDAT[8:0] Bits (Receive Data)

After one frame of data is received, the received data is transferred from the RSR register to this registers, thus allowing the RSR register to receive the next data.
The RSR and RDR registers have a double-buffered construction to enable continuous reception.
In non-FIFO mode, read the RDR register only once when a receive data full interrupt (RXI) request is issued. Without reading received data from RDR register, if the next one frame is received, an overrun error occurs.
In FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive FIFO (RDR register), the value is undefined. When the receive FIFO (RDR register) are full of received data, subsequent serial receive data is lost.
The CPU cannot write to RDR register.
0 is stored in the bit position which isn't received (RDAT[8] or RDAT[7]) at the time of 7bit or 8bit communication of asynchronous and Manchester mode.

MPB Flag (Multi-Processor Bit Monitor Flag)

In asynchronous mode and Manchester mode, during multi-processor communication (SCR3.MP bit = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0] bits) can be read.

PER Flag (Parity Error Flag)

Indicates whether the data read from the receive FIFO has a parity error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

FER Flag (Framing Error Flag)

Indicates whether the data read from receive FIFO has a framing error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

42.2.3 Transmit Data Register (TDR)

Address(es): RSCI0.TDR 8745 8004h, RSCI1.TDR 8745 8104h, RSCI2.TDR 8745 8204h, RSCI3.TDR 8745 8304h, RSCI4.TDR 8745 8404h, RSCI5.TDR 8745 8504h, RSCI6.TDR 8745 8604h, RSCI7.TDR 8745 8704h, RSCI8.TDR 8745 8804h, RSCI9.TDR 8745 8904h, RSCI10.TDR 8745 8A04h, RSCI11.TDR 8745 8B04h, RSCI12.TDR 8745 8C04h,
RSCI0_NS.TDR 0745 8004h, RSCI1_NS.TDR 0745 8104h, RSCI2_NS.TDR 0745 8204h, RSCI3_NS.TDR 0745 8304h, RSCI4_NS.TDR 0745 8404h, RSCI5_NS.TDR 0745 8504h, RSCI6_NS.TDR 0745 8604h, RSCI7_NS.TDR 0745 8704h, RSCI8_NS.TDR 0745 8804h, RSCI9_NS.TDR 0745 8904h, RSCI10_NS.TDR 0745 8A04h, RSCI11_NS.TDR 0745 8B04h, RSCI12_NS.TDR 0745 8C04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	TDAT[8:0]	Transmit Data	TDAT[8:0] bits are a 9-bit field for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When writing the TDR register in 8-bit units, write first to the TDR.LH and then TDR.LL.	R/W
b9	MPBT	Transmit Multi-Processor	Value of the multi-processor bit in the transmission frame. This bit is use in asynchronous and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
b11, b10	—	Reserved	These bits are read as 1. The write value should be 1.	R
b12	SYNC	Sync Pulse Select	It is valid when MMCR.SBLEN bit = 1 and MMCR.SYNCE bit = 1 in Manchester mode. When writing to this bit when not used, write the initial value. 0: The Start Bit is transmitted as DATA Sync. 1: The Start Bit is transmitted as COMMAND Sync.	R/W
b31 to b13	—	Reserved	These bits are read as 1. The write value should be 1.	R

In FIFO mode (SCR3.FM bit = 1), this register is 16-stage FIFO buffer configuration.

TDAT[8:0] Bits (Transmit Data)

The TDAT[8:0] bits are a 9-bit field for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR register is transferred to TSR register, and transmitting is started.

The TSR and TDR registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in TDR register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

In non-FIFO mode, the TDR register is always readable and writable by the CPU. When a transmit data empty interrupt (TXI) request is issued and SCR0.TE bit is 1, write transmit data to the TDR register only once.

In FIFO mode, continuous serial transmission is executed until there is no transmit data left in the transmit FIFO (TDR register). When transmit FIFO is full of transmit data (16 frames), no more data can be written. If writing of new data is attempted, the data is ignored.

When writing the TDR register in 8-bit units, write first to the TDR.LH and then the TDR.LL.

MPBT Bit (Transmit Multi-Processor)

Selects the multi-processor bit of transmit frame.

SYNC Bit (Sync Pulse Select)

This bit is valid when the MMCR.SYNCE and MMCR.SBLEN bits are set to 1 in Manchester mode (SCR3.MOD[2:0] bits = 101b).

The Sync type of start bit area in the transmission frame can be set to Data Sync or Command Sync.

42.2.4 Transmit Shift Register (TSR)

TSR register is a shift register that transmits serial data. TSR register cannot be directly accessed by the CPU.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR register to TSR register, and then sends the data to the TXDn pin.

42.2.5 Control Register 0 (SCR0)

Address(es): RSCI0.SCR0 8745 8008h, RSCI1.SCR0 8745 8108h, RSCI2.SCR0 8745 8208h, RSCI3.SCR0 8745 8308h, RSCI4.SCR0 8745 8408h, RSCI5.SCR0 8745 8508h, RSCI6.SCR0 8745 8608h, RSCI7.SCR0 8745 8708h, RSCI8.SCR0 8745 8808h, RSCI9.SCR0 8745 8908h, RSCI10.SCR0 8745 8A08h, RSCI11.SCR0 8745 8B08h, RSCI12.SCR0 8745 8C08h,
RSCI0_NS.SCR0 0745 8008h, RSCI1_NS.SCR0 0745 8108h, RSCI2_NS.SCR0 0745 8208h, RSCI3_NS.SCR0 0745 8308h, RSCI4_NS.SCR0 0745 8408h, RSCI5_NS.SCR0 0745 8508h, RSCI6_NS.SCR0 0745 8608h, RSCI7_NS.SCR0 0745 8708h, RSCI8_NS.SCR0 0745 8808h, RSCI9_NS.SCR0 0745 8908h, RSCI10_NS.SCR0 0745 8A08h, RSCI11_NS.SCR0 0745 8B08h, RSCI12_NS.SCR0 0745 8C08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W *1, *2
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode and Manchester mode when SCR3.MP is 1.) This bit should set 0 in smart card interface mode. 0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.	R/W *3
b9	DCME	Data Compare Match Enable	(Valid only in asynchronous mode) 0: Data match detection function is disabled 1: Data match detection function is enabled	R/W *3
b10	IDSEL	ID Frame Select	(Valid only in asynchronous mode with multi-processor) 0: It's always compared data in spite of the value of the multi-processor bit. 1: It's compared data when the multi-processor bit is 1 (ID frame) only.	R/W *4
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TIE	Transmit Interrupt Enable	0: TXI interrupt request is disabled 1: TXI interrupt request is enabled	R/W
b21	TEIE	Transmit End Interrupt Enable	This bit should set 0 in smart card interface mode. 0: TEI interrupt request is disabled 1: TEI interrupt request is enabled	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b24	SSE	SSn# Pin Function Enable	(Valid in simple SPI mode.) In slave mode (SCR3.CKE[1:0] bits = 1xb), set this bit to 1. 0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W *4
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. In clock synchronous mode (SCR3.MOD[2:0] bits = 010b), simple SPI mode (SCR3.MOD[2:0] bits = 011b), and simple I2C mode (SCR3.MOD[2:0] bits = 100b), 1 can be written only when TE bit = 0 and RE bit = 0. After setting TE bit or RE bit to 1, only 0 can be written in TE bit and RE bit. In other mode, writing is enabled under any condition.
Note 2. In clock synchronous mode and simple SPI mode, receive only setting with internal clock (master mode) is prohibited (TE bit = 0 and RE bit = 1 setting prohibited).
Note 3. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.
Note 4. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

RE Bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception starts when RSCI detects the start bit in synchronous mode, the falling edge of the RXD input in Manchester mode, the synchronous clock input in clock synchronous mode, or the start bit in smart card interface mode.

Note that the SCR0 and SCR3 registers should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.RDRF, AFER, APER, ORER, MMSR.MCER, SBER, SYER, and PFER flags in non-FIFO mode, and RFSR.DR flag in FIFO mode are not affected and the previous values is retained. In smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.AFER, APER, and ORER flags are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission becomes possible. Transmission is started by writing transmit data to TDR register. Note that SCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (SSR.RDRF, ORER, AFER, RFSR.DR, MMSR.MCER, SYER, PFER, SBER) are not set.

When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, refer to section 42.4, Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the multi-processor bit set to 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and AFER, MCER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the multi-processor bit set to 1, the MPB flag is set to 1, the MPIE bit is automatically set to 0, the RXI and ERI interrupt requests are enabled (if SCR0.RIE bit is set to 1), and setting the flags ORER, AFER, MCER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME Bit (Data Compare Match Enable)

It can select whether the data match detection function uses or not.

When DCME bit is 1, if RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, DCME bit is cleared automatically, and after that, RSCI operation mode will be receive mode without data match

detection function.
Refer to section 42.3.6, Data Match Detection.
The write value should be 0 other than asynchronous mode.

IDSEL Bit (ID Frame Select)

This bit specifies the condition of the received data to be compared. The bit is valid only when the DCME bit is 1.
When setting this bit to 1, only data in received frames (ID frames) in which the multi-processor bit has the value 1 are compared.
When this bit is set to 0, all received data is compared.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
RXI and ERI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the SSR.ORER, AFER, or APER flag and then setting the flag to 0, or setting the RIE bit to 0.
In the case of Manchester mode, the MMSR.MCER, SYER, PFER, and SBER flags are also the cause of ERI interrupt request, so the same processing is necessary. For details of these flags, see section 42.2.12, Manchester Mode Control Register (MMCR) and section 42.2.21, Manchester Mode Status Register (MMSR).

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
An TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to SCR0.TE bit and SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.

TEIE Bit (Transmit End Interrupt Enable)

T Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).
In the slave mode (SCR3.CKE[1:0] bits = 10b or 11b), SSE should be set 1.
In the master mode (SCR3.CKE[1:0] bits = 00b or 01b) and single-master, the SSn# pin on the master side is not required to control reception and transmission, so SSE should be set 0.

42.2.6 Control Register 1 (SCR1)

Address(es): RSCI0.SCR1 8745 800Ch, RSCI1.SCR1 8745 810Ch, RSCI2.SCR1 8745 820Ch, RSCI3.SCR1 8745 830Ch, RSCI4.SCR1 8745 840Ch, RSCI5.SCR1 8745 850Ch, RSCI6.SCR1 8745 860Ch, RSCI7.SCR1 8745 870Ch, RSCI8.SCR1 8745 880Ch, RSCI9.SCR1 8745 890Ch, RSCI10.SCR1 8745 8A0Ch, RSCI11.SCR1 8745 8B0Ch, RSCI12.SCR1 8745 8C0Ch,
RSCI0_NS.SCR1 0745 800Ch, RSCI1_NS.SCR1 0745 810Ch, RSCI2_NS.SCR1 0745 820Ch, RSCI3_NS.SCR1 0745 830Ch, RSCI4_NS.SCR1 0745 840Ch, RSCI5_NS.SCR1 0745 850Ch, RSCI6_NS.SCR1 0745 860Ch, RSCI7_NS.SCR1 0745 870Ch, RSCI8_NS.SCR1 0745 880Ch, RSCI9_NS.SCR1 0745 890Ch, RSCI10_NS.SCR1 0745 8A0Ch, RSCI11_NS.SCR1 0745 8B0Ch, RSCI12_NS.SCR1 0745 8C0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	NFM	NFEN	—	NFCS[2:0]			—	—	—	HDSEL	—	—	—	LOOP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CRSEP	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W *1
b1	CRSEP	CTS/RTS Separation*2	0: Use either CTS or RTS function 1: Use both CTS and RTS functions at the same time	R/W *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SPB2DT	Serial Port Break Data	The output level of TXDn (TXDAn/TXDBn*3) pin is selected when SCR0.TE bit = 0 and SPB2IO bit = 1.*4 When TINV is 0, 0: Low level is output in TXDn (TXDAn/TXDBn*3) pin. 1: High level is output in TXDn (TXDAn/TXDBn*3) pin. When TINV is 1, 0: High level is output in TXDn (TXDAn/TXDBn*3) pin. 1: Low level is output in TXDn (TXDAn/TXDBn*3) pin.	R/W
b5	SPB2IO	Serial Port Break I/O	It's selected whether the value of SPB2DT is output to TXDn (TXDAn/TXDBn*3) pin when SCR0.TE = 0.*4 0: The value of SPB2DT bit isn't output in TXDn (TXDAn/TXDBn*3) pin. 1: The value of SPB2DT bit is output in TXDn (TXDAn/TXDBn*3) pin.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	PE	Parity Enable	(Valid only in asynchronous mode and Manchester mode. In smart card interface mode, set 1 to this bit.) When transmitting 0: Parity bit addition is not performed 1: The parity bit is added When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W *1
b9	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	TINV	Transmitter Output Invert*5	0: Transmit data is not inverted and output to TXDn (TXDAn/TXDBn*3) pin. 1: Transmit data is inverted and output to TXDn (TXDAn/TXDBn*3) pin.	R/W *1
b13	RINV	Receiver Input Invert*5	0: Received data from RXDn is not inverted and input. 1: Received data from RXDn is inverted and input.	R/W *1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b16	LOOP	Loopback Mode Setting	It can be used when internal clock operation in asynchronous mode, internal mode operation in Manchester mode, internal clock operation in clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	HDSEL	Half-Duplex Communication Select	In the smart card interface mode, the simple I2C mode, or in the simple SPI mode, this bit should be set 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (Half-duplex communication using TXDn pin)	R/W *1
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	NFCS[2:0]	Noise Filter Clock Select	(Valid in asynchronous mode and Manchester mode, extended serial mode, and simple I2C mode.) Select for the noise filter's clock source. b26 b24 0 0 0: The base clock signal divided by 1. 0 0 1: The on-chip baud rate generator source clock*6 divided by 1. 0 1 0: The on-chip baud rate generator source clock*6 divided by 2. 0 1 1: The on-chip baud rate generator source clock*6 divided by 4. 1 0 0: The on-chip baud rate generator source clock*6 divided by 8. Settings other than above are prohibited. In simple I2C mode, 000b setting is prohibited. “The on-chip baud rate generator source clock” means the clock selected by SCR2.CKS[1:0] bits.	R/W *1
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	NFEN	Digital Noise Filter Enable	(Valid in asynchronous mode, Manchester mode, extended serial mode and simple I2C) In asynchronous mode, Manchester mode and extended serial mode 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. In simple I2C mode 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled.	R/W *1
b29	NFM	Digital Noise Filter Mode	(Valid in asynchronous mode with the SCR2.ABCSE2 bit set to 1) 0: Three times match mode 1: Majority mode	R/W *1
b31 to b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. This bit is available in asynchronous mode and Manchester mode. Set this bit to 0 in other mode.
Note 3. When the alternate output is enabled in HBS support mode.
Note 4. Use this bit in asynchronous mode and Manchester mode only. Movement by other mode isn’t guaranteed.
Note 5. The RINV and TINV bits should be set to 0 in smart card interface mode and simple I2C mode.
Note 6. The clock is selected by the SCR2.CKS[1:0] bits.

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, simple I2C mode, and extended serial mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CRSEP Bit (CTS/RTS Separation)

This bit selects usage of the CTSn#, RTSn#, and CTSn#/RTSn# pins when the CTSE bit is 1.
When either CTS or RTS function is to be used, set this bit to 0.
When both CTS and RTS functions are to be used, set this bit to 1.
When the CTSE bit is set to 0, set this bit to 0.
Refer to Table 42.6 for the relationship between the CRSEP and CTSE bit settings and the pin functions.

Table 42.6 Relationship between the CRSEP and CTSE Bit Settings and Pin Functions

CTSE Bit	CRSEP Bit	CTSn#/RTSn# Multiplexed Pin	CTSn# Dedicated Pin	RTSn# Dedicated Pin
0	0	RTSn# signal output	Disabled	RTSn# signal output
1	0	CTSn# signal input	CTSn# signal input	Disabled
1	1	RTSn# signal output	CTSn# signal input	RTSn# signal output

SPB2DT Bit (Serial Port Break Data), SPB2IO Bit (Serial Port Break I/O)

The TXDn (TXDAn/TXDBn) pins status decided by combination of SCR0.TE bit, SCR1.SPB2IO bit and SCR1.SPB2DT bit is indicated in Table 42.7.

Table 42.7 Controlling the TXDn (TXDAn/TXDBn) Pins

SCR0.TE Bit Setting	SCR1.SPB2IO Bit Setting	SCR1.SPB2DT Bit Setting	TINV Bit Setting	TXDn Pin Status
0 (Transmission disabled)	0 (Input)	0 or 1	0 or 1	Hi-Z
	1 (Output)	0	0	Low is output
			1	High is output
		1	0	High is output
			1	Low is output
1 (Transmission enabled)	0 or 1	0 or 1	0 or 1	Transmit data output pin

PE Bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.
For details on the usage of this bit in smart card interface mode, refer to section 42.7.2, Data Format (Except in Block Transfer Mode).

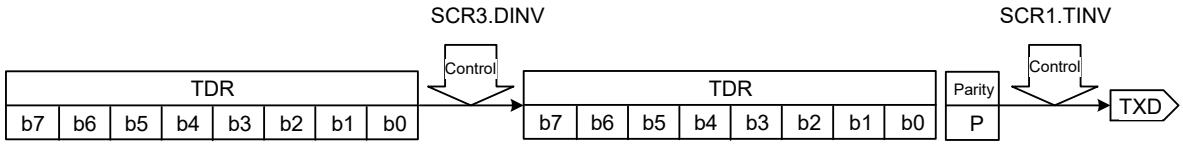
TINV Bit (Transmitter Output Invert), RINV Bit (Receiver Input Invert)

The data of RDR register is controlled by RINV bit and SCR3.DINV bit. And the data from TXDn pin is controlled by TINV bit and SCR3.DINV bit. The control by RINV/TINV bits are done to communication pins (RXDn/TXDn), so they can control not only data bits but also other bits (start bit, stop bit, parity bit). Please refer to Figure 42.2 in detail. When the TXDAn/TXDBn pins are used, the data is also inverted according to the TINV value.
During half-duplex communication and slave operation in simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Sentences and a timing chart of the RSCI operation explanation are mentioned by TINV bit = 0 and RINV bit = 0 when TINV's value and RINV's value are not specified.

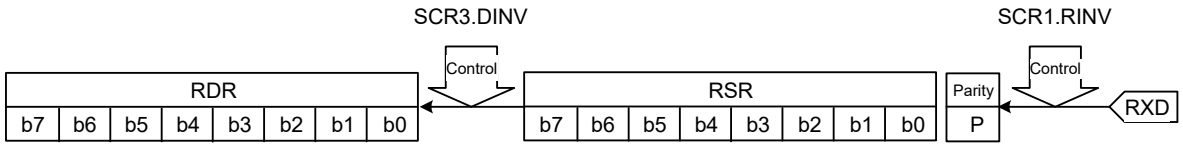
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

The transmit data is controlled by SCR1.TINV and SCR3.DINV.



SCR3. DINV	SCR1. TINV	TDR	TSR	Parity (even)	TXDn waveform
0	0	BEh	BEh	0	
0	1	BEh	BEh	0	
1	0	BEh	41h	0	
1	1	BEh	41h	0	

The received data is controlled by SCR1.RINV and SCR3.DINV.



SCR3. DINV	SCR1. RINV	RDR	RSR	Parity (even)	RXDn waveform
0	0	BEh	BEh	0	
1	0	41h	BEh	0	
0	1	BEh	BEh	0	
1	1	41h	BEh	0	

Figure 42.2 Example of the Receive/Transmit Data Control

LOOP Bit (Loopback Mode Setting)

When this bit is 1, RSCI blocks the input path from RXD and connects the output path to TXD to the reception data register.
Transmit data can be inverted and received by combining it with TINV bit.
Clock synchronous mode Set to 0 at slave operation and asynchronous mode use of external clock, and extended serial mode.

HDSEL Bit (Half-Duplex Communication Select)

Setting this bit to 1 enables half-duplex communication using the TXDn pin. However, it cannot be used in simple SPI mode, simple I^2C mode and smart card interface mode.
If this bit is set to 1 and SCR0.TE bit = 1, SCR0.RE bit = 0, the TXDn pin becomes communication output. If this bit is set to 1 and SCR0.TE bit = 0, SCR0.RE bit = 1, the TXDn pin becomes the communication input. For details, see section 42.16, Half-Duplex Communication Function.

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter.
To use the noise filter in asynchronous mode, Manchester mode and extended serial mode set these bits from 000b to 100b. In simple I^2C mode, set the bits to a value in the range from 001b to 100b.

NFEN Bit (Digital Noise Filter Enable)

This bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, Manchester mode, extended serial mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

NFM Bit (Digital Noise Filter Mode)

This bit selects the operation mode of the digital noise filter function. Normally, set this bit to 0. When the SCR2.ABCSE2 bit is set to 1 in asynchronous mode, set this bit to 1.

42.2.7 Control Register 2 (SCR2)

Address(es): RSCI0.SCR2 8745 8010h, RSCI1.SCR2 8745 8110h, RSCI2.SCR2 8745 8210h, RSCI3.SCR2 8745 8310h, RSCI4.SCR2 8745 8410h, RSCI5.SCR2 8745 8510h, RSCI6.SCR2 8745 8610h, RSCI7.SCR2 8745 8710h, RSCI8.SCR2 8745 8810h, RSCI9.SCR2 8745 8910h, RSCI10.SCR2 8745 8A10h, RSCI11.SCR2 8745 8B10h, RSCI12.SCR2 8745 8C10h,
RSCI0_NS.SCR2 0745 8010h, RSCI1_NS.SCR2 0745 8110h, RSCI2_NS.SCR2 0745 8210h, RSCI3_NS.SCR2 0745 8310h, RSCI4_NS.SCR2 0745 8410h, RSCI5_NS.SCR2 0745 8510h, RSCI6_NS.SCR2 0745 8610h, RSCI7_NS.SCR2 0745 8710h, RSCI8_NS.SCR2 0745 8810h, RSCI9_NS.SCR2 0745 8910h, RSCI10_NS.SCR2 0745 8A10h, RSCI11_NS.SCR2 0745 8B10h, RSCI12_NS.SCR2 0745 8C10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BRR[7:0]								ABCSE2	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BCP[2:0]	Base Clock Pulse	Selects the number of base clock cycles in smart card interface mode. b2 b0 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	BGDM	Baud Rate Generator Double-Speed Mode Select	Valid in asynchronous/Manchester/clock synchronous/simple SPI mode and SCR3.CKE[1] bit = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W *1
b5	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode, Manchester mode and extended serial mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W *1
b6	ABCSE	Asynchronous Mode Base Clock Select Extended*3	(Valid only in asynchronous mode and SCR3.CKE[1] bit = 0) 0: Clock cycles for 1-bit period is decided with combination between SCR2.BGDM bit and SCR2.ABCS bit. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W *1
b7	ABCSE2	Asynchronous Mode Base Clock Select Extended 2*3	(Valid only in asynchronous mode and SCR3.CKE[1] bit = 0) 0: Clock cycles for 1-bit period is decided with combination between SCR2.BGDM bit and SCR2.ABCS bit. 1: Baud rate is 4 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W *1
b15 to b8	BRR[7:0]	Bit Rate Setting	An 8-bit field that adjusts the bit rate.	R/W *1
b16	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b21, b20	CKS[1:0]	Clock Select	b21 b20 0 0: TCLK (n = 0)*4 0 1: TCLK/4 (n = 1)*4 1 0: TCLK/16 (n = 2)*4 1 1: TCLK/64 (n = 3)*4	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b24	MDDR[7:0]	Modulation Duty Setting	MDDR[7:0] bits corrects the bit rate adjusted by the BRR[7:0] bits.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. S is the value of S in BRR[7:0] bits explanation.
Note 3. Do not set both the ABCSE and ABCSE2 bits to 1 at the same time.
Note 4. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.
For details, refer to section 42.7.4, Receive Data Sampling Timing and Reception Margin.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR3.CKE[1] bit = 0) in asynchronous mode, Manchester mode, clock synchronous mode, simple SPI mode. When external clock is selected (SCR3.CKE[1] bit = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode or Manchester mode or clock synchronous mode or simple SPI.

ABCS Bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.
Set it to 0 in modes other than asynchronous mode, Manchester mode and extended serial mode.

ABCSE Bit (Asynchronous Mode Base Clock Select Extended)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, please use this bit and set SCR2.CKS[1:0] bits = 00b and BRR[7:0] bits = 0.
Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

ABCSE2 Bit (Asynchronous Mode Base Clock Select Extended 2)

The pulse number for a base clock at 1-bit period is 4 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 4 dividing frequency of the bus clock, please use this bit and set SCR2.CKS[1:0] bits = 00b and BRR[7:0] bits = 0.
Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 42.8 Base Clock Cycle Number per 1-Bit Period

ABCSE2 Bit	ABCSE Bit	ABCS Bit	BGDM Bit	Number of Base Clocks Per 1-Bit Period	The Output Frequency of the Baud Rate Generator
0	0	0	0	16	×1
0	0	0	1	16	×2
0	0	1	0	8	×1
0	0	1	1	8	×2
0	1	0 or 1	0 or 1	6	×2
1	0	0 or 1	0 or 1	4	×2

BRR[7:0] Bits (Bit Rate Setting)

BRR[7:0] bits are an 8-bit field that adjusts the bit rate.

RSCI has independent baud rate generator control, different bit rates can be set for each. Table 42.9 shows the relationship between the setting (N) in the BRR[7:0] bits and the bit rate (B) for asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.

Table 42.9 Relationship between N Setting in BRR[7:0] Bits and Bit Rate B

Mode	SCR2 Settings				BRR[7:0] Bit Setting	Error (%)
	BGDM bit	ABCS bit	ABCSE bit	ABCSE2 bit		
Asynchronous, multi-processor communication, Manchester*1, extended serial*2	0	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0	0		
	1	1	0	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	0	N = 0*3	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	0	1	N = 0*3	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 8 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI	0	0	0	0	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
	1	0	0	0	$N = \frac{TCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	
Smart card interface					$N = \frac{TCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*4					$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)

N: BRR[7:0] bit setting ($0 \le N \le 255$)

TCLK: Operating frequency (MHz)

n and S: Determined by the settings of the SCR2 registers as listed in the table below.Please be careful about "$2^{(2n+1)}$" is used in the expression for smart card interface, "$2^{(2n-1)}$" is used in other mode.

Note 1. In Manchester mode, set the ABCSE and ABCSE2 bits to 0.
Note 2. In extended serial mode, set the BGDM, ABCSE, and ABCSE2 bits to 0.
Note 3. When the ABCSE or ABCSE2 bit is set to 1, set N = 0 and n = 0.
Note 4. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

Table 42.10 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I2C	High period (minimum value)	$(N+1)\times 4\times 2^{2n-1}\times 7\times \frac{1}{TCLK\times 10^{6}}$
	Low period (minimum value)	$(N+1)\times 4\times 2^{2n-1}\times 8\times \frac{1}{TCLK\times 10^{6}}$

Table 42.11 Clock Source Settings

SCR2 Setting		
CKS[1:0] Bits	Clock Source	n
0 0	TCLK	0
0 1	TCLK/4	1
1 0	TCLK/16	2
1 1	TCLK/64	3

Table 42.12 Base Clock Settings in Smart Card Interface Mode

SCR2 Setting		
BCP[2:0] Bits	Base Clock Cycles for 1-bit Period	S
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 42.13 and Table 42.14 list examples of N settings in BRR[7:0] in asynchronous mode and Manchester mode. Table 42.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR[7:0] bits (N) settings in clock synchronous mode and simple SPI mode are listed in Table 42.18. Examples of BRR[7:0] bits (N) settings in smart card interface mode are listed in Table 42.20. Examples of BRR[7:0] bits (N) settings in simple I2C mode are listed in Table 42.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 42.7.4, Receive Data Sampling Timing and Reception Margin. Table 42.16 and Table 42.19 list the maximum bit rates with external clock input. Table 42.17 shows the maximum bit rate with TMR clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and Manchester mode, the bit rate becomes twice that listed in Table 42.13 and Table 42.14. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 42.13 Examples of BRR[7:0] Bit Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (1)

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 42.14 Examples of BRR[7:0] Bit Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2)

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	50			60			100			120			150		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16	3	243	0.06
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35	3	121	0.06
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16	3	60	0.06
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35	2	121	0.06
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16	2	60	0.06
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35	1	121	0.06
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16	1	60	0.06
31250	0	49	0.00	0	59	0.00	1	24	0.00	0	119	0.00	0	149	0.00
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35	0	121	0.06

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 42.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester Mode) (1/2)

TCLK (MHz)	SCR2 Settings BGDM Bit	ABCS Bit	ABCSE Bit	ABCSE2 Bit	n	N	Maximum Bit Rate (bps)	TCLK (MHz)	SCR2 Settings BGDM Bit	ABCS Bit	ABCSE Bit	ABCSE2 Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	0	0	250000	9.8304	0	0	0	0	0	0	307200
		1	0	0	0	0	500000			1	0	0	0	0	614400
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	1000000			1	0	0	0	0	1228800
	0 or 1	0 or 1	1	0	0	0	1333333		0 or 1	0 or 1	1	0	0	0	1638400
	0 or 1	0 or 1	0	1	0	0	2000000		0 or 1	0 or 1	0	1	0	0	2457600
10	0	0	0	0	0	0	312500	12	0	0	0	0	0	0	375000
		1	0	0	0	0	625000			1	0	0	0	0	750000
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	1250000			1	0	0	0	0	1500000
	0 or 1	0 or 1	1	0	0	0	1666667		0 or 1	0 or 1	1	0	0	0	2000000
	0 or 1	0 or 1	0	1	0	0	2500000		0 or 1	0 or 1	0	1	0	0	3000000
12.288	0	0	0	0	0	0	384000	14	0	0	0	0	0	0	437500
		1	0	0	0	0	768000			1	0	0	0	0	875000
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	1536000			1	0	0	0	0	1750000
	0 or 1	0 or 1	1	0	0	0	2048000		0 or 1	0 or 1	1	0	0	0	2333333
	0 or 1	0 or 1	0	1	0	0	3072000		0 or 1	0 or 1	0	1	0	0	3500000
16	0	0	0	0	0	0	500000	17.2032	0	0	0	0	0	0	537600
		1	0	0	0	0	1000000			1	0	0	0	0	1075200
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	2000000			1	0	0	0	0	2150400
	0 or 1	0 or 1	1	0	0	0	2666667		0 or 1	0 or 1	1	0	0	0	2867200
	0 or 1	0 or 1	0	1	0	0	4000000		0 or 1	0 or 1	0	1	0	0	4300800
18	0	0	0	0	0	0	562500	19.6608	0	0	0	0	0	0	614400
		1	0	0	0	0	1125000			1	0	0	0	0	1228800
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	2250000			1	0	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	0	3000000		0 or 1	0 or 1	1	0	0	0	3276800
	0 or 1	0 or 1	0	1	0	0	4500000		0 or 1	0 or 1	0	1	0	0	4915200
20	0	0	0	0	0	0	625000	25	0	0	0	0	0	0	781250
		1	0	0	0	0	1250000			1	0	0	0	0	1562500
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	2500000			1	0	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	0	3333333		0 or 1	0 or 1	1	0	0	0	4166667
	0 or 1	0 or 1	0	1	0	0	5000000		0 or 1	0 or 1	0	1	0	0	6250000
30	0	0	0	0	0	0	937500	33	0	0	0	0	0	0	1031250
		1	0	0	0	0	1875000			1	0	0	0	0	2062500
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	3750000			1	0	0	0	0	4125000
	0 or 1	0 or 1	1	0	0	0	5000000		0 or 1	0 or 1	1	0	0	0	5500000
	0 or 1	0 or 1	0	1	0	0	7500000		0 or 1	0 or 1	0	1	0	0	8250000

Table 42.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester Mode) (2/2)

TCLK (MHz)	SCR2 Settings						Maximum Bit Rate (bps)	TCLK (MHz)	SCR2 Settings						Maximum Bit Rate (bps)
	BGDM Bit	ABCS Bit	ABCSE Bit	ABCSE2 Bit	n	N			BGDM Bit	ABCS Bit	ABCSE Bit	ABCSE2 Bit	n	N	
40	0	0	0	0	0	0	1250000	50	0	0	0	0	0	0	1562500
		1	0	0	0	0	2500000			1	0	0	0	0	3125000
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	5000000			1	0	0	0	0	6250000
	0 or 1	0 or 1	1	0	0	0	6666667		0 or 1	0 or 1	1	0	0	0	8333333
	0 or 1	0 or 1	0	1	0	0	10000000		0 or 1	0 or 1	0	1	0	0	12500000
60	0	0	0	0	0	0	1875000	120	0	0	0	0	0	0	3750000
		1	0	0	0	0	3750000			1	0	0	0	0	7500000
	1	0	0	0	0	0			1	0	0	0	0	0	
		1	0	0	0	0	7500000			1	0	0	0	0	15000000
	0 or 1	0 or 1	1	0	0	0	10000000		0 or 1	0 or 1	1	0	0	0	20000000
	0 or 1	0 or 1	0	1	0	0	15000000		0 or 1	0 or 1	0	1	0	0	30000000
150	0	0	0	0	0	0	4687500								
		1	0	0	0	0	9375000								
	1	0	0	0	0	0									
		1	0	0	0	0	18750000								
	0 or 1	0 or 1	1	0	0	0	25000000								
	0 or 1	0 or 1	0	1			37500000								

Table 42.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

TCLK (MHz)	Maximum Frequency of External Clock (MHz)	Maximum Bit Rate (bps)	
		SCR2.ABCS Bit = 0	SCR2.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000
120	30.0000	1875000	3750000
150	37.5000	2343750	4687500

Table 42.17 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

TCLK (MHz)	Maximum Frequency of TMR Clock (MHz)	Maximum Bit Rate (bps)	
		SCR2.ABCS Bit = 0	SCR2.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000
33	16.5	1031250	2062500
40	20	1250000	2500000
50	25	1562500	3125000
60	30	1875000	3750000
75	37.5	2343750	4687500
120	30	1875000	3750000
150	37.5	2343750	4687500

Table 42.18 Examples of BRR[7:0] Bit Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency TCLK (MHz)																	
	8			10			30			60			120			150		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	155	—	—	—	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—	—	—	—
1 k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—	—	—	—
2.5 k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187	0	3	233
5 k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93	0	3	116
10 k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46	0	2	233
25 k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74	0	2	93
50 k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149	0	2	46
100 k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74	0	1	93
250 k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29	0	0	149
500 k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14	0	0	74
1 M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29	1	0	74
2.5 M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2	1	0	29
5 M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5	1	0	14
7.5 M	—	—	—	—	—	—	0	0	0	0	0	1	0	1	0	1	0	9
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0	—	—	—

—: Can be set, but an error over 10% will occur.

Table 42.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

TCLK (MHz)	Maximum Frequency of External Clock (MHz)	Maximum Bit Rate (Mbps)
8	4	4
10	5	5
12	6	6
14	7	7
16	8	8
18	9	9
20	10	10
25	12.5	12.5
30	15	15
33	16.5	16.5
40	20	20
50	25	25
60	30	30
120	60	60
150	75	75

Table 42.20 Examples of BRR[7:0] Bit Settings for Various Bit Rates (Smart Card Interface Mode, S = 372)

Bit Rate (bps)	TCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01
	33.00	0	4	–7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01
	120.00	0	16	–1.17
	150.00	0	20	0.01

Table 42.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

TCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0
120.00	1875000	0	0
150.00	2343750	0	0

Table 42.22 Examples of BRR[7:0] Bit Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	30	0.8	0	49	0.0	0	62	–0.8	0	77	0.2
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	0	24	0.0	0	30	0.8
50 k	0	4	0.0	0	5	4.2	0	9	0.0	0	12	–3.8	2	0	–2.3
100 k	0	2	–16.7	1	0	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	0	25.0	0	1	0.0	0	2	–16.7	0	2	4.2
350 k										0	1	–10.7	0	2	–25.6
384 k										0	1	–18.6	0	2	–32.2

Bit Rate (bps)	Operating Frequency TCLK (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	93	–0.3	0	102	0.1	0	124	0.0	1	38	0.2	1	46	–0.3
25 k	0	37	–1.3	0	40	0.6	0	49	0.0	0	62	–0.8	0	74	0.0
50 k	0	18	–1.3	0	20	–1.8	0	24	0.0	0	30	0.8	0	37	–1.3
100 k	0	9	–6.3	0	10	–6.3	0	12	–3.8	2	0	–2.3	0	18	–1.3
250 k	1	0	–6.3	1	0	3.1	0	4	0.0	0	5	4.2	1	1	–6.3
350 k	0	2	–10.7	0	2	–1.8	1	0	–10.7	0	4	–10.7	0	4	7.1
384 k	0	2	–18.6	0	2	–10.5	1	0	–18.6	0	4	–18.6	0	4	–2.3

Bit Rate (bps)	Operating Frequency TCLK (MHz)					
	120			150		
	n	N	Error (%)	n	N	Error (%)
10 k	1	93	–0.3	1	116	0.2
25 k	0	149	0.0	1	46	–0.3
50 k	0	74	0.0	0	93	–0.3
100 k	0	37	–1.3	0	46	–0.3
250 k	0	14	0.0	0	18	–1.3
350 k	0	10	–2.6	0	12	3.0
384 k	0	9	–2.3	0	12	–6.1

Table 42.23 Minimum High and Low Periods of the SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)
10 k	0	24	43.75/50.00	0	30	43.40/49.60	0	49	43.75/50.00	0	62	44.10/50.40
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	0	24	17.50/20.00
50 k	0	4	8.75/10.00	0	5	8.40/9.60	0	9	8.75/10.00	0	12	9.10/10.40
100 k	0	2	5.25/6.00	1	0	5.60/6.40	0	4	4.38/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	0	1.40/1.60	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60
384 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)
10 k	0	77	43.68/49.92	0	93	43.87/50.13	0	102	43.70/49.94	0	124	43.75/50.00
25 k	0	30	17.36/19.84	0	37	17.73/20.27	0	40	17.39/19.88	0	49	17.50/20.00
50 k	2	0	8.96/10.24	0	18	8.87/10.13	0	20	8.91/10.18	0	24	8.75/10.00
100 k	1	1	4.48/5.12	0	9	4.67/5.33	0	10	4.67/5.33	0	12	4.55/5.20
250 k	0	2	1.68/1.92	1	0	1.87/2.13	1	0	1.70/1.94	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	1	0	1.40/1.60
384 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	1	0	1.40/1.60

Bit Rate (bps)	Operating Frequency TCLK (MHz)											
	50			60			120			150		
	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)	n	N	Min. High/Low Period of SCL (µs)
10 k	1	38	43.68/49.92	1	46	43.87/50.13	1	93	43.87/50.13	1	116	43.68/49.92
25 k	0	62	17.64/20.16	0	74	17.50/20.00	0	149	17.50/20.00	1	46	17.55/20.05
50 k	0	30	8.68/9.92	0	37	8.87/10.13	0	74	8.75/10.00	0	93	8.77/10.03
100 k	2	0	4.48/5.12	0	18	4.43/5.07	0	37	4.43/5.07	0	46	4.39/5.01
250 k	0	5	1.68/1.92	1	1	1.87/2.13	0	14	1.75/2.00	0	18	1.77/2.03
350 k	0	4	1.40/1.60	0	4	1.17/1.33	0	10	1.28/1.47	0	12	1.21/1.39
384 k	0	4	1.40/1.60	0	4	1.17/1.33	0	9	1.17/1.33	0	12	1.21/1.39

BRME Bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

This bit can only be set to 1 in asynchronous mode and simple I²C mode. Set this bit to 0 in clock synchronous mode, simple SPI mode, smart card interface mode, Manchester mode, and extended serial mode. Also, Set this bit to 0 when the ABCSE2 bit is set to 1.

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.
For the relationship between the settings of these bits and the baud rate, refer to BRR[7:0] bits explanation.

MDDR[7:0] Bits (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR[7:0] bit setting (M) and the bit rate (B) is given in Table 42.24.
The initial value of MDDR[7:0] bits is FFh. Bit 7 in this register is fixed to 1.

Table 42.24 Relationship between MDDR[7:0] Bit Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	SCR2 Settings			BRR[7:0] Bit Setting	Error (%)
	BGDM Bit	ABCS Bit	ABCSE Bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	N = 0*2	$Error = \left\{ \frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I2C*3				$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR[7:0] bit setting (128 ≤ M ≤ 255)
N: BRR[7:0] bit setting (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n: Determined by the settings of the SCR2.CKS[1:0] bits as listed in Table 42.11, Clock Source Settings.
Note 1. Do not use this function in clock synchronous mode, simple SPI mode, smart card Interface mode, Manchester mode and extended serial mode.
Note 2. When the ABCSE bit is set to 1, set N = 0 and n = 0.
Note 3. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

Table 42.25 and Table 42.26 list examples of N settings in BRR[7:0] bits and M settings in MDDR[7:0] bits in asynchronous mode.

Table 42.25 Examples of BRR[7:0] Bits and MDDR[7:0] Bit Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency TCLK (MHz)														
	8					9.8304					10				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0.00	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

	Operating Frequency TCLK (MHz)														
	12					12.288					14				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

	Operating Frequency TCLK (MHz)														
	16					17.2032					18				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0.00	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.20	0	0	210	0	0.14

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

Table 42.26 Examples of BRR[7:0] Bits and MDDR[7:0] Bit Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency TCLK (MHz)														
	19.6608					20					25				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	15	(256) *1	0	0.00	0	10	173	0	–0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

	Operating Frequency TCLK (MHz)														
	30					33					40				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

	Operating Frequency TCLK (MHz)														
	50					60					120				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	23	151	0	0.00	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0.00	0	21	173	0	–0.01	0	58	232	0	0.00
115200	0	7	151	0	0.00	0	10	173	0	–0.01	0	21	173	0	–0.01
230400	0	3	151	0	0.00	0	10	173	1	–0.01	0	10	173	0	–0.01
460800	0	1	151	0	0.00	0	6	220	1	–0.09	0	10	173	1	–0.01

	Operating Frequency TCLK (MHz)				
	150				
Bit Rate (bps)	n	N	M	BGDM Bit	Error (%)
38400	0	71	151	0	0.00
57600	0	47	151	0	0.00
115200	0	23	151	0	0.00
230400	0	11	151	0	0.00
460800	0	5	151	0	0.00

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

42.2.8 Control Register 3 (SCR3)

Address(es): RSCI0.SCR3 8745 8014h, RSCI1.SCR3 8745 8114h, RSCI2.SCR3 8745 8214h, RSCI3.SCR3 8745 8314h, RSCI4.SCR3 8745 8414h, RSCI5.SCR3 8745 8514h, RSCI6.SCR3 8745 8614h, RSCI7.SCR3 8745 8714h, RSCI8.SCR3 8745 8814h, RSCI9.SCR3 8745 8914h, RSCI10.SCR3 8745 8A14h, RSCI11.SCR3 8745 8B14h, RSCI12.SCR3 8745 8C14h,
RSCI0_NS.SCR3 0745 8014h, RSCI1_NS.SCR3 0745 8114h, RSCI2_NS.SCR3 0745 8214h, RSCI3_NS.SCR3 0745 8314h, RSCI4_NS.SCR3 0745 8414h, RSCI5_NS.SCR3 0745 8514h, RSCI6_NS.SCR3 0745 8614h, RSCI7_NS.SCR3 0745 8714h, RSCI8_NS.SCR3 0745 8814h, RSCI9_NS.SCR3 0745 8914h, RSCI10_NS.SCR3 0745 8A14h, RSCI11_NS.SCR3 0745 8B14h, RSCI12_NS.SCR3 0745 8C14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	BLK	GM	—	ACS0	CKE[1:0]		—	—	DEEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	STOP	DINV	DDIR	—	—	CHR[1:0]		SYNDIS	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	Clock Phase Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed.)	R/W *1
b1	CPOL	Clock Polarity Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W *1
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	SYNDIS	Synchronizer Disable	0: TCLK = SCICLK, synchronizer enabled 1: TCLK = PCLK, synchronizer disabled	R/W *1
b9, b8	CHR[1:0]	Character Length Select	(Valid in asynchronous mode and Manchester mode)*2 Select the data length for transmission and reception. b9 b8 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	DDIR	Transfer Data Direction Select	0: MSB first 1: LSB first Set this bit to 0 in simple I^2C mode and set this bit to 1 in extended serial mode.	R/W *1
b13	DINV	Transfer Data Invert	0: TDR register contents are transmitted to TSR register as they are. RSR register contents are stored to RDR register as they are. 1: TDR register contents are inverted before being transmitted to TSR register. RSR register contents are inverted and stored to RDR register. Set this bit to 0 in simple I^2C mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and SCR1.TINV/RINV. Please refer to Figure 42.2 for details.	R/W *1
b14	STOP	Stop Bit Length Select	(Valid in asynchronous mode, Manchester mode, extended serial mode) 0: 1 stop bit/break delimiter length is 1bit 1: 2 stop bits/break delimiter length is 2bits	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b15	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) Set this bit to 1 in extended serial mode. 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W *1
b18 to b16	MOD[2:0]	Communication Mode Select	Select the RSCI communication mode. b18 b16 0 0 0: Asynchronous mode 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple I^2C mode 1 0 1: Manchester mode 1 1 0: Extended serial mode 1 1 1: Setting prohibited	R/W *1
b19	MP	Multi-Processor Mode	(Valid in asynchronous mode, Manchester mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W *1
b20	FM	FIFO Mode Select	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) 0: TDR register, RDR register is non-FIFO buffer configuration 1: TDR register, RDR register is FIFO buffer configuration	R/W *1
b21	DEEN	Driver Control Function Enable	(Valid only in asynchronous mode) 0: RS-485 driver control function disable. 1: RS-485 driver control function enable.	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25, b24	CKE[1:0]	Clock Enable	In the case of asynchronous mode b25 b24 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock A clock with a frequency of 16 times (when the SCR2.ABCS bit is 0) or 8 times (when the SCR2.ABCS bit is 1) the bit rate should be input from the SCKn pin or TMR. When the TMR clock is used, the SCKn pin can be used as an I/O port by setting the I/O port. In the case of Manchester mode and extended serial mode b25 b24 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. Settings other than above are prohibited. In the case of clock synchronous mode and simple SPI mode b25 b24 0 x: Internal clock (master operation) The SCKn pin functions as the clock output pin. 1 x: External clock (slave operation) The SCKn pin functions as the clock input pin. In the case of smart card interface mode When SCR3.GM bit = 0 b25 b24 0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.) 0 1: Clock output 1 x: Prohibited When SCR3.GM bit = 1 b25 b24 0 0: Output fixed low 0 1: Clock output 1 0: Output fixed high 1 1: Clock output	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b26	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock 1: TMR clock (logical AND of two compare-match outputs from the TMR)*4	R/W *1
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	GM	GSM Mode	(Valid only in smart card interface mode) 0: Non-GSM mode operation 1: GSM mode operation	R/W *1
b29	BLK	Block Transfer Mode	(Valid only in smart card interface mode) 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. In other than asynchronous mode and Manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. Set these bits to 10b in extended serial mode.
Note 3. LSB first is fixed and the MSB (bit 7) in TDR register is not transmitted in transmission.
Note 4. When the ACS0 bit is set to 1, the SYNDIS bit should also be set to 1.

CPHA Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 42.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 42.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

SYNDIS Bit (Synchronizer Disable)

This bit is used to bypass the synchronizers for clock domain crossing. When this bit is set to 1, the peripheral module clock (PCLK) is selected as the operating clock (TCLK) and the synchronizers are bypassed. For details, refer to section 42.17, Synchronizer Bypass Function.

CHR[1:0] Bits (Character Length Select)

Selects the data length for transmission and reception.
Except of asynchronous mode and Manchester mode, a fixed data length of 8 bits is used.

DDIR Bit (Transfer Data Direction Select)

Select whether to transmit/receive data in MSB first or LSB first.

DINV Bit (Transfer Data Invert)

DINV bit can invert the transmit data bit from TDR register to TSR register, and also can invert the received data from RSR register to RDR register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SCR1.PM bit.

STOP Bit (Stop Bit Length Select)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.
In addition, it is used as break delimiter length setting when sending Start Frame in extended serial mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 1 in extended serial mode. Set this bit to 0 in modes except of asynchronous mode and extended serial mode.

MOD[2:0] Bits (Communication Mode Select)

Selects the RSCI communication mode.

Table 42.27 Relationship between Communication Mode Selection Bits (MOD[2:0]), Other Operation Mode Setting Bits

Communication mode	Asynchronous								Smart Card I/F	Clock Synchronous		Simple SPI				Simple I²C	Manchester		Extended Serial
SCR3.MOD[2:0]	000b								001b	010b		011b				100b	101b		110b
SCR3.MP	0				1				—	—		—				—	0	1	—
SCR3.FM	0		1		0		1		—	0	1	0		1		—	—		—
SCR3.DEEN	0	1	0	1	0	1	0	1	—	—		—				—	—		—
SCR3.SSE	—								—	—		0	1	0	1	—	—		—

—: Prohibited setting

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM Bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register/RDR register switches to FIFO configuration, and transmit FIFO (TDR register)/receive FIFO (RDR register) can be used for serial transmission/reception.

DEEN Bit (Driver Control Function Enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

The combination of the settings of these bits and of the SCR3.ACS0 bit sets the TMR clock.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 42.7.8, Clock Output Control.

ACS0 Bit (Asynchronous Mode Clock Source Select)

This bit is used to select the clock source in the asynchronous mode. Set this bit to 0 in other than asynchronous mode. The ACS0 bit is valid when an external clock is selected (SCR3.CKE[1] bit = 1) in asynchronous mode (SCR3.MOD[2:0] bits = 000b). The external clock or the TMR clock can be selected. When the TMR clock is selected, set the SYNDIS bit to 1 to select PCLK as the operating clock (TCLK).

The TMR clock is the logical AND of two compare-match outputs from the TMR. The even channels of the RSCI use the outputs of TMR unit 0 (TMO0, TMO1), and the odd channels use the outputs of TMR unit 1 (TMO2, TMO3). A setting example is shown in Figure 42.3.

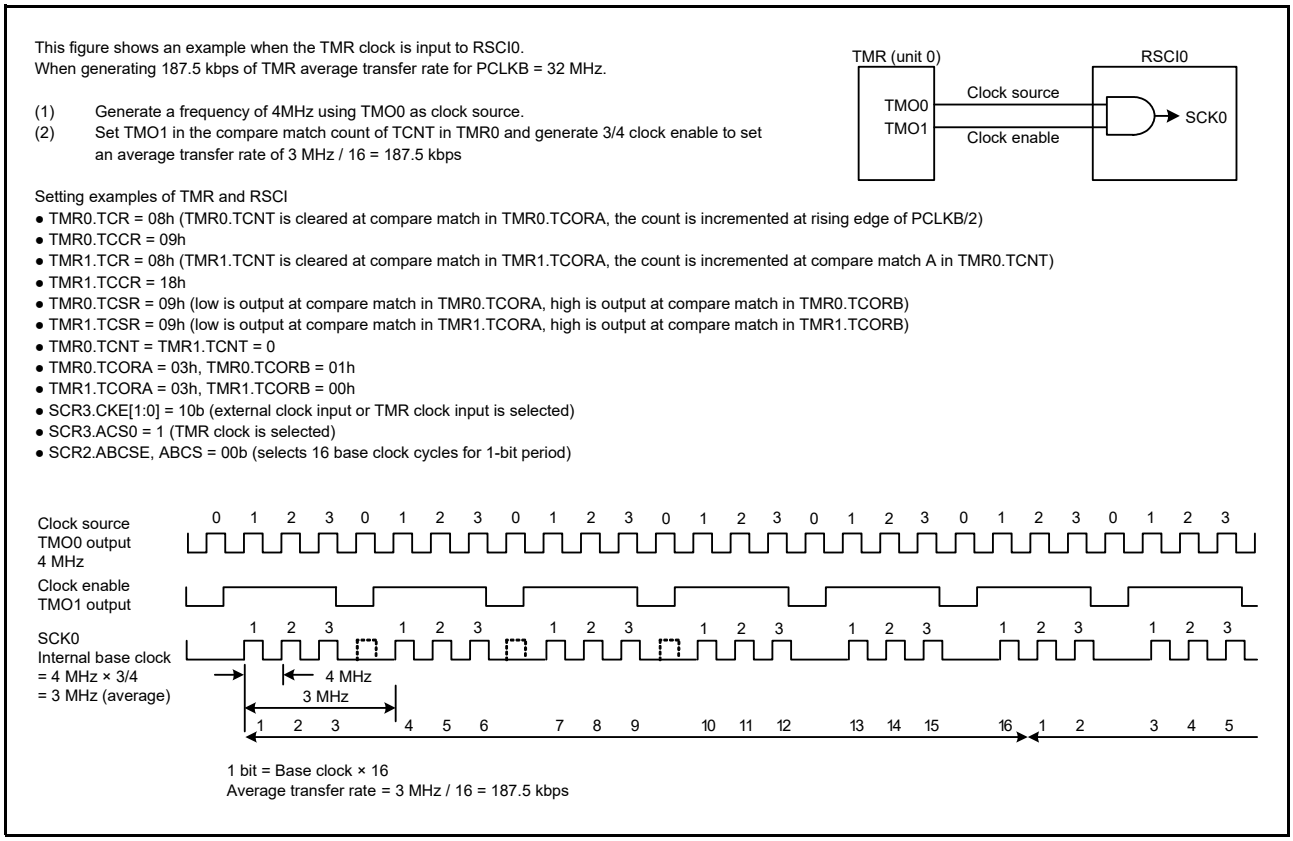


Figure 42.3 Example of Average Transfer Rate Setting When TMR Clock is Input.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 42.7.6, Serial Data Transmission (Except in Block Transfer Mode) and section 42.7.8, Clock Output Control.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 42.7.3, Block Transfer Mode.

42.2.9 Control Register 4 (SCR4)

Address(es): RSCI0.SCR4 8745 8018h, RSCI1.SCR4 8745 8118h, RSCI2.SCR4 8745 8218h, RSCI3.SCR4 8745 8318h, RSCI4.SCR4 8745 8418h, RSCI5.SCR4 8745 8518h, RSCI6.SCR4 8745 8618h, RSCI7.SCR4 8745 8718h, RSCI8.SCR4 8745 8818h, RSCI9.SCR4 8745 8918h, RSCI10.SCR4 8745 8A18h, RSCI11.SCR4 8745 8B18h, RSCI12.SCR4 8745 8C18h,
RSCI0_NS.SCR4 0745 8018h, RSCI1_NS.SCR4 0745 8118h, RSCI2_NS.SCR4 0745 8218h, RSCI3_NS.SCR4 0745 8318h, RSCI4_NS.SCR4 0745 8418h, RSCI5_NS.SCR4 0745 8518h, RSCI6_NS.SCR4 0745 8618h, RSCI7_NS.SCR4 0745 8718h, RSCI8_NS.SCR4 0745 8818h, RSCI9_NS.SCR4 0745 8918h, RSCI10_NS.SCR4 0745 8A18h, RSCI11_NS.SCR4 0745 8B18h, RSCI12_NS.SCR4 0745 8C18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TTMG[3:0]				RTMG[3:0]				—	—	—	—	—	—	TTADJ	RTADJ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	CMPD[8:0]	Compare Match Data	(Valid only in asynchronous mode) Set the compared data when using data match detection function	R/W *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RTADJ	Receive Data Sampling Timing Adjustment	(Valid in asynchronous mode using internal clock and extended serial mode using internal clock) 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W *1
b17	TTADJ	Transmit Signal Transition Timing Adjustment	(Valid only in asynchronous mode using internal clock) 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W *1
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27 to b24	RTMG[3:0]	Receive Data Sampling Timing Select	b27 b24 1 1 1 1: Data are sampled 7 clocks earlier than default point. 1 1 1 0: Data are sampled 6 clocks earlier than default point. 1 1 0 1: Data are sampled 5 clocks earlier than default point. 1 1 0 0: Data are sampled 4 clocks earlier than default point. 1 0 1 1: Data are sampled 3 clocks earlier than default point. 1 0 1 0: Data are sampled 2 clocks earlier than default point. 1 0 0 1: Data are sampled 1 clocks earlier than default point. x 0 0 0: Data are sampled at default point. 0 0 0 1: Data are sampled 1 clock later than default point. 0 0 1 0: Data are sampled 2 clock later than default point. 0 0 1 1: Data are sampled 3 clock later than default point. 0 1 0 0: Data are sampled 4 clock later than default point. 0 1 0 1: Data are sampled 5 clock later than default point. 0 1 1 0: Data are sampled 6 clock later than default point. 0 1 1 1: Data are sampled 7 clock later than default point.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	TTMG[3:0]	Transmit Signal Transition Timing Select	b31 b28 1 1 1 1: Delays the 1 to 0 transitions for 7 clocks. 1 1 1 0: Delays the 1 to 0 transitions for 6 clocks. 1 1 0 1: Delays the 1 to 0 transitions for 5 clocks. 1 1 0 0: Delays the 1 to 0 transitions for 4 clocks. 1 0 1 1: Delays the 1 to 0 transitions for 3 clocks. 1 0 1 0: Delays the 1 to 0 transitions for 2 clocks. 1 0 0 1: Delays the 1 to 0 transitions for 1 clocks. x 0 0 0: Does not change the waveform. 0 0 0 1: Delays the 0 to 1 transitions for 1 clock. 0 0 1 0: Delays the 0 to 1 transitions for 2 clock. 0 0 1 1: Delays the 0 to 1 transitions for 3 clock. 0 1 0 0: Delays the 0 to 1 transitions for 4 clock. 0 1 0 1: Delays the 0 to 1 transitions for 5 clock. 0 1 1 0: Delays the 0 to 1 transitions for 6 clock. 0 1 1 1: Delays the 0 to 1 transitions for 7 clock.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CMPD[8:0] Bits (Compare Match Data)

Set the comparison data for receive data, when data match detection function is enabled (SCR0.DCME bit = 1). SCR4.CMPD[8:0] bits should be written while SCR0.DCME bit is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7bit length enable, CMPD[7:0] with 8bit, and CMPD[8:0] with 9bit length.

RTADJ Bit (Receive Data Sampling Timing Adjustment)

When this bit is 1, the receive sampling timing adjustment function is enabled.

In asynchronous mode using internal clock, refer to section 42.3.10, Receive Data Sampling Timing Adjustment for details. The operation when the extended serial mode internal clock is selected is the same as when the asynchronous clock internal clock is selected.

TTADJ Bit (Transmit Signal Transition Timing Adjustment)

When this bit is 1, the transmit signal transition timing adjustment function is enabled. The transmit signal transition timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. Refer to section 42.3.11, Transmit Data Transition Timing Adjustment for details.

RTMG[3:0] Bits (Receive Data Sampling Timing Select)

When the RTADJ bit is 1, the receive sampling timing can be adjusted according to this bit setting value. The adjustment value is the base clock × RTMG[2:0] setting value.

TTMG[3:0] Bits (Transmit Signal Transition Timing Select)

The edge timing of the TXDn pin specified by the TTMG[3:0] bits is adjusted by the base clock × TTMG[2:0] setting value. Make sure that the TTMG[2:0] bit setting is less than the number of base clock cycles for 1-bit period.

42.2.10 I²C Mode Register (SIMR)

Address(es): RSCI0.SIMR 8745 8020h, RSCI1.SIMR 8745 8120h, RSCI2.SIMR 8745 8220h, RSCI3.SIMR 8745 8320h, RSCI4.SIMR 8745 8420h, RSCI5.SIMR 8745 8520h, RSCI6.SIMR 8745 8620h, RSCI7.SIMR 8745 8720h, RSCI8.SIMR 8745 8820h, RSCI9.SIMR 8745 8920h, RSCI10.SIMR 8745 8A20h, RSCI11.SIMR 8745 8B20h, RSCI12.SIMR 8745 8C20h, RSCI0_NS.SIMR 0745 8020h, RSCI1_NS.SIMR 0745 8120h, RSCI2_NS.SIMR 0745 8220h, RSCI3_NS.SIMR 0745 8320h, RSCI4_NS.SIMR 0745 8420h, RSCI5_NS.SIMR 0745 8520h, RSCI6_NS.SIMR 0745 8620h, RSCI7_NS.SIMR 0745 8720h, RSCI8_NS.SIMR 0745 8820h, RSCI9_NS.SIMR 0745 8920h, RSCI10_NS.SIMR 0745 8A20h, RSCI11_NS.SIMR 0745 8B20h, RSCI12_NS.SIMR 0745 8C20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IICDL[4:0]	SDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b4 b0 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts	R/W *1
b9	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W *1
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*2, *4, *5, *6	R/W
b17	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*3, *4, *5, *6	R/W
b18	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*3, *4, *5, *6	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b21, b20	IICSDAS[1:0]	SDA Output Select	b21 b20 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b23, b22	IICSCLS[1:0]	SCL Output Select	b23 b22 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

Note 2. In the bus free state, perform the start condition generation.
Note 3. In the bus busy state, perform restart or stop condition generation when the SSCLn pin after acknowledgment described in Figure 42.79 and Figure 42.80 is low level.
Note 4. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] Bits (SDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing TCLK by the divisor set in SCR2.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator.
Set these bits to 00000b unless operation is in simple I^2C mode. In simple I^2C mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM Bit (I^2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I^2C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCL signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.
The internal SCL signal is not synchronized if the IICCSC bit is 0. The internal SCL signal is generated in accordance with the rate selected in the BRR[7:0] bits regardless of the level being input on the SSCLn pin.
Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.
If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt request output.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSDAS[1:0] Bits (SDA Output Select)

These bits control output from the SSDAn pin.

IICSCLS[1:0] Bits (SCL Output Select)

These bits control output from the SSCLn pin.

42.2.11 FIFO Control Register (FCR)

Address(es): RSCI0.FCR 8745 8024h, RSCI1.FCR 8745 8124h, RSCI2.FCR 8745 8224h, RSCI3.FCR 8745 8324h, RSCI4.FCR 8745 8424h, RSCI5.FCR 8745 8524h, RSCI6.FCR 8745 8624h, RSCI7.FCR 8745 8724h, RSCI8.FCR 8745 8824h, RSCI9.FCR 8745 8924h, RSCI10.FCR 8745 8A24h, RSCI11.FCR 8745 8B24h, RSCI12.FCR 8745 8C24h,
RSCI0_NS.FCR 0745 8024h, RSCI1_NS.FCR 0745 8124h, RSCI2_NS.FCR 0745 8224h, RSCI3_NS.FCR 0745 8324h, RSCI4_NS.FCR 0745 8424h, RSCI5_NS.FCR 0745 8524h, RSCI6_NS.FCR 0745 8624h, RSCI7_NS.FCR 0745 8724h, RSCI8_NS.FCR 0745 8824h, RSCI9_NS.FCR 0745 8924h, RSCI10_NS.FCR 0745 8A24h, RSCI11_NS.FCR 0745 8B24h, RSCI12_NS.FCR 0745 8C24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRES	Receive Data Ready Interrupt Select	(Valid in asynchronous mode) This bit select the interrupt request for a reception data ready detection. 0: Reception data full interrupt (RXI) 1: Receive error interrupt (ERI)	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	TTRG[4:0]	Transmit FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b12 b8 0 0 0 0 0: Trigger number 0 : : 0 1 1 1 1: Trigger number 15 Settings other than above are prohibited.	R/W *1
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	TFRST	Transmit FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in transmit FIFO (TDR register) are made 0 The read value is always 0.	W*1
b20 to b16	RTRG[4:0]	Receive FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b20 b16 0 0 0 0 0: Trigger number 0 : : 0 1 1 1 1: Trigger number 15 Settings other than above are prohibited.	R/W *1
b22, b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23	RFRST	Receive FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in receive FIFO (RDR register) are made 0 The read value is always 0.	W*1
b28 to b24	RSTRG[4:0]	RTS# Output Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode) This bit enables only when SCR3.FM bit = 1, SCR1.CTSE bit = 0, and SCR0.SSE bit = 0. b28 b24 0 0 0 0 0: Trigger number 0 : : 0 1 1 1 1: Trigger number 15 Settings other than above are prohibited.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DRES Bit (Receive Data Ready Interrupt Select)

Select whether the detection of receive data ready (RFSR.DR flag = 1) is the cause of RXI interrupt request or the cause of ERI interrupt request.

TTRG[4:0] Bits (Transmit FIFO Threshold Setting)

The TDRE flag is set to 1 when the quantity of transmit data in the transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number. If SCR0.TIE bit = 1, TXI interrupt request is occurred.

TFRST Bit (Transmit FIFO Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in transmit FIFO (TDR register) is made 0.

RTRG[4:0] Bits (Receive FIFO Threshold Setting)

The SSR.RDRF flag is set to 1 when the quantity of receive data in the receive FIFO (RDR register) is equal to or greater than the specified receive triggering number. If SCR0.RIE bit = 1, RXI interrupt request is occurred. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

RFRST Bit (Receive FIFO Reset)

When the RFRST bit is set to 1, the number of the reception data stored in receive FIFO (RDR register) is made 0.

RSTRG[4:0] Bits (RTS# Output Threshold Setting)

When the quantity of receive data stored in the receive FIFO (RDR register) is equal to or greater than this number, the RTSn# signal is in the High state. When FCR.RSTRG[4:0] bits are set to 0, RTSn# is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

42.2.12 Manchester Mode Control Register (MMCR)

Address(es): RSCI0.MMCR 8745 802Ch, RSCI1.MMCR 8745 812Ch, RSCI2.MMCR 8745 822Ch, RSCI3.MMCR 8745 832Ch, RSCI4.MMCR 8745 842Ch, RSCI5.MMCR 8745 852Ch, RSCI6.MMCR 8745 862Ch, RSCI7.MMCR 8745 872Ch, RSCI8.MMCR 8745 882Ch, RSCI9.MMCR 8745 892Ch, RSCI10.MMCR 8745 8A2Ch, RSCI11.MMCR 8745 8B2Ch, RSCI12.MMCR 8745 8C2Ch,
RSCI0_NS.MMCR 0745 802Ch, RSCI1_NS.MMCR 0745 812Ch, RSCI2_NS.MMCR 0745 822Ch, RSCI3_NS.MMCR 0745 832Ch, RSCI4_NS.MMCR 0745 842Ch, RSCI5_NS.MMCR 0745 852Ch, RSCI6_NS.MMCR 0745 862Ch, RSCI7_NS.MMCR 0745 872Ch, RSCI8_NS.MMCR 0745 882Ch, RSCI9_NS.MMCR 0745 892Ch, RSCI10_NS.MMCR 0745 8A2Ch, RSCI11_NS.MMCR 0745 8B2Ch, RSCI12_NS.MMCR 0745 8C2Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SBERIE	SYERIE	PFERIE	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBLEN	SYNCE	SBPTN	—	SADJE	ENCS	DECS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECS	Decoding Convention Select	Sets the polarity of the received Manchester code 0: Low to high transition is decoded to a logic 0 and high to low transition is decoded to a logic 1. 1: high to low transition is decoded to a logic 0 and low to high transition is decoded to a logic 1.	R/W *1
b1	ENCS	Encoding Convention Select	Sets the polarity of the transmit Manchester code 0: Logic 0 is encoded to a low to high transition and logic 1 is encoded to a high to low transition. 1: Logic 0 is encoded to a high to low transition and logic 1 is encoded to a low to high transition.	R/W *1
b2	SADJE	Receive Timing Self Adjustment Enable	Sets the receive retiming function 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	SBPTN	Start Bit Pattern Select	Sets the Sync type of the start bit(s) in the Manchester code When the start bit area consists of one bit. (SBLEN bit = 0) • When transmitting 0: The start bit is added as a low to high transition. 1: The start bit is added as a high to low transition. • When receiving 0: Only when the start bit is a low to high transition, the data is received. The other cases are judged as an error. 1: Only when the start bit is a high to low transition, the data is received. The other cases are judged as an error. When the start bit area consists of three bits. (SBLEN bit = 1) • When transmitting 0: The start bits are added as a low to high transition. (DATA Sync) 1: The start bits are coded as a high to low transition. (COMMAND Sync) • When receiving When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W *1
b5	SYNCE	Sync Enable	0: The start bit pattern is set with the SBPTN bit. 1: The start bit pattern is set with the SYNC bit.	R/W *1
b6	SBLEN	Start Bit Length Select	0: The start bit area consists of one bit. 1: The start bit area consists of three bits. (COMMAND Sync or DATA Sync)	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b11 to b8	TPLEN[3:0]	Transmit Preface Length Setting	Set the preface length of the transmit data in Manchester mode 0: Disables the transmit preface generation 1 to 15: Transmit preface length (bit length)	R/W *1
b13, b12	TPPAT[1:0]	Transmit Preface Pattern Select	Set the preface pattern of the transmit data b13 b12 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b19 to b16	RPLEN[3:0]	Receive Preface Length Setting	Set the preface length in received frames when Manchester mode is enabled 0: Disables the receive preface generation 1 to 15: Receive preface length (bit length)	R/W *1
b21, b20	RPPAT[1:0]	Receive Preface Pattern Select	Set the preface pattern of received frames b21 b20 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PFERIE	Preface Error Interrupt Enable	Specifies whether to handle a preface error as an interrupt source 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W *1
b25	SYERIE	Sync Error Interrupt Enable	Specifies whether to handle a receive Sync error as an interrupt source 0: Does not handle a receive Sync error as an interrupt source 1: Handles a receive Sync error as an interrupt source	R/W *1
b26	SBERIE	Start Bit Error Interrupt Enable	Specifies whether to handle a start bit error as an interrupt source 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W *1
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DECS Bit (Decoding Convention Select)

This bit sets the polarity of the received Manchester code. For details on the data reception, see section 42.5.7, Serial Data Reception in Manchester Mode.

ENCS Bit (Encoding Convention Select)

This bit sets the polarity of the transmit Manchester code. For details on the data transmission, see section 42.5.6, Serial Data Transmission in Manchester Mode.

SADJE Bit (Receive Timing Self Adjustment Enable)

This bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see section 42.5.9, Receive Retiming.

SBPTN Bit (Start Bit Pattern Select)

This bit is valid when the SYNCE bit is set to 0.

The Sync type can be set by combining this bit and the SBLEN bit.

For the start bit area determined by the combination of this bit and the SBLEN bit, see Figure 42.37 and Figure 42.38.

SYNCE Bit (Sync Enable)

This bit is valid when the SBLEN bit is set to 1. This bit determines the destination to be referred to for setting the Sync type of the start bit area added to Manchester frames.
When this bit is set to 0, the SBPTN bit is referred to.
When this bit is set to 1, the TDR.SYNC bit is referred to.

SBLEN Bit (Start Bit Length Select)

This bit sets the start bit area in Manchester frames.
When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNCE and SBPTN bits are valid.
When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] Bits (Transmit Preface Length Setting)

These bits set the preface bit length of the transmit data in Manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the transmit preface, which is not added.

TPPAT[1:0] Bits (Transmit Preface Pattern Select)

These bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 42.36.
When these bits are set to 00b, the preface area is set to all zeros.
When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is set to all ones.

RPLEN[3:0] Bits (Receive Preface Length Setting)

These bits set the preface bit length of the received frames in Manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the receive preface, which is not added. When 1h to Fh is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] Bits (Receive Preface Pattern Select)

These bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the TPPAT[1:0] bits are set, see Figure 42.36.
When these bits are set to 00b, the preface area is handled as all zeros.
When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is handled as all ones.

PFERIE Bit (Preface Error Interrupt Enable)

This bit specifies whether to handle a preface error as an interrupt source.
When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYERIE Bit (Sync Error Interrupt Enable)

This bit specifies whether to handle a receive Sync error as an interrupt source.
When it is set to 0, a receive Sync error is not handled as an interrupt source. When it is set to 1, a receive Sync error is handled as an interrupt source.

SBERIE Bit (Start Bit Error Interrupt Enable)

This bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

42.2.13 DE Signal Control Register (DECR)

Address(es): RSCI0.DECR 8745 8030h, RSCI1.DECR 8745 8130h, RSCI2.DECR 8745 8230h, RSCI3.DECR 8745 8330h, RSCI4.DECR 8745 8430h, RSCI5.DECR 8745 8530h, RSCI6.DECR 8745 8630h, RSCI7.DECR 8745 8730h, RSCI8.DECR 8745 8830h, RSCI9.DECR 8745 8930h, RSCI10.DECR 8745 8A30h, RSCI11.DECR 8745 8B30h, RSCI12.DECR 8745 8C30h,
RSCI0_NS.DECR 0745 8030h, RSCI1_NS.DECR 0745 8130h, RSCI2_NS.DECR 0745 8230h, RSCI3_NS.DECR 0745 8330h, RSCI4_NS.DECR 0745 8430h, RSCI5_NS.DECR 0745 8530h, RSCI6_NS.DECR 0745 8630h, RSCI7_NS.DECR 0745 8730h, RSCI8_NS.DECR 0745 8830h, RSCI9_NS.DECR 0745 8930h, RSCI10_NS.DECR 0745 8A30h, RSCI11_NS.DECR 0745 8B30h, RSCI12_NS.DECR 0745 8C30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	DEHLD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DESU[4:0]					—	—	—	—	—	—	—	DELVL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DELVL	DE Signal Active Level Select	(Valid only in asynchronous mode) 0: The DE signal is active high. 1: The DE signal is active low.	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	DESU[4:0]	DE Signal Setup Time Setting	(Valid only in asynchronous mode) Set the DE signal setup time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20 to b16	DEHLD[4:0]	DE Signal Hold Time Setting	(Valid only in asynchronous mode) Set the DE signal hold time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DELVL Bit (DE Signal Active Level Select)

Select the active level of the DE (driver enable) signal.

DESU[4:0] Bits (DE Signal Setup Time Setting)

Set the DE signal setup time (time from the assertion of the DE signal to the start of transmission of the start bit). It is expressed in the number of the base clock cycles (1/4, 1/6, 1/8, or 1/16 bit period). The actual transmission of the start bit starts after the setup time and transmission wait time have elapsed.

DEHLD[4:0] Bits (DE Signal Hold Time Setting)

Set the DE signal hold time (time from the completion of transmission of the stop bit of the last transmission message to negation of the DE signal). It is expressed in the number of the base clock cycles (1/4, 1/6, 1/8, or 1/16 bit period).
If the transmission data is written during the hold time, transmit starting operation is different depends on the writing timing (following two cases: the transmission of the start bit starts after the transmission wait time has elapsed without negating the DE signal, or it starts after the DE signal is negated and asserted again and then the setup time and transmission wait time have elapsed.).

42.2.14 Extended Serial Mode Control Register 0 (XCR0)

Address(es): RSCI0.XCR0 8745 8034h, RSCI1.XCR0 8745 8134h, RSCI2.XCR0 8745 8234h, RSCI3.XCR0 8745 8334h, RSCI4.XCR0 8745 8434h, RSCI5.XCR0 8745 8534h, RSCI6.XCR0 8745 8634h, RSCI7.XCR0 8745 8734h, RSCI8.XCR0 8745 8834h, RSCI9.XCR0 8745 8934h, RSCI10.XCR0 8745 8A34h, RSCI11.XCR0 8745 8B34h, RSCI12.XCR0 8745 8C34h,
RSCI0_NS.XCR0 0745 8034h, RSCI1_NS.XCR0 0745 8134h, RSCI2_NS.XCR0 0745 8234h, RSCI3_NS.XCR0 0745 8334h, RSCI4_NS.XCR0 0745 8434h, RSCI5_NS.XCR0 0745 8534h, RSCI6_NS.XCR0 0745 8634h, RSCI7_NS.XCR0 0745 8734h, RSCI8_NS.XCR0 0745 8834h, RSCI9_NS.XCR0 0745 8934h, RSCI10_NS.XCR0 0745 8A34h, RSCI11_NS.XCR0 0745 8B34h, RSCI12_NS.XCR0 0745 8C34h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	BCCS[1:0]		—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCSS[1:0]	Timer Count Clock Source Select	(Valid in extended serial mode) Select the clock source of the timer in the extended serial module. b1 b0 0 0: TCLK 0 1: TCLK/4 1 0: TCLK/16 1 1: TCLK/64	R/W *1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BFE	Break Field Detection Enable	Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W *1, *3
b9	CF0RE	Control Field 0 Reception Enable	Set the presence or absence of Control Field 0 of Start Frame. 0: No Control Field 0 1: With Control Field 0	R/W *1, *3
b11, b10	CF1DS[1:0]	Control Field 1 Compare Data Select	Select the compare data for Control Field 1 b11 b10 0 0: Select XCR1.PCF1D[7:0] bits as the compare data 0 1: Select XCR1.SCF1D[7:0] bits as the compare data 1 0: Select both XCR1.PCF1D[7:0] bits and XCR1.SCF1D[7:0] bits as the compare data 1 1: Prohibition	R/W *1, *3
b12	PIBE	Priority Interrupt Bit Enable	0: Priority interrupt bit disable 1: Priority interrupt bit enable	R/W *1, *3
b15 to b13	PIBS[2:0]	Priority Interrupt Bit Select	Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. b15 b13 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W *1, *3
b16	BFOIE	Break Field Low Width Output Complete Interrupt Enable	Select whether to include Break Field transmission completion as a TXI interrupt factor. 0: Break Field transmission completion is not included in TXI interrupt factor 1: Break Field transmission completion is included in TXI interrupt factor	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b17	BCDIE	Bus Collision Detected Interrupt Enable	Select whether to output an ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in ERI interrupt factor 1: Bus conflict detection is included in ERI interrupt factor	R/W *1
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	BFDIE	Break Field Low Width Detected Interrupt Enable	Select whether to output a BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disable 1: Break Field detection interrupt enable	R/W *1
b21	COFIE	Count Overflow Interrupt Enable	Select whether to include counter overflow as an ERI interrupt factor. 0: Counter overflow is not included in ERI interrupt factor 1: Counter overflow is included in ERI interrupt factor	R/W *1
b22	AEDIE	Effective Edge Detected Interrupt Enable	Select whether to output an AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disable 1: Active edge detection interrupt enable	R/W *1
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b25, b24	BCCS[1:0]	Bus Collision Detection Clock Select	Select the sampling clock for the bus conflict detection circuit. When SCR2.ABCS bit = 1, setting BCCS[1:0] bits = 1xb is prohibited. b25 b24 0 0: Base clock*4 0 1: Base clock/2 1 0: Base clock/4 1 1: Prohibition	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. Rewrite the TCSS[1:0] bits only when the timer is stopped (XCR1.TCST bit = 0, XCR1.SDST bit = 0, and XCR1.BRME bit = 0).
Note 3. This bit is a setting bit required for Start Frame reception operation. Rewrite this bit when Start Frame reception or transmission is not in progress (XCR1.SDST bit = 0 and XCR1.TCST bit = 0).
Note 4. Base clock: 1/16 period of 1 bit period when SCR2.ABCS bit = 0, 1/8 period of 1 bit period when SCR2.ABCS bit = 1.

TCSS[1:0] Bits (Timer Count Clock Source Select)

Select clock source of timer in extended serial module.

BFE Bit (Break Field Detection Enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE Bit (Control Field 0 Reception Enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] Bits (Control Field 1 Compare Data Select)

Select the compare data for Control Field 1.

PIBE Bit (Priority Interrupt Bit Enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When this bit is 1, regardless of the XCR1.CF1CE[7:0] bit setting value, the bit specified in the PIBS[2:0] bits is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D[7:0] bits).

PIBS[2:0] Bits (Priority Interrupt Bit Select)

Specify bit N (N = 0 to 7) of Control Field 1 as the priority interrupt bit.

BFOIE Bit (Break Field Low Width Output Complete Interrupt Enable)

Select whether to include Break Field transmission completion as a TXI interrupt factor. Set SCR0.TIE bit = 1 and SCR3.MOD[2:0] bits = 110b, to output TXI upon completion of Break Field transmission.

BCDIE Bit (Bus Collision Detected Interrupt Enable)

Select whether to output an ERI interrupt when a bus collision is detected. In extended serial mode (SCR3.MOD[2:0] bits = 110b), ERI output control is performed with this bit. When SCR3.MOD[2:0] bits = 110b and BCDIE bit = 1, an ERI interrupt is issued when a bus collision is detected even if SCR0.RIE bit = 0.

COFIE Bit (Count Overflow Interrupt Enable)

Select whether to include counter overflow as an ERI interrupt factor. Set SCR0.RIE bit = 1 and SCR3.MOD[2:0] bits = 110b are required to output ERI upon counter overflow.

AEDIE Bit (Effective Edge Detected Interrupt Enable)

Select whether to output an AED interrupt when a valid edge is detected. To output AED with valid edge detection, XCR1.BRME bit = 1 and SCR3.MOD[2:0] bits = 110b must be set.

BCCS[1:0] Bits (Bus Collision Detection Clock Select)

Select the sampling clock for the bus conflict detection circuit.

42.2.15 Extended Serial Mode Control Register 1 (XCR1)

Address(es): RSCI0.XCR1 8745 8038h, RSCI1.XCR1 8745 8138h, RSCI2.XCR1 8745 8238h, RSCI3.XCR1 8745 8338h, RSCI4.XCR1 8745 8438h, RSCI5.XCR1 8745 8538h, RSCI6.XCR1 8745 8638h, RSCI7.XCR1 8745 8738h, RSCI8.XCR1 8745 8838h, RSCI9.XCR1 8745 8938h, RSCI10.XCR1 8745 8A38h, RSCI11.XCR1 8745 8B38h, RSCI12.XCR1 8745 8C38h,
RSCI0_NS.XCR1 0745 8038h, RSCI1_NS.XCR1 0745 8138h, RSCI2_NS.XCR1 0745 8238h, RSCI3_NS.XCR1 0745 8338h, RSCI4_NS.XCR1 0745 8438h, RSCI5_NS.XCR1 0745 8538h, RSCI6_NS.XCR1 0745 8638h, RSCI7_NS.XCR1 0745 8738h, RSCI8_NS.XCR1 0745 8838h, RSCI9_NS.XCR1 0745 8938h, RSCI10_NS.XCR1 0745 8A38h, RSCI11_NS.XCR1 0745 8B38h, RSCI12_NS.XCR1 0745 8C38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCF1D[7:0]								—	—	BRME	SDST	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Break Field Low Width Output Timer Count Start	0: Break Field transmission timer count stopped 1: Break Field transmission timer count start Do not set this bit and SDST bit to 1 at the same time.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SDST	Start Frame Detection Start	0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W *1
b5	BRME	Bit Rate Measurement Enable	0: Bit rate measurement disabled 1: Bit rate measurement enabled Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 0, it can be set to 0 at any timing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15 to b8	PCF1D[7:0]	Primary Control Field 1 Compare Data	The priority compare data for Control Field 1	R/W *1
b23 to b16	SCF1D[7:0]	Secondary Control Field 1 Compare Data	The secondary compare data for Control Field 1	R/W *1
b31 to b24	CF1CE[7:0]	Control Field 1 Compare Enable	Select whether to compare bit N of Control Field 1. (N = 0 to 7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

TCST Bit (Break Field Low Width Output Timer Count Start)

[Clearing condition]

- When 0 is written to TCST bit. Break Field transmission timer count is stopped and TXDn pin output becomes idle level.
- When Break Field transmission for the period set in XCR2.BFLW[15:0] bits is completed.

[Setting condition]

- When 1 is written to TCST bit. Start Break Field transmission from TXDn pin. Holds 1 during Break Field transmission.

SDST Bit (Start Frame Detection Start)

When 1 is written to this bit, Start Frame detection starts. When XCR0.BFE bit = 1 is set, Break Field can be detected during Start Frame is detected and after Start frame is detected. When XCR0.BFE bit = 0 is set, Break Field is not

detected.
When 0 is written to this bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF flag = 0 at the time of stop, it is not possible to stop data reception with this bit. Write 0 to SCR0.RE bit to stop the reception operation or perform reception completion processing (clearing the SSR.RDRF flag or reading the RDR register) after reception is completed.

BRME Bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] Bits (Primary Control Field 1 Compare Data)

Set the priority compare data for Control Field 1.

SCF1D[7:0] Bits (Secondary Control Field 1 Compare Data)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] Bits (Control Field 1 Compare Enable)

Select whether to compare bit N of Control Field 1. (N = 0 to 7)
When all of these bits are set to 0 (CF1CE[7:0] bits = 00h), it is always judged that Control Field 1 matches when reception is completed, and XSR0.CF1MF flag is set. This bit is a comparison enable with PCF1D[7:0] bits or SCF1D[7:0] bits, and it is not a priority interrupt bit comparison enable.

42.2.16 Extended Serial Mode Control Register 2 (XCR2)

Address(es): RSCI0.XCR2 8745 803Ch, RSCI1.XCR2 8745 813Ch, RSCI2.XCR2 8745 823Ch, RSCI3.XCR2 8745 833Ch, RSCI4.XCR2 8745 843Ch, RSCI5.XCR2 8745 853Ch, RSCI6.XCR2 8745 863Ch, RSCI7.XCR2 8745 873Ch, RSCI8.XCR2 8745 883Ch, RSCI9.XCR2 8745 893Ch, RSCI10.XCR2 8745 8A3Ch, RSCI11.XCR2 8745 8B3Ch, RSCI12.XCR2 8745 8C3Ch,
RSCI0_NS.XCR2 0745 803Ch, RSCI1_NS.XCR2 0745 813Ch, RSCI2_NS.XCR2 0745 823Ch, RSCI3_NS.XCR2 0745 833Ch, RSCI4_NS.XCR2 0745 843Ch, RSCI5_NS.XCR2 0745 853Ch, RSCI6_NS.XCR2 0745 863Ch, RSCI7_NS.XCR2 0745 873Ch, RSCI8_NS.XCR2 0745 883Ch, RSCI9_NS.XCR2 0745 893Ch, RSCI10_NS.XCR2 0745 8A3Ch, RSCI11_NS.XCR2 0745 8B3Ch, RSCI12_NS.XCR2 0745 8C3Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CF0D[7:0]	Control Field Compare Data	The compare data for Control Field 0	R/W *1
b15 to b8	CF0CE[7:0]	Control Field Compare Enable	Select whether to compare bit N of Control Field 0. (N = 0 to 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W *1
b31 to b16	BFLW[15:0]	Break Field Low Width Setting	This bit sets the Break Field length. The break field length is (BFLW[15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is FFFEh. (Setting prohibited for FFFFh)	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CF0D[7:0] Bits (Control Field Compare Data)

The compare data for Control Field 0.

CF0CE[7:0] Bits (Control Field Compare Enable)

Select whether to compare bit N of Control Field 0. (N = 0 to 7)

When all of these bits are set to 0 (CF0CE[7:0] bits = 00h), it is always judged that Control Field 0 matches when reception is completed, and the XSR0.CF0MF flag is set.

BFLW[15:0] Bits (Break Field Low Width Setting)

The BFLW[15:0] bits are 16-bit Break Field length setting bits and the initial value is FFFEh.

Set the break field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When sending the Break Field, writing 1 to the TCST bit leads RSCI to output the Break Field on the TXDn pin. At the same time, the timer starts counting with the timer count clock source selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, up-counting is stopped and Break Field transmission from the TXDn pin is also stopped.

When detecting the Break Field, writing 1 to the SDST bit leads Start Frame detection to be enabled. RSCI starts counting from the negative edge of the RXDn signal. The timer count clock source is selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, it is determined that a break field has been detected. Up-counting continues until the next valid edge or counter overflow.

42.2.17 Status Register (SSR)

Address(es): RSCI0.SSR 8745 8048h, RSCI1.SSR 8745 8148h, RSCI2.SSR 8745 8248h, RSCI3.SSR 8745 8348h, RSCI4.SSR 8745 8448h, RSCI5.SSR 8745 8548h, RSCI6.SSR 8745 8648h, RSCI7.SSR 8745 8748h, RSCI8.SSR 8745 8848h, RSCI9.SSR 8745 8948h, RSCI10.SSR 8745 8A48h, RSCI11.SSR 8745 8B48h, RSCI12.SSR 8745 8C48h,
RSCI0_NS.SSR 0745 8048h, RSCI1_NS.SSR 0745 8148h, RSCI2_NS.SSR 0745 8248h, RSCI3_NS.SSR 0745 8348h, RSCI4_NS.SSR 0745 8448h, RSCI5_NS.SSR 0745 8548h, RSCI6_NS.SSR 0745 8648h, RSCI7_NS.SSR 0745 8748h, RSCI8_NS.SSR 0745 8848h, RSCI9_NS.SSR 0745 8948h, RSCI10_NS.SSR 0745 8A48h, RSCI11_NS.SSR 0745 8B48h, RSCI12_NS.SSR 0745 8C48h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRF	TEND	TDRE	AFER	APER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b4	ERS	Error Signal Status Flag	(Valid only in Smart card interface mode) 0: Error signal Low not responded 1: Error signal Low responded	R
b14 to b5	—	Reserved	These bits are read as 0.	R
b15	RXDMON	RXD Line Monitoring Flag	The state of the RXDn pin is shown. When RINV is 0, 0: RXDn pin is the Low level. 1: RXDn pin is the High level. When RINV is 1, 0: RXDn pin is the High level. 1: RXDn pin is the Low level.	R
b16	DCMF	Data Match Flag	(Valid only in asynchronous mode) 0: No matched 1: Matched	R
b17	DPER	Matched Data Parity Error Flag	(Valid only in asynchronous mode) 0: No parity error occurred at data match detection 1: A parity error has occurred at data match detection	R
b18	DFER	Matched Data Framing Error Flag	(Valid only in asynchronous mode) 0: No framing error occurred at data match detection 1: A framing error has occurred at data match detection	R
b23 to b19	—	Reserved	These bits are read as 0.	R
b24	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R
b25	—	Reserved	This bit is read as 0.	R
b26	MFF	Mode Fault Flag	(Valid only in simple SPI mode.) 0: No mode fault 1: Mode fault	R
b27	APER	Aggregate Parity Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No parity error occurred 1: A parity error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No parity error in all received data in receive FIFO 1: One or more parity errors occurred in received data in receive FIFO	R

Bit	Symbol	Bit Name	Description	R/W
b28	AFER	Aggregate Framing Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No framing error occurred 1: A framing error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No framing error in all received data in receive FIFO 1: One or more framing errors occurred in received data in receive FIFO	R
b29	TDRE	Transmit Data Empty Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: Transmit data is in TDR register 1: No transmit data is in TDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of transmit data written in transmit FIFO exceeds the specified transmit triggering number. 1: The quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number.	R
b30	TEND	Transmit End Flag	0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
b31	RDRF	Receive Data Full Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No received data is in RDR register 1: Received data is in RDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of receive data written in receive FIFO falls below the specified receive triggering number. 1: The quantity of receive data written in receive FIFO is equal to or greater than the specified receive triggering number.	R

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When an error signal Low is sampled.

[Clearing condition]

- When write 1 to SSCR.ERSC bit.

DCMF Flag (Data Match Flag)

Indicates that RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data.

Clearing the SCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (SCR4.CMPD[8:0] bits) with receive data, while SCR0.DCME bit = 1.

[Clearing condition]

- When write 1 to SSCR.DCMFC bit.

DPER Flag (Matched Data Parity Error Flag)

It indicates that a parity error occurred at data match detection.

Clearing the SCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error was detected by the frame in which an data match was detected.

[Clearing condition]

- When write 1 to SSCR.DPERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DPER flag to 0.

DFER Flag (Matched Data Framing Error Flag)

It indicates that a framing error occurred at data match detection.
Clearing the SCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.
[Setting condition]

- When a stop bit of the frame in which an data match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and doesn't check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to SSCR.DFERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DFER flag to 0.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In simple I^2C mode, this bit is not use.
[Setting condition with non-FIFO mode (SCR3.FM bit = 0)]

- When the next data is received before reading out RDR register with no error reception data stored in RDR register.

In RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data isn't forwarded to RDR register. Note that, in clock synchronous mode and simple SPI mode, serial reception will be stop.
[Setting condition with FIFO mode (SCR3.FM bit = 1)]

- When the next serial reception is completed while the receive FIFO is full of 16 receive data.

[Clearing condition]

- When write 1 to SSCR.ORERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the ORER flag to 0.

MFF Flag (Mode Fault Flag)

This bit indicates mode faults. In a multi-master configuration, determine the mode fault occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation (SCR3.CKE[1:0] bits = 00b or 01b) in simple SPI mode.

[Clearing condition]

- When write 1 to SSCR.MFFC bit.

APER Flag (Aggregate Parity Error Flag)

Indicates that a parity error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the APER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when a parity error is detected during reception.
- In FIFO mode, when one or more parity error is detected for data in receive FIFO.

In non-FIFO mode, the received data when the parity error occurs is transferred to RDR register, but no RXI interrupt request occurs. Note that when the APER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
[Clearing condition]

- When write 1 to SSCR.APERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the APER flag to 0.

AFER Flag (Aggregate Framing Error Flag)

Indicates that a framing error has occurred during reception and the reception ends abnormally. Clearing the SCR0.RE bit to 0 does not affect the AFER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when 0 is sampled as the stop bit during reception.
- In FIFO mode, when one or more framing error is detected for data in receive FIFO.
- In Manchester mode, when both sampling results (1/4 and 3/4 sampling points) are not 1 for 1 stop bit.

In 2 stop bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked.
In non-FIFO mode, the received data when the framing error occurs is transferred to RDR register, but no RXI interrupt request occurs. In addition, when the AFER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
In extended serial mode, even if a condition that changes to 1 occurs when XCR1.SDST bit = 1, the AFER flag set timing is delayed up to the Break Field judgment timing at the longest, since it may be a Break Field. If an edge is detected on the RXD signal before the Break Field judgment timing, AFER is detected. If no edge is detected in the RXD signal before the Break Field judgment timing, Break Field is detected.
[Clearing condition]

- When write 1 to SSCR.AFERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the AFER flag to 0.

TDRE Flag (Transmit Data Empty Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)
Indicates the presence of transmit data in the TDR register.
The condition of SCR0.TE bit = 0 has priority over the condition of 0.
If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.
[Setting condition]

- When SCR0.TE bit is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to the TDR register during SCR0.TE bit = 1.

(2) FIFO selected (SCR3.FM bit = 1)
Indicates that data has been transferred from the transmit FIFO (TDR register) into the transmit shift register (TSR), the quantity of data in transmit FIFO has fallen below the specified transmit triggering number.
When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit FIFO is judged, and if that is equal to or less than TTRG[4:0] value, TDRE flag is set to 1 after 1 PCLK cycle.
[Setting condition]

- When the quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number*1.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to transmit FIFO by the DTC or DMAC (the last block transfer when block transfer).

Note 1. Because the transmit FIFO is a 16-stage FIFO register, the maximum number of data that can be written to the TDR register when the TDRE flag is 1 is “16 – TFSR.T[5:0]”. All other data written to the TDR register above that value is ignored.

TEND Flag (Transmit End Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0), and not smart card interface mode (SCR3.MOD[2:0] bits ≠ 001b)

Indicates completion of transmission.

[Setting condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).
- When Break Field is sending.

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the TEND flag to 0.

(2) Non-FIFO selected (SCR3.FM bit = 0), and smart card interface mode (SCR3.MOD[2:0] bits = 001b)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.
  The set timing is determined by register settings as listed below.
  When GM bit = 0 and BLK bit = 0, 12.5 etu after the start of transmission
  When GM bit = 0 and BLK bit = 1, 11.5 etu after the start of transmission
  When GM bit = 1 and BLK bit = 0, 11.0 etu after the start of transmission
  When GM bit = 1 and BLK bit = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

(3) FIFO selected (SCR3.FM bit = 1)

Indicates that the transmit FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).

[Clearing condition]

- When the transmission data was written to the TDR.TDAT[7:0] bits during SCR0.TE bit = 1.

RDRF Flag (Receive Data Full Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)

Indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the read data is read from the RDR register.

(2) FIFO selected (SCR3.FM bit = 1)

Indicates that receive data has been transferred to the receive FIFO (RDR register), and the quantity of data in receive FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

[Setting condition]

RDRF is set to 1 when the quantity of receive data in receive FIFO is equal to or greater than the specified receive triggering number*2.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the reception data is read form receive FIFO by the DTC or DMAC (the last block transfer when block transfer).

When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, RDRF flag will be 0. After that, when the number of stored data in receive FIFO is judged, and if that is equal to or greater than RTRG[4:0] value, RDRF flag is set to 1 after 1 PCLK cycle.

Note 2. Since the receive FIFO is a 16-stage FIFO register, the maximum number of data that can be read when RDRF is 1 is indicated by the RFSR.R[5:0] bits. After reading all the data in the RDR register, continuing a read access results in an undefined value.

Note: In non-FIFO mode, do not clear the RDRF and TDRE flags by the SSCR register except when the communication is to be aborted.

42.2.18 I²C Status Register (SISR)

Address(es): RSCI0.SISR 8745 804Ch, RSCI1.SISR 8745 814Ch, RSCI2.SISR 8745 824Ch, RSCI3.SISR 8745 834Ch, RSCI4.SISR 8745 844Ch, RSCI5.SISR 8745 854Ch, RSCI6.SISR 8745 864Ch, RSCI7.SISR 8745 874Ch, RSCI8.SISR 8745 884Ch, RSCI9.SISR 8745 894Ch, RSCI10.SISR 8745 8A4Ch, RSCI11.SISR 8745 8B4Ch, RSCI12.SISR 8745 8C4Ch, RSCI0_NS.SISR 0745 804Ch, RSCI1_NS.SISR 0745 814Ch, RSCI2_NS.SISR 0745 824Ch, RSCI3_NS.SISR 0745 834Ch, RSCI4_NS.SISR 0745 844Ch, RSCI5_NS.SISR 0745 854Ch, RSCI6_NS.SISR 0745 864Ch, RSCI7_NS.SISR 0745 874Ch, RSCI8_NS.SISR 0745 884Ch, RSCI9_NS.SISR 0745 894Ch, RSCI10_NS.SISR 0745 8A4Ch, RSCI11_NS.SISR 0745 8B4Ch, RSCI12_NS.SISR 0745 8C4Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R
b1	—	Reserved	This bit is read as 0.	R
b2	—	Reserved	The read value is undefined.	R
b3	IICSTIF	Condition Generation Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
b5, b4	—	Reserved	The read value is undefined.	R
b31 to b6	—	Reserved	These bits are read as 0.	R

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this flag.

The IICACKR flag is updated at the rising of SSCLn signal for the ACK/NACK receiving bit.

IICSTIF Flag (Condition Generation Completed Flag)

After generating a condition, this flag indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR0.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to SISCR.IICSTIFC bit
- When operation is not in simple I²C
- Writing 0 to SCR0.TE bit

42.2.19 Receive FIFO Status Register (RFSR)

Address(es): RSCI0.RFSR 8745 8050h, RSCI1.RFSR 8745 8150h, RSCI2.RFSR 8745 8250h, RSCI3.RFSR 8745 8350h, RSCI4.RFSR 8745 8450h, RSCI5.RFSR 8745 8550h, RSCI6.RFSR 8745 8650h, RSCI7.RFSR 8745 8750h, RSCI8.RFSR 8745 8850h, RSCI9.RFSR 8745 8950h, RSCI10.RFSR 8745 8A50h, RSCI11.RFSR 8745 8B50h, RSCI12.RFSR 8745 8C50h,
RSCI0_NS.RFSR 0745 8050h, RSCI1_NS.RFSR 0745 8150h, RSCI2_NS.RFSR 0745 8250h, RSCI3_NS.RFSR 0745 8350h, RSCI4_NS.RFSR 0745 8450h, RSCI5_NS.RFSR 0745 8550h, RSCI6_NS.RFSR 0745 8650h, RSCI7_NS.RFSR 0745 8750h, RSCI8_NS.RFSR 0745 8850h, RSCI9_NS.RFSR 0745 8950h, RSCI10_NS.RFSR 0745 8A50h, RSCI11_NS.RFSR 0745 8B50h, RSCI12_NS.RFSR 0745 8C50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FEC[5:0]						—	—	PEC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data has remained in receive FIFO after normally completed receiving. (receive FIFO is empty) 1: The following receive data does not come for a fixed period after storing data below the threshold in the receive FIFO	R
b1	—	Reserved	This bit is undefined when read.	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b13 to b8	R[5:0]	Receive FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of receive data stored in receive FIFO	R
b15, b14	—	Reserved	These bits are read as 0.	R
b21 to b16	PEC[5:0]	Parity Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a parity error among the receive data stored in the receive FIFO data register.	R
b23, b22	—	Reserved	These bits are read as 0.	R
b29 to b24	FEC[5:0]	Framing Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a framing error among the receive data stored in the receive FIFO data register.	R
b31, b30	—	Reserved	These bits are read as 0.	R

DR Flag (Receive Data Ready Flag)

Indicates that the quantity of data stored in the receive FIFO (RDR register) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This flag becomes 1 only when the FIFO buffer is enabled in synchronous mode (including multiprocessor mode), and does not become 1 in other modes.

[Setting conditions]

DR is set to 1 when the following conditions are met.

- After receive FIFO (RDR register) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
- SSR.AFER and APER flags are 0.

[Clearing conditions]

- When all receive data in the receive FIFO (RDR register) is read and 1 is written to RFSCR.DRC bit.

- When SCR3.FM bit is 0.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DR flag to 0 when the DR flag is set to the source of the receive error interrupt by setting the FCR.DRES bit to 1.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

R[5:0] Bits (Receive FIFO Data Count)

Indicate the quantity of receive data stored in receive FIFO.
00h means no receive data. 10h means receive FIFO is full.

PEC[5:0] Bits (Parity Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a parity error.

FEC[5:0] Bits (Framing Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a framing error.

42.2.20 Transmit FIFO Status Register (TFSR)

Address(es): RSCI0.TFSR 8745 8054h, RSCI1.TFSR 8745 8154h, RSCI2.TFSR 8745 8254h, RSCI3.TFSR 8745 8354h, RSCI4.TFSR 8745 8454h, RSCI5.TFSR 8745 8554h, RSCI6.TFSR 8745 8654h, RSCI7.TFSR 8745 8754h, RSCI8.TFSR 8745 8854h, RSCI9.TFSR 8745 8954h, RSCI10.TFSR 8745 8A54h, RSCI11.TFSR 8745 8B54h, RSCI12.TFSR 8745 8C54h,
RSCI0_NS.TFSR 0745 8054h, RSCI1_NS.TFSR 0745 8154h, RSCI2_NS.TFSR 0745 8254h, RSCI3_NS.TFSR 0745 8354h, RSCI4_NS.TFSR 0745 8454h, RSCI5_NS.TFSR 0745 8554h, RSCI6_NS.TFSR 0745 8654h, RSCI7_NS.TFSR 0745 8754h, RSCI8_NS.TFSR 0745 8854h, RSCI9_NS.TFSR 0745 8954h, RSCI10_NS.TFSR 0745 8A54h, RSCI11_NS.TFSR 0745 8B54h, RSCI12_NS.TFSR 0745 8C54h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	T[5:0]	Transmit FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of non-transmit data stored in transmit FIFO	R
b31 to b6	—	Reserved	These bits are read as 0.	R

T[5:0] Bits (Transmit FIFO Data Count)

Indicate the quantity of non-transmitted data stored in transmit FIFO.

00h means no untransmitted data. 10h means transmit FIFO is full.

42.2.21 Manchester Mode Status Register (MMSR)

Address(es): RSCI0.MMSR 8745 8058h, RSCI1.MMSR 8745 8158h, RSCI2.MMSR 8745 8258h, RSCI3.MMSR 8745 8358h, RSCI4.MMSR 8745 8458h, RSCI5.MMSR 8745 8558h, RSCI6.MMSR 8745 8658h, RSCI7.MMSR 8745 8758h, RSCI8.MMSR 8745 8858h, RSCI9.MMSR 8745 8958h, RSCI10.MMSR 8745 8A58h, RSCI11.MMSR 8745 8B58h, RSCI12.MMSR 8745 8C58h,
RSCI0_NS.MMSR 0745 8058h, RSCI1_NS.MMSR 0745 8158h, RSCI2_NS.MMSR 0745 8258h, RSCI3_NS.MMSR 0745 8358h, RSCI4_NS.MMSR 0745 8458h, RSCI5_NS.MMSR 0745 8558h, RSCI6_NS.MMSR 0745 8658h, RSCI7_NS.MMSR 0745 8758h, RSCI8_NS.MMSR 0745 8858h, RSCI9_NS.MMSR 0745 8958h, RSCI10_NS.MMSR 0745 8A58h, RSCI11_NS.MMSR 0745 8B58h, RSCI12_NS.MMSR 0745 8C58h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	RSYNC	—	MCER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFER	Preface Error Flag	This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
b1	SYER	Sync Error Flag	This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive Sync error detected 1: Receive Sync error detected	R
b2	SBER	Start Bit Error Flag	This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	MCER	Manchester Code Error Flag	Valid for Manchester mode only 0: No Manchester code error occurred 1: Manchester code error has occurred	R
b5	—	Reserved	This bit is read as 0.	R
b6	RSYNC	Received Sync Data	It is valid when MMCR.SBLEN bit = 1 in Manchester mode, 0 is read otherwise. 0: The received the Start Bit is DATA Sync 1: The received the Start Bit is COMMAND Sync	R
b31 to b7	—	Reserved	These bits are read as 0.	R

PFER Flag (Preface Error Flag)

This bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

When detecting a preface error when receiving frames in Manchester mode

The following operations are performed when a preface error occurs.

<When MMCR.PFERIE = 1>

The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the PFER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MMCR.PFERIE = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.PFERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the PFER flag to 0.

SYER Flag (Sync Error Flag)

This bit indicates that a receive Sync error was detected when receiving frames in Manchester mode with MMCR.SADJE bit = 1 (Manchester edge retiming enabled).
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.
[Setting condition]
When detecting a receive Sync error when receiving frames in Manchester mode
The following operations are performed when a receive Sync error occurs.
<When MMCR.SYERIE = 1>
Although the received data is transferred to the RDR register, no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SYER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SYERIE = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SYERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the SYER flag to 0.

SBER Flag (Start Bit Error Flag)

This bit indicates that a start bit error was detected when receiving frames in Manchester mode.
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.
[Setting condition]
When detecting a start bit error when receiving frames in Manchester mode
The following operations are performed when a start bit error occurs.
<When MMCR.SBERIE bit = 1>
The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SBER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SBERIE bit = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SBERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the SBER flag to 0.

MCER Flag (Manchester Code Error Flag)

When data is received in Manchester mode, Manchester code error is detected, and it is displayed. Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the MCER flag is not affected and retains its previous value.
[Setting conditions]
When receiving in Manchester mode and detecting Manchester code error in data area of received frame.
Received data when an error occurs is transferred to the RDR register, but the RXI interrupt request is not generated.
When the Manchester code error flag is set to 1, subsequent receive data is not transferred to the RDR register.
For details on Manchester code error, see section 42.5.11, Errors in Manchester Mode.
[Clearing condition]
Write 1 to MMSCR.MCERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the MCER flag to 0.

RSYNC Flag (Received Sync Data)

When Manchester mode (SCR3.MOD[2:0] = 101b) and MMCR.SBLEN = 1, this bit indicates the type of Sync of the received the start bit. For other settings, it is fixed to 0.

42.2.22 Extended Serial Mode Status Register 0 (XSR0)

Address(es): RSCI0.XSR0 8745 805Ch, RSCI1.XSR0 8745 815Ch, RSCI2.XSR0 8745 825Ch, RSCI3.XSR0 8745 835Ch, RSCI4.XSR0 8745 845Ch, RSCI5.XSR0 8745 855Ch, RSCI6.XSR0 8745 865Ch, RSCI7.XSR0 8745 875Ch, RSCI8.XSR0 8745 885Ch, RSCI9.XSR0 8745 895Ch, RSCI10.XSR0 8745 8A5Ch, RSCI11.XSR0 8745 8B5Ch, RSCI12.XSR0 8745 8C5Ch, RSCI0_NS.XSR0 0745 805Ch, RSCI1_NS.XSR0 0745 815Ch, RSCI2_NS.XSR0 0745 825Ch, RSCI3_NS.XSR0 0745 835Ch, RSCI4_NS.XSR0 0745 845Ch, RSCI5_NS.XSR0 0745 855Ch, RSCI6_NS.XSR0 0745 865Ch, RSCI7_NS.XSR0 0745 875Ch, RSCI8_NS.XSR0 0745 885Ch, RSCI9_NS.XSR0 0745 895Ch, RSCI10_NS.XSR0 0745 8A5Ch, RSCI11_NS.XSR0 0745 8B5Ch, RSCI12_NS.XSR0 0745 8C5Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SFSF	Start Frame Status Flag	0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
b1	RXDSF	RXD Input Status Flag	0: RXD input to RSCI core is enabled 1: RXD input to RSCI core is disabled (RXD is not input to the RSCI core)	R*1
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	BFOF	Break Field Low Width Output Complete Flag	0: When Break Field transmission is not completed 1: When Break Field transmission is completed	R
b9	BCDF	Bus Collision Detected Flag	0: When bus conflict is not detected 1: When bus conflict is detected	R
b10	BFDF	Break Field Low Width Detection Flag	0: When Break Field is not detected 1: When Break Field is detected	R
b11	CF0MF	Control Field 0 Match Flag	0: When Control Field 0 data and the compare data do not match 1: When Control Field 0 data and the compare data match	R
b12	CF1MF	Control Field 1 Match Flag	0: When Control Field 1 data and the compare data do not match 1: When Control Field 1 data and the compare data match	R
b13	PIBDF	Priority Interrupt Bit Detection Flag	0: When priority interrupt bit is not detected 1: When Priority interrupt bit is detected	R
b14	COF	Count Overflow Flag	0: When the counter for Break Field detection does not overflow 1: When the counter for Break Field detection overflows	R
b15	AEDF	Effective Edge Detection Flag	0: When Active edge is not detected 1: When Active edge is detected	R
b23 to b16	CF0RD[7:0]	Control Field 0 Received Data	Control Field 0 received data.	R
b31 to b24	CF1RD[7:0]	Control Field 1 Received Data	Control Field 1 received data.	R

Note 1. Wait at least 1 TCLK cycle after the receive data full interrupt (RXI) before reading this register.

SFSF Flag (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.
[Setting condition]

- When 1 is written to XCR1.SDST bit.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST bit is 0.
- When Start Frame detection is completed.

RXDSF Flag (RXD Input Status Flag)

Indicates the RXD input status to the RSCI core. When this bit is 1, RXD input is received only by the extended serial module and the Break Field is detected and is not input to the RSCI core.

42.2.23 Extended Serial Mode Status Register 1 (XSR1)

Address(es): RSCI0.XSR1 8745 8060h, RSCI1.XSR1 8745 8160h, RSCI2.XSR1 8745 8260h, RSCI3.XSR1 8745 8360h, RSCI4.XSR1 8745 8460h, RSCI5.XSR1 8745 8560h, RSCI6.XSR1 8745 8660h, RSCI7.XSR1 8745 8760h, RSCI8.XSR1 8745 8860h, RSCI9.XSR1 8745 8960h, RSCI10.XSR1 8745 8A60h, RSCI11.XSR1 8745 8B60h, RSCI12.XSR1 8745 8C60h,
RSCI0_NS.XSR1 0745 8060h, RSCI1_NS.XSR1 0745 8160h, RSCI2_NS.XSR1 0745 8260h, RSCI3_NS.XSR1 0745 8360h, RSCI4_NS.XSR1 0745 8460h, RSCI5_NS.XSR1 0745 8560h, RSCI6_NS.XSR1 0745 8660h, RSCI7_NS.XSR1 0745 8760h, RSCI8_NS.XSR1 0745 8860h, RSCI9_NS.XSR1 0745 8960h, RSCI10_NS.XSR1 0745 8A60h, RSCI11_NS.XSR1 0745 8B60h, RSCI12_NS.XSR1 0745 8C60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CCV[15:0]	Captured Count Value	Stores the 16-bit counter capture value.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

CCV[15:0] Bits (Captured Count Value)

Stores the capture value of the 16-bit counter of the extended serial module.

When sending Start Frame

This register holds the previous value.

When receiving Start Frame with bit rate measurement disabled

If a Break Field is detected in the Break Field detection state (refer to Figure 42.74), the Break Field length is captured and held (counter value is captured at the rising edge of RXD).

If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.

If the counter overflows, it will not be captured.

When receiving Start Frame with bit rate measurement enabled

The count value is captured and held at the valid edge (both RXD edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by this register read. Even if a valid edge occurs before reading, the counter value is not captured.

42.2.24 Status Clear Register (SSCR)

Address(es): RSCI0.SSCR 8745 8068h, RSCI1.SSCR 8745 8168h, RSCI2.SSCR 8745 8268h, RSCI3.SSCR 8745 8368h, RSCI4.SSCR 8745 8468h, RSCI5.SSCR 8745 8568h, RSCI6.SSCR 8745 8668h, RSCI7.SSCR 8745 8768h, RSCI8.SSCR 8745 8868h, RSCI9.SSCR 8745 8968h, RSCI10.SSCR 8745 8A68h, RSCI11.SSCR 8745 8B68h, RSCI12.SSCR 8745 8C68h,
RSCI0_NS.SSCR 0745 8068h, RSCI1_NS.SSCR 0745 8168h, RSCI2_NS.SSCR 0745 8268h, RSCI3_NS.SSCR 0745 8368h, RSCI4_NS.SSCR 0745 8468h, RSCI5_NS.SSCR 0745 8568h, RSCI6_NS.SSCR 0745 8668h, RSCI7_NS.SSCR 0745 8768h, RSCI8_NS.SSCR 0745 8868h, RSCI9_NS.SSCR 0745 8968h, RSCI10_NS.SSCR 0745 8A68h, RSCI11_NS.SSCR 0745 8B68h, RSCI12_NS.SSCR 0745 8C68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRFC	—	TDREC	AFERC	APERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	ERSC	ERS Clear	Setting this bit to 1 clears the SSR.ERS flag. The read value is always 0.	W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DCMFC	DCMF Clear	Setting this bit to 1 clears the SSR.DCMF flag. The read value is always 0.	W
b17	DPERC	DPER Clear	Setting this bit to 1 clears the SSR.DPER flag. The read value is always 0.	W
b18	DFERC	DFER Clear	Setting this bit to 1 clears the SSR.DFER flag. The read value is always 0.	W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	ORERC	ORER Clear	Setting this bit to 1 clears the SSR.ORER flag. The read value is always 0.	W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26	MFFC	MFF Clear	Setting this bit to 1 clears the SSR.MFF flag. The read value is always 0.	W
b27	APERC	APER Clear	Setting this bit to 1 clears the SSR.APER flag. The read value is always 0.	W
b28	AFERC	AFER Clear	Setting this bit to 1 clears the SSR.AFER flag. The read value is always 0.	W
b29	TDREC	TDRE Clear	Setting this bit to 1 clears the SSR.TDRE flag. The read value is always 0.	W
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	RDRFC	RDRF Clear	Setting this bit to 1 clears the SSR.RDRF flag. The read value is always 0.	W

42.2.25 I²C Status Clear Register (SISCR)

Address(es): RSCI0.SISCR 8745 806Ch, RSCI1.SISCR 8745 816Ch, RSCI2.SISCR 8745 826Ch, RSCI3.SISCR 8745 836Ch, RSCI4.SISCR 8745 846Ch, RSCI5.SISCR 8745 856Ch, RSCI6.SISCR 8745 866Ch, RSCI7.SISCR 8745 876Ch, RSCI8.SISCR 8745 886Ch, RSCI9.SISCR 8745 896Ch, RSCI10.SISCR 8745 8A6Ch, RSCI11.SISCR 8745 8B6Ch, RSCI12.SISCR 8745 8C6Ch,
RSCI0_NS.SISCR 0745 806Ch, RSCI1_NS.SISCR 0745 816Ch, RSCI2_NS.SISCR 0745 826Ch, RSCI3_NS.SISCR 0745 836Ch, RSCI4_NS.SISCR 0745 846Ch, RSCI5_NS.SISCR 0745 856Ch, RSCI6_NS.SISCR 0745 866Ch, RSCI7_NS.SISCR 0745 876Ch, RSCI8_NS.SISCR 0745 886Ch, RSCI9_NS.SISCR 0745 896Ch, RSCI10_NS.SISCR 0745 8A6Ch, RSCI11_NS.SISCR 0745 8B6Ch, RSCI12_NS.SISCR 0745 8C6Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	IICSTIFC	IICSTIF Clear	Setting this bit to 1 clears the SISR.IICSTIF flag. The read value is always 0.	W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

42.2.26 Receive FIFO Status Clear Register (RFSCR)

Address(es): RSCI0.RFSCR 8745 8070h, RSCI1.RFSCR 8745 8170h, RSCI2.RFSCR 8745 8270h, RSCI3.RFSCR 8745 8370h, RSCI4.RFSCR 8745 8470h, RSCI5.RFSCR 8745 8570h, RSCI6.RFSCR 8745 8670h, RSCI7.RFSCR 8745 8770h, RSCI8.RFSCR 8745 8870h, RSCI9.RFSCR 8745 8970h, RSCI10.RFSCR 8745 8A70h, RSCI11.RFSCR 8745 8B70h, RSCI12.RFSCR 8745 8C70h,
RSCI0_NS.RFSCR 0745 8070h, RSCI1_NS.RFSCR 0745 8170h, RSCI2_NS.RFSCR 0745 8270h, RSCI3_NS.RFSCR 0745 8370h, RSCI4_NS.RFSCR 0745 8470h, RSCI5_NS.RFSCR 0745 8570h, RSCI6_NS.RFSCR 0745 8670h, RSCI7_NS.RFSCR 0745 8770h, RSCI8_NS.RFSCR 0745 8870h, RSCI9_NS.RFSCR 0745 8970h, RSCI10_NS.RFSCR 0745 8A70h, RSCI11_NS.RFSCR 0745 8B70h, RSCI12_NS.RFSCR 0745 8C70h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRC	DR Clear	Setting this bit to 1 clears the RFSR.DR flag. The read value is always 0.	W
b1	—	Reserved	This bit is read as 0.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

42.2.27 Manchester Mode Status Clear Register (MMSCR)

Address(es): RSCI0.MMSCR 8745 8074h, RSCI1.MMSCR 8745 8174h, RSCI2.MMSCR 8745 8274h, RSCI3.MMSCR 8745 8374h, RSCI4.MMSCR 8745 8474h, RSCI5.MMSCR 8745 8574h, RSCI6.MMSCR 8745 8674h, RSCI7.MMSCR 8745 8774h, RSCI8.MMSCR 8745 8874h, RSCI9.MMSCR 8745 8974h, RSCI10.MMSCR 8745 8A74h, RSCI11.MMSCR 8745 8B74h, RSCI12.MMSCR 8745 8C74h,
RSCI0_NS.MMSCR 0745 8074h, RSCI1_NS.MMSCR 0745 8174h, RSCI2_NS.MMSCR 0745 8274h, RSCI3_NS.MMSCR 0745 8374h, RSCI4_NS.MMSCR 0745 8474h, RSCI5_NS.MMSCR 0745 8574h, RSCI6_NS.MMSCR 0745 8674h, RSCI7_NS.MMSCR 0745 8774h, RSCI8_NS.MMSCR 0745 8874h, RSCI9_NS.MMSCR 0745 8974h, RSCI10_NS.MMSCR 0745 8A74h, RSCI11_NS.MMSCR 0745 8B74h, RSCI12_NS.MMSCR 0745 8C74h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	MCERC	—	SBERC	SYERC	PFERC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFERC	PFER Clear	Setting this bit to 1 clears the MMSR.PFER flag. The read value is always 0.	W
b1	SYERC	SYER Clear	Setting this bit to 1 clears the MMSR.SYER flag. The read value is always 0.	W
b2	SBERC	SBER Clear	Setting this bit to 1 clears the MMSR.SBER flag. The read value is always 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	MCERC	MCER Clear	Setting this bit to 1 clears the MMSR.MCER flag. The read value is always 0.	W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

42.2.28 Extended Serial Mode Status Clear Register (XSCR)

Address(es): RSCI0.XSCR 8745 8078h, RSCI1.XSCR 8745 8178h, RSCI2.XSCR 8745 8278h, RSCI3.XSCR 8745 8378h, RSCI4.XSCR 8745 8478h, RSCI5.XSCR 8745 8578h, RSCI6.XSCR 8745 8678h, RSCI7.XSCR 8745 8778h, RSCI8.XSCR 8745 8878h, RSCI9.XSCR 8745 8978h, RSCI10.XSCR 8745 8A78h, RSCI11.XSCR 8745 8B78h, RSCI12.XSCR 8745 8C78h,
RSCI0_NS.XSCR 0745 8078h, RSCI1_NS.XSCR 0745 8178h, RSCI2_NS.XSCR 0745 8278h, RSCI3_NS.XSCR 0745 8378h, RSCI4_NS.XSCR 0745 8478h, RSCI5_NS.XSCR 0745 8578h, RSCI6_NS.XSCR 0745 8678h, RSCI7_NS.XSCR 0745 8778h, RSCI8_NS.XSCR 0745 8878h, RSCI9_NS.XSCR 0745 8978h, RSCI10_NS.XSCR 0745 8A78h, RSCI11_NS.XSCR 0745 8B78h, RSCI12_NS.XSCR 0745 8C78h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AEDCL	COFC	PIBDCL	CF1MCL	CF0MCL	BFDCL	BCDCL	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BFOC	BFOF Clear	Setting this bit to 1 clears the XSR0.BFOF flag. The read value is always 0.	W
b9	BCDCL	BCDF Clear	Setting this bit to 1 clears the XSR0.BCDF flag. The read value is always 0.	W
b10	BFDCL	BFDF Clear	Setting this bit to 1 clears the XSR0.BFDF flag. The read value is always 0.	W
b11	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the XSR0.CF0MF flag. The read value is always 0.	W
b12	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the XSR0.CF1MF flag. The read value is always 0.	W
b13	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the XSR0.PIBDF flag. The read value is always 0.	W
b14	COFC	COF Clear	Setting this bit to 1 clears the XSR0.COF flag. The read value is always 0.	W
b15	AEDCL	AEDF Clear	Setting this bit to 1 clears the XSR0.AEDF flag. The read value is always 0.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

42.2.29 Internal Signal Status Register (SISSR)

Address(es): RSCI0.SISSR 8745 801Ch, RSCI1.SISSR 8745 811Ch, RSCI2.SISSR 8745 821Ch, RSCI3.SISSR 8745 831Ch, RSCI4.SISSR 8745 841Ch, RSCI5.SISSR 8745 851Ch, RSCI6.SISSR 8745 861Ch, RSCI7.SISSR 8745 871Ch, RSCI8.SISSR 8745 881Ch, RSCI9.SISSR 8745 891Ch, RSCI10.SISSR 8745 8A1Ch, RSCI11.SISSR 8745 8B1Ch, RSCI12.SISSR 8745 8C1Ch,
RSCI0_NS.SISSR 0745 801Ch, RSCI1_NS.SISSR 0745 811Ch, RSCI2_NS.SISSR 0745 821Ch, RSCI3_NS.SISSR 0745 831Ch, RSCI4_NS.SISSR 0745 841Ch, RSCI5_NS.SISSR 0745 851Ch, RSCI6_NS.SISSR 0745 861Ch, RSCI7_NS.SISSR 0745 871Ch, RSCI8_NS.SISSR 0745 881Ch, RSCI9_NS.SISSR 0745 891Ch, RSCI10_NS.SISSR 0745 8A1Ch, RSCI11_NS.SISSR 0745 8B1Ch, RSCI12_NS.SISSR 0745 8C1Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ITEF	—	—	—	IREF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IREF	Internal RE Signal Monitor Flag	0: Internal RE signal state is 0. 1: Internal RE signal state is 1.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	ITEF	Internal TE Signal Monitor Flag	0: Internal TE signal state is 0. 1: Internal TE signal state is 1.	R
b7 to b5	—	Reserved	These bits are read as 0.	R

A different clock than the operating clock of the control registers (PCLK) can be selected for the operating clock of the transmitter and receiver (TCLK). Synchronizers are inserted between the control registers and the transmitter/receiver to ensure correct operation even when a different clock is selected. For this reason, after rewriting the control register, it takes time for the value to propagate internally.

This register is used to monitor the status of the SCR0.TE bit and SCR0.RE bit, which are used to permit transmission and reception, after they have passed through the synchronizer. When changing the SCR0.TE and SCR0.RE bits from 1 to 0 to rewrite control registers for the next communication, be sure to rewrite them after the ITEF and IREF flags have both become 0.

42.2.30 HBS Support Mode Control Register (HBSCR)

Address(es): RSCI0.HBSCR 8745 801Eh, RSCI1.HBSCR 8745 811Eh, RSCI2.HBSCR 8745 821Eh, RSCI3.HBSCR 8745 831Eh, RSCI4.HBSCR 8745 841Eh, RSCI5.HBSCR 8745 851Eh, RSCI6.HBSCR 8745 861Eh, RSCI7.HBSCR 8745 871Eh, RSCI8.HBSCR 8745 881Eh, RSCI9.HBSCR 8745 891Eh, RSCI10.HBSCR 8745 8A1Eh, RSCI11.HBSCR 8745 8B1Eh, RSCI12.HBSCR 8745 8C1Eh,
RSCI0_NS.HBSCR 0745 801Eh, RSCI1_NS.HBSCR 0745 811Eh, RSCI2_NS.HBSCR 0745 821Eh, RSCI3_NS.HBSCR 0745 831Eh, RSCI4_NS.HBSCR 0745 841Eh, RSCI5_NS.HBSCR 0745 851Eh, RSCI6_NS.HBSCR 0745 861Eh, RSCI7_NS.HBSCR 0745 871Eh, RSCI8_NS.HBSCR 0745 881Eh, RSCI9_NS.HBSCR 0745 891Eh, RSCI10_NS.HBSCR 0745 8A1Eh, RSCI11_NS.HBSCR 0745 8B1Eh, RSCI12_NS.HBSCR 0745 8C1Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LPS	AOE	—	HBSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HBSE	HBS Support Mode Enable	0: Pulse width for data 0 is specified as 100% of a bit period (NRZ coding). 1: Pulse width for data 0 is specified as 50% of a bit period (RZI coding and inversion).	R/W *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	AOE	Alternate Output Enable	0: Data is output from the TXDn pin. 1: Data 0 is alternately output from the TXDAn and TXDBn pins.	R/W *1
b3	LPS	Leading Output Pin Select	0: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDAn pin 1: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDBn pin	R/W *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

HBSE Bit (HBS Support Mode Enable)

When this bit is 1, the transmitter encodes the transmit data to the negative logic RZI code and the receiver decodes the receive data to the NRZ code. Also, transmit data can be output from the TXDAn/TXDBn pin. This function should be used only in asynchronous mode.

AOE Bit (Alternate Output Enable)

This bit is used to select whether the data is output from the TXDn pin or data 0 is alternately output from the TXDAn and TXDBn pins in HBS support mode.

LPS Bit (Leading Output Pin Select)

This bit is used when the AOE bit is 1 in HBS support mode.
When it is set to 0, the start bit is transmitted from TXDAn pin and the data 0 is output alternately from the TXDBn and TXDAn pins.
When it is set to 1, the start bit is transmitted from TXDBn pin and the data 0 is output alternately from the TXDAn and TXDBn pins.
For details, refer to the operation description in section 42.6, HBS Support Mode.

42.2.31 RXD Input Path Select Register (RXDSEL)

Address(es): RSCI2.RXDSEL 8780 7000h, RSCI3.RXDSEL 8780 7020h, RSCI2_NS.RXDSEL 0780 7000h, RSCI3_NS.RXDSEL 0780 7020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ISEL	Input Path Select*1	This bit selects the signal for input to the RXD pin of RSCI. For RSCI2: 0: Input signal from the RXD2 pin 1: The GTIOC10A signal output by GPTW10 For RSCI3: 0: Input signal from the RXD3 pin 1: The GTIOC12A signal output by GPTW12	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the ISEL bit is set to 1, the RSCI and GPTW should have the same security attribute.

42.2.32 GTIOC Input Path Select Register (GTIOCSEL)

Address(es): GPTW11.GTIOCSEL 8780 7010h, GPTW13.GTIOCSEL 8780 7030h,
GPTW11_NS.GTIOCSEL 0780 7010h, GPTW13_NS.GTIOCSEL 0780 7030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ISEL	Input Path Select*1	This bit selects the signals for input to the GTIOCnA and GTIOCnB pins of GPTW (n = 11, 13). For GPTW11: 0: Input signals from the GTIOC11A and GTIOC11B pins 1: The TXD2 signal output by RSCI2 For GPTW13: 0: Input signals from the GTIOC13A and GTIOC13B pins 1: The TXD3 signal output by RSCI3	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the ISEL bit is set to 1, the RSCI and GPTW should have the same security attribute.

42.3 Operation in Asynchronous Mode

Figure 42.4 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.
The RSCI monitors the communications line. When the RSCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the SCR3.RXDESEL bit setting. RSCI regards space (low level) as a start bit when SCR3.RXDESEL bit is 0. RSCI regards a fall edge as a start bit when RXDESEL bit is 1.
Inside the RSCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure (it has also FIFO mode), so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

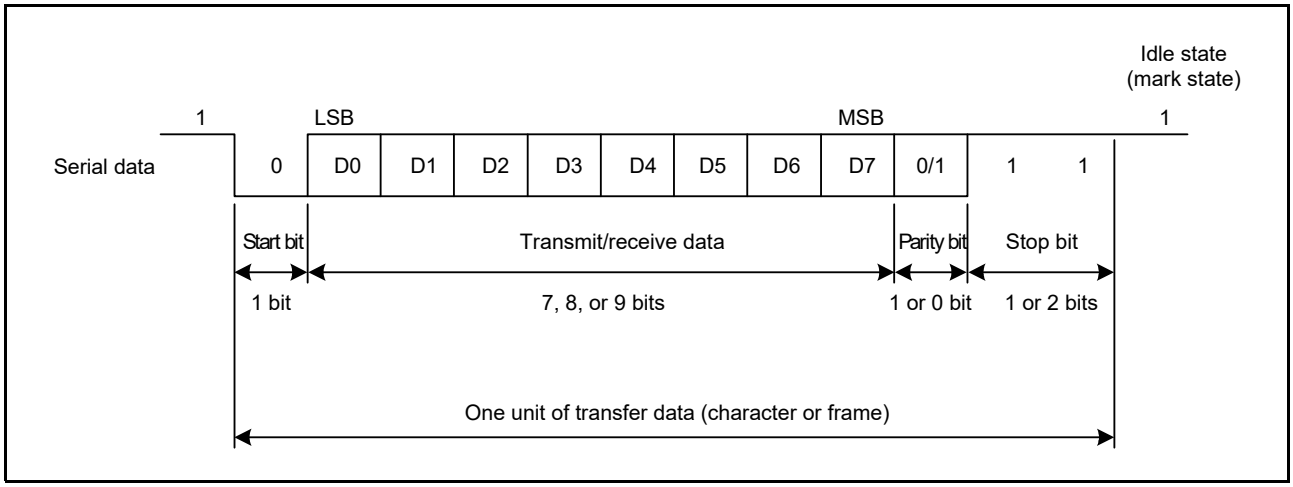


Figure 42.4 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

42.3.1 Serial Data Transfer Format

Table 42.28 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SCR1 and SCR3 setting. For details of multi-processor function, refer to section 42.4, Multi-Processor Communication Function.

Table 42.28 Serial Transfer Formats (Asynchronous Mode)

SCR3		SCR1	SCR3		Serial Transfer Format and Frame Length
CHR[1]	CHR[0]	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

42.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the RSCI samples the falling edge of the start bit using the base clock, and performs internal synchronization*2. When sampling timing does not adjust (SCR4.RTADJ bit = 0 or SCR4.RTADJ bit = 1 and SCR4.RTMG[2:0] bits = 000b), receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 42.5. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SCR2.ABCSE2 = 0, SCR2.ABCSE = 0, and SCR2.ABCS = 0, N = 8 when SCR2.ABCSE2 = 0, SCR2.ABCSE = 0, and SCR2.ABCS = 1, N = 6 when SCR2.ABCSE2 = 0 and SCR2.ABCSE = 1, N = 4 when SCR2.ABCSE2 = 1 and SCR2.ABCSE = 0)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ (\%)$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCSE2, ABCSE, and ABCS bits in the SCR2 register is 0. When the ABCSE2 and ABCSE bits are 0 and the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When the ABCSE2 bit is 0 and the ABCSE bit is 1, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock. When the ABCSE2 bit is 1 and the ABCSE bit is 0, a frequency of 4 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 2nd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.
In the case of the function of adjust sampling timing is OFF (RTADJ bit = 0):
The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing.
In Figure 42.5, the low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the RSCI judges this as a noise. So, the RSCI does not start reception and wait start bit.
In the case of the function of adjust sampling timing is ON (RTADJ bit = 1):
The determination condition of a start bit is that Low keeps up until the sampling timing.
Adjusting the sampling timing forward (RTMG[3] bit = 1) increases the possibility of erroneously determining a noise as the start bit.

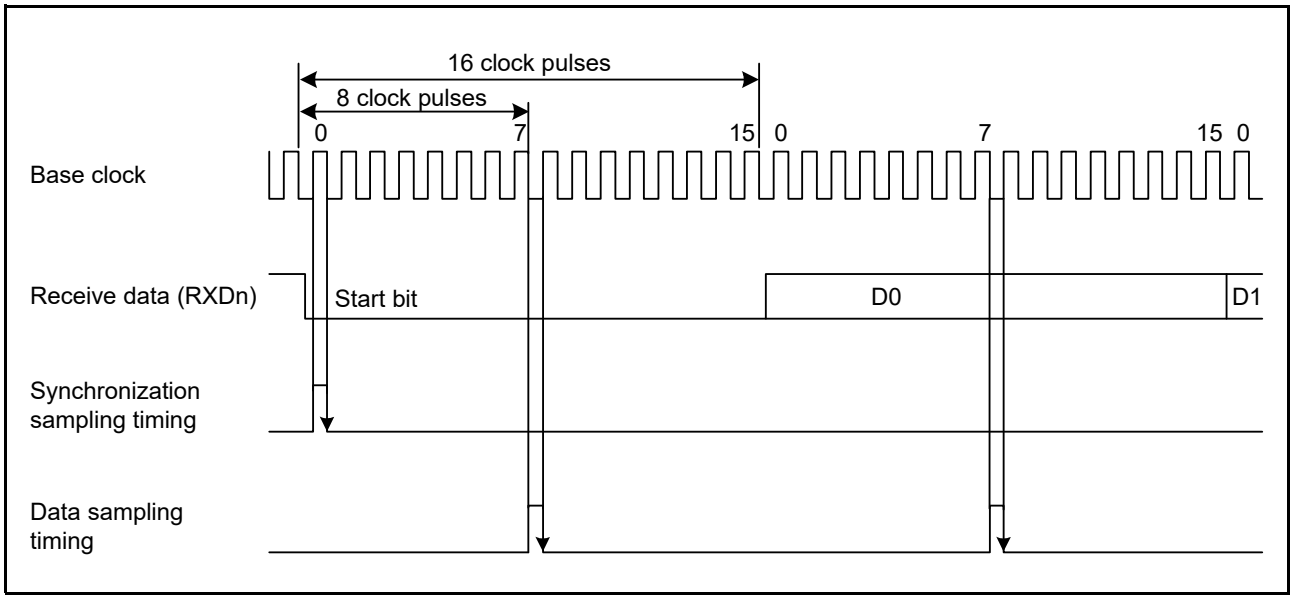


Figure 42.5 Receive Data Sampling Timing in Asynchronous Mode

42.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the RSCI's transfer clock, according to the setting of the SCR3.CKE[1:0] bits.

When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SCR2.ABCS bit = 0) and 8 times the bit rate (when SCR2.ABCS bit = 1). In addition, when an external clock is selected, the base clock generated by TMR0 and TMR1 or TMR2 and TMR3 can be selected by the SCR3.ACS0 bit.

When the RSCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 42.6.

If you selected an internal clock, the SCKn pin is outputted after SCR0.TE bit is set to 1 or SCR0.RE bit is set to 1.

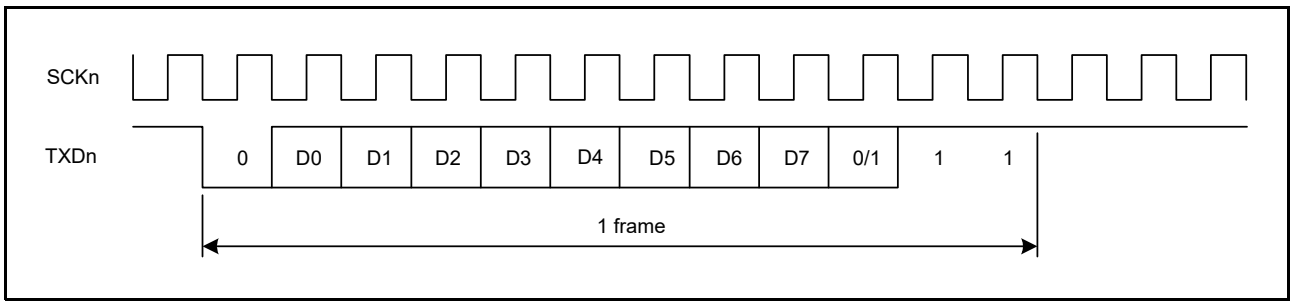


Figure 42.6 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SCR1.PE Bit = 1, SCR3.CHR[1:0] Bits = 10b, MP Bit = 0, STOP Bit = 1)

42.3.4 Double-Speed Mode, Divide-by-6 Mode, and Divided-by-4 Mode

When SCR2.ABCS bit is set to 1, the RSCI operates on the bit rate twice that in the case where ABCS bit is set to 0. And when SCR2.BGDM bit is set to 1, the cycle of the base clock is halved and the bit rate is doubled from that in the case where BGDM bit is set to 0. When SCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the RSCI to operate on a bit rate four times that in the case ABCS bit = 0 and BGDM bit = 0.

When SCR2.ABCSE2 bit is set to 0 and SCR2.ABCSE bit is set to 1, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of ABCS bit = 0, BGDM bit = 0, ABCSE bit = 0, and ABCSE2 bit = 0.

When SCR2.ABCSE2 bit is set to 1 and SCR2.ABCSE bit is set to 0, the number of base clock pulses are 4 during a period of 1 bit, and the base clock frequency is half. And RSCI works 8 times of bit rate compared with a case of ABCS bit = 0, BGDM bit = 0, ABCSE bit = 0, and ABCSE2 bit = 0.

As shown by Formula (1) in section 42.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when ABCS bit is set to 1, ABCSE bit is set to 1, or ABCSE2 bit is set to 1. Therefore, if the desired bit rate can be obtained with ABCS bit set to 0, ABCSE bit set to 0, and ABCSE2 bit set to 0, it is recommended to use the RSCI with ABCS bit set to 0, ABCSE bit set to 0, and ABCSE2 bit set to 0.

42.3.5 CTS and RTS Functions

The CTS function is the transmission control function by the CTSn# pin. Setting the SCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting that uses either function with one pin or the dedicated setting that uses each function independently with two pins. This setting is done with the SCR1.CRSEP bit.

When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.

Even if the CTSn# pin goes high after transmission starts, the frame being transmitted is not affected and transmission will continue.

The RTS function is the transmission request function by the RTSn# pin. In the RTS function, the RTSn# pin output low level when reception becomes possible. Conditions for output of the low and high level are shown below.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]

When the following conditions are all satisfied

- The SCR0.RE bit is 1
- There are no received data yet to be read and reception is not in progress
- The ORER, AFER, and APER flags in the SSR are all 0

[Condition for high-level output]

When the conditions for low-level output are not satisfied

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

[Conditions for low-level output]

When the following conditions are all satisfied

- The SCR0.RE bit is 1
- The number of data stored in the receive FIFO is less than the threshold (FCR.RSTRG[4:0] bits)
- The SSR.ORER (RDR.ORER) flag is 0

[Condition for high-level output]

When the conditions for low-level output are not satisfied

42.3.6 Data Match Detection

The data match detection function can use only the asynchronous mode.
If SCR0.DCME bit is set to 1*2, when one frame of data has been received, RSCI compares that receive data with the data which is set to SCR4.CMPD[8:0] bits. If RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits*1) with receive data, RSCI can issue RXI interrupt request.
If SCR3.MP bit is set to 0, this comparative target in communication data is valid only data field in receive format. In multi-processor mode (SCR3.MP bit = 1), if SCR0.IDSEL bit is set to 1, the reception data at which MPB is 1 detects data match or unmatch, and the reception data at which MPB is 0 detects always unmatch. If SCR0.IDSEL bit is set to 0, RSCI detects data match or unmatch despite the value of the MPB of the reception data at every reception complete.
Until RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, the communication data is skipped (discarded), and RSCI can not detect parity error, framing error. When RSCI detects the match, the SCR0.DCME bit is automatically cleared, and SSR.DCMF flag is set to 1. If SCR0.IDSEL bit is set to 1 at this time, SCR0.MPIE bit is automatically cleared. And if SCR0.IDSEL bit is set to 0 at this time, SCR0.MPIE bit is kept. At the same time, if SCR0.RIE bit is set to 1, RSCI issues RXI interrupt request. If RSCI detects framing error in comparative receive data which is detected the match, SSR.DFER flag is set to 1, and if RSCI detects parity error in that frame, SSR.DPER flag is set to 1. That comparative receive data and MPB are not stored to RDR register, and SSR.RDRF flag is retained to 0.
After RSCI detects the match, and the SCR0.DCME bit is automatically cleared, it receives next data continuously in current register setting.
When SSR.DFER flag or SSR.DPER flag is set, the data match isn't detected. Before making the data match detection function effective, please be sure to set SSR.DFER and SSR.DPER flags as 0.
Figure 42.7 and Figure 42.8 show the data match detection example.

Note 1. This comparative target can select one length of 3 types, they are CMPD[6:0] bits with 7bit length enable, CMPD[7:0] bits with 8bit, and CMPD[8:0] bits with 9bit length.

Note 2. Set the SCR0.DCME bit to 1 before receiving the start bit of the received frame that performs data matching.

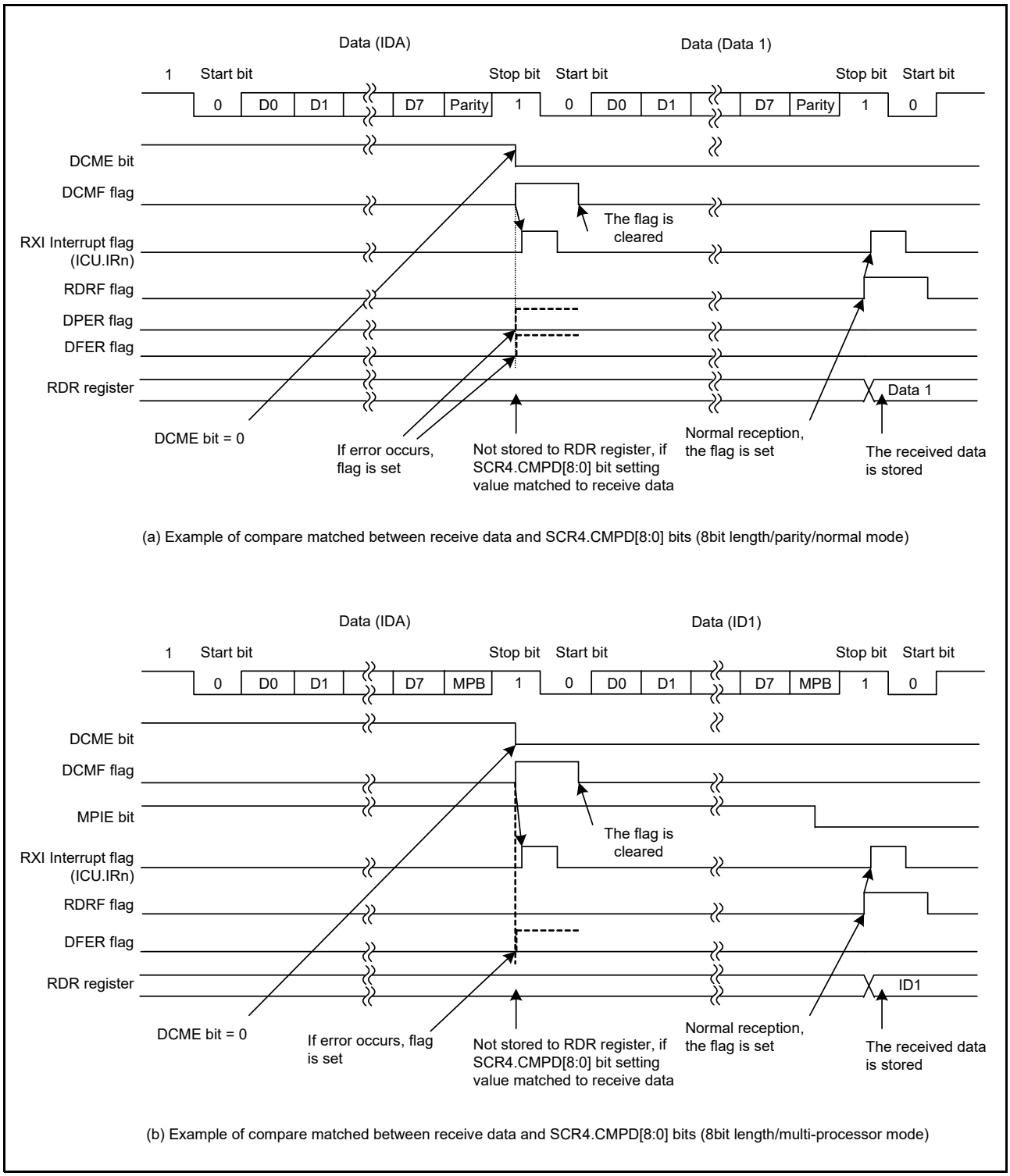
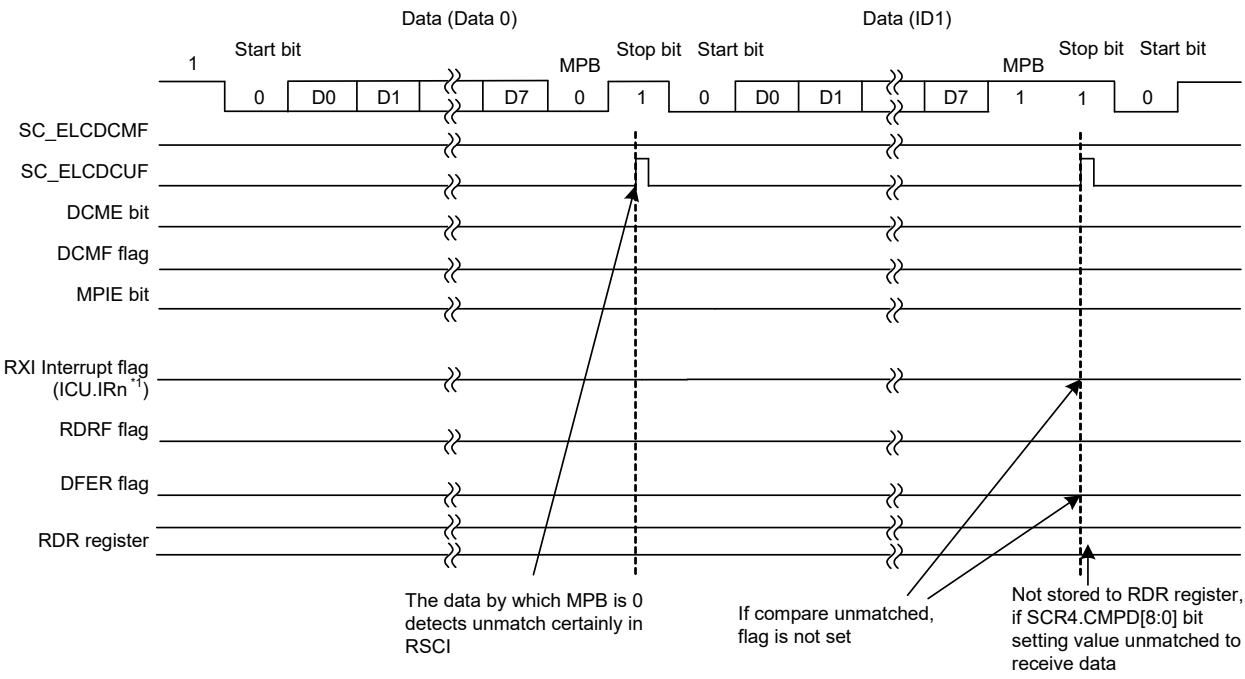
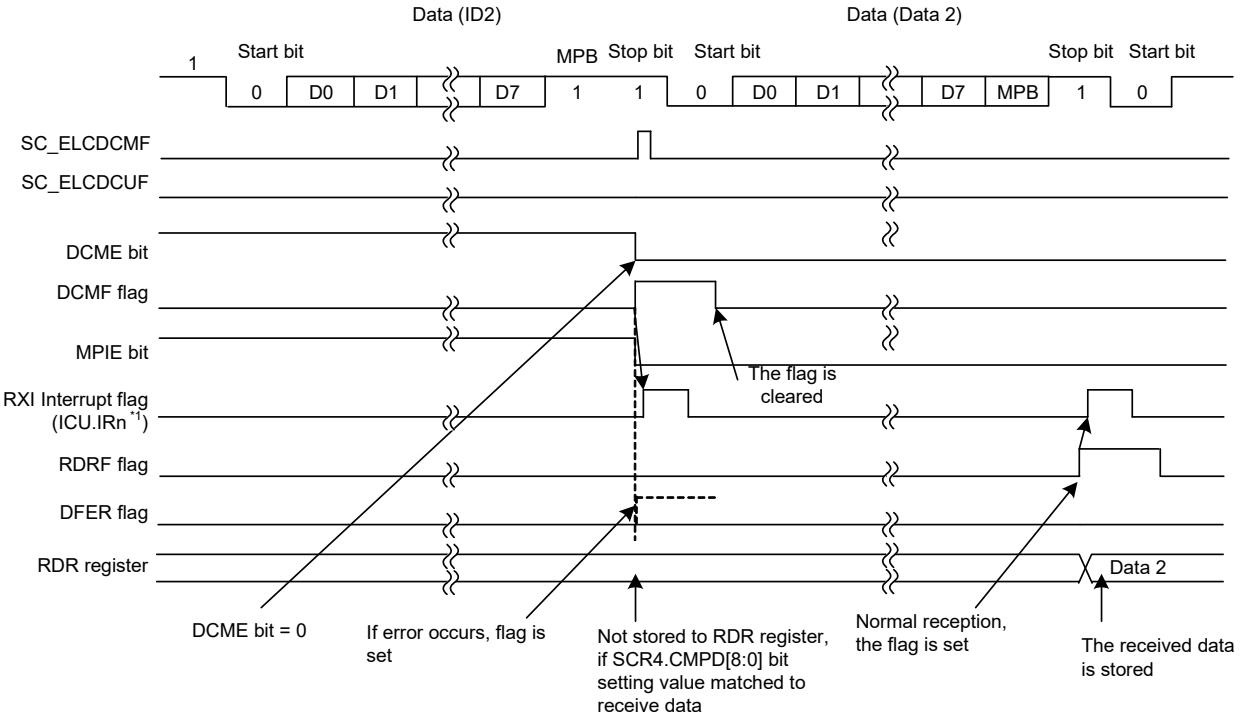


Figure 42.7 Example of Data Match Detection (1) 8bit-Data



(a) Example of compare unmatched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)



(b) Example of compare matched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)

Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.8 Example of Data Match Detection (2) Multi-Processor Mode/8bit-Data

42.3.7 RSCI Initialization (Asynchronous Mode)

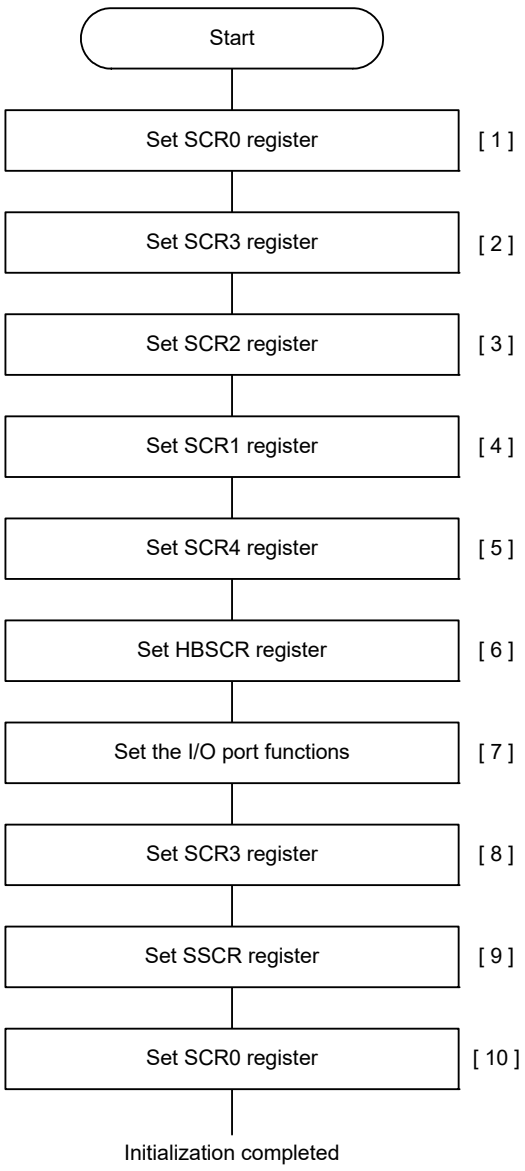
Before transmitting and receiving data, start by writing 0 to SCR0.TE and SCR0.RE bits (or writing the initial value to SCR0 register) and then continue through the procedure (select to non-FIFO or FIFO) for RSCI given in Figure 42.9 or Figure 42.10. Whenever the operating mode or transfer format is changed, SCR0.TE and SCR0.RE bits must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR0.RE bit to 0 initializes neither the ORER, AFER, APER, RDRF, DR flags, and RDR register.

In FIFO mode, even if the TE bit is set as 0, the TEND flag isn't initialized, so please be careful. Please be also careful at the time of change in the operation mode.

Moreover, note that switching the value of the SCR0.TE bit from 0 to 1 while the SCR0.TIE bit is 1 leads to the generation of a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO no-use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, *2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] When using the HBS support mode, enable or set the function. If this function is not used, leave the default value.

[7] Make I/O port settings to enable input and output functions as required for TXDn (TXDAn/TXDBn), RXDn, and SCKn pins.

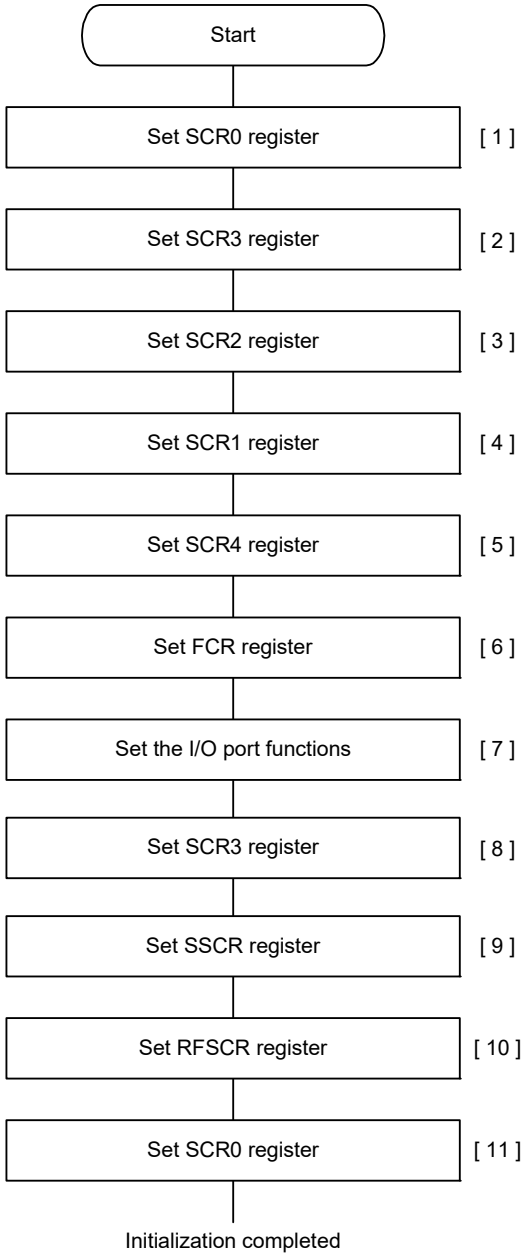
[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[10] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function. Setting the TE and RE bits allows TXDn (TXDAn/TXDBn) and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 42.9 Sample RSCI Initialization Flowchart (Asynchronous Mode/Non-FIFO Mode)



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, 2,select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Set the TFRST and RFRST to 1 to empty FIFO.
Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it is initialization flow after a reset, you can skip this step.

[10] Write 1 to The RFSCR.DRC bit and clear the corresponding flag.
If it is initialization flow after a reset, you can skip this step.

[11] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 42.10 Sample RSCI Initialization Flowchart (Asynchronous Mode, FIFO Mode)

Figure 42.11 shows an example of the timing when data is transmitted after reset is released, and RSCI is set to asynchronous mode according to Figure 42.9 or Figure 42.10. As shown in the figure, when the pin function is set to the TXDn pin, the SCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the SCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR register to data transmission starts. In asynchronous mode, TXDn is high during this period.

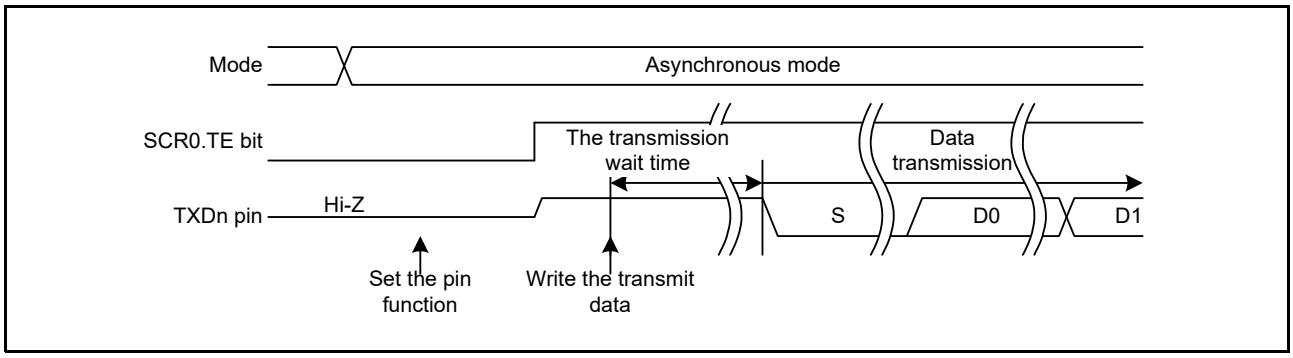


Figure 42.11 Example of Data Transmission Timing in Asynchronous Mode

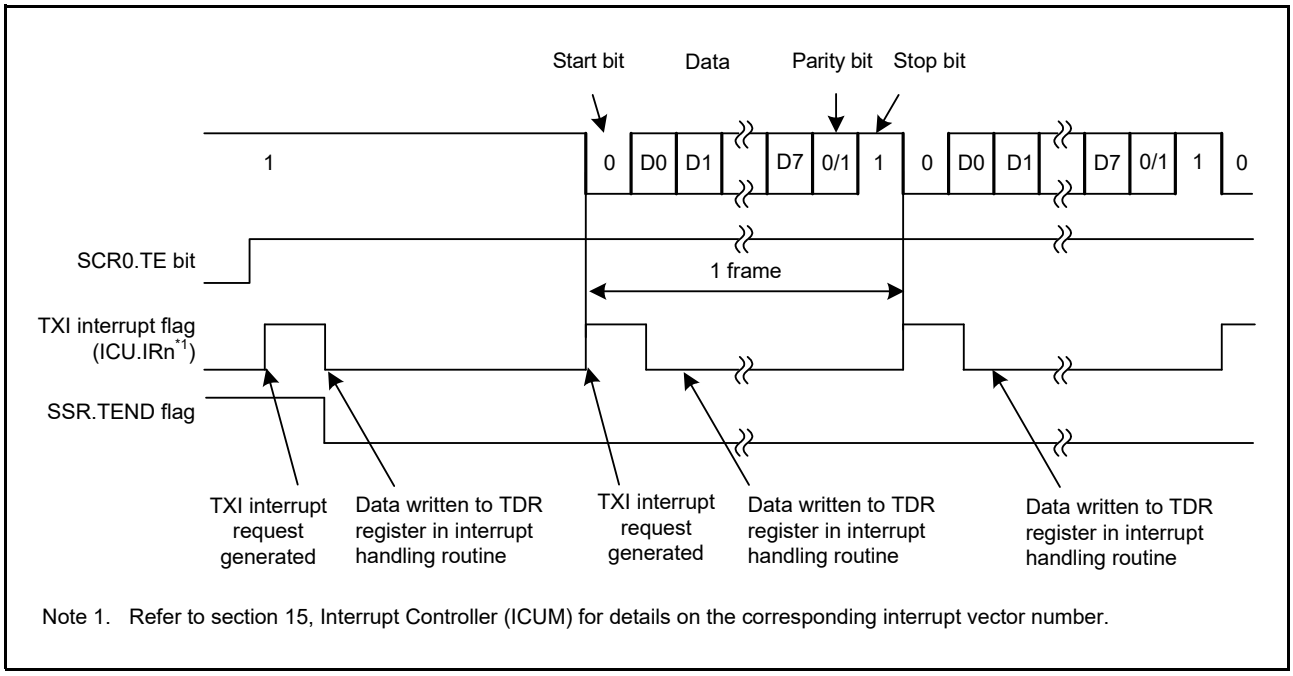
42.3.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO Mode

Figure 42.12 to Figure 42.14 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the RSCI operates as described below.

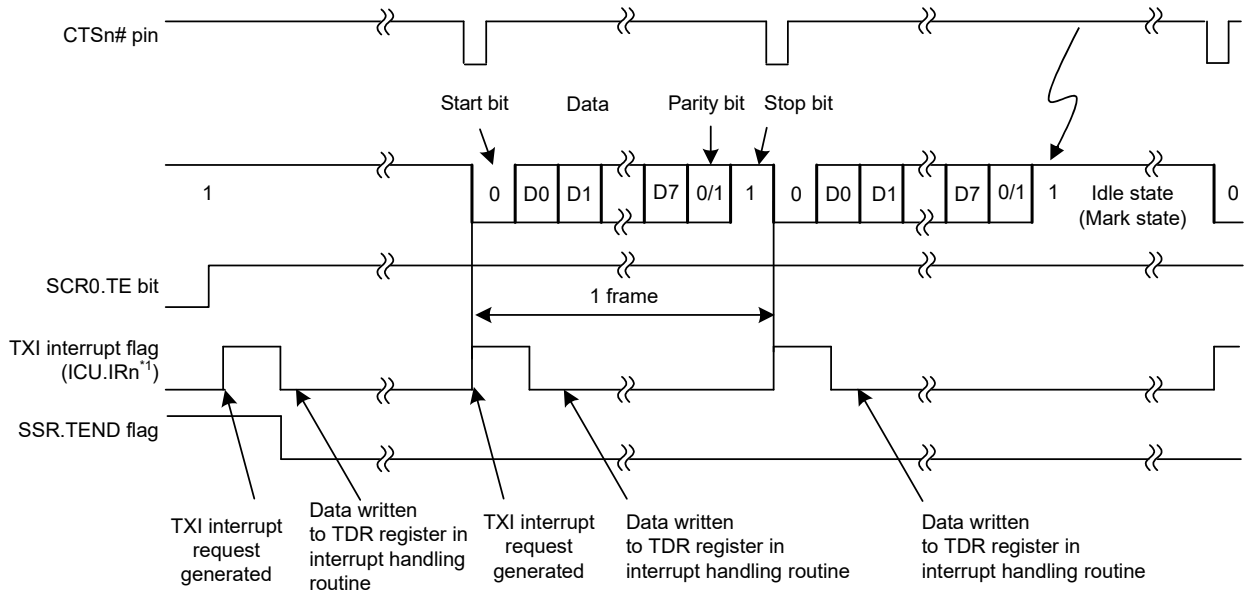
1. The RSCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. At the beginning of transmission, set 1 to the SCR0.TE bit and the SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from the TDR register to the TSR register when the SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. If the SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The RSCI checks for updating of (writing to) the TDR register at the time of stop bit output.
5. When the TDR register is updated, setting of the SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 42.16 shows the example of Serial Transmission Flowchart.



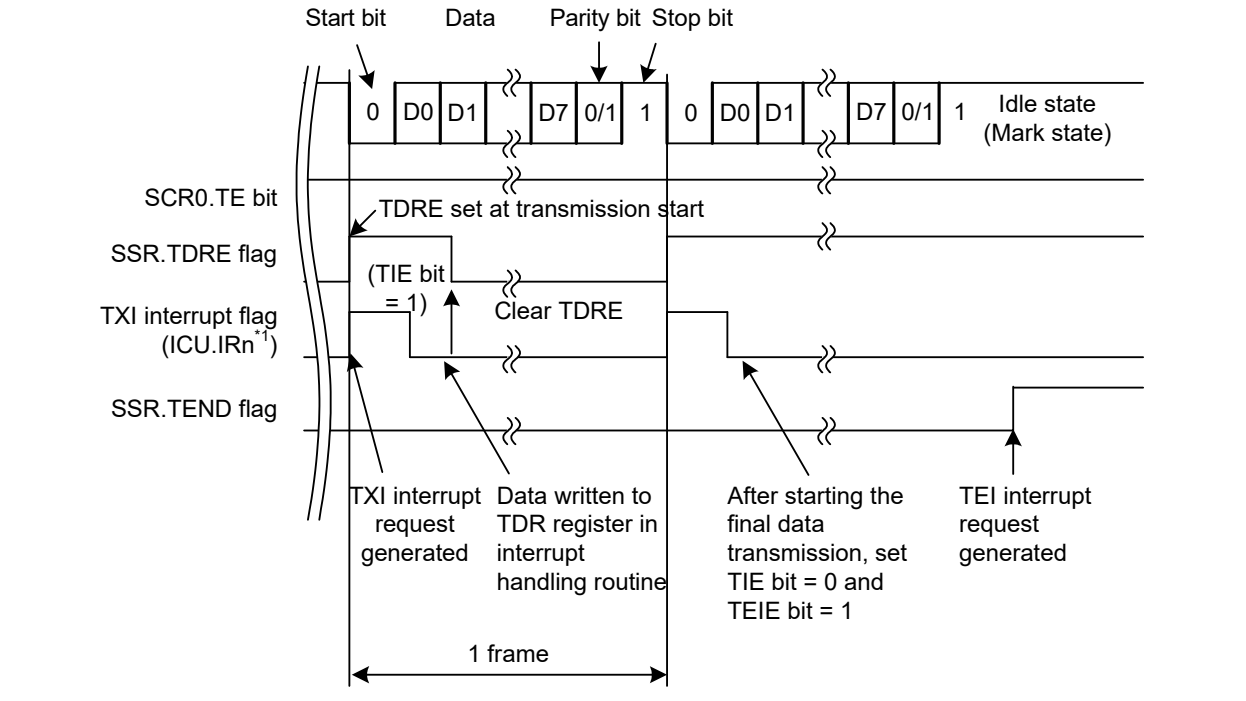
Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

**Figure 42.12 Example of Operation for Serial Transmission in Asynchronous Mode (1)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)**



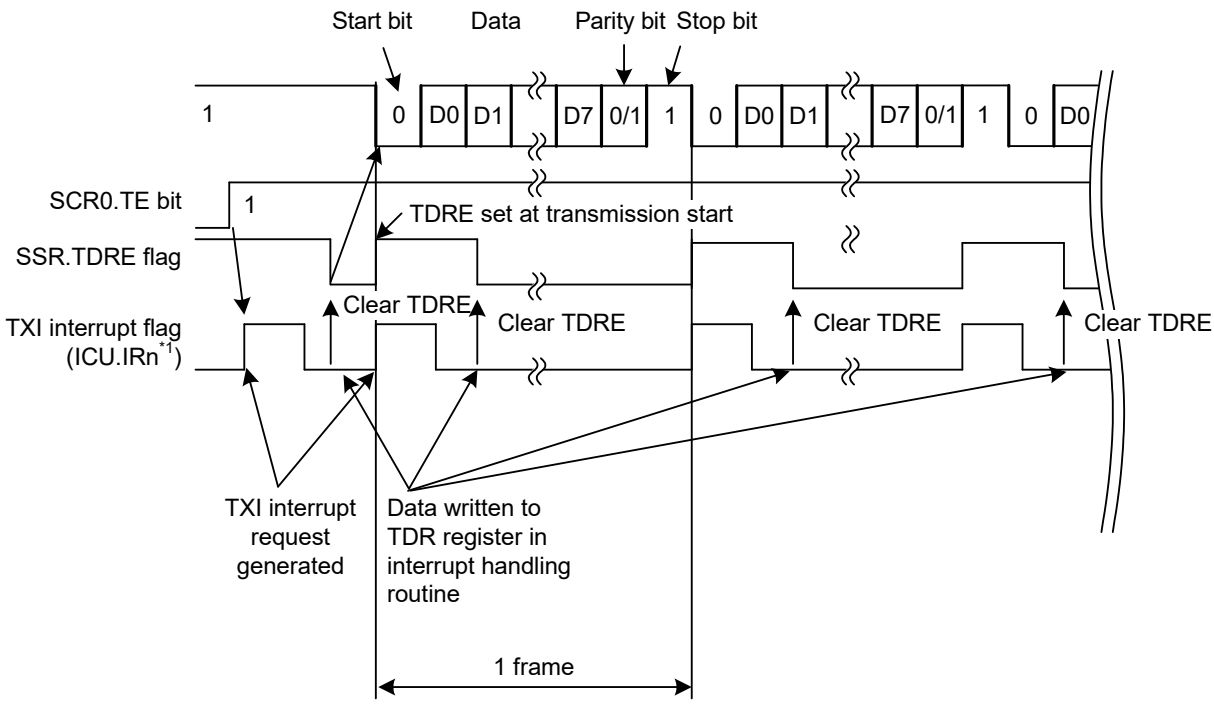
Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.13 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.14 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.15 Example of Operation for Serial Transmission in Asynchronous Mode (4) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, during Transmission)

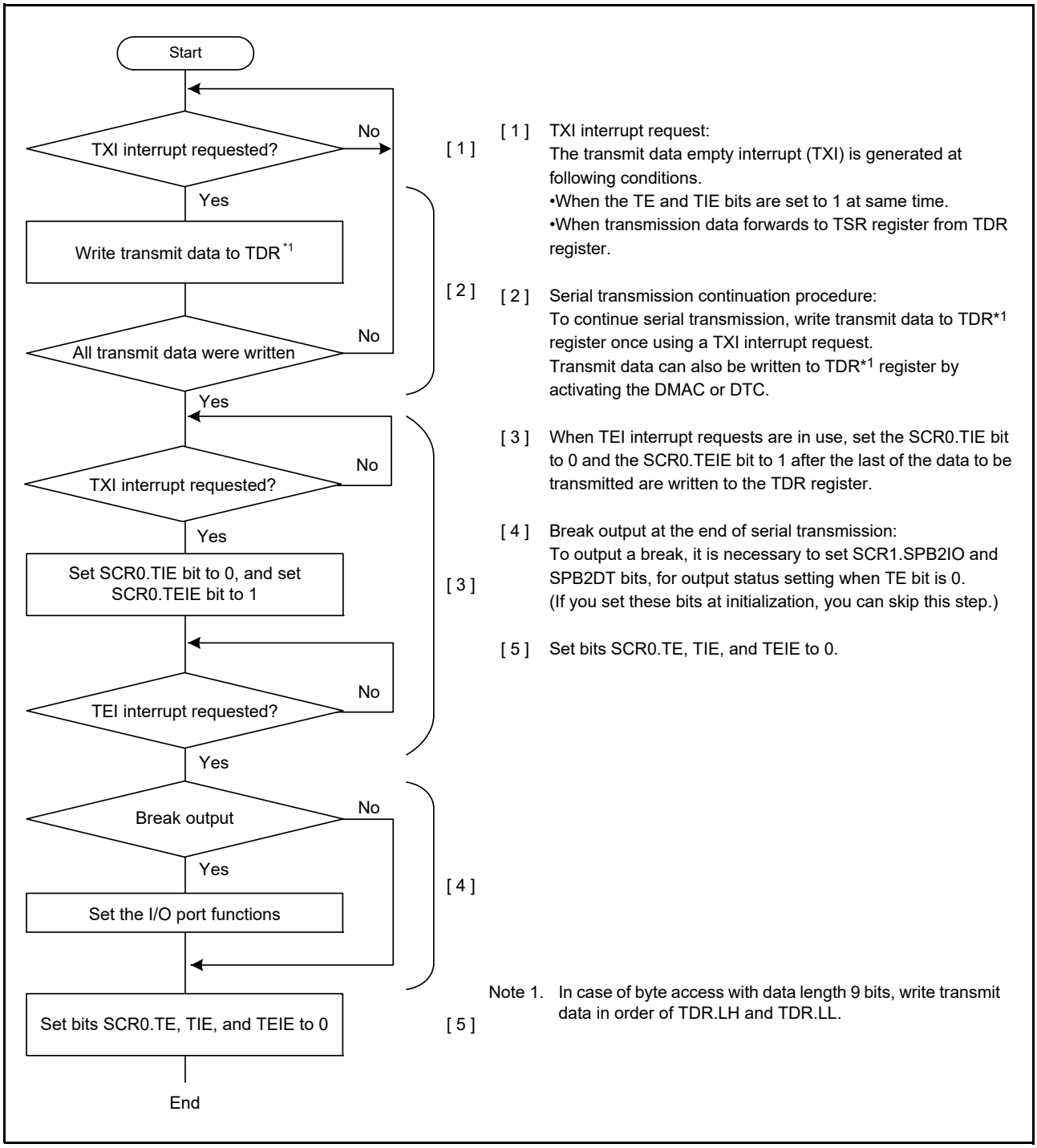


Figure 42.16 Example of Serial Transmission Flowchart in Asynchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Table 42.29 shows an example of data format that is written to transmit FIFO (TDR register) in asynchronous mode with FIFO selected.

Write the MPBT bit to bit 9 of the transmit FIFO (TDR register). When 7-bit data length is selected, write data to the TDR.TDAT[6:0] bits; when 8-bit data length is selected, write data to the TDR.TDAT[7:0] bits; when 9-bit data length is selected, write data to the TDR.TDAT[8:0] bits. Write 0 to unused bits. When accessing in byte units, write data in the order of the TDR.LH to the TDR.LL.

Table 42.29 Data Format That is Written to Transmit FIFO (TDR) (FIFO Mode)

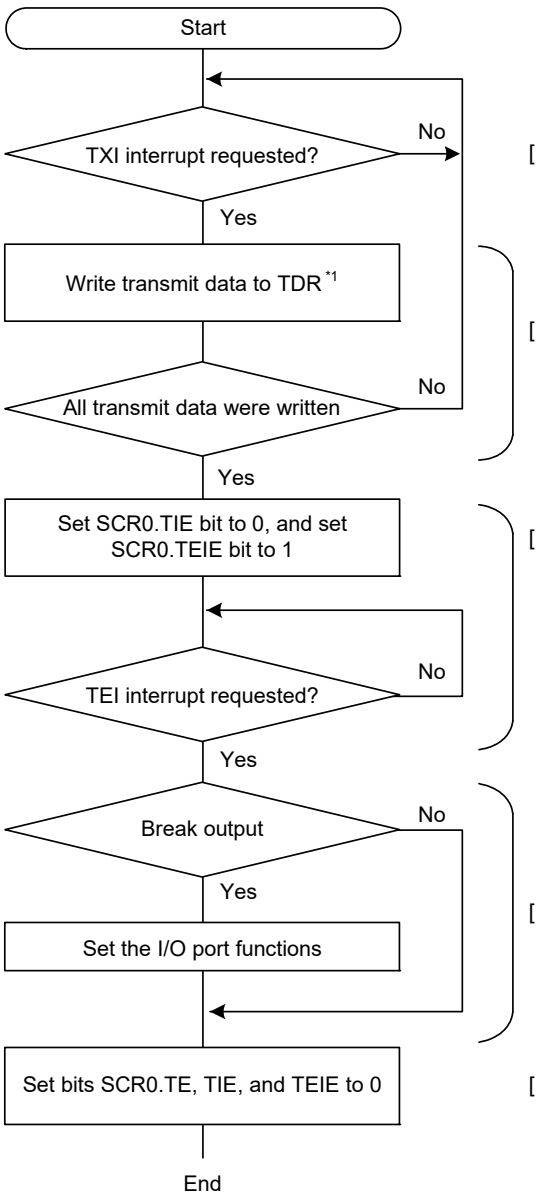
Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

In serial transmission, the RSCI operates as described below.

1. The RSCI transfers data from transmit FIFO (TDR register) to TSR register when data is written to transmit FIFO (TDR register) in the TXI interrupt handling routine. The writable transmit data number is until (16 – transmit FIFO (TDR register)) bytes. At the beginning of transmission, set 1 to SCR0.TE and SCR0.TIE bits simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from transmit FIFO (TDR register) to TSR register when SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. When the quantity of transmit data written in transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number, SSR.TDRE flag is set to 1. If SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to transmit FIFO (TDR register) in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The RSCI checks whether non-transmitted data in transmit FIFO (TDR register) or not at the time of stop bit output.
5. When data is set to transmit FIFO (TDR register), setting of SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from transmit FIFO (TDR register) to TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set to transmit FIFO (TDR register), SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 42.17 shows a sample flowchart for serial transmission in asynchronous mode at FIFO selected.



[1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated at following conditions.
•When the TE and TIE bits are set to 1 at same time.
•When transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit FIFO (TDR register)*1 by using a TXI interrupt and clear the SSR.TDRE flag to 0. (Transmit data can also be written to the transmit FIFO (TDR register)*1 by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 16 (the number of stored transmit FIFO data).

[3] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[4] Break output at the end of serial transmission:
To output a break, it is necessary to set SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[5] Set bits SCR0.TE, TIE, and TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR.LH and TDR.LL.

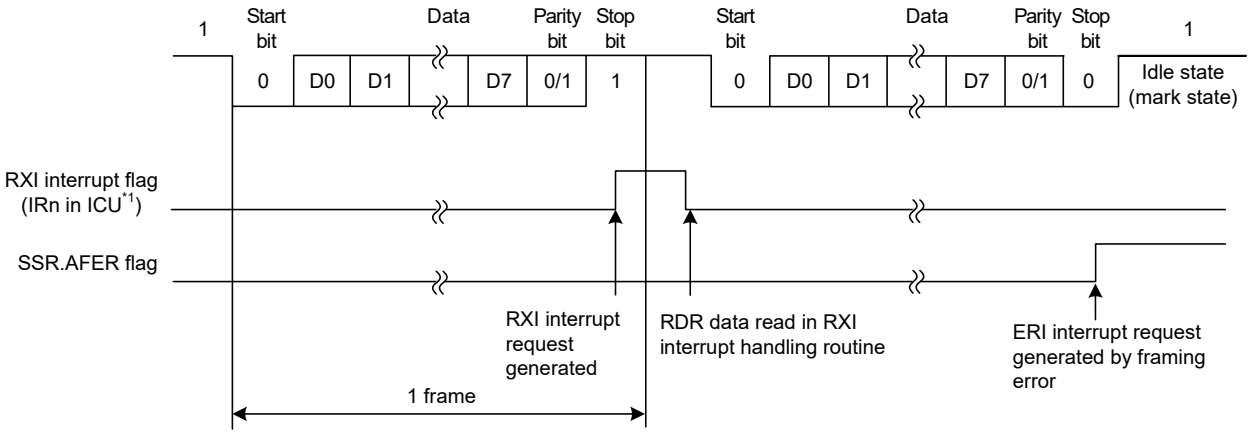
Figure 42.17 Example of Serial Transmission Flowchart in Asynchronous Mode (FIFO Mode)

42.3.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO Mode

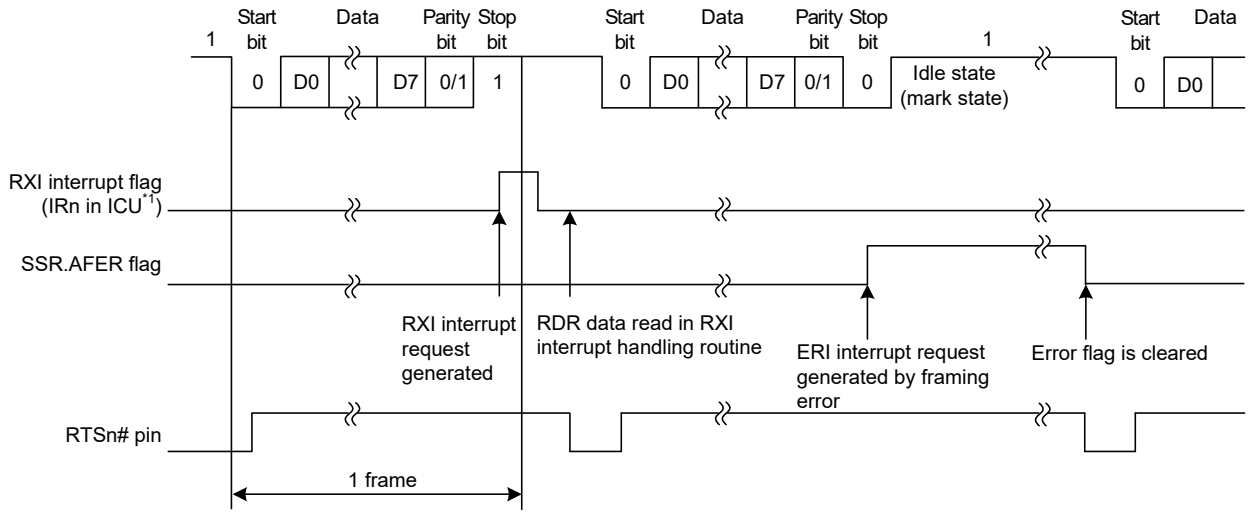
Figure 42.18 and Figure 42.19 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the RSCI operates as described below.

1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR register.
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to RDR register. If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR register causes the RTSn# pin to output the low level in the case of RTS function use. If you do not want to turn the RTSn# pin output low after receiving the last data, set SCR0.RE bit to 0, before reading the RDR register.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.18 Example of RSCI Operation for Serial Reception in Asynchronous Mode (1) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Not Used)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.19 Example of RSCI Operation for Serial Reception in Asynchronous Mode (2) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Used)

Table 42.30 lists the states of the flags in SSR status register and receive data handling when a receive error is detected. If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, and APER flags to 0 before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR register.
Figure 42.20 and Figure 42.21 show samples of flowcharts for serial data reception.

Table 42.30 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	**AFER**	**APER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

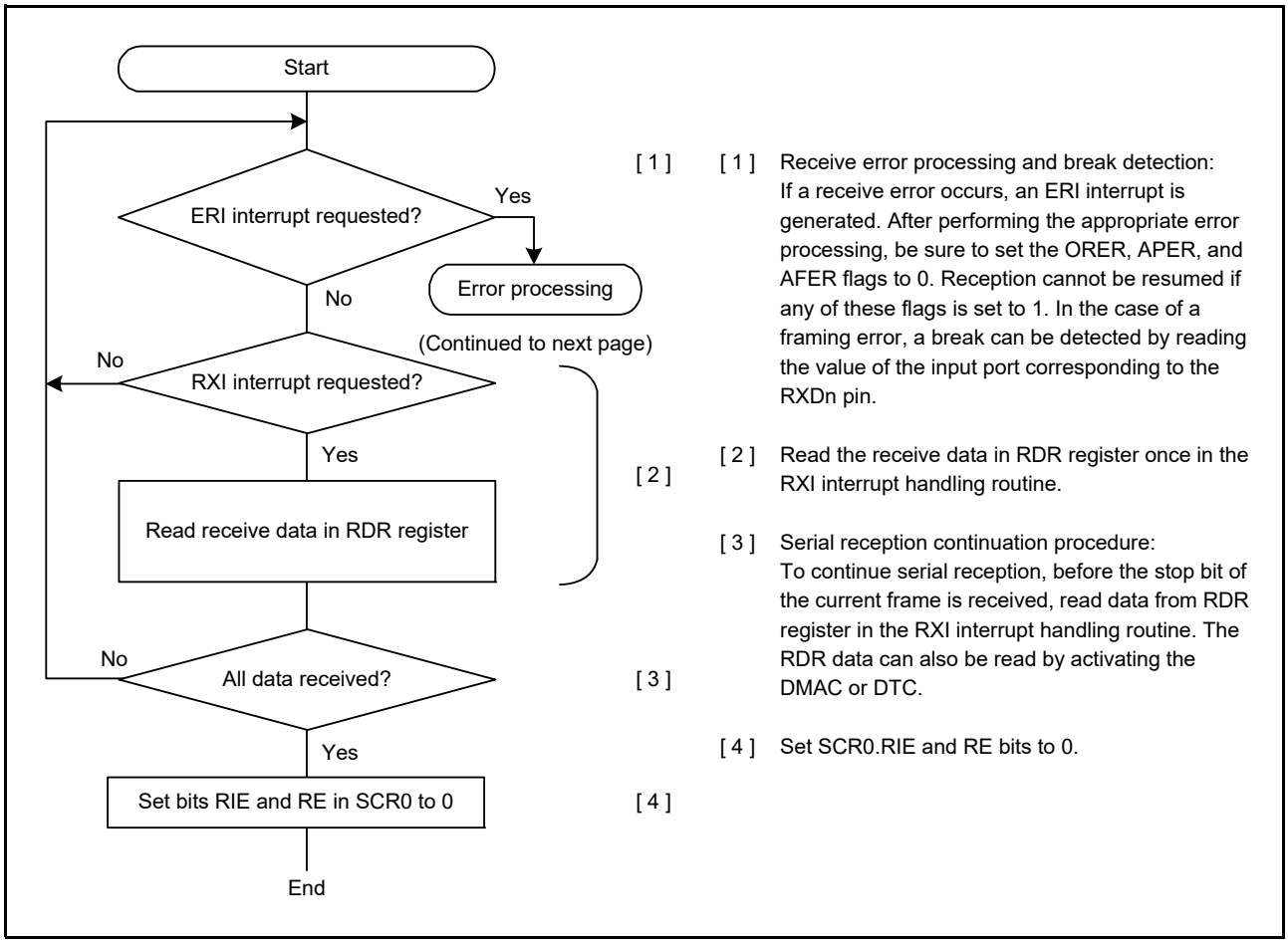


Figure 42.20 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (1)

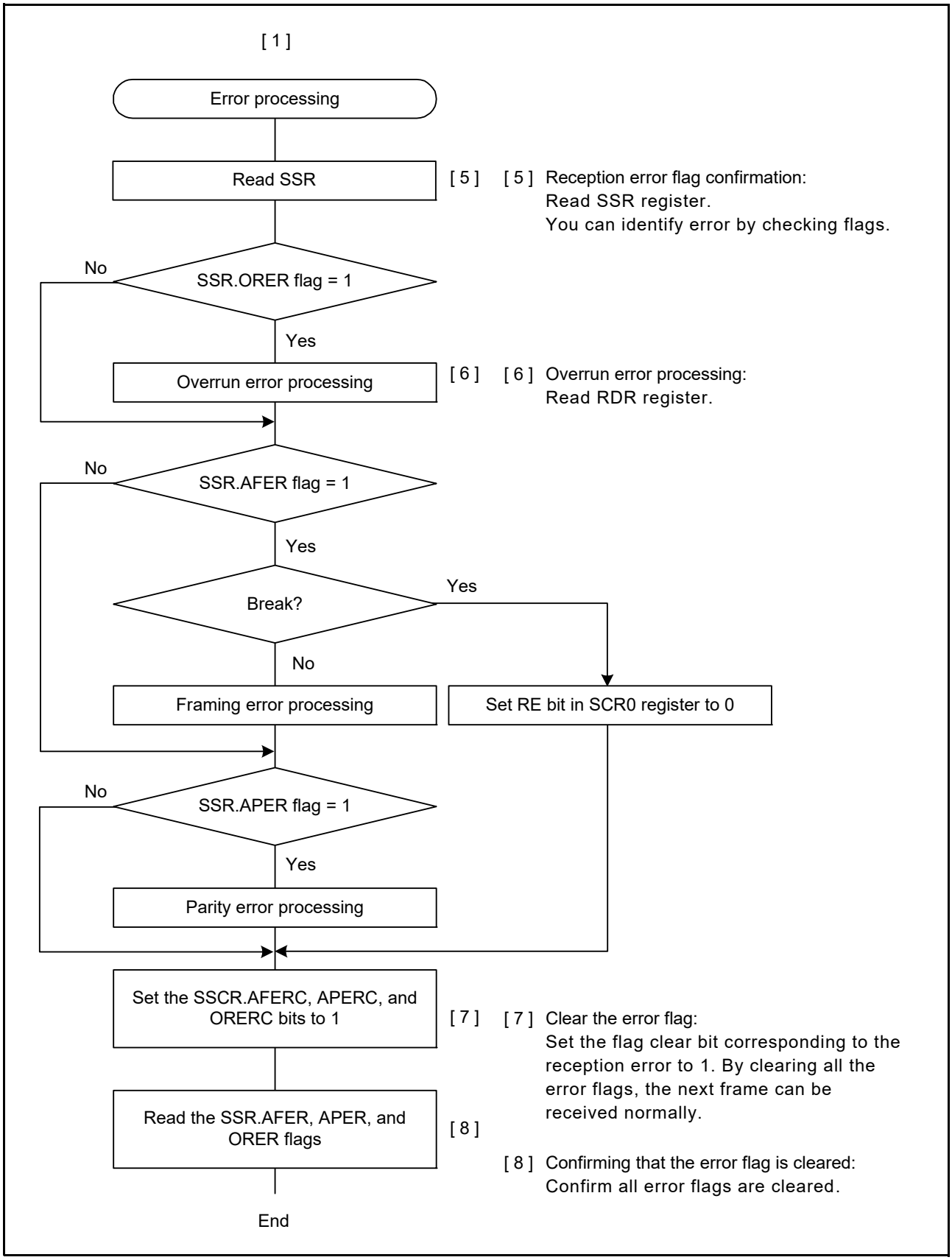


Figure 42.21 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (2)

(2) FIFO Mode

Table 42.31 shows an example of data format stored to receive FIFO (RDR register) in asynchronous mode.
MPB flag (Receive FIFO (RDR register) bit9) is stored 0. Data is stored to receive FIFO (RDR register) corresponded to data length. It is stored to 0 for unused bits. If receive FIFO (RDR register) is read, RSCI updates to next data which are FER flag, PER flag, and receive data (RDAT[8:0] bits) in receive FIFO. The flags which are AFER, APER, ORER, and DR flag in receive FIFO, are always indicated to the flags corresponded to SSR register and RFSR register.

Table 42.31 Data Format Stored in the Receive FIFO (RDR) (FIFO Mode)

Data Length	Register setting		Arrangement of Receive Flag, MPB Flag and Received Data in RDR Register															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	—	—	—	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	—	—	—	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3**	**CHR[1:0]**	**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
8 bits	1	0	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
9 bits	0	0 or 1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—

Note: 0 is always read from the MPB flag (RDR register bit9).
When a 7-bit data length is selected, 0 is read from the RDAT[8:7] bits.
When a 8-bit data length is selected, 0 is read from the RDAT[8] bit.

Table 42.32 lists the states of the flags in SSR register status register and receive data handling when a receive error is detected in FIFO mode. Figure 42.22 and Figure 42.23 show samples of flowcharts for serial data reception in FIFO mode.
In serial data reception, the RSCI operates as described below.

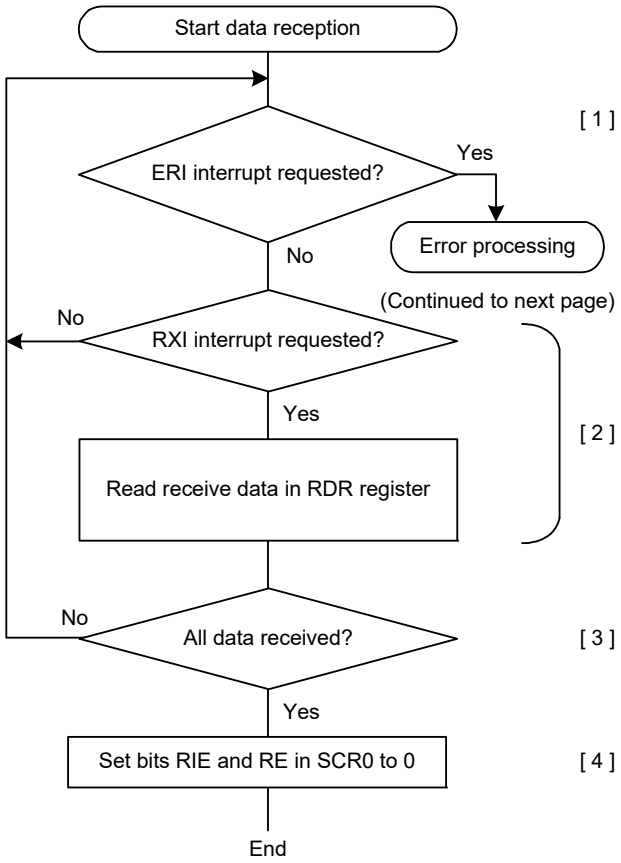
1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If there is no space in receive FIFO (RDR register), an overrun error occurs and the SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to receive FIFO (RDR register).
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to receive FIFO (RDR register). If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. After framing error is detected, when RSCI detects that continuous receive data is 0 for 1 frame, the reception stops.
7. When quantity of data stored in the receive FIFO data register (RDR register) falls the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode, RFSR.DR flag is set to 1. If RIE bit is set to 1, RSCI occurs RXI interrupt request when FCR.DRES bit is 0 and RSCI occurs ERI interrupt request when FCR.DRES bit is 1.
8. When reception finishes successfully, receive data is transferred to receive FIFO (RDR register). The SSR.RDRF flag is set to 1 when the quantity of receive data which is equal to or greater than the specified receive triggering number are stored in receive FIFO (RDR register). If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to receive FIFO (RDR register)

in this RXI interrupt handling routine, before overrun error is occurred. Reading the received data that have been transferred to receive FIFO (RDR register), and if it is less than RTS trigger number, causes the RTSn# pin to output the low level. in the case of RTS function use.

Table 42.32 Flags in the SSR Status Register and Receive Data Handling (FIFO Mode)

SSR Register				
ORER	**AFER*1**	**APER*1**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	0	0	Lost	Overrun error + framing error + parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flag to 0. Reception cannot be resumed if ORER flag is set to 1. Reception can continue even if APER, AFER, and DR flags are 1.

[2] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG[4:0] value at least. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[3] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive FIFO (RDR register) in the RXI interrupt handling routine and clear SSR.RDRF flag and RFSR.DR flag to 0. The receive FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[4] Set SCR0.RIE and RE bits to 0.

Figure 42.22 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (1)

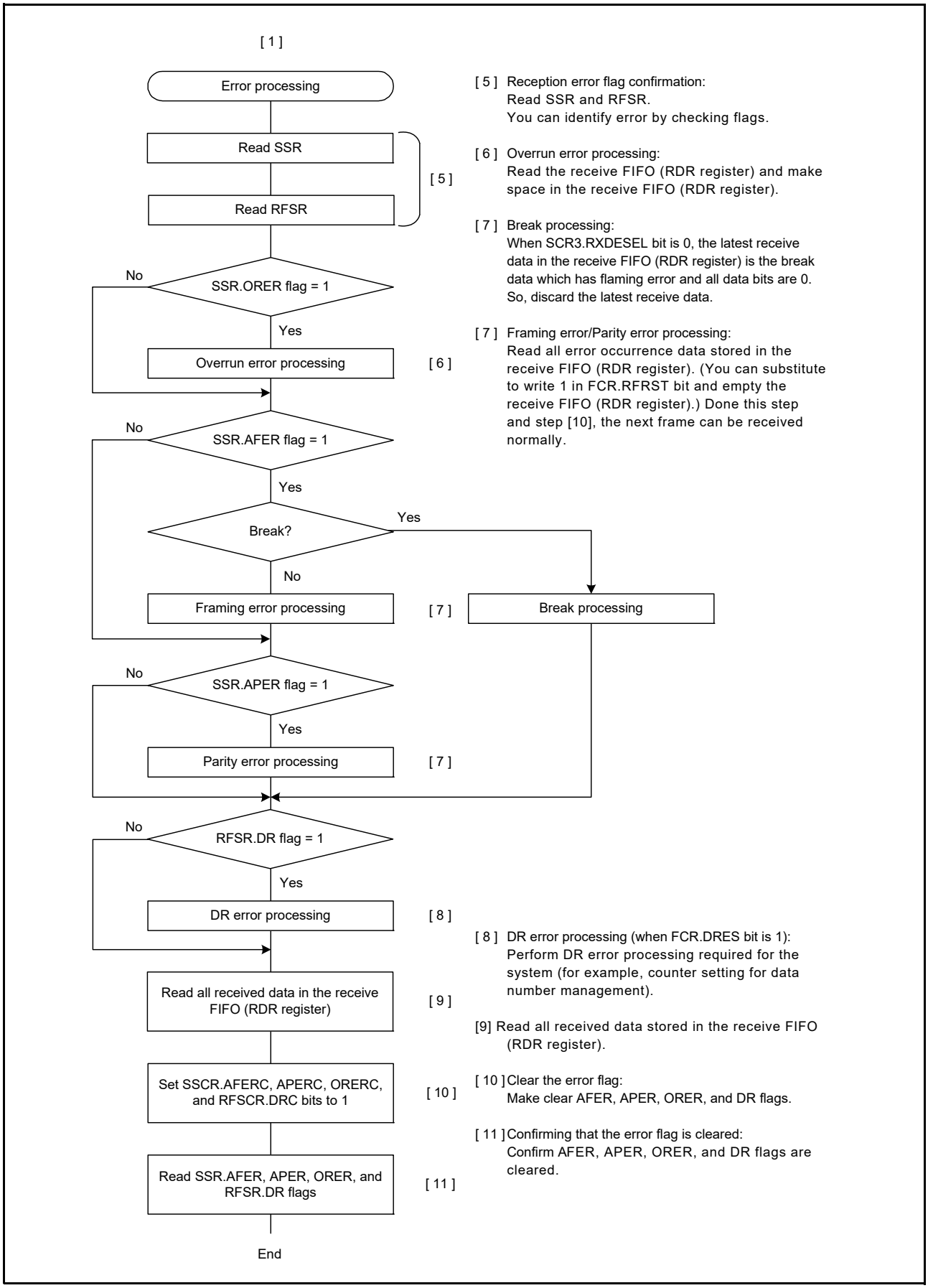


Figure 42.23 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (2)

42.3.10 Receive Data Sampling Timing Adjustment

When a waveform such that high width is different from low width because the difference between rising time and falling time is large is received, adjust the timing for data to be sampled at the center of the narrower pulse. When low width is narrower than high width, move the sampling timing forward, and when high width is narrower than low width, move the sampling timing backward.

When the SCR4.RTMG[3:0] bits are set to an offset to the default sampling point and then the SCR4.RTADJ bit is set to 1, received data is sampled at the specified position.

Figure 42.24 shows an example of the sampling timing adjustment.

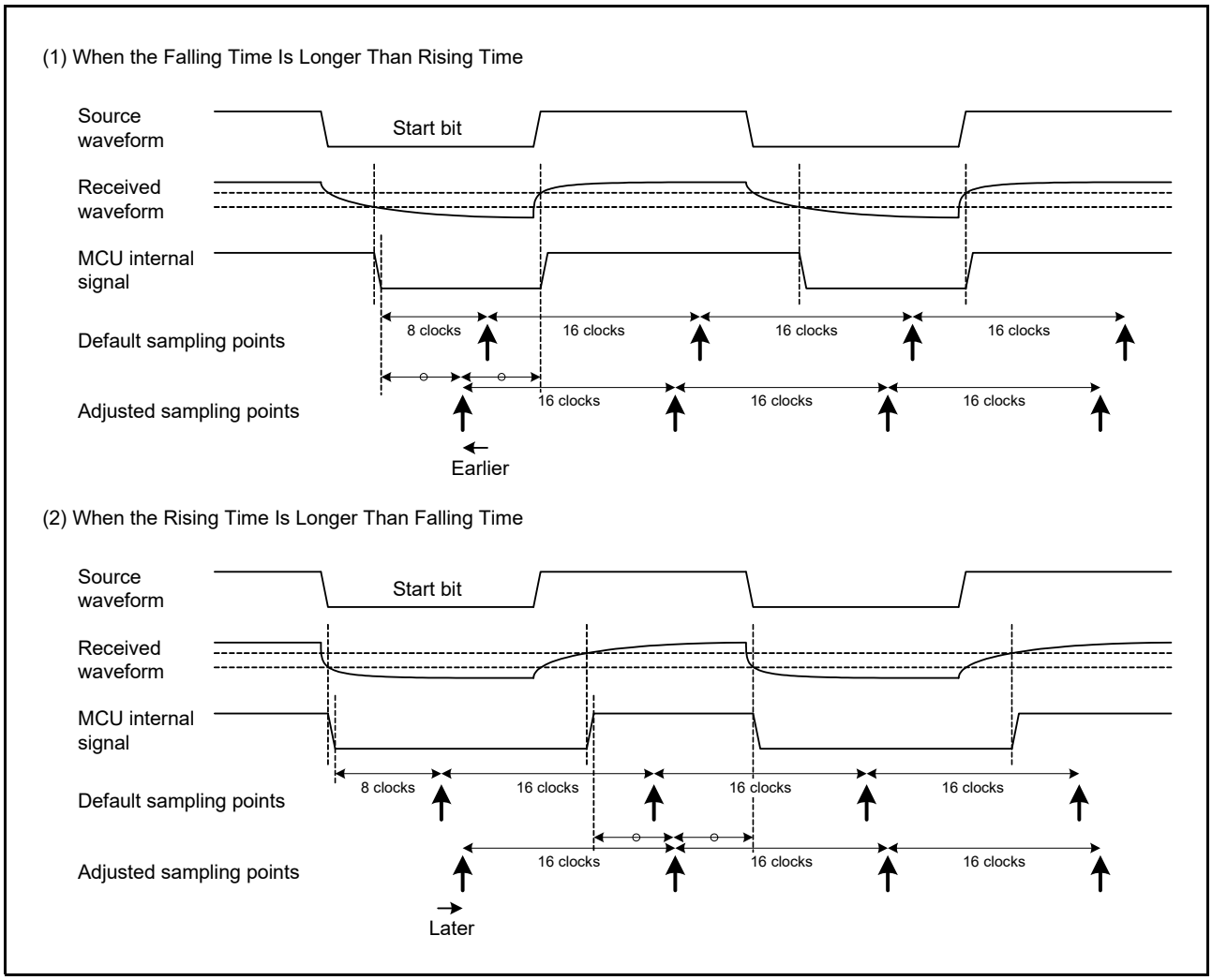


Figure 42.24 Example of Sampling Timing Adjustment (SCR2.ABCSE2 Bit = 0, SCR2.ABCSE Bit = 0, and SCR2.ABCS Bit = 0)

42.3.11 Transmit Data Transition Timing Adjustment

When a difference between high width and low width for a waveform transmitted by this MCU arises at the receiver, it is also possible to make a difference between the high width and the low width beforehand at transmission to adjust so that the difference disappears at the receiver. When high width becomes narrower at the receiver, expand the high width of the transmit signal by delaying the falling edges, and when low width becomes narrower at the receiver, expand the low width of the transmit signal by delaying the rising edges.

When the SCR4.TTMG[3:0] bits are set to the transition direction and the delay amount and the SCR4.TTADJ bit is set to 1, transmit data changes at the specified point.

Figure 42.25 shows an example of the transition timing adjustment.

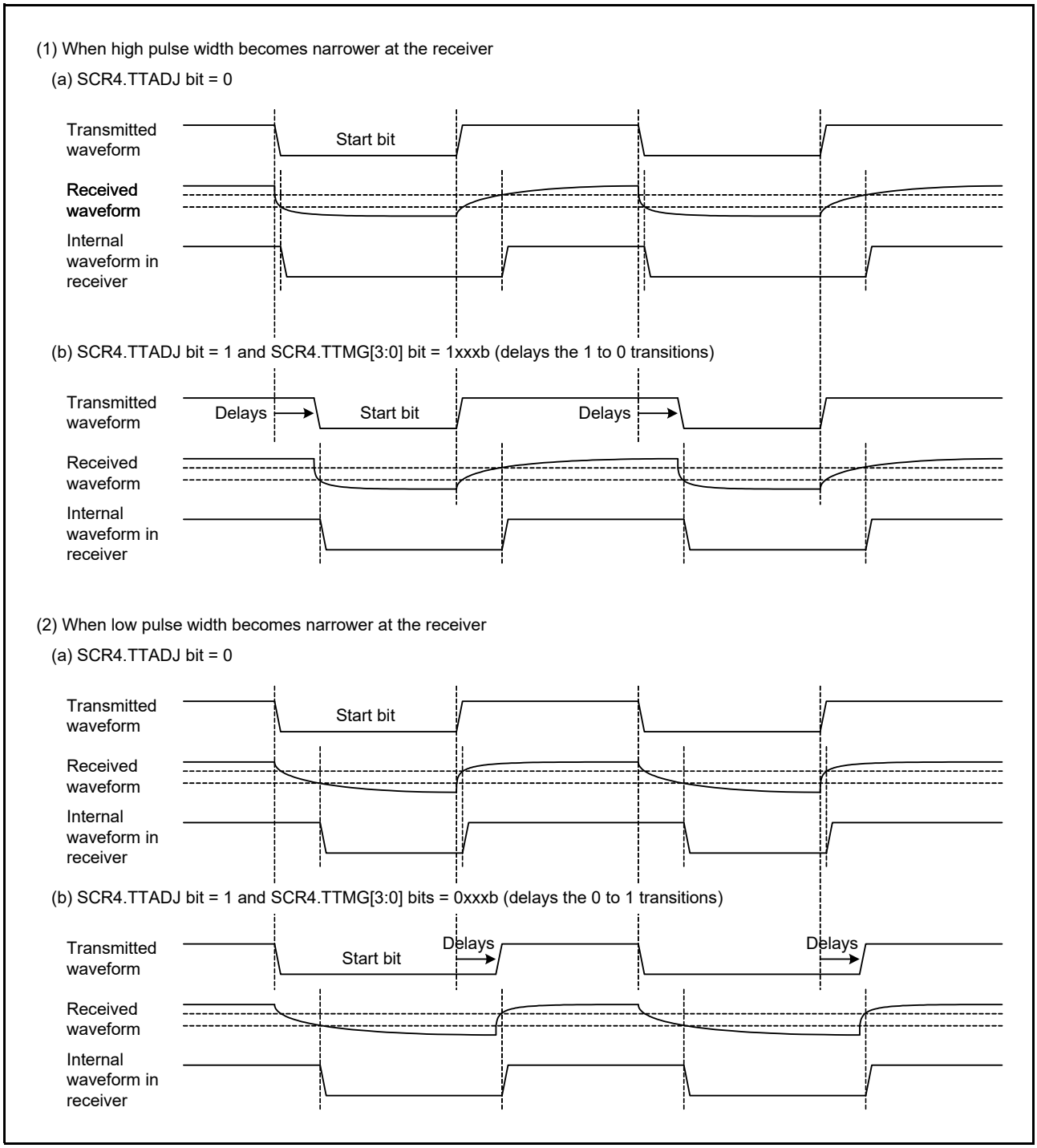


Figure 42.25 Example of Transition Timing Adjustment

42.4 Multi-Processor Communication Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multiprocessor bit is set to 0, it indicates the data transmission cycle. Figure 42.26 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two matches, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.
RTS control cannot be used at the time of multi-processor communication function use, because this is a function corresponding to one-to-many communications.

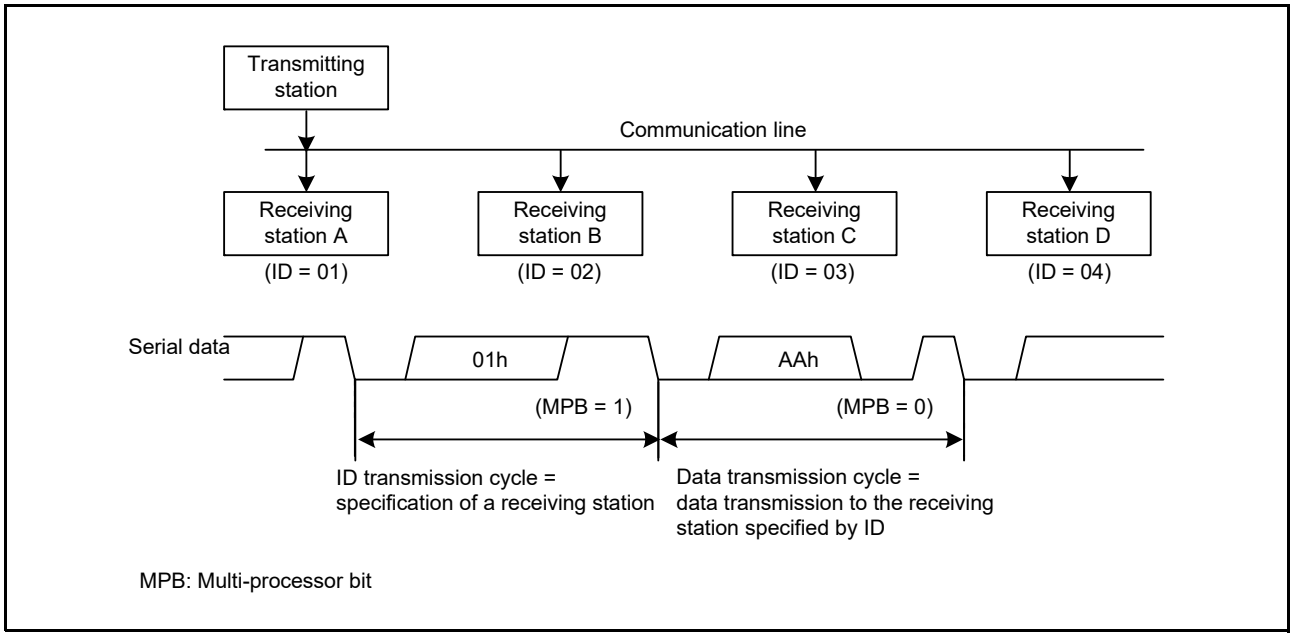


Figure 42.26 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(1) Non-FIFO Mode

For supporting this function, the RSCI provides the MPIE bit in SCR0 register. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register, detection of a receive error, and setting the respective status flags RDRF, ORER and AFER in SSR are disabled until reception of data in which the multi-processor bit is set to 1.
Upon receiving a reception character in which the multi-processor bit is set to 1, the MPB flag in RDR register is set to 1 and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register is set.
When the multi-processor format is specified, specification of the parity bit is disabled.
Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the non-multi-processor asynchronous mode.

(2) FIFO Mode

For transmission, SW should write to TDR.MPBT (Multi-Processor Bit Transfer) which corresponds to transmit data in TDR.TDAT[8:0] bits. For reception, multi-processor bit that is a part of receive data, is stored to RDR.MPB flag, and receive data is stored to RDR.RDAT[8:0] bits. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR.RDAT[8:0] bits, detection of a receive error, and detection of DR flag, and setting the respective status flags RDRF, ORER, and AFER in SSR are disabled until reception of data in which the multi-processor bit is set to 1.

Upon receiving a reception character in which the multi-processor bit is set to 1, the MPB flag in RDR register is set to 1,and receive data is stored to receive FIFO (RDR.RDAT[8:0] bits), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode and FIFO mode.

42.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO Mode

Figure 42.27 shows a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

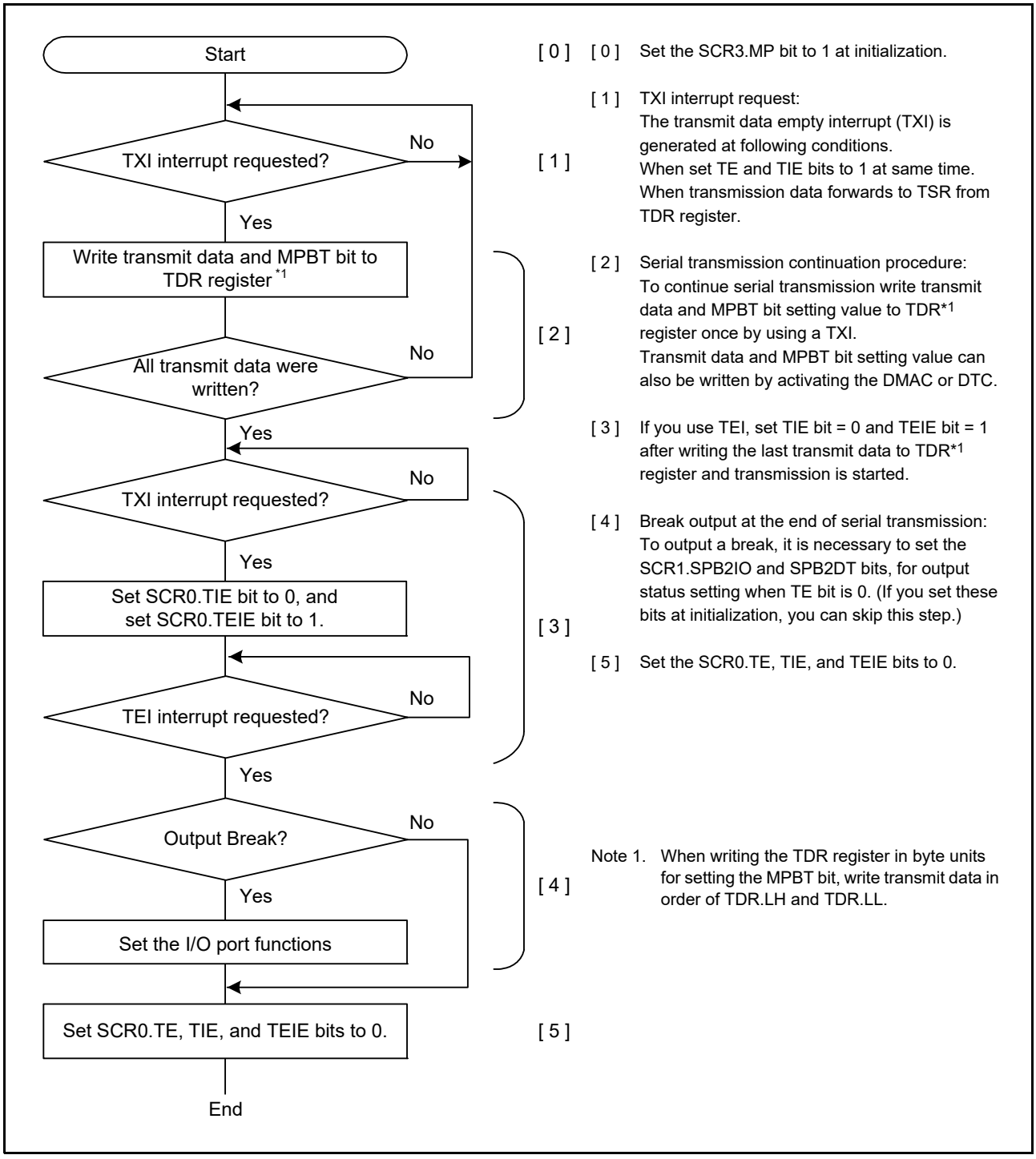


Figure 42.27 Example of Multi-Processor Serial Transmission Flowchart (Non-FIFO Mode)

(2) FIFO Mode

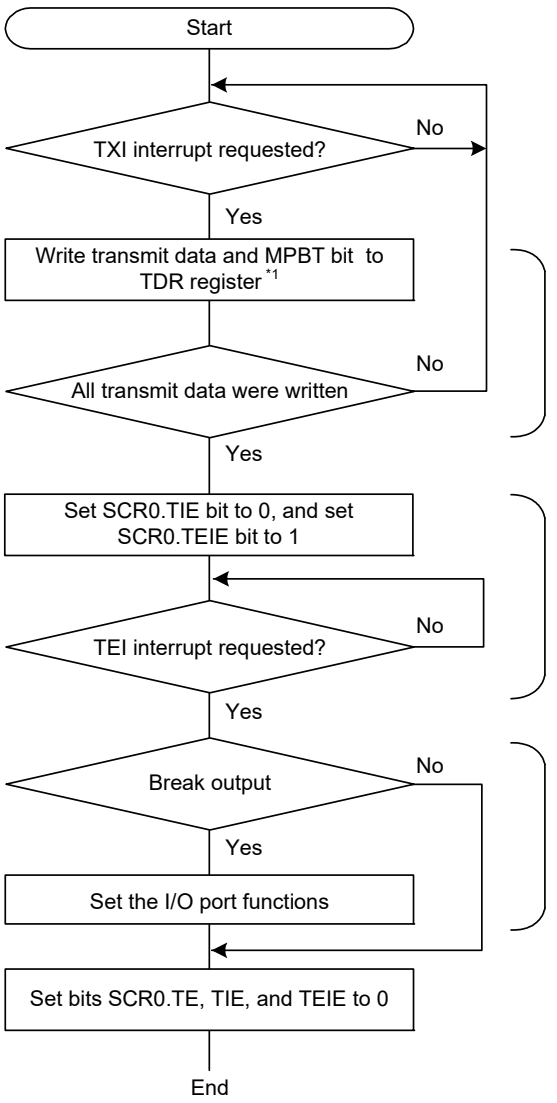
Table 42.33 shows an example of data format that is written to transmit FIFO (TDR register) in multi-processor mode. Write MPBT in bit9 of TDR. And write data to transmit FIFO (TDR register) corresponded to data length. It should write to 0 for unused bits.

Table 42.33 Data Format in Multi-processor Mode that is Written to Transmit FIFO (TDR Register)

Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

Figure 42.28 shows a sample flowchart for multi-processor data transmission in FIFO mode. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1 at initialization.

[1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT bit setting value to the transmit FIFO (TDR register) by using a TXI interrupt. Transmit data and MPBT bit setting value can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC.

[4] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set the SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set bits SCR0.TE, TIE, and TEIE to 0.

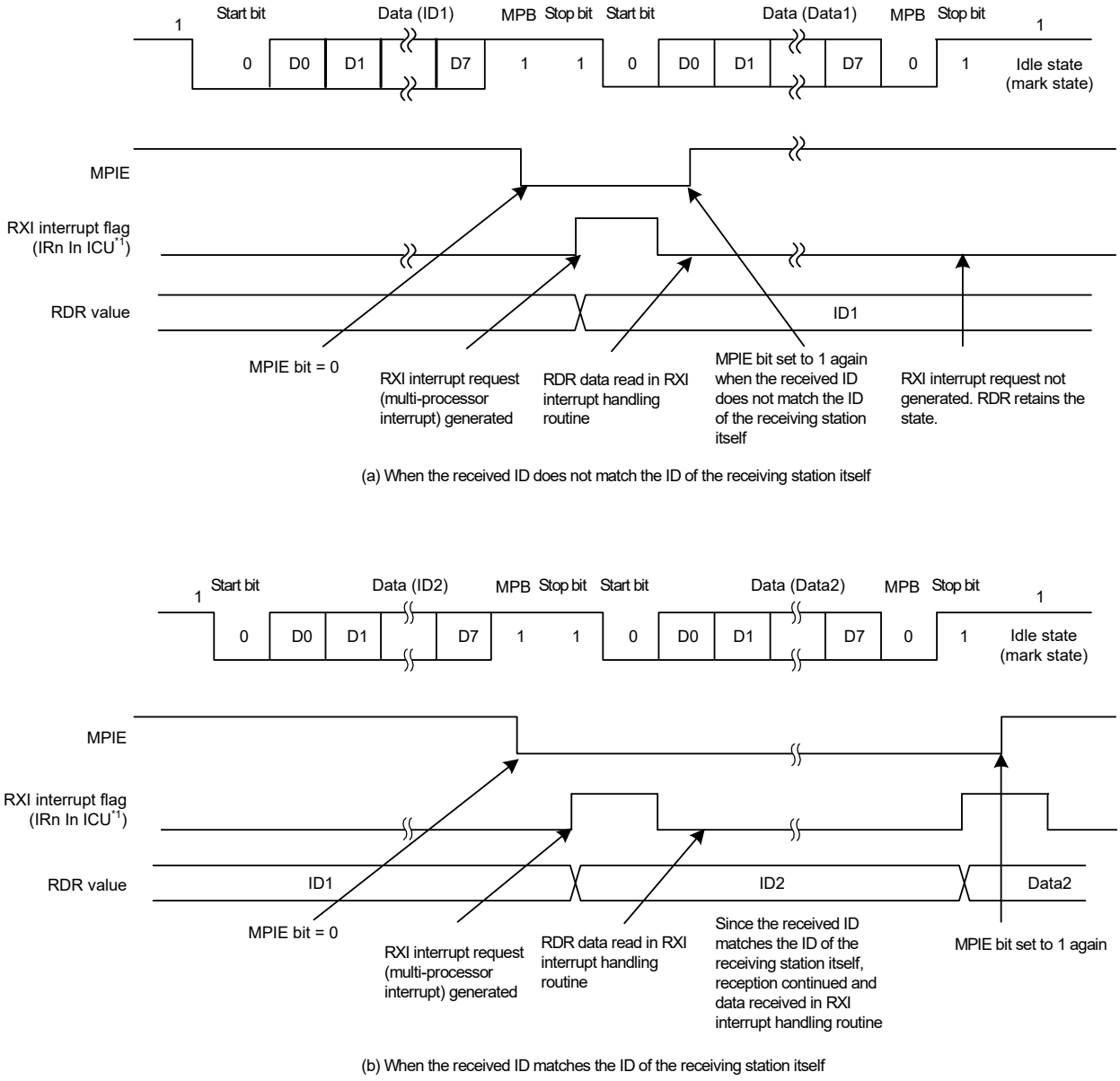
Note 1. When writing the TDR register in byte units for setting the MPBT bit, write transmit data in order of TDR.LH and TDR.LL.

Figure 42.28 Example of Multi-Processor Serial Transmission Flowchart (FIFO Mode)

42.4.2 Multi-Processor Serial Data Reception

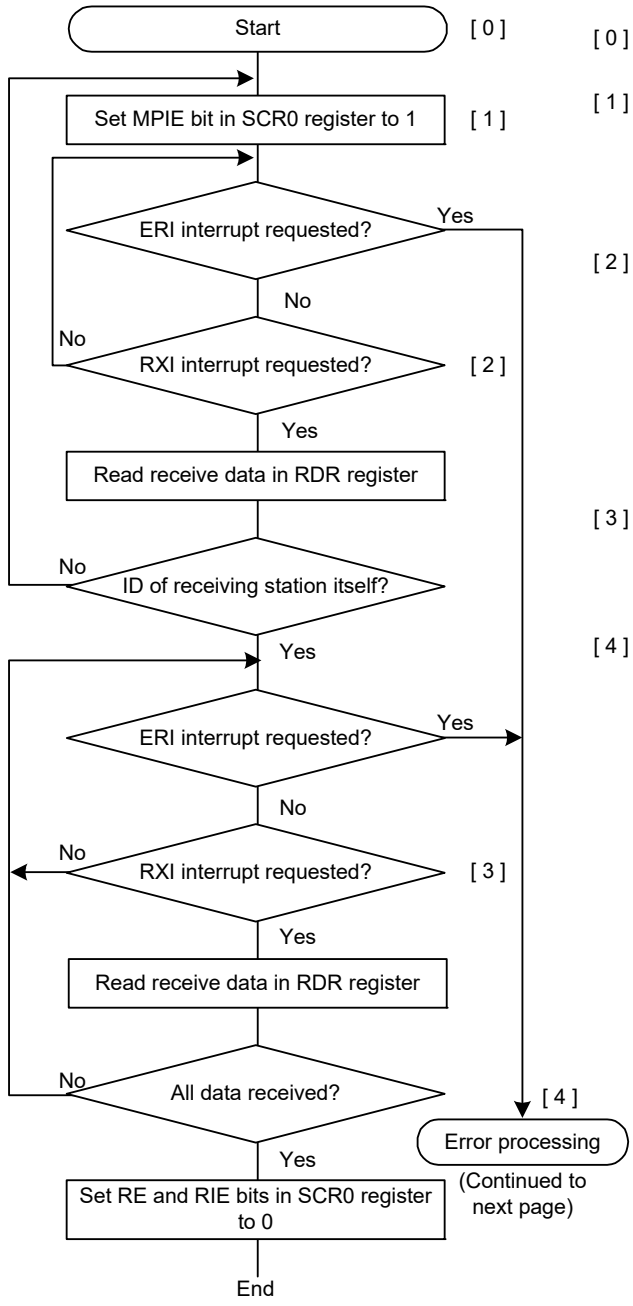
(1) Non-FIFO Mode

Figure 42.30 and Figure 42.31 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR0 register is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR register. At this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 42.29 is the example of operation for reception.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.29 Example of RSCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



[0] Set the SCR3.MP bit to 1 at initialization.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1 and wait for ID reception.
If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
Read data in RDR register at the first RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[3] Data reception at an RXI interrupt:
Read data in RDR register once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR. After performing the appropriate error processing be sure to clear the ORER and AFER flags. Reception cannot be resumed if any of these flags is set to 1.
In the case of a framing error, a break can be detected by reading SSR.RXDMON flag.

Figure 42.30 Example of Multi-Processor Serial Reception Flowchart (1) (Non-FIFO Mode)

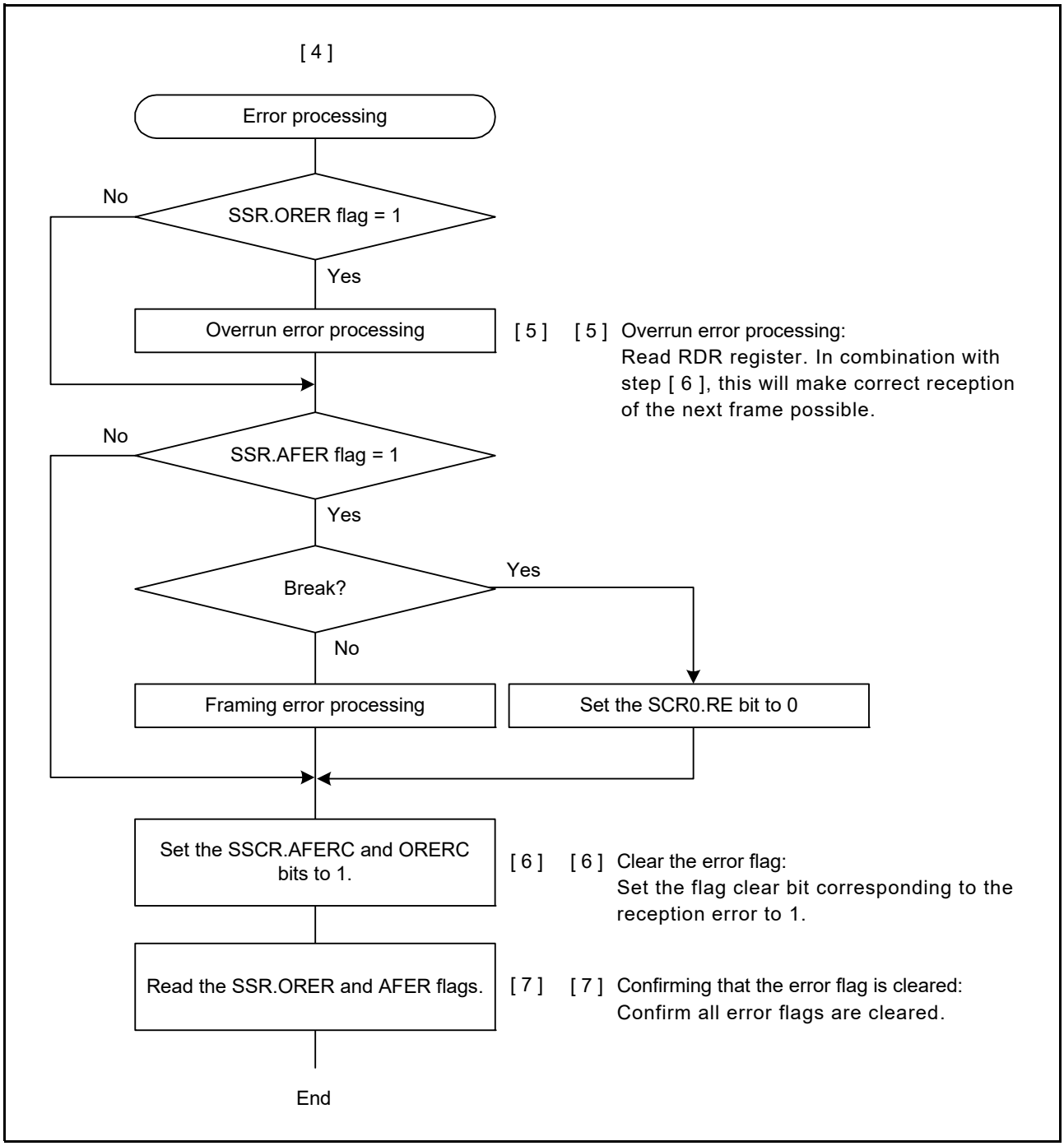


Figure 42.31 Example of Multi-Processor Serial Reception Flowchart (2) (Non-FIFO Mode)

(2) FIFO Mode

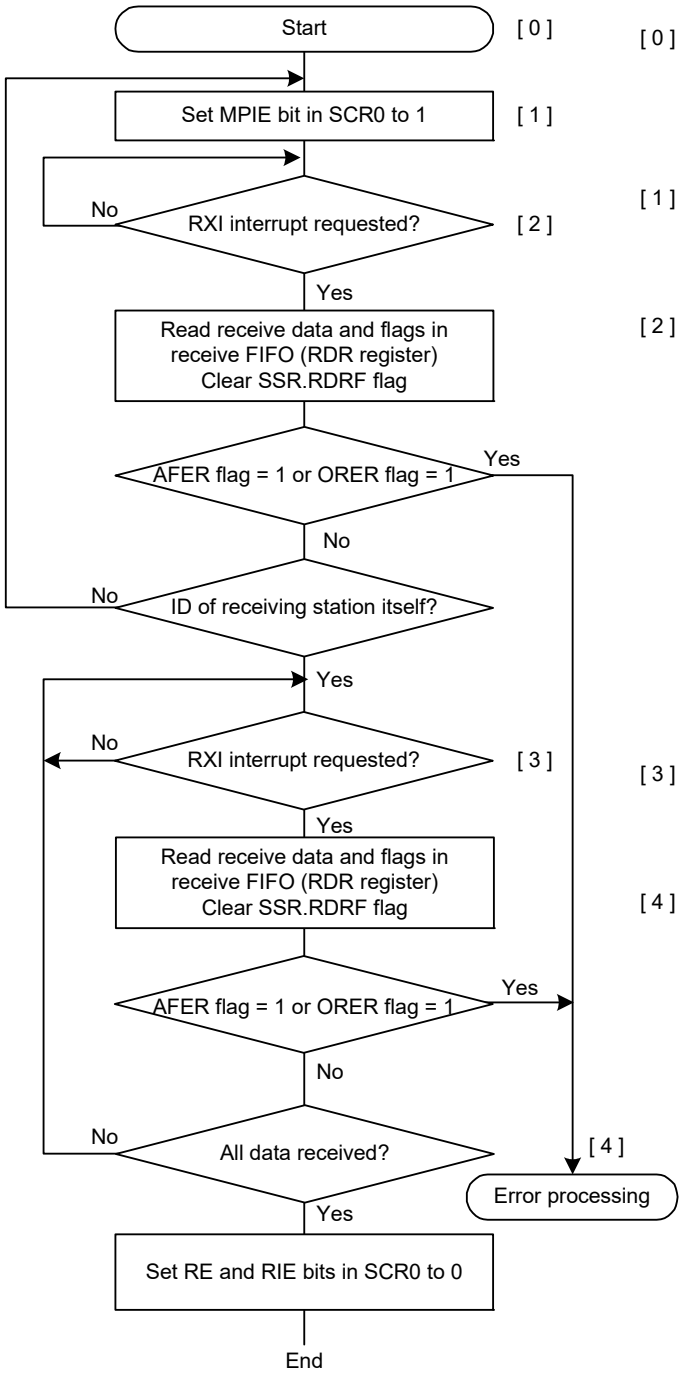
Table 42.34 shows an example of data format that is stored to receive FIFO (RDR register) in multi-processor mode. MPB flag is stored in bit9 of RDR register. 0 is stored to APER and PER flags. Data is stored to receive FIFO RDAT[8:0] bits corresponded to data length. It is stored to 0 for unused bits. Reading the receive FIFO (RDR register) updates the FER, PER, MPB flags and receive data (RDAT[8:0] bits) in the receive FIFO (RDR register) with the next received data. The AFER, APER, and ORER flags in the receive FIFO (RDR register) always reflect the status of the corresponding flags in the SSR and RFSR registers.

Table 42.34 Data Format in Multi-Processor Mode that is Stored to Receive FIFO (FIFO Mode)

Data Length	Register setting		Received Data in RDR Register															
	SCR3 CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	0	0	0	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	0	0	0	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	0	0	0	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3 CHR[1:0]**		**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
8 bits	1	0	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
9 bits	0	0 or 1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7] bits.
When data length is 8bit, it can read always 0 in RDAT[8] bit.

Figure 42.32 shows a sample flowchart for multi-processor data reception in FIFO mode. When the MPIE bit in SCR0 register is set to 1, reading the communication data is skipped until reception of the communication data in which the multiprocessor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data, MPB flag and each errors are transferred to receive FIFO (RDR register), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a normal reception operation. After a framing error occurred and SSR.AFER flag is set to 1, but RSCI continues data reception.
The other operations are the same as the operations in asynchronous mode with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1.
Also, set the FCR.RTRG[4:0] bits to 00001b and DRES bit to 0 to generate an RXI interrupt request when data with the MPB flag set to 1 is received.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1. And wait for ID reception. If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
RSCI skips the received data until it receives the data with the MPB flag set to 1. When receiving data with the MPB flag set to 1, RSCI generates an RXI interrupt request.
In the RXI interrupt routine, read the data from the receive FIFO (RDR register) and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again and wait for another RXI interrupt request. If it matches the ID of the receiving station itself, set the receive FIFO threshold to the desired value.

[3] Data reception at an RXI interrupt:
Read data in the receive FIFO (RDR register) once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR register (There is no problem reading the SSR.ORER and AFER flags again.). After performing the appropriate error processing be sure to clear the ORER and AFER flags. In the case of a framing error, a break can be detected by reading the SSR.RXDMON flag.

Figure 42.32 Example Flowchart of Serial Reception in Multi-Processor Mode (1) (FIFO Mode)

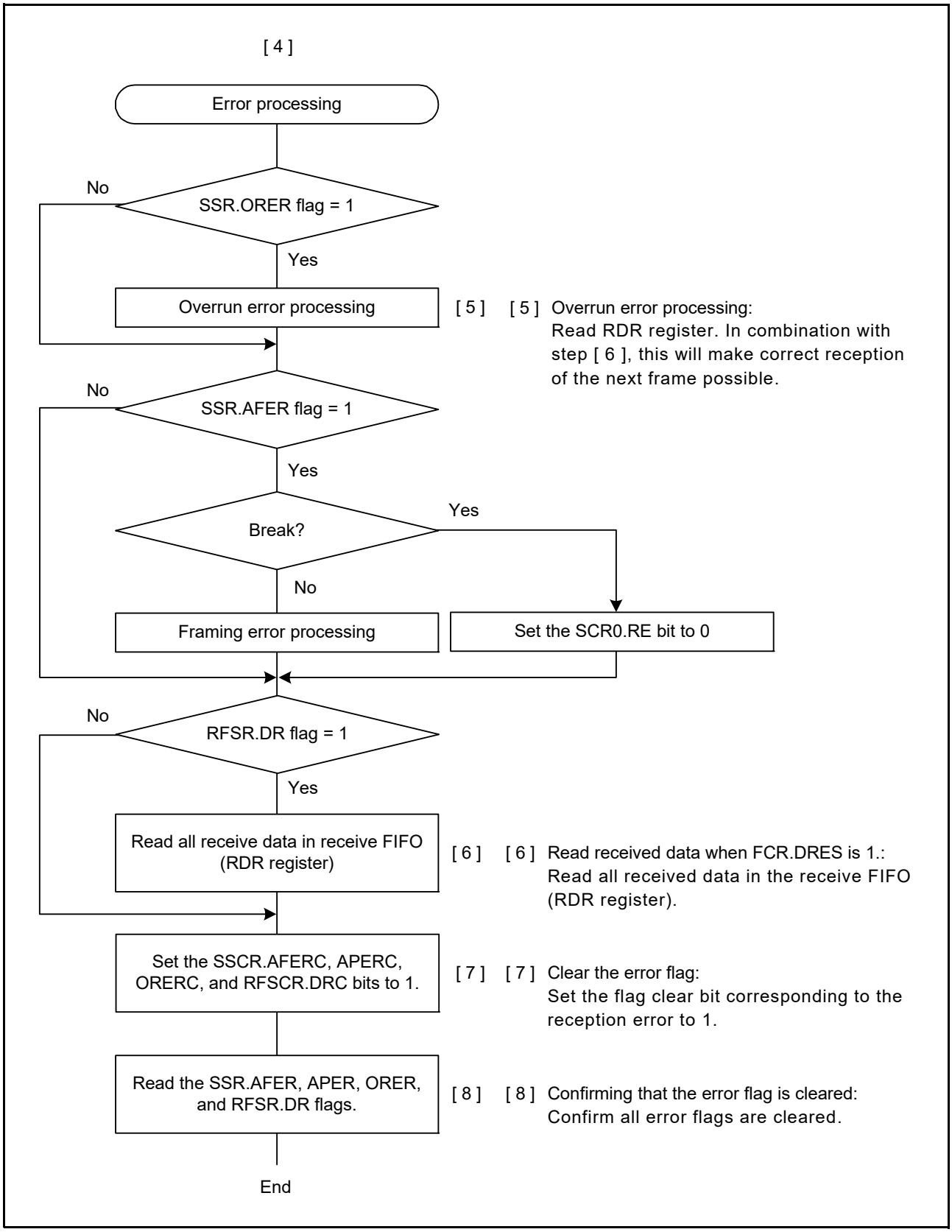


Figure 42.33 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO Mode)

42.5 Manchester Mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.
Figure 42.34 shows the conceptual image of Manchester encoding.

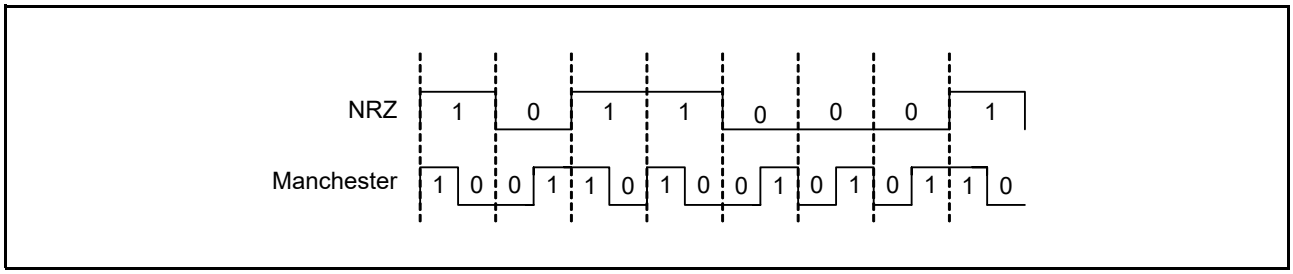


Figure 42.34 Example of Manchester Encoding

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.
For details on the frame format, see section 42.5.1, Frame Format.

42.5.1 Frame Format

Figure 42.35 shows the frame format in Manchester mode.
In the upper half of the figure, relevant setting registers are shown.
The preface area and the data area are encoded in Manchester encoding.

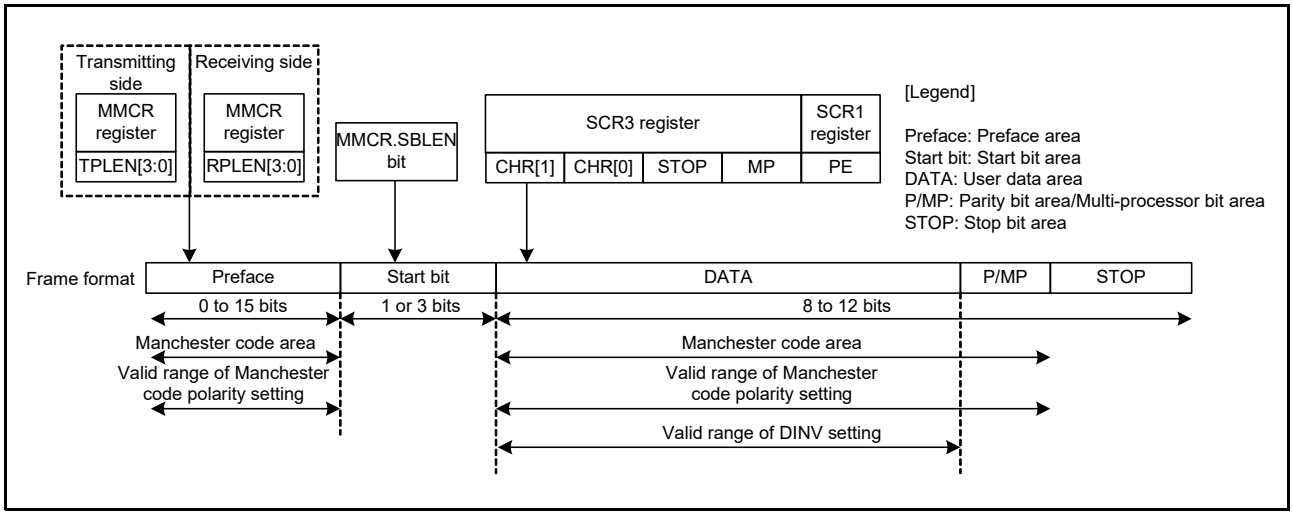


Figure 42.35 Frame Format in Manchester Mode

(1) Preface Area

This is a fixed pattern area located at the beginning of each frame.
Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MMCR.TPLEN[3:0] bits for transmission. It is determined by setting MMCR.RPLEN[3:0] bits for reception.
If it is set to 0, the transmit preface is disabled and is not added.
If it is set to 1d to 15d, a preface whose length is determined by this setting is added.
(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)
In addition, the preface pattern can be changed by setting, and it can be selected from four types of patterns by setting

MMCR.TPPAT[1:0] bits for transmission and MMCR.RPPAT[1:0] bits for reception.
Figure 42.36 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

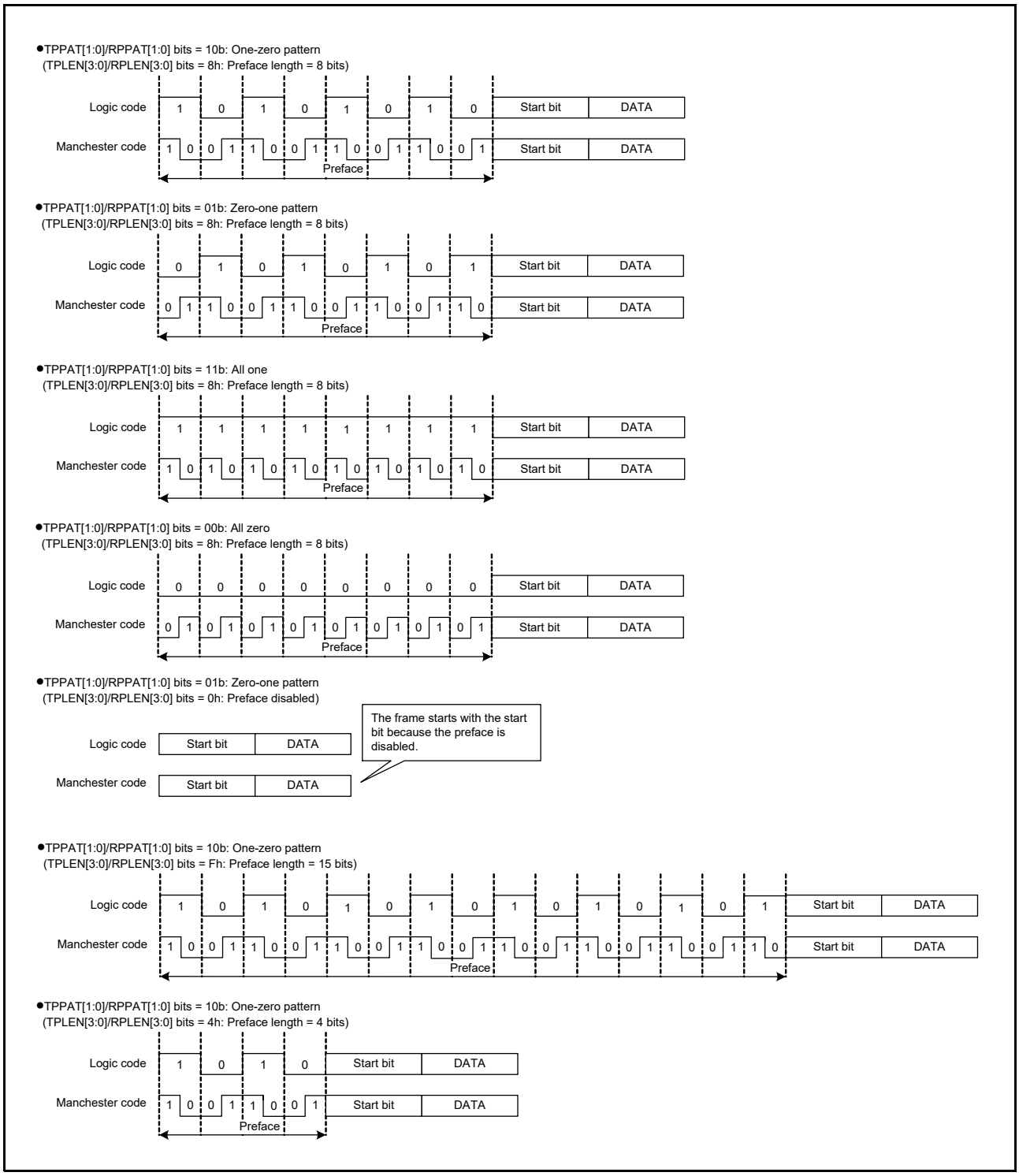
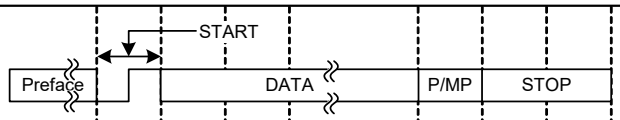
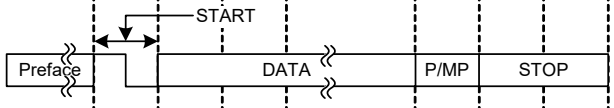
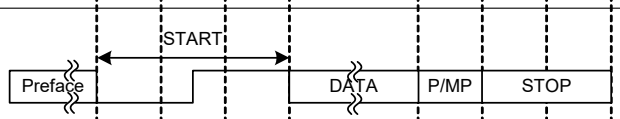
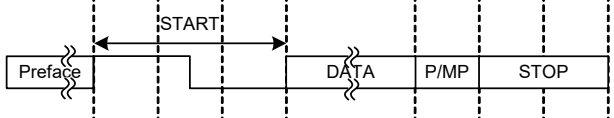
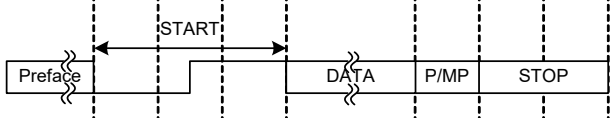
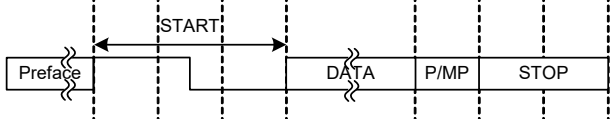


Figure 42.36 Preface Pattern Setting Example

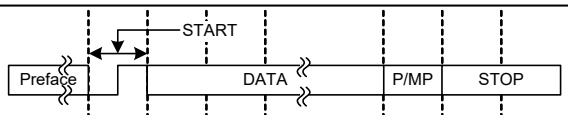
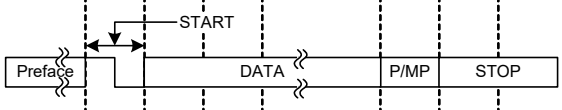
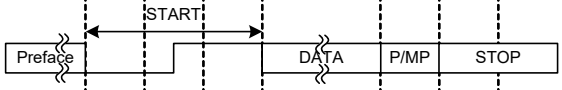
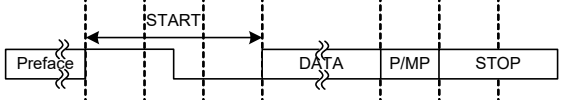
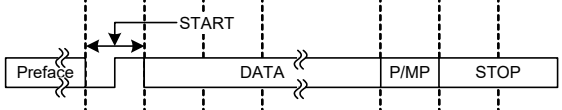
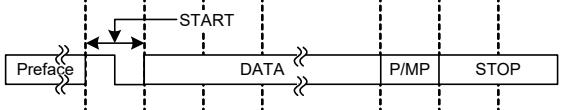
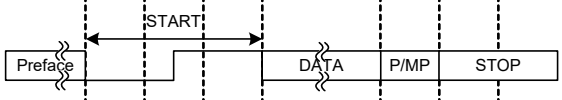
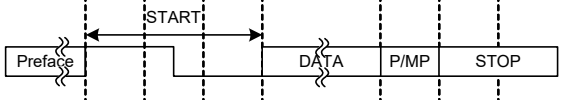
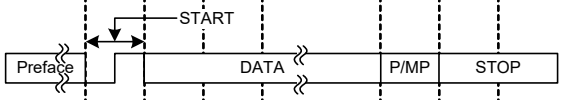
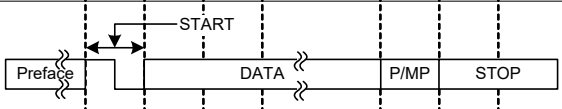
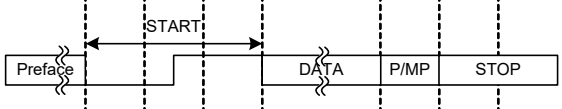
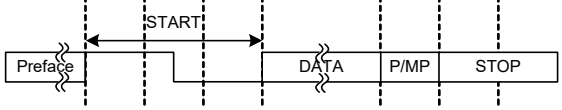
(2) Start Bit Area

This is an area indicating the start of valid data in a frame. It is added after the preface area.
The start bit length is determined by MMCR.SBLEN bit setting. When MMCR.SBLEN bit = 0, the start bit length is 1 bit. When MMCR.SBLEN bit = 1, the start bit length is 3 bits.
When MMCR.SBLEN bit = 1, the Sync type can be selected from command Sync and data Sync.
Command Sync means the three start bits are added as a one-to-zero transition.
Data Sync means the three start bits are added as a zero-to-one transition.
The Sync type is determined by the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bit settings. (When receiving, the received result is applied to MMSR.RSYNC bit.)
When MMCR.SBLEN bit = 0, the start bit is added as a zero-to-one or one-to-zero transition. The selection is determined by the MMCR.SBPTN setting.
The MMCR.SYNCE bit specifies the destination to be referred to when setting for transmission. When the MMCR.SYNCE bit is set to 1, the MMCR.SBPTN setting is referred to. When the MMCR.SYNCE bit is set to 0, the TDR.SYNC bit setting is referred to.
Figure 42.37 shows the state of the start bit area according to the settings in the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bits in the case of transmission. Figure 42.38 shows that in the case of reception.
The start bit(s) is not affected by the MMCR.ENCS bit or MMCR.DECS bit setting.

Register Setting				Output Signal	Output Start Bit
MMCR			TDR	Output TXD Waveform	
SBLEN	SYNCE	SBPTN	SYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition
1	0	1	d.c.		3bit 1 to 0 transition
1	1	d.c.	0		3bit 0 to 1 transition
1	1	d.c.	1		3bit 0 to 1 transition

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Figure 42.37 Settings Related to and Format of the Start Bit Area at Transmission

Register Setting				Input Signal	Start Bit Detection Result *1	Register Indication
MMCR			TDR	RXD Input Waveform		MMSR.RSYNC
SBLEN	SYNCE	SBPTN	SYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0 to 1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1 to 0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					DATA Sync	0
					COMMAND Sync	1

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Note 1. Data other than the start bit is assumed to be normal.

Figure 42.38 Settings Related to and Judgment of the Start Bit Area at Reception

(3) Data

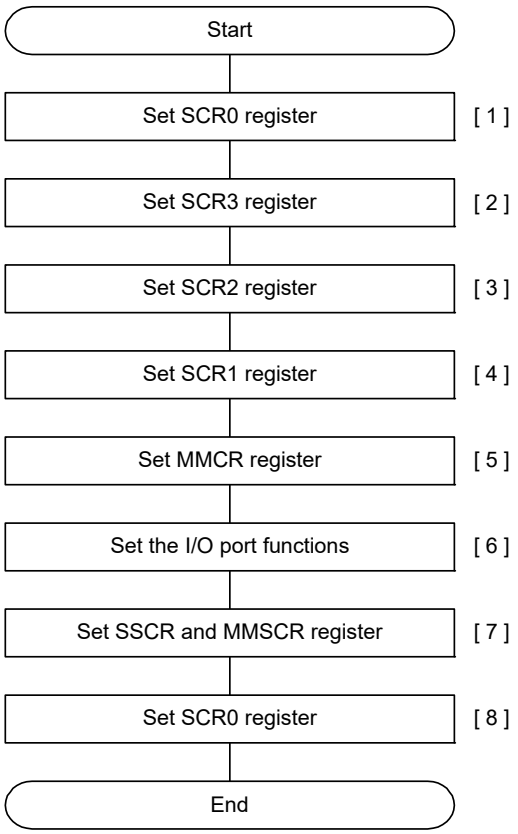
Since the format of the data area is the same as that of the asynchronous mode, see section 42.3.1, Serial Data Transfer Format.
As shown in Figure 42.35, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

42.5.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the SCR2.CKS[1:0] bits.
Also it is possible to set the oversampling (transfer rate of one bit period) by SCR2.ABCS bit. When the SCR2.ABCS bit is set to 0, oversampling ×16 is selected with the one-bit period being 16 cycles of the base clock. When the SCR2.ABCS bit is set to 1, oversampling ×8 is selected with the one-bit period being 8 cycles of the base clock.

42.5.3 RSCI Initialization of Manchester Mode

Before transferring data, write 0 to SCR0.TE and SCR0.RE bits (or write the initial value to SCR0 register), and initialize the RSCI following the example of flowchart shown in Figure 42.39.
Be sure to write 0 to SCR0.TE and SCR0.RE bits before changing the operation mode or communication format.
Note that setting the SCR0.RE bit to 0 initializes none of the SSR.ORER, AFER, APER, RDRF, MMSR.MCER, SYER, PFER and SBER flags, and the RDR registers.
Note also that switching the value of SCR0.TE bit from 0 to 1 when SCR0.TIE bit is 1 generates a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, multi-processor communication, the communication mode, transmission/reception format, and clock enable.
Leave unused bits such as CPOL and CPHA bits at their initial values.

[3] Set the clock selection, bit rate setting.
Leave unused function bits at their initial values.

[4] Set the noise filter, loopback control, communication terminal status, parity check, and CTS/RTS function.

[5] Set Manchester transmission/reception setting.
Start bit setting, retiming setting, Manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I/O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the SSCR and MMSCR registers to clear the target flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MMSCR.MCERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[8] Set the SCR0.TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Set the multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 42.39 RSCI Initialization Flow in Manchester Mode

42.5.4 Double Speed Operation

When the ABCS bit in SCR2 register is set to 1 and eight pulses of the base clock for a 1 bit period is selected, the RSCI operates on the bit rate twice that of when ABCS is set to 0.
When the BGDM bit in SCR2 register is set to 1, the cycle of the base clock is reduced to half and the RSCI operates on the bit rate twice that of when BGDM is set to 0.
When the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 1, the RSCI operates on the bit rate four times that of when the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 0.

42.5.5 CTS, RTS Functions

The CTS function controls transmission using the CTSn# pin input. Setting the CTSE bit in SCR1 register to 1 enables the CTS function. The CTSn#/RTSn# pin can be set as a multiplexed pin which allows one pin to be used for either function, or as dedicated pins with each pin for a single function. Use the CRSEP bit in SCR1 register for this setting.
When the CTS function is enabled, transmission starts only when the CTSn# pin is at the low level.
Even if the level of CTSn# pin goes High after transmission starts, does not affect transmission of the current frame, which continues.
The RTS function uses output on the RTSn# pin to request transmission. When the RSCI is ready to receive, it outputs a low level to the RTSn# pin, Conditions for output of the low level and high level are as follows:
[Conditions for low-level output]
When all conditions listed below are satisfied:

- The value of the SCR0.RE bit is 1.
- There are no received data yet to be read and reception is not in progress.
- All of the following flags are set to 0:
  SSR.ORER, AFER, APER and MMSR.MCER, SBER (when SBERIE bit = 1), SYER (when SYERIE bit = 1), PFER (when PFERIE bit = 1)

[Conditions for high-level output]
When the conditions for low output are not satisfied.

42.5.6 Serial Data Transmission in Manchester Mode

The RSCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.
When the polarity setting (MMCR.ENCS bit) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.
When the polarity setting (MMCR.ENCS bit) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.
For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 42.34)
The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.
For details on the frame format, see section 42.5.1, Frame Format.
Figure 42.40 shows the flowchart in transmission. At transmission starts, set the SCR0.TIE and SCR0.TE bits to 1 simultaneously with one instruction. Then, a TXI interrupt request is generated.
Figure 42.41 to Figure 42.43 show examples of the operation for serial transmission in Manchester mode.

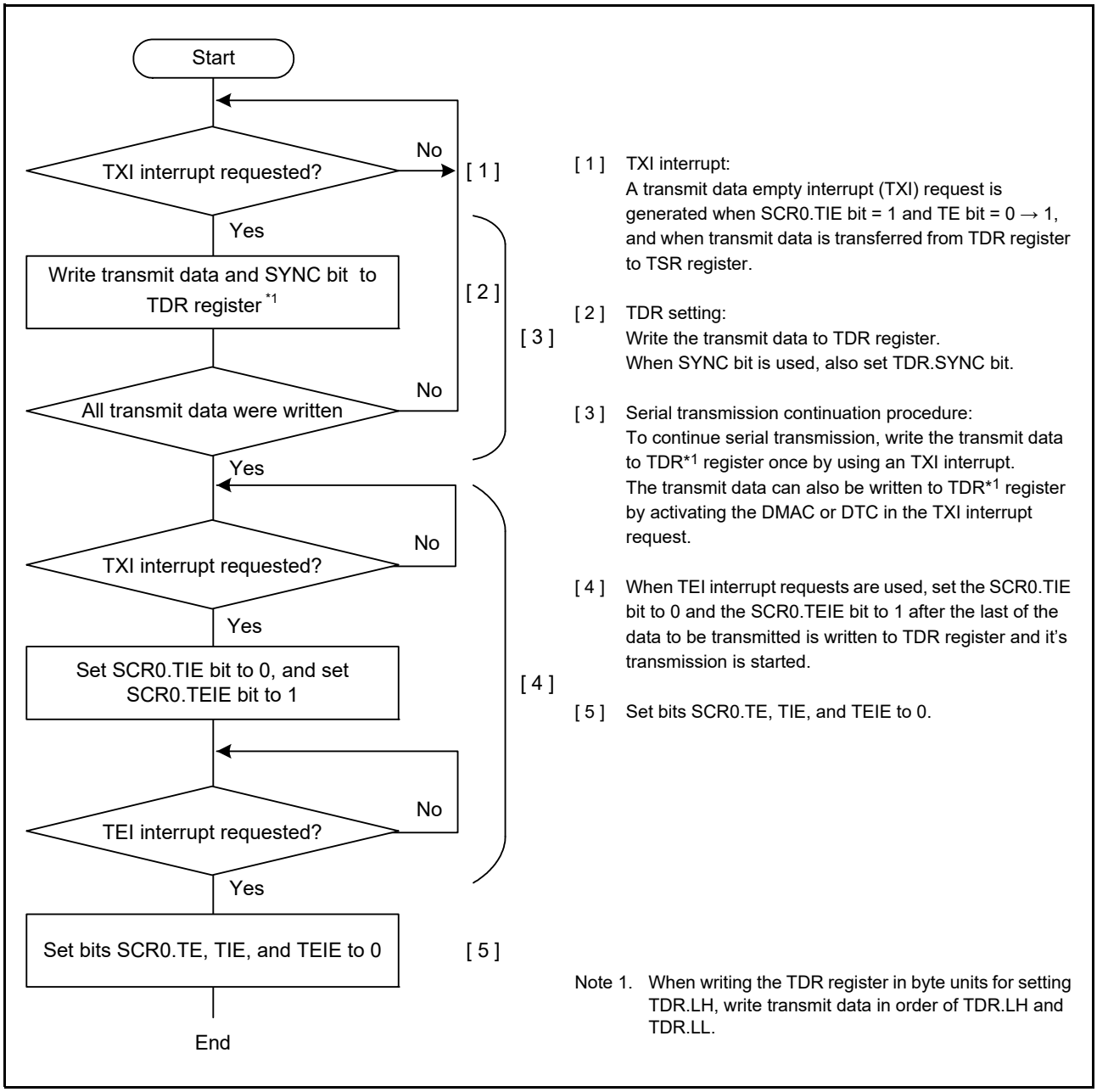
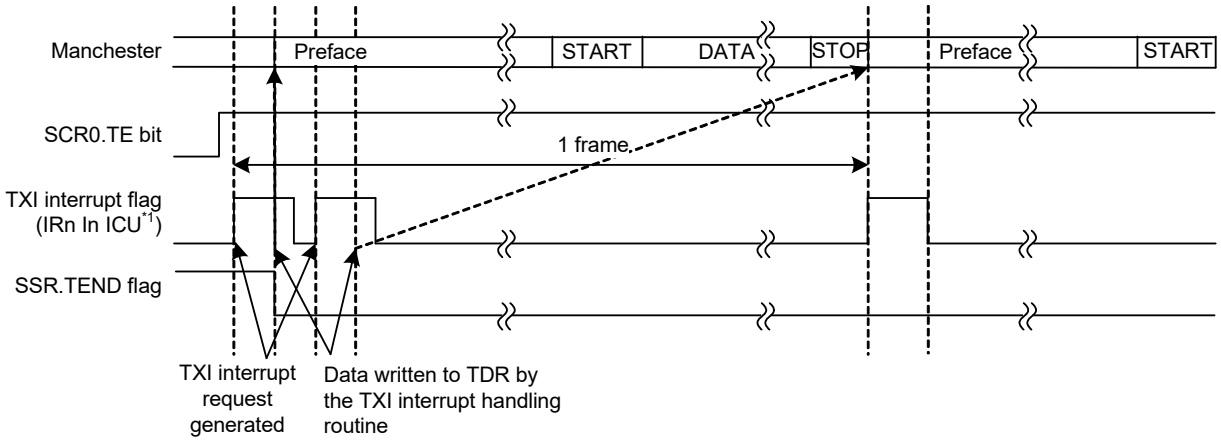
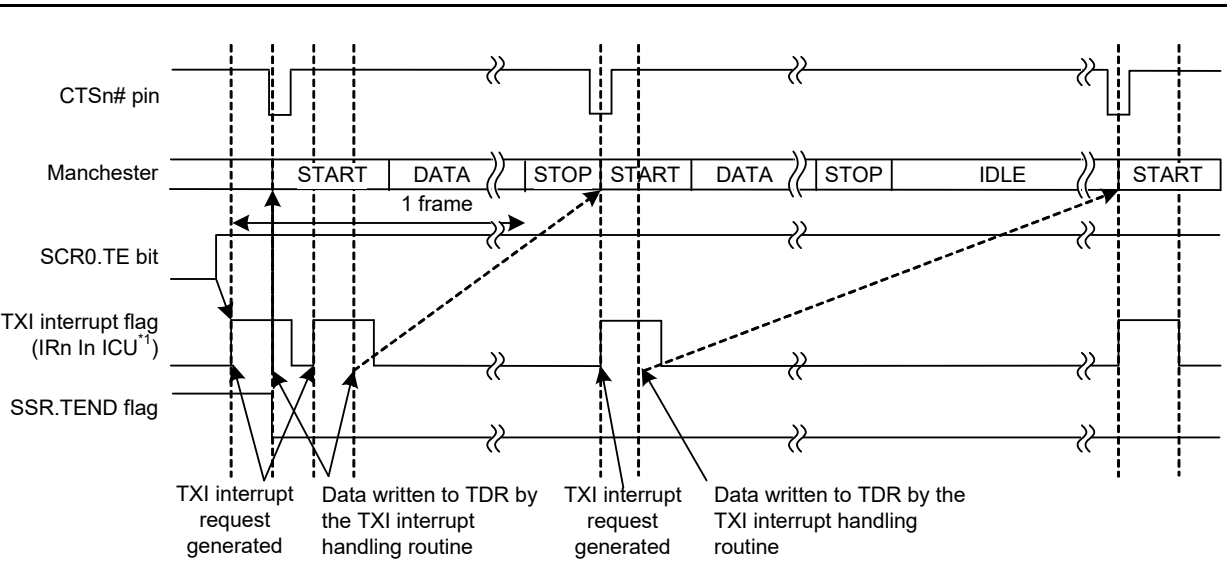


Figure 42.40 Serial Transmission Flowchart in Manchester Mode



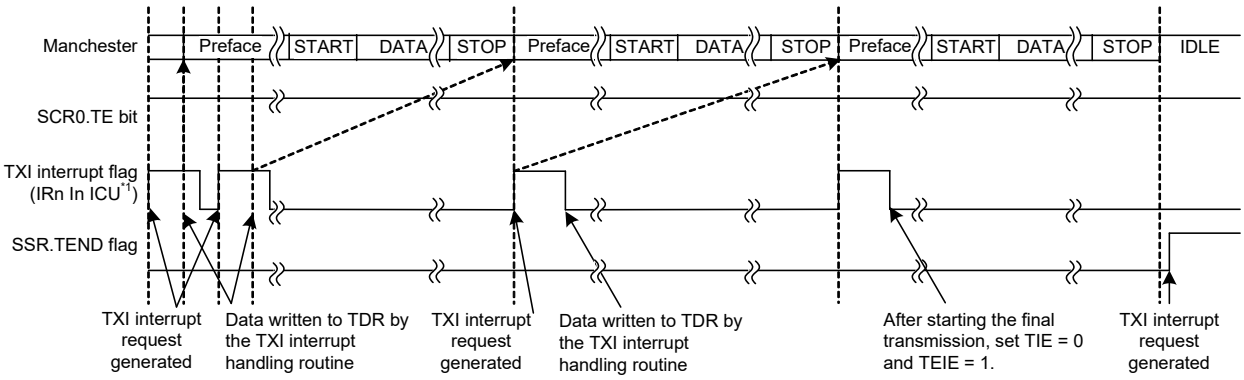
Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.41 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.42 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (without Preface but with the CTS Function)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.43 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)

42.5.7 Serial Data Reception in Manchester Mode

In Manchester mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 42.44, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the RSCI judges it as an error and waits for a falling edge again.
If a high level is expected in the first half of a bit in the received data, the RSCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when SCR2.ABCS bit = 0. When SCR2.ABCS bit = 1, the RSCI operates on a base clock with a frequency of 8 times the bit rate.

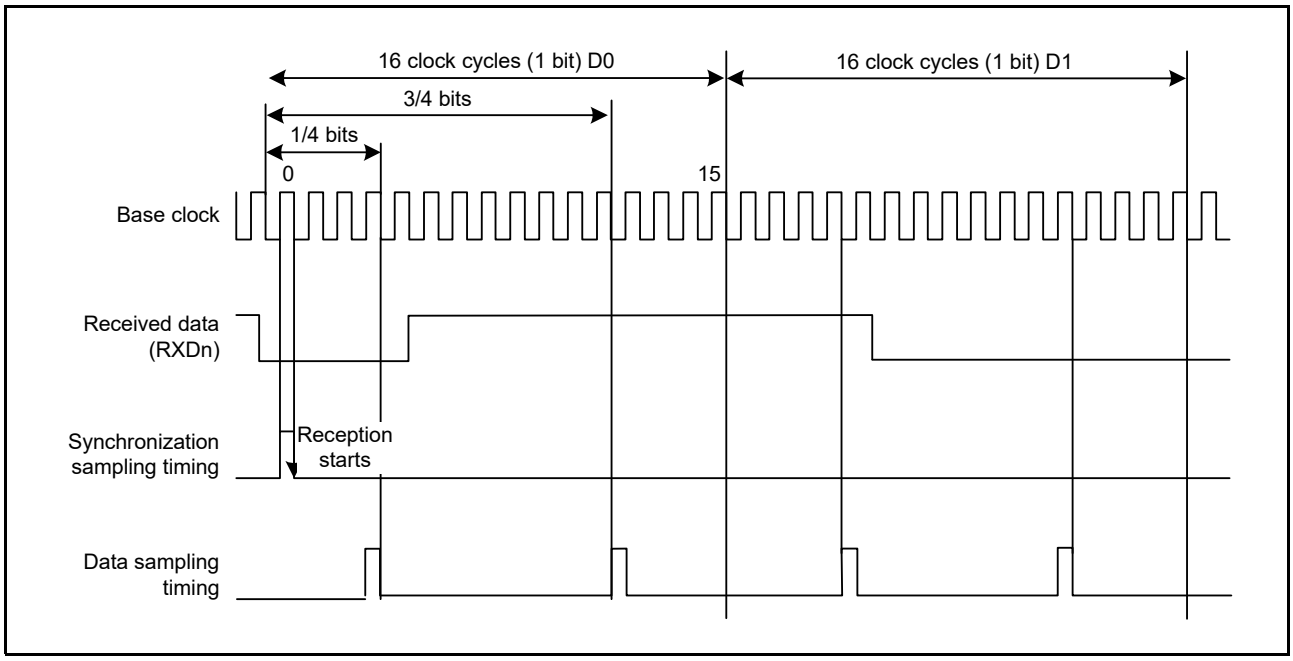
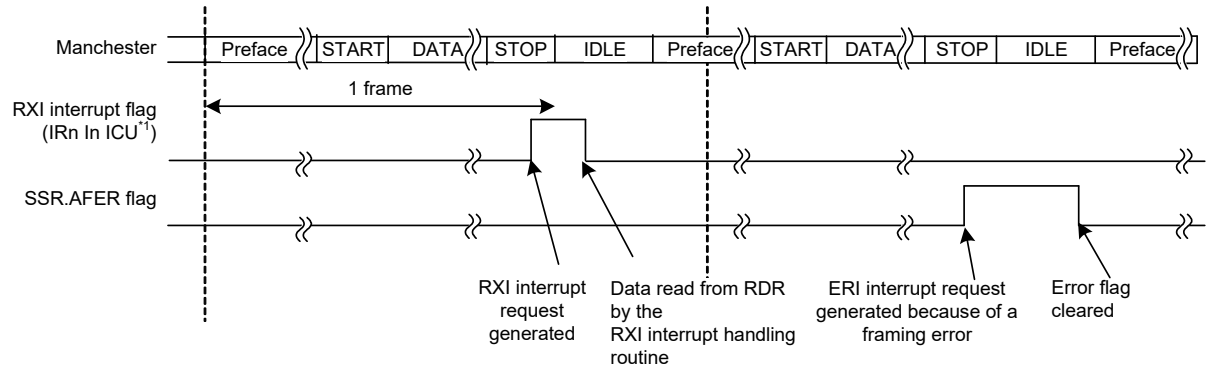


Figure 42.44 Data Reception Sampling Timing in Manchester Mode

In Manchester mode, data reception starts with detection of a preface and start bit area.
The RSCI checks the input from the RXDn pin to see whether a preface is added based on the value of MMCR.RPLEN[3:0] bits.
If the preface is disabled (MMCR.RPLEN[3:0] bits = 0), it moves on to the detection of a start bit area without detecting a preface.
When a preface is enabled, it identifies a preface pattern setting according to the set value in MMCR.RPPAT[1:0] bits, and compares it with the RXD input for a pattern match to detect a preface pattern.
Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.
If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).
For start bit detection, the RSCI selects an expected value based on the register settings (MMCR.SBLEN and SBPTN bits), compares it with the RXD input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.
Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.
Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).
In data processing, the RSCI shifts the data by the expected received data length based on the register settings (SCR3.CHR[1:0] bits) through the RSR register. If two sampling points in a bit of the received data are identical, the RSCI judges this as a Manchester code error. For details, see section 42.5.11, Errors in Manchester Mode (4).

When the parity function is disabled (SCR1.PE bit = 0), the RSCI moves on to the next phase of stop bit detection. When the parity function is enabled (SCR1.PE bit = 1), the RSCI performs parity checking. If detecting a parity error, it asserts a parity error flag (APER), and then moves on to stop-bit detection.
In stop bit detection, the RSCI checks the following in the stop bit area of the received frame:
It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (AFER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.
Figure 42.45 shows an example of the operation for serial data reception in Manchester mode.

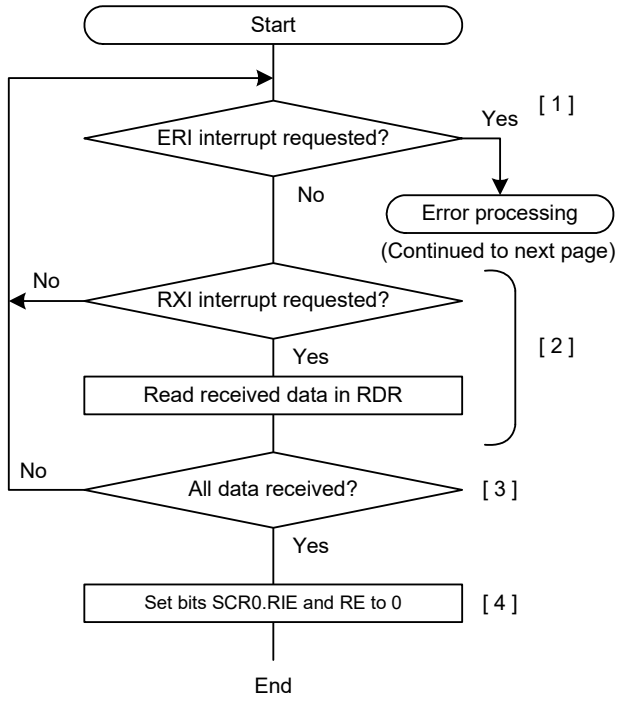


Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.45 Example of Operation for Serial Data Reception in Manchester Mode (with a Preface)

For the state of each status flag in the SSR and MMSR registers and RXD input processing when a receive error is detected, see section 42.5.11, Errors in Manchester Mode.
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated.
Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, APER, MCER, SYER*2, PFER*2, SBER*2 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.
Figure 42.46 and Figure 42.47 show examples of serial data reception flowchart in Manchester mode.

Note 2. Effective when the corresponding bit is enabled.



[1] Receive error processing:
When a receive error occurs, RSCI generates an ERI interrupt. An error is identified by reading the ORER, APER, AFER in SSR register and MCER, SBER, SYER, PFER in MMSR register. After performing the appropriate error processing, always set the ORER, APER, AFER, MCER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MMSR register has an enable bit (PFERIE, SYERIE, and SBERIE bits in MMCR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[2] Read the received data in RDR register once in the RXI interrupt handling routine.

[3] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[4] Write 0 to SCR0.RIE and RE bits.

Figure 42.46 Example of Serial Data Reception in Manchester Mode (Normal)

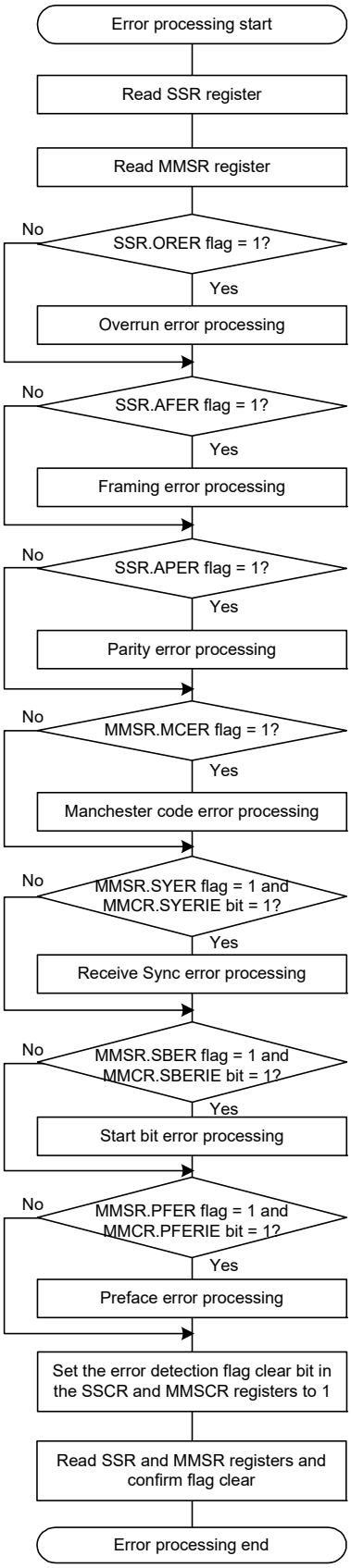


Figure 42.47 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

42.5.8 Operation When Multi-Processor Bit is Used

See section 42.4, Multi-Processor Communication Function (1) for the operation in Manchester mode when using multi-processor mode because the operation is the same.
A preface and a start bit area are added to the frame format in Manchester mode.
See Figure 42.47 for error processing in Manchester mode for the reception flowchart (Figure 42.30). Refer to Table 42.37 for the operation status when detecting various errors.

42.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.
The receive retiming function can be turned on or off by setting the SADJE bit in the MMCR register.
When the receive retiming function is turned off (MMCR.SADJE bit = 0), retiming is not performed, causing misalignment between the internal clock and the RXD input to be accumulated and the receive margin to be reduced.
When the receive retiming function is turned on (MMCR.SADJE bit = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling ×16 is selected is shown below.
When detecting an RXD input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.
When detecting a RXD input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.
(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)
Figure 42.48 shows the conceptual image of receive retiming range.
When detecting an edge in the tolerance area in the figure, data is received as is without making correction.
When detecting an edge in the SyncJump area in the figure, data is corrected for reception.
When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.
For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the RSCI reports a code error.

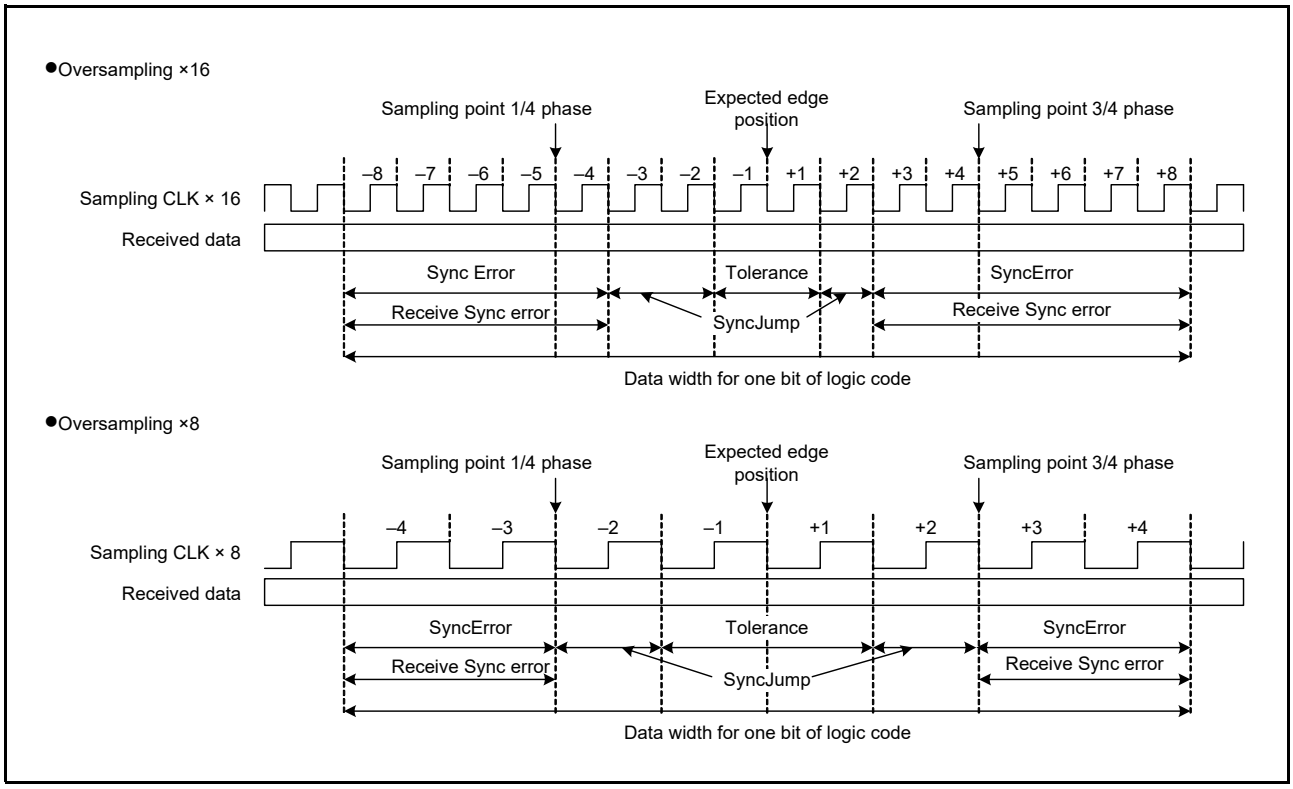


Figure 42.48 Conceptual Image of Reception Retiming Range

42.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MMCR).
It can be set separately for transmission and reception. Use the MMCR.ENCS bit to set the polarity for transmission and the MMCR.DECS bit to set the polarity for reception.
The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.
When the initial settings (ENCS/DECS bit = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code.
If the settings are changed to ENCS/DECS bit = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 42.49 shows the conceptual image of the settings and operation.
Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (SCR3.DINV bit). Since the polarity of Manchester code (MMCR.ENCS/DECS bit) can be set separately from the transmitted/received data invert function (SCR3.DINV bit), if both are set to inversion (MMCR.ENCS/DECS bit = 1 and SCR3.DINV bit = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).
The polarity of the start bit area can be set by a register different from the ones mentioned above.
Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.
For details on the setting for the start bit area, see section 42.5.1, Frame Format (2).

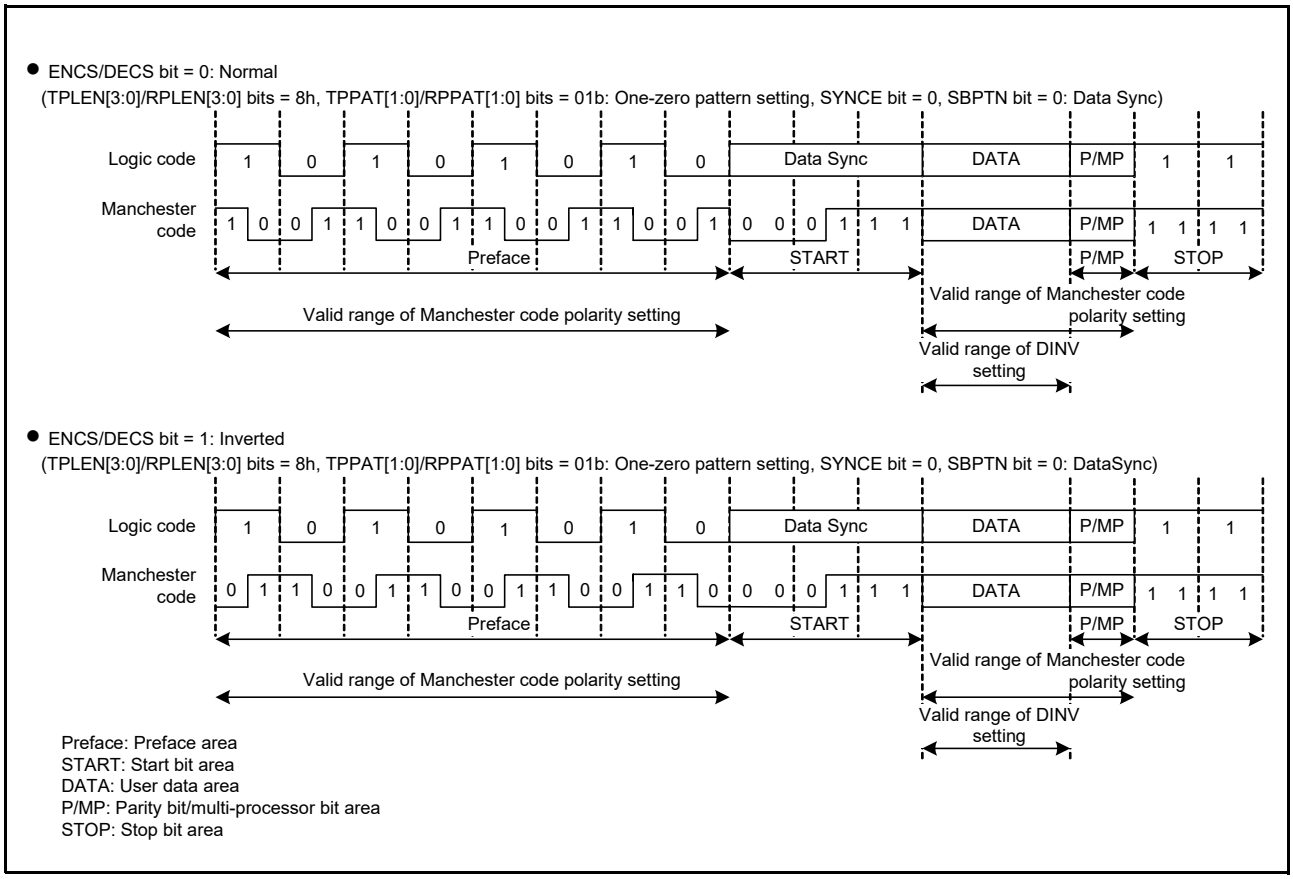


Figure 42.49 Valid Range of the Manchester Code Polarity Setting

42.5.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

(1) Parity error
(2) Over run error
(3) Framing error
(4) Manchester code error
(5) Preface error
(6) Start bit error
(7) Receive Sync error

For errors (1) to (3), see section 42.3.9, Serial Data Reception (Asynchronous Mode) (1) because they are the same as in asynchronous mode.
Each error is judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.
Table 42.35 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR register.
Table 42.36 lists the errors that can be detected in each area of a Manchester frame. If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous frame, data will not be received, but errors in the preface area and start bit area will update that flag. Table 42.37 shows the flags and actions in this case.

(4) Manchester Code Error

A Manchester code error is generated when a Manchester code error is detected.
In Manchester code, there must be an edge (transition) in the center of the bit.
In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values matches.
If a Manchester code error is detected, the Manchester code error flag (MMSR.MCER flag) is asserted.
If a Manchester code error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface Error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MMSR.PFER flag) is asserted.
It is possible to set whether to use this error flag as an interrupt source with the setting of the MMCR register.
When MMCR.PFERIE bit = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.PFERIE bit = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MMSR.PFER flag.

(6) Star Bit Error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MMSR.SBER) is asserted.
It is possible to set whether to use the start bit error as an interrupt source with the setting of the MMCR register.
When MMCR.SBERIE bit = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MMCR.SBERIE bit = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MMSR.SBER flag.

(7) Receive Sync Error (SyncError)

When the receive retiming function described in section 42.5.9, Receive Retiming is enabled, the receive retiming operation is performed.
If no edges are detected within the receive retiming range (SyncError area in Figure 42.48) when receive timing operation is being performed, a receive Sync error is generated. Upon detection of a receive Sync error, a receive Sync error flag (MMSR.SYER) is asserted. In areas not subject to receive retiming, receive Sync errors are not detected.
The preface area *1, the start bit area*1, *2, and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.
It is possible to set whether to use the receive Sync error as an interrupt source with the setting of the MMCR register.
When MMCR.SYERIE bit = 1, a receive Sync error is handled as an interrupt source or event source. If a receive Sync error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.SYERIE bit = 0, a receive Sync error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive Sync error is notified to MMSR.SYER flag.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming. Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 42.35 Flags in the Status Register and Receive Data Handling in Manchester

Flags in the SSR Register			Flags in the MMSR Register					
ORER	AFER	APER	MCER	SBER *1	PFER *1	SYER	Received Data	Received Error Status (ERI Interrupt/Event Generation)
0	0	0	0	0	0	0	Transfer to RDR	No error
0	1	0	0	0	0	0	Transfer to RDR	Framing error
0	0	1	0	0	0	0	Transfer to RDR	Parity error
0	1	1	0	0	0	0	Transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	Transfer to RDR	Manchester code error
0	1	0	1	0	0	0	Transfer to RDR	Framing error + Manchester code error
0	0	1	1	0	0	0	Transfer to RDR	Parity error + Manchester code error
0	1	1	1	0	0	0	Transfer to RDR	Framing error + Parity error + Manchester code error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun + Framing error
1	0	1	0	0	0	0	Lost	Overrun + Parity error
1	1	1	0	0	0	0	Lost	Overrun + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun + Manchester code error
1	1	0	1	0	0	0	Lost	Overrun + Framing error + Manchester code error
1	0	1	1	0	0	0	Lost	Overrun + Parity error + Manchester code error
1	1	1	1	0	0	0	Lost	Overrun + Framing error + Parity error + Manchester code error
0	Combination of the above errors			0	0	1	Transfer to RDR	Error above + Receive Sync error*2
1				0	0	1	Lost	Error above + Receive Sync error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive Sync error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive Sync error*2

Note 1. Start bit error and preface error never become 1 at the same time.
Note 2. When MMCR.SYERIE bit = 1, ERI interrupt/event is generated by SYER factor.
Note 3. If MMCR.PFERIE bit = 1 or MMCR.SBERIE bit = 1, an ERI interrupt/event is generated when the corresponding flag is set.

Table 42.36 Errors Detectable in Each Area

	Preface Error (PFER)	Start Bit Error (SBER)	Manchester Code Error (MCER)	Receive Sync Error (SYER)	Parity Error (APER)	Framing Error (AFER)
Preface area	✓	—	—*1	✓*2	—	—
Start bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop bit area	—	—	—	—	—	✓

✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to receive Sync error detection. For details see section 42.5.11, Errors in Manchester Mode (7)

Table 42.37 Operation Status due to Presence/Absence of Error in Previous Frame and Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 0)

Previous Frame	Each Area of the Frame					PFE RIE	SBE RIE	SYE RIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
No error	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Not output	Not output
						1					Output	Output
	No error	SBER, No SYER*1	—	—	—	—	0	—	Lost	Set SBER*1	Not output	Not output
							1				Output	Output
	SYER, No PFER	No error	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	SYER, No SBER	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	No error	SYER		No error	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1			Output	Output
	No error	No error	MCER		No error	—	—	—	Transfer to RDR	Set MCER	Output	Output
	No error	No error	—	APER	No error	—	—	—	Transfer to RDR	Set APER	Output	Output
	No error	No error	—	—	AFER	—	—	—	Transfer to RDR	Set AFER	Output	Output
	There is some error ORER					—	—	—	Lost	Set some flags*2	Output	Output
	No error	No error	No error	No error	No error, ORER	—	—	—	Lost	Set ORER	Output	Output
Some error*3, *6	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Output*4	Not output*5
						1						
	No error	SBER, No SYER*1	—	—	—	—	0	—		Set SBER*1		
							1					
	SYER, No PFER	No error	—	—	—	—	—	0		Set SYER		
								1				
	No error	SYER, No SBER	—	—	—	—	—	0		Set SYER		
								1				
	No error	No error	SYER		No error	—	—	0		don't set any flags		
								1				
	No error	No error	MCER		No error	—	—	—				
	No error	No error	—	APER	No error	—	—	—				
	No error	No error	—	—	AFER	—	—	—				
	There is some error ORER					—	—	—				
	No error	No error	No error	No error	No error, ORER	—	—	—				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.

Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of errors in the relevant frame.
Note 5. Since the error cause is continuously detected, the ERI event is not newly output regardless of the presence of errors in the relevant frame.
Note 6. For MMSR.PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 42.38 Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 1)

MPB*1	Each Area of the Frame					PFERIE	SBERIE	SYERIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
1	No error	No error	—	—	—	—	—	—	Transfer to RDR	Set some flags	Output	Output
	No PFER, SYER	No SBER, SYER	—	—	—	—	—	0				
								1	Lost	don't set any flags	Not output	Not output
	PFER	No error	—	—	—	—	—	—				
	No error	SBER	—	—	—	—	—					

Note 1. If the received MPB flag is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, RXI interrupt request or event is output, and if it is detected, ERI interrupt request or event is output.
Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the MMCR.SYERIE bit changes.

42.6 HBS Support Mode

Setting the HBSCR.HBSE bit to 1 supports the negative logic RZI coding to generate waveforms (AMI, 50% duty cycle, negative logic) required by the home bus system. Since this function operates only in the asynchronous mode, refer to the asynchronous mode for the setting, transmission flow, and reception flow.

42.6.1 Reception in HBS Support Mode

When receiving in HBS support mode, the falling edge of the input from the RXDn pin is detected and the signal after the start bit is recognized is received. One frame is sampled according to the set bit rate, and if the stop bit is correctly received without error, the data value is stored in the receive data register RDR.

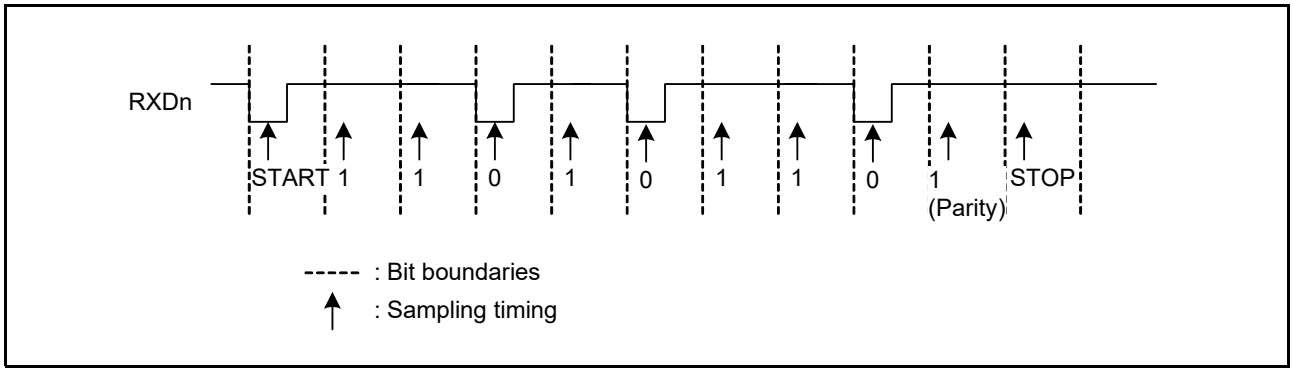


Figure 42.50 HBS Support Mode Reception Timing Chart

To receive in the HBS support mode, it is necessary to sample at the timing of 1/4 of 1 bit in order to capture the pulse in the first half of the 1 bit period. Sampling operates with a frequency 16 times the bit rate*1 as the base clock, similar to the asynchronous mode. The start bit is detected by detecting the Low level from the falling edge of RXD four times continuously with the base clock. When High is detected on the way, it is regarded as noise and waits for the next fall edge.

To set the sampling timing to 1/4 of the 1-bit period, enable the reception sampling timing adjustment function with the SCR4.RTADJ bit, set the SCR4.RTMG[3:0] bits to 1100b, and adjust Adjust from the center of the bit, which is the previous sampling timing, four clocks ahead of the base clock.

Since the sampling timing can be adjusted backward and forward using the reception sampling timing adjustment function, this timing can be adjusted according to the reception status. Increasing the SCR4.RTMG[3:0] bits value from 1100b will move the sampling timing forward, and decreasing it will move it backward. Refer to section 42.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode for the details of adjustment.

After recognizing the start bit, sampling is performed at the timing according to the set bit rate, but the low width and high width of the waveform are not checked. Therefore, it is possible to receive even a normal asynchronous waveform.

Note 1. HBS support mode supports only following setting: SCR2.ABCS bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0.

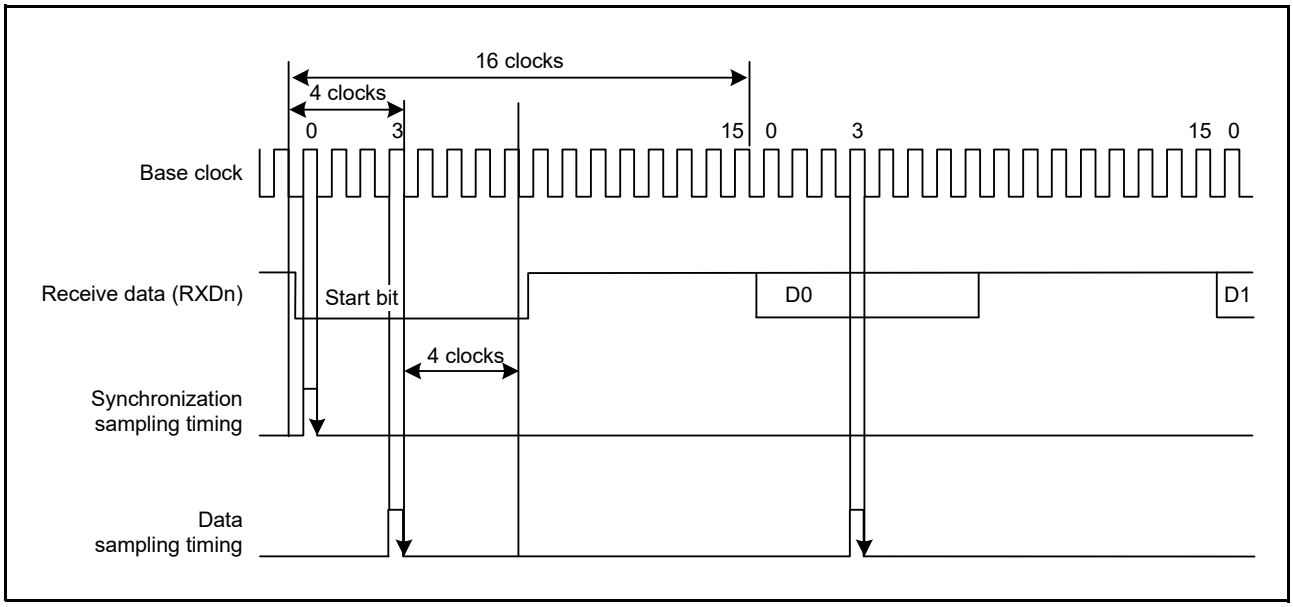


Figure 42.51 Details of Reception Sampling Timing in HBS Support Mode

42.6.2 Transmission in HBS Support Mode

Transmission in HBS support mode, data 0 is output as a low pulse for the first half of the 1-bit period.

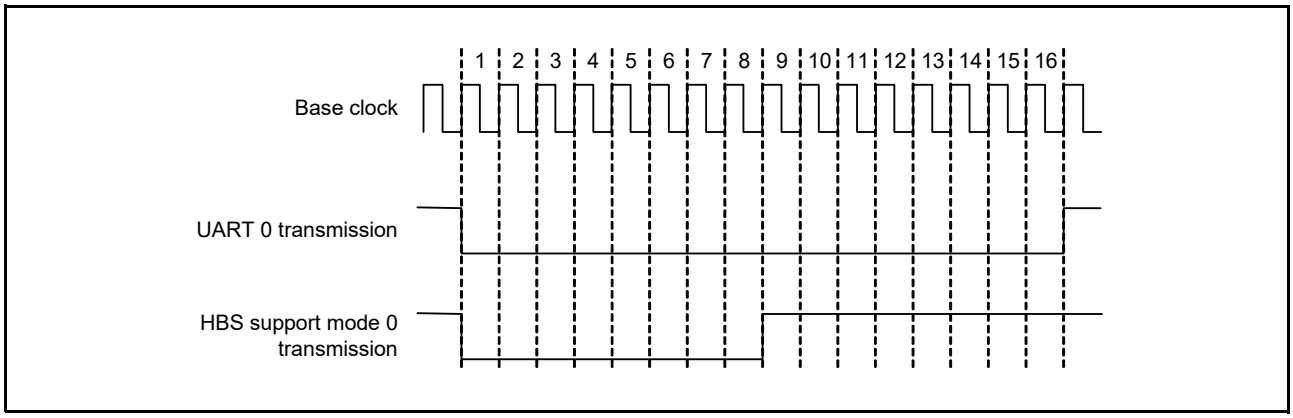


Figure 42.52 Transmission Waveform of HBS Support Mode

When HBSCR.AOE bit = 0, the all bits are output from TXDn pins, and when HBSCR.AOE bit = 1, data 0 is output alternately from TXDAn and TXDBn pins. Use the HBSCR.LPS bit to select which transmission pin starts output of the start bit.

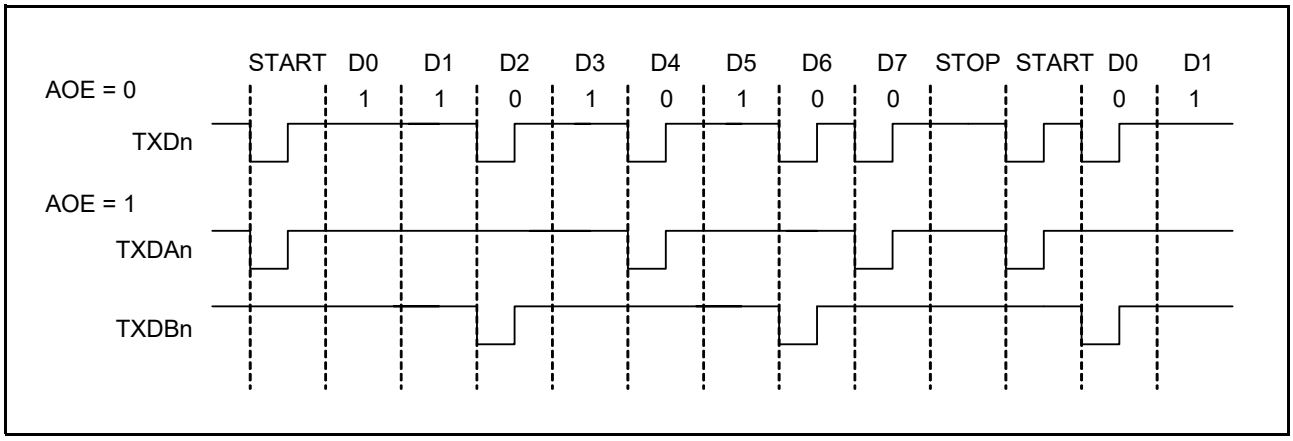


Figure 42.53 Difference in Transmission Waveform Depending on AOE Bit (When LPS Bit = 0)

Figure 42.53 shows an example of the transmission waveform for each HBSCR.AOE bit value. When the AOE bit is set to 0, the waveform is output from the TXDn pin, but when the AOE bit is set to 1, data 0 including start bit is output alternately from the TXDAn pin and the TXDBn pin.

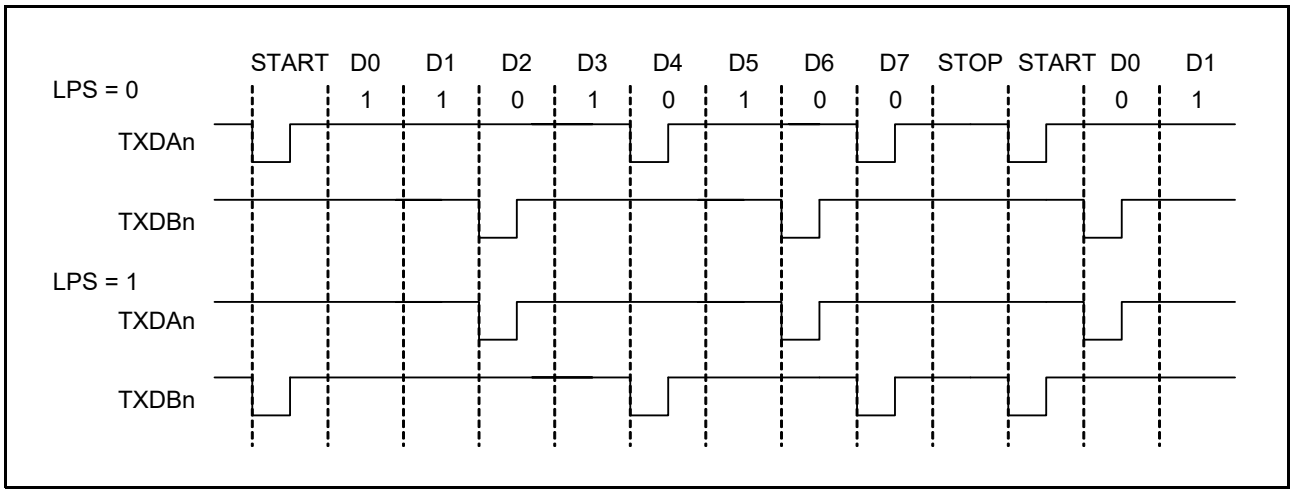


Figure 42.54 Difference in Transmission Waveform Depending on LPS Bit (When AOE Bit = 1)

Figure 42.54 shows an example of the transmission waveform for each HBSCR.LPS bit.
When the HBSCR.LPS bit = 0, the start bit is output from the TXDAn pin, and when the HBSCR.LPS bit is 1, the start bit is output from the TXDBn pin, and data 0 is output to each pin alternately. The start bit of the next frame starts output again from the pin specified by the HBSCR.LPS bit.
If the HBSCR.HBSE bit = 0, the TXDBn pin becomes High regardless of the settings of other bits.
When SCR0.TE bit = 0, both TXDn/TXDAn/TXDBn pins become high impedance, but can be controlled by SCR1.SPB2IO bit and SCR1.SPB2DT bit. At this time, the same output is applied to the TXDn/TXDAn/TXDBn pins.

42.6.3 Register Setting for HBS Support Mode

The HBS support mode is a part of asynchronous mode function, but there are some settings that are not supported when using this function. Set each bit of the control register as shown in Table 42.39 before use. Register bits not described can be set in the same way as the asynchronous mode.

Table 42.39 Control Register Setting Value for HBS Support Mode

Register Bit Name	Value	Remarks
SCR0.DCME	0	Use it when the data match detection function is disabled.
SCR1.NFCS[2:0]	000b	Use this setting when using the noise filter.
SCR1.HDSEL	0	Half-duplex communication with the TXDn pin cannot be used.
SCR1.CTSE	0	Please use it without the CTS function.
SCR2.BRME	0	Bit rate modulation function cannot be used.
SCR2.ABCSE2	0	The setting that 4 cycles of the base clock becomes 1 bit cannot be used.
SCR2.ABCSE	0	The setting that 6 cycles of the base clock becomes 1 bit cannot be used.
SCR2.ABCS	0	Only the setting that 16 cycles of the base clock becomes 1 bit can be used.
SCR3.CKE[1:0]	00b	Use with internal clock and without clock output.
SCR3.DEEN	0	Use without the RS-485 driver function.
SCR3.FM	0	Use without FIFO function.
SCR3.MOD[2:0]	000b	Set to asynchronous mode.
SCR3.RXDESEL	1	Detect the start bit at the falling edge of the RXDn pin input.
SCR3.STOP	0	Use with stop bit 1.
SCR3.DINV	0	Use without data inversion.
SCR3.DDIR	1	Use with LSB first.
SCR3.CHR[1:0]	10b	Use with 8 bit length.
SCR4.RTMG[3:0]	1100b	Use this setting when receiving in HBS support mode.*1
SCR4.TTADJ	0	Use without adjust transmit timing function.
SCR4.RTADJ	1	Use this setting when receiving in HBS support mode.

Note 1. This is the timing to sample at the center of the effective pulse. It can be adjusted if needed.

42.7 Smart Card Interface Mode

The RSCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function.
Smart card interface mode can be selected using the appropriate register.

42.7.1 Sample Connection

Figure 42.55 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR0 register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the RSCI, input the SCKn pin output to the CLK pin of an IC card.
The output port can be used to output the reset signal.

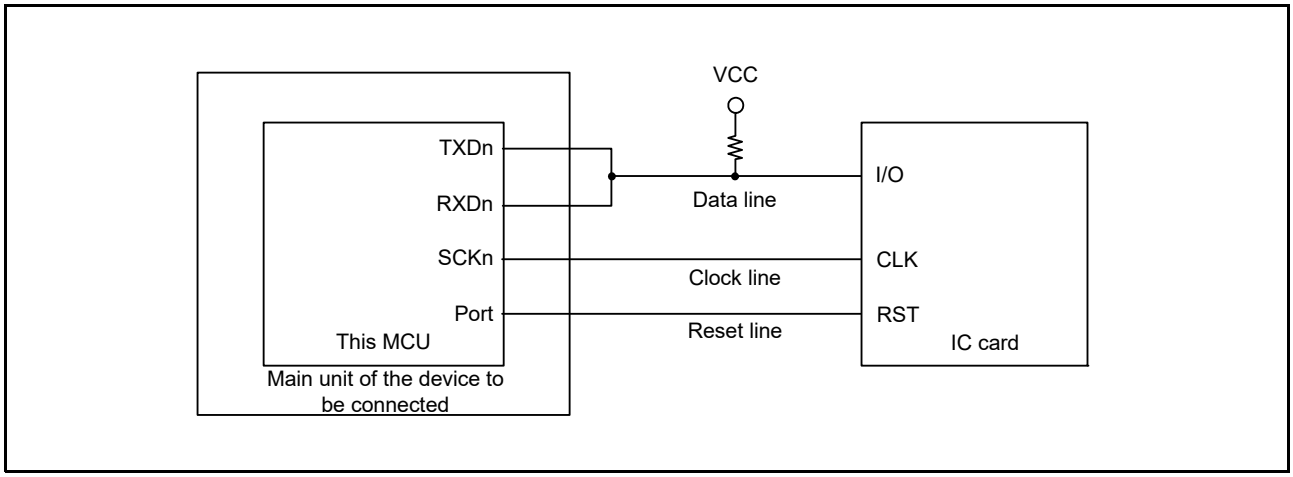


Figure 42.55 Sample Connection with a Smart Card (IC Card)

42.7.2 Data Format (Except in Block Transfer Mode)

Figure 42.56 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

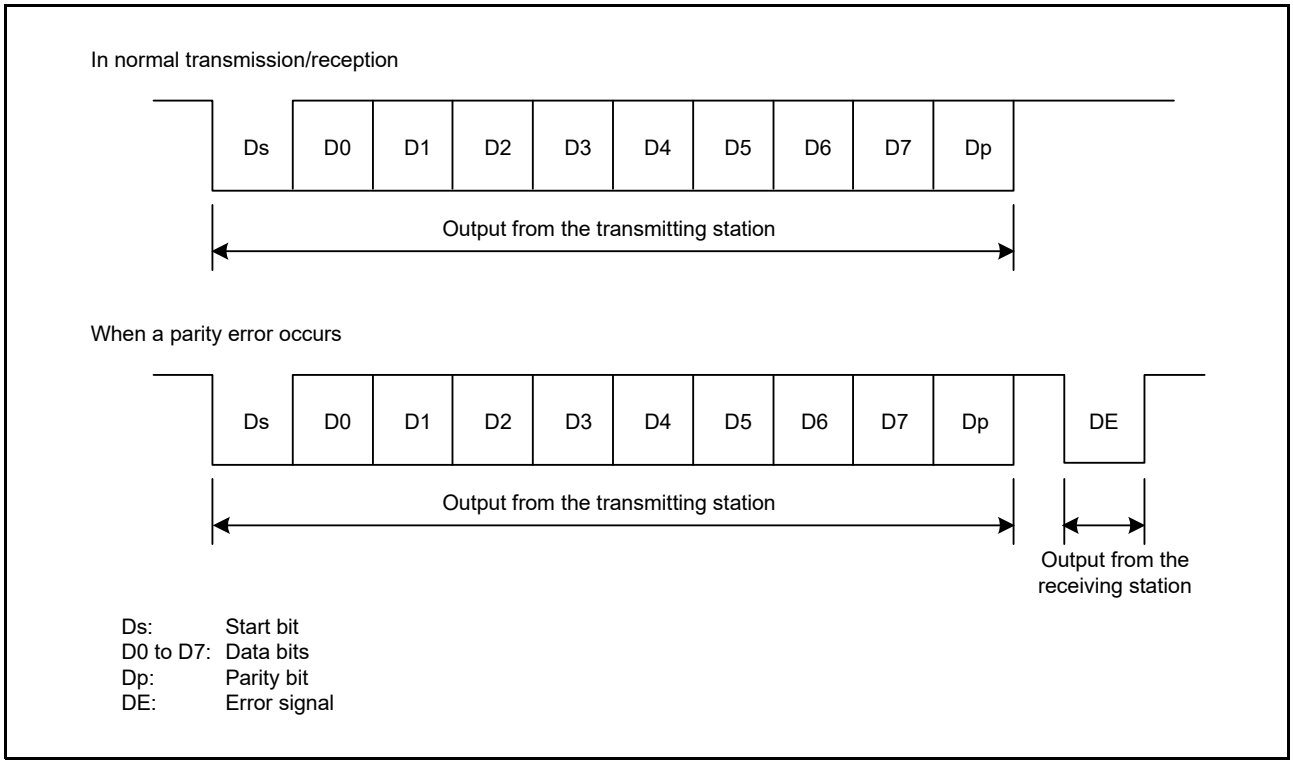


Figure 42.56 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 42.57. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 1 to the SCR3.DDIR bit and 0 to the SCR3.DINV bit. Write 0 to the PM bit in the SCR1 register in order to use even parity, which is prescribed by the smart card standard.

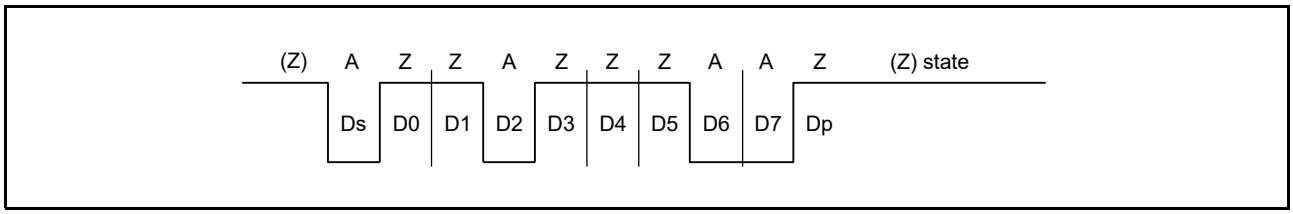


Figure 42.57 Direct Convention (SCR3.DDIR Bit = 1, SCR3.DINV Bit = 0, SCR1.PM Bit = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 42.58. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 0 to the SCR3.DDIR bit and 1 to the SCR3.DINV bit. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the DINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in the SCR1 register to invert the parity bit for both transmission and reception.

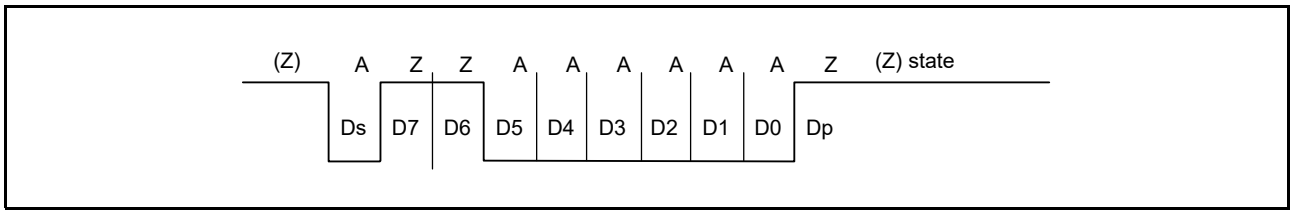


Figure 42.58 Inverse Convention (SCR3.DDIR Bit = 0, SCR3.DINV Bit = 1, SCR1.PM Bit = 1)

42.7.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the APER flag in the SSR register is set by error detection, clear the APER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in the SSR register is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in the SSR register indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

42.7.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive clock in smart card interface mode.

In this mode, the RSCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the SCR2.BCP[2:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 42.59. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 372)} × 100 (%) = 49.866 (%)

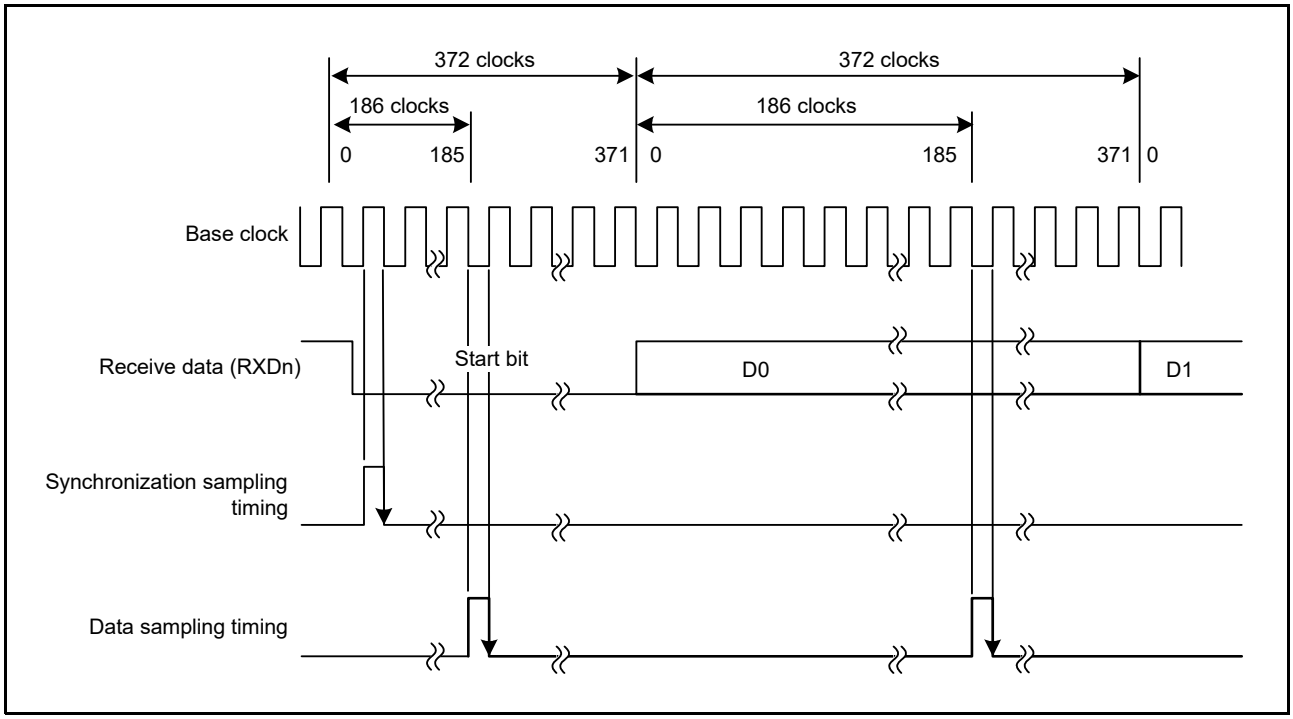


Figure 42.59 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

42.7.5 RSCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write 0 to SCR0.TE bit and SCR0.RE bit (or write the initial value to SCR0 register). And initialize RSCI following example flowchart in Figure 42.60.
Be sure to set the initial value in the TIE, RIE, TE, RE, and TEIE bits in SCR0 register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized. In transmission mode, set 1 to the TE bit and TIE bit simultaneously, then the TXI interrupt request is generated. To change reception mode to transmission mode, first check that reception has completed, and then initialize RSCI. At the end of initialization, set TE bit = 1 and RE bit = 0. Reception completion can be verified by RXI interrupt request, SSR.ORER, or SSR.APER flag. To change transmission mode to reception mode, first check that transmission has completed, and then initialize RSCI. At the end of initialization, set TE bit = 0 and RE bit = 1. Transmission completion can be verified by reading SSR.TEND flag.

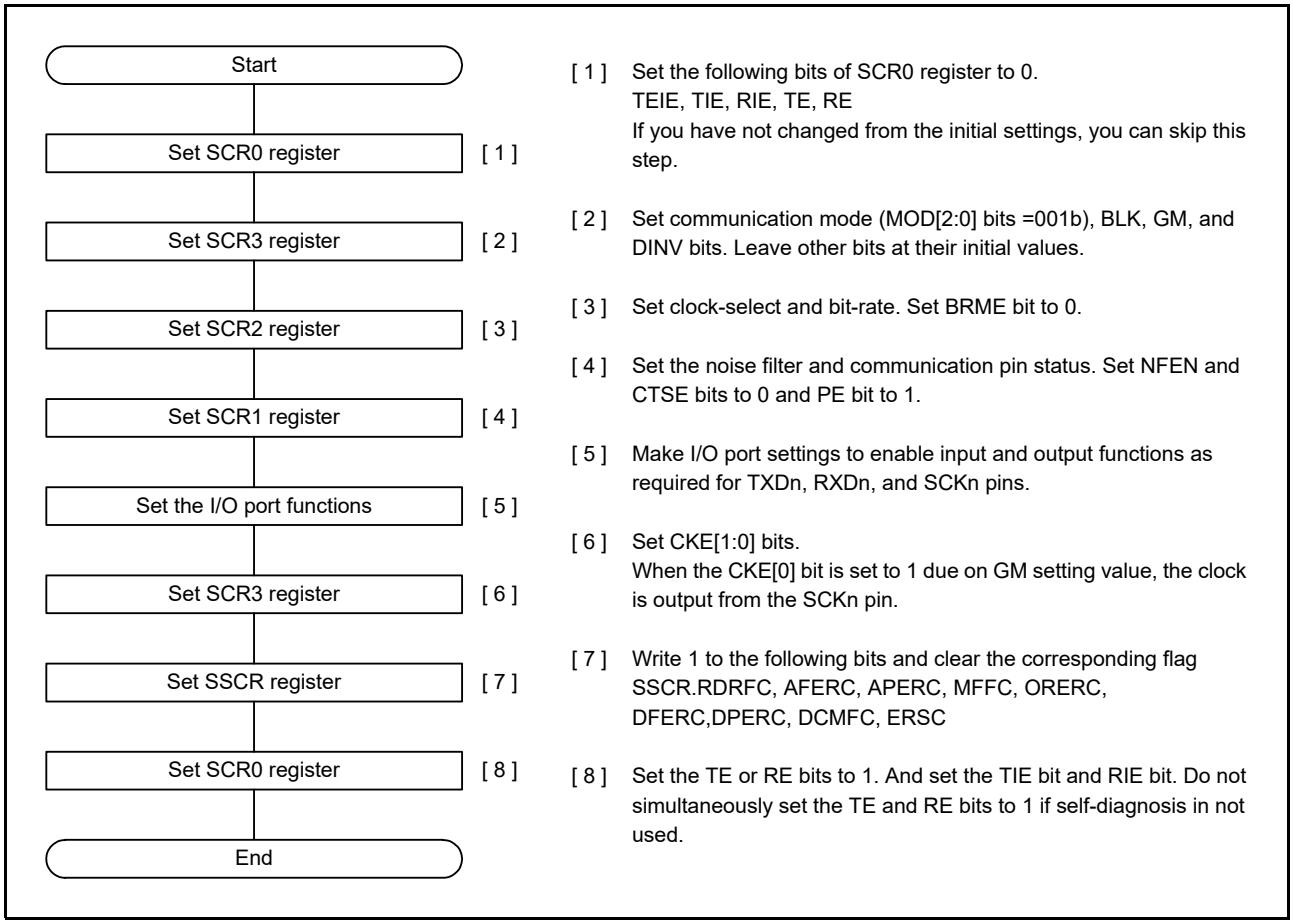


Figure 42.60 Example of RSCI Initialization Flowchart (Smart Card Interface Mode)

Figure 42.61 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when SCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the SCR3.CKE[0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the SCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting SCR3.CKE[0] bit to 1, start data transmission by writing transmit data after setting SCR0.TE bit to 1.
In the smart card interface mode, even if not communicating at SCR0.TE bit = 0 and SCR0.RE bit = 0, the clock is continuously output if the clock output setting is used.

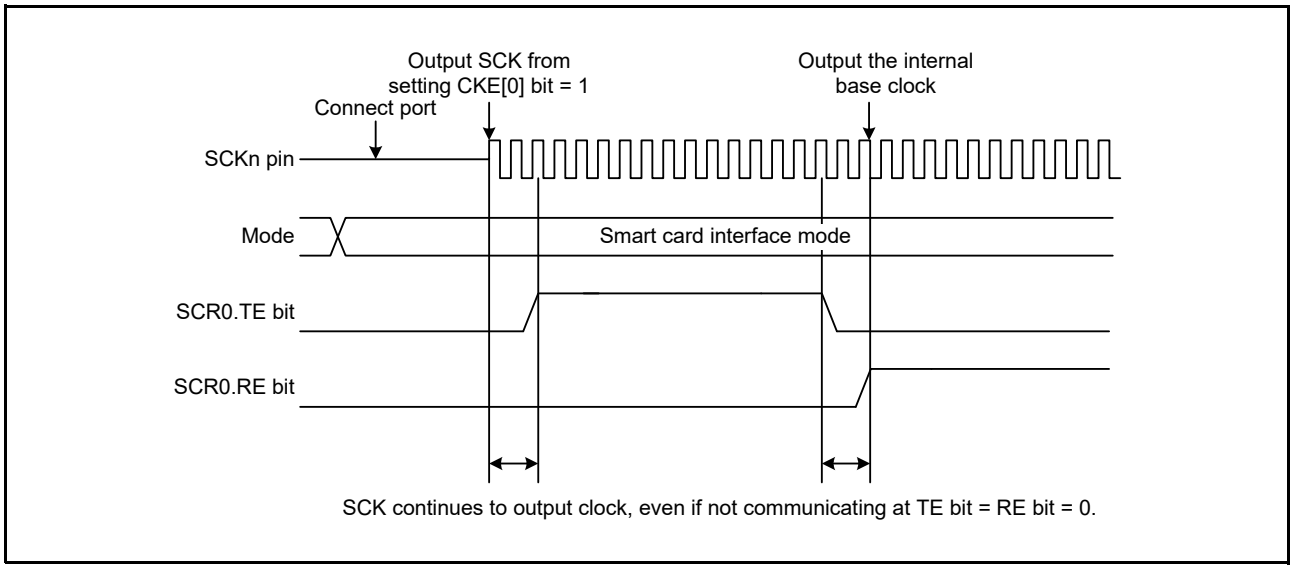
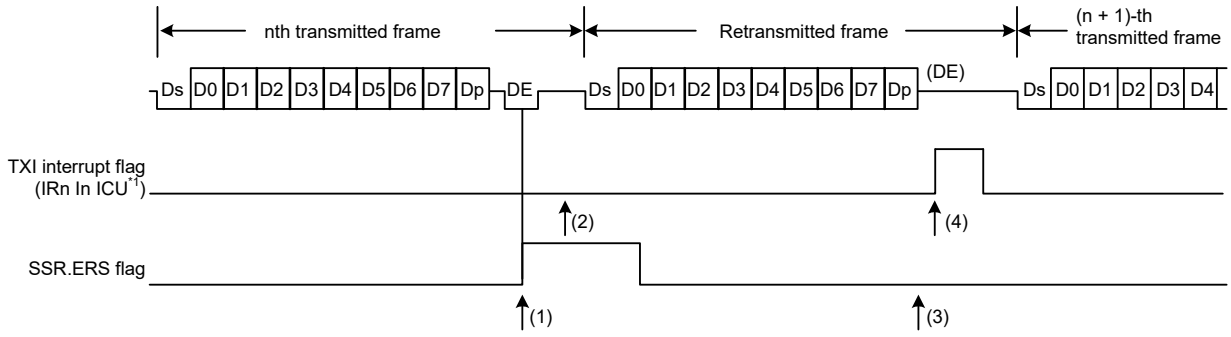


Figure 42.61 Example of Data Transmission Timing in Smart Card Interface Mode

42.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 42.62 shows the data retransmit operation during transmission.

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in the SSR register is set to 1. If the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
(2) For a frame in which an error signal is received, the TEND flag in the SSR register is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.
(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.
(4) In this case, the RSCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the TIE bit in the SCR0 register is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.62 Data Retransmit Operation in RSCI Transmit Mode

Figure 42.64 shows a sample flowchart of serial transmission. All the processing steps are automatically performed by using a TXI interrupt request to activate the DTC or DMAC. When SSR.TEND flag is set to 1 in transmission, if the SCR0.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data. If an error occurs, the RSCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making RSCI settings. For DTC or DMAC settings, refer to section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCA).
Note that SSR.TEND flag is set in different timings depending on the SCR3.GM bit setting. Figure 42.63 shows the timing for setting the TEND flag.

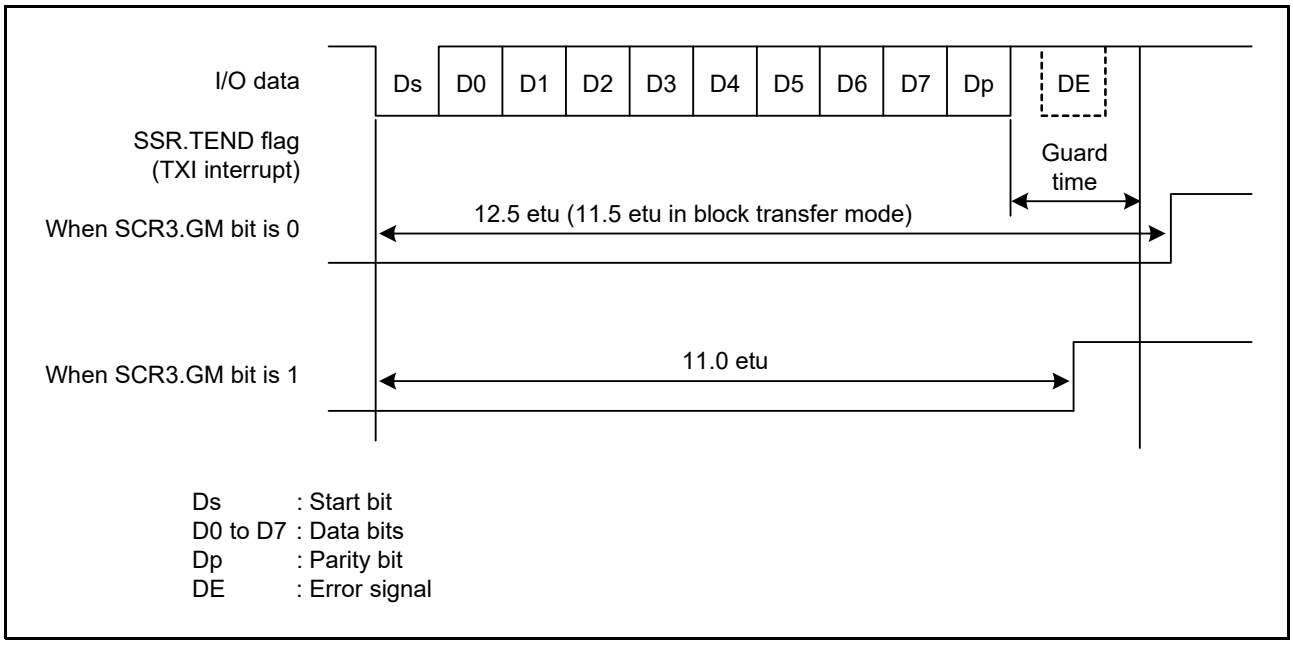


Figure 42.63 Timing for Setting the SSR.TEND Flag during Transmission

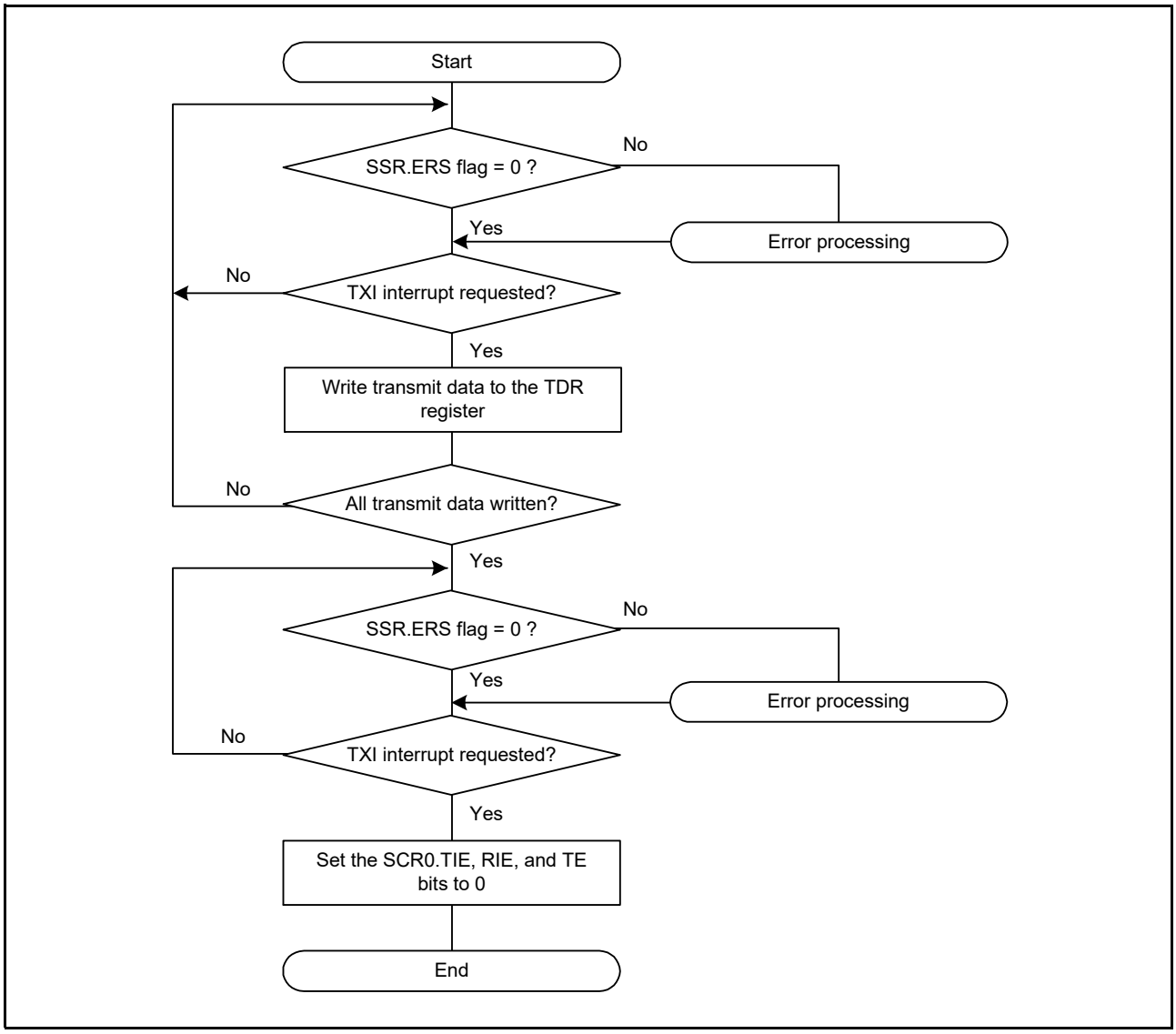


Figure 42.64 Sample Smart Card Interface Transmission Flowchart

42.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 42.65 shows the data retransmit operation in receive mode.

(1) If a parity error is detected in receive data, the APER flag in the SSR register is set to 1. When the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the APER flag to 0 before the next parity bit is sampled.
(2) For a frame in which a parity error is detected, no RXI interrupt is generated.
(3) When no parity error is detected, the APER flag in the SSR register is not set to 1.
(4) In this case, data is determined to have been received successfully. When the RIE bit in the SCR0 register is 1, an RXI interrupt request is generated.

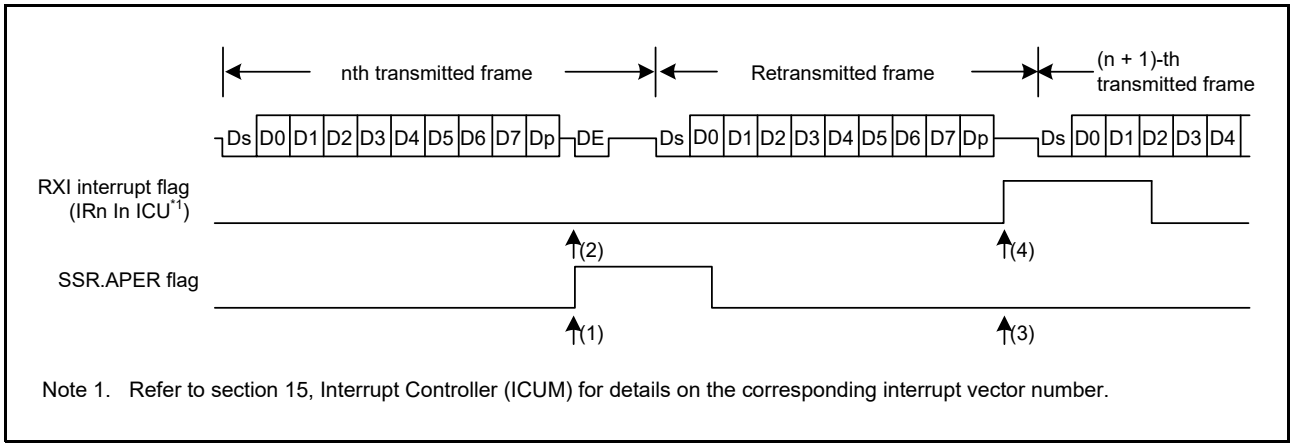


Figure 42.65 Data Retransmit Operation in RSCI Receive Mode (Data Retransmit Operation during Reception)

Figure 42.66 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data. If an error occurs during reception and either the SSR.ORER or APER flag is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred. Even if a parity error occurs and the APER flag is set to 1 during reception, receive data is transferred to RDR register, thus allowing the data to be read. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR register.

Note: For operations in block transfer mode, refer to section 42.3, Operation in Asynchronous Mode.

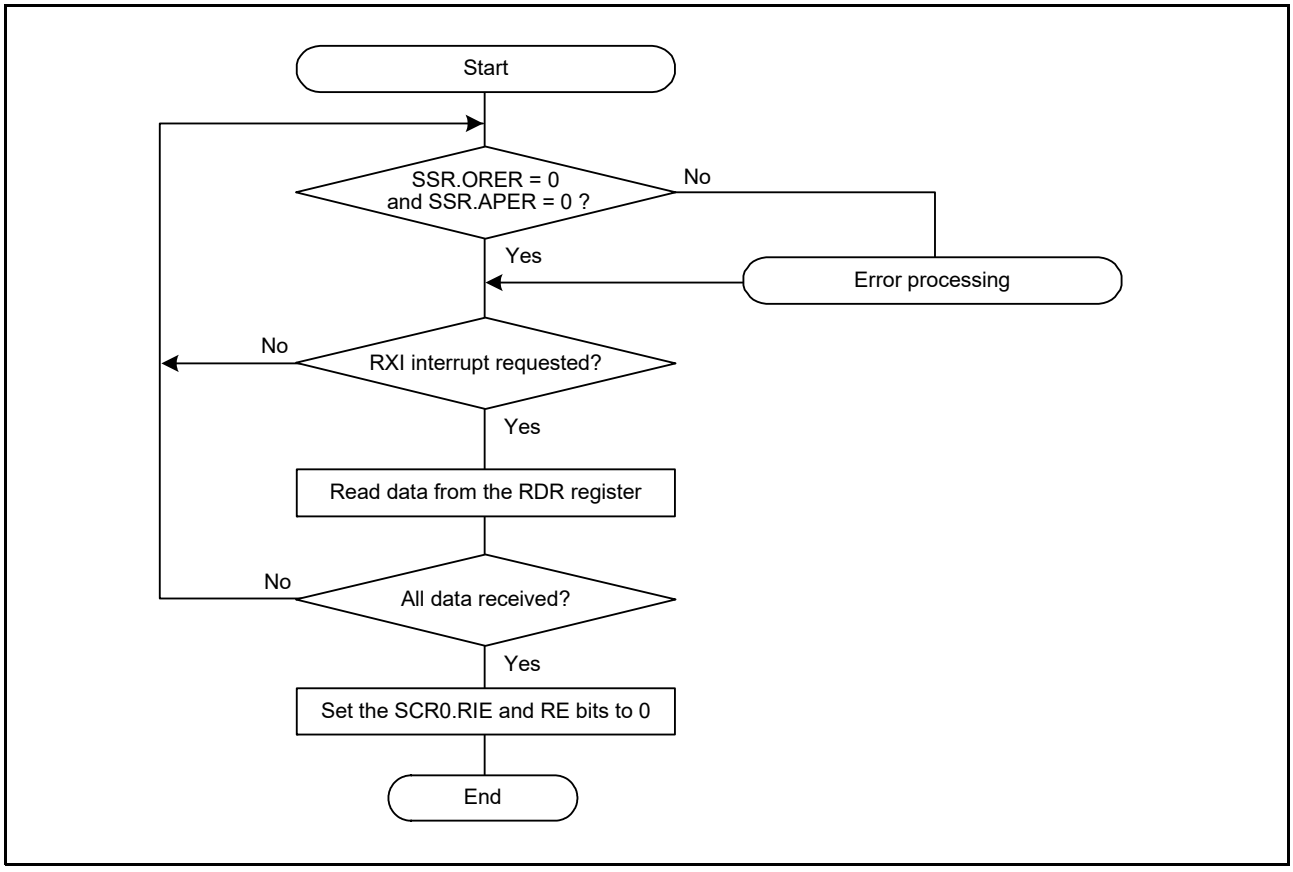


Figure 42.66 Sample Smart Card Interface Reception Flowchart

42.7.8 Clock Output Control

When the SCR3.GM bit is set to 1, the clock output can be controlled by the SCR3.CKE[1:0] bits. Refer to the description of the SCR3.CKE[1:0] bits in section 42.2.8, Control Register 3 (SCR3). When setting the clock output, the base clock described in section 42.7.4, Receive Data Sampling Timing and Reception Margin is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. It is described in section 42.2.7, Control Register 2 (SCR2), the bit rate is set by SCR2.CKS[1:0] bits, SCR2.BCP[2:0] bits and BRR[7:0] bits.
Figure 42.67 shows the timing chart for explaining clock output control. This is an example when the SCR3.CKE[1] bit is set to 0 and the SCR3.CKE[0] bit is controlled.
When the SCR3.GM bit is 0, output control by the SCR3.CKE[0] bit is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.
When the SCR3.GM bit is 1, the output pulse control by the SCR3.CKE[0] bit controls the pulse width set to be based on the state of the base clock.

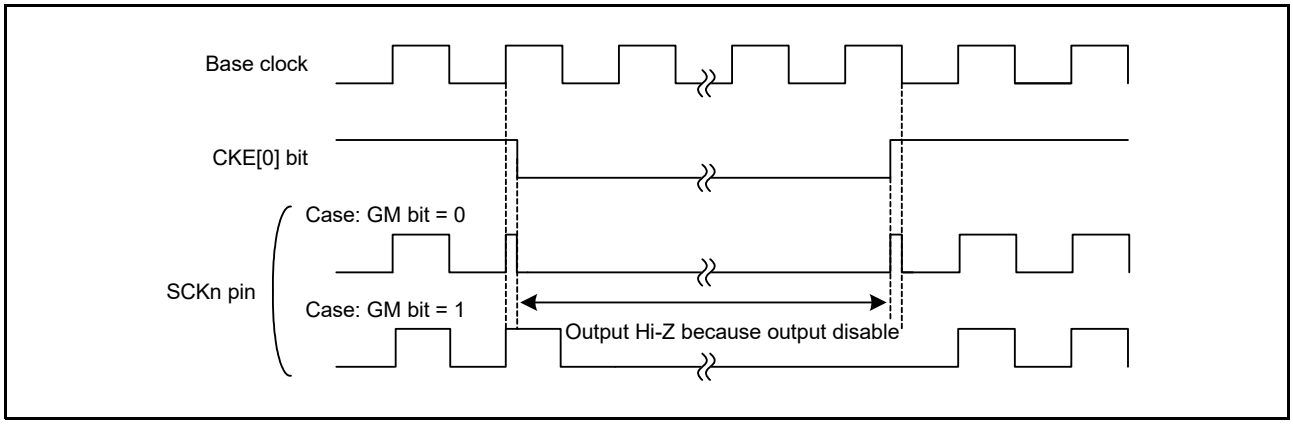


Figure 42.67 Clock Control Timing Chart by SCR3.GM Bit

42.8 Extended Serial Mode

42.8.1 Serial Transfer Protocol

As an extended function of the RSCI, the RSCI supports the serial communication protocol (Figure 42.68) consisting of a Start Frame and an Information Frame. Extended serial mode is enabled by the SCR3.MOD[2:0] bits = 110b. Since the extended serial mode uses the same circuit as the asynchronous mode for transmission/reception control other than Break Field, the basic communication settings required for the asynchronous mode are also required for the extended serial mode (however, set the SCR3.RXDESEL bit to 1).
The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

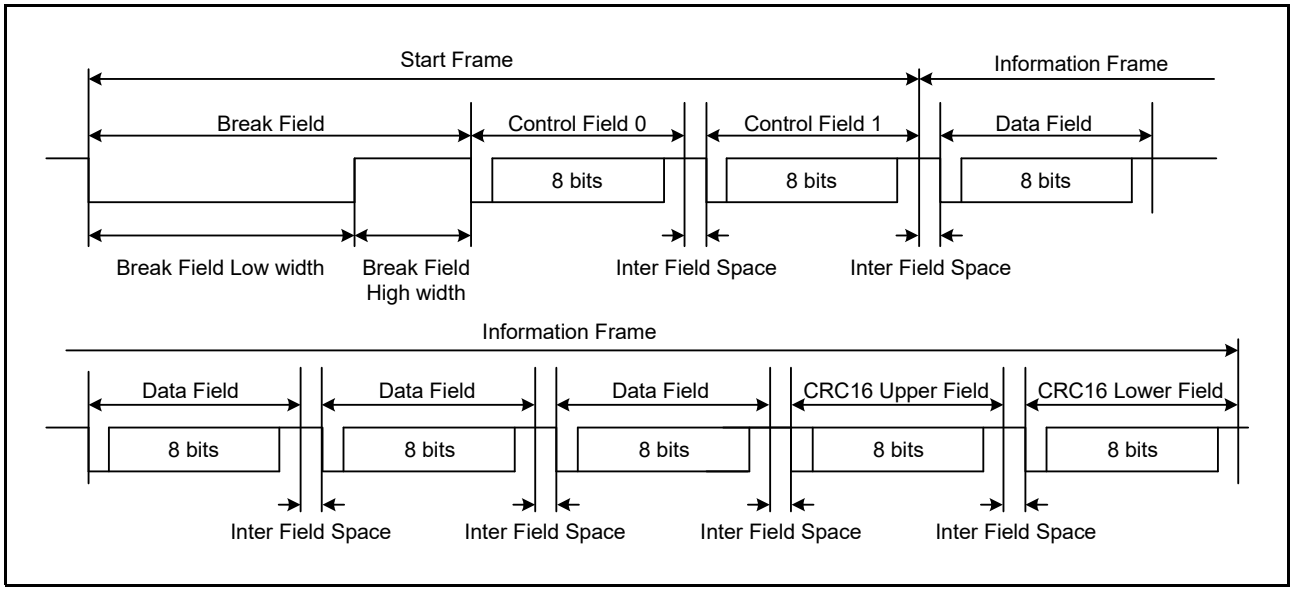


Figure 42.68 Protocol for Serial Transfer by the Extended Serial Mode

The following describes operations in extended serial mode. In this section, operations are described with the following conditions:
Communication pin (RXDn/TXDn) level inversion function: OFF (RINV bit = TINV bit = 0)
When using the communication pin (RXDn/TXDn) level inversion function enabled, replace the RXD and TXD signal levels with their inverted levels.

42.8.2 Transmitting a Start Frame

Figure 42.69 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field 0, and Control Field 1. (Omit Break Field and Control Field 0 according to the Start Frame configuration.)
Figure 42.70 shows a flowchart for Start Frame transmission.
The RSCI operates as follows during Start Frame transmission.

(1) Make the initial settings for the RSCI according to the RSCI initialization flow (Figure 42.9) in asynchronous mode. In extended serial mode, do not set SCR0.TE and TIE bits to 1 at the same time to avoid TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the RSCI initialization flow (asynchronous mode) procedure [10].

- Set the bits except SCR0.TIE bit. (SCR0.TIE bit = 0, SCR0.TE bit = 1, and SCR0.RE bit = 0)
- Set SCR0.TIE bit to 1.

(2) When 1 is written to TCST, the timer in the extended serial module starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0] bits. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
Writing 0 to XCR1.TCST bit suspends output of the Break Field. After the suspension, set SCR0.TE bit = 0 and turn off the transmission.

(3) When the extended serial module timer count value matches the set XCR2.BFLW[15:0] bits value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1*1. Furthermore, if XCR0.BFOIE bit has been set to 1 at this time, a TXI interrupt is generated.

(4) After confirming the BFOF flag is set to 1, send the Control Field 0 data.*2

(5) After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.

(6) After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF flag is set to 1, if 1 is written to XCR1.TCST bit without clearing it, no TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF flag before writing 1 to XCR1.TCST bit.

Note 2. LIN communication requires a Break delimiter (IDLE period) of 1 bit or more from the end of Break Field transmission until the next data transmission starts. For this reason, the Break delimiter length is counted upon completion of Break Field transmission. If transmit data is written while the Break delimiter length is being counted, transmission does not start until the Break delimiter length counting is completed. When transmit data is written after the Break delimiter length has been counted, transmission starts at the same timing as normal data transmission.
Break delimiter length count time after Break Field transmission: 1-bit to 2-bit length (SCR3.STOP bit = 0) 2-bit to 3-bit length (SCR3.STOP bit = 1)

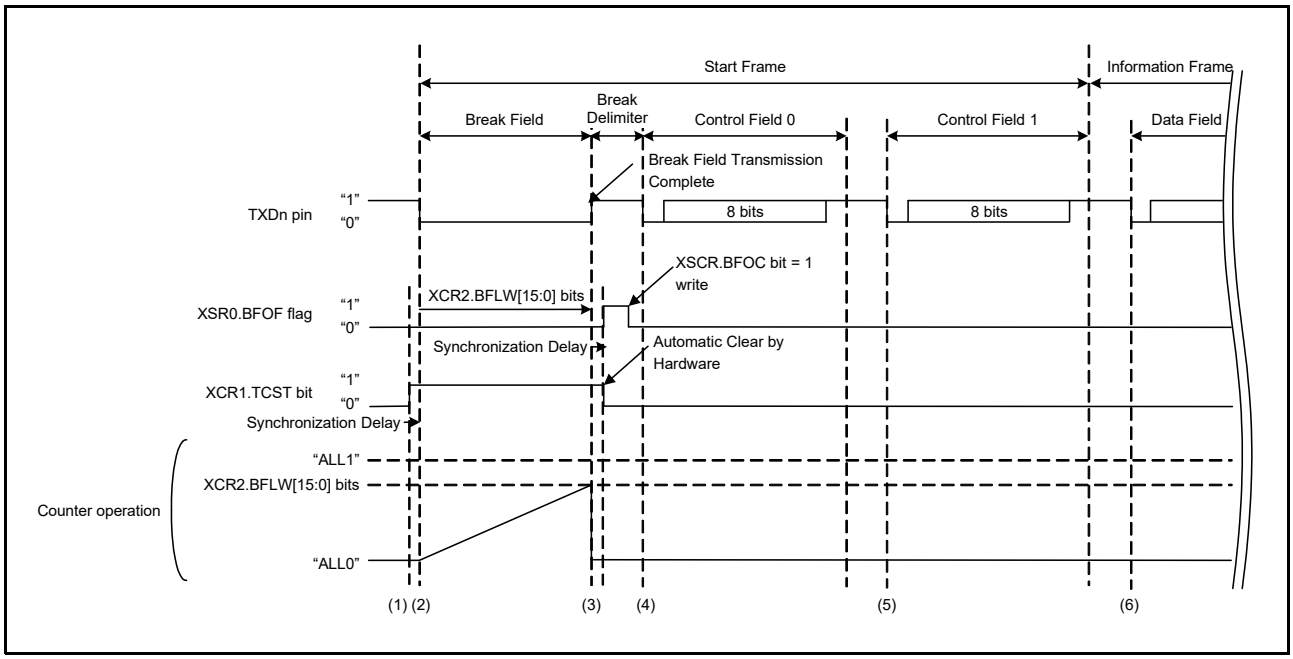


Figure 42.69 Example of Operations When Transmitting a Start Frame

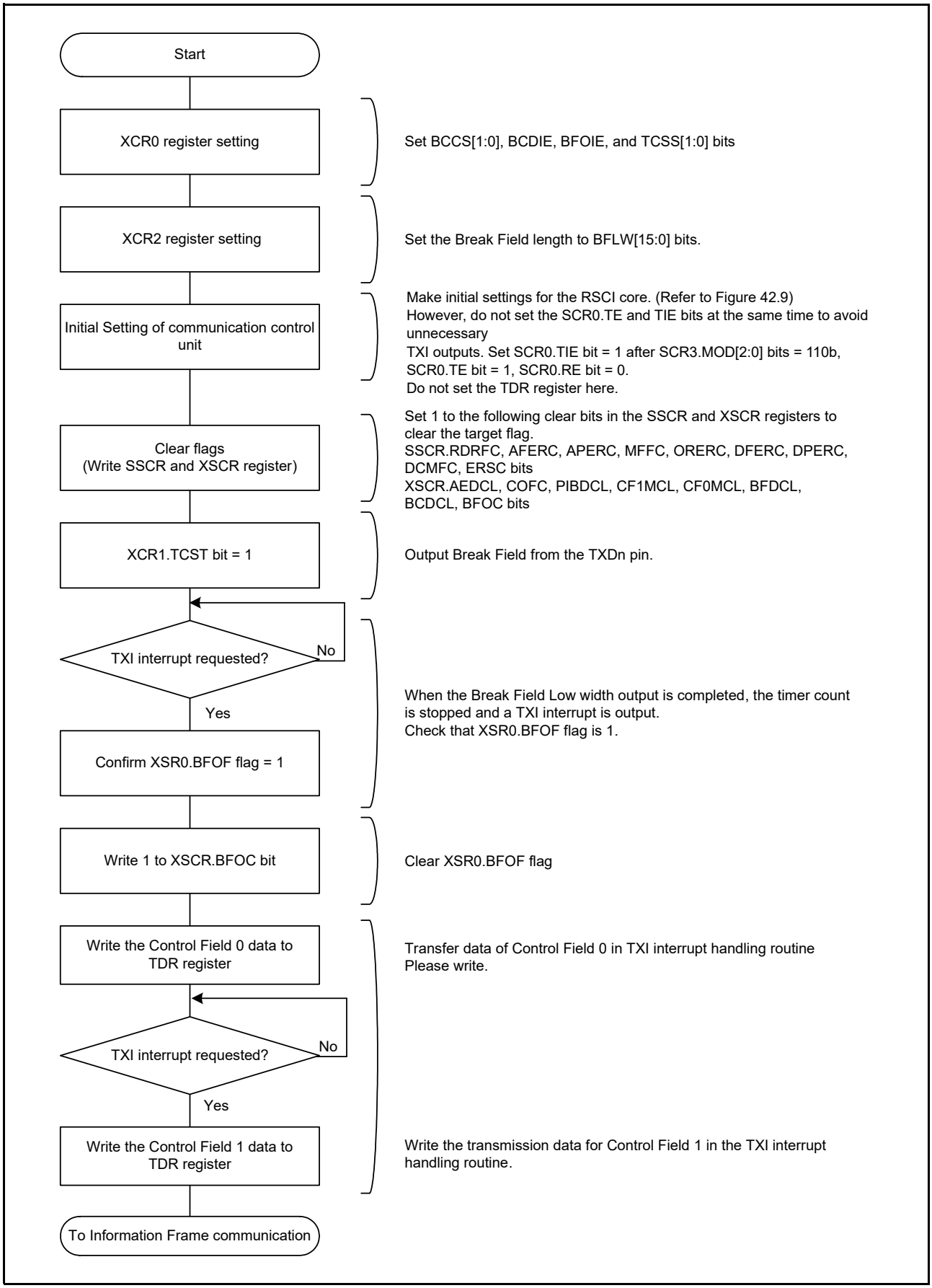


Figure 42.70 Example of Start Frame Transmission

42.8.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 42.40.

Table 42.40 Structures of Start Frames

XCR0		Start Frame Configuration
BFE	CF0RE	
0	0	Control Field1 (8 bits) → Information Frame
0	1	Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame
1	0	Break Field (Low width) → Control Field1 (8 bits) → Information Frame
1	1	Break Field (Low width) → Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame

42.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used)

Figure 42.71 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 42.72 shows an example of reception to detect the Break Field during Control Field 1. Figure 42.73 shows a flowchart to receive the Start Frame, and Figure 42.74 shows a state transition diagram.
When receiving the Start Frame, the RSCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

(1) Writing 1 to XCR1.SDST bit makes it possible to detect the Start Frame. When XCR0.BFE bit = 1, RXD input to the RSCI core is disabled until the Break Field is detected (because XSR0.RXDSF flag is set to 1). Once the Break Field is detected, RXD input can be received to the RSCI core (XSR0.RXDSF flag = 0).
(2) When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
(3) When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF flag is set to 1. If XCR0.BFDIE bit has been set to 1 at this time, a BFD interrupt is generated.
The timer continues counting until the RXD rising edge or counter overflow.
(4) After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.CCV[15:0] bits when XCR1.BRME bit = 0. At this time, XSR0.RXDSF flag is cleared to 0 and the RSCI core starts receiving the RXD input.
(5) The RSCI core starts receiving Control Field 0. Because the extended serial continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the RSCI core waits for reception of Control Field 0 again (Figure 42.72).
(6) When Control Field 0 has been received, an RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0] bits. When the received data matches the set XCR2.CF0D[7:0] bits value, XSR0.CF0MF flag is set to 1. If the received data differs from the set XCR2.CF0D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.
(7) The RSCI core starts receiving Control Field 1. When BFE bit = 1, the Break Field detection function is continuously enabled while SDST bit = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the RSCI core waits for reception of Control Field 0 again.
(8) When Control Field 1 has been received, an RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0] bits. When the received data matches the set XCR1.PCF1D[7:0] bits value or the set

XCR1.SCF1D[7:0] bits value, XSR0.CF1MF flag is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

(10) When communication is completed, write 0 to XCR1.SDST bit and 0 to SCR0.RE bit to stop reception.

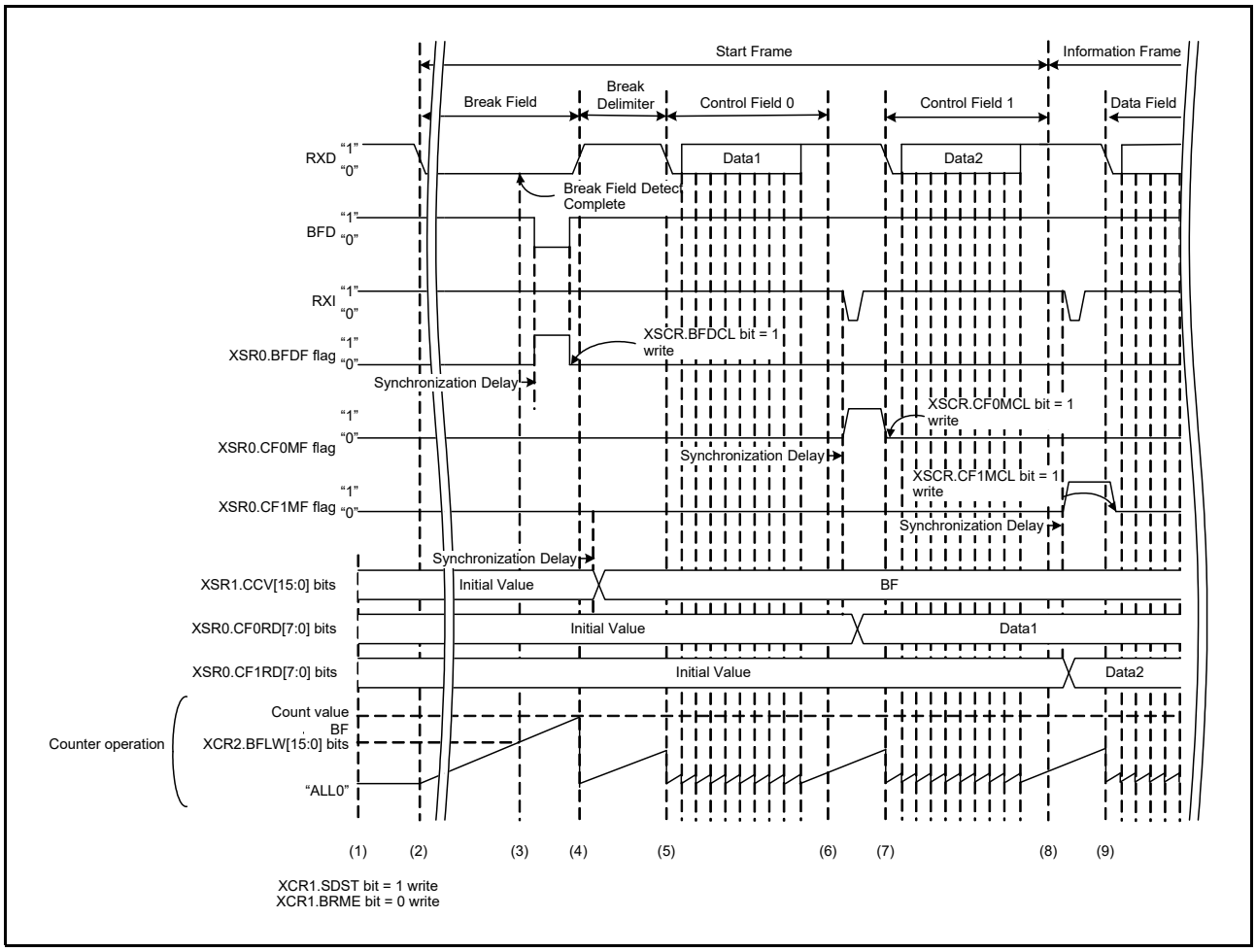


Figure 42.71 Normal Reception Example of Start Frame (PIB Not Used)

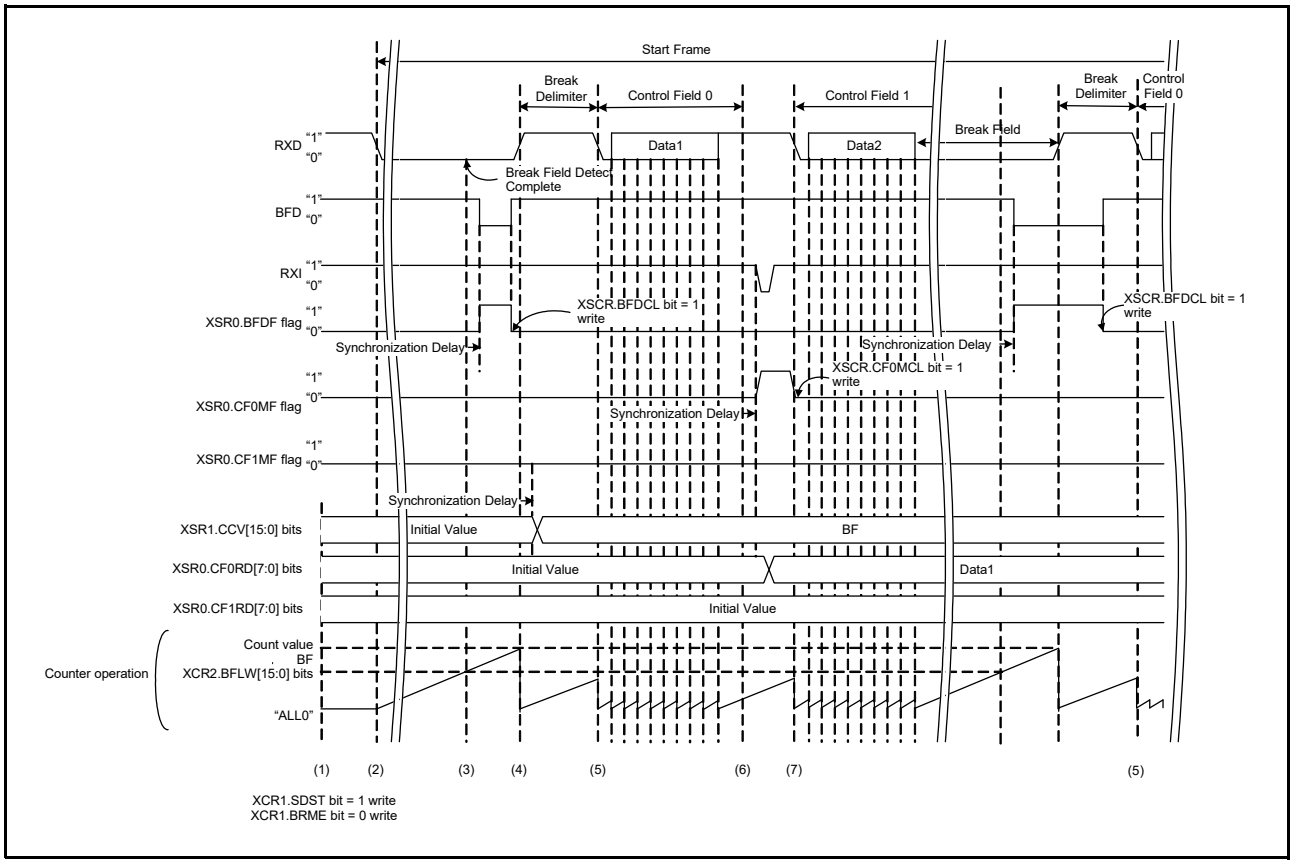


Figure 42.72 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

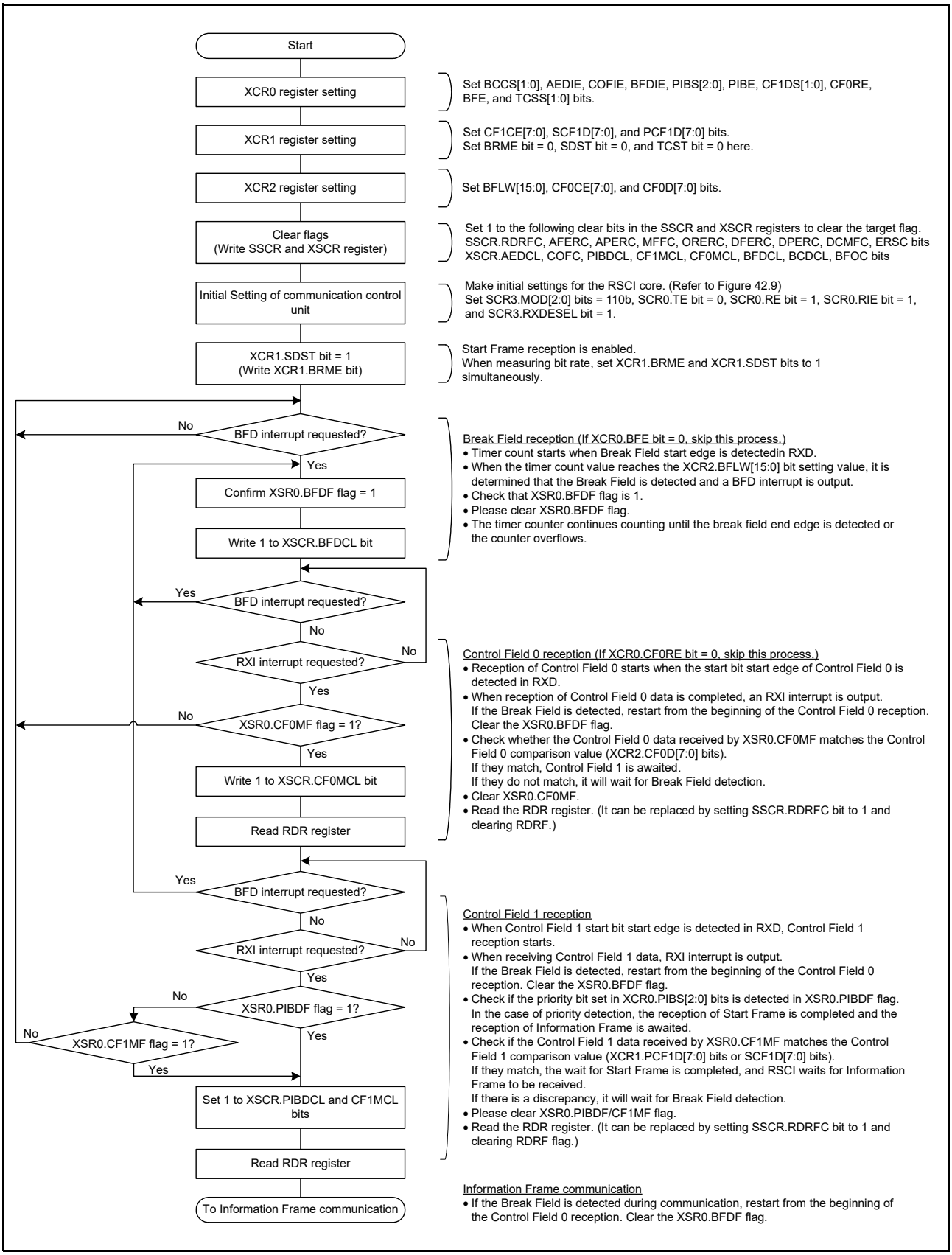


Figure 42.73 Sample Flowchart for Reception of a Start Frame

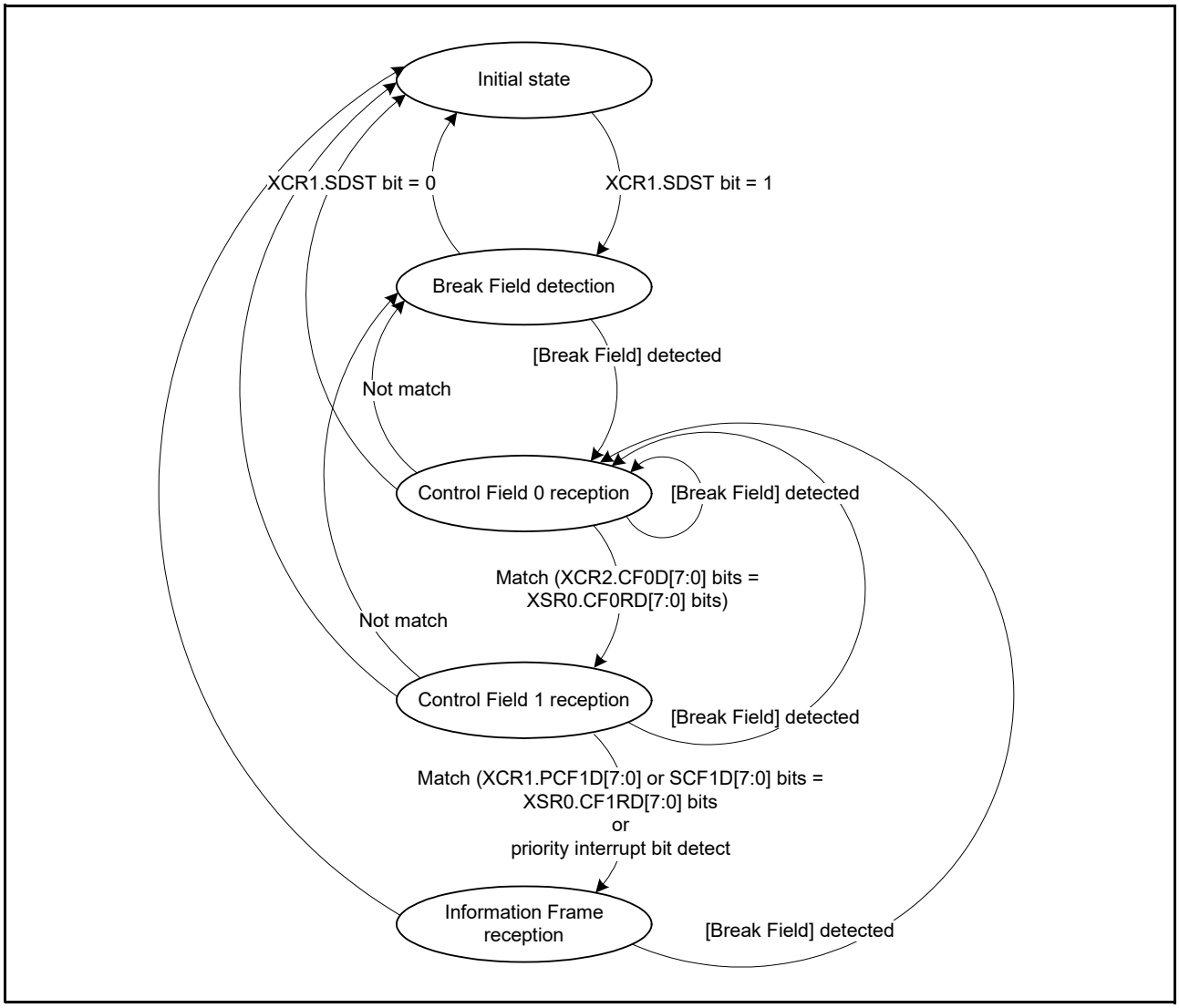


Figure 42.74 State Transition Diagram of Start Frame Reception

42.8.3.2 Priority Interrupt Bit

Figure 42.75 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE bit to 1.

The RSCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as in Figure 42.71, for Start Frame reception.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] bits value, XSR0.PIBDF flag is set to 1and the RSCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value and the priority interrupt bit is not detected, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

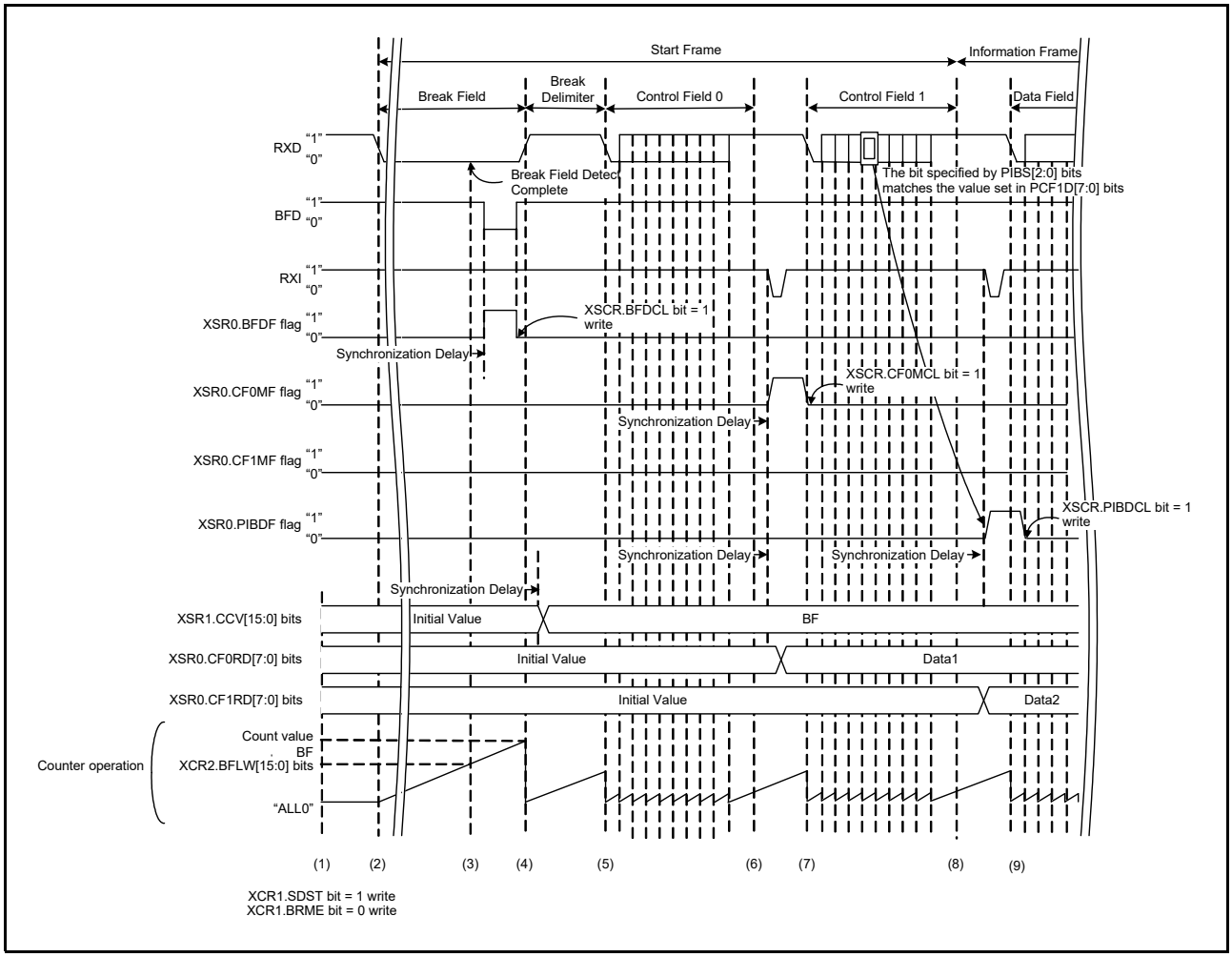


Figure 42.75 Start Frame Reception Example (Priority Interrupt Bit Used)

42.8.4 Detection of Bus Collisions

In extended serial mode (SCR3.MOD[2:0] bits = 110b) when TE bit = 1, the bus conflict detection function works during Break Field transmission and during data transmission.

Figure 42.76 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0] bits. When a mismatch occurs three times in a row, XSR0.BCDF flag is set to 1, and if XCR0.BCDIE bit has been set to 1 at this time, an ERI interrupt is generated.

When an ERI interrupt is generated, stop transmission according to Figure 42.77. Check the bus state to decide whether to resume transmission.

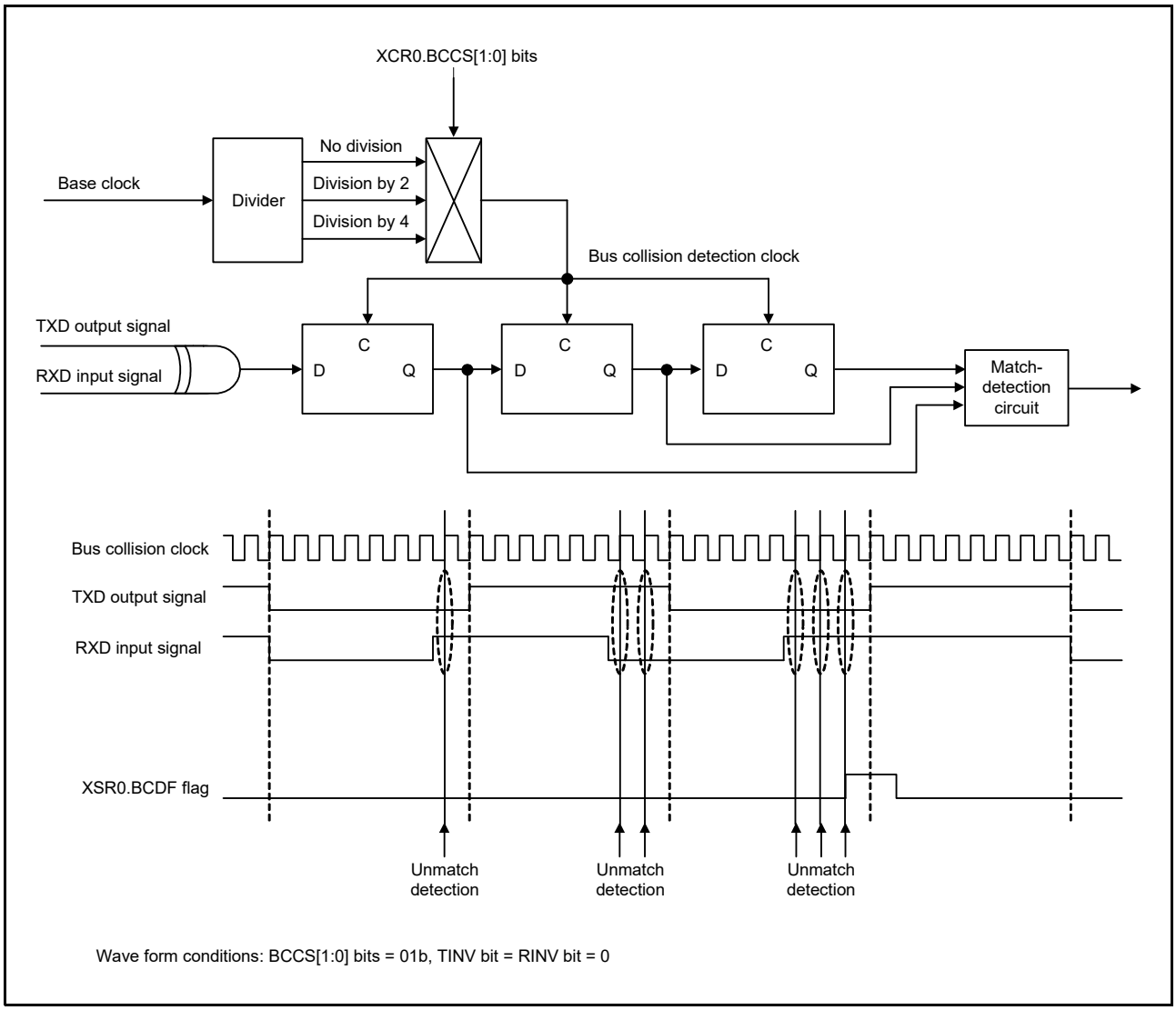


Figure 42.76 Example of Operations with Bus Collision Detection

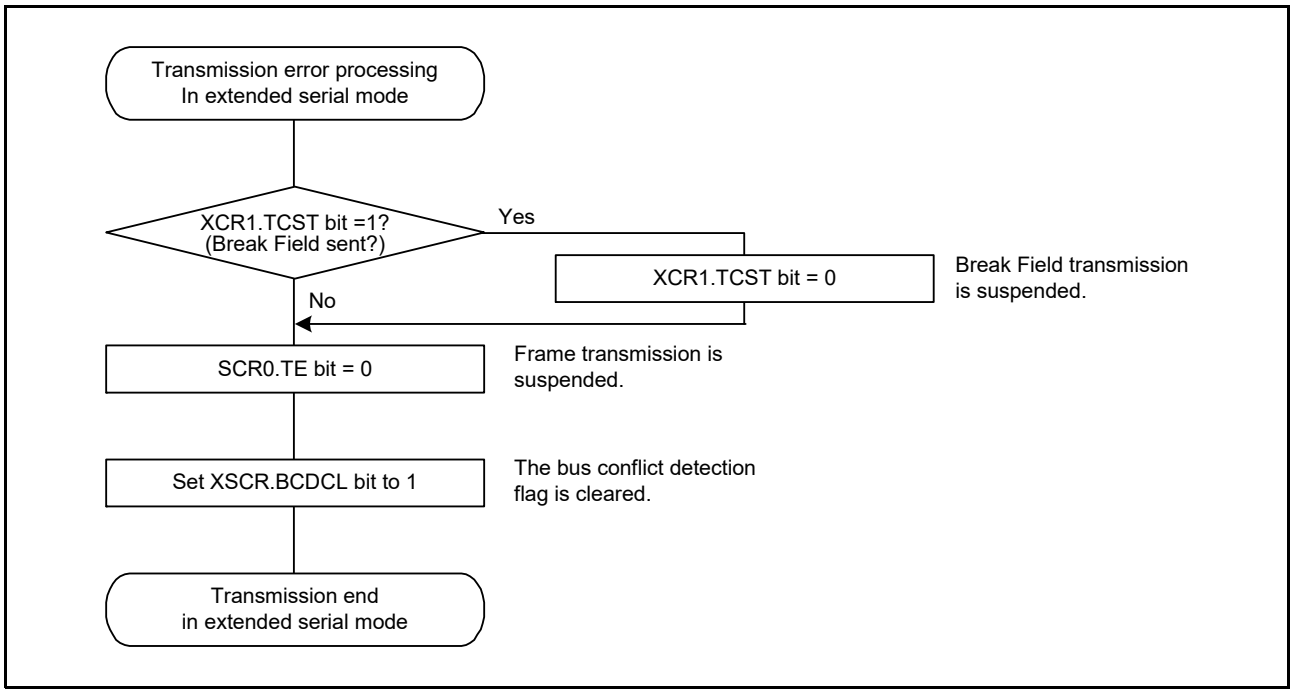


Figure 42.77 ERI Interrupt Handling Flow at Transmission in Extended Serial Mode

42.8.5 Bit Rate Measurement

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 42.78 shows an operation example of the bit rate measurement function.

(1) Writing 1 to XCR1.SDST and XCR1.BRME bits enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BRME and XCR1.SDST bits to 1 simultaneously, only when measuring bit rate.
(2) Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.CCV[15:0] bits.
(3) The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.CCV[15:0] bits.
(4) The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE bit has been set to 1 at this time, an AED interrupt is output. The start bit count value is stored in XSR1.CCV[15:0] bits. The XSR1.CCV[15:0] value is retained until the effective capture value is read.
(5) Even if an effective edge is input from the RXD input pin, the count value of this effective edge timing is not captured because the XSR1.CCV[15:0] bits value has not been read and retention has not been released. In this case, an AED interrupt is not output.
(6) The XSR1.CCV[15:0] value is read. Then the retention of XSR1.CCV[15:0] bits is released and the XSR0.AEDF flag is cleared by hardware.
(7) Because the retention of XSR1.CCV[15:0] bits has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE bit has been set to 1, an AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the RSCI settings.
(8) To disable bit rate measurement, write 0 to XCR1.BRME bit.
(9) The XSR0.AEDF value and the XSR1.CCV[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

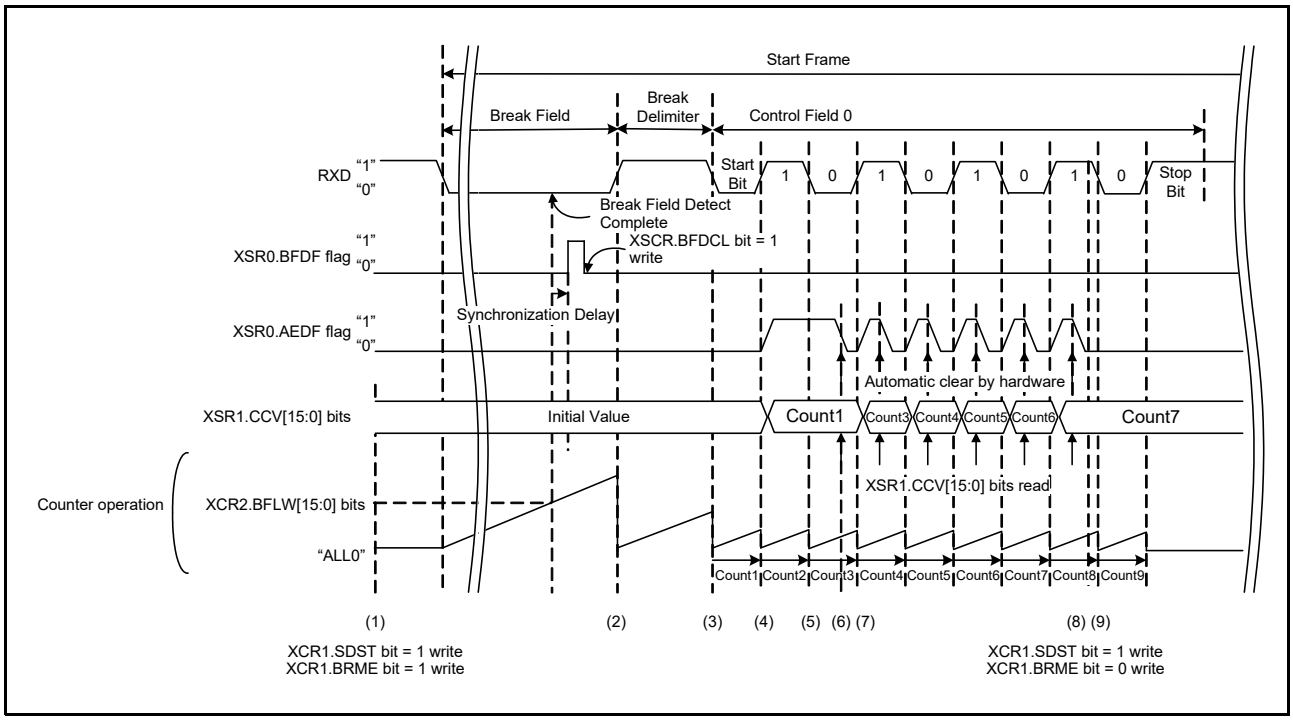


Figure 42.78 Operation Example of the Bit Rate Measurement Function

42.9 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. The frame following the start condition or restart condition is the slave-address frame, which is used by the master device to specify a slave device with which it is communicating. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8-bit data of each frame is transmitted from MSB.
Figure 42.79 shows I2C bus format, and Figure 42.80 shows the timing of I2C.

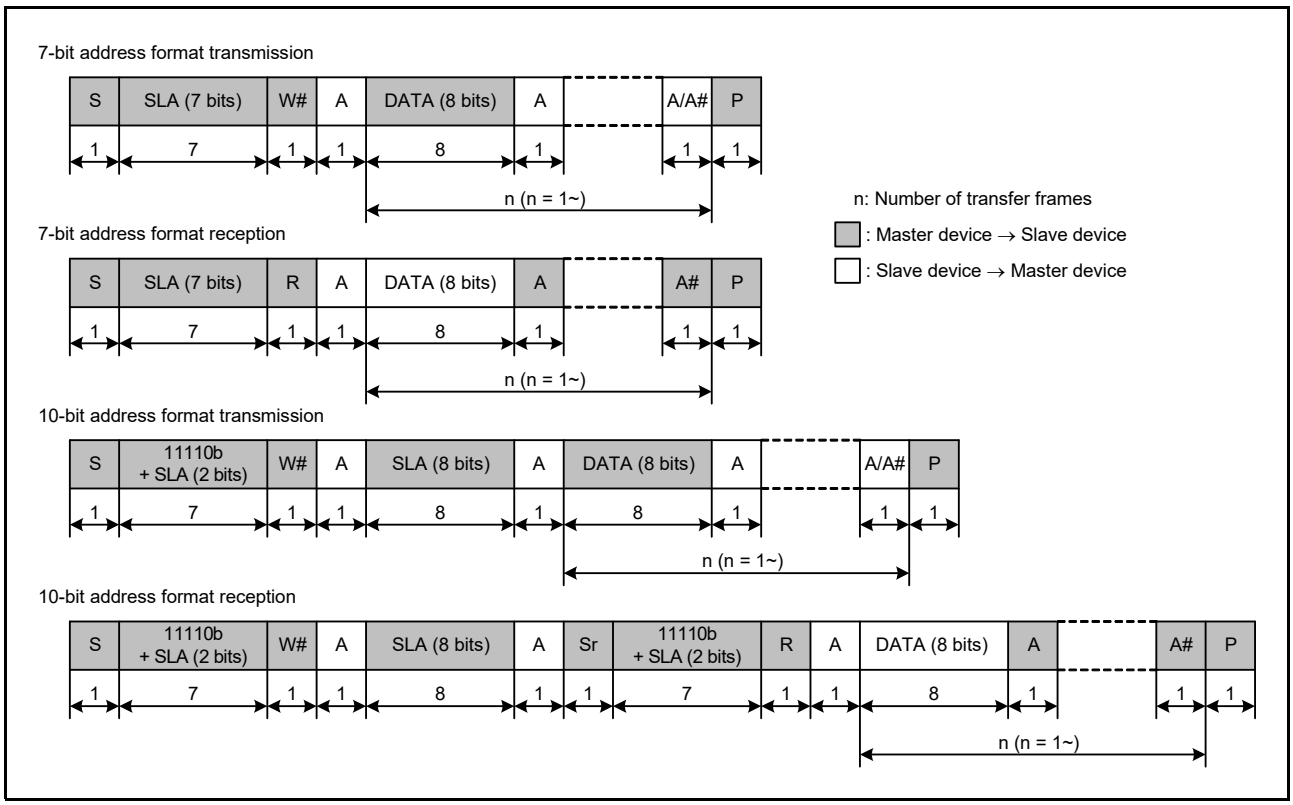


Figure 42.79 I2C-Bus Format

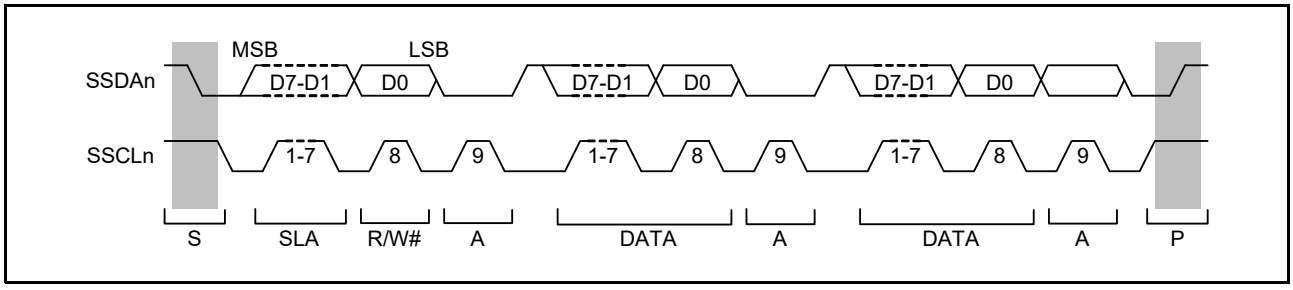


Figure 42.80 I2C-Bus Timing (When SLA is 7 Bits)

S: Indicates a start condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
SLA: Indicates a slave address. The master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). When R/W# is high, the transfer direction is from the slave device to the master device. When R/W# is low, the transfer direction is from the master device to the slave device.
A/A#: Indicates an acknowledge. This is returned by the slave device for master transmission and by the master device for master reception. The low level indicates ACK and the high level indicates NACK.
Sr: Indicates a restart condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
DATA: Indicates the received or transmitted data.
P: Indicates a stop condition. The master device changing the level on the SSDAn line from the low to the high level when the SSCLn line is at the high level.

42.9.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to SIMR.IICSTAREQ bit causes the generation of a start condition. The following operations are done at the generation of a start condition.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to SIMR.IICRSTAREQ bit causes the generation of a restart condition. The following operations are done at the generation of a restart condition.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to SIMR.IICSTPREQ bit causes the generation of a stop condition. The following operations are done at the generation of a stop condition.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSDAn is released (transition from the low to the high level), the SIMR.IICSTPREQ bit is set (to 0), and a stop condition generated interrupt is output.

Figure 42.81 shows the timing of operations in the generation of start, restart and stop conditions.

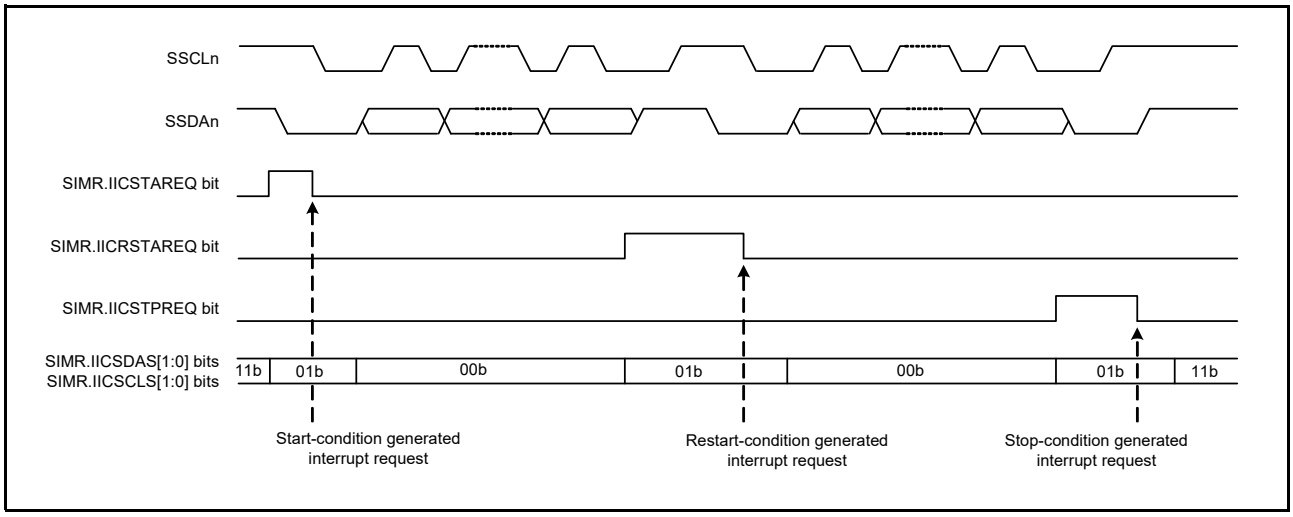


Figure 42.81 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

42.9.2 Clock Synchronization

The slave device of the communication partner may make SSCLn line Low-level with a view to insert a wait. Setting the SIMR.IICCSC bit to 1, applies control to obtain synchronization when the levels of the internal SCL signal and the level being input on the SSCLn pin differ.

When the SIMR.IICCSC bit is set to 1, the level of the internal SCL signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total time which contains the SSCLn input delay, the noise filtering delay of the SSCLn pin (2 or 3 cycles of the filtering clock), and the internal processing delay (1 or 2 cycles of TCLK). The period at high level of the internal SCL signal is extended even if other devices are not placing the low level on the SSCLn line.

If the SIMR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SCL signal. If the SIMR.IICCSC bit is 0, synchronization with the internal SCL signal is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SCL signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SCL signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed.

Figure 42.82 shows an example of operations to synchronize the clocks.

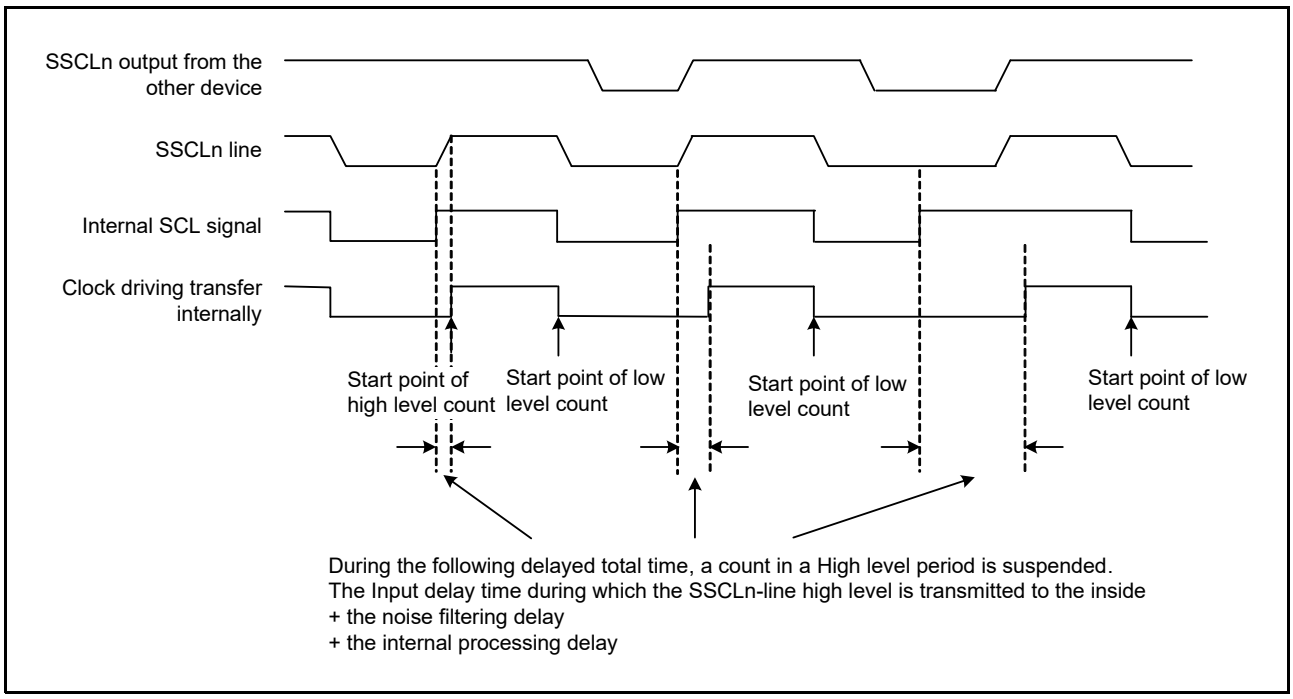


Figure 42.82 Example of Operations for Clock Synchronization

42.9.3 SDA Output Delay

The SIMR.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. The delay-time settings are selectable from 0 to 31 cycles of the clock signal from the on-chip baud rate generator (The base is TCLK and selected the divided clock by the SCR2.CKS[1:0] bits). About Start/Restart/Stop conditions, 8bit-transmission data and acknowledge, the SSDAn pin output can be delayed.

If the SDA output delay is shorter than the falling time of the SSCLn output pin, the change of the SSDAn output pin will start while the SSCLn output pin level is falling, then there is a possibility of erroneous operation for slave devices. Ensure setting the SDA output delay greater than the SSCLn maximum falling time. (300 ns for I^2C normal/fast mode.)

Figure 42.83 shows the timing of delays in SDA output.

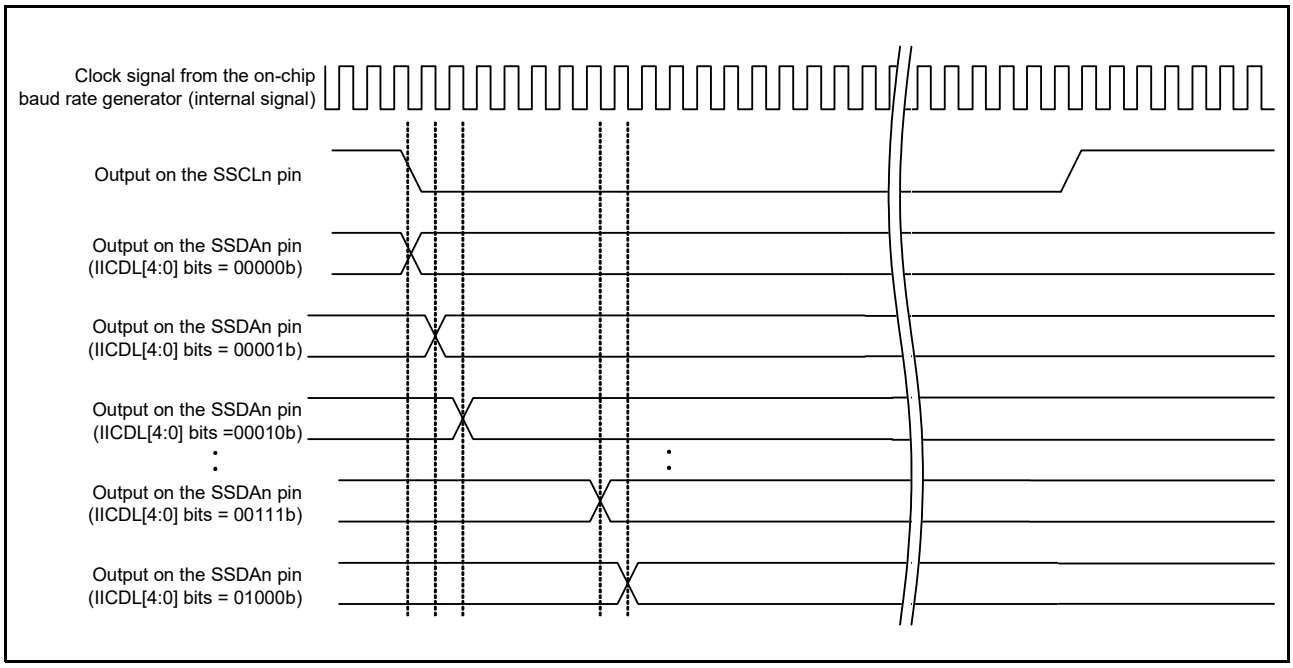
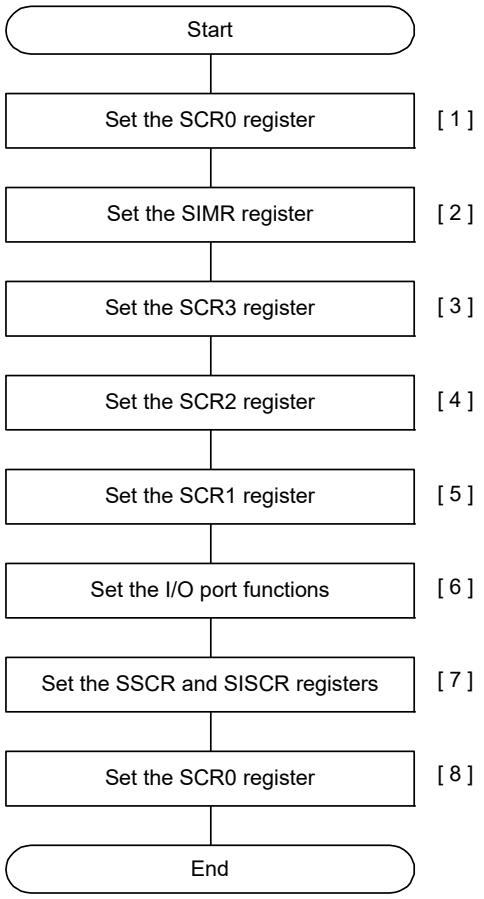


Figure 42.83 Timing of Delays in SDA Output

42.9.4 RSCI Initialization (Simple I2C Mode)

Write initial value (0000 0000h) to SCR0, then initialize RSCI according to Figure 42.84.
When changing the operating mode, transfer format, and so on, be sure to set 0 to SCR0.TE bit and SCR0.RE bit before proceeding with the changes. (Or write initial value to SCR0 again.) In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the IICSDAS[1:0] and IICSCLS[1:0] bits to 11b.
Set the IICDL[4:0] and IICINTM bits as required.
Set the IICACKT and IICCSC bits to 1.

[3] Set the transmission/reception format as the communication mode (MOD[2:0] bits = 100b) and CKE[1:0] bits = 00b.

[4] Set the bit rate modulation function*1, the clock selection, and the bit rate.

[5] Set noise filter, communication pin status, parity check, CTS/RTS function.

[6] Set I/O port settings that allow use (on NMOS opendrain output pins and Hi-Z) of the SSCLn and SSDAn pin functions.

[7] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
SISCR.IICSTIFC

[8] Set the TE and RE bits to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
(for transmission and when the SIMR.IICINTM bit is 1, clear the RIE bit.) Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note 1. If you do not use the bit rate modulation unction, you do not need to set it.

Note: Set the SCR0.TE and RE bits to 0 or 1 at the same time.

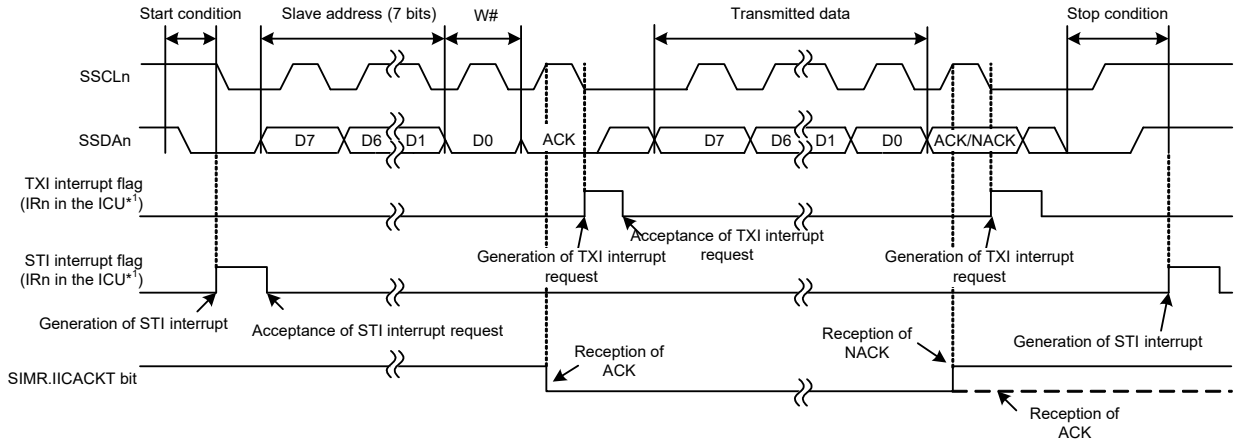
Figure 42.84 Example of the Flowchart of RSCI Initialization (for Simple I2C Mode)

42.9.5 Operation in Master Transmission (Simple I2C Mode)

Figure 42.85 and Figure 42.86 show examples of operations in master transmission, Figure 42.87 to Figure 42.89 show the example flowcharts. See Table 42.47 about the STI interrupt.

Figure 42.85 shows the operation example when SIMR.IICINTM bit is 1 (Reception/Transmission interrupt are in use). In this case, you can start DMAC or DTC by TXI interrupt. However, if you use DMAC or DTC, ACK/NACK can not be confirmed. So, if you want to confirm ACK/NACK, prepare transmit data by CPU. In simple I2C mode, TXI interrupt is generated when communication of one frame is completed. And it isn't used reception interrupt in master transmission, so the SCR0.RIE bit set to 0.

Figure 42.87 shows a flow chart in the case of SIMR.IICINTM bit is 1 and address transmission by CPU and data transmission by DTC or DMAC. Figure 42.88 shows a flow chart of address and data transmission by CPU. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

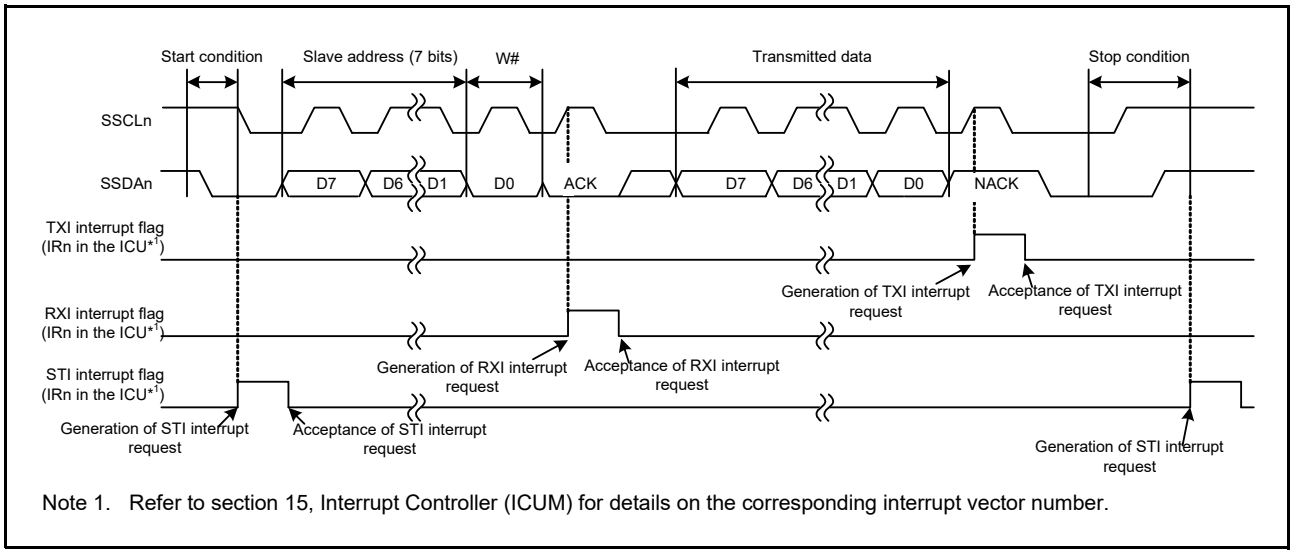
Figure 42.85 Example 1 of Operations for Master Transmission in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))

Figure 42.86 shows an example of operations when SIMR.IICINTM bit is 0 (ACK and NACK interrupt in use). In this case, DTC or DMAC is activated by the ACK interrupt, and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

Figure 42.89 shows a flow chart of SIMR.IICINTM bit is 0

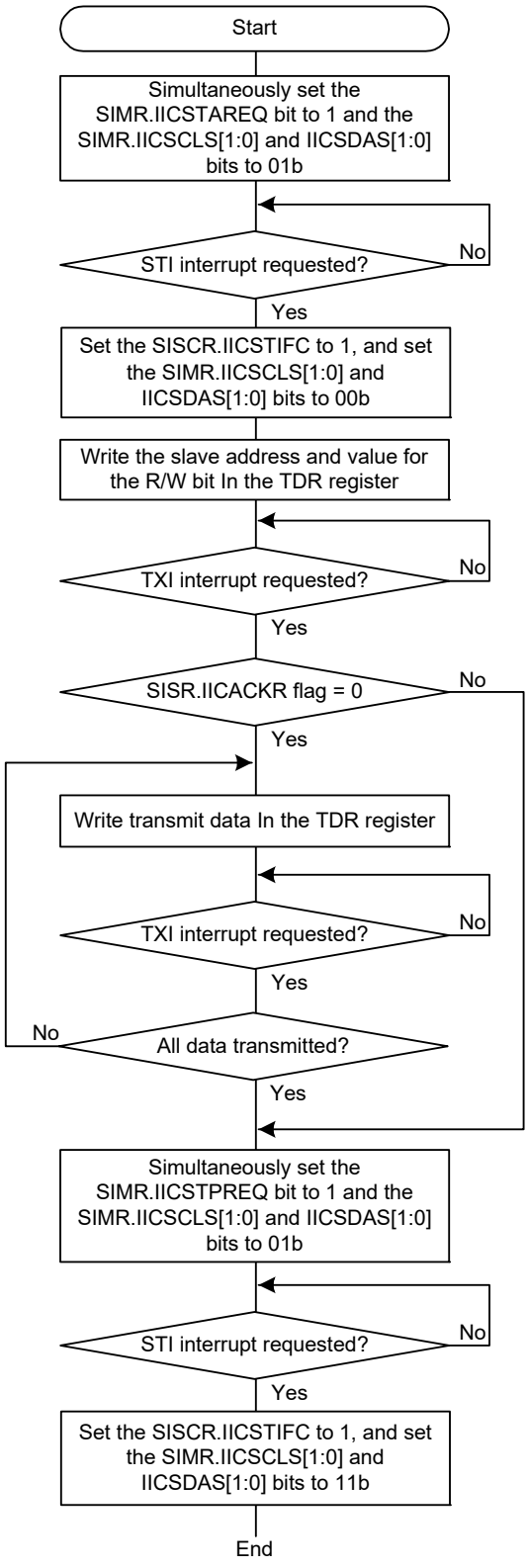
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

1. Set the SCR0.TE and SCR0.RE bits to 0 to stop communication.
2. Set SIMR.IICSCLS[1:0] and SIMR.IICSDAS[1:0] bits to 11b, release the I2C bus, and clear various condition generation requests.
3. When the SSR.RDRF flag is 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the SCR0.TE and SCR0.RE bits to 1 and restart communication.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.86 Example 2 of Operations for Master Transmission in Simple I²C Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use (SIMR.IICINTM Bit = 0))



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] [3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

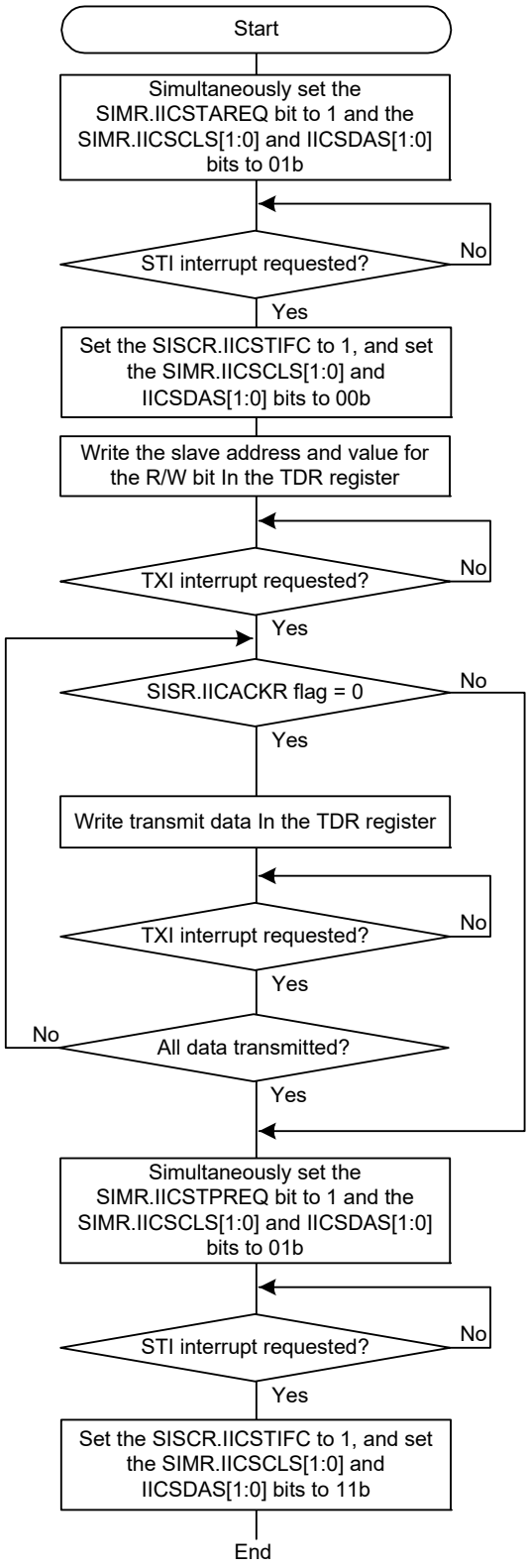
If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] [4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR register. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR register.
But it cannot be confirmed status of ACK or NACK in this case.

[5] [5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 42.87 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK by Address Transmission Only)



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. To checking whether ACK or NACK in every transmission, you cannot use DMA or DTC.

[5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 42.88 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK in All Transmissions)

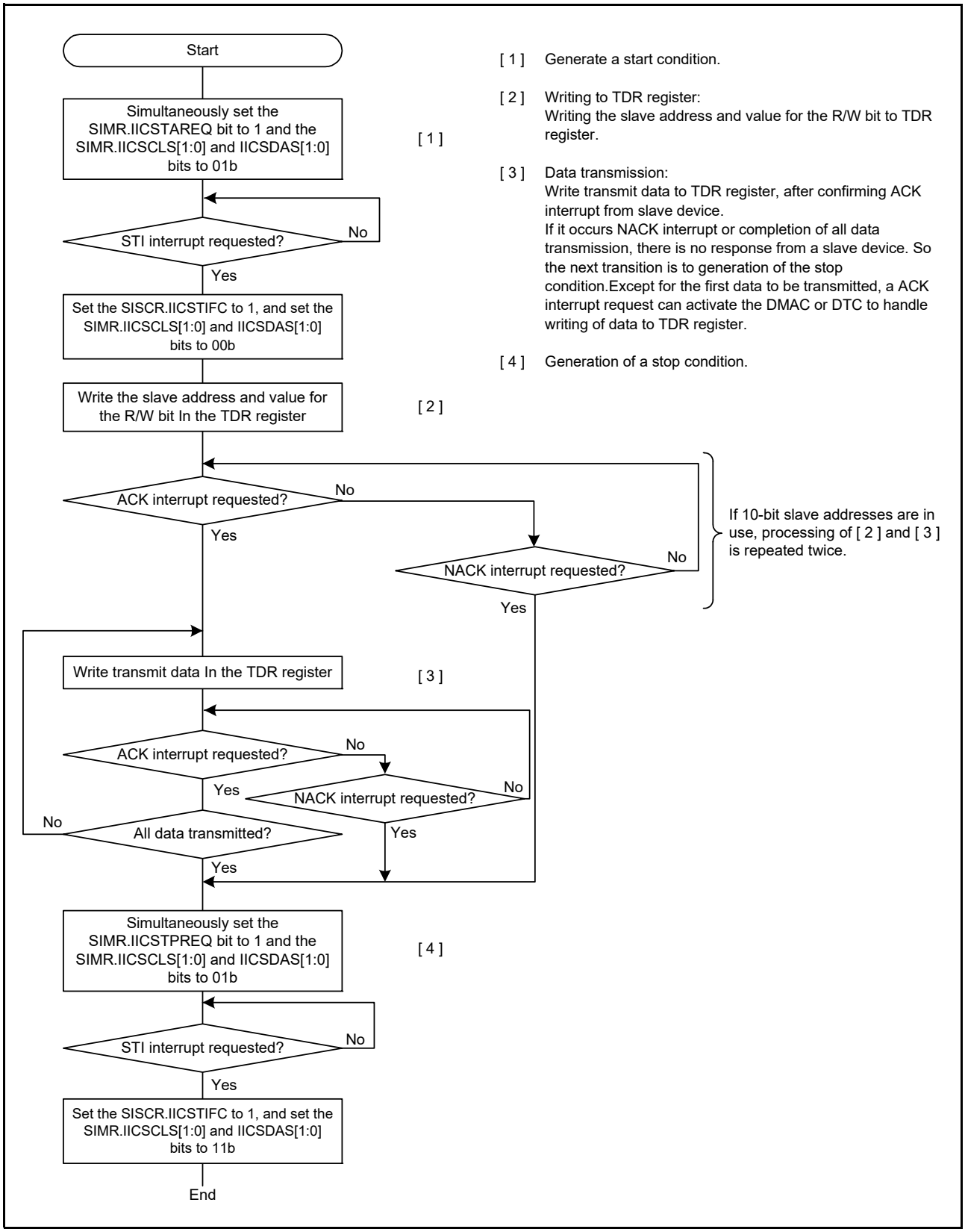
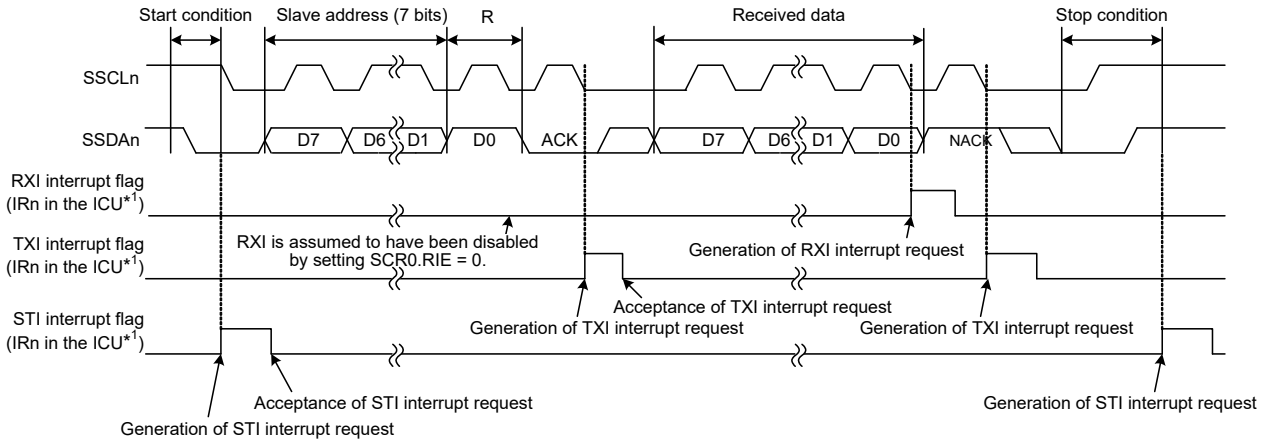


Figure 42.89 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (when SIMR.IICINTM Bit is 0)

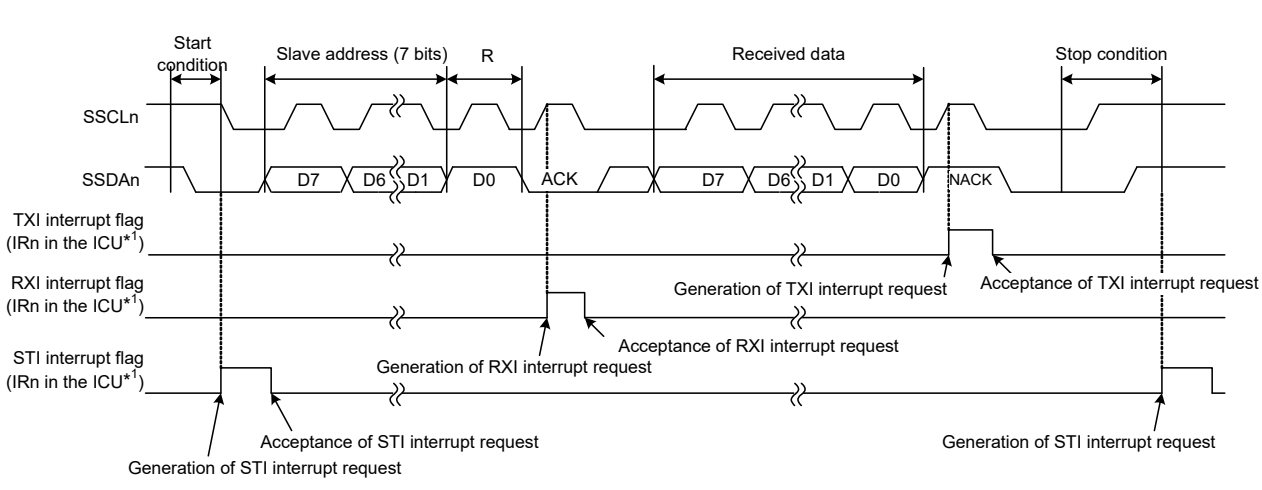
42.9.6 Master Reception (Simple I2C Mode)

Figure 42.90 and Figure 42.91 show example of operations in simple I2C mode master reception. Figure 42.92 and Figure 42.93 show flowchart of the master reception. The value of the SIMR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).
In simple I2C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.90 Example of Operations for Master Reception in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.91 Example of Operations for Master Reception in Simple I2C Mode (with 7-Bit Slave Addresses, ACK and NACK Interrupt in Use (SIMR.IICINTM Bit = 0))

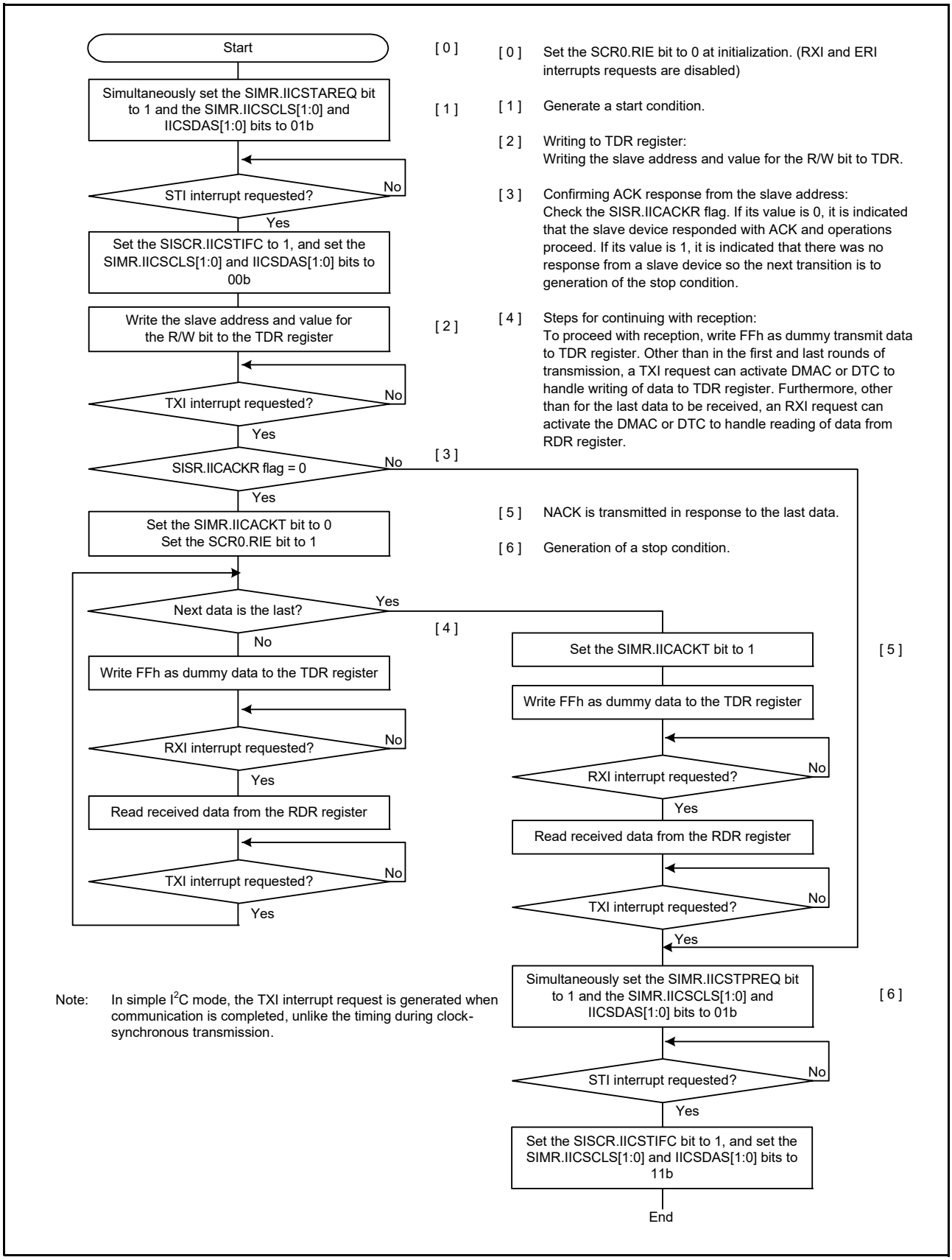


Figure 42.92 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 1, and Transmission Interrupts and Reception Interrupts in Use)

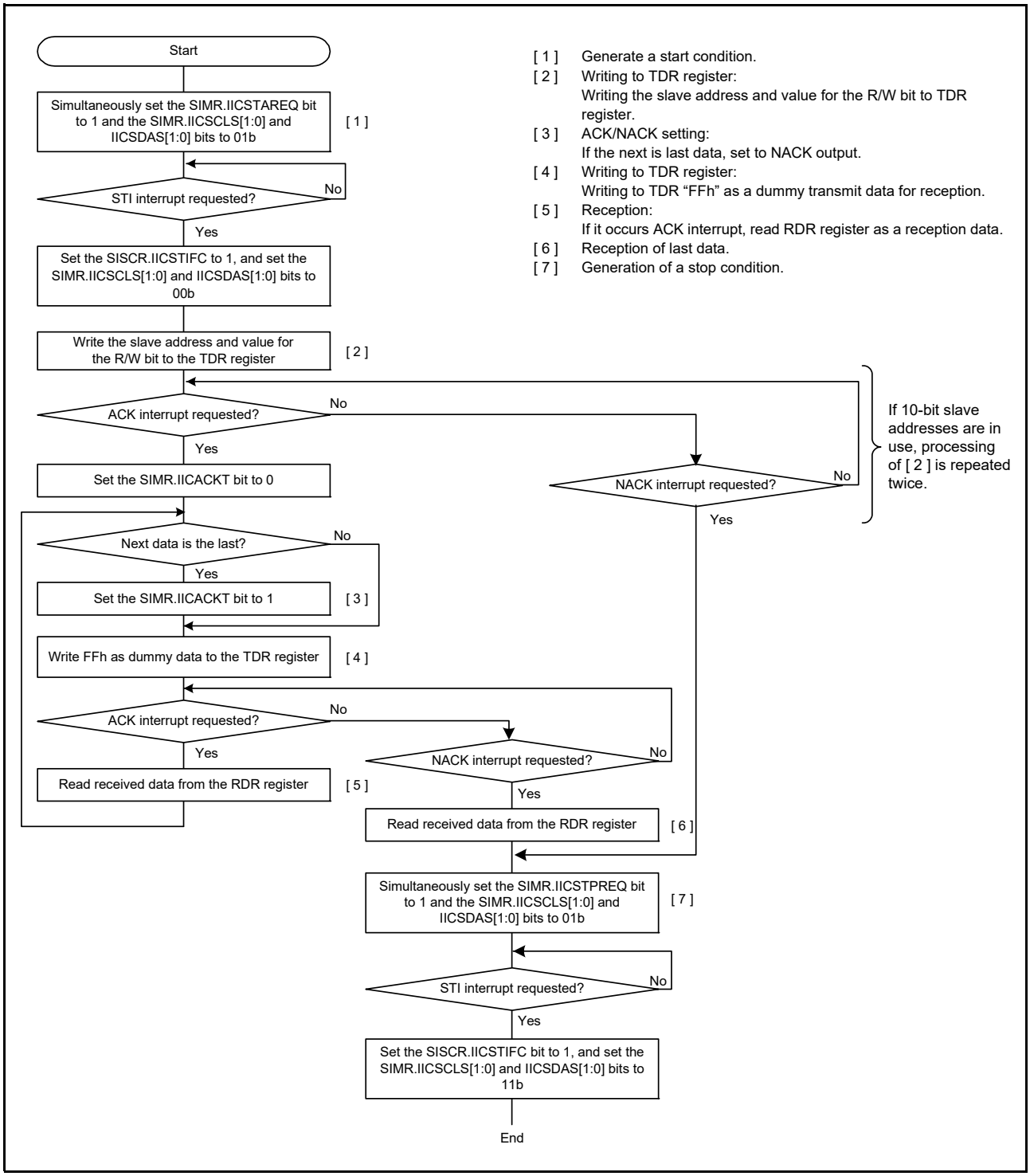
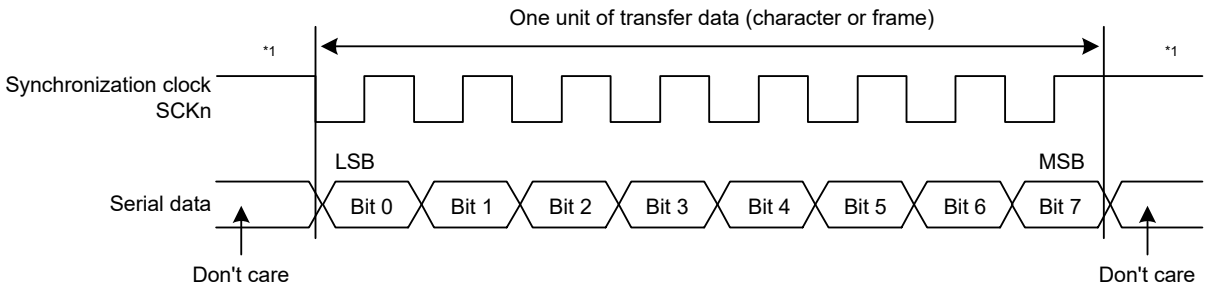


Figure 42.93 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 0, and ACK Interrupts and NACK Interrupts are in Use)

42.10 Operation in Clock Synchronous Mode

Figure 42.94 shows the communication data format of clock synchronous serial communication.
In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. A communication data character consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission when CPHA bit = 1 and CPOL bit = 1, the RSCI outputs data from the falling edge of the sync clock until the next falling edge. In data reception, data is read at the rising edges of the sync clock. After 8-bit data is output, the communication line holds the final-bit output state. In slave communication when CPHA bit = 0, however, the communication line holds the first-bit output state.
Because the RSCI has an internal transmitter and a receiver independently, RSCI enable full-duplex communication by sharing a communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver have a double- buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.



Note 1. Holds a high level except during continuous transfer or reception.

Figure 42.94 Data Format in Clock Synchronous Serial Communications (LSB First, CPHA Bit = 1, CPOL Bit = 1)

42.10.1 Clock

When using the SCKn at its highest frequency (= 1/2 TCLK), the PCLK frequency should be at least 1/2 the TCLK frequency.

(1) When the Internal Clock is Selected

When the SCR3.CKE[1:0] bits are set to 00b or 01b (master mode), the internal clock generated by the on-chip baud rate generator can be selected and the sync clock is output from the SCKn pin. Eight pulses of the sync clock are output during single-character transmission/reception. The sync clock remains at a high level*1 while no transmission or reception is performed. In transmission-only or transmission/reception, the sync clock is not output unless transmit data is prepared.
When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 1 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 1, the sync clock stops at a high level. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 0 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 0, the sync clock stops at a low level.

(2) When the External Clock is Selected

When the SCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

42.10.2 CTS and RTS Functions

The CTS function performs transmission/reception and controls transmission start using the CTSn# pin input when the internal clock is selected. Setting the SCR1.CTSE bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time.
When the CTS function is enabled, transmission/reception and transmission start only when the CTSn# pin input level is low.
When using the FIFO, if the CTSn# signal remains high before transmission, transmission will not start, but the number of data stored will be “number written to the TDR register – 1” (unlike using FIFO in asynchronous mode). This is because data is transferred to the TSR register after writing to the TDR register, but if the CTSn# signal is set to low level, transmission starts from the TSR register, so there is no problem.
Even if the CTSn# pin input becomes high level during transmission/reception or transmission operation, frames that are being transmitted/received or being transmitted are not affected and transmission/reception or transmission operation continues.
The RTS function makes a serial communication start request using the RTSn# pin output when the external sync clock is selected. When serial communication is enabled, the RTSn# pin outputs a low level. A low level and a high level are output under the following conditions.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- No receive data are present before reading and reception is not in progress. (when SCR0.RE bit = 1)
- Data written in the TDR register is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.
When SCR0.RE bit is set to 0 without reading the RDR register to terminate reception after reception is complete, the RTSn# pin output level remains high. At this time, write 0 to SCR0.RE bit.

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

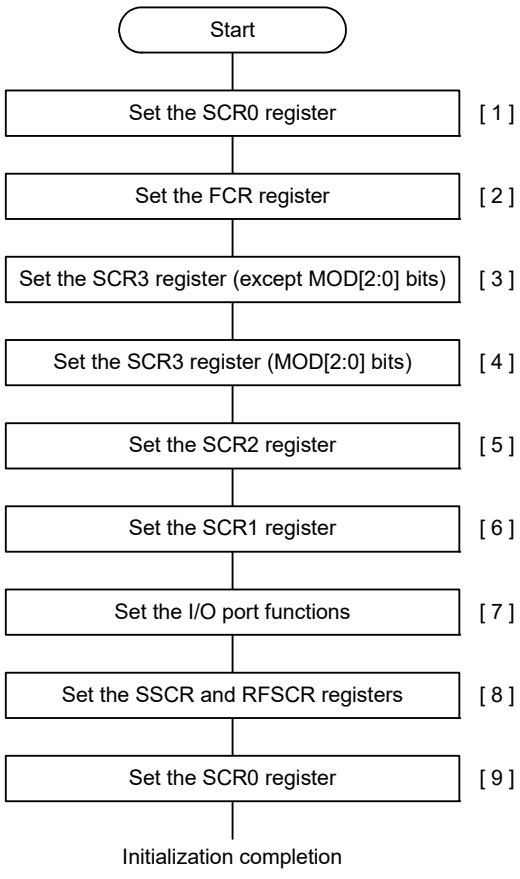
[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when SCR0.RE bit = 1)
- Data written in the transmit FIFO (TDR register) is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

42.10.3 RSCI Initialization (Clock Synchronous Mode)

Before starting data transmission/reception, write 0 to SCR0.TE bit and SCR0.RE bit (or write initial values to the SCR0 register) and initialize the RSCI according to the flowchart example in Figure 42.95.
Before changing operating mode or communication format, also be sure to write 0 to TE bit and RE bit.
Note that writing 0 to the RE bit does not initialize the ORER, AFER, APER, and RDRF flags in SSR register and the RDR register. Also note that writing 0 to the TE bit does not initialize the SSR.TEND flag in FIFO mode. Attention is also needed for changing operating mode.
When the SCR0.TIE bit = 1, note that setting the TE bit to 1 from 0 generates a TXI interrupt.



[1] Set the following bits of SCR0 register to 0[*1].
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the TFRST and RFRST bits to 1 to empty FIFO.
Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

[3] Set SCR3 except of communication mode.
FIFO use/no-use
Transmission/reception format
Clock setting
Leave unused bits (CHR[1:0], STOP, RXDESEL, MP, DEEN, GM, BLK) at their initial values.[*2]

[4] Set communication mode (MOD[2:0] bits = 010b)[*2].

[5] Select clock, set bit rate[*3]. Leave unused bits (SCR2.BCP[2:0], ABCS, ABCSE, ABCSE2, BRME, and MDDR[7:0]) at their initial values.

[6] Set up the half-duplex communication, the loop-back function, communication pin status, and the CTS/RTS function.

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Write 1 to the following bits to clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
RFSCR.DRC

[9] Set the TE or RE bit to 1.[*1, *4] To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in SCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set the clock (CPOL and CPHA bits) before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

Figure 42.95 Example of RSCI Initialization Flowchart (Clock Synchronous Mode)

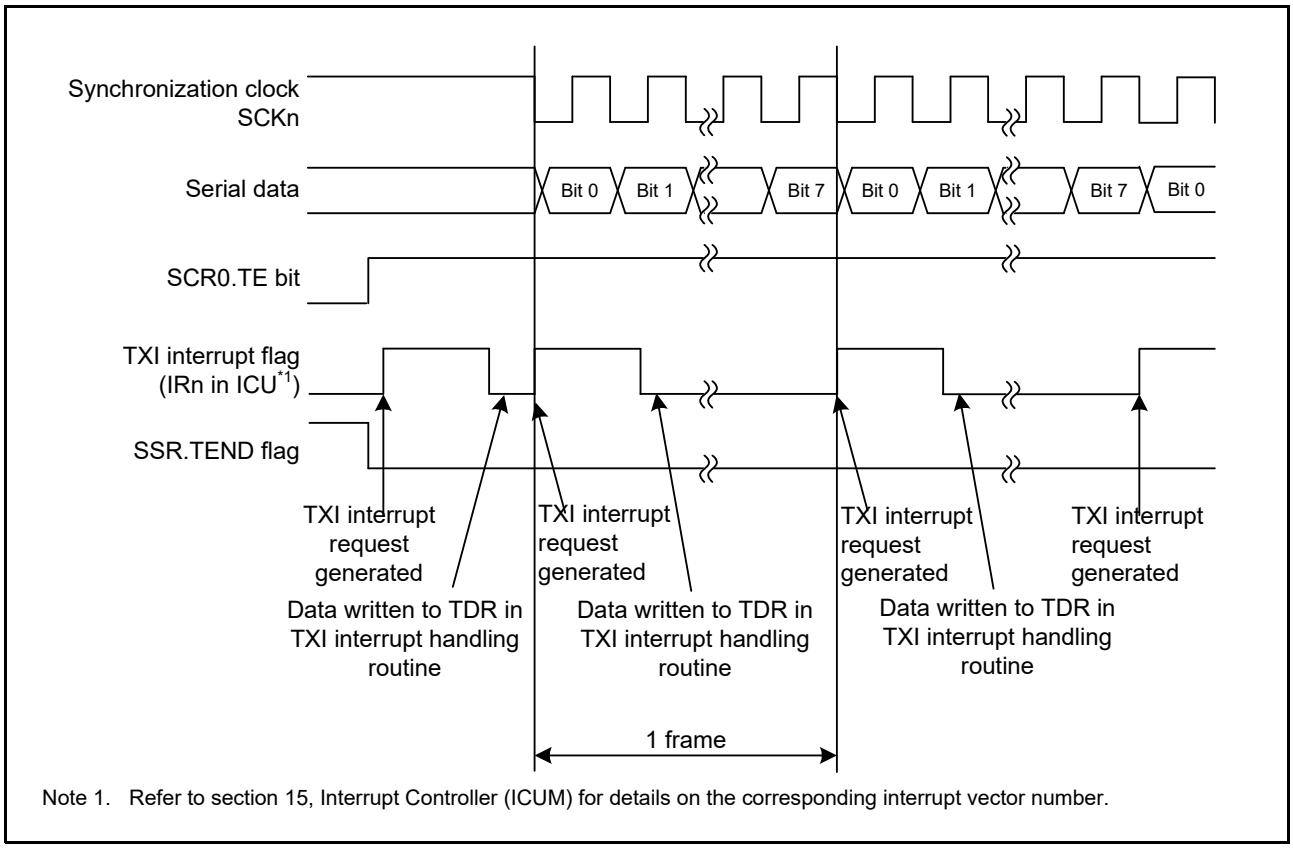
42.10.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO Mode

Figure 42.96 to Figure 42.98 show an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the RSCI operates as described below.

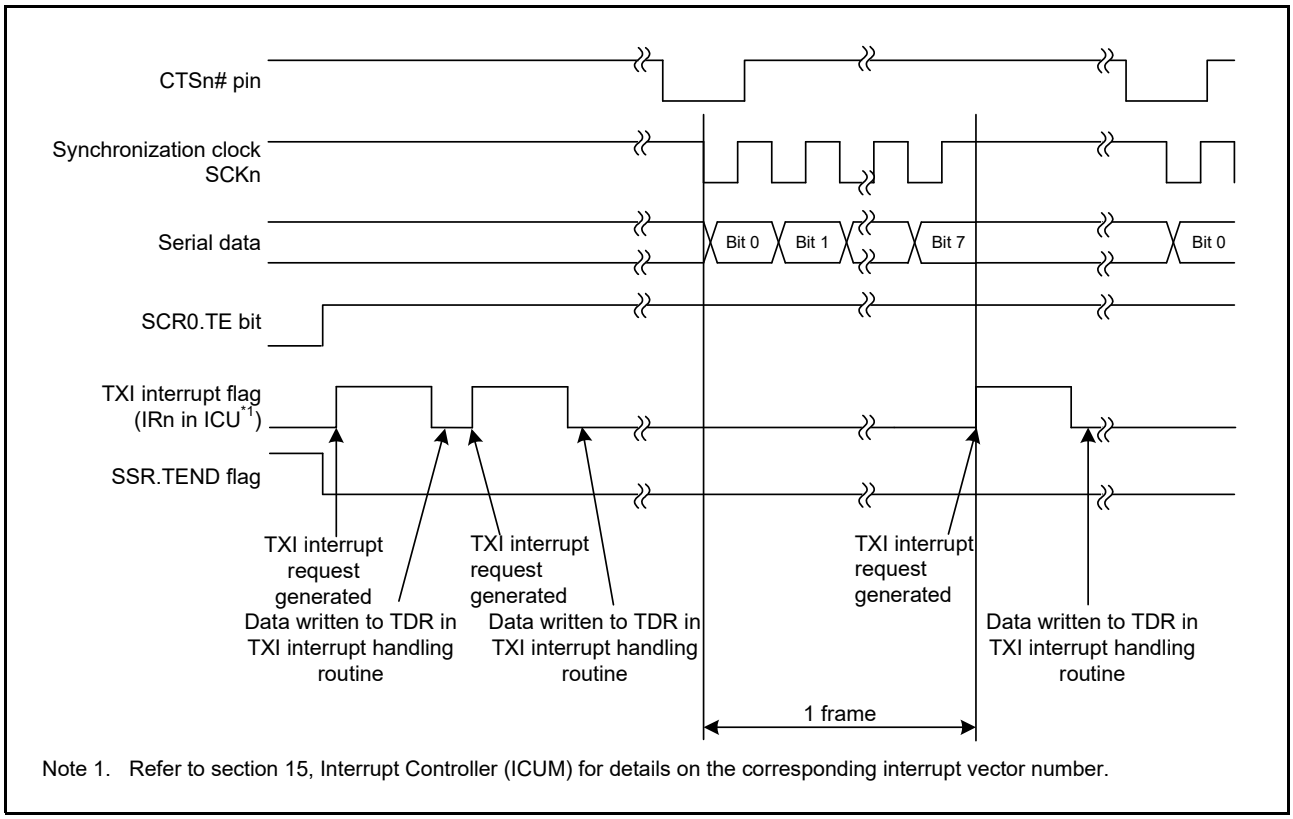
1. When data is written to the TDR register in the TXI interrupt routine, the RSCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the SCR0.TIE bit and the SCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. The written data is transferred from the TDR register to the TSR register, which starts transmission. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the TDR register before transmission of data transferred previously in the TXI interrupt routine is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the TDR register in the TXI interrupt request processing routine and the final data's transmission is started, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. Update (data write) of the TDR register is checked at the final-bit transmission timing.
5. When the TDR register has been updated, data is transferred from the TDR register to the TSR register to start sending the next frame.
6. If the TDR register has not been updated, the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 42.99 shows a sample flowchart of serial data transmission.



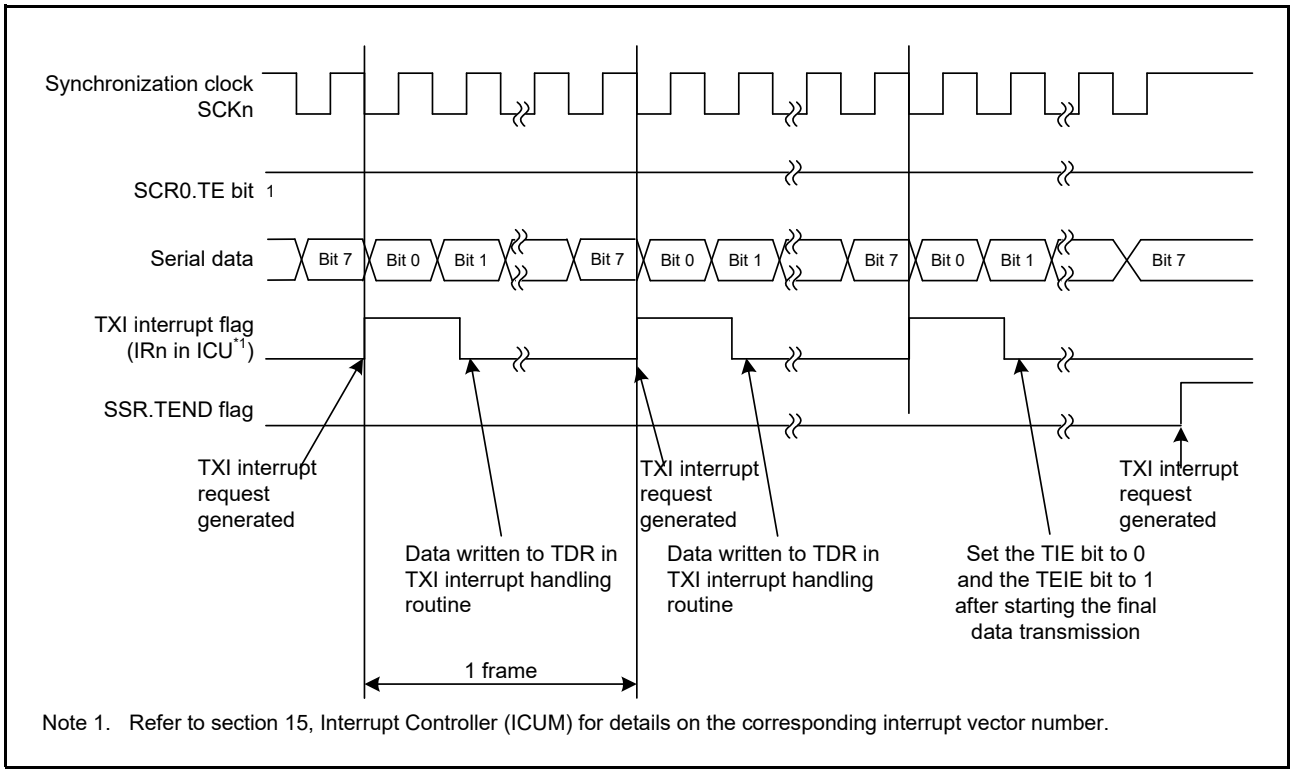
Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.96 Serial Transmission Example in Clock Synchronous Mode (1) (CTS Function Not Used/ Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



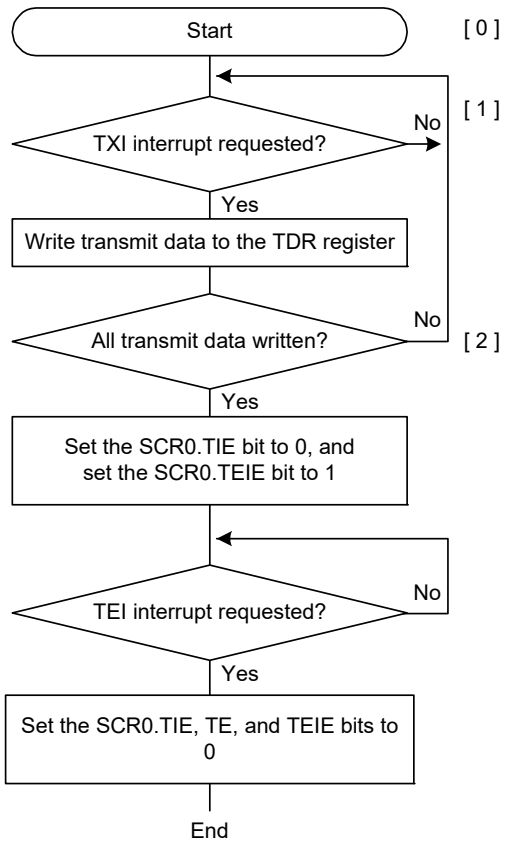
Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.97 Serial Transmission Example in Clock Synchronous Mode (2) (CTS Function Used/Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.98 Serial Transmission Example in Clock Synchronous Mode (3) (During Transmission to Transmission End/CPHA Bit = 1, CPOL Bit = 1)



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to TDR register by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from TDR register. Write transmit data to TDR register once in the TXI interrupt handling routine.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR register in the TXI interrupt handling routine. Transmit data can also be written by activating the DMAC or DTC.
If you use TEI interrupt request, set SCR0.TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the SCR3.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

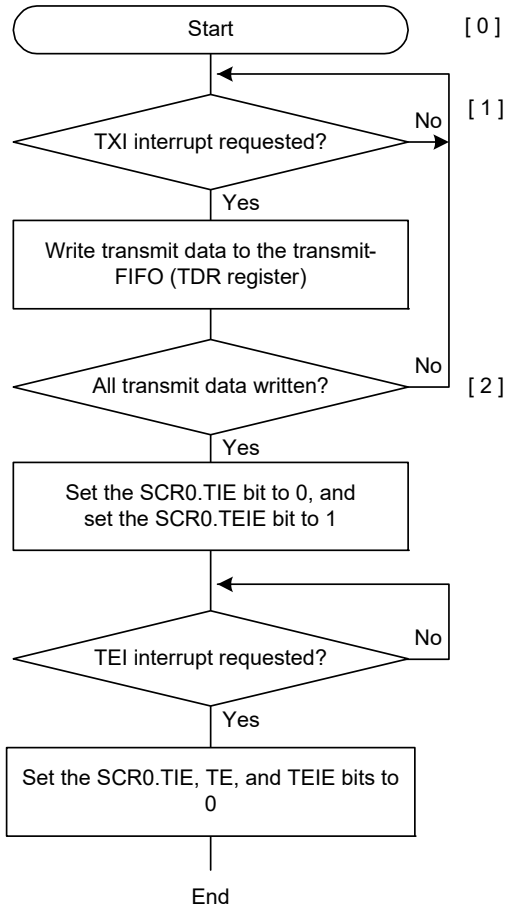
Figure 42.99 Example Flowchart of Serial Transmission in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 42.100 shows an example of flowchart of serial transmission (a FIFO buffer selected) in clock synchronous mode with FIFO enabled.
The RSCI operates as follows when serial data transmission.

1. When data is written to the transmit FIFO (TDR register) in the TXI interrupt routine, the RSCI transfers the data from the transmit FIFO (TDR register) to the TSR register. The number of writable transmit data is [16 – number of unsent transmit data stored in the transmit FIFO (TDR register)]. If the SCR0.TIE and SCR0.TE bits are simultaneously set to 1 at the start of data transmission, a TXI interrupt request is generated.
2. Data is transferred from the transmit FIFO (TDR register) to the TSR register and transmission starts. When the number of data stored in the transmit FIFO (TDR register) is equal to or less than the threshold value of the transmit FIFO, the SSR.TDRE flag is set to 1. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the transmit FIFO (TDR register) in the TXI interrupt routine before transmission of data written to the transmit FIFO (TDR register) is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the transmit FIFO (TDR register) in the TXI interrupt request processing routine, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTSn# pin input becomes low level.
4. The RSCI checks whether unsent transmit data is remaining in the transmit FIFO (TDR register)*1 at the final-bit transmission timing.
5. When data is remaining in the transmit FIFO (TDR register), the data is transferred from the transmit FIFO (TDR register) to the TSR register to start sending the next frame.
6. If no data is remaining in the transmit FIFO (TDR register), the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the TFSR.T[5:0] bits.



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to transmit FIFO (TDR) by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the transmit FIFO (TDR register) by using a TXI interrupt and clear the SSR.TDRE flag to 0.
The number of transmission data it's possible to write in is 16 minus (the number of stored data in transmit FIFO).
Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by TXI interrupt request. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.
If you use TEI, set TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

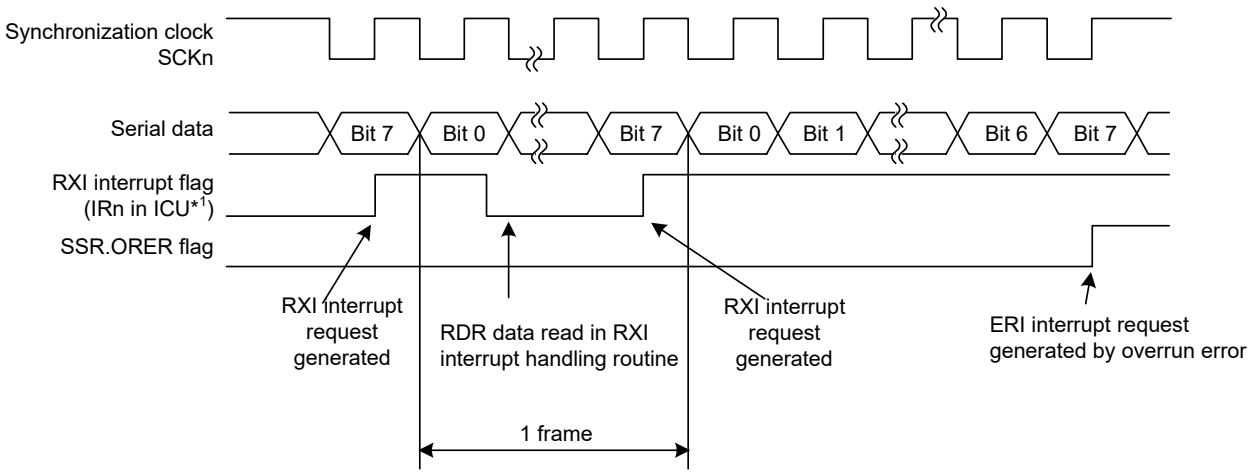
Figure 42.100 Example Flowchart of Serial Transmission in Clock Synchronous Mode (FIFO Mode)

42.10.5 Serial Data Reception (Clock Synchronous Mode)

(1) Non-FIFO Mode

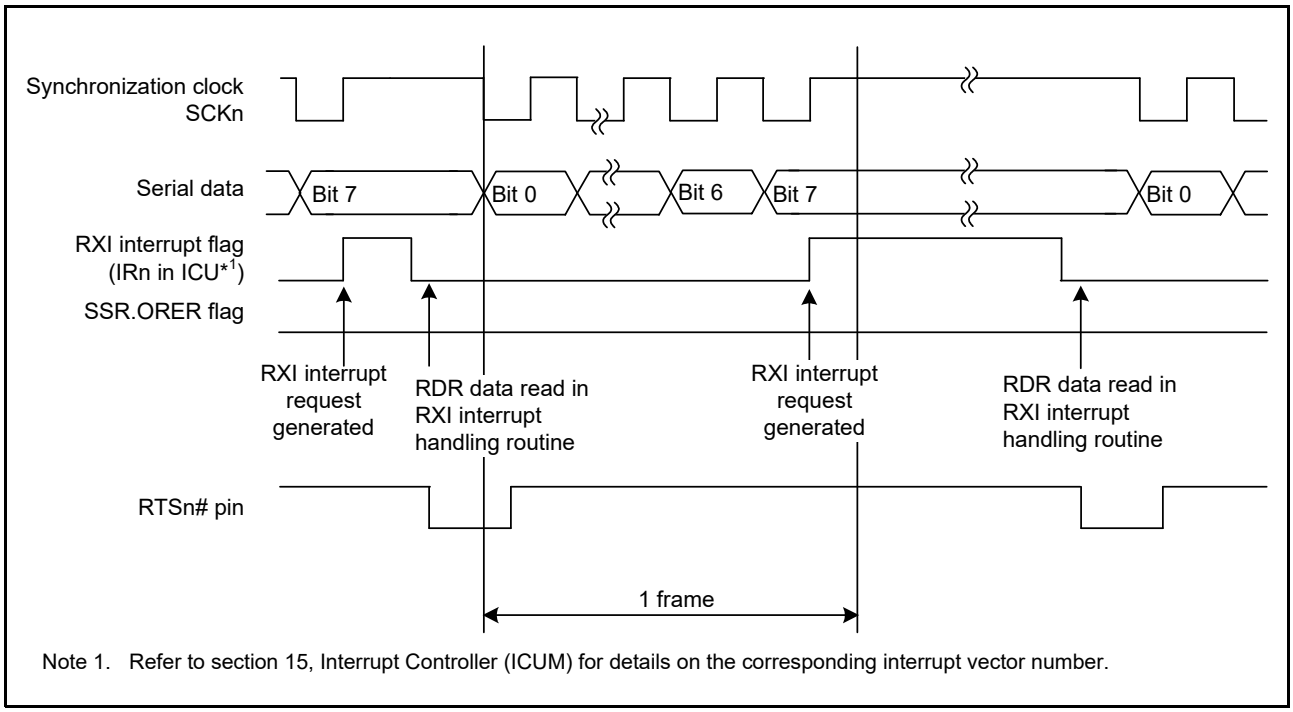
Figure 42.101 and Figure 42.102 show operation examples of serial data reception in clock synchronous mode.
The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output becomes low (when the RTS function is used).
2. The RSCI starts data reception in synchronization with input or output of the sync clock, and transfers receive data to the RSR register.
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the RDR register.
4. When data is normally received, the received data is transferred to the RDR register. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the RDR register in the RXI interrupt handling routine before the next data is completely received enables continuous reception. When the received data transferred to the RDR register is read, the RTSn# pin output becomes low (when the RTS function is used).
   If you want to prevent the RTSn# pin output from turning low after the final data is received, clear the SCR0.RE bit to 0 and then read the RDR register.



Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.101 Example of Operation for Serial Reception in Clock Synchronous Mode (1)
(When RTS Function is Not Used/CPHA Bit = 1, CPOL Bit = 1)

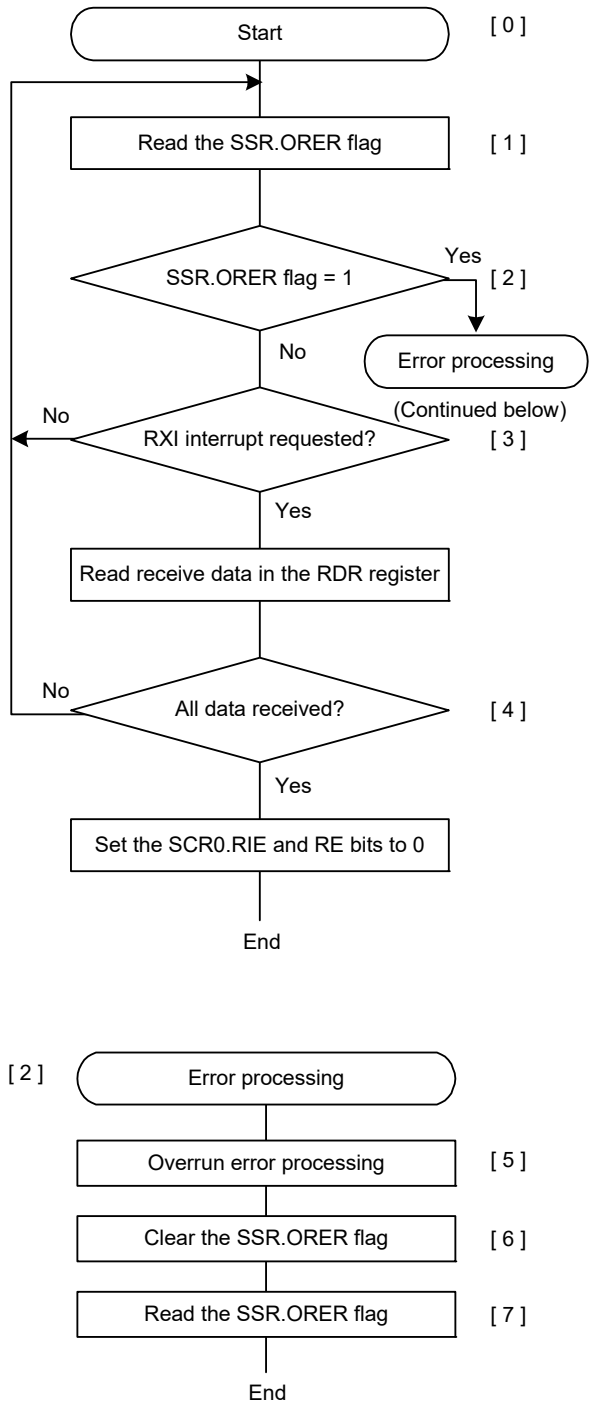


Note 1. Refer to section 15, Interrupt Controller (ICUM) for details on the corresponding interrupt vector number.

Figure 42.102 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used/CPHA Bit = 1, CPOL Bit = 1)

While the reception error flag is set to 1, subsequent reception are disabled. Therefore, before continuing reception, be sure to clear the ORER, AFER, and APER flag in SSR to 0. Also be sure to read the RDR register in the overrun error processing. If the SCR0.RE bit is set to 0 during reception to forcibly terminate the reception operation, unread receive data may be remaining in the RDR register. In this case, read the RDR register.

Figure 42.103 shows a sample flowchart for serial data reception.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

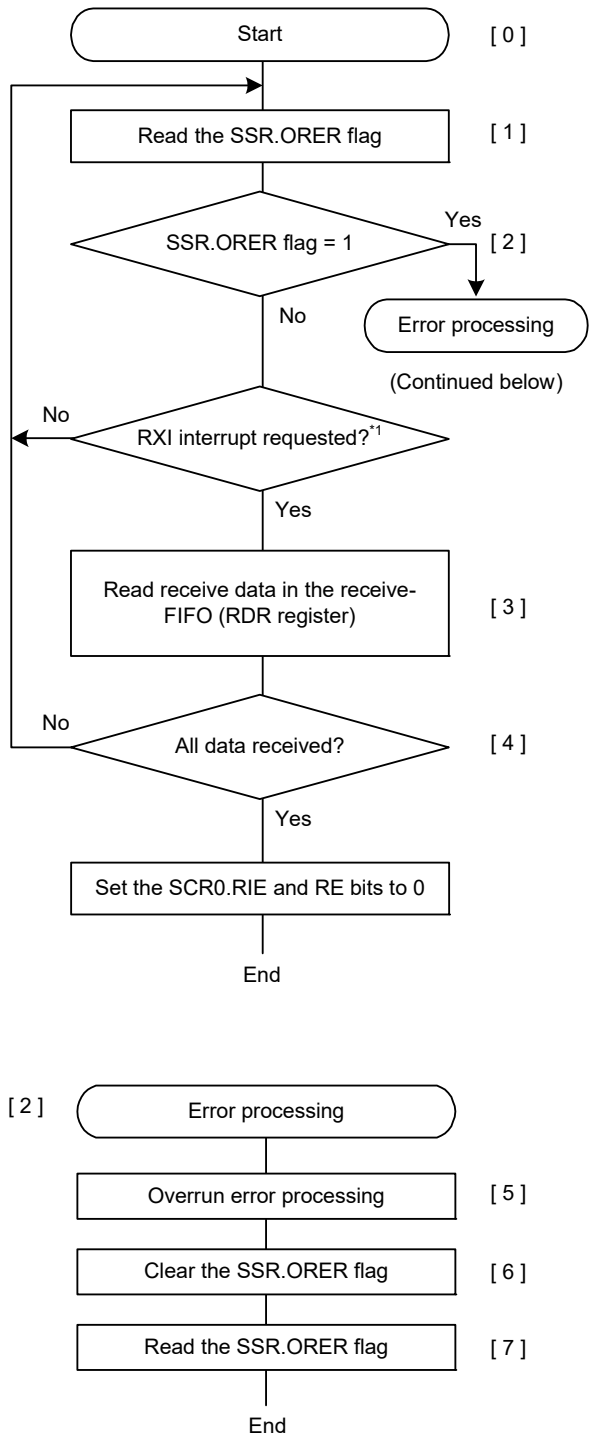
Figure 42.103 Example Flowchart of Serial Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 42.104 shows an example of serial data reception flowchart in clock synchronous mode with FIFO enabled. The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output turns low (when the RTS function is used).
2. The RSCI starts receiving data in synchronization with input or output of the sync clock, and transfers the received data to the receive FIFO (RDR register).
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the receive FIFO (RDR register)*1.
4. When data is normally received, the received data is transferred to the receive FIFO (RDR register)*1. When the number of receive data stored in the receive FIFO (RDR register) is equal to or more than the threshold value of the receive FIFO, the SSR.RDRF flag is set to 1. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the receive FIFO (RDR register) in the RXI interrupt handling routine before an overrun error occurs enables continuous reception. When the received data transferred to the receive FIFO (RDR register) is read and the number of data becomes lower than the RTS# output threshold value, the RTSn# pin output becomes low (when the RTS function is used).

Note 1. The RDR.RDAT[8] bit is not used.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG[4:0] value at least. When the RDR register is read, the SSR.RDRF flag is cleared. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR register. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 42.104 Example Flowchart of Serial Reception in Clock Synchronous Mode (FIFO Mode)

42.10.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

In clock synchronous mode, transmission and reception are simultaneously performed, so the number of transmitted data and the number of received data are the same.

(1) Non-FIFO Mode

Figure 42.105 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode.
After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.
When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the RTSn# pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in SCR0 register to 0 simultaneously, and then read the RDR register.

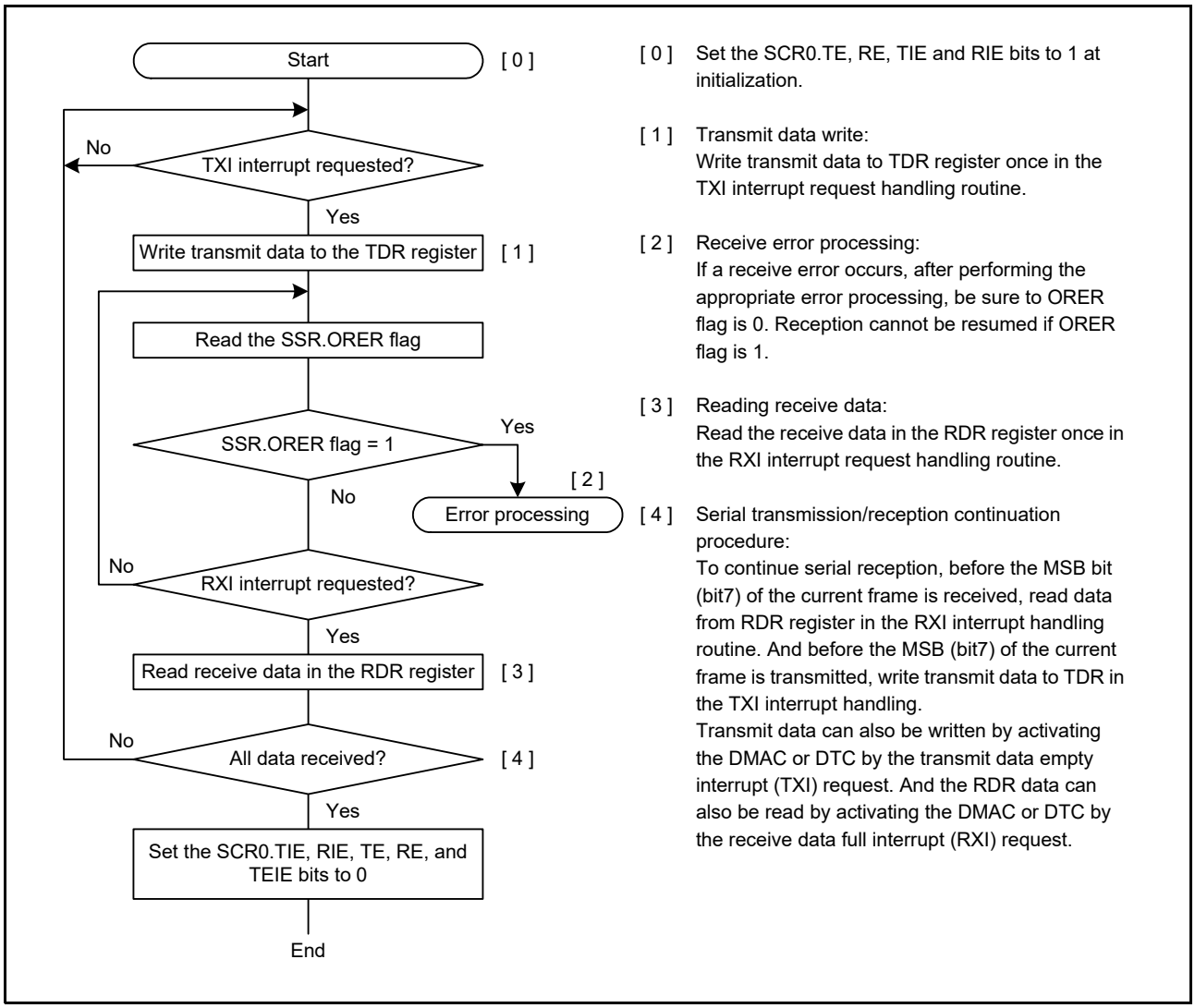


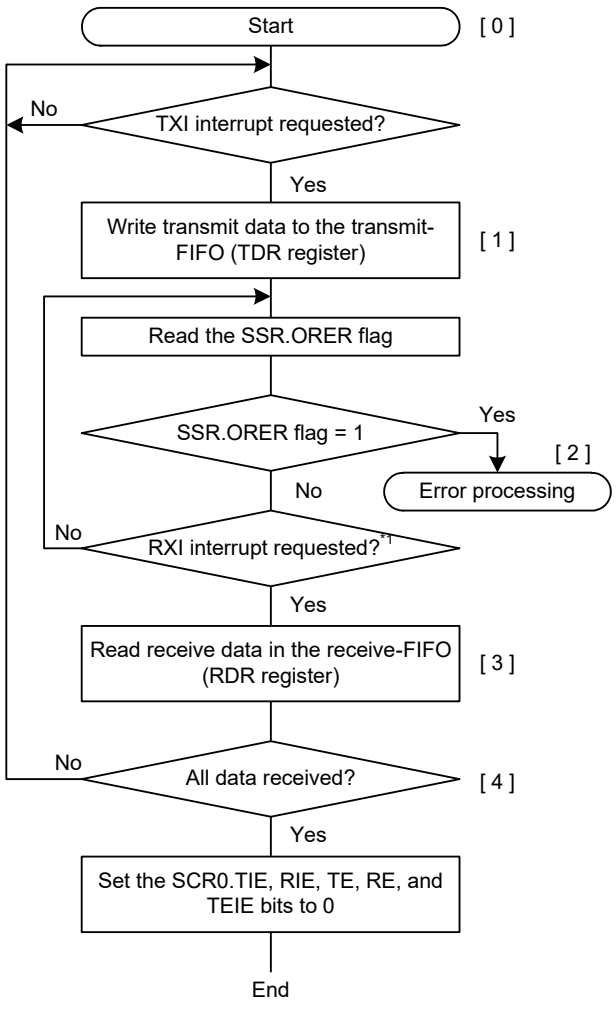
Figure 42.105 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 42.106 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode with FIFO enabled.

After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 to 1 simultaneously by a single instruction.



Note 1. Set the number of all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[0] Set the SCR0.TE, RE, TIE and RIE bits to 1 at initialization.

[1] Transmit data write:
Write transmit data to transmit FIFO (TDR register) in the TXI Interrupt request handling routine. The number of data to be written should match the number specified in the receive FIFO threshold.

[2] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the appropriate error processing such as stopping communication and reading all received data to clear the SSR.RDRF flag, and then set the ORER flag to 0. Reception cannot be resumed if ORER flag is 1.

[3] Reading receive data:
Read the same number of received data stored in the receive FIFO (RDR register) as the number specified in the receive FIFO threshold.

[4] Serial transmission/reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from the receive FIFO (RDR register) and clear SSR.RDRF flag. To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0. Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by the transmit data empty interrupt (TXI) request. And The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request. In this case, the TDRE and RDRF flags are cleared automatically, so please don't operate the TDRE and RDRF flags.

Figure 42.106 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (FIFO Mode)

42.11 Operation in Simple SPI Mode

As an extended function, the RSCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for simple SPI mode (SCR3.MOD[2:0] bits = 011b) plus setting the SSE bit in the SCR0 register to 1 places the RSCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SSE bit in the SCR0 to 0 in such cases.

Figure 42.107 shows an example of connections for simple SPI mode.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this.

Since the receiver and transmitter are independent of each other within the RSCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.

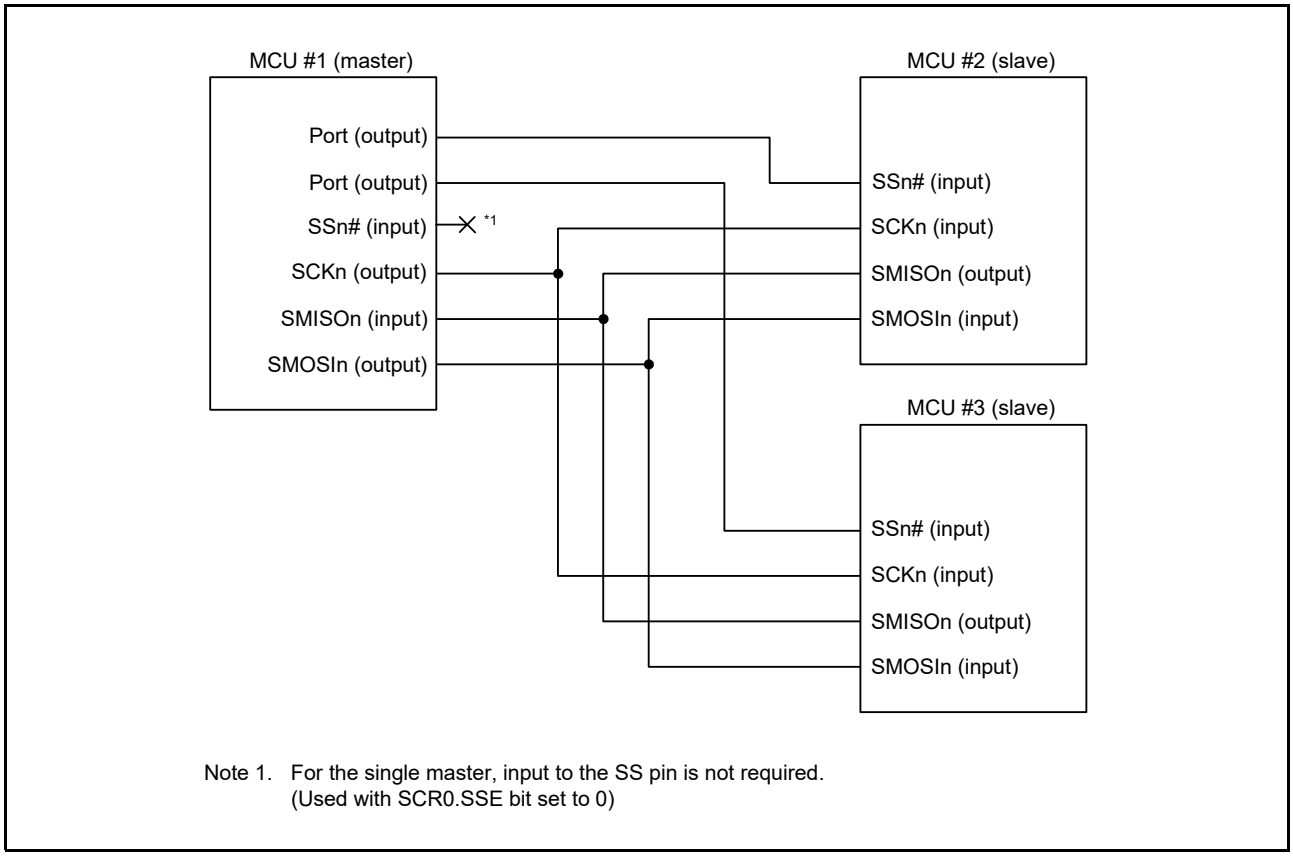


Figure 42.107 Example of Connections via a Simple SPI Mode

42.11.1 Clock

When using the SCKn at its highest frequency (= 1/2 TCLK), the PCLK frequency should be at least 1/2 the TCLK frequency.

42.11.2 States of Pins in Master and Slave Modes

In simple SPI mode, input and output directions of each pin vary depending on master mode (SCR3.CKE[1:0] bits = 00b or 01b) and slave mode (SCR3.CKE[1:0] bits = 10b or 11b).
Table 42.41 lists the states of pins according to the mode and the level on the SSn# pin.

Table 42.41 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission*2	Clock input

Note 1. When there is only a single master (SCR0.SSE bit = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMISOn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0).
Note 3. The SCKn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0 and RE bit = 0) in a multi-master configuration (SCR0.SSE bit = 1).

42.11.3 SS Function in Master Mode

Setting the SCR3.CKE[1:0] bits to 00b or 01b enables master mode.
In single master mode (SSE bit = 0), the SSn# pin is not used and data transmission and reception are enabled regardless of the SSn# pin input level. The SSn# pin is available for other purposes.
When in multi-master mode (SSE bit = 1) and the SSn# pin input is high, the master outputs a clock from the SCKn pin and performs transmission and reception operations. And outputting clock indicates “There are no other masters” or “Another master is not performing reception or transmission”. When the SSn# pin input level is low in multi-master mode (SSE bit = 1), this means that another master exists and it is performing data transmission/reception. At this time, the RSCI makes the TXDn pin output and the SCKn pin output high impedance and does not start data transmission/reception. In addition, the SSR.MFF flag is set to 1 as a mode fault. In multi-master mode, read this flag bit to perform the error processing. If a mode fault occurs during the transmission/reception operation, the SCKn pin and the TXDn pin output are made high impedance while the SSn# pin input level is low. In this case, any of TXI, RXI, and TEI interrupts occurs.
Control the SS signal output in master mode with a general-purpose port.

42.11.4 SS Function in Slave Mode

Setting the SCR3.CKE[1:0] bits to 10b or 11b enables slave mode.
When the SSn# pin input level is high, the RXDn pin output becomes high impedance and the clock input from the SCKn pin is ignored. When the SSn# pin input level is low, the clock input from the SCKn pin becomes effective, enabling data transmission and reception.
When the SSn# pin input changes from low to high level during the transmission/reception operation, the RXDn pin output is made high impedance and the transmission/reception operation is immediately suspended. If the transmission is in progress, the SSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn# pin during slave transmission/reception. If an abnormal stop occurs, set the SCR0.RE and SCR0.TE bits to 0 to stop transmission/reception. To resume transmission/reception, set the SCR0.RE and SCR0.TE bits to 1 after at least 3 TCLK cycles plus 3 PCLK cycles.

42.11.5 Relationship between Clock and Transmit/Receive Data

The clock to be used for data transmission/reception is selectable from four types using the SCR3.CPOL and CPHA bits. Figure 42.108 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

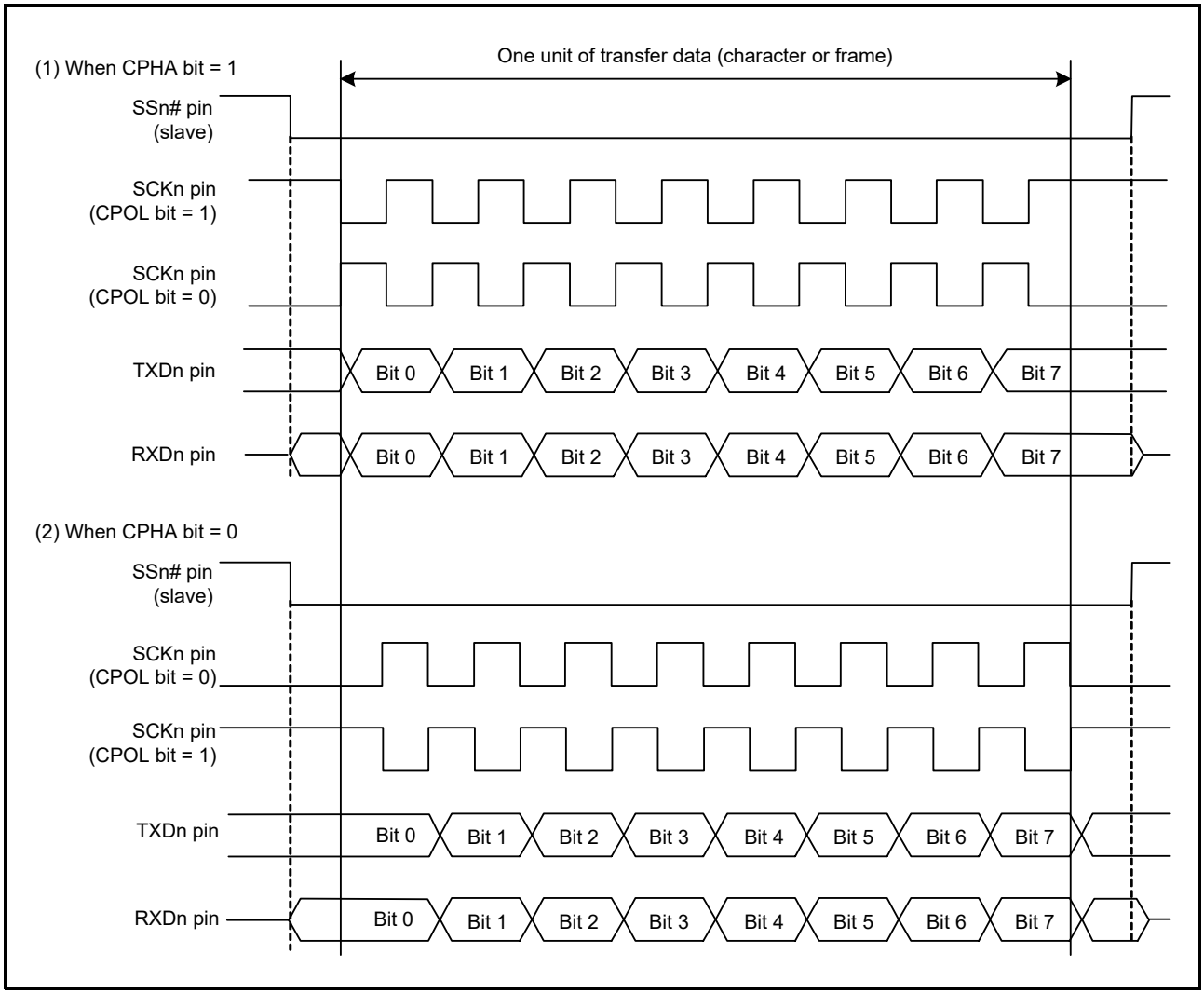


Figure 42.108 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

42.11.6 RSCI Initialization (Simple SPI Mode)

The RSCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 42.95, Example of RSCI Initialization Flowchart (Clock Synchronous Mode)). The master devices and slave devices use the same clock type selected by the SCR3.CPOL and CPHA bits.
Before performing initialization or changing operating mode or communication format, be sure to stop communication (SCR0.RE bit = 0 and SCR0.TE bit = 0).
Note that setting the RE bit to 0 does not initialize the ORER, AFER, and APER flags in SSR register and the RDR register.
Note that, when SCR0.TIE bit = 1, setting the TE bit to 1 from 0 generates a TXI interrupt.

42.11.7 Transmission and Reception of Serial Data (Simple SPI Mode)

In master mode, set the SSn# pin of the destination slave device to low level before starting data transmission/reception and set to high level after the end of data transmission/reception. In multiple master operation with SCR0.SSE bit = 1 even in master mode, a mode fault will occur if the SSn# pin goes low. Therefore, make sure that no mode fault has occurred before starting communication, and start communication, and make sure that no mode fault has occurred even after communication ends. If a mode fault has occurred, communication may be incomplete, so measures such as retransmission are required. The other procedures are the same as in clock synchronous mode.
In slave mode, it operates according to the SSn# pin input level. Other steps are the same as those of clock synchronous mode.

42.12 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be corrected by evenly enabling the clocks of the number specified in the SCR2.MDDR[7:0] bits among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in SCR2 register.

Figure 42.109 shows an example where the TCLK is selected by the CKS[1:0] bits in SCR2 register and the BRR[7:0] bits and MDDR[7:0] bits are set to 0 and 160 respectively in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). Note that there is an imbalance in enabling the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in clock synchronous mode, simple-SPI mode, smart card Interface mode, Manchester mode and extended serial mode.

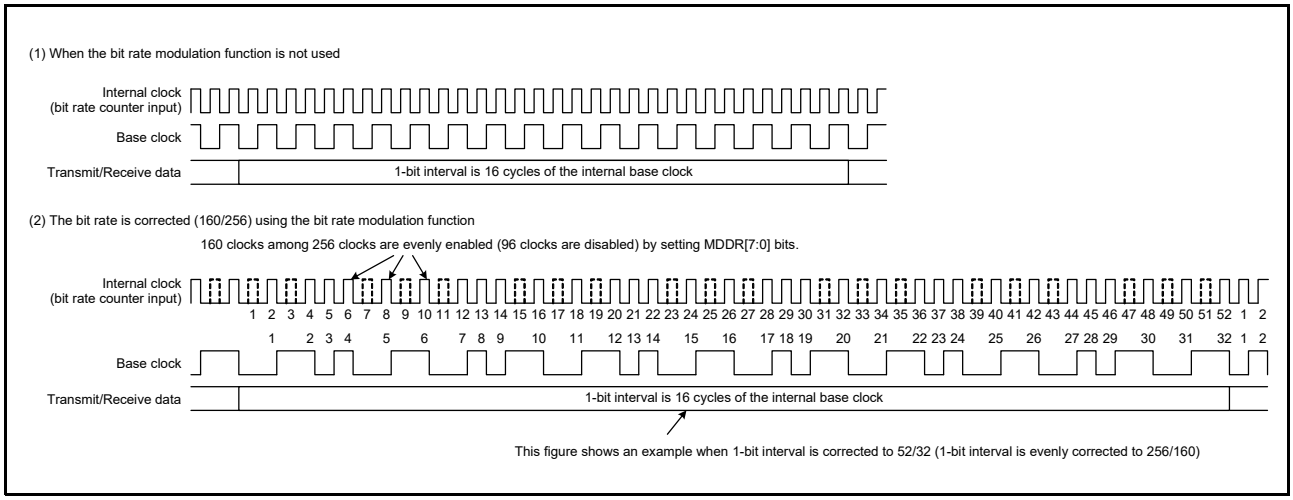


Figure 42.109 Example of Base Clock when Using the Bit Rate Modulation Function

42.13 Noise Cancellation Function

Figure 42.110 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits, a match detection circuit, and a majority voting circuit. When the SCR1.NFM bit is set to 0, when the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is propagated as an internal signal. Unless otherwise matched, the previous value is retained. (When the same level is retained for three cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for three cycles or shorter is considered as a noise, not as a receive signal). When the SCR2.ABCSE2 bit is set to 1 in asynchronous mode, the above method cannot be used because there are only four base clock cycles in one bit period. In this case, set the NFM bit to 1 to select majority mode.

In asynchronous mode, Manchester mode and extended serial mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period (1/16 of one bit period when SCR2.ABCS bit = 0, SCR2.ABCSE bit = 0, and SCR2.ABCSE2 bit = 0. 1/8 of one bit period when ABCS bit = 1, ABCSE bit = 0, and ABCSE2 bit = 0. 1/6 of one bit period when ABCSE2 bit = 0 and ABCSE bit = 1. 1/4 of one bit period when ABCSE2 bit = 1 and ABCSE bit = 0.) or the baud rate generator source clock divided by 1, 2, 4, or 8 by setting the SCR1.NFCS[2:0] bits. When setting the ABCSE2 or ABCSE bit to 1, set the NFCS[2:0] bits to 000b or 001b.

In simple I2C mode, the noise elimination function can be used for the input pins of TXDn/SSDAn and RXDn/SSCLn. The sampling period of the noise filter can be selected from the divided clock of the baud rate generator clock source by SCR1.NFCS[2:0] bits.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR0.TE and SCR0.RE bits are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is propagated as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles (NFM bit = 0) or two or more sampling cycles (NFM bit = 1).

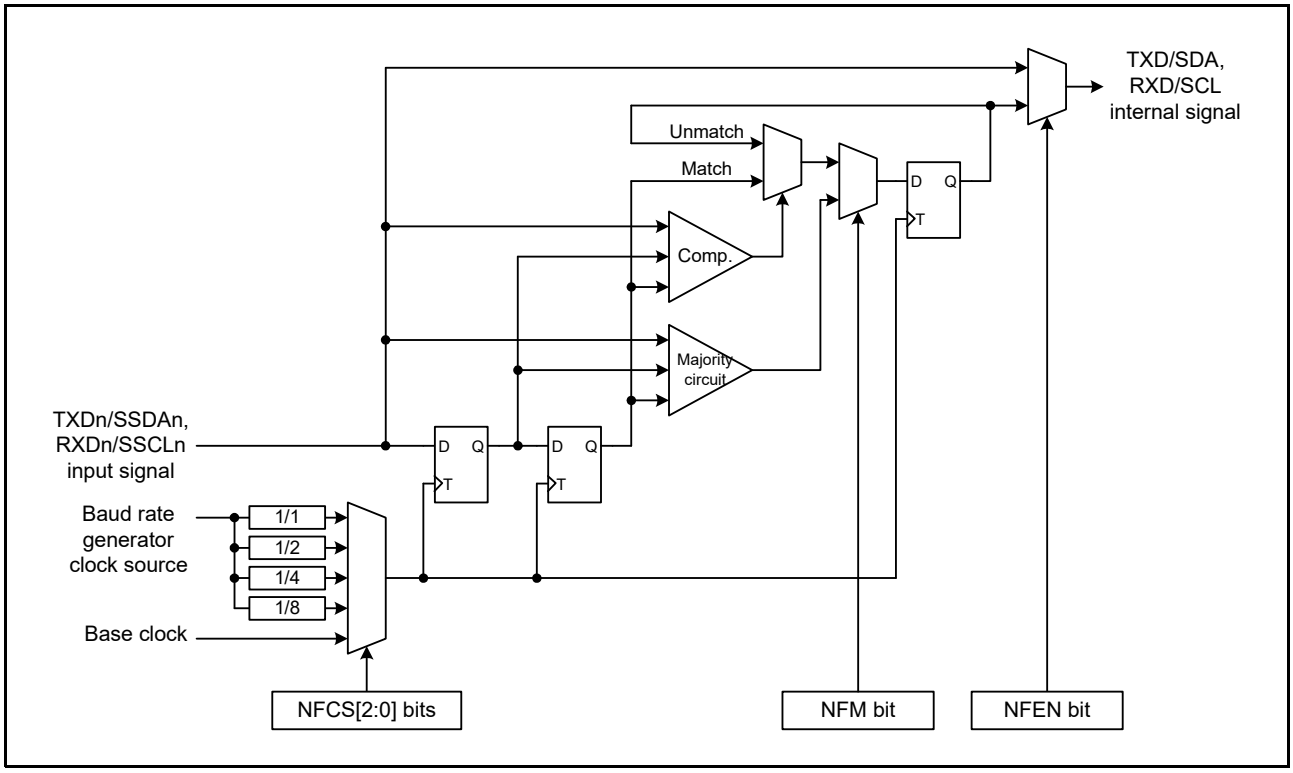


Figure 42.110 Block Diagram of Digital Noise Filter

42.14 RS-485 Driver Control Function

Setting the DEEN bit in the RSCI control register 3 (SCR3) to 1 enables the RS-485 driver control function and generates a DE (Driver Enable) signal that enables the external transceiver transmission mode. The DE signal outputs a valid level for the period with setup time and hold time added before and after data transmission. The DE signal valid level is set by the DELVL bit in the DE signal control register (DECR).

The setup time is the time from when the DE signal is valid until the start bit starts. Set by DESU[4:0] bits of DE signal control register (DECR).

The hold time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DE signal. Set with DEHLD[4:0] bits of the DE signal control register (DECR).

DESU[4:0] and DEHLD[4:0] bits are expressed in base clock units (1/4, 1/6, 1/8, or 1/16 bit time). For details, refer to section 42.2.13, DE Signal Control Register (DECR).

When this function is used (DEEN bit = 1), the TEND set timing and TEI interrupt output timing are at the end of the DE signal hold time.

When transmission is completed and the next transmission data is not written before the DE signal is negated, the DE signal is negated once. If the timing for writing the next transmit data is not in time, the DE signal is negated and asserted again, the setup time is inserted, and the next data is transmitted. If you want to perform the next transmission with the DE signal asserted, write the next transmission data to the TDR quickly enough.

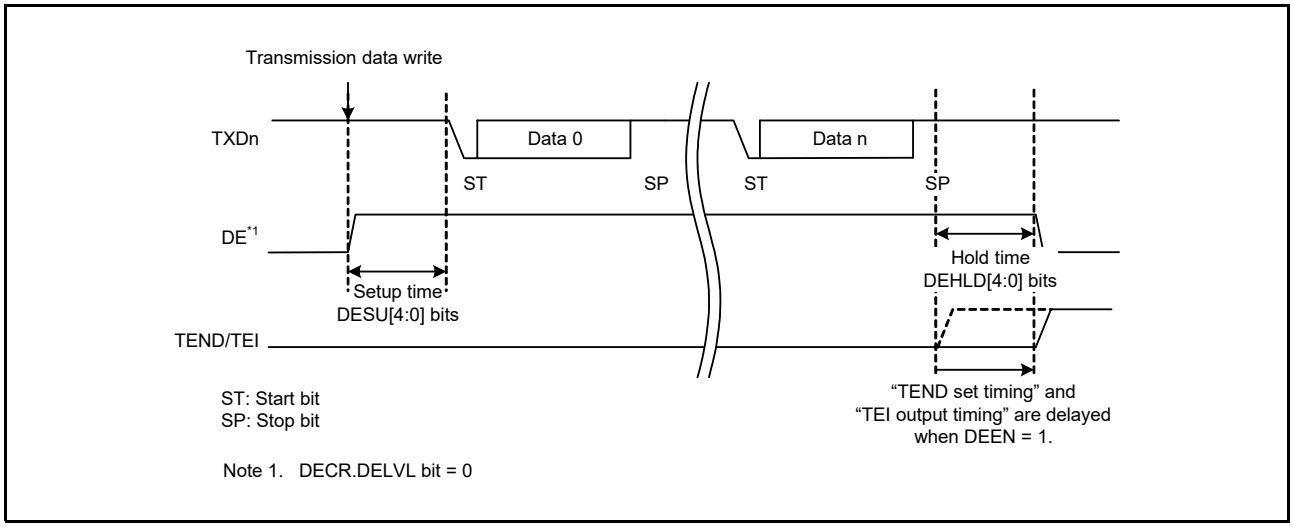


Figure 42.111 Image Waveform for RS-485 Driver Control DE Signal Output

42.15 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and Manchester mode with the internal clock, and Clock synchronous mode with the internal clock.
When 1 is written to the LOOP bit in the SCR1 register, RSCI blocks the external input (RXD) path and connects the output path of the transmit data register and the input path of the receive data register.
When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV bit = 1 only when operating in clock synchronous mode internal clock.
Table 42.42 shows the relationship between the TINV and LOOP bit settings and the received data.

Table 42.42 TINV and LOOP Bit Settings and Received Data

			Communication Mode		
TINV	**LOOP**	**Receive Data**	**Async Internal Clock**	**Manchester Internal Clock**	**Clock Sync Internal Clock**
0 or 1	0	Receive data from RXDn pin	Possible	Possible	Possible
0	1	Transmit data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Figure 42.112 shows the configuration of the shift register input/output path in loopback mode.

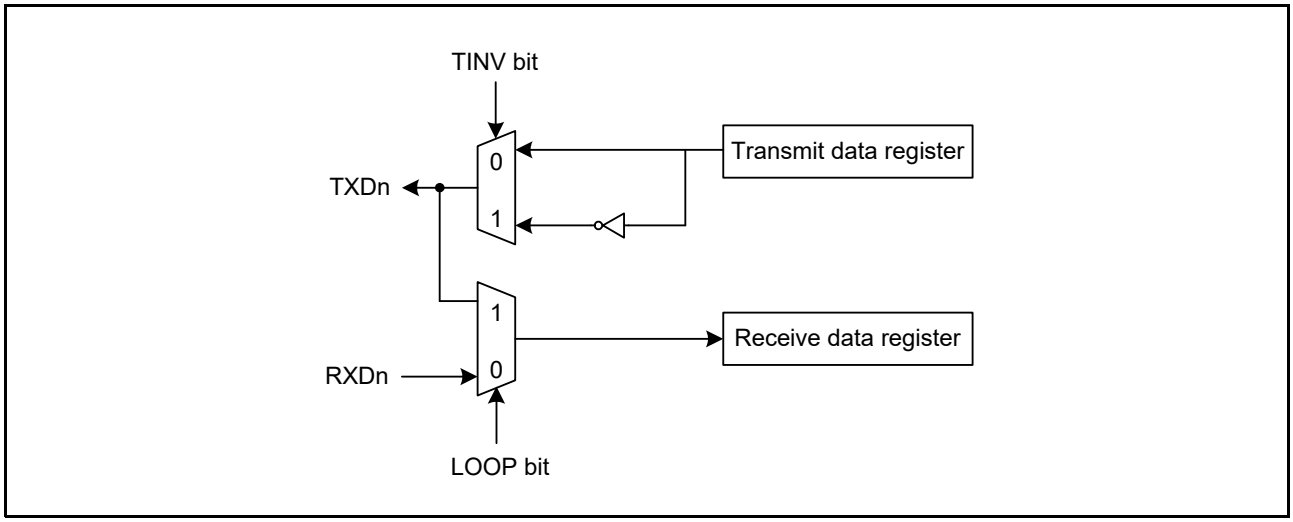


Figure 42.112 Shift Register Input/Output Configuration Image in Loopback Mode

42.16 Half-Duplex Communication Function

Do not use the half-duplex communication function in simple I^2C mode, simple-SPI mode, and smart card interface mode.

In other communication modes, if the SCR1.HDSEL bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission/reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the RSCI and is not actually transmitted.

During half-duplex communication, the only communication port terminal used is the TXDn pin. Output when SCR0.TE bit = 1, input when SCR0.TE bit = 0.

42.17 Synchronizer Bypass Function

Each RSCI has a circuit that operates in synchronization with the peripheral module clock (PCLK) and a circuit that operates in synchronization with the operating clock (TCLK). From the entry of signals for transfer between these two types of circuit to the given synchronizer, the signals take two to three cycles of PCLK or TCLK to pass through the synchronizer, producing a corresponding delay. However, setting the SCR3.SYNDIS bit to 1 selects the peripheral module clock as the operating clock, and the synchronizer is bypassed. In this case, the above delay is reduced to zero, which improves the responsiveness. Furthermore, for the RSCI, although there is also a synchronizer between the circuit that operates in synchronization with the transfer clock and the circuit that operates in synchronization with the operating clock, bypassing that synchronizer is not possible.

42.18 Interrupt Signal

Table 42.43 lists RSCI interrupt signals.

The interrupt explanation corresponding to each operation mode is described in sections 42.18.2 to 42.18.5. Also, TXI and RXI have an interrupt buffer function. Refer to section 42.18.1, Buffer Operations for TXI and RXI Interrupts. When performing transmission and reception using DTC or DMAC, be sure to set DTC or DMAC first, and then enable RSCI before setting. Refer to section 20, Data Transfer Controller (DTCA) and section 18, DMA Controller (DMACAa) for how to set DTC or DMAC.

Table 42.43 RSCI Interrupt List

Interrupt Symbol	Interrupt Function	Pulse/ Level	Pulse Width	Active Level	SYNC Clock	Note
ERI	Error interrupt Bus collision detection interrupt	Level	—	Low	PCLK	
RXI	Simple I²C: Reception end interrupt Other mode: Receive data full interrupt	Pulse	1 cycle	Low	PCLK	
TXI	Simple I²C and smart card interface: Transmit end interrupt Other mode: Transmit data empty interrupt, Break Field transmission completion	Pulse	1 cycle	Low	PCLK	
TEI	Simple I²C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Level	—	Low	PCLK	
AED	Active edge detection interrupt	Pulse	1 cycle	Low	PCLK	
BFD	Break Field detection interrupt	Level	—	Low	PCLK	Only when extended serial mode

42.18.1 Buffer Operations for TXI and RXI Interrupts

The TXI and RXI interrupts have an interrupt buffer function. When the first interrupt request is generated during interrupt handling and the next interrupt request is generated (when the status flag of the interrupt controller (ICU) is 1), the RSCI does not output the interrupt request, and holds it internally. The interrupt that can be held is up to one.

42.18.2 Interrupt in Asynchronous Mode, Manchester Mode, Clock Synchronous Mode, and Simple SPI Mode

A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR0 register.

(1) Non-FIFO Mode

Table 42.44 lists interrupt sources in asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. At the start of transmission, a TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.*1

When the SCR0.TEIE bit is 1, the SSR.TEND flag becomes 1 and the TEI interrupt request is generated if the next data is not written to the TDR register by the timing to transmit the last bit of transmission data. In addition, the TEND flag holds 1 during the period from setting the SCR0.TE bit to 1 until writing transmit data to the TDR register, and if the TEIE bit is set to 1, a TEI interrupt request is generated.

Writing data to the TDR register clears the TEND flag and cancels the TEI interrupt request, but it takes time to cancel it.

If the SCR0.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, AFER, and APER flags in the SSR register or the MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2 flags in MMSR register to 1 while the SCR0.RIE bit is 1 leads to the generation of an ERI interrupt request.

An RXI interrupt request is not generated at this time. Clearing all flags (ORER, AFER, APER, MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmit end interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Note 2. In Manchester mode only, MMSR.SYER (if SYERIE bit = 1), PFER (if PFERIE bit = 1), SBER (if SBERIE bit = 1) flags are added to the ERI interrupt sources.

(2) FIFO Mode

Table 42.45 lists interrupt sources in asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode with FIFO enabled.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when the stored number of data in transmit FIFO becomes equal to or less than the transmit FIFO threshold. A TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.TIE bit to 1 at the same time. A TXI interrupt request is not generated by setting the SCR0.TE bit to 1 while the setting of the SCR0.TIE bit is 0 or by setting the SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.

If SCR0.TEIE bit is 1, when the next data isn't being written in transmit FIFO by the timing to which the last bit of the transmission data is sent, SSR.TEND flag will be 1 and TEI interrupt request is generated.

If the SCR0.RIE bit is 1, RXI interrupt request is generated when the stored number of data in receive FIFO exceeds the threshold. When the threshold value is set to 0, RXI interrupt request is occurred if the quantity of data in receive FIFO is greater than or equal to 1.

If the SCR0.RIE bit is 1, when SSR.ORER flag is set to 1 or the data a framing error or a parity error generated is stored in receive FIFO, ERI interrupt request is generated.

When the number of data stored in a receive FIFO at this time is a threshold value or above, RXI interrupt request is also generated. The ERI interrupt request can be canceled by clearing all flags (SSR.ORER, AFER, APER).

Table 42.44 RSCI Interrupt Sources (in Non-FIFO Mode)

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, MCER, SYER (SYERIE = 1), PFER (PFERIE = 1), SBER (SBERIE = 1)	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
	Receive data match	DCMF		
TXI	Transmit data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

Table 42.45 RSCI Interrupt Sources (in FIFO Mode)

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, DR (FCR.DRES bit = 1)	RIE	Not possible
RXI	Receive FIFO data full	RDRF	RIE	Possible
	Receive data ready	DR (FCR.DRES bit = 0)		
	Receive data match	DCMF		
TXI	Transmit FIFO data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

42.18.3 Interrupt in Smart Card Interface Mode

Table 42.46 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 42.46 RSCI Interrupt Sources in Smart Card Interface Mode

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error or error signal detection	ORER, APER, ERS	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
TXI	Transmit end	TEND	TIE	Possible
	TE = 0 → 1 detection			

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode. In transmission operation, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.
If an error occurs, the RSCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR0.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.
In reception operation, an RXI interrupt request is generated when receive data is set to RDR register. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making RSCI settings. For DTC or DMAC settings, refer to section 20, Data Transfer Controller (DTCA) and section 18, DMA Controller (DMACAa).

42.18.4 Interrupts in Simple I2C Mode

Table 42.47 lists RSCI interrupts in Simple I2C mode.

The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR.IICINTM bit is 1, a RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data. (In this case, ACK/NACK judging are impossible.)

When the value of the SIMR.IICINTM bit is 0, RSCI moves as follows. RXI request (ACK detection) is generated if the input on the SSDAn pin is at the low level on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). TXI request (NACK detection) is generated if the input on the SSDAn pin is at the high level on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the RSCI.

When the SIMR.IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 42.47 RSCI Interrupt Sources in Simple I2C Mode

Name	Interrupt Source		Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
	IICINTM bit = 1	IICINTM bit = 0			
RXI	Reception end	—	—	RIE	Possible*1
	—	ACK detection	—		Possible
TXI	Transmit end	—	—	TIE	Possible*1
	—	NACK detection	—		Possible
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you can not confirm whether ACK or NACK.

42.18.5 Interrupts in Extended Serial Mode

Table 42.48 lists interrupt sources in extended serial mode.

Table 42.48 RSCI Interrupt Sources in Extended Serial Mode

Name	Interrupt Source	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
ERI	Error	ORER, AFER, APER	—	RIE	Not possible
		BCDF		BCDIE	
		COF		RIE, COFIE	
RXI	Reception data full	RDRF	CF0MF, CF1MF, PIBDF	RIE	XSR0.SFSF flag = 0: Possible XSR0.SFSF flag = 1: Not possible
AED	Active edge detection	AEDF	—	AEDIE	Possible
TXI	Transmit data empty	TDRE	—	TIE	Possible
	TE = 0 → 1 detection				
	Break Field transmission completion	BFOF	—	TIE, BFOIE	
TEI	Transmit end	TEND	—	TEIE	Not possible
BFD	Break Field detection	BFDF	—	BFDIE	Not Possible (Unnecessary)

In extended serial mode, in addition to reception errors (overrun, framing, and parity errors), an ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the extended serial module occurs. At this time, a RXI interrupt request is not output. The ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if SCR0.TIE bit = 1 and XCR0.BFOIE bit = 1, a TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set SCR0.TEIE bit = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF flag = 1), reception using DTC or DMAC by RXI interrupt is not possible. Check the SSR register and XSR0 register, check the reception status (See Figure 42.73), and then clear the flag.When data is received, read the RDR register so that an overrun error does not occur (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF flag = 1), Start Frame detection is disabled (XSR0.SFSF flag = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1), if a Break Field longer than the period set in XCR2.BFLW[15:0] bits is received, the BFDF flag is set and a BFD interrupt request is output. Then RSCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1) and the bit rate measurement function is enabled (XCR1.BRME bit = 1), an AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.CCV[15:0] bits).

42.19 Event Linking

By employing interrupt request signals as event signals, RSCI can provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits. And even when the next interrupt source occurs while the corresponding interrupt status flag is 1, the event signal can be output.
All event output is a pulsed output and becomes negated behind 1 PCLK of asserting.
Table 42.49 lists RSCI event link signals.

Table 42.49 RSCI Event Link Signal List

Function	Pulse/Level	Pulse Width	Active Level	Synchronize Clock
Error event	Pulse	1 cycle	High	PCLK
Receive data full event	Pulse	1 cycle	High	PCLK
Transmit data empty event	Pulse	1 cycle	High	PCLK
Transmit end event	Pulse	1 cycle	High	PCLK
Receive data match event	Pulse	1 cycle	High	PCLK
Receive data unmatch event	Pulse	1 cycle	High	PCLK
Active edge detect event	Pulse	1 cycle	High	PCLK

(1) Error Event Output (Receive Error, Error Signal Detection)

- This output indicates abnormal termination due to a parity error during reception.
- This output indicates abnormal termination due to a framing error during reception.
- This output indicates abnormal termination due to an overrun error during reception.
- This output indicates abnormal termination due to a Manchester code error during reception (only in Manchester mode).
- This output indicates that a preface error occurred upon reception and abnormal termination occurred (only in Manchester mode and MMCR.PFERIE bit = 1).
- This output indicates that a start bit error occurred during reception and abnormal termination occurred (only in Manchester mode and MMCR.SBERIE bit = 1).
- This output indicates that a reception sync error occurred during reception and abnormal termination occurred (only in Manchester mode and only when MMCR.SYERIE bit = 1).
- This output indicates detection of the error signal during transmission in smart card interface mode.
- SSR.AFER and APER flags are 0, and reception data less than receive FIFO data trigger number is set in a receive FIFO buffer, and this output indicates that time of 15 etu has passed when FIFO enabled and FCR.DRES bit is 1.
- In extended serial mode, this output indicates that the 16-bit counter in the extended serial module has overflowed.
- In extended serial mode, a bus collision is detected during transmission (SCR0.TE bit = 1).

(2) Receive Data Full Event Output

- In non-FIFO mode, this output indicates that received data have been set in the receive data register (RDR).
- This output indicates that ACK has been detected if the SIMR.IICINTM bit is 0 in simple I^2C mode.
- This output indicates that the 8th-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

Note: Do not use this event in FIFO mode.

(3) Transmit Data Empty Event Output

- This output indicates that the SCR0.TE bit has been changed from 0 to 1.
- In non-FIFO mode, this output indicates that transmit data have been transferred from the transmit data register (TDR) to the transmit shift register (TSR).
- This output indicates that transmission has been completed in smart card interface mode.
- This output indicates that NACK has been detected if the SIMR.IICINTM bit is 0 in simple I^2C mode.
- This output indicates that the ninth-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I^2C mode.
- In extended serial mode, this output indicates that Break Field transmission is complete.

Note: Do not use this event in FIFO mode.

(4) Transmit End Event Output

- This output indicates the completion of transmission.
- This output indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.
- In smart card interface mode, the transmit end event is not output.

Note: Do not use this event in FIFO mode.

(5) Receive Data Match Event Output

- This output indicates that the received data matched the value of the SCR4.CMPD[8:0] bits in asynchronous mode (including multi-processor mode).

(6) Receive Data Unmatch Event Output

- This output indicates that the received data did not match the value of the SCR4.CMPD[8:0] bits in asynchronous mode (including multi-processor mode).

(7) Active Edge Detection Event Output

- In extended serial mode, when XCR1.BRME bit is 1, this output indicates that a valid edge has been detected in the RXD input signal.

42.20 Usage Notes

42.20.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start RSCI operations. With the value after a reset, RSCI operations are stopped. The registers of the modules only become accessible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

42.20.2 RSCI Operations during Low Power Consumption State

(1) Transmission

Before using the power consumption reduction function to reduce RSCI's power consumption, please do the following to confirming transmission end (SSR.TEND flag = 1):

- Set the output terminal state after transmission operation is stopped by SCR1.SPB2DT and SPB2IO bits.
- Stop the transmission (SCR0.TIE bit = 0, TE bit = 0, TEIE bit = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.
To transmit data in the same operating mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR register, and write data to TDR sequentially to start data transmission. To transmit data with a different operating mode, initialize the RSCI first.
To start transmission using the DMAC/DTC after cancellation from software standby mode, set the TE and TIE bit to 1 simultaneously. The TXI interrupt is generated and transmission starts using the DMAC/DTC.
Figure 42.113 shows a sample flowchart for transition to software standby mode during transmission. Figure 42.114 and Figure 42.115 show the port pin states during transition to software standby mode.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR0.RE bit = 0). If transition is made during data reception, the data being received will be invalid.
Figure 42.116 shows a sample flowchart for reception to software standby mode during reception.

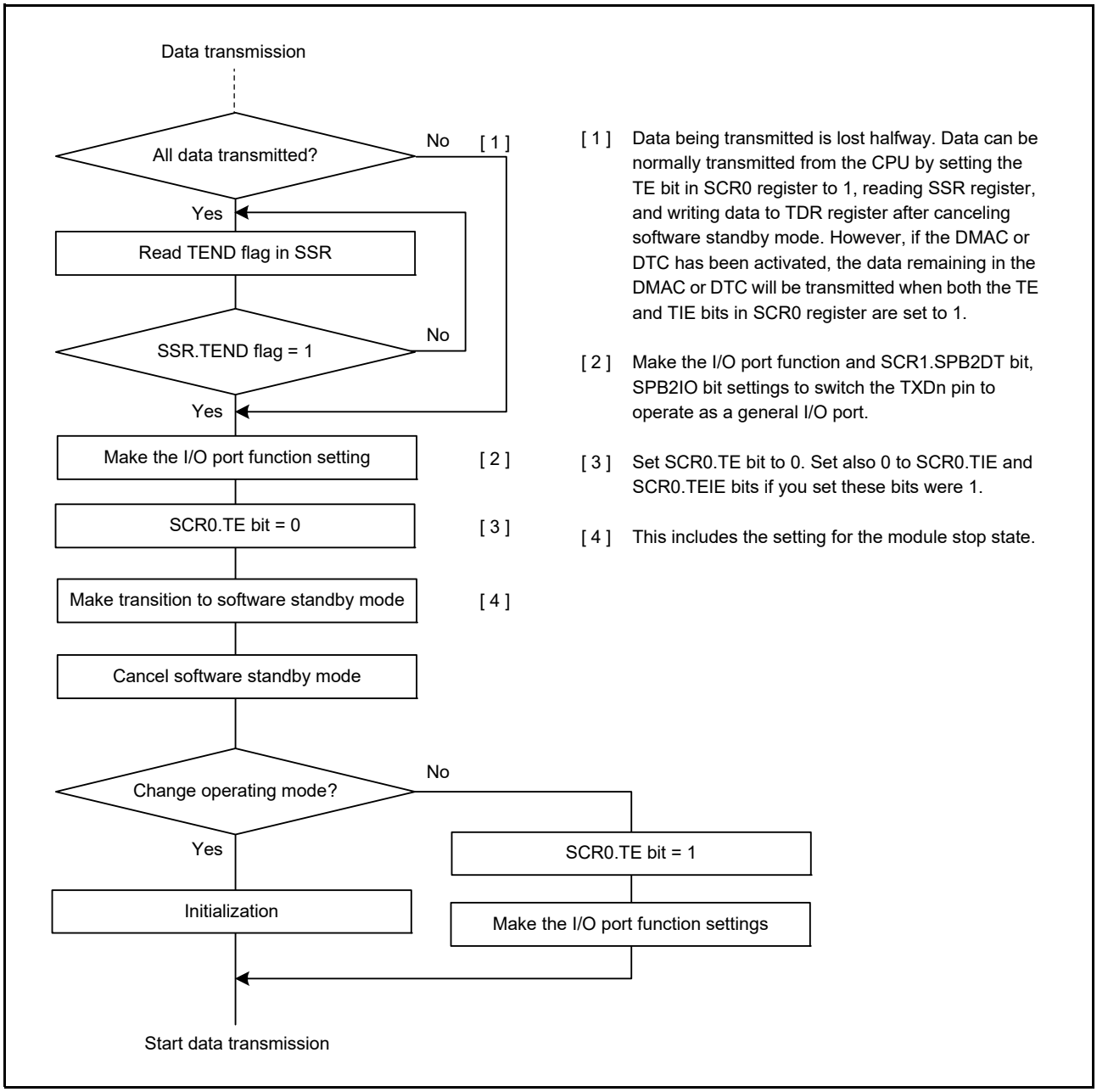


Figure 42.113 Example of Flowchart for Transition to Software Standby Mode during Transmission

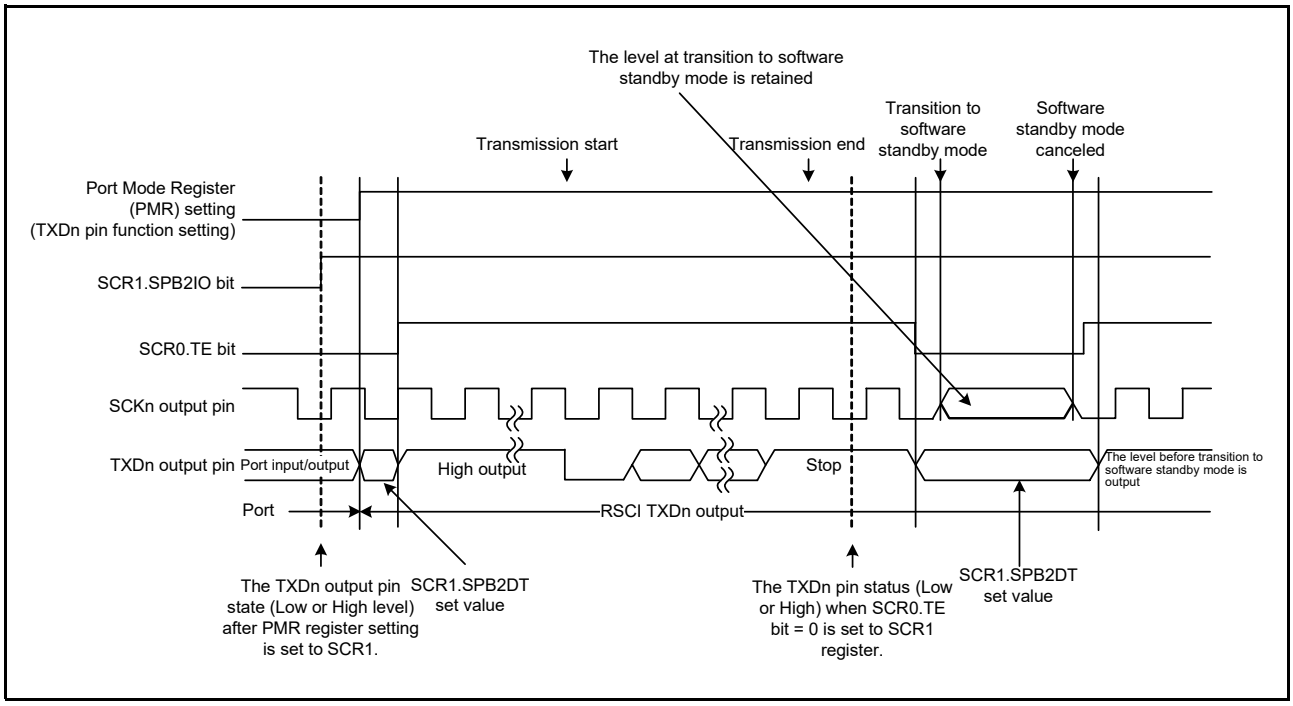


Figure 42.114 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

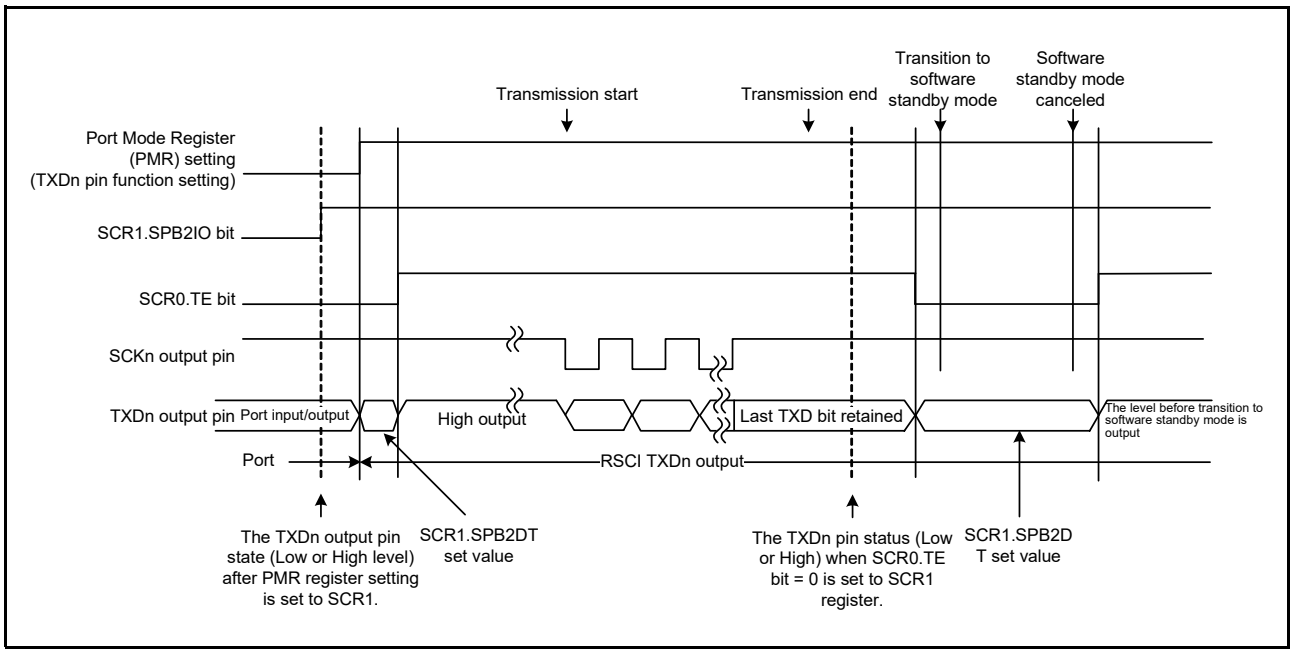


Figure 42.115 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

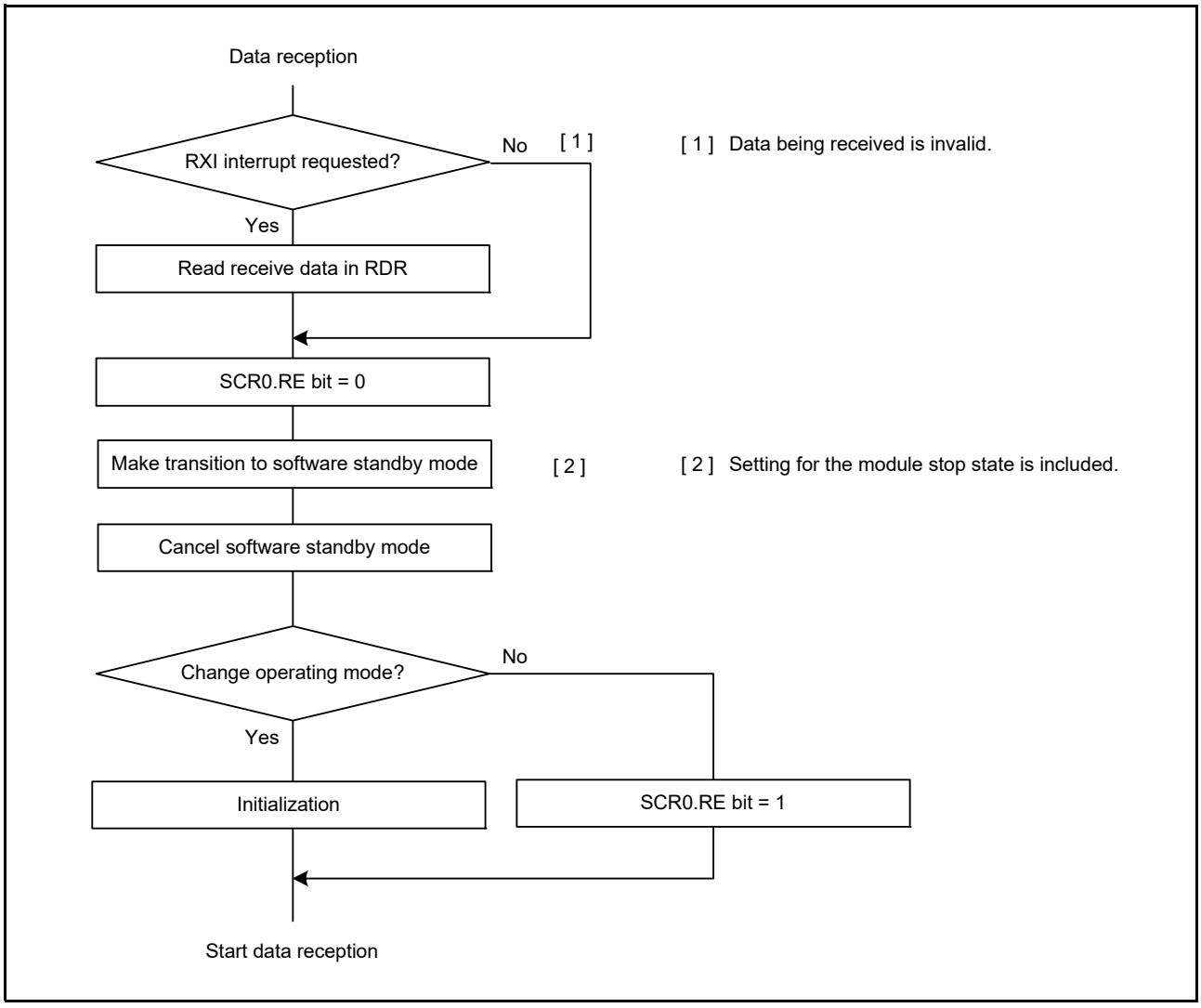


Figure 42.116 Example of Flowchart for Transition to Software Standby Mode during Reception

42.20.3 Break Detection and Processing

(1) Non-FIFO Mode

When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. In a break, the input from the RXDn pin becomes all low, and so the SSR.AFER flag is set to 1 (framing error has occurred), and the SSR.APER flag may also be set to 1 (parity error has occurred). When the SCR3.RXDESEL bit is 0, the RSCI continues the receive operation even after a break is received. Therefore, note that even if the AFER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SCR3.RXDESEL bit is 1, the RSCI sets the SSR.AFER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.AFER flag is set to 0 at this time, the SSR.AFER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the start bit at the first falling edge of the RXDn pin allows the RSCI to start the receiving operation.

(2) FIFO Mode

After a framing error is detected, when RSCI detects that continuous receive data is low for 1 frame, the data stored into the receive FIFO (RDR register) and reception stops. When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. After the RXD signal is in the mark state and it has finished the break, a stock of reception data to the receive FIFO (RDR register) resumes.

42.20.4 Mark State and Sending Breaks

When the SCR0.TE bit is 0 (serial transmission is disabled), the state of the TXDn pin can be set by SCR1.SPB2IO bit and SCR1.SPB2DT bit. Using this, it's possible to do a TXDn pin in the mark state and send a break out. When you want to make a communication-line is in the mark state (the state of 1) until the SCR0.TE bit is set to 1 (serial transmission is enabled). First, set it as High-level output by setting SPB2IO bit and SPB2DT. Next, it's changed to a TXDn pin by I/O port function. On the other hand, if you want to send a break when sending data, set the SCR0.TE bit to 0 after setting the SPB2IO and SPB2DT bits in the SCR1 register to low level output. Setting the TE bit to 0 initializes the transmitter regardless of the current transmission status.

42.20.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR register even if SSR.ORER flag is 1. However, reception can not be started. Note also that the receive error flags cannot be set to 0 even if the SCR0.RE bit is set to 0 (serial reception is disabled).

42.20.6 Writing Data to the TDR Register

(1) Non-FIFO Mode

Data can be written to TDR register anytime when TE bit is 1. However, if new data is written to TDR register when transmit data is remaining in TDR register, the previous data in TDR register is lost because it has not been transferred to TSR register yet. If you use DTC or DMAC, be sure to write transmit data to TDR register in the TXI interrupt request handling routine.

(2) FIFO Mode

Data can be written to transmit FIFO (TDR register) when TE bit is 1. Check the number of writable data with the TFSR.T[5:0] bits.

42.20.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of Transmission

Update TDR register by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 42.117)
Time considering the output AC characteristics of the SMISO pin of this product and the input AC characteristics of the master reception + 1 PCLK cycle + 2 to 3 TCLK cycles (delay time of the synchronizer).

(2) Continuous Transmission

The next transmit data must be transferred to the TSR register before the falling edge*1 of the transmit clock for bit 7. Write the transmit data to TDR register taking into account the delay time of the synchronizer. If the transmit data can not be written in time, the previous frame data is resent. (See Figure 42.117)

Note 1. When SCR3.CPOL bit = 1 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 0 and SCR3.CPHA bit = 1. In the case of SCR3.CPOL bit = 0 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 1 and SCR3.CPHA bit = 1, it's the rising edge.

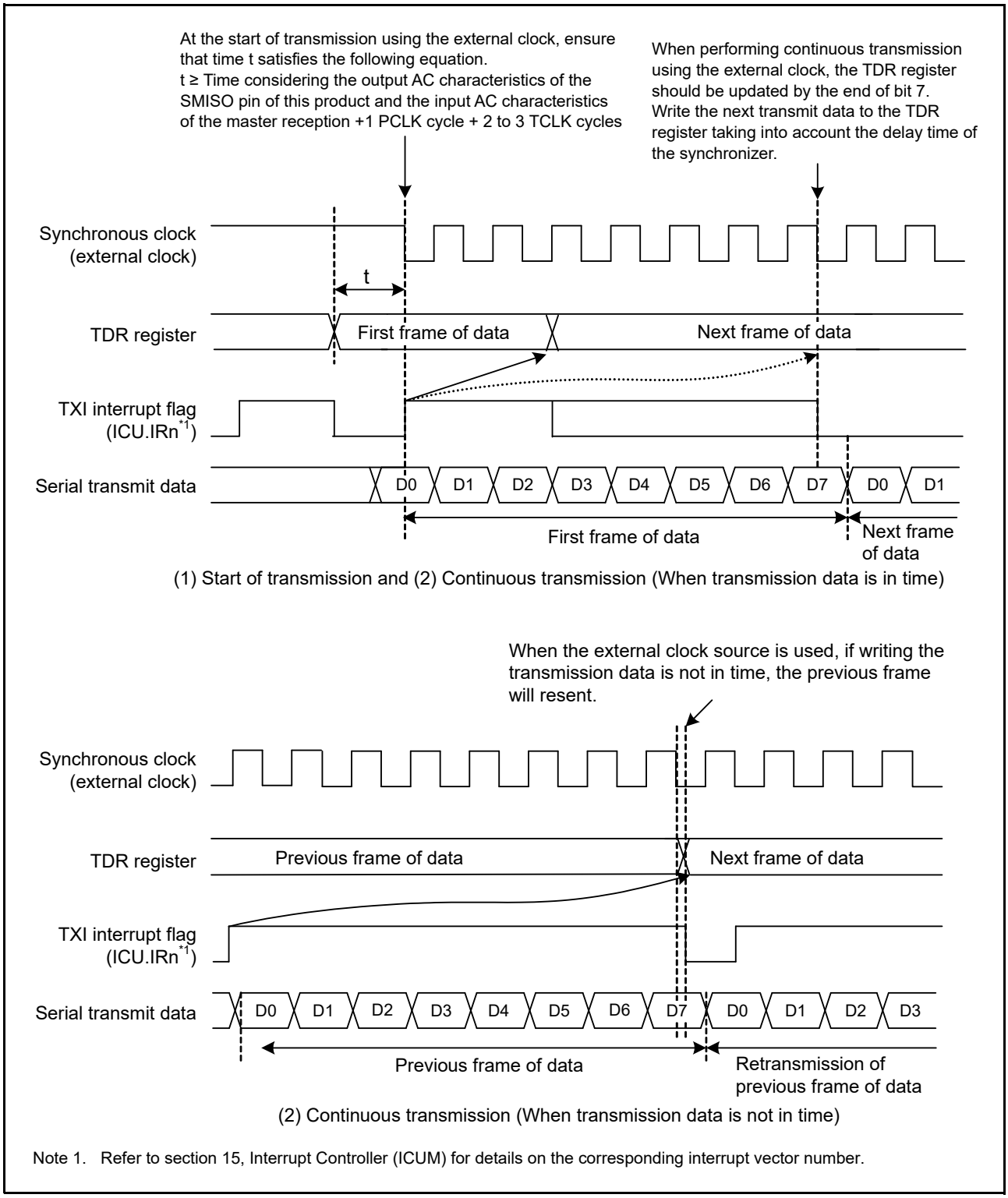


Figure 42.117 Restrictions on Use of External Clock in Clock Synchronous Transmission

42.20.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR register, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant RSCI.
During the operation in transmission/reception using the DMAC or DTC, it should not set transfer information of DMAC/DTC.

42.20.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR0.TE or SCR0.RE bit to 1). For details on the interrupt status flag, refer to section 15, Interrupt Controller (ICUM).

- Confirm that transfer has stopped (the setting of the SCR0.TE or SCR0.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

42.20.10 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SCR3.CPHA and CPOL bits when the SCR0.SSE bit is 1. This prevents the clock line from being placed in the high-impedance state when the SCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR0.TE bit is changed from 0 to 1.
  In a single-master configuration, pull up or pull down the clock line is not necessary because the clock line does not become high-impedance state when SCR0.SSE bit = 0 and SCR0.TE bit = 0.
- In the case of the setting for clock delay (SCR3.CPHA bit is 0), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 42.118. If the TE and RE bits become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault occurs as the current character is being transferred. And stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

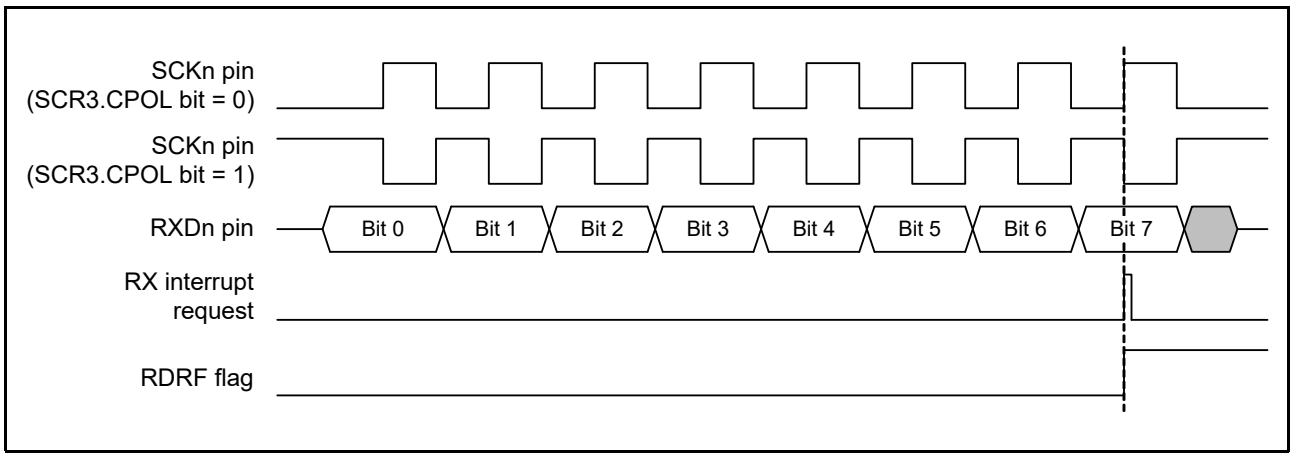


Figure 42.118 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- It takes “1 PCLK + 2 to 3 TCLK cycles (delay time of the synchronizer) + data output delay time (AC characteristics)” from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer.
- Secure the SS input setup time (AC characteristics) from the SSn# low-level input to the start of external clock input.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, the transmission and reception is stopped immediately. Set the TE and RE bits in the SCR0 register to 0 and, after remaking the settings, restart transfer of the first byte.

42.20.11 Note on Transmit Enable Bit (TE Bit)

In the initial register value, when setting the SCR0.TE bit to 0 (serial transmission is disabled) while the pin function is "TXDn", output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(a) Connect a pull-up or pull-down resistor to the TXDn line.
(b) Set the SCR1 register to determine the level of the TXDn pin when the TE bit is 0.

42.20.12 Note on Extended Serial Mode

In extended serial mode (SCR3.MOD[2:0] bits = 110b), the following functions cannot be used.

- CTS and RTS functions
- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

42.20.13 Notes on RS-485 Driver Control Function

- RS-485 Driver control function is valid only in Asynchronous mode.
- When RS-485 Driver control function is active (SCR3.DEEN bit = 1), the TEND set timing/TEI output timing changes as follows. Wait for the TEI interrupt and set the TE bit to 0.
  When RS-485 Driver control function is inactive: When STOP bit output is completed.
  When RS-485 Driver control function is active: At the end of DE signal hold time.

42.20.14 Notes on Loopback Function

The Loopback function is valid in Asynchronous mode with internal clock, in Manchester mode with internal clock and Clock synchronous mode with internal clock.
It can also operate in the asynchronous HBS support mode, and when the HBSCR.AOE bit = 1, it loops back the signal with the logical AND of the TXDAn and TXDBn pin outputs (Use with TINV bit = RINV bit = 0).

42.20.15 Note on Register Access When the Frequency of the Operating Clock (TCLK) is Lower Than the Frequency of the Peripheral Module Clock (PCLK)

If the TCLK frequency is lower than the PCLK frequency, when a new value is written to the SCR0.TE or RE bit, the time until the value is propagated to the transmitter or receiver is longer than the register access time. Before rewriting registers and bits that can only be rewritten while the TE and RE bits are set to 0, be sure to set the TE and RE bits to 0 and to confirm that the values of the SISSR.ITEF and IREF flags have become 0.

42.20.16 Note on Aborting Operation

If 0 is written to SCR0.RE bit during data reception and the receive operation is aborted, the status may become invalid depending on the timing. Therefore, do not use the received data (value stored in the RDR register) or the flag value of each status register. To abort the receive operation, disable the interrupt and event link related to reception, and then write 0 to the SCR0.RE bit.

43. I3C Bus Interface (RI3C)

This MCU has a single-channel I3C bus interface (RI3C0).
The RI3C module conforms with the MIPI Alliance I3C bus interface and the NXP I2C-bus (Inter-IC bus) interface, and provides a subset of their functions.

43.1 Overview

Table 43.1 and Table 43.2 list the specifications of the RI3C and Figure 43.1 and Figure 43.2 show block diagrams of the RI3C. Table 43.3 lists the I/O pins of the RI3C.

Table 43.1 RI3C Specifications (I2C Protocol Mode) (1/2)

Item	Specification
Operation mode	Four • Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Communication format/ Transfer rate	• I2C-bus format - Standard-mode (Sm): 0 to 100 kbps - Fast-mode (Fm): 0 to 400 kbps - Fast-mode Plus (Fm+): 0 to 1 Mbps - High-speed mode (Hs-mode): 0 to 3.4 Mbps • SMBus format: 10 to 100 kbps • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Serial clock (SCL)	• For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	START, repeated START, and STOP conditions are automatically generated. START conditions (including repeated START conditions) and STOP conditions are detectable.
Address detection	• Slave address (static address) (max 3 address) • 7-bit and 10-bit address formats are supported. • General call address • Hs-mode master code • Device ID • Host address
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function (clock stretch)	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	It is possible to delay the change timing of the output of the transmission data bits and the acknowledgment bit relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the START condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the START condition while the bus is busy is detectable (to prevent the generating of double START conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise filter	• Analog noise filter • Digital noise filter

Table 43.1 RI3C Specifications (I^2C Protocol Mode) (2/2)

Item	Specification
Interrupt source	Five sources: • Transmit data empty (TXI) • Receive data full (RXI) • Transmission end (TEI) • Communication error or communication event (EEI) - START condition detection (including repeated START condition) - STOP condition detection - NACK detection - Arbitration lost - Timeout detection • Wake-up (WUI) - Slave address match detection
Event link function (output)	• Transmit data empty • Receive data full • Transmission end • Communication error or communication event
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attributes can be set.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 43.2 RI3C Specifications (I3C Protocol Mode) (1/2)

Item	Specification
Operation mode	Controller (Primarily Controller/Secondary Controller) mode and Target mode selectable
Data handler	• Controller: - High-priority FIFO buffer transfer - FIFO buffer transfer • Target: - FIFO buffer transfer
Communication protocol	• SDR (I3C single data rate) mode - Private message - Broadcast message (common command code (CCC)) - Direct message (common command code (CCC)) • Legacy I^2C message - Fast-mode (Fm): 0 to 400 kbps - Fast-mode Plus (Fm+): 0 to 1 Mbps
IBI	• Target Interrupt Request (TIR) • Controller Role Request (CRR) (Secondary Controller only)
Address format	7-bit address
Address detection	• Target address (Static Address or Dynamic Address) • Broadcast Address (7Eh)
Clock stalling	The I3C Controller can stall the I3C bus while the SCL line is low.
Timing control	• Synchronous time control - Sync Mode • Asynchronous time control - Async Mode 0: Asynchronous Basic Mode - Async Mode 1: Asynchronous Advanced Mode

Table 43.2 RI3C Specifications (I3C Protocol Mode) (2/2)

Item	Specification
Interrupt source	12 sources: • Transmit data empty (TXI) • Receive data full (RXI) • Command queue empty (CMDI) • Response queue full (RESPI) • IBI queue empty/full (IBII) • Receive status queue full (RCVI) • High-priority transmit data empty (HTXI) • High-priority receive data full (HRXI) • High-priority command queue empty (HCMDI) • High-priority response queue full (HRESPI) • Communication error or communication event (EEI) - START condition detection (including repeated START condition) - STOP condition detection - HDR exit pattern detection - Timeout detection - Buffer access error detection - Data transfer abort - Data transfer error - High-priority data transfer abort - High-priority data transfer error • Wake-up (WUI) - Low detection on the SDA line (START condition detection) - Target address match detection
Error detection	• Buffer access error • Address header error • Address NACKed or Dynamic Address Assignment NACKed • Receive overflow or transmit underflow error • Transfer abort • NACK is received during an I^2C write data transfer • Timeout
Event link output	• Transmit data empty event • Receive data full event • Command queue empty • Response queue full • IBI queue empty/full • Receive status queue full • High-priority transmit data empty • High-priority receive data full • High-priority command queue empty • High-priority response queue full • Communication error or communication event • Synchronous timing event • C_REF counter overflow event • C_REF capture event • Additional bus event
Event link input	• T_C1 capture event
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attributes can be set.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

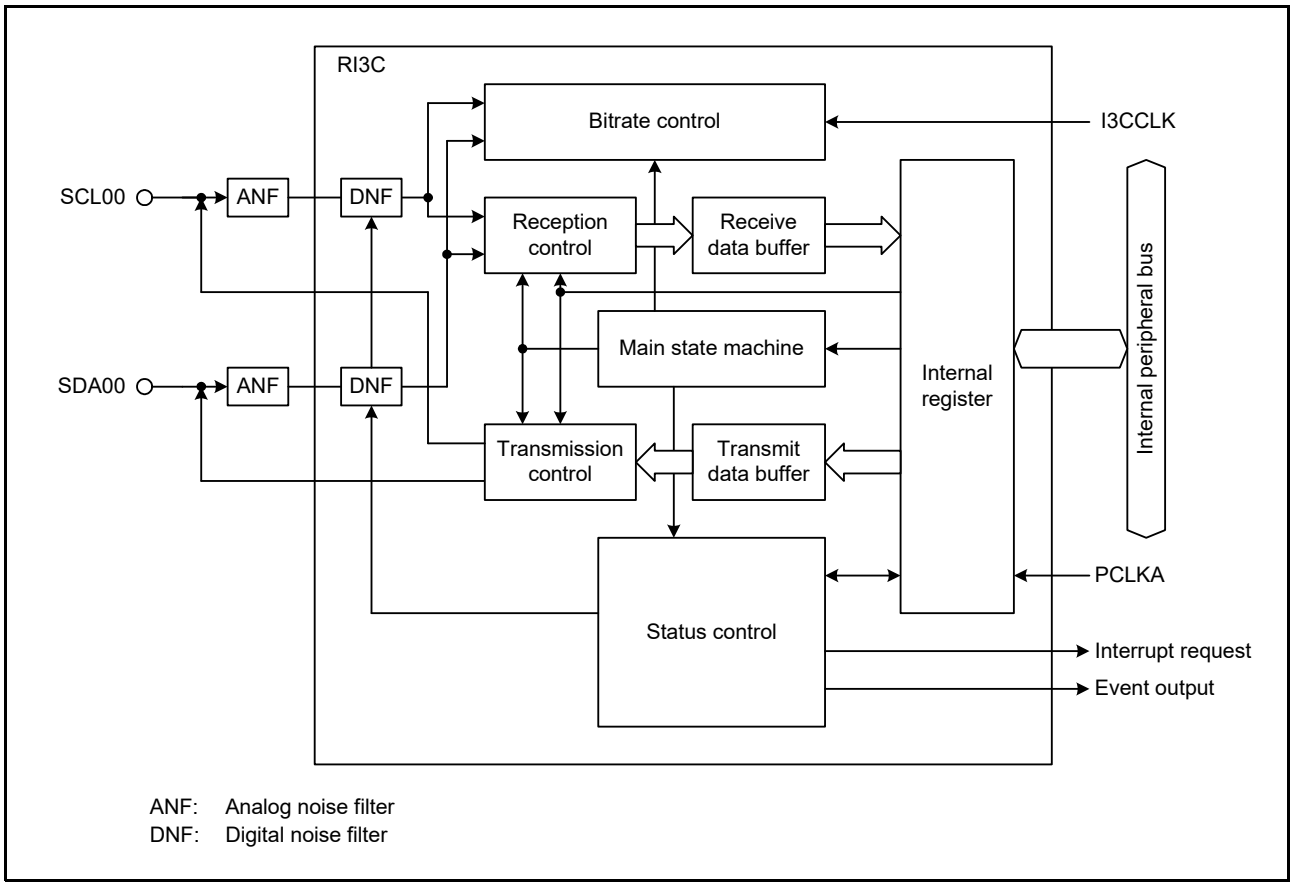


Figure 43.1 RI3C Block Diagram (I²C Protocol Mode)

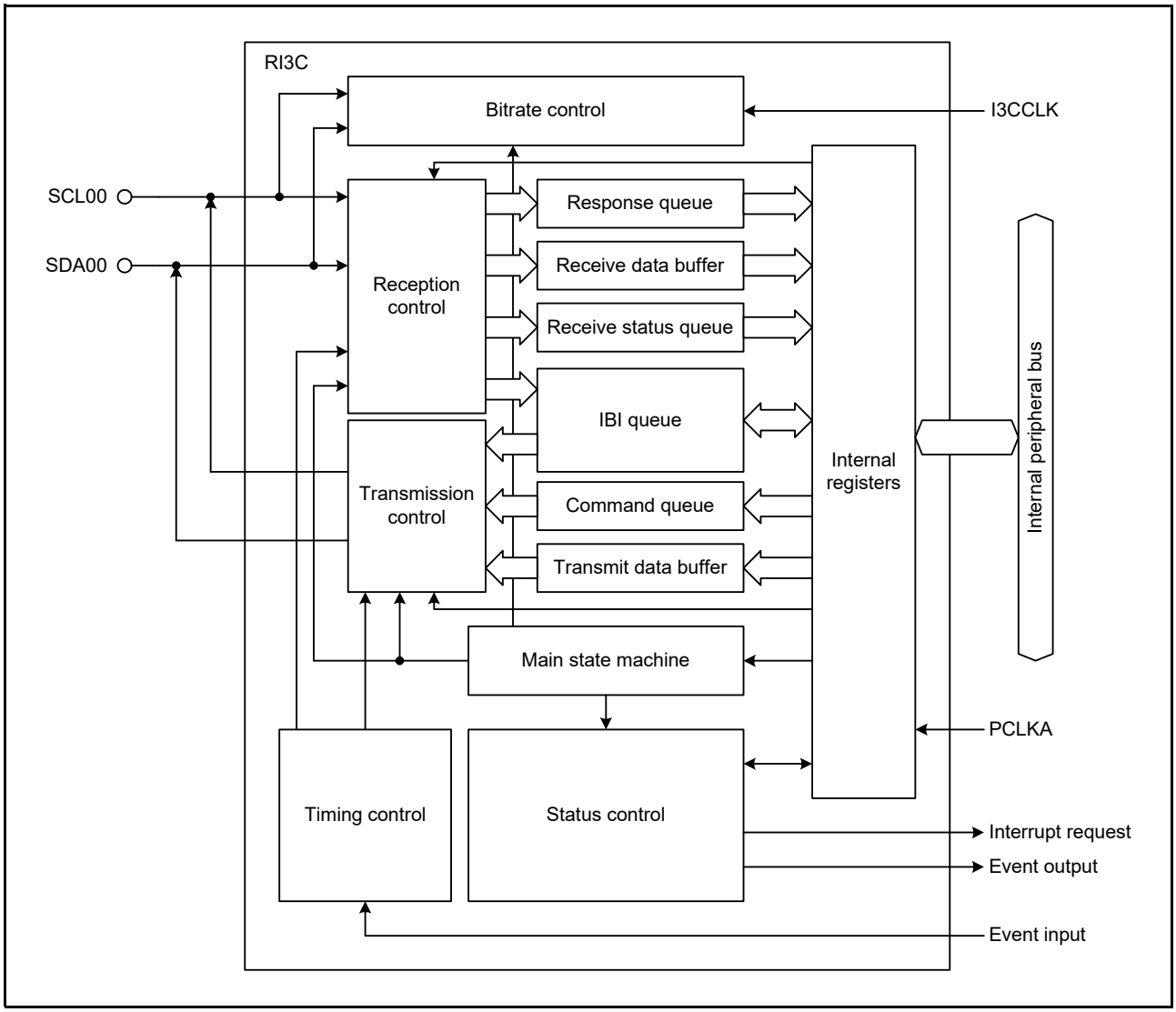


Figure 43.2 RI3C Block Diagram (I3C Protocol Mode)

Table 43.3 RI3C Pin Configuration

Channel	Pin Name	I/O	Function
RI3C0	SCL00	I/O	Serial clock I/O pin
	SDA00	I/O	Serial data I/O pin

43.2 Register Descriptions

43.2.1 Mode Register (ICMR)

Address(es): RI3C0.ICMR 8745 F000h, RI3C0_NS.ICMR 0745 F000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OMS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	OMS	Operating Mode Select*1	0: I3C protocol mode 1: I2C protocol mode	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Rewrite this bit while the ICRCR.ISRST bit is set to 1.

43.2.2 Control Register (ICCR)

Address(es): RI3C0.ICCR 8745 F014h, RI3C0_NS.ICCR 0745 F014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICE	RESUME	ABORT	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BAEXITMD	IBAINC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IBAINC	I3C Broadcast Address Include *1	0: Do not include I3C Broadcast Address for private transfers 1: Include I3C Broadcast Address for private transfers	R/W
b1	BAEXITMD	Broadcast Address Exit Mode*1	0: Does not add an HDR Exit Pattern 1: Add an HDR Exit Pattern	R/W
b28 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29	ABORT	Host Controller Abort*1	Writing 1 to this bit aborts the currently issued transfer. Write 0 to this bit to clear the Aborted state. Values when read: 0: Aborted state is cleared. 1: RI3C remains in Aborted state.	R/W
b30	RESUME	Host Controller Resume*1	Writing 1 to this bit while RI3C is in the Halt state can resume the operation. Values when read: 0: RI3C is running. 1: RI3C is suspended.	R/W
b31	ICE	Bus Interface Enable	0: The I3C bus operation is disabled. (SCL00 and SDA00 pins in inactive state) 1: The I3C bus operation is enabled. (SCL00 and SDA00 pins in active state)	R/W

Note 1. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.
Note 2. This bit is used in I3C protocol mode.

IBAINC Bit (I3C Broadcast Address Include)

This bit controls whether the I3C Broadcast Address (7Eh) is included for private transfers.
If the I3C Broadcast Address is not included for private transfers, then IBIs driven from Target devices might not win the arbitration, potentially delaying acceptance of the IBIs.

BAEXITMD Bit (Broadcast Address Exit Mode)

This bit is used to generate the HDR Exit Pattern on the I3C bus for error recovery.
When this bit is set to 1, the HDR Exit Pattern is sent following the I3C Broadcast Address (7Eh).

ABORT Bit (Host Controller Abort)

When 1 is written to this bit, RI3C starts the abort operation which terminates the currently issued transfer.
In response to an Abort request, RI3C issues the STOP condition to terminate the current transaction on the I3C bus after the complete data byte is transferred or received. At this time, this bit becomes 1.
To resume operation, write 0 to this bit.
If this bit is set to 1 to abort the transaction, ignore the ERR_STATUS field in the Response Descriptor.

RESUME Bit (Host Controller Resume)

This bit is used to resume I3C operation following the Halt state.
RI3C enters the Halt state as a result of any type of error occurring in a transfer. The error type is indicated by the ERR_STATUS field in Response Descriptor.
After RI3C has entered the Halt state, the application must write 1 to this bit to resume operation. This bit is automatically set to 0 when RI3C restarts the transfer (starting the next command).

ICE Bit (Bus Interface Enable)

Enables or disables the operation on the I3C bus.
When using RI3C, set this bit to 1. When this bit is 1, the SCL00 and SDA00 pins are active.
Set this bit to 0 when RI3C is not used. When the ICE bit is 0, the SCL00 and SDA00 pins are inactive.
When this bit is read as 0, it indicates that I3C bus operation is prohibited.

43.2.3 Controller Device Address Register (ICCAR)

Address(es): RI3C0.ICCAR 8745 F018h, RI3C0_NS.ICCAR 0745 F018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAV	—	—	—	—	—	—	—	—	DADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b16	DADR[6:0]	Dynamic Address	Set the Dynamic Address of RI3C.	R/W
b30 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	DAV	Dynamic Address Valid	0: The DADR[6:0] bits are not valid. 1: The DADR[6:0] bits are valid.	R/W

Note: This register is used in I3C Primarily Controller mode.

This register is used to store the Dynamic Address of RI3C.
When the ICCR.ICE bit is set to 1 after setting this register, the device will act as Primary Controller.
If the ICCR.ICE bit is set to 1 after setting the ICDCTR.ROLE[1:0] bits to 00b (I3C Target) or the ICTDCTRm.ROLE[1:0] bits to 01b (I3C Controller capable) without setting this register, the device will act as Target.

DADR[6:0] Bits (Dynamic Address)

These bits are used to program RI3C Controller Dynamic Address. RI3C uses this address to respond to Controller transactions in I3C interface mode (Target or Secondary Controller mode).
In I3C Active Controller mode, the software shall program the Dynamic Address as it self-assigns its Dynamic Address.

DAV Bit (Dynamic Address Valid)

This bit indicates whether or not the value in the DADR[6:0] bits is valid.
In I3C Active Controller mode, set this bit to 1 as it self-assigns its Dynamic Address.

43.2.4 Reset Control Register (ICRCR)

Address(es): RI3C0.ICRCR 8745 F020h, RI3C0_NS.ICRCR 0745 F020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	HRBRST	HTBRST	HRQRST	HCQRST	—	—	SQRST	IQRST	RBRST	TBRST	RQRST	CQRST	MRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRST	Module Reset*1	Writing 1 to this bit resets all registers and internal circuits. This bit automatically returns to 0 when the reset is complete.	R/W
b1	CQRST	Command Queue Reset*2	Writing 1 to this bit flushes the command queue. This bit automatically returns to 0 when the command queue has been flushed.	R/W
b2	RQRST	Response Queue Reset*2	Writing 1 to this bit flushes the response queue. This bit automatically returns to 0 when the response queue has been flushed.	R/W
b3	TBRST	Transmit Data Buffer Reset*2	Writing 1 to this bit flushes the transmit data buffer. This bit automatically returns to 0 when the transmit data buffer has been flushed.	R/W
b4	RBRST	Receive Data Buffer Reset*2	Writing 1 to this bit flushes the receive data buffer. This bit automatically returns to 0 when the receive data buffer has been flushed.	R/W
b5	IQRST	IBI Queue Reset*2	Writing 1 to this bit flushes the IBI queue. This bit automatically returns to 0 when the IBI queue has been flushed.	R/W
b6	SQRST	Receive Status Queue Reset*3	Writing 1 to this bit flushes the receive status queue. This bit automatically returns to 0 when the receive status queue has been flushed.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	HCQRST	High-Priority Command Queue Reset*4	Writing 1 to this bit flushes the high-priority command queue. This bit automatically returns to 0 when the high-priority command queue has been flushed.	R/W
b10	HRQRST	High-Priority Response Queue Reset*4	Writing 1 to this bit flushes the high-priority response queue. This bit automatically returns to 0 when the high-priority response queue has been flushed.	R/W
b11	HTBRST	High-Priority Transmit Data Buffer Reset*4	Writing 1 to this bit flushes the high-priority transmit data buffer. This bit automatically returns to 0 when the high-priority transmit data buffer has been flushed.	R/W
b12	HRBRST	High-Priority Receive Data Buffer Reset*4	Writing 1 to this bit flushes the high-priority receive data buffer. This bit automatically returns to 0 when the high-priority receive data buffer has been flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ISRST	Internal Status Reset	0: Some registers and internal circuits are released from reset. 1: Some registers and internal circuits are reset.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not write any value to this register when this bit is 1.
Note 2. This bit is used in I3C protocol mode.
Note 3. This bit is used in I3C Secondary Controller and I3C Target modes.
Note 4. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.

ISRST Bit (Internal Status Reset)

When this bit is set to 1, some registers and internal circuits are reset. Refer to section 43.8, Reset Description for detail on the registers.

Set this bit to 0 to release from the reset.

43.2.5 Operating Mode Monitor Register (ICMMR)

Address(es): RI3C0.ICMMR 8745 F024h, RI3C0_NS.ICMMR 0745 F024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WP	—	—	TRS	—	ACF	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	ACF	Active Controller Flag*1, *2	• In I3C protocol mode 0: RI3C is not the Active Controller. 1: RI3C is the Active Controller. • In I2C protocol mode 0: Slave mode 1: Master mode	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	TRS	Transmit/Receive Mode Flag*3	0: Receive mode 1: Transmit mode	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	WP	Write Protect*1	0: Writing to the ACF flag is disabled. 1: Writing to the ACF flag is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C Primarily Controller, I3C Secondary Controller, and I2C protocol modes.
Note 2. When setting a value to the ACF flag, set the WP bit to 1 at the same time.
Note 3. This bit is used in I2C protocol mode.

ACF Flag (Active Controller Flag)

In I3C protocol mode:
This flag indicates whether the RI3C is presently the Active Controller.
[Setting conditions]

- When 1 is written to the WP bit and the ACF flag at the same time.
- When 1 is written to the ICCAR.DAV bit. (in I3C Primarily Controller mode)
- When GETACCCR reception is successfully completed by issuing STOP condition, after the ACK is responded to the Controller Role Request transmitted to the Secondary Controller. (in I3C Primarily Controller mode)
- When GETACCCR reception is successfully completed by issuing STOP condition, after the ACK is responded to the Controller Role Request transmitted to the Active Controller. (in I3C Secondary Controller mode)

[Clearing conditions]

- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When 1 is written to the WP bit and 0 is written to the ACF flag at the same time.
- When 0 written to the ICCAR.DAV bit. (in I3C Primarily Controller mode)
- When GETACCCR transmission is successfully completed by issuing STOP condition, after responding ACK to the Controller Role Request received from the Secondary Controller. (in I3C Primarily Controller mode)
- When GETACCCR transmission is successfully completed by issuing STOP condition, after responding ACK to

the Controller Role Request received from another Controller-capable device. (in I3C Secondary Controller mode)

In I^2C protocol mode:

This flag indicates master or slave mode.

The RI3C is in slave mode when the ACF flag is set to 0 and is in master mode when the flag is set to 1. Combination of this flag and the TRS flag indicates the operating mode of the RI3C.

The value of the ACF flag is automatically changed to 1 for master mode or 0 for slave mode by generating of a START condition and generating or detection of a STOP condition, etc. Although writing to the ACF flag is possible when the WP bit is set to 1, writing to this flag is not necessary during normal usage.

[Setting conditions]

- When 1 is written to the WP bit and the ACF flag at the same time.
- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).

[Clearing conditions]

- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.
- When 1 is written to the WP bit and 0 is written to the ACF flag at the same time.
- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).

TRS Flag (Transmit/Receive Mode Flag)

This flag indicates transmit or receive mode. It is used in I^2C protocol mode.

The RI3C is in receive mode when the TRS flag is set to 0 and is in transmit mode when the bit is set to 1. Combination of this flag and the ACF flag indicates the operating mode of the RI3C.

The value of TRS flag is automatically changed to 1 for transmit mode or 0 for receive mode by generating or detection of a START condition and setting of the R/W# bit.

[Setting conditions]

- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).
- When a repeated START condition is generated normally according to the repeated START condition generation request (when a repeated START condition is detected with the ICCGR.RS bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in the ICSCR register, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended.
- In slave mode, a match between the received address and the address enabled in the ICSCR register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a repeated START condition is detected (a START condition is detected with ICBSR.BFREE flag = 0 and ACF flag = 0).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to reset the RI3C or the internal status.

WP Bit (Write Protect)

This bit is used to rewrite the ACF flag.

When setting a value to the ACF flag, set this bit to 1 at the same time.

This bit is automatically set to 0. The read value is 0.

43.2.6 Internal Status Register (ICISR)

Address(es): RI3C0.ICISR 8745 F030h, RI3C0_NS.ICISR 0745 F030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERF	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERF	Buffer Access Error Flag	0: Buffer access error has not been detected. 1: Buffer access error was detected.	R/W*1
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.
Note 1. To clear this flag, confirm that the flag is 1 and then set it to 0.

BERF Flag (Buffer Access Error Flag)

This flag indicates that an access error to data buffers or queue buffers has been detected.

[Setting conditions]

When any of the following conditions is satisfied while the ICISER.BERDE bit is 1.

- When a transmit data is written while the transmit data buffer is completely full.
- When a received data is read while the receive data buffer is completely empty.
- When a Command Descriptor is written while the command queue is completely full.
- When a Response Descriptor is read while the response queue is completely empty.
- When a Receive Status Descriptor is read while the receive status queue is completely empty.
- When an IBI Status Descriptor is read while the ICMMR.ACF flag is 1 (Active Controller) and the IBI queue is completely empty.
- When an IBI data is written while the ICMMR.ACF flag is 0 (not Active Controller) and the IBI queue is completely full.
- When the response queue, IBI queue, or receive status queue overflows.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

43.2.7 Internal Status Detection Enable Register (ICISER)

Address(es): RI3C0.ICISER 8745 F034h, RI3C0_NS.ICISER 0745 F034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERDE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERDE	Buffer Access Error Detection Enable	0: Buffer access error detection is disabled. 1: Buffer access error detection is enabled.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.

43.2.8 Internal Status Interrupt Enable Register (ICISIER)

Address(es): RI3C0.ICISIER 8745 F038h, RI3C0_NS.ICISIER 0745 F038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERIE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERIE	Buffer Access Error Detection Interrupt Enable	0: Buffer access error detection interrupt is disabled. 1: Buffer access error detection interrupt is enabled.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.

BERIE Bit (Buffer Access Error Detection Interrupt Enable)

When the BERF flag becomes 1 while this bit is 1, an interrupt is generated.

43.2.9 Device Characteristics Table Index Register (ICDCTIR)

Address(es): RI3C0.ICDCTIR 8745 F044h, RI3C0_NS.ICDCTIR 0745 F044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	INDEX[4:0]					—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b18 to b0	—	Reserved	These bits are read as 0.	R
b23 to b19	INDEX[4:0]	DCT Table Index	Current index of the DCT, which is used as the starting index for the I3C ENTDAA CCC.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

INDEX[4:0] Bits (DCT Table Index)

Once the complete characteristics of device that won the arbitration are written to the Device Characteristics Table (DCT) during ENTDAA using address assignment command, this index is incremented by 1.

43.2.10 IBI Notify Control Register (ICINCR)

Address(es): RI3C0.ICINCR 8745 F058h, RI3C0_NS.ICINCR 0745 F058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RTIRN	—	RCRRN	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b1	RCRRN	Rejected Controller Role Request Notify	0: Do not pass the rejected IBI status to the IBI queue, if a Controller Role Request is rejected. 1: Pass the rejected IBI status to the IBI queue, if a Controller Role Request is rejected.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	RTIRN	Rejected Target Interrupt Request Notify	0: Do not pass the rejected IBI status to the IBI queue, if a Target Interrupt Request is rejected. 1: Pass the rejected IBI status to the IBI queue, if a Target Interrupt Request is rejected.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

RCRRN Bit (Rejected Controller Role Request Notify)

This bit enables or disables reporting rejection of individual Controller Role Requests (CRR), if the incoming CRR is NACKed and is auto-disabled based on the setting of the relevant ICTDATRm.CRRRJ bit (m = 0 to 7).

RTIRN Bit (Rejected Target Interrupt Request Notify)

This bit enables or disables reporting rejection of individual Target Interrupt Requests (TIR), if the incoming TIR is NACKed and is auto-disabled based on the setting of the relevant ICTDATRm.TIRRJ bit.

43.2.11 Function Enable Register (ICFER)

Address(es): RI3C0.ICFER 8745 F060h, RI3C0_NS.ICFER 0745 F060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HSME	FMPE	—	SMBS	—	—	—	SCLE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICMMR.ACF and ICMMR.TRS flags automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICMMR.ACF and ICMMR.TRS flags automatically when arbitration is lost.)	R/W
b1	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b2	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b11 to b9	—	Reserved	These bits are read as 0 and cannot be modified.	R
b12	SMBS	SMBus/I2C-bus Selection	0: The I2C-bus select 1: The SMBus select	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	FMPE	Fast-mode Plus Enable	0: No Fm+ slope control circuit is used for the SCL00 pin and SDA00 pin. 1: An Fm+ slope control circuit is used for the SCL00 pin and SDA00 pin.	R/W
b15	HSME	Hs-mode Enable	0: High speed mode is disabled. 1: High speed mode is enabled.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I2C protocol mode.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit uses to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit uses to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit uses to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

SCLE Bit (SCL Synchronization Enable)

This bit uses to specify whether to synchronize the SCL output with the SCL input. Normally, set this bit to 1.
When the SCLE bit set to 0 (SCL synchronization is disabled), the RI3C does not synchronize the SCL output with the SCL input. In this setting, the RI3C outputs the clock with the transfer rate set in the ICBRH and ICBRL registers regardless of the SCL00 line state. For this reason, if the load of the I^2C bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not, it also affects the generation of a START condition, repeated START condition, and STOP condition, and the continuous output of additional SCL.
This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE Bit (Fast-mode Plus Enable)

This bit uses to specify whether to use a slope control circuit for Fast-mode Plus (Fm+).
When this bit set to 1, a slope control circuit conforming to the Fast-mode Plus (Fm+) slope control standard (tof) of the I^2C-bus is selected. When this bit set to 0, a slope control circuit conforming to the Standard-mode (Sm) and Fast-mode (Fm) slope control standard (tof) of the I^2C-bus is selected.
Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus (Fm+)) of the I^2C-bus standard. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps (Sm), up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps). When performing Hs-mode communication, set this bit according to the mode when sending the Hs-mode master code.

HSME Bit (Hs-mode Enable)

This bit is used for communicating in Hs-mode.
When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.
After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from repeated START condition after receiving the NACK response.
It communicates at the bit rate set in the ICFBR register until the NACK response, and automatically switches from repeated START condition issuance after receiving the NACK response to the bit rate set in the ICHBR register.
Hs-mode continues until a STOP condition is detected.
When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in the ICFBR register.

Note: When this bit is set to 1, the ICSR2.NACKF flag will not be set even if a NACK response is received after sending the Hs-mode master code.

43.2.12 Target Mode Control Register (ICTCR)

Address(es): RI3C0.ICTCR 8745 F064h, RI3C0_NS.ICTCR 0745 F064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TA2DE	TA1DE	TA0DE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	TA0DE	Target Address 0 Detection Enable	0: The value of the ICDAMR0.TADR[9:0] bits is disabled. 1: The value of the ICDAMR0.TADR[9:0] bits is enabled.	R/W
b17	TA1DE	Target Address 1 Detection Enable	0: The value of the ICDAMR1.TADR[9:0] bits is disabled. 1: The value of the ICDAMR1.TADR[9:0] bits is enabled.	R/W
b18	TA2DE	Target Address 2 Detection Enable	0: The value of the ICDAMR2.TADR[9:0] bits is disabled. 1: The value of the ICDAMR2.TADR[9:0] bits is enabled.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

TAyDE Bit (Target Address y Detection Enable) (y = 0 to 2)

This bit is used to enable or disable the Target address set in ICDAMRy.TADR[9:0] bits.

When this bit is set to 1, the Target address set in ICDAMRy.TADR[9:0] bits is enabled and is compared with the received Target address.

When this bit is set to 0, the Target address set in ICDAMRy.TADR[9:0] bits is disabled and is ignored even if it matches the received Target address.

43.2.13 Slave Mode Control Register (ICSCR)

Address(es): RI3C0.ICSCR 8745 F064h, RI3C0_NS.ICSCR 0745 F064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	—	—	—	—	—	—	—	DIDE	HSMCE	—	—	—	—	GCAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GCAE	General Call Address Detection Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMCE	Hs-mode Master Code Detection Enable	0: Hs-mode master code detection is disabled. 1: Hs-mode master code detection is enabled.	R/W
b6	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	HOAE	Host Address Detection Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W
b16	SAR0E	Slave Address 0 Detection Enable	0: The value of the SAMR0.SVA[9:0] bits is disabled. 1: The value of the SAMR0.SVA[9:0] bits is enabled.	R/W
b17	SAR1E	Slave Address 1 Detection Enable	0: The value of the SAMR1.SVA[9:0] bits is disabled. 1: The value of the SAMR1.SVA[9:0] bits is enabled.	R/W
b18	SAR2E	Slave Address 2 Detection Enable	0: The value of the SAMR2.SVA[9:0] bits is disabled. 1: The value of the SAMR2.SVA[9:0] bits is enabled.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I²C protocol mode.

GCAE Bit (General Call Address Detection Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RI3C recognizes the received slave address as the general call address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE Bit (Hs-mode Master Code Detection Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (0000 1XXXb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, RI3C recognizes that the Hs-mode master code has been received.

The first byte after repeated START condition after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set in the SAMRy.SVA[9:0] bits (y = 0 to 2).

If the addresses match, the transmission/reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, RI3C will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode

master code.
When this bit is set to 1, set the ICCSCR.WAITAE bit to 0 and ICCSCR.WAITRE bit to 1.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a START condition or repeated START condition is detected.
When this bit is set to 1, if the received first frame matches the device ID, the RI3C recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RI3C recognizes the second and the following frames as slave addresses and continues the receive operation.
When this bit is set to 0, the RI3C ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.
For details on the device-ID address detection, refer to section 43.5.4.3, Device-ID Address Detection.

HOAE Bit (Host Address Detection Enable)

This bit uses to specify whether to ignore received host address (0001 000b) when the ICFER.SMBS bit is 1.
When this bit set to 1 while the ICFER.SMBS bit is 1, if the received slave address matches the host address, the RI3C recognizes the received slave address as the host address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs the receive operation.
When the ICFER.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SARyE Bit (Slave Address y Detection Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in SAMRy.SVA[9:0] bits.
When this bit is set to 1, the slave address set in SAMRy.SVA[9:0] bits is enabled and is compared with the received slave address.
When this bit is set to 0, the slave address set in SAMRy.SVA[9:0] bits is disabled and is ignored even if it matches the received slave address.

43.2.14 Reference Clock Control Register (ICRCCR)

Address(es): RI3C0.ICRCCR 8745 F070h, RI3C0_NS.ICRCCR 0745 F070h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Internal Reference Clock Select	Selects the internal reference clock source (IICφ) for the RI3C. b2 b0 0 0 0: I3CCLK/1 0 0 1: I3CCLK/2 0 1 0: I3CCLK/4 0 1 1: I3CCLK/8 1 0 0: I3CCLK/16 1 0 1: I3CCLK/32 1 1 0: I3CCLK/64 1 1 1: I3CCLK/128	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I^2C protocol mode.

43.2.15 Standard Bitrate Register (ICSBR)

Address(es): RI3C0.ICSBR 8745 F074h, RI3C0_NS.ICSBR 0745 F074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ODDBL	—	PPHW[5:0]						—	—	PPLW[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODHW[7:0]								ODLW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ODLW[7:0]	Open-Drain Low Width Setting	Count value for the low period of SCL in Open-Drain mode.	R/W
b15 to b8	ODHW[7:0]	Open-Drain High Width Setting	Count value for the high period of SCL in Open-Drain mode.	R/W
b21 to b16	PPLW[5:0]	Push-Pull Low Width Setting	Count value for the low period of SCL in Push-Pull mode.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	PPHW[5:0]	Push-Pull High Width Setting	Count value for the high period of SCL in Push-Pull mode.	R/W
b30	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	ODDBL	Open-Drain High/Low Width Doubling	0: Set the high and low widths to the set value in Open-Drain mode. 1: Double the high and low widths of the set values in Open-Drain mode.	R/W

Note 1. This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to set the bit rate when the I3C SDR0 mode is specified in the MODE field of the Command Descriptor. When the I3C SDR2 mode is specified, a bit rate of 1/2 the bit rate set in this register is used.

ODLW[7:0] Bits (Open-Drain Low Width Setting)

The ODLW[7:0] bits are used to set the low period of SCL in Open-Drain mode.
The RI3C counts the low period with I3CCLK.

ODHW[7:0] Bits (Open-Drain High Width Setting)

The ODHW[7:0] bits are used to set the high period of SCL in Open-Drain mode.
These bits are valid in master/Controller mode.
The RI3C counts the high period with I3CCLK.

PPLW[5:0] Bits (Push-Pull Low Width Setting)

The PPLW[5:0] bits are used to set the low period of SCL in Push-Pull mode.
RI3C counts the low period with I3CCLK.

PPHW[5:0] Bits (Push-Pull High Width Setting)

The PPHW[5:0] bits are used to set the high period of SCL in Push-Pull mode.
These bits are valid in Controller mode.
RI3C counts the high period with I3CCLK.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = ODLW[7:0] / I3CCLK
t_{HIGH} = ODHW[7:0] / I3CCLK
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I3C bus standard from MIPI Alliance.

43.2.16 F/S-Mode Bitrate Register (ICFBR)

Address(es): RI3C0.ICFBR 8745 F074h, RI3C0_NS.ICFBR 0745 F074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DBL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HIGH[7:0]								LOW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b21 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	DBL	High/Low Width Doubling	0: Set the high and low widths to the set value. 1: Double the high and low widths of the set values.	R/W

Note: This register is used in I^2C protocol mode.

This register is used to specify the bit rate other than Hs-mode.

LOW[7:0] Bits (Low Width Setting)

The LOW[7:0] bits are used to set the low period of SCL.

The RI3C counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits. These bits also work to generate the data setup time for automatic SCL low-hold operation (refer to section 43.5.5, Automatic Low-Hold Function for SCL (I^2C Protocol Mode)); when the RI3C is used in slave mode, these bits need to be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time ($t_{SU;DAT}$)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))
50 ns (up to 1 Mbps: Fast-mode Plus (Fm+))

HIGH[7:0] Bits (High Width Setting)

The HIGH[7:0] bits are used to set the high period of SCL.

These bits are valid in master/Controller mode. If the RI3C is used only in I^2C slave mode, these bits need not to set the high period.

The RI3C counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = (LOW[7:0] + NF[3:0] +1) / IICφ
t_{HIGH} = (HIGH[7:0] + NF[3:0] +1) / IICφ
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I²C-bus standard from NXP Semiconductors.

Table 43.4 Example of the ICFBR Register Settings for Transfer Rate (I3CCLK = 120 MHz)

Transfer Rate (kbps)	CKS[2:0] Bits	HIGH[7:0] Bits	LOW[7:0] Bits	NFE Bit	NF[3:0] Bits
10	101b	122 (7Ah)	247 (F7h)	0	—
	101b	121 (79h)	246 (F6h)	1	0001b
50	011b	91 (5Bh)	188 (BCh)	0	—
	011b	90 (5Ah)	187 (BBh)	1	0001b
100	010b	108 (6Ch)	151 (97h)	0	—
	010b	107 (6Bh)	150 (96h)	1	0001b
400	000b	68 (44h)	158 (9Eh)	0	—
	000b	67 (43h)	157 (9Dh)	1	0001b
	001b	34 (22h)	78 (4Eh)	0	—
	001b	33 (21h)	77 (4Dh)	1	0001b
1000	000b	27 (1Bh)	63 (3Fh)	0	—
	000b	26 (1Ah)	62 (3Eh)	1	0001b
	001b	13 (0Dh)	31 (1Fh)	0	—
	001b	12 (0Ch)	30 (1Eh)	1	0001b

Rise time of SCL signal (t_r)
- 100 kbps or less (Sm): 1000 ns
- 400 kbps (Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

Fall time of SCL signal (t_f)
- 400 kbps (Sm/Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

43.2.17 Extended Bitrate Register (ICEBR)

Address(es): RI3C0.ICEBR 8745 F078h, RI3C0_NS.ICEBR 0745 F078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	PPHW[5:0]						—	—	PPLW[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODHW[7:0]								ODLW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ODLW[7:0]	Open-Drain Low Width Setting	Count value for the low period of SCL in Open-Drain mode.	R/W
b15 to b8	ODHW[7:0]	Open-Drain High Width Setting	Count value for the high period of SCL in Open-Drain mode.	R/W
b21 to b16	PPLW[5:0]	Push-Pull Low Width Setting	Count value for the low period of SCL in Push-Pull mode.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	PPHW[5:0]	Push-Pull High Width Setting	Count value for the high period of SCL in Push-Pull mode.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to set the bit rate when the I3C SDR1 mode is specified in the MODE field of the Command Descriptor. When the I3C SDR3 mode is specified, a bit rate of 1/2 the bit rate set in this register is used. When the I3C SDR4 mode is specified, a bit rate of 1/4 the bit rate set in this register is used.

ODLW[7:0] Bits (Open-Drain Low Width Setting)

The ODLW[7:0] bits are used to set the low period of SCL in Open-Drain mode.
The RI3C counts the low period with I3CCLK.

ODHW[7:0] Bits (Open-Drain High Width Setting)

The ODHW[7:0] bits are used to set the high period of SCL in Open-Drain mode.
These bits are valid in master/Controller mode.
The RI3C counts the high period with I3CCLK.

PPLW[5:0] Bits (Push-Pull Low Width Setting)

The PPLW[5:0] bits are used to set the low period of SCL in Push-Pull mode.
The RI3C counts the low period with the I3CCLK.

PPHW[5:0] Bits (Push-Pull High Width Setting)

The PPHW[5:0] bits are used to set the high period of SCL in Push-Pull mode.
These bits are valid in Controller mode.
The RI3C counts the high period with the I3CCLK.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = ODLW[7:0] / I3CCLK
t_{HIGH} = ODHW[7:0] / I3CCLK
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I3C bus standard from MIPI Alliance.

43.2.18 Hs-Mode Bitrate Register (ICHBR)

Address(es): RI3C0.ICHBR 8745 F078h, RI3C0_NS.ICHBR 0745 F078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HIGH[7:0]								LOW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b21 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I2C protocol mode.

This register is used to specify the bit rate in Hs-mode.

LOW[7:0] Bits (Low Width Setting)

The LOW[7:0] bits are used to set the low period of SCL.
The RI3C counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits. These bits also work to generate the data setup time for automatic SCL low-hold operation (refer to section 43.5.5, Automatic Low-Hold Function for SCL (I2C Protocol Mode)); when the RI3C is used in slave mode, these bits need to be set to a value longer than the data setup time*1.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time ($t_{SU;DAT}$)
10 ns (up to 3.4 Mbps: Hs-mode (Hs))

HIGH[7:0] Bits (High Width Setting)

The HIGH[7:0] bits are used to set the high period of SCL.
These bits are valid in master/Controller mode. If the RI3C is used only in I2C slave mode, these bits need not to set the high period.
The RI3C counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = (LOW[7:0] + NF[3:2] +1) / IICφ
t_{HIGH} = (HIGH[7:0] + NF[3:2] +1) / IICφ
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I2C-bus standard from NXP Semiconductors.

43.2.19 Bus Free Time Setting Register (ICBFTR)

Address(es): RI3C0.ICBFTR 8745 F07Ch, RI3C0_NS.ICBFTR 0745 F07Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BFREE flag is set to 1. The upper 23 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using IICφ (in I2C protocol mode) or I3CCLK (in I3C protocol mode) as the count source.
Set a value that satisfies the t_{BUF} (bus free time between a STOP and START condition) specified in the I2C-bus specification.

43.2.20 Bus Available Time Setting Register (ICBATR)

Address(es): RI3C0.ICBATR 8745 F080h, RI3C0_NS.ICBATR 0745 F080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C protocol mode.

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BAVL flag is set to 1.
The upper 23 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using I3CCLK as the count source.

43.2.21 Bus Idle Time Setting Register (ICBITR)

Address(es): RI3C0.ICBITR 8745 F084h, RI3C0_NS.ICBITR 0745 F084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C protocol mode.

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BIDL flag is set to 1.
The upper 14 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using I3CCLK as the count source.

43.2.22 Output Signal Control Register (ICOCR)

Address(es): RI3C0.ICOCR 8745 F088h, RI3C0_NS.ICOCR 0745 F088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	—	—	—	—	SDDL[2:0]			—	—	—	CLO	—	SOWP	SCLO	SDAO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAO	SDA Output Control*1	0: The RI3C drives the SDA00 pin low. 1: The RI3C releases the SDA00 pin.	R/W
b1	SCLO	SCL Output Control*1	0: The RI3C drives the SCL00 pin low. 1: The RI3C releases the SCL00 pin.	R/W
b2	SOWP	SCLO/SDAO Write Protect*1	0: Writing to the SCLO and SDAO bits is disabled. 1: Writing to the SCLO and SDAO bits is enabled (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	CLO	Additional SCL Output*2	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (This bit automatically becomes 0 after output of one clock pulse.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	SDDL[2:0]	SDA Output Delay Setting*2	• When ICOCR.DLCS bit is 0 (IICφ) b10 b8 0 0 0: No output delay 0 0 1: 1 IICφ cycle 0 1 0: 2 IICφ cycles 0 1 1: 3 IICφ cycles 1 0 0: 4 IICφ cycles 1 0 1: 5 IICφ cycles 1 1 0: 6 IICφ cycles 1 1 1: 7 IICφ cycles • When ICOCR.DLCS bit is 1 (IICφ/2) b10 b8 0 0 0: No output delay 0 0 1: 1 or 2 IICφ cycles 0 1 0: 3 or 4 IICφ cycles 0 1 1: 5 or 6 IICφ cycles 1 0 0: 7 or 8 IICφ cycles 1 0 1: 9 or 10 IICφ cycles 1 1 0: 11 or 12 IICφ cycles 1 1 1: 13 or 14 IICφ cycles	R/W
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	DLCS	SDA Output Delay Clock Source Select*2	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*3	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C Primarily Controller, I3C Secondary Controller, and I²C protocol modes.
Note 2. This bit is used in I²C protocol mode.
Note 3. This setting (IICφ/2) is valid only when the SCL pin is low. When the SCL pin is high, this setting is invalid and the clock source becomes the internal reference clock (IICφ).

SDAO Bit (SDA Output Control) and SCLO Bit (SCL Output Control)

These bits are used to directly control the SDA00 and SCL00 signals output from the RI3C.
When writing to these bits, also write 1 to the SOWP bit at the same time.
The result of setting these bits is input to the RI3C via the input buffer. When slave/Target mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.
Do not rewrite these bits during a START condition, STOP condition, repeated START condition, transmission, or reception. Operation after rewriting under the above conditions is not guaranteed.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 43.5.8.2, Additional SCL Output Function.

43.2.23 Input Signal Control Register (ICICR)

Address(es): RI3C0.ICICR 8745 F08Ch, RI3C0_NS.ICICR 0745 F08Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ANFE	NFE	NF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	NF[3:0]	Noise Filter Stage Select	• Except in Hs-mode b3 b0 0 0 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 0 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 0 0 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). : : 1 1 1 1: Noise of up to 16 IICφ cycles is filtered out (16-stage filter). • In Hs-mode b3 b2 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to 4 IICφ cycles is filtered out (4-stage filter). The lower 2 bits (b1 and b0) are ignored.	R/W
b4	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b5	ANFE	Analog Noise Filter Enable	0: Analog noise filters are not used. 1: Analog noise filters are used.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I²C protocol mode.

NF[3:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.
For details on the digital noise filter function, refer to section 43.5.3, Digital Noise Filters (I²C Protocol Mode).

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL00 line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – {1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) + 120 ns (pulse width suppressed by the analog noise filter, reference values)} or a greater value, the serial clock is regarded as noise by the noise filter function of the RI3C, which may prevent the RI3C from operating normally.

43.2.24 Timeout Control Register (ICTOR)

Address(es): RI3C0.ICTOR 8745 F090h, RI3C0_NS.ICTOR 0745 F090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TMOM[1:0]		TMOH	TMOL	—	—	TMOS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMOS[1:0]	Timeout Detection Time Select	b1 b0 0 0: Timeout counter is 16 bits (up to 65536 counts) 0 1: Timeout counter is 14 bits (up to 16384 counts) 1 0: Timeout counter is 8 bits (up to 256 counts) 1 1: Timeout counter is 6 bits (up to 64 counts)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TMOL	Timeout L Count Control	0: Count is disabled while the SCL00 line is low. 1: Count is enabled while the SCL00 line is low.	R/W
b5	TMOH	Timeout H Count Control	0: Count is disabled while the SCL00 line is high. 1: Count is enabled while the SCL00 line is high.	R/W
b7, b6	TMOM[1:0]	Timeout Detection Mode Select	b7 b6 0 0: Timeout function is enabled in any of the following conditions. • When the ICBSR.BFREE flag is 0 (bus busy) in master/Controller mode • When the ICBSR.BFREE flag is 0 (bus busy) when the received slave/Target address matches its own slave/Target address in slave/Target mode • When the ICBSR.BFREE flag is 1 (bus free) while the ICCGR.ST is 1 (request to generate a START condition) 0 1: Timeout function is enabled during ICBSR.BFREE flag is 0 (bus busy). 1 0: Timeout function is enabled during ICBSR.BFREE flag is 1 (bus free). 1 1: Setting prohibited	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TMOS[1:0] Bits (Timeout Detection Time Select)

These bits are used to select for the timeout detection time when the timeout function is enabled (ICSER.TMOE bit = 1). While the SCL00 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) or I3CCLK as a count source.

For details on the timeout function, refer to section 43.5.8.1, Timeout Function or 43.5.16.3 Timeout Detection in section 43.5.16, Error Detection (I3C Protocol Mode).

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL00 line is held low when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL00 line is held high when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOM[1:0] Bits (Timeout Detection Mode Select)

These bits are used to set the timeout detection condition when the timeout function is enabled (ICSER.TMOE bit = 1).
When these bits are set to 00b, timeout detection is enabled during one of the following periods.

- When the bus is busy in master/Controller mode
- When RI3C is the target of the communication in slave/Target mode and the bus is busy
- When the bus is free while a START condition is requested

When these bits are set to 01b, timeout detection is enabled when the bus is busy.
When these bits are set to 10b, timeout detection is enabled when the bus is free.

43.2.25 Wake-Up Unit Control Register (ICWUCR)

Address(es): RI3C0.ICWUCR 8745 F098h, RI3C0_NS.ICWUCR 0745 F098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WUFE	WUFSYNE	—	WUANFE	—	—	—	WUACKS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	WUACKS	Wake-Up Acknowledge Select*1	The response mode can be selected in combination with the ICRCR.ISRST bit. Refer to Table 43.5 for details.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	WUANFE	Analog Noise Filter Enable*1	0: Analog noise filter for wake-up is disabled. 1: Analog noise filter for wake-up is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	WUFSYNE	Synchronous Operation Select	0: Asynchronous operation 1: Internal clock synchronous operation	R/W
b7	WUFE	Wake-Up Function Enable	0: Wake-Up function is disabled. 1: Wake-Up function is enabled.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I2C protocol mode.

WUACKS Bit (Wake-Up Acknowledge Select)

This bit selects the ACK response when a wake-up condition is detected. This bit is used in I2C protocol mode.

Table 43.5 Response Mode

ICRCR.ISRST Bit	WUACKS Bit	Response Mode	Description
0	0	Normal wake-up mode 1	Responds with an ACK and holds the SCL line low.
0	1	Normal wake-up mode 2*1	Holds the SCL line low after eighth clock pulse, and when the low power consumption mode is released, release the SCL line and responds with an ACK.
1	0	Command recovery mode	Responds with an ACK but does not hold the SCL line low.
1	1	EEP response mode	Responds with a NACK but does not hold the SCL line low.

Note 1. This mode cannot be used in Hs mode.

WUANFE Bit (Analog Noise Filter Enable)

This bit enables/disables the analog noise filter for the SCL and SDA input signals. This bit is used in I2C protocol mode.

WUFSYNE Bit (Synchronous Operation Select)

This bit selects whether to synchronize the SCL and SDA lines with the internal clock or to use them as asynchronous. Asynchronous operation can be selected only when the WUFE bit is 1.

When this bit is set to 0, the ICWUSR.WUASYNF flag becomes 1 when bus-free is detected and the wake-up detection circuit is enabled. At this time, data reception is performed asynchronously with the internal clock.

When this bit is set to 1 after the wake-up condition is detected, the wake-up detection circuit is disabled and the normal reception circuit operates. If this bit is set to 1 when the wake-up condition is not detected, the wake-up detection circuit

is disabled when a STOP condition is detected.

43.2.26 Acknowledgment Bit Control Register (ICACKR)

Address(es): RI3C0.ICACKR 8745 F0A0h, RI3C0_NS.ICACKR 0745 F0A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACKWP	ACKBT	ACKBR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b1	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b2	ACKWP	ACKBT Write Protect	0: Writing to the ACKBT bit is disabled. 1: Writing to the ACKBT bit is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I²C protocol mode.
Note 1. When setting a value to the ACKBT bit, set the ACKWP bit to 1 at the same time

ACKBR Bit (Received Acknowledge)

This bit is used to store the acknowledgment bit received from the receiver in transmit mode.
[Setting condition]

- When 1 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

ACKBT Bit (Transmit Acknowledge)

[Setting condition]

- When 1 is written to the ACKWP and ACKBT bits at the same time.

[Clearing conditions]

- When 1 is written to the ACKWP bit and 0 is written to the ACKBT bit at the same time.
- When a STOP condition generation is detected. (When a STOP condition is detected with the ICCGR.SP bit set to 1.)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.
When setting a value to the ACKBT bit, set this bit to 1 at the same time.
This bit is automatically set to 0. The read value is 0.

43.2.27 Clock Stretch Control Register (ICCSCR)

Address(es): RI3C0.ICCSCR 8745 F0A4h, RI3C0_NS.ICCSCR 0745 F0A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WAITRE	WAITAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WAITAE	Acknowledgment Generation Wait Enable	0: The ICCSR.RDRF flag becomes 1 at the rising edge of the ninth SCL. (The SCL00 line is not held low at the falling edge of the eighth clock pulse.) 1: The ICCSR.RDRF flag becomes 1 at the rising edge of the eighth SCL. (The SCL00 line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ICACKR.ACKBT bit.	R/W
b1	WAITRE	Receive Data Read Wait Enable	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading ICDR register.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I²C protocol mode.

WAITAE Bit (Acknowledgment Generation Wait Enable)

This bit is used to select the ICCSR.RDRF flag set timing in receive mode and also to select whether to hold the SCL00 line low at the falling edge of the eighth SCL.

When the WAITAE bit is 0, the SCL00 line is not held low at the falling edge of the eighth SCL, and the ICCSR.RDRF flag is set to 1 at the rising edge of the ninth SCL.

When the WAITAE bit is 1, the ICCSR.RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCL00 line is held low at the falling edge of the eighth SCL. The low-hold of the SCL00 line is released by writing a value to the ICACKR.ACKBT bit. After data is received with this setting, the SCL00 line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ICACKR.ACKBT bit = 0) or NACK (ICACKR.ACKBT bit = 1) according to receive data.

WAITRE Bit (Receive Data Read Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the receive data buffer (ICDR register) is completely read each time single-byte data is received in receive mode.

When the WAITRE bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL low. When both the WAITAE and WAITRE bits are 0, continuous receive operation is enabled with the double buffer.

When the WAITRE bit is 1, the SCL00 line is held low from the falling edge of the ninth SCL until the ICDR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: Read the ICDR register before setting the WAITRE bit to 0.

43.2.28 Clock Stall Control Register (ICSTCR)

Address(es): RI3C0.ICSTCR 8745 F0B0h, RI3C0_NS.ICSTCR 0745 F0B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	APSE	PBSE	—	AASE	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	STT[15:0]	Stall Time Setting	Set the clock stall time for each phase.	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	AASE	Assigned Address Phase Stall Enable	0: Do not stall SCL before the Assigned Address phase. 1: Stall SCL before the Assigned Address phase.	R/W
b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	PBSE	Parity Bit Stall Enable	0: Does not stall the SCL during the parity bit period. 1: Stall the SCL during the parity bit period.	R/W
b31	APSE	ACK/NACK Phase Stall Enable	0: Does not stall the SCL during the ACK/NACK phase. 1: Stall the SCL during the ACK/NACK phase.	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

When setting this register, follow section 6.10, Controller Clock Stalling of MIPI I3C Specification Version 1.2, and use it only when necessary because of its negative impacts on bus performance.

STT[15:0] Bits (Stall Time Setting)

These bits is used to set the SCL stall time. The SCL stall time is counted by I3CCLK. This is a counter common to the enable bits of each phase.

AASE Bit (Assigned Address Phase Stall Enable)

This bit is used to select whether the Controller stalls SCL during the low period of the first bit of the Assigned Address phase of the ENTDAA CCC command. It can gain time in assigning Dynamic Address to the Device based on the BCR and DCR of the Target. However, because the Dynamic Address Assignment procedure sends the Dynamic Address set in the ICTDATRm register in sequence, it is not necessary to set this bit and it is prohibited.

PBSE Bit (Parity Bit Stall Enable)

This bit is used to select whether the Controller stalls SCL at the parity bit of the transmit data of I3C write transfer to avoid underrun of the transmit FIFO. However, when the transmit FIFO of the I3C Controller becomes empty, SCL stalling is performed regardless of the setting of this bit, it is not necessary to set this bit and it is prohibited.

APSE Bit (ACK/NACK Phase Stall Enable)

Determine the need to perform SCL stalling during the ACK/NACK phase based on the following criteria:

- It is necessary to set this bit when the I3C and I^2C Targets connected to the bus require preparation time to receive or transmit data.
- In legacy I^2C communication, if there is a possibility that the FIFO of the I3C Controller might underrun or overrun. It is not necessary to set this bit because SCL stalling is performed by FIFO Empty or Full regardless of the setting

of this bit.

- Other than legacy I2C communication, the FIFO of I3C Controller might underrun or overrun, and if SCL stalling is required in ACK phase, this bit can be set. However, it is necessary to build the software so that the FIFO does not underrun or overrun due to the interrupt generated according to the FIFO threshold setting (ICQBTCR, ICDBTCR, ICSQTCR).
- When I3C Controller responds ACK/NACK to IBI, it is not necessary to set this bit because ACK/NACK response can be set in advance by ICTDATRm.CRRRJ and ICTDATRm.TIRRJ.
- It is necessary to set this bit when the I3C Target connected to the bus requires preparation time to transmit data for Direct GET CCC.

43.2.29 Target Transmit/Receive Data Length Register (ICTDLR)

Address(es): RI3C0.ICTDLR 8745 F0C0h, RI3C0_NS.ICTDLR 0745 F0C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b16	DLGTH[15:0]	Data Length	Indicates the number of bytes to be transferred.	R/W

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to set the number of bytes of data to be transmitted with the regular data transfer command.

43.2.30 Synchronous Timing Control Register (ICSYTCR)

Address(es): RI3C0.ICSYTCR 8745 F120h, RI3C0_NS.ICSYTCR 0745 F120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STOE	Synchronous Timing Event Output Enable	0: Synchronous timing event output is disabled. 1: Synchronous timing event output is enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.

43.2.31 Asynchronous Timing Control Register (ICASTCR)

Address(es): RI3C0.ICASTCR 8745 F124h, RI3C0_NS.ICASTCR 0745 F124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CDIV[7:0]								—	—	—	—	—	ABEOE	CREFOE	ATTRGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ATTRGS	Synchronous Timing Trigger Select*1	0: Software trigger 1: T_C1 capture event input	R/W
b1	CREFOE	C_REF Event Output Enable*2	0: C_REF event output is disabled. 1: C_REF event output is enabled.	R/W
b2	ABEOE	Additional Bus Event Output Enable*2	0: Additional bus event output is disabled. 1: Additional bus event output is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15 to b8	CDIV[7:0]	I3CCLK Divide Setting	The count clock for the asynchronous timing counters (C_REF, C_C2, T_C1, and T_C2) is generated by dividing I3CCLK. Set the division value of I3CCLK in these bits.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.
Note 1. This bit is used in I3C Secondary Controller and I3C Target modes.
Note 2. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.32 Asynchronous Timing Trigger Register (ICASTTR)

Address(es): RI3C0.ICASTTR 8745 F128h, RI3C0_NS.ICASTTR 0745 F128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ATSTRG	Software Trigger	Writing 1 to this bit generates a software trigger signal.	W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

43.2.33 Asynchronous Timing Counter Control Register (ICASTCCR)

Address(es): RI3C0.ICASTCCR 8745 F12Ch, RI3C0_NS.ICASTCCR 0745 F12Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ATCE	Asynchronous Timing Counter Operation Enable	0: Asynchronous timing counters are disabled. 1: Asynchronous timing counters are enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.

43.2.34 Condition Generation Request Register (ICCGR)

Address(es): RI3C0.ICCGR 8745 F140h, RI3C0_NS.ICCGR 0745 F140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SP	RS	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ST	START Condition Generation Request	0: Do not request to generate a START condition. 1: Requests to generate a START condition.	R/W
b1	RS	Repeated START Condition Generation Request	0: Do not request to generate a repeated START condition. 1: Requests to generate a repeated START condition.	R/W
b2	SP	STOP Condition Generation Request	0: Do not request to generate a STOP condition. 1: Requests to generate a STOP condition.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I2C protocol mode.

ST Bit (START Condition Generation Request)

This bit is used to request transition to master mode and generation of a START condition.

For details on the START condition generation, refer to section 43.5.7, START Condition/Repeated START Condition/STOP Condition Generating Function (I2C Protocol Mode).

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit.
- When a START condition has been generated (a START condition is detected).
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RI3C reset or an internal reset.

Note: Set the ST bit to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 1 (bus free state).
Note that arbitration may be lost if the ST bit is set to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 0 (bus busy state).

RS Bit (Repeated START Condition Generation Request)

This bit is used to request that a repeated START condition be generated in master mode.

When this bit is set to 1 to request to generate a repeated START condition, a repeated START condition is generated when the BFREE flag is set to 0 (bus busy state) and the ICMMR.ACF flag is set to 1 (master mode).

For details on the repeated START condition generation, refer to section 43.5.7, START Condition/Repeated START Condition/STOP Condition Generating Function (I2C Protocol Mode).

[Setting condition]

- When 1 is written to the RS bit with the ICBSR.BFREE flag set to 0.

[Clearing conditions]

- When 0 is written to the RS bit.
- When a repeated START condition has been generated. (a repeated START condition is detected.)
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RI3C reset or an internal reset.

Note: Do not set the RS bit to 1 while generating a STOP condition.
Note: If 1 (requests to generate a repeated START condition) is written to the RS bit in slave mode, the repeated START condition is not generated but the RS bit remains set to 1.
If the operating mode changes to master mode with the bit not being cleared, note that the repeated START condition may be generated.

SP Bit (STOP Condition Generation Request)

This bit is used to request that a STOP condition be generated in master mode.
When this bit is set to 1 to request to generate a STOP condition, a STOP condition is generated when the ICBSR.BFREE flag is set to 0 (bus busy state) and the ICMMR.ACF flag is set to 1 (master mode).
For details on the STOP condition generation, refer to section 43.5.7, START Condition/Repeated START Condition/STOP Condition Generating Function (I^2C Protocol Mode).
[Setting condition]

- When 1 is written to the SP bit with the ICBSR.BFREE flag set to 0 and the ICMMR.ACF flag set to 1.

[Clearing conditions]

- When 0 is written to the SP bit.
- When a STOP condition has been generated. (a STOP condition is detected.)
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When a START condition and a repeated START condition are detected.
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply an RI3C reset or an internal reset.

Note: When the ICBSR.BFREE flag is 1 (bus free state), writing 1 to the SP bit is ignored.
Note: Do not set the SP bit to 1 while a repeated START condition is being generated.

43.2.35 Command Queue Register (ICCQR)

Address(es): RI3C0.ICCQR 8745 F150h, RI3C0_NS.ICCQR 0745 F150h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C protocol mode.

The ICCQR register is a 32-bit write-only register used to enqueue Command Descriptors into the command queue. Command Descriptors should be written in order from lower (b31-b0) to higher (b63-b32).
Refer to section 43.3.1, Command Descriptor for details on Command Descriptors.

43.2.36 Response Queue Register (ICRQR)

Address(es): RI3C0.ICRQR 8745 F154h, RI3C0_NS.ICRQR 0745 F154h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C protocol mode.

The ICRQR register is a 32-bit read-only register used to read the Response Descriptors generated in the response queue after a command has been processed.
Refer to section 43.3.2, Response Descriptor for details on Response Descriptors.

43.2.37 Transmit/Receive Data Register (ICDR)

(1) I3C Protocol Mode

Address(es): RI3C0.ICDR 8745 F158h, RI3C0_NS.ICDR 0745 F158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICDR register is a 32-bit read/write register used to read data from the receive data buffer and write data to the transmit data buffer.

For Read Operation:

Receive data is stored in the receive data buffer every 4 bytes. If the number of data received is not an integer multiple of 4 bytes, the last 32 bits of the stored data will contain invalid bytes. Check the number of valid data bytes in the DATA_LENGTH field of the Response Descriptor.

For Write Operation:

Transmit data must be written to the ICDR register in 4-byte units. If the number of data to be transmitted is not an integer multiple of 4 bytes, add dummy data after the final data to make it 4 bytes before writing to the ICDR register. RI3C sends the number of bytes of data specified in the DATA_LENGTH field of the Command Descriptor.

(2) I²C Protocol Mode

Address(es): RI3C0.ICDR 8745 F158h, RI3C0_NS.ICDR 0745 F158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICDR register is a 32-bit read/write register used to read data from the receive data buffer and write data to the transmit data buffer. This register can be accessed in 32-bit or 8-bit units. The upper 24 bits are reserved and should be written as 0 when data is written in 32-bit units.

For Read Operation:

When 1 byte of data has been received, the received data is transferred from the internal shift register to the ICDR register to enable the next data to be received.

The double-buffer structure of the shift register and the ICDR register allows continuous receive operation if the received data has been read from the ICDR register while the shift register is receiving data.

Read data from the ICDR register once when a receive data full interrupt (RXI) request is generated.

If the ICDR register receives the next receive data before the current data is read from the ICDR register (while the ICCSR.RDRF flag is 1), the RI3C automatically holds the SCL00 line low one cycle before the RDRF flag is set to 1 next.

For Write Operation:

When the ICDR register detects a space in the internal shift register, it transfers the transmit data that has been written to the ICDR register to the shift register and starts transmitting data in transmit mode.

The double-buffer structure of the ICDR register and the shift register allows continuous transmit operation if the next transmit data has been written to the ICDR register while the shift register data is being transmitted.

Write transmit data to the ICDR register once when a transmit data empty interrupt (TXI) request is generated.

43.2.38 IBI Queue Register (ICIQR)

Address(es): RI3C0.ICIQR 8745 F17Ch, RI3C0_NS.ICIQR 0745 F17Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C protocol mode.

The ICIQR register is a 32-bit read-only register used to read an IBI Status Descriptor and IBI data from the IBI queue after an IBI has been received.
Refer to section 43.3.3, IBI Status Descriptor for details on IBI Status Descriptors.

43.2.39 Receive Status Queue Register (ICSQR)

Address(es): RI3C0.ICSQR 8745 F180h, RI3C0_NS.ICSQR 0745 F180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The ICSQR register is a 32-bit read-only register used to read an Receive Status Descriptor from the receive status queue.
Refer to section 43.3.4, Receive Status Descriptor for details on Receive Status Descriptors.

43.2.40 High-Priority Command Queue Register (ICHCQR)

Address(es): RI3C0.ICHCQR 8745 F184h, RI3C0_NS.ICHCQR 0745 F184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICHCQR register is a 32-bit write-only register used to enqueue Command Descriptors into the high-priority command queue. Command Descriptors should be written in order from lower (b31-b0) to higher (b63-b32).
Refer to section 43.3.1, Command Descriptor for details on Command Descriptors.

43.2.41 High-Priority Response Queue Register (ICHRQR)

Address(es): RI3C0.ICHRQR 8745 F188h, RI3C0_NS.ICHRQR 0745 F188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICHRQR register is a 32-bit read-only register used to read the Response Descriptors generated in the high-priority response queue after a high-priority command has been processed.
Refer to section 43.3.2, Response Descriptor for details on Response Descriptors.

43.2.42 High-Priority Transmit/Receive Data Register (ICHDR)

Address(es): RI3C0.ICHDR 8745 F18Ch, RI3C0_NS.ICHDR 0745 F18Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICHDR register is a 32-bit read/write register used to read data from the high-priority receive data buffer and write data to the high-priority transmit data buffer.

For Read Operation:

Receive data is stored in the high-priority receive data buffer every 4 bytes. If the number of data received is not an integer multiple of 4 bytes, the last 32 bits of the stored data will contain invalid bytes. Check the number of valid data bytes in the DATA_LENGTH field of the Response Descriptor.

For Write Operation:

High-priority transmit data must be written to the ICHDR register in 4-byte units. If the number of data to be transmitted is not an integer multiple of 4 bytes, add dummy data after the final data to make it 4 bytes before writing to the ICHDR register. RI3C sends the number of bytes of data specified in the DATA_LENGTH field of the Command Descriptor.

43.2.43 Queue Buffer Threshold Control Register (ICQBTCR)

Address(es): RI3C0.ICQBTCR 8745 F190h, RI3C0_NS.ICQBTCR 0745 F190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IQTH[7:0]								IDSS[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RQTH[7:0]								CQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CQTH[7:0]	Command Queue Empty Interrupt Threshold Setting	00h: Interrupt is generated when command queue is completely empty. 01h: Interrupt is generated when command queue has one free entry. 02h: Interrupt is generated when command queue has two free entry. 03h: Interrupt is generated when command queue has three free entry. Settings other than above are prohibited.	R/W
b15 to b8	RQTH[7:0]	Response Queue Full Interrupt Threshold Setting	00h: Interrupt is generated when response queue contains one entry (4 bytes). 01h: Interrupt is generated when response queue contains two entries (8 bytes). 02h: Interrupt is generated when response queue contains three entries (12 bytes). 03h: Interrupt is generated when response queue contains four entries (16 bytes). Settings other than above are prohibited.	R/W
b23 to b16	IDSS[7:0]	IBI Data Segment Size Setting*1	Set the maximum size of IBI data segments in 4-byte units. Setting range: 1 (4 bytes) to 7 (28 bytes)	R/W
b31 to b24	IQTH[7:0]	IBI Queue Empty/Full Interrupt Threshold Setting	• I3C Controller mode: 00h: Interrupt is generated when there is at least one outstanding IBI status. 01h: Interrupt is generated when there are two outstanding IBI status. Settings other than above are prohibited. • I3C Target mode: 00h: Interrupt is generated when IBI queue is completely empty. 01h: Interrupt is generated when IBI queue has one free entry. : 07h: Interrupt is generated when IBI queue has seven free entries. Settings other than above are prohibited.	R/W

Note: This register is used in I3C protocol mode.
Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

This register controls the interrupt generation thresholds for the command queue, response queue, and IBI queue.

CQTH[7:0] Bits (Command Queue Empty Interrupt Threshold Setting)

These bits control the minimum number of free command queue entries needed to generate the command queue empty interrupt (CMDI).

RQTH[7:0] Bits (Response Queue Full Interrupt Threshold Setting)

These bits control the minimum number of response queue entries needed to generate the response queue full interrupt (RESPI).

IDSS[7:0] Bits (IBI Data Segment Size Setting)

These bits specify the maximum size of IBI data segments in 4-byte units.
These bits allow the incoming IBI data to be sliced into multiple segments generating status individually, to support cut-through readout of a long IBI payload data.

IQTH[7:0] Bits (IBI Queue Empty/Full Interrupt Threshold Setting)

(1) For I3C Controller mode (ICMMR.ACF flag = 1):
These bits control generation of the IBI queue empty/full interrupt (IBII), based on the value of the number of outstanding IBI status in the IBI queue.
Each IBI status entry can represent either the complete IBI payload (if the IBI payload size is 4 × IDSS[7:0] bytes or less), or a segment of the IBI payload (if the IBI payload size is more than 4 × IDSS[7:0] bytes).
(2) For I3C Target mode (ICMMR.ACF flag = 0):
These bits control the minimum number of free IBI data buffer entries needed to generate the IBI queue empty/full interrupt (IBII).

43.2.44 Data Buffer Threshold Control Register (ICDBTCR)

Address(es): RI3C0.ICDBTCR 8745 F194h, RI3C0_NS.ICDBTCR 0745 F194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	RSTH[2:0]			—	—	—	—	—	TSTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	RFTH[2:0]			—	—	—	—	—	TETH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TETH[2:0]	Transmit Data Empty Interrupt Threshold Setting	b2 b0 0 0 0: Interrupt is generated when the free level of the transmit data buffer reaches 2 stages (8 bytes). 0 0 1: Interrupt is generated when the free level of the transmit data buffer reaches 4 stages (16 bytes). 0 1 0: Interrupt is generated when the free level of the transmit data buffer reaches 8 stages (32 bytes). 0 1 1: Interrupt is generated when the free level of the transmit data buffer reaches 16 stages (64 bytes). Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	RFTH[2:0]	Receive Data Full Interrupt Threshold Setting	b10 b8 0 0 0: Interrupt is generated when the fill level of the receive data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Interrupt is generated when the fill level of the receive data buffer reaches 4 stages (13 to 16 bytes). 0 1 0: Interrupt is generated when the fill level of the receive data buffer reaches 8 stages (29 to 32 bytes). 0 1 1: Interrupt is generated when the fill level of the receive data buffer reaches 16 stages (61 to 64 bytes). Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	TSTH[2:0]	Transmit Start Threshold Setting *1	b18 b16 0 0 0: Transmission starts when the fill level of the transmit data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Transmission starts when the fill level of the transmit data buffer reaches 4 stages (13 to 16 bytes). 0 1 0: Transmission starts when the fill level of the transmit data buffer reaches 8 stages (29 to 32 bytes). 0 1 1: Transmission starts when the fill level of the transmit data buffer reaches 16 stages (61 to 64 bytes). Settings other than above are prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	RSTH[2:0]	Receive Start Threshold Setting *1	b26 b24 0 0 0: Reception starts when the free level of the receive data buffer reaches 2 stages (8 bytes). 0 0 1: Reception starts when the free level of the receive data buffer reaches 4 stages (16 bytes). 0 1 0: Reception starts when the free level of the receive data buffer reaches 8 stages (32 bytes). 0 1 1: Reception starts when the free level of the receive data buffer reaches 16 stages (64 bytes). Settings other than above are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C protocol mode.
Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICDBTCR register controls the interrupt trigger thresholds for the receive data buffer and the transmit data buffer.

TETH[2:0] Bits (Transmit Data Empty Interrupt Threshold Setting)

These bits specify the minimum number of free transmit data buffer stages needed to generate the transmit data empty interrupt (TXI).

RFTH[2:0] Bits (Receive Data Full Interrupt Threshold Setting)

These bits specify the minimum number of filled receive data buffer stages needed to generate the receive data full interrupt (RXI).

TSTH[2:0] Bits (Transmit Start Threshold Setting)

When preparing to initiate a Write Transfer on the I3C bus, RI3C waits until the indicated number of stages of data have been written to the transmit buffer.

RSTH[2:0] Bits (Receive Start Threshold Setting)

When preparing to initiate a Read Transfer on the I3C bus, RI3C waits until the receive buffer has the indicated number of free stages to receive the data bytes.

43.2.45 Receive Status Queue Threshold Control Register (ICSQTCR)

Address(es): RI3C0.ICSQTCR 8745 F1C0h, RI3C0_NS.ICSQTCR 0745 F1C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SQTH[7:0]	Receive Status Queue Full Interrupt Threshold Setting	00h: Interrupt is generated when receive status queue contains one entry (4 bytes). 01h: Interrupt is generated when receive status queue contains two entries (8 bytes). Settings other than above are prohibited.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

SQTH[7:0] Bits (Receive Status Queue Full Interrupt Threshold Setting)

These bits control the minimum number of receive status queue entries needed to generate the receive status queue full interrupt (RCVI).

43.2.46 High-Priority Queue Buffer Threshold Control Register (ICHQBTCR)

Address(es): RI3C0.ICHQBTCR 8745 F1C4h, RI3C0_NS.ICHQBTCR 0745 F1C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HRQTH[7:0]								HCQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	HCQTH[7:0]	High-Priority Command Queue Empty Interrupt Threshold Setting	00h: Interrupt is generated when high-priority command queue is completely empty. 01h: Interrupt is generated when high-priority command queue has one free entry. Settings other than above are prohibited.	R/W
b15 to b8	HRQTH[7:0]	High-Priority Response Queue Full Interrupt Threshold Setting	00h: Interrupt is generated when high-priority response queue contains one entry (4 bytes). 01h: Interrupt is generated when high-priority response queue contains two entries (8 bytes). Settings other than above are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

This register controls the interrupt generation thresholds for the high-priority command queue and high-priority response queue.

HCQTH[7:0] Bits (High-Priority Command Queue Empty Interrupt Threshold Setting)

These bits control the minimum number of free high-priority command queue entries needed to generate the high-priority command queue empty interrupt (HCMDI).

HRQTH[7:0] Bits (High-Priority Response Queue Full Interrupt Threshold Setting)

These bits control the minimum number of high-priority response queue entries needed to generate the high-priority response queue full interrupt (HRESPI).

43.2.47 High-Priority Data Buffer Threshold Control Register (ICHDBTCR)

Address(es): RI3C0.ICHDBTCR 8745 F1C8h, RI3C0_NS.ICHDBTCR 0745 F1C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	HRSTH[2:0]			—	—	—	—	—	HTSTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HRFTH[2:0]			—	—	—	—	—	HTETH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	HTETH[2:0]	High-Priority Transmit Data Empty Interrupt Threshold Setting	b2 b0 0 0 0: Interrupt is generated when the free level of the high-priority transmit data buffer reaches 2 stages (8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	HRFTH[2:0]	High-Priority Receive Data Full Interrupt Threshold Setting	b10 b8 0 0 0: Interrupt is generated when the fill level of the high-priority receive data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	HTSTH[2:0]	High-Priority Transmit Start Threshold Setting	b18 b16 0 0 0: Transmission starts when the fill level of the high-priority transmit data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	HRSTH[2:0]	High-Priority Receive Start Threshold Setting	b26 b24 0 0 0: Reception starts when the free level of the high-priority receive data buffer reaches 2 stages (8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICHDBTCR register controls the interrupt trigger thresholds for the high-priority receive data buffer and the high-priority transmit data buffer.

HTETH[2:0] Bits (High-Priority Transmit Data Empty Interrupt Threshold Setting)

These bits specify the minimum number of free high-priority transmit data buffer stages needed to generate the high-priority transmit data empty interrupt (HTXI).

HRFTH[2:0] Bits (High-Priority Receive Data Full Interrupt Threshold Setting)

These bits specify the minimum number of filled high-priority receive data buffer stages needed to generate the high-priority receive data full interrupt (HRXI).

HTSTH[2:0] Bits (High-Priority Transmit Start Threshold Setting)

When preparing to initiate a Write Transfer on the I3C bus, RI3C waits until the indicated number of stages of data have been written to the high-priority transmit buffer.

HRSTH[2:0] Bits (High-Priority Receive Start Threshold Setting)

When preparing to initiate a Read Transfer on the I3C bus, RI3C waits until the high-priority receive buffer has the indicated number of free stages to receive the data bytes.

43.2.48 Status Register 2 (ICSR2)

Address(es): RI3C0.ICSR2 8745 F1D0h, RI3C0_NS.ICSR2 0745 F1D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDF	—	—	—	TMOF	—	—	—	AL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEND	—	—	—	NACKF	—	HDRXDF	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	START	START Condition Detection Flag	0: START condition is not detected. 1: START condition is detected.	R/(W) *1
b1	STOP	STOP Condition Detection Flag	0: STOP condition is not detected. 1: STOP condition is detected.	R/(W) *1
b2	HDRXDF	HDR Exit Pattern Detection Flag *2	0: HDR Exit Pattern is not detected. 1: HDR Exit Pattern is detected.	R/(W) *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	NACKF	NACK Detection Flag*3	0: NACK is not detected 1: NACK is detected.	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEND	Transmission End Flag*3	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	AL	Arbitration Lost Flag*3	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDF	Wake-Up Condition Detection Flag	0: Wake-up condition is not detected. 1: Wake-up condition is detected.	R/(W) *1
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.
Note 2. This bit is used in I3C protocol mode.
Note 3. This bit is used in I^2C protocol mode.

START Flag (START Condition Detection Flag)

[Setting condition]

- When a START condition (or a repeated START condition) is detected while the ICSER.STDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

STOP Flag (STOP Condition Detection Flag)

[Setting condition]

- When a STOP condition is detected while the ICSER.SPDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

HDRXDF Flag (HDR Exit Pattern Detection Flag)

[Setting condition]

- When a HDR Exit Pattern is detected while the ICSER.HDRXDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

NACKF Flag (NACK Detection Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.NAKDE bit is 1 (NACK detection is enabled).
  2. When ACK is not received (NACK is received) from the receiver in transmit mode.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TEND Flag (Transmission End Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.TEDE bit is 1 (transmission end detection is enabled).
  2. At the rising edge of the ninth SCL while the ICCSR.TDRE flag is 1. Excluding when sending an address.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When data is written to the ICDR register.
- When a STOP condition is detected.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

AL Flag (Arbitration Lost Flag)

[Setting condition]

When master arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.MALE bit = 1):

- When the internal SDA output state does not match the SDA00 line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA00 line is driven low while the internal SDA output is high (the SDA00 pin is in the high-impedance state)).
- The following 1 and 2 are satisfied.
  1. When the START condition is detected while the ICCGR.ST bit is 1.
  2. When the internal SDA output state does not match the SDA00 line level.
- When the ICCGR.ST bit is set to 1 (requests to generate a START condition) while the ICBSR.BFREE flag is 0.

When NACK transmission arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.NALE bit = 1):

- When the internal SDA output state does not match the SDA00 line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.SALE bit = 1):

- When the internal SDA output state does not match the SDA00 line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TMOF Flag (Timeout Detection Flag)

[Setting condition]

- All of the following 1 to 3 are satisfied.
  1. The ICSER.TMOE bit is 1 (timeout detection is enabled).
  2. When the master/Controller mode or the received slave/Target address matches the slave/Target address n in slave/Target mode.
  3. When the SCL00 line state remains unchanged for the period specified by ICTOR register.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

WUDF Flag (Wake-Up Condition Detection Flag)

This flag indicates that the condition for generating a wake-up interrupt has been met.

[Setting condition]

- When any of the following conditions is satisfied while the ICWUCR.WUFE bit is 1 (wake-up function is enabled), the ICSER.WUDE bit is 1 (wake-up condition detection is enabled), and ICWUSR.WUASYNF flag is 1 (PCLKA and I3CCLK are stopped).
  1. When the SDA line goes low (in I3C Controller mode)
  2. When the address received after the Broadcast Address (7Eh) matches its own Dynamic Address (in I3C Target mode)
  3. When the received slave address matches any of the addresses permitted for detection by the ICSCR.SAR0E to SAR2E bits, the ICSCR.HOAE bit, and the ICSCR.GCAE bit (in I^2C slave mode)

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1 while the ICWUSR.WUASYNF flag is 0 (PCLKA and I3CCLK are operating).

43.2.49 Status Detection Enable Register (ICSER)

Address(es): RI3C0.ICSER 8745 F1D4h, RI3C0_NS.ICSER 0745 F1D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDE	—	—	—	TMOE	—	—	—	ALE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEDE	—	—	—	NAKDE	—	HDRXDE	SPDE	STDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STDE	START Condition Detection Enable	0: START condition detection is disabled. 1: START condition detection is enabled.	R/W
b1	SPDE	STOP Condition Detection Enable	0: STOP condition detection is disabled. 1: STOP condition detection is enabled.	R/W
b2	HDRXDE	HDR Exit Pattern Detection Enable*1	0: HDR Exit Pattern detection is disabled. 1: HDR Exit Pattern detection is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	NAKDE	NACK Detection Enable*2	0: NACK detection is disabled. 1: NACK detection is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEDE	Transmission End Detection Enable*2	0: Transmission end detection is disabled. 1: Transmission end detection is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALE	Arbitration Lost Detection Enable *2	0: Arbitration lost detection is disabled. 1: Arbitration lost detection is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOE	Timeout Detection Enable	0: Timeout detection is disabled. 1: Timeout detection is enabled.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDE	Wake-Up Condition Detection Enable	0: Wake-up condition detection is disabled. 1: Wake-up condition detection is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C protocol mode.
Note 2. This bit is used in I2C protocol mode.

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICSR2 register is set to 1.

43.2.50 Status Interrupt Enable Register (ICSIER)

Address(es): RI3C0.ICSIER 8745 F1D8h, RI3C0_NS.ICSIER 0745 F1D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDIE	—	—	—	TMOIE	—	—	—	ALIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEIE	—	—	—	NAKIE	—	HDRXIE	SPIE	STIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STIE	START Condition Detection Interrupt Enable	0: START condition detection interrupt is disabled. 1: START condition detection interrupt is enabled.	R/W
b1	SPIE	STOP Condition Detection Interrupt Enable	0: STOP condition detection interrupt is disabled. 1: STOP condition detection interrupt is enabled.	R/W
b2	HDRXIE	HDR Exit Pattern Detection Interrupt Enable*1	0: HDR Exit Pattern detection interrupt is disabled. 1: HDR Exit Pattern detection interrupt is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	NAKIE	NACK Detection Interrupt Enable*2	0: NACK detection interrupt is disabled. 1: NACK detection interrupt is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEIE	Transmission End Interrupt Enable*2	0: Transmission end interrupt is disabled. 1: Transmission end interrupt is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALIE	Arbitration Lost Detection Interrupt Enable*2	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	TMOIE	Timeout Detection Interrupt Enable	0: Timeout detection interrupt is disabled. 1: Timeout detection interrupt is enabled.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDIE	Wake-Up Interrupt Enable	0: Wake-up interrupt is disabled. 1: Wake-up interrupt is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C protocol mode.
Note 2. This bit is used in I2C protocol mode.

This register is used to enable or disable interrupts for each status. When a flag in the ICSR2 register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is output.

43.2.51 Communication Status Register (ICCSR)

Address(es): RI3C0.ICCSR 8745 F1E0h, RI3C0_NS.ICCSR 0745 F1E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEF	—	—	—	DTAF	RQFF	CQEF	IQEFF	RDRF	TDRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDRE	Transmit Data Empty Flag	• In I3C protocol mode 0: The free level of the transmit data buffer is less than the threshold specified by the ICDBTCR.TETH[2:0] bits. 1: The free level of the transmit data buffer is equal to or more than the threshold specified by the TETH[2:0] bits. • In I2C protocol mode 0: Transmit data buffer contains transmit data. 1: Transmit data buffer contains no transmit data.	R/(W) *1
b1	RDRF	Receive Data Full Flag	• In I3C protocol mode 0: The fill level of the receive data buffer is less than the threshold specified by the ICDBTCR.RFTH[2:0] bits 1: The fill level of the receive data buffer is equal to or more than the threshold specified by the RFTH[2:0] bits. • In I2C protocol mode 0: Receive data buffer contains no receive data. 1: Receive data buffer contains receive data.	R/(W) *1
b2	IQEFF	IBI Queue Empty/Full Flag*2	• In I3C Controller mode 0: The number of IBI statuses in IBI queue is less than the threshold specified by the ICQBTCR.IQTH[7:0] bits. 1: The number of IBI statuses in IBI queue is equal to or more than the threshold specified by the IQTH[7:0] bits. • In I3C Target mode and if the IQTH[7:0] bits are 00h: 0: IBI queue contains entry. 1: IBI queue contains no entries. • In I3C Target mode and if the IQTH[7:0] bits are not 00h: 0: The number of free IBI queue entries is less than the threshold specified by the IQTH[7:0] bits. 1: The number of free IBI queue entries is equal to or more than the threshold specified by the IQTH[7:0] bits.	R/(W) *1
b3	CQEF	Command Queue Empty Flag*2	• If the ICQBTCR.CQTH[7:0] bits are 00h: 0: Command queue contains data. 1: Command queue contains no data. • If the CQTH[7:0] bits are not 00h: 0: The number of free command queue entries is less than the threshold specified by the CQTH[7:0] bits. 1: The number of free command queue entries is equal to or more than the threshold specified by the CQTH[7:0] bits.	R/(W) *1
b4	RQFF	Response Queue Full Flag*2	0: The number of response queue entries is less than the threshold specified by the ICQBTCR.RQTH[7:0] bits. 1: The number of response queue entries is equal to or more than the threshold specified by the RQTH[7:0] bits.	R/(W) *1
b5	DTAF	Data Transfer Abort Flag*2	0: Data transfer abort does not occur. 1: Data transfer abort occurs.	R/(W) *1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEF	Data Transfer Error Flag*2	0: Data transfer error does not occur. 1: Data transfer error occurs.*3	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFF	Receive Status Queue Full Flag *4	0: The number of receive status queue entries is less than the threshold specified by the ICSQTCR.SQTH[7:0] bits. 1: The number of receive status queue entries is equal to or more than the threshold specified by the SQTH[7:0] bits.	R/(W) *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. This bit is used in I3C protocol mode.
Note 3. The Error type for this error is available in the Response or Receive Status Descriptor structure corresponding to the transfer or command.
Note 4. This bit is used in I3C Secondary Controller and I3C Target modes.

TDRE Flag (Transmit Data Empty Flag)

[Setting condition]

In I3C protocol mode:

- When the free level of the transmit data buffer is equal to or more than the threshold specified by the ICDBTCR.TETH[2:0] bits while the ICCSER.TDE bit is 1 (detection of transmit data empty is enabled).

In I^{2}C protocol mode:

- When any of the following conditions is satisfied while the ICCSER.TDE bit is 1 (detection of transmit data empty is enabled).
  1. When data is transferred from transmit data buffer to shift register and transmit data buffer becomes empty.
  2. When the ICMMR.TRS flag is set to 1.
  3. When the received slave address matches while the TRS flag is 1.

[Clearing condition]

In I3C protocol mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the transmit data buffer by DMAC/DTC.
- When the free level of the transmit data buffer is less than the threshold specified by the TETH[2:0] bits.

In I^{2}C protocol mode:

- When data is written to ICDR register.
- When the ICMMR.TRS flag is set to 0.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

Note: When the ICSR2.NACKF flag is set to 1 while the ICSER.NAKDE bit is 1, RI3C aborts data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the shift register and the transmit data buffer becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

RDRF Flag (Receive Data Full Flag)

[Setting condition]

In I3C protocol mode:

- When the fill level of the receive data buffer is equal to or more than the threshold specified by the ICDBTCR.RFTH[2:0] bits while the ICCSER.RDE bit is 1 (detection of receive data full is enabled).

In I^{2}C protocol mode:

- When any of the following conditions is satisfied while the ICCSER.RDE bit is 1 (detection of receive data full is enabled).
  1. When receive data is transferred from shift register to receive data buffer.

The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (selected in the ICCSCR.WAITAE bit).

2. When the received slave address matches after a START (or repeated START) condition is detected with the ICMMR.TRS flag set to 0.

[Clearing condition]

In I3C protocol mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the receive data buffer by DMAC/DTC.
- When the fill level of the receive data buffer is less than the threshold specified by the RFTH[2:0] bits.

In I^2C protocol mode:

- When data is read from ICDR register.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

IQEFF Flag (IBI Queue Empty/Full Flag)

[Setting condition]

When any of the following conditions is satisfied while the ICCSER.IQEFDE bit is 1 (detection of IBI queue empty/full is enabled).

In I3C Controller mode:

- When the number of the IBI queue entries is equal to or more than the threshold specified by the ICQBTCR.IQTH[7:0] bits.

In I3C Target mode:

- When the IBI queue is completely empty (when the IQTH[7:0] bits are 00h).
- When the number of the free IBI queue entries is equal to or more than the threshold specified by the IQTH[7:0] bits (when the IQTH[7:0] bits are other than 00h).

[Clearing condition]

In I3C Controller mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the IBI queue by DMAC/DTC.
- When the number of the IBI queue entries is less than the threshold specified by the IQTH[7:0] bits.

In I3C Target mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the IBI queue by DMAC/DTC.
- When the IBI queue is not completely empty (when the IQTH[7:0] bits are 00h).
- When the number of the free IBI queue entries is less than the threshold specified by the IQTH[7:0] bits (when the IQTH[7:0] bits are other than 00h).

CQEF Flag (Command Queue Empty Flag)

[Setting condition]

When any of the following conditions is satisfied while the ICCSER.CQEDE bit is 1 (detection of command queue empty is enabled).

- When the command queue is completely empty (when the ICQBTCR.CQTH[7:0] bits are 00h).
- When the number of the free command queue entries is equal to or more than the threshold specified by the CQTH[7:0] bits (when the CQTH[7:0] bits are other than 00h).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the command queue by DMAC/DTC.

- When the command queue is not completely empty (when the CQTH[7:0] bits are 00h).
- When the number of the free command queue entries is less than the threshold specified by the CQTH[7:0] bits (when the CQTH[7:0] bits are other than 00h).

RQFF Flag (Response Queue Full Flag)

[Setting condition]

- When the number of the response queue entries is equal to or more than the threshold specified by the ICQBTCR.RQTH[7:0] bits while the ICCSER.RQFDE bit is 1 (detection of response queue full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the response queue by DMAC/DTC.
- When the number of the response queue entries is less than the threshold specified by the RQTH[7:0] bits.

DTAF Flag (Data Transfer Abort Flag)

[Setting condition]

- When any data transfer is aborted while the ICCSER.DTADE bit is 1 (detection of data transfer abort is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

DTEF Flag (Data Transfer Error Flag)

[Setting condition]

- When any data transfer error occurs on the I3C bus while the ICCSER.DTEDE bit is 1 (detection of data transfer error is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

SQFF Flag (Receive Status Queue Full Flag)

[Setting condition]

- When the number of the receive status queue entries is equal to or more than the threshold specified by the ICSQTCR.SQTH[7:0] bits while the ICCSER.SQFDE bit is 1 (detection of receive status queue full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the receive status queue by DMAC/DTC.
- When the number of the receive status queue entries is less than the threshold specified by the SQTH[7:0] bits.

43.2.52 Communication Status Detection Enable Register (ICCSER)

Address(es): RI3C0.ICCSER 8745 F1E4h, RI3C0_NS.ICCSER 0745 F1E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFDE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEDE	—	—	—	DTADE	RQFDE	CQEDE	IQEFDE	RDE	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Transmit Data Empty Detection Enable	0: Detection of transmit data empty is disabled. 1: Detection of transmit data empty is enabled.	R/W
b1	RDE	Receive Data Full Detection Enable	0: Detection of receive data full is disabled. 1: Detection of receive data full is enabled.	R/W
b2	IQEFDE	IBI Queue Empty/Full Detection Enable*1	0: Detection of IBI queue empty/full is disabled. 1: Detection of IBI queue empty/full is enabled.	R/W
b3	CQEDE	Command Queue Empty Detection Enable*1	0: Detection of command queue empty is disabled. 1: Detection of command queue empty is enabled.	R/W
b4	RQFDE	Response Queue Full Detection Enable*1	0: Detection of response queue full is disabled. 1: Detection of response queue full is enabled.	R/W
b5	DTADE	Data Transfer Abort Detection Enable*1	0: Detection of data transfer abort is disabled. 1: Detection of data transfer abort is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEDE	Data Transfer Error Detection Enable*1	0: Detection of data transfer error is disabled. 1: Detection of data transfer error is enabled.	R/W
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFDE	Receive Status Queue Full Detection Enable*2	0: Detection of receive status queue full is disabled. 1: Detection of receive status queue full is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C protocol mode.
Note 2. This bit is used in I3C Secondary Controller and I3C Target modes.

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICCSR register is set to 1.

43.2.53 Communication Status Interrupt Enable Register (ICCSIER)

Address(es): RI3C0.ICCSIER 8745 F1E8h, RI3C0_NS.ICCSIER 0745 F1E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEIE	—	—	—	DTAIE	RQFIE	CQEIE	IQEFIE	RIE	TIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b1	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b2	IQEFIE	IBI Queue Empty/Full Interrupt Enable*1	0: IBI queue empty/full interrupt is disabled. 1: IBI queue empty/full interrupt is enabled.	R/W
b3	CQEIE	Command Queue Empty Interrupt Enable*1	0: Command queue empty interrupt is disabled. 1: Command queue empty interrupt is enabled.	R/W
b4	RQFIE	Response Queue Full Interrupt Enable*1	0: Response queue full interrupt is disabled. 1: Response queue full interrupt is enabled.	R/W
b5	DTAIE	Data Transfer Abort Interrupt Enable*1	0: Data transfer abort interrupt is disabled. 1: Data transfer abort interrupt is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEIE	Data Transfer Error Interrupt Enable*1	0: Data transfer error interrupt is disabled. 1: Data transfer error interrupt is enabled.	R/W
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFIE	Receive Status Queue Full Interrupt Enable*2	0: Receive status queue full interrupt is disabled. 1: Receive status queue full interrupt is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C protocol mode.
Note 2. This bit is used in I3C Secondary Controller and I3C Target modes.

This register is used to enable or disable interrupts for each status. When a flag in the ICCSR register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is generated.

43.2.54 High-Priority Communication Status Register (ICHCSR)

Address(es): RI3C0.ICHCSR 8745 F200h, RI3C0_NS.ICHCSR 0745 F200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HDTEF	—	—	—	HDTAF	HRQFF	HCQEF	—	HRDRF	HTDRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HTDRE	High-Priority Transmit Data Empty Flag	0: The free level of the high-priority transmit data buffer is less than the threshold specified by the ICHDBTCR.HTETH[2:0] bits. 1: The free level of the high-priority transmit data buffer is equal to or more than the threshold specified by the HTETH[2:0] bits.	R/(W) *1
b1	HRDRF	High-Priority Receive Data Full Flag	0: The fill level of the high-priority receive data buffer is less than the threshold specified by the ICHDBTCR.HRFTH[2:0] bits 1: The fill level of the high-priority receive data buffer is equal to or more than the threshold specified by the HRFTH[2:0] bits.	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	HCQEF	High-Priority Command Queue Empty Flag	• If the ICHQBTCR.HCQTH[7:0] bits are 00h: 0: High-priority command queue contains data. 1: High-priority command queue contains no data. • If the HCQTH[7:0] bits are not 00h: 0: The number of free high-priority command queue entries is less than the threshold specified by the HCQTH[7:0] bits. 1: The number of free high-priority command queue entries is equal to or more than the threshold specified by the HCQTH[7:0] bits.	R/(W) *1
b4	HRQFF	High-Priority Response Queue Full Flag	0: The number of high-priority response queue entries is less than the threshold specified by the ICHQBTCR.HRQTH[7:0] bits. 1: The number of high-priority response queue entries is equal to or more than the threshold specified by the HRQTH[7:0] bits.	R/(W) *1
b5	HDTAF	High-Priority Data Transfer Abort Flag	0: High-priority data transfer abort does not occur. 1: High-priority data transfer abort occurs.	R/(W) *1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	HDTEF	High-Priority Data Transfer Error Flag	0: High-priority data transfer error does not occur. 1: High-priority data transfer error occurs.*2	R/(W) *1
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.
Note 1. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. The Error type for this error is available in the Response Descriptor structure corresponding to the transfer or command.

HTDRE Flag (High-Priority Transmit Data Empty Flag)

[Setting condition]

- When the free level of the high-priority transmit data buffer is equal to or more than the threshold specified by the ICHDBTCR.HTETH[2:0] bits while the ICHCSER.HTDE bit is 1 (detection of high-priority transmit data empty is

enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the high-priority transmit data buffer by DMAC/DTC.
- When the free level of the high-priority transmit data buffer is less than the threshold specified by the HTETH[2:0] bits.

HRDRF Flag (High-Priority Receive Data Full Flag)

[Setting condition]

- When the fill level of the high-priority receive data buffer is equal to or more than the threshold specified by the ICHDBTCR.HRFTH[2:0] bits while the ICHCSER.HRDE bit is 1 (detection of high-priority receive data full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the high-priority receive data buffer by DMAC/DTC.
- When the fill level of the high-priority receive data buffer is less than the threshold specified by the HRFTH[2:0] bits.

HCQEF Flag (High-Priority Command Queue Empty Flag)

[Setting condition]

When any of the following conditions is satisfied while the ICHCSER.HCQEDE bit is 1 (detection of high-priority command queue empty is enabled).

- When the high-priority command queue is completely empty (when the ICHQBTCR.HCQTH[7:0] bits are 00h).
- When the number of the free high-priority command queue entries is equal to or more than the threshold specified by the HCQTH[7:0] bits (when the HCQTH[7:0] bits are other than 00h).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the high-priority command queue by DMAC/DTC.
- When the high-priority command queue is not completely empty (when the HCQTH[7:0] bits are 00h).
- When the number of the free high-priority command queue entries is less than the threshold specified by the HCQTH[7:0] bits (when the HCQTH[7:0] bits are other than 00h).

HRQFF Flag (High-Priority Response Queue Full Flag)

[Setting condition]

- When the number of the high-priority response queue entries is equal to or more than the threshold specified by the ICHQBTCR.HRQTH[7:0] bits while the ICHCSER.HRQFDE bit is 1 (detection of high-priority response queue full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the high-priority response queue by DMAC/DTC.
- When the number of the high-priority response queue entries is less than the threshold specified by the HRQTH[7:0] bits.

HDTAF Flag (High-Priority Data Transfer Abort Flag)

[Setting condition]

- When any high-priority data transfer is aborted while the ICHCSER.HDTADE bit is 1 (detection of high-priority

data transfer abort is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

HDTEF Flag (High-Priority Data Transfer Error Flag)

[Setting condition]

- When any high-priority data transfer error occurs on the I3C bus while the ICHCSER.HDTEDE bit is 1 (detection of high-priority data transfer error is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

43.2.55 High-Priority Communication Status Detection Enable Register (ICHCSER)

Address(es): RI3C0.ICHCSER 8745 F204h, RI3C0_NS.ICHCSER 0745 F204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HDTEDE	—	—	—	HDTADE	HRQFDE	HCQEDE	—	HRDE	HTDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HTDE	High-Priority Transmit Data Empty Detection Enable	0: Detection of high-priority transmit data empty is disabled. 1: Detection of high-priority transmit data empty is enabled.	R/W
b1	HRDE	High-Priority Receive Data Full Detection Enable	0: Detection of high-priority receive data full is disabled. 1: Detection of high-priority receive data full is enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	HCQEDE	High-Priority Command Queue Empty Detection Enable	0: Detection of high-priority command queue empty is disabled. 1: Detection of high-priority command queue empty is enabled.	R/W
b4	HRQFDE	High-Priority Response Queue Full Detection Enable	0: Detection of high-priority response queue full is disabled. 1: Detection of high-priority response queue full is enabled.	R/W
b5	HDTADE	High-Priority Data Transfer Abort Detection Enable	0: Detection of high-priority data transfer abort is disabled. 1: Detection of high-priority data transfer abort is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	HDTEDE	High-Priority Data Transfer Error Detection Enable	0: Detection of high-priority data transfer error is disabled. 1: Detection of high-priority data transfer error is enabled.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICHCSR register is set to 1.

43.2.56 High-Priority Communication Status Interrupt Enable Register (ICHCSIER)

Address(es): RI3C0.ICHCSIER 8745 F208h, RI3C0_NS.ICHCSIER 0745 F208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HDTEIE	—	—	—	HDTAIE	HRQFIE	HCQEIE	—	HRIE	HTIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HTIE	High-Priority Transmit Data Empty Interrupt Enable	0: High-priority transmit data empty interrupt is disabled. 1: High-priority transmit data empty interrupt is enabled.	R/W
b1	HRIE	High-Priority Receive Data Full Interrupt Enable	0: High-priority receive data full interrupt is disabled. 1: High-priority receive data full interrupt is enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	HCQEIE	High-Priority Command Queue Empty Interrupt Enable	0: High-priority command queue empty interrupt is disabled. 1: High-priority command queue empty interrupt is enabled.	R/W
b4	HRQFIE	High-Priority Response Queue Full Interrupt Enable	0: High-priority response queue full interrupt is disabled. 1: High-priority response queue full interrupt is enabled.	R/W
b5	HDTAIE	High-Priority Data Transfer Abort Interrupt Enable	0: High-priority data transfer abort interrupt is disabled. 1: High-priority data transfer abort interrupt is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	HDTEIE	High-Priority Data Transfer Error Interrupt Enable	0: High-priority data transfer error interrupt is disabled. 1: High-priority data transfer error interrupt is enabled.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to enable or disable interrupts for each status. When a flag in the ICHCSR register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is generated.

43.2.57 Bus Status Register (ICBSR)

Address(es): RI3C0.ICBSR 8745 F210h, RI3C0_NS.ICBSR 0745 F210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BIDL	BAVL	BFREE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFREE	Bus Free Flag	0: Have not detected bus free. 1: Have detected bus free.	R
b1	BAVL	Bus Available Detection Flag*1	0: Have not detected bus available. 1: Have detected bus available.	R
b2	BIDL	Bus Idle Detection Flag*1	0: Have not detected bus idle. 1: Have detected bus idle.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

Note 1. This bit is used in I3C protocol mode.

BFREE Flag (Bus Free Flag)

This flag indicates whether the I3C bus is occupied (bus busy) or released (bus free).

[Setting condition]

- When the number of cycles set in the ICBFTR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBFTR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than high (in I3C protocol mode).
- When a START condition is detected (in I^2C protocol mode).
- When the ICCR.ICE bit is set to 0.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

BAVL Flag (Bus Available Detection Flag)

The Bus Available condition is a period during which the Bus Free condition is sustained continuously for a duration of at least tAVAL (refer to section 67.5.7.10, RI3C). A Target can only issue a START Request (for an IBI or for a CRR) after a Bus Available condition.

[Setting condition]

- When the number of cycles set in the ICBATR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBATR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than high.

- When the ICCR.ICE bit is set to 0.

BIDL Flag (Bus Idle Detection Flag)

The Bus Idle condition is a period during which the Bus Available condition is sustained continuously for a duration of at least tIDLE (refer to section 67.5.7.10, RI3C).

[Setting condition]

- When the number of cycles set in the ICBITR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBITR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than High.
- When the ICCR.ICE bit is set to 0.

43.2.58 Slave Mode Status Register (ICSSR)

Address(es): RI3C0.ICSSR 8745 F214h, RI3C0_NS.ICSSR 0745 F214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	—	—	—	—	—	—	—	DID	HSMC	—	—	—	—	GCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMC	Hs-mode Master Code Detection Flag	0: Hs-mode master code is not detected. 1: Hs-mode master code is detected.	R/(W) *1
b6	DID	Device-ID Address Detection Flag	0: Device-ID address is not detected. 1: Device-ID address is detected. • This bit is set to 1 when the first frame received immediately after a START condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1
b16	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b17	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b18	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I²C protocol mode.
Note 1. Only 0 can be written to this bit.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.GCAE bit is 1 (general call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000b + 0 (write)).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

HSMC Flag (Hs-mode Master Code Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HSMCE bit is 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXXb) + 1 (NACK).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of device ID (1111 100b) + 0 (write).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected does not match a value of device ID (1111 100b).
  3. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2.

HOA Flag (Host Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HOAE bit is 1 (host address detection is enabled).
  2. When the received slave address matches the host address (0001 000b).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting condition]

For 7-bit address format (SAMRy.FS flag = 0):

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches the SAMRy.SVA[6:0] bits value.

For 10-bit address format (SAMRy.FS flag = 1):

- This flag is set to 1 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 1111 0b + SAMRy.SVA[9:8] bits and the following address matches the SAMRy.SVA[7:0] bits value.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.

For 7-bit address format (SAMRy.FS flag = 0):

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match SAMRy.SVA[6:0] bits value.

For 10-bit address format (SAMRy.FS flag = 1):

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match a value of 1111 0b + SAMRy.SVA[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 1111 0b + SAMRy.SVA[9:8] bits and the following address does not match the SAMRy.SVA[7:0] bits value.

43.2.59 Wake-Up Unit Status Register (ICWUSR)

Address(es): RI3C0.ICWUSR 8745 F218h, RI3C0_NS.ICWUSR 0745 F218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WUASYNF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WUASYNF	Wake-Up Detection Circuit Status Flag	0: Wake-up detection circuit is disabled (PCLKA and I3CCLK are operating). 1: Wake-up detection circuit is enabled (PCLKA and I3CCLK are stopped).	R
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

WUASYNF Flag (Wake-Up Detection Circuit Status Flag)

This bit indicates whether the wake-up detection circuit is enabled or disabled.

[Setting condition]

- When the ICBSR.BFREE flag is set to 1 (bus free) while the ICWUCR.WUFE bit is 1 (wake-up function is enabled) and the WUFSYNE bit is 0 (asynchronous operation).

[Clearing condition]

- When the ICWUCR.WUFE bit is set to 0 (wake-up function is disabled).
- When a wake-up event is detected while the WUFE bit is 1, and then the ICWUCR.WUFSYNE bit is set to 1 (internal clock synchronous operation) while this flag is set to 1 (in I^2C slave mode).
- When a STOP condition is detected after a wake-up event is detected while the WUFE bit is 1 and the WUFSYNE bit is set to 1 while this flag is set to 1 (in I3C Target mode).
- When a STOP condition is detected while the WUFE bit is 1, this flag is 1, and the WUFSYNE bit is 1 (in I^2C slave mode or in I3C Target mode).
- When a wake-up event is detected while the WUFE bit is 1 and the WUFSYNE bit is set to 1 while this flag is set to 1 (in I3C Controller mode).

43.2.60 C_SyncCNT Capture Register (ICCSCCR)

Address(es): RI3C0.ICCSCCR 8745 F21Ch, RI3C0_NS.ICCSCCR 0745 F21Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICCSCCR register is a 32-bit read-only register used to read the value of C_SyncCNT captured by aBE. This register is used only in Async Mode 1.

43.2.61 Target Device Address Table Register m (ICTDATRm) (m = 0 to 7)

Address(es): RI3C0.ICTDATR0 8745 F224h, RI3C0.ICTDATR1 8745 F22Ch, RI3C0.ICTDATR2 8745 F234h, RI3C0.ICTDATR3 8745 F23Ch, RI3C0.ICTDATR4 8745 F244h, RI3C0.ICTDATR5 8745 F24Ch, RI3C0.ICTDATR6 8745 F254h, RI3C0.ICTDATR7 8745 F25Ch, RI3C0_NS.ICTDATR0 0745 F224h, RI3C0_NS.ICTDATR1 0745 F22Ch, RI3C0_NS.ICTDATR2 0745 F234h, RI3C0_NS.ICTDATR3 0745 F23Ch, RI3C0_NS.ICTDATR4 0745 F244h, RI3C0_NS.ICTDATR5 0745 F24Ch, RI3C0_NS.ICTDATR6 0745 F254h, RI3C0_NS.ICTDATR7 0745 F25Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TYPE	NACKRC[1:0]		—	—	—	—	—	DADR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IBITSE	CRRRJ	TIRRJ	IBIPL	—	—	—	—	—	SADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	SADR[6:0]	Static Address	I3C/I2C static address	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	IBIPL	IBI Payload	0: IBIs from this device do not carry data payload. 1: IBIs from this device carry data payload.	R/W
b13	TIRRJ	Target Interrupt Request Reject	0: RI3C ACKs Target Interrupt Requests from this device. 1: RI3C NACKs Target Interrupt Requests from this device and automatically sends DISEC CCC.	R/W
b14	CRRRJ	Controller Role Request Reject	0: RI3C ACKs Controller Role Requests from this device. 1: RI3C NACKs Controller Role Requests from this device and automatically sends DISEC CCC.	R/W
b15	IBITSE	IBI Timestamp Enable	0: RI3C does not timestamp IBIs from this device with Controller timestamps. 1: RI3C timestamps IBIs from this device with Controller timestamps.	R/W
b23 to b16	DADR[7:0]	Dynamic Address	DADR[6:0] bits: Dynamic Address for this device DADR[7] bit: Parity bit	R/W
b28 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30, b29	NACKRC[1:0]	NACK Retry Count Setting	0h: No retries on NACK 1h to 3h: Up to 1 to 3 retries on NACK	R/W
b31	TYPE	Device Type	0: I3C device 1: I2C device	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The Device Address Table (DAT) stores the device addresses and control information of Target devices attached to the I3C bus and is implemented in RI3C as the ICTDATRm registers. The Command Descriptor of a command that requires this register information has a DEV_INDEX field, and the register is selected by the field.

IBIPL Bit (IBI Payload)

This bit indicates whether IBIs from the corresponding device have a data payload. This bit reflects the ICTDCTRm.IBIPL bit (BCR[2]: IBI Payload bit in the Bus Characteristics Register).
During IBI handling for this device, the Controller shall use this bit to determine whether or not to drive reception of the IBI data payload. Data continuation is indicated by the T-Bit.

TIRRJ Bit (Target Interrupt Request Reject)

This bit controls whether RI3C, when operating in the Active Controller role, will accept or reject Target Interrupt Requests (TIR) from the Target device indicated by this register.

CRRRJ Bit (Controller Role Request Reject)

This bit controls whether RI3C, when operating in the Active Controller role, will accept or reject Controller Role Requests (CRR) from the Secondary Controller indicated by this register.

IBITSE Bit (IBI Timestamp Enable)

This bit enables or disables IBI timestamping for a specific device.

Note: The IBI Status Descriptor for each IBI event indicates whether or not the individual IBI event was actually timestamped. Set to 0 except for Async Mode 0 and Async Mode 1 of Timing Control Mode.

NACKRC[1:0] Bits (NACK Retry Count Setting)

These bits set the number of retries when a NACK response is received from the Target for the transaction set in the Command Descriptor.

Note: When ENTDAA CCC is executed by address assignment command (refer to section 43.3.1.1, Address Assignment Command), the setting of this bit is ignored and the transaction ends when NACK is received once.
Note: When a NACK is received for a Broadcast Address, RI3C does not retry regardless of the setting of the NACKRC[1:0] bits.
Note: When the NACKRC[1:0] bits are set to 00b, the RI3C does not retry even for Direct CCCs.

43.2.62 Extended Target Device Address Table Register (ICEDATR)

Address(es): RI3C0.ICEDATR 8745 F2A0h, RI3C0_NS.ICEDATR 0745 F2A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TYPE	NACKRC[1:0]		—	—	—	—	—	DADR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	SADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	SADR[6:0]	Static Address	I3C/I²C static address	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23 to b16	DADR[7:0]	Dynamic Address	DADR[6:0] bits: Dynamic Address for this device DADR[7] bit: Parity bit	R/W
b28 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30, b29	NACKRC[1:0]	NACK Retry Count Setting	0h: No retries on NACK 1h to 3h: Up to 1 to 3 retries on NACK	R/W
b31	TYPE	Device Type	0: I3C device 1: I²C device	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.63 Device Address Register y (ICDARy) (y = 0 to 2)

Address(es): RI3C0.ICDAR0 8745 F2B0h, RI3C0.ICDAR1 8745 F2B4h, RI3C0.ICDAR2 8745 F2B8h,
RI3C0_NS.ICDAR0 0745 F2B0h, RI3C0_NS.ICDAR1 0745 F2B4h, RI3C0_NS.ICDAR2 0745 F2B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	DADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IBIPL	—	—	SADR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SADR[9:0]	Static Address	Set a static address.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	IBIPL	IBI Payload	This bit is the mirror bit of the ICDCTR.IBIPL. 0: IBIs from this device do not carry a data payload. 1: IBIs from this device carry a data payload.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b16	DADR[6:0]	Dynamic Address	Dynamic Address is set.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.
Note: Do not write to this register when RI3C is Primary Controller.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICDARy register is used to store the values of the assigned Dynamic Address.

SADR[9:0] Bits (Static Address)

Set the static address in Target mode to the lower 7 bits.
The upper 3 bits are reserved. Set them to 0.

IBIPL Bit (IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This bit reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).
During IBI handling for this Device, the Controller shall use this bit to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

DADR[6:0] Bits (Dynamic Address)

These bits store the Dynamic Address assigned in Target mode.
These bits are updated when one any of the following events occurs.

- When writing Dynamic Address value.
- When Target Address is equal to its own Static Address in receiving SETDASA CCC (Direct), these bits are updated to the value of the subsequent Dynamic Address.
- When Dynamic Address Assignment procedure that starts by receiving ENTDAA CCC (Broadcast) is established.
- When receiving RSTDAA CCC (Broadcast), all bits are set to 0.
- When Target Address is equal to its own Dynamic Address in receiving RSTDAA CCC (Direct), all bits are set to 0.

- When Target Address is equal to its own Dynamic Address in receiving SETNEWDA CCC (Direct), these bits are updated to the value of the subsequent new Dynamic Address.
- When receiving SETAASA CCC (Broadcast), these bits are updated to the value of the SADR[6:0] bits.

43.2.64 Slave Address Register y (SARy) (y = 0 to 2)

Address(es): RI3C0.SAR0 8745 F2B0h, RI3C0.SAR1 8745 F2B4h, RI3C0.SAR2 8745 F2B8h,
RI3C0_NS.SAR0 0745 F2B0h, RI3C0_NS.SAR1 0745 F2B4h, RI3C0_NS.SAR2 0745 F2B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FS	SVA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SVA[9:0]	Slave Address	Set a slave address.	R/W
b10	FS	Address Format Select	0: 7-bit address format 1: 10-bit address format	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I^2C protocol mode.

FS Bit (Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address (SVA[9:0] bits).
When this bit is 0, the 7-bit address format is selected for slave address, the SVA[6:0] bit setting is valid, and the SVA[9:7] bit setting is ignored.
When this bit is 1, the 10-bit address format is selected for slave address and the SVA[9:0] bit setting is valid.
When the ICSCR.SARyE bit is 0, the setting of this bit is ignored.

43.2.65 Target Device Characteristics Table Register m (ICTDCTRm) (m = 0 to 7)

Address(es): RI3C0.ICTDCTR0 8745 F2D0h, RI3C0.ICTDCTR1 8745 F2D4h, RI3C0.ICTDCTR2 8745 F2D8h, RI3C0.ICTDCTR3 8745 F2DCh, RI3C0.ICTDCTR4 8745 F2E0h, RI3C0.ICTDCTR5 8745 F2E4h, RI3C0.ICTDCTR6 8745 F2E8h, RI3C0.ICTDCTR7 8745 F2ECh, RI3C0_NS.ICTDCTR0 0745 F2D0h, RI3C0_NS.ICTDCTR1 0745 F2D4h, RI3C0_NS.ICTDCTR2 0745 F2D8h, RI3C0_NS.ICTDCTR3 0745 F2DCh, RI3C0_NS.ICTDCTR4 0745 F2E0h, RI3C0_NS.ICTDCTR5 0745 F2E4h, RI3C0_NS.ICTDCTR6 0745 F2E8h, RI3C0_NS.ICTDCTR7 0745 F2ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ROLE[1:0]		ADVCAP	VTSUP	OFLC	IBIPL	IBIRQC	LIMIT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	LIMIT	Max Data Speed Limitation	0: No limitation 1: Limitation	R/W
b9	IBIRQC	IBI Request Capable	0: Not capable 1: Capable	R/W
b10	IBIPL	IBI Payload	0: No data byte follows the accepted IBI. 1: One or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
b11	OFLC	Offline Capable	0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
b12	VTSUP	Virtual Target Support	0: Device is not a Virtual Target and does not expose other downstream Device(s). 1: Device is a Virtual Target or expose other downstream Device(s).	R/W
b13	ADVCAP	Advanced Capabilities	0: Device does not support optional advanced capabilities. 1: Device supports optional advanced capabilities. Use GETCAPS CCC to determine which ones.	R/W
b15, b14	ROLE[1:0]	Device Role Setting	b15 b14 0 0: I3C Target 0 1: I3C Controller capable*1 Other than above are reserved.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.
Note 1. For an I3C Device acting as I3C Primary Controller, the ROLE[1:0] bits will contain the value 01b.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICTDCTRm register contains the value of the Bus Characteristic Register (BCR) of the Target device captured by the RI3C during Controller operation.
This register is updated when the Bus Characteristics Register (BCR) is received from the device in the Dynamic Address Allocation procedure starting by receiving the ENTDAA CCC (Broadcast).

LIMIT Bit (Max Data Speed Limitation)

This bit corresponds to bit 0 in the BCR (BCR[0]).
Controller shall use the GETMXDS CCC to interrogate the Target for specific limitation.

IBIRQC Bit (IBI Request Capable)

This bit corresponds to bit 1 in the BCR (BCR[1]).

IBIPL Bit (IBI Payload)

This bit corresponds to bit 2 in the BCR (BCR[2]).

OFLC Bit (Offline Capable)

This bit corresponds to bit 3 in the BCR (BCR[3]).
Offline capable devices retain the Dynamic Address.

VTSUP Bit (Virtual Target Support)

This bit corresponds to bit 4 in the BCR (BCR[4]).

ADVCAP Bit (Advanced Capabilities)

This bit corresponds to bit 5 in the BCR (BCR[5]).

ROLE[1:0] Bits (Device Role Setting)

This bit corresponds to bits 7 and 6 in the BCR (BCR[7:6]).

43.2.66 Device Characteristics Table Register (ICDCTR)

Address(es): RI3C0.ICDCTR 8745 F320h, RI3C0_NS.ICDCTR 0745 F320h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ROLE[1:0]		ADVCAP	VTSUP	OFLC	IBIPL	IBIRQC	LIMIT	DCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DCR[7:0]	Device Characteristics	255 available codes for describing the type of sensor, or device. Examples: Accelerometer, gyroscope, composite devices Default value is 00h: Generic device	R/W
b8	LIMIT	Max Data Speed Limitation*1	0: No limitation 1: Limitation	R/W
b9	IBIRQC	IBI Request Capable	0: Not capable 1: Capable	R/W
b10	IBIPL	IBI Payload	0: No data byte follows the accepted IBI. 1: One or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
b11	OFLC	Offline Capable*2	0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
b12	VTSUP	Virtual Target Support	0: Device is not a Virtual Target and does not expose other downstream Device(s). 1: Device is a Virtual Target or expose other downstream Device(s).	R/W
b13	ADVCAP	Advanced Capabilities	0: Device does not support optional advanced capabilities. 1: Device supports optional advanced capabilities.	R/W
b15, b14	ROLE[1:0]	Device Role Setting	b15 b14 0 0: I3C Target 0 1: I3C Controller capable*3 Settings other than above are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.
Note 1. Controller shall use the GETMXDS CCC to interrogate the Target for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. For an I3C Device acting as I3C Primarily Controller, the ROLE[1:0] bits will contain the value 01b.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICDCTR register is used to store the values of the RI3C Bus Characteristics Register (BCR) and Device Characteristics Register (DCR). The BCR consists of the ROLE[1:0], OFLC, IBIPL, IBIRQC, and LIMIT bits.

DCR[7:0] Bits (Device Characteristics)

Each I3C device that is connected to the I3C bus has an associated Device Characteristics Register (DCR). This register describes the I3C compliant device type such as accelerometer and gyroscope, for use in Dynamic Address assignment and Common Command Codes.

LIMIT Bit (Max Data Speed Limitation)

This bit corresponds to bit 0 in the BCR (BCR[0]).

IBIRQC Bit (IBI Request Capable)

This bit corresponds to bit 1 in the BCR (BCR[1]).

IBIPL Bit (IBI Payload)

This bit corresponds to bit 2 in the BCR (BCR[2]).

OFLC Bit (Offline Capable)

This bit corresponds to bit 3 in the BCR (BCR[3]).

VTSUP Bit (Virtual Target Support)

This bit corresponds to bit 4 in the BCR (BCR[4]).

ADVCAP Bit (Advanced Capabilities)

This bit corresponds to bit 5 in the BCR (BCR[5]).

ROLE[1:0] Bits (Device Role Setting)

This bit corresponds to bits 7 and 6 in the BCR (BCR[7:6]).

When RI3C is in Target mode and issues IBI by Command Descriptor, set the BCR as follows:

(1) Target Interrupt Request: No IBI Payload follow the accepted IBI:

- IBIRQC = 1
- IBIPL = 0

Note: Set DATA_LENGTH field of Command Descriptor to 0000h.

(2) Target Interrupt Request: IBI Payload follow the accepted IBI:

- IBIRQC = 1
- IBIPL = 1

Note: Set DATA_LENGTH field of Command Descriptor to any value.

(3) Controller Role Request:

- IBIRQC = 1
- ROLE[1:0] = 01b

When RI3C is in Target mode and receives CCC from I3C Controller, it performs the following operations according to the setting of the BCR:

- When the IBIPL bit is 1, RI3C sends the value set in the ICMRLR.IBIPL[7:0] bits as the 3rd byte data to GETMRL CCC from I3C Controller.
- When the LIMIT bit is 0, RI3C NACKs GETMXDS CCC from I3C Controller.
- When the LIMIT bit is 1, RI3C ACKs GETMXDS CCC from I3C Controller and sends data set in the ICMWSR, ICMRSR, and ICMTTR registers.

43.2.67 Provisioned ID Low Register (ICPIDLR)

Address(es): RI3C0.ICPIDLR 8745 F324h, RI3C0_NS.ICPIDLR 0745 F324h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
This register stores the lower 16 bits of the provisioned ID (b15 to b0 of the PID).
The upper 16 bits are reserved. These bits are read as 0 and the write value should be 0.

43.2.68 Provisioned ID High Register (ICPIDHR)

Address(es): RI3C0.ICPIDHR 8745 F328h, RI3C0_NS.ICPIDHR 0745 F328h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
This register stores the upper 32 bits of the provisioned ID (b47 to b16 of the PID).

43.2.69 Device Address Monitor Register y (ICDAMRy) (y = 0 to 2)

Address(es): RI3C0.ICDAMR0 8745 F330h, RI3C0.ICDAMR1 8745 F334h, RI3C0.ICDAMR2 8745 F338h, RI3C0_NS.ICDAMR0 0745 F330h, RI3C0_NS.ICDAMR1 0745 F334h, RI3C0_NS.ICDAMR2 0745 F338h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAV	SAV	—	—	—	—	TADR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0.	R
b25 to b16	TADR[9:0]	Target Address	Target address is set.	R
b29 to b26	—	Reserved	These bits are read as 0.	R
b30	SAV	Static Address Valid Flag	0: Static address is disabled. 1: Static address is enabled.	R
b31	DAV	Dynamic Address Valid Flag	0: Dynamic Address is disabled. 1: Dynamic Address is enabled.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

TADR[9:0] Bits (Target Address)

The TADR[9:0] bits indicate a valid Target address.

- When the DAV flag is 1:
  The TADR[9:7] bits are 000b.
  The TADR[6:0] bits are the ICDARy.DADR[6:0] bits.
- When the SAV flag is 1:
  The TADR[9:7] bits are 000b.
  The TADR[6:0] bits are the ICDARy.SADR[6:0] bits.

SAV Flag (Static Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICTCR.TAyDE bit is 1 (detection of Target address y is enabled).
  2. The DAV flag is 0 (Dynamic Address is disabled).
  3. The ICDARy.SADR[6:0] bits are not 0000000b.

[Clearing condition]

- [Setting condition] is not satisfied.

DAV Flag (Dynamic Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICTCR.TAyDE bit is 1 (detection of Target address y is enabled).
  2. The ICDARy.DADR[6:0] bits are not 0000000b.

[Clearing condition]

- [Setting condition] is not satisfied.

43.2.70 Slave Address Monitor Register y (SAMRy) (y = 0 to 2)

Address(es): RI3C0.SAMR0 8745 F330h, RI3C0.SAMR1 8745 F334h, RI3C0.SAMR2 8745 F338h, RI3C0_NS.SAMR0 0745 F330h, RI3C0_NS.SAMR1 0745 F334h, RI3C0_NS.SAMR2 0745 F338h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SAV	—	—	FS	—	SVA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0.	R
b25 to b16	SVA[9:0]	Slave Address	Slave address is set.	R
b26	—	Reserved	This bit is read as 0.	R
b27	FS	Address Format Flag	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
b29, b28	—	Reserved	These bits are read as 0.	R
b30	SAV	Slave Address Valid Flag	0: Slave address is disabled. 1: Slave address is enabled.	R
b31	—	Reserved	This bit is read as 0.	R

Note: This register is used in I²C protocol mode.

SVA[9:0] Bits (Slave Address)

The SVA[9:0] bits indicate a valid slave address.

- When the SAV flag is 1 and the FS flag is 0:
  The SVA[9:7] bits are 000b.
  The SVA[6:0] bits are the SARy.SVA[6:0] bits.
- When the SAV flag is 1 and the FS flag is 1:
  The SVA[9:0] bits are the SARy.SVA[9:0] bits.

FS Flag (Address Format Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (detection of slave y is enabled).
  2. The SARy.FS bit is 1 (the address length is 10 bits).

[Clearing condition]

- [Setting condition] is not satisfied.

SAV Flag (Slave Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (detection of slave address y is enabled).
  2. The SARy.SVA[6:0] bits are not 0000000b when the FS flag is 0.
     The SARy.SVA[9:0] bits are not 0000000000b when the FS flag is 1.

[Clearing condition]

- [Setting condition] is not satisfied.

43.2.71 Target Event Register (ICTEVR)

Address(es): RI3C0.ICTEVR 8745 F350h, RI3C0_NS.ICTEVR 0745 F350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENCR	ENINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENINT	Target Interrupt Request Enable Flag	0: Target-initiated interrupts is disabled by the Controller. 1: Target-initiated interrupts is enabled by the Controller.	R/W
b1	ENCR	Controller Role Request Enable Flag	0: Controller Role Requests from Secondary Controllers is disabled by the Active Controller. 1: Controller Role Requests from Secondary Controllers is enabled by the Active Controller.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

ENINT Flag (Target Interrupt Request Enable Flag)

This flag allows the Controller to control when Target-initiated interrupts are allowed on the I3C bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with the ENINT field set to 1.
- When receiving ENEC CCC (Direct) with the ENINT field set to 1 and with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with the DISINT field set to 1.
- When receiving DISEC CCC (Direct) with the DISINT field set to 1 and with its own Target address.

ENCR Flag (Controller Role Request Enable Flag)

This flag allows the Active Controller to control when Controller Role Requests from Secondary Controller are allowed on the I3C bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with the ENCR field set to 1.
- When receiving ENEC CCC (Direct) with the ENCR field set to 1 and with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with the DISCR field set to 1.
- When receiving DISEC CCC (Direct) with the DISCR field set to 1 and with its own Target address.

43.2.72 Activity State Register (ICASR)

Address(es): RI3C0.ICASR 8745 F354h, RI3C0_NS.ICASR 0745 F354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ENTAS3	ENTAS2	ENTAS1	ENTAS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENTAS0	Activity State 0 Flag*1	0: Other than activity state 0 1: Activity state 0	R/W
b1	ENTAS1	Activity State 1 Flag*1	0: Other than activity state 1 1: Activity state 1	R/W
b2	ENTAS2	Activity State 2 Flag*1	0: Other than activity state 2 1: Activity state 2	R/W
b3	ENTAS3	Activity State 3 Flag*1	0: Other than activity state 3 1: Activity state 3	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.
Note 1. Do not set multiple bits to 1 at the same time.

ENTAS0 Flag (Activity State 0 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS0 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS1 CCC (Broadcast), ENTAS2 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS1 CCC (Direct), ENTAS2 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS1 Flag (Activity State 1 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS1 CCC (Broadcast).
- When receiving ENTAS1 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS2 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS2 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS2 Flag (Activity State 2 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS2 CCC (Broadcast).
- When receiving ENTAS2 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS1 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS1 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS3 Flag (Activity State 3 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS3 CCC (Broadcast).
- When receiving ENTAS3 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS1 CCC (Broadcast), or ENTAS2 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS1 CCC (Direct), or ENTAS2 CCC (Direct) with its own Target address.

43.2.73 Max Write Length Register (ICMWLR)

Address(es): RI3C0.ICMWLR 8745 F358h, RI3C0_NS.ICMWLR 0745 F358h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MWL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MWL[15:0]	Max Write Length	Set the maximum data write length in bytes to respond to GETMWL CCC. Set a value of 8 or more. These bits are automatically updated when SETMWL CCC is received.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

MWL[15:0] Bits (Max Write Length)

These bits are used when the I3C Controller sets the maximum data write length in RI3C or gets the maximum data write length from RI3C.

The value of these bits is automatically transmitted in response to GETMRL CCC.

When receiving SETMWL CCC (Broadcast) or SETMWL CCC (Direct) with its own Target address, these bits are automatically updated to the received MWL value.

43.2.74 Max Read Length Register (ICMRLR)

Address(es): RI3C0.ICMRLR 8745 F35Ch, RI3C0_NS.ICMRLR 0745 F35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IBIPL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MRL[15:0]	Max Read Length	Set the maximum data read length in bytes to respond to GETMRL CCC. Set a value of 16 or more. These bits are automatically updated when SETMRL CCC is received.	R/W
b23 to b16	IBIPL[7:0]	IBI Payload Size	Set the maximum IBI payload size in bytes to respond to GETMRL CCC. These bits are automatically updated when SETMRL CCC is received.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

MRL[15:0] Bits (Max Read Length)

These bits are used when the I3C Controller sets the maximum data read length in RI3C or gets the maximum data read length from RI3C.

The value of these bits is automatically transmitted as the first two bytes of the response to GETMRL CCC.

When receiving SETMRL CCC (Broadcast) or SETMRL CCC (Direct) with its own Target address, these bits are automatically updated to the received MRL value.

IBIPL[7:0] Bits (IBI Payload Size)

These bits are used when the I3C Controller sets the maximum IBI payload size in RI3C or gets the maximum IBI payload size from RI3C.

When the ICDCTR.IBIPL bit is 1, the value of these bits is transmitted as the third byte of the response to GETMRL CCC. When the IBIPL bit is 0, these bits are not transmitted.

When receiving SETMRL CCC (Broadcast) or SETMRL CCC (Direct) with its own Target address, these bits are automatically updated to the received IBI Payload Size value.

43.2.75 Test Mode Register (ICTMR)

Address(es): RI3C0.ICTMR 8745 F360h, RI3C0_NS.ICTMR 0745 F360h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TM[7:0]	Test Mode	00h: Exit test mode This value removes all I3C devices from test mode. 01h: Vendor test mode This value indicates that I3C devices shall return a random 32-bit value in the provisioned ID during the Dynamic Address Assignment procedure. Settings other than above are prohibited.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

TM[7:0] Bits (Test Mode)

These bits are used when the I3C Controller puts RI3C into test mode.

When receiving ENTTM CCC (Broadcast), these bits are automatically updated to the received Test Mode Byte value.

43.2.76 Device Status Register (ICDSR)

Address(es): RI3C0.ICDSR 8745 F364h, RI3C0_NS.ICDSR 0745 F364h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VRSV[7:0]								CAS[1:0]		PERR	—	PNDINT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PNDINT[3:0]	Pending Interrupt Number	Contains the interrupt number of any pending interrupt, or 0000b if no interrupts are pending.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R
b5	PERR	Protocol Error Flag	0: The Target has not detected a protocol error since the last status read. 1: The Target has detected a protocol error since the last status read.	R
b7, b6	CAS[1:0]	Current Activity State	b7 b6 0 0: Activity state 0 0 1: Activity state 1 1 0: Activity state 2 1 1: Activity state 3	R/W
b15 to b8	VRSV[7:0]	Vendor Reserved Area	Reserved for vendor-specific meaning	R/W
b31 to b16	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the status to be returned when current status is requested from I3C Controller.
The value of bits b15 to b0 in this register is automatically transmitted when a GETSTATUS CCC (Direct) is received.

PNDINT[3:0] Bits (Pending Interrupt Number)

These bits are used to indicate whether any interrupts are pending. If no interrupts are pending, set these bits to 0000b.
If multiple interrupts are pending, set the interrupt number of the highest priority interrupt.

PERR Flag (Protocol Error Flag)

This bit indicates whether the Target has detected a protocol error since the last status read.
This flag is automatically cleared each time the Controller successfully reads the Target status.
[Setting condition]

- When the Target detected a protocol error.

[Clearing condition]

- When a transmission of status (value set in this register) is successfully completed in response to GETSTATUS CCC (Direct) with its own Target address.

CAS[1:0] Bits (Current Activity State)

These bits are used to indicate the current Activity state of RI3C.

43.2.77 Max Write Speed Register (ICMWSR)

Address(es): RI3C0.ICMWSR 8745 F368h, RI3C0_NS.ICMWSR 0745 F368h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MSWDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MSWDR[2:0]	Maximum Sustained Write Data Rate	b2 b0 0 0 0: Maximum value of f_{SCL} 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Settings other than above are prohibited.	R/W
b31 to b3	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the maximum write speed to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b7 to b0 in this register is automatically transmitted as MaxWr when a GETMXDS CCC (Direct) is received.

43.2.78 Max Read Speed Register (ICMRSR)

Address(es): RI3C0.ICMRSR 8745 F36Ch, RI3C0_NS.ICMRSR 0745 F36Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	TSCO[2:0]			MSRDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MSRDR[2:0]	Maximum Sustained Read Data Rate	b2 b1 b0 0 0 0: Maximum value of f_{SCL} 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Settings other than above are prohibited.	R/W
b5 to b3	TSCO[2:0]	Clock-to-Data Turnaround Time	b5 b4 b3 0 0 0: 8 ns or less 0 0 1: 9 ns or less 0 1 0: 10 ns or less 0 1 1: 11 ns or less 1 0 0: 12 ns or less 1 1 1: t_{SCO} is more than 12 ns, and is reported by private agreement. Settings other than above are prohibited.	R/W
b31 to b6	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the maximum read speed to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b7 to b0 in this register is automatically transmitted as MaxRd when a GETMXDS CCC (Direct) is received.

43.2.79 Maximum Read Turnaround Time Register (ICMTTR)

Address(es): RI3C0.ICMTTR 8745 F370h, RI3C0_NS.ICMTTR 0745 F370h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRTTE	—	—	—	—	—	—	—	MRTT[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRTT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	MRTT[23:0]	Maximum Read Turnaround Time	Set the read turnaround time in 1 µs units in the range of 0 to 16 seconds. 000000h: 0 µs (minimum value) 000001h: 1 µs : : F42400h: 16 seconds (maximum value) Settings other than above are prohibited.	R/W
b30 to b24	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	MRTTE	Maximum Read Turnaround Time Transmit Enable	0: Disables transmission of the maximum read turnaround time. (GETMXDS format 1: without turnaround) 1: Enables transmission of the maximum read turnaround time. (GETMXDS format 2: with turnaround)	R/W

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the three-byte maximum read turnaround time to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b23 to b0 in this register is automatically transmitted as MaxRdTurn when a GETMXDS CCC (Direct) is received.

43.2.80 Timing Support Information Register (ICTSIR)

Address(es): RI3C0.ICTSIR 8745 F374h, RI3C0_NS.ICTSIR 0745 F374h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	INAC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FREQ[7:0]								—	—	—	—	—	SAM1	SAM0	SSM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSM	Supports Sync Mode	0: Sync Mode is not supported. 1: Sync Mode is supported.	R/W
b1	SAM0	Supports Async Mode 0	0: Async Mode 0 is not supported. 1: Async Mode 0 is supported.	R/W
b2	SAM1	Supports Async Mode 1	0: Async Mode 1 is not supported. 1: Async Mode 1 is supported.	R/W
b7 to b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b15 to b8	FREQ[7:0]	Frequency Setting	These bits specify the Target's internal oscillation frequency in increments of 0.5 MHz. b15 b8 00000000: Approximately 32.0 kHz 00000001: 0.5 MHz 00000010: 1.0 MHz : : 11111101: 126.5 MHz 11111110: 127.0 MHz 11111111: 127.5 MHz	R/W
b23 to b16	INAC[7:0]	Frequency Inaccuracy Setting	These bits specify the inaccuracy of the oscillation frequency from 0.0% to 25.5% in increments of 0.1%. b23 b16 00000000: 0.0% 00000001: 0.1% : : 11111110: 25.4% 11111111: 25.5%	R/W
b31 to b24	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the value to be returned when the timing information is requested from I3C Controller. When a GETXTIME CCC (Direct) is received, the values of bits b7 to b0, b15 to b8, and b23 to b16 in this register are automatically transmitted as Supported Mode Byte, Frequency Byte, and Inaccuracy Byte, respectively.

43.2.81 Timing Support Information Register 2 (ICTSIR2)

Address(es): RI3C0.ICTSIR2 8745 F378h, RI3C0_NS.ICTSIR2 0745 F378h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OVF	—	—	—	—	AM1E	AM0E	SME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SME	Sync Mode Enabled	0: Sync Mode is disabled. 1: Sync Mode is enabled.	R/W
b1	AM0E	Async Mode 0 Enabled	0: Async Mode 0 is disabled. 1: Async Mode 0 is enabled.	R/W
b2	AM1E	Async Mode 1 Enabled	0: Async Mode 1 is disabled. 1: Async Mode 1 is enabled.	R/W
b6 to b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b7	OVF	Overflow Flag	0: No counter overflow has occurred since the last check. 1: Counter overflow has occurred since the last check.	R/W
b31 to b8	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the value to be returned when the timing information is requested from I3C Controller. When a GETXTIME CCC (Direct) is received, the value of bits b7 to b0 in this register is automatically transmitted as State Byte.

When receiving SETXTIME CCC (Broadcast) or SETXTIME CCC (Direct) with its own Target address, the AM0E and AM1E bits are automatically updated according to the received Sub-Command Byte value (DFh: Enter Async Mode 0, EFh: Enter Async Mode 1).

43.2.82 Bit Count Register (ICBCR)

Address(es): RI3C0.ICBCR 8745 F380h, RI3C0_NS.ICBCR 0745 F380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BC[4:0]	Bit Counter	These bits indicate the number of bits remaining to be transferred. Refer to Table 43.6 and Table 43.7 for details.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

BC[4:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL00 line.

Table 43.6 Counter Value and Number of Remaining Bits (I^2C/Legacy I^2C Transfer)

	Master/Controller		Slave/Target	
BC[4:0]	Address Phase	Data Phase	Address Phase	Data Phase
00h	2 or 1 bits	2 or 1 bits	3 to 1 bits	2 or 1 bits
01h	3 bits	3 bits	4 bits	3 bits
02h	4 bits	4 bits	5 bits	4 bits
03h	5 bits	5 bits	6 bits	5 bits
04h	6 bits	6 bits	7 bits	6 bits
05h	7 bits	7 bits	8 bits	7 bits
06h	8 bits	8 bits	9 bits	8 bits
07h	9 bits	9 bits	—	9 bits

Table 43.7 Counter Value and Number of Remaining Bits (I3C Transfer)

	SDR*1	
BC[4:0]	Transmission	Reception
00h	1 bit	2 or 1 bits
01h	2 bits	3 bits
02h	3 bits	4 bits
03h	4 bits	5 bits
04h	5 bits	6 bits
05h	6 bits	7 bits
06h	7 bits	8 bits
07h	8 bits	9 bits
08h	9 bits	—

Note 1. The address phase is the same as in Table 43.6.

43.2.83 Queue Buffer Status Register (ICQBSR)

Address(es): RI3C0.ICQBSR 8745 F394h, RI3C0_NS.ICQBSR 0745 F394h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	ISC[4:0]					IQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RQFL[7:0]								CQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CQFL[7:0]	Command Queue Free Level	Number of free entries currently in the command queue.	R
b15 to b8	RQFL[7:0]	Response Queue Fill Level	Number of entries currently in the response queue.	R
b23 to b16	IQFL[7:0]	IBI Queue Fill Level	Number of entries currently in the IBI queue.	R
b28 to b24	ISC[4:0]	IBI Status Count*1	Number of IBI status entries currently in the IBI queue.	R
b31 to b29	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C protocol mode.
Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.84 Data Buffer Status Register (ICDBSR)

Address(es): RI3C0.ICDBSR 8745 F398h, RI3C0_NS.ICDBSR 0745 F398h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBFL[7:0]								TBFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TBFL[7:0]	Transmit Data Buffer Free Level	Indicates the free level of the transmit data buffer.	R
b15 to b8	RBFL[7:0]	Receive Data Buffer Fill Level	Indicates the fill level of the receive data buffer.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C protocol mode.

43.2.85 Receive Status Queue Status Register (ICSQSR)

Address(es): RI3C0.ICSQSR 8745 F3C0h, RI3C0_NS.ICSQSR 0745 F3C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SQFL[7:0]	Receive Status Queue Fill Level	Number of entries currently in the receive status queue.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

43.2.86 High-Priority Queue Buffer Status Register (ICHQBSR)

Address(es): RI3C0.ICHQBSR 8745 F3C4h, RI3C0_NS.ICHQBSR 0745 F3C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HRQFL[7:0]								HCQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	HCQFL[7:0]	High-Priority Command Queue Free Level	Number of free entries currently in the high-priority command queue.	R
b15 to b8	HRQFL[7:0]	High-Priority Response Queue Fill Level	Number of entries currently in the high-priority response queue.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.87 High-Priority Data Buffer Status Register (ICHDBSR)

Address(es): RI3C0.ICHDBSR 8745 F3C8h, RI3C0_NS.ICHDBSR 0745 F3C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HRBFL[7:0]								HTBFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	HTBFL[7:0]	High-Priority Transmit Data Buffer Free Level	Indicates the free level of the high-priority transmit data buffer.	R
b15 to b8	HRBFL[7:0]	High-Priority Receive Data Buffer Fill Level	Indicates the fill level of the high-priority receive data buffer.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.88 Internal Status Monitor Register (ICIMR)

Address(es): RI3C0.ICIMR 8745 F3CCh, RI3C0_NS.ICIMR 0745 F3CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SDAO	SCLO	SDAI	SCLI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SCLI	SCL Line Monitor Flag	0: SCL00 line is low. 1: SCL00 line is high.	R
b1	SDAI	SDA Line Monitor Flag	0: SDA00 line is low. 1: SDA00 line is high.	R
b2	SCLO	SCL Output Monitor Flag	0: The RI3C drives the SCL00 pin low. 1: The RI3C releases the SCL00 pin.	R
b3	SDAO	SDA Output Monitor Flag	0: The RI3C drives the SDA00 pin low. 1: The RI3C releases the SDA00 pin.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

SCLO Flag (SCL Output Monitor Flag) and SDAO Flag (SDA Output Monitor Flag)

When reading these bits, the state of signals output from the RI3C can be read.

43.2.89 Controller Error Count Register (ICCECR)

Address(es): RI3C0.ICCECR 8745 F3D0h, RI3C0_NS.ICCECR 0745 F3D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CE2C[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CE2C[7:0]	CE2 Error Counter	These bits counts I3C Type CE2 errors on the I3C bus. These bits are cleared when read.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

43.2.90 T_C1 Capture Register (ICTC1CR)

Address(es): RI3C0.ICTC1CR 8745 F3E0h, RI3C0_NS.ICTC1CR 0745 F3E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The ICTC1CR register is a 32-bit read-only register used to read the captured T_C1 counter value. This register is used in Async Mode 0 and Async Mode 1.
The upper 16 bits are reserved and the read value is 0.

43.2.91 T_C2 Capture Register (ICTC2CR)

Address(es): RI3C0.ICTC2CR 8745 F3E4h, RI3C0_NS.ICTC2CR 0745 F3E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The ICTC2CR register is a 32-bit read-only register used to read the captured T_C2 counter value. This register is used in Async Mode 0 and Async Mode 1.
The upper 16 bits are reserved and the read value is 0.

43.2.92 Communication Clock Control Register (ICCCR)

Address(es): RI3C0.ICCCR 8745 F010h, RI3C0_NS.ICCCR 0745 F010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCE	Communication Clock Enable	0: Clock disable 1: Clock enable	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

CCE Bit (Communication Clock Enable)

This bit enables or disables the clock supply to the transmit/receive circuits.
Set this bit to 1 before setting the ICCR.ICE bit to 1 (the I3C bus operation is enabled), and set this bit to 0 after setting the ICE bit to 0 (the I3C bus operation is disabled).

43.3 Data Structures

43.3.1 Command Descriptor

The Command Descriptor is 64 bits in length. The Command Descriptor is added to the command queue via writes to the command queue register (ICCQR).

Write to the ICCQR register in the following order:

1. First write: The lower 32 bits of Command Descriptor
2. Second write: The upper 32 bits of Command Descriptor.

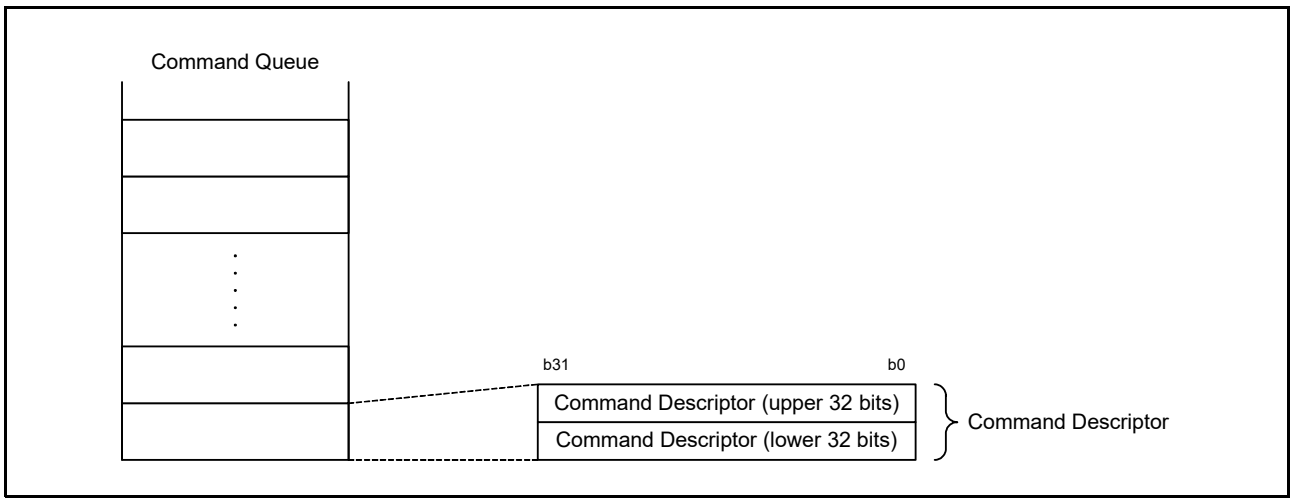


Figure 43.3 Command Descriptor Data Structure

RI3C provides a Command Descriptor structure for each command type as follows:

- Address Assignment Command
- Immediate Data Transfer Command
- Regular Data Transfer Command
- Combo Transfer Command
- Internal Control Command

43.3.1.1 Address Assignment Command

This command is used for Address Assignment (ENTDAA, SETDASA)*1.
Details of the address assignment command structure are as follows.

Note 1. When issuing SETAASA CCC, use the immediate data transfer command.

	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOC	ROC	DEV_COUNT				—	—	—	—	EXT_DEVICE	DEV_INDEX				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMD								TID				CMD_ATTR		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attributes	Set this field to 010b (address assignment command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used. ENTDAA (Broadcast): 07h SETDASA (Direct): 87h	W
b15	—	Reserved	Set this field to 0.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being assigned an address via Dynamic Address Assignment.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm register indicated by the DEV_INDEX field. 1: Use the ICEDATR register.	W
b25 to b22	—	Reserved	The write value should be 0.	W
b29 to b26	DEV_COUNT	Device Count	Specify the number of devices that a Dynamic Address shall be assigned to.	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b63 to b32	—	Reserved	The write value should be 0.	W

TID Field (Transaction ID)

The value specified in this field is reflected in the TID field of the Response Descriptor.

DEV_INDEX Field (Device Index)

Indicates the DAT table index for the Target device being assigned an address via Dynamic Address Assignment.
Write a Dynamic Address into each DAT entry (the ICTDATRm registers) that will be used with this command type, as well as a Static Address (when used with the SETDASA CCC).

DEV_COUNT Field (Device Count)

Indicates the number of devices that a Dynamic Address is assigned to. The DAT entries are processed starting at index DEV_INDEX for up to DEV_COUNT times.

ROC Field (Response on Completion)

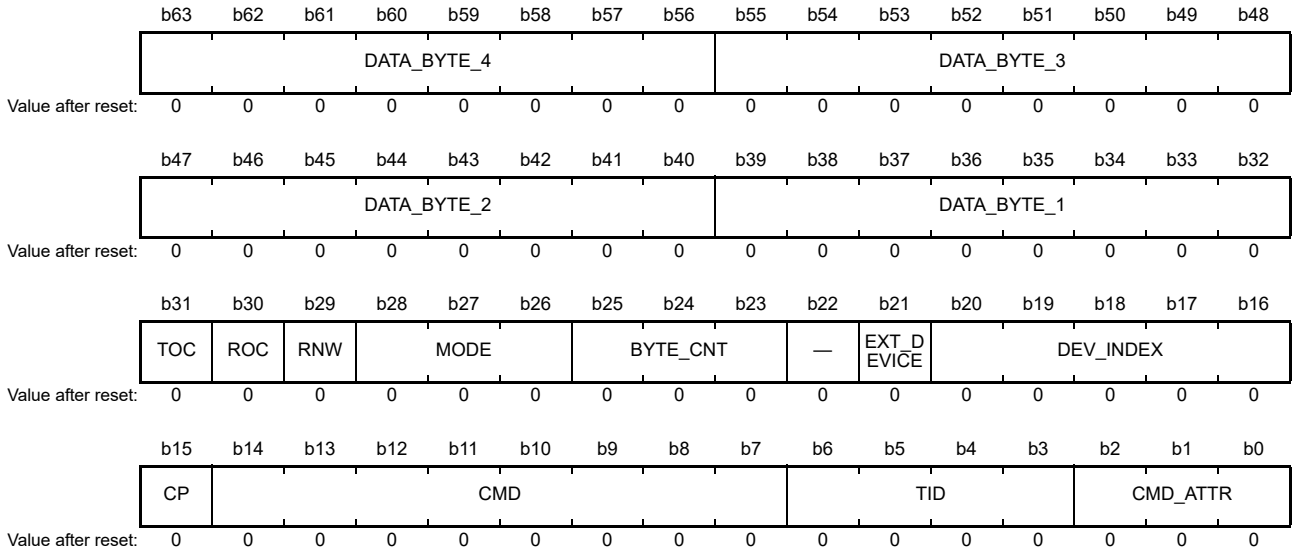
Controls whether Response status is sent after successful completion of the transfer command. The successful completion shall be read from the ICRQR register. Upon unsuccessful transfer the Response status shall always be sent.

TOC Field (Terminate on Completion)

Controls what bus condition to issue after the transfer command completes.
For ENTDAA, a STOP condition is issued regardless of the value set in this field.
When sending SETDASA CCC with the TOC field set to 0 (RESTART), the next command must also be set to an address assignment command specifying SETDASA CCC. When the next command is not SETDASA CCC, the TOC field must be set to 1 (STOP).

43.3.1.2 Immediate Data Transfer Command

This Command Descriptor directly contains data bytes (up to 4 bytes) to be transferred, and as a result is only useful for shorter write-type transfers or CCCs that write data. This command shall not be used for read-type transfers.
Details of the immediate data transfer command structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 001b (immediate data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used.	W
b15	CP	Command Present	0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b22	—	Reserved	The write value should be 0.	W
b25 to b23	BYTE_CNT	Byte Count	Number of valid data bytes to use in this descriptor. Specify in the range of 0h (no payload) to 4h (4 bytes).	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 0: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 0 1 1: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 0: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I^2C transfer b28 b26 0 0 0: I^2C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I^2C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W
b29	RNW	Transfer Direction (RnW)	Set this field to 0 (write transfer)	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W

Bit	Symbol	Bit Name	Description	R/W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b39 to b32	DATA_BYTE_1	Data Byte 1	First byte of data to be transmitted	W
b47 to b40	DATA_BYTE_2	Data Byte 2	Second byte of data to be transmitted	W
b55 to b48	DATA_BYTE_3	Data Byte 3	Third byte of data to be transmitted	W
b63 to b56	DATA_BYTE_4	Data Byte 4	Fourth byte of data to be transmitted	W

DEV_INDEX Field (Device Index)

Indicates the DAT table (the ICTDATRm register) index for the Target device being addressed with the transfer. The ICTDATRm register indicated by this field must contain a valid Dynamic Address. This field is ignored for Broadcast CCCs.

BYTE_CNT Field (Byte Count)

Number of valid data bytes to use in this descriptor. This field should be set to non-zero value, except for CCCs that do not have any payload defined.

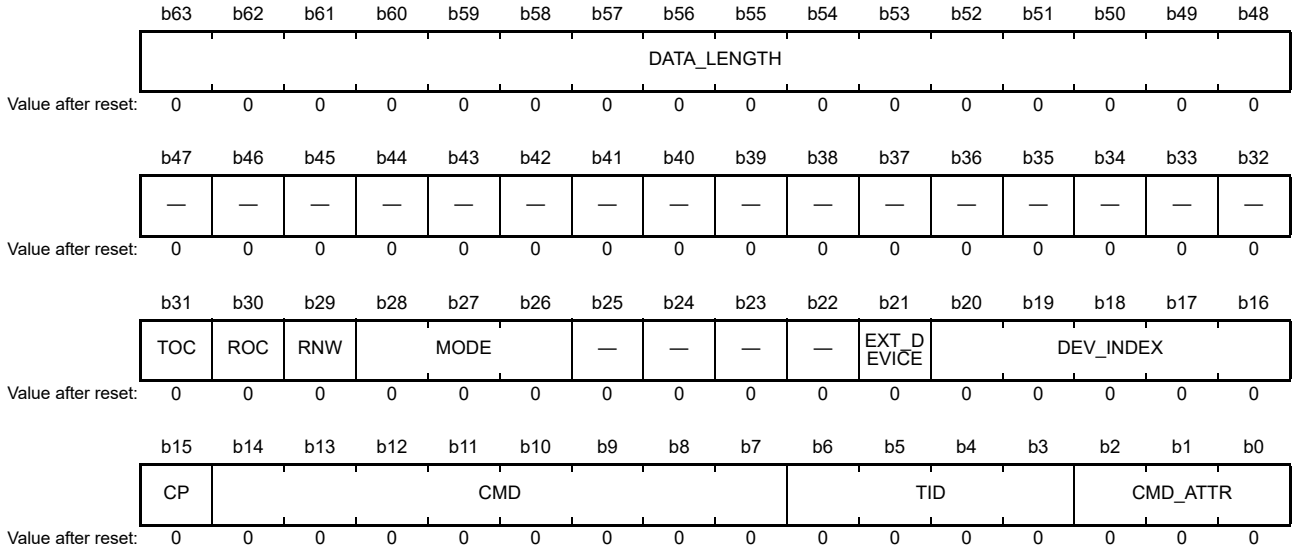
MODE Field (Mode and Speed)

Sets the mode and speed for the I3C or I^2C transfer.
Interpretation of this field depends on the TYPE bit in the ICTDATRm register indexed by the DEV_INDEX field.

43.3.1.3 Regular Data Transfer Command

This Command Descriptor does not contain data to be transferred. For I3C Controller mode, data is transferred via the transmit/receive data register (ICDR), and for I3C Target mode, data is transferred via the IBI queue register (ICIQR). Details of the regular data transfer command structure of each mode are as follows.

(1) I3C Controller Mode



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 000b (regular data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used.	W
b15	CP	Command Present	0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b25 to b22	—	Reserved	The write value should be 0.	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 0: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 0 1 1: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 0: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I^2C transfer b28 b26 0 0 0: I^2C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I^2C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer 1: READ: Read transfer	W

Bit	Symbol	Bit Name	Description	R/W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b47 to b32	—	Reserved	The write value should be 0.	W
b63 to b48	DATA_LENGTH	Data Length	Indicates the number of bytes to be transferred.	W

RNW Field (Transfer Direction (RnW))

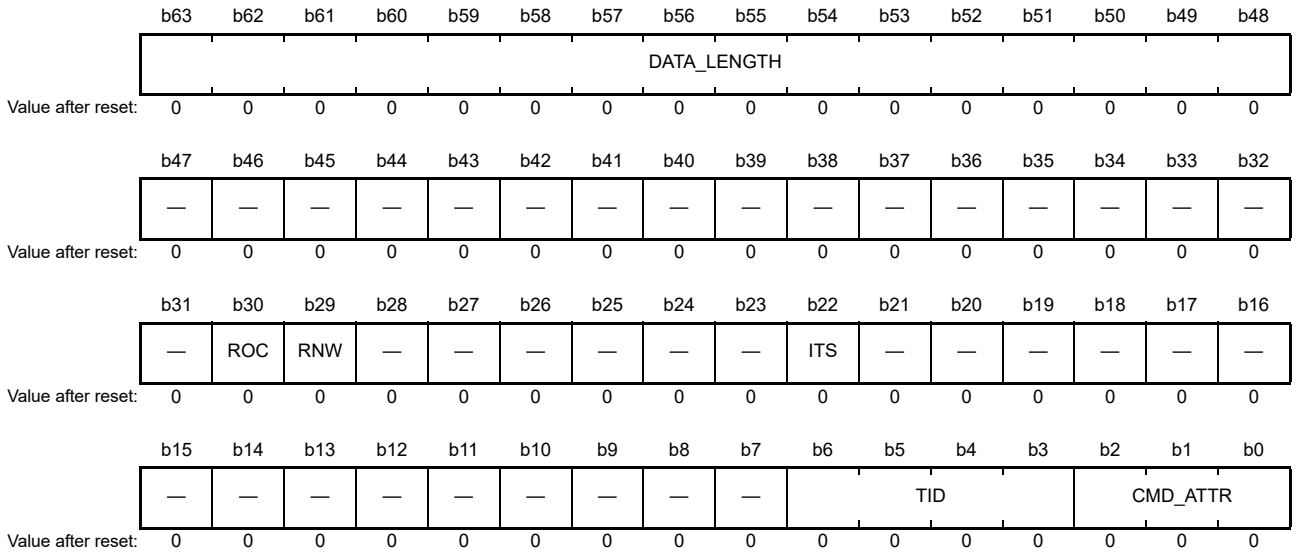
Identifies the direction of this transfer.

DATA_LENGTH Field (Data Length)

Number of bytes to be transferred.

This field should be set to a non-zero value, except for CCCs that do not have payload defined. Length setting of GETMXDS command should be fixed to “0005h”.

(2) I3C Target Mode



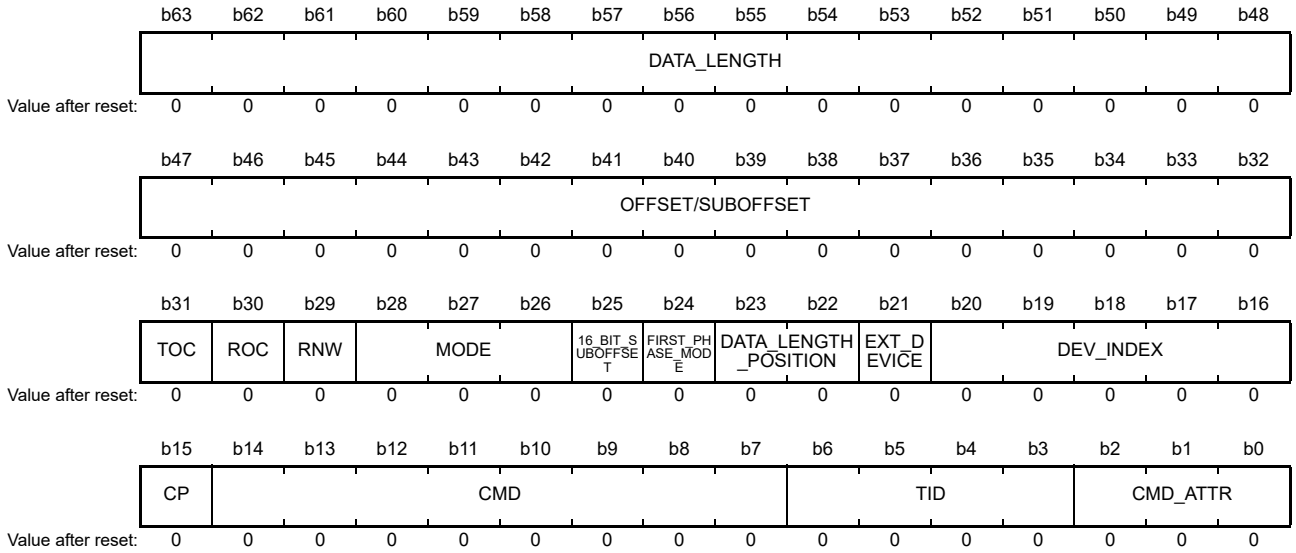
Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 000b (regular data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b21 to b7	—	Reserved	The write value should be 0.	W
b22	ITS	Include Timestamp	0: Do not include timestamp. 1: Include timestamp.	W
b28 to b23	—	Reserved	The write value should be 0.	W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer (Controller Role Request) 1: READ: Read transfer (Target Interrupt Request)	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b47 to b31	—	Reserved	The write value should be 0.	W
b63 to b48	DATA_LENGTH	Data Length	Indicates the number of bytes to be transferred.	W

RNW Field (Transfer Direction (RnW))

Identifies the direction of this transfer.

43.3.1.4 Combo Transfer Command

This Command Descriptor describes a combined Write + Read or Write + Write operation.
Data is transferred via the transmit/receive data register (ICDR).
Details of the combo transfer command structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 011b (combo transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	This field is reserved for this command. Set it to 00h.	W
b15	CP	Command Present	Set this field to 0 for this command.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b23, b22	DATA_LENGTH_POSITION	Data Length Field Position	Set this field to 00b.	W
b24	FIRST_PHASE_MODE	First Phase Mode	0: SDR: First phase is executed in SDR mode. 1: MODE: First phase is executed in the mode indicated by the MODE field.	W
b25	16_BIT_SUBOFFSET	Sub Offset Size	0: 8_BIT_SUBOFFSET: Sub-offset is 8 bits long. Value is encoded in lower byte of OFFSET/SUBOFFSET field. 1: 16_BIT_SUBOFFSET: Sub-offset is 16 bits long.	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 0: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 0 1 1: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 0: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I²C transfer b28 b26 0 0 0: I²C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I²C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W

Bit	Symbol	Bit Name	Description	R/W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer 1: READ: Read transfer	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b47 to b32	OFFSET/ SUBOFFSET	Offset/Sub-Offset	Set the value of the Offset field to be transferred in the first phase.	W
b63 to b48	DATA_LENGTH	Data Length	Specify the number of bytes to be transferred in the second phase. This field must be set to non-zero value.	W

DATA_LENGTH_POSITION Field (Data Length Field Position)

Indicates whether and where to put Data Length (DATA_LENGTH) in the first phase of the transfer.

FIRST_PHASE_MODE Field (First Phase Mode)

Indicates whether the first phase of the combo transfer is executed in SDR mode, vs. the mode indicated by the MODE field.

16_BIT_SUBOFFSET Field (Sub Offset Size)

Indicates whether the length of the Offset and Length fields in the first phase of the transfer is 8 bits or 16 bits. For 8-bit, data should be written in lower bytes of the OFFSET/SUBOFFSET and DATA_LENGTH fields.

RNW Field (Transfer Direction (RnW))

Identifies the direction of the second phase of the transfer.

43.3.1.5 Internal Control Command

This Command Descriptor is used for controlling RI3C itself (not for I3C transfer commands).
Details of the internal control command structure are as follows.

	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ON_OFF	MIPI_CMD				—	TID				CMD_ATTR		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 111b (internal control command).	W
b6 to b3	TID	Transaction ID*1	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b7	—	Reserved	The write value should be 0.	W
b11 to b8	MIPI_CMD	MIPI Alliance Command	b11 b8 0 0 0 0: NoOp (performs no transfer) 0 0 1 0: Broadcast Address automatic transmission Settings other than above are prohibited.	W
b12	ON_OFF	On/Off	0: disables automatic transmission of Broadcast Address 1: enables automatic transmission of Broadcast Address*2	W
b63 to b13	—	Reserved	The write value should be 0.	W

Note 1. The Response Descriptor is not stored when the internal control command is executed.
Note 2. Automatic transmission of Broadcast Address enabled by this command is disabled when the ICRCR.ISRST is set to 1.

43.3.2 Response Descriptor

The Response Descriptor is a 32-bit read-only structure describing the success or failure of a command, and the amount of data transferred.
The Response Descriptor is read from the response queue via the response queue register (ICRQR).

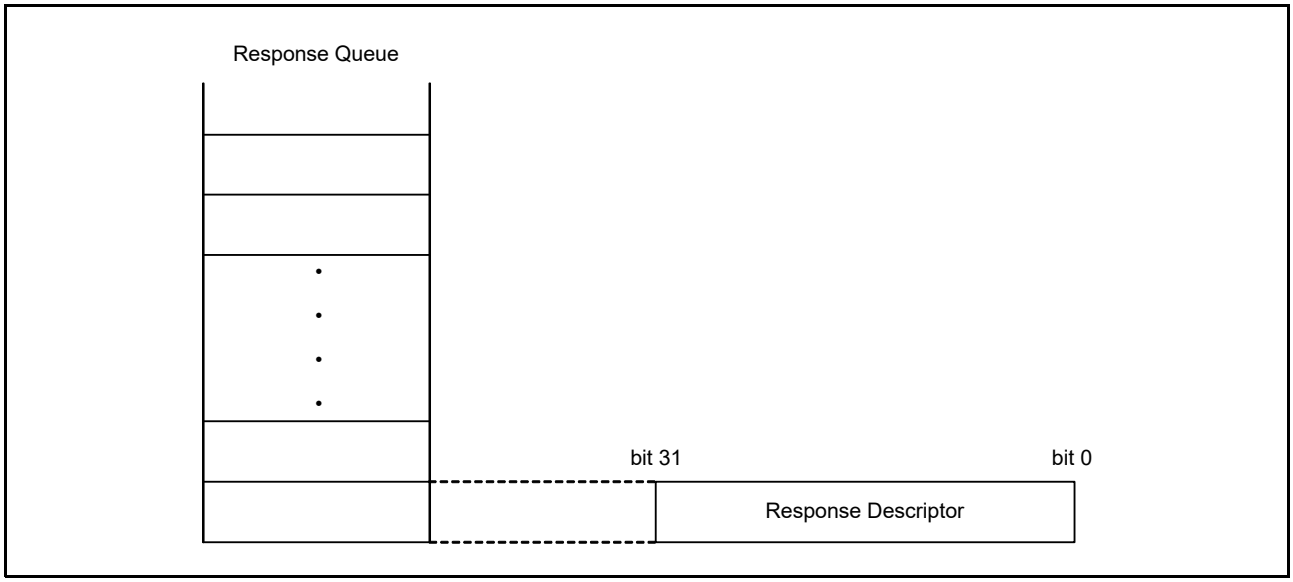
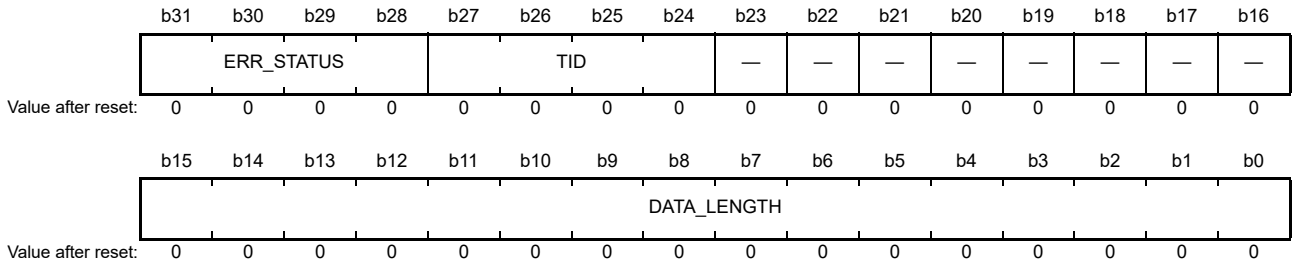


Figure 43.4 Response Descriptor Data Structure

The Response Descriptor structure differs between I3C Controller mode and I3C Target mode.
Details of the Response Descriptor structure of each mode are as follows.

(1) I3C Controller Mode



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length/Device Count	For write transfer: Remaining data length (in bytes) For read transfer: Received data length (in bytes) For address assignment: Remaining device count	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b27 to b24	TID	Transaction ID	This value shall match the value of the TID field, for a previously enqueued Command Descriptor that was sent on the bus.	R
b31 to b28	ERR_STATUS	Error Status	b31 b28 0 0 0 0: SUCCESS: Transfer successful, no error 0 1 0 0: ADDR_HEADER: Address header error 0 1 0 1: NACK: Address NACKed or Dynamic Address Assignment NACKed 0 1 1 0: OVL: Receive overflow or transmit underflow error 1 0 0 0: ABORTED: Transfer aborted 1 0 0 1: I2C_WR_DATA_NACK: NACK is received during an I^2C write data transfer 1 0 1 0: NOT_SUPPORTED: Command is not supported	R

TID Field (Transaction ID)

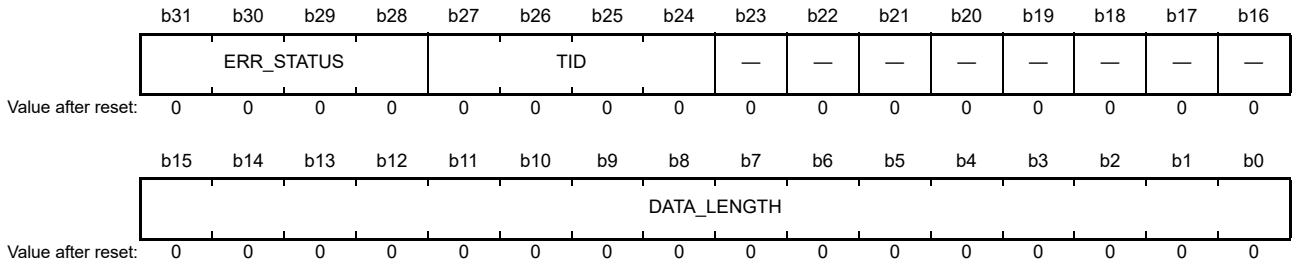
This field reflects the value specified in the TID field of the Command Descriptor. Software can determine which command the response is for by checking this field.

ERR_STATUS Field (Error Status)

This field indicates the transfer status of the command indicated by the TID field.
The details of each error are as follows.

- ADDR_HEADER: No Target devices acknowledge the Broadcast Address.
- NACK: Any Target device that responds to the ENTDAA procedure does not accept the Dynamic Address or an addressed Target device does not acknowledge its Dynamic Address.
- OVL: Transmit data buffer underflowed or receive data buffer overflowed.
- ABORTED: A transaction was aborted.
- I2C_WR_DATA_NACK: The I^2C Target device does not acknowledge the I^2C write data transfer.
- NOT_SUPPORTED: A Command Descriptor structure with an illegal or invalid combination of field values was stored.

(2) I3C Target Mode



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length	For Target Interrupt Request: Remaining data length (in bytes)	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b27 to b24	TID	Transaction ID	This value shall match the value of the TID field, for a previously enqueued Command Descriptor that was sent on the bus.	R
b31 to b28	ERR_STATUS	Error Status	b31 b28 0 0 0 0: SUCCESS: Transfer successful, no error 0 1 0 0: ADDR_HEADER: Address header error 0 1 0 1: NACK: Address NACKed or Dynamic Address Assignment NACKed 0 1 1 0: OVL: Receive overflow or transmit underflow error 1 0 0 0: ABORTED: Transfer aborted 1 0 1 0: NOT_SUPPORTED: Command is not supported	R

ERR_STATUS Field (Error Status)

This field indicates the transfer status of the command indicated by the TID field.
The details of each error are as follows.

- ADDR_HEADER: No Target devices acknowledge the Broadcast Address.
- NACK: Any Target device that responds to the ENTDAA procedure does not accept the Dynamic Address or an addressed Target device does not acknowledge its Dynamic Address.
- OVL: Transmit data buffer underflowed or receive data buffer overflowed.
- ABORTED: A transaction was aborted.
- NOT_SUPPORTED: A Command Descriptor structure with an illegal or invalid combination of field values was stored, an IBI disabled by the ICTEVR register is requested, or an IBI is disabled by DISEC CCC after a Command Descriptor structure for the IBI was enqueued in the command queue.

43.3.3 IBI Status Descriptor

The IBI Status Descriptor is a 32-bit read-only structure describing an IBI event received from a Target device on the I3C bus. This Descriptor is used in I3C Controller mode.
The IBI Status Descriptor is read from the IBI queue via the IBI queue register (ICIQR).

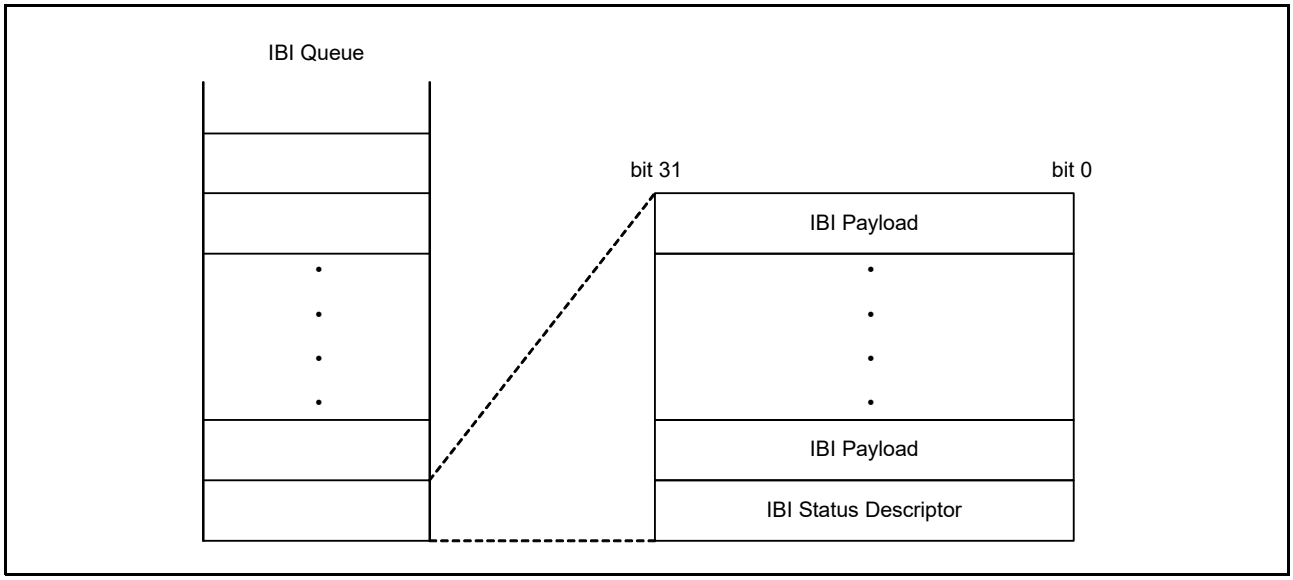


Figure 43.5 IBI Status Descriptor Data Structure

Details of the IBI Status Descriptor Structure are as follows.

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IBI_STS	—	—	ERR_STATUS			TS	LAST_STATUS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IBI_ID								DATA_LENGTH							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA_LENGTH	IBI Data Length	Number of data bytes in IBI data.	R
b15 to b8	IBI_ID	IBI Received ID	• Target Interrupt Request or Controller Role Request: b15 to b9: Target's Device Address b8: Value of RnW bit	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b24	LAST_STATUS	Last IBI Status	0: This is not the last IBI Status Descriptor. 1: This is the last IBI Status Descriptor.	R
b25	TS	IBI Timestamp Present	Indicates whether a timestamp is available for the IBI. 0: OFF: IBI is not timestamped. 1: ON: IBI is timestamped.	R
b28 to b26	ERR_STATUS	IBI Error Status	b28 b26 0 0 0: SUCCESS 0 1 1: FRAME: Frame error 1 0 0: ADDR_HEADER: Address header error 1 0 1: NACK: Address NACKed 1 1 1: ABORT: Aborted to Controller	R

Bit	Symbol	Bit Name	Description	R/W
b30, b29	—	Reserved	These bits are read as 0.	R
b31	IBI_STS	IBI Received Status	0: The IBI was handled with ACK. 1: The IBI was handled with NACK, and then Auto-Disabled.	R

LAST_STATUS Field (Last IBI Status)

For some IBIs, multiple IBI Status Descriptors might be needed to describe a single IBI event. A 0 in this field indicates that there is a subsequent IBI Status Descriptor.
Even if this field is set to 0, examine the DATA_LENGTH field to evaluate the data payload length.

IBI_STS Field (IBI Received Status)

Indicates how the received IBI was handled by Active Controller.

43.3.4 Receive Status Descriptor

The Receive Status Descriptor is a 32-bit read-only structure describing the success or failure of read/write transfer issued by the Controller, and the amount of data transferred. This Descriptor is used in I3C Target mode.
The Receive Status Descriptor is read from the receive status queue via the receive status queue register (ICSQR).

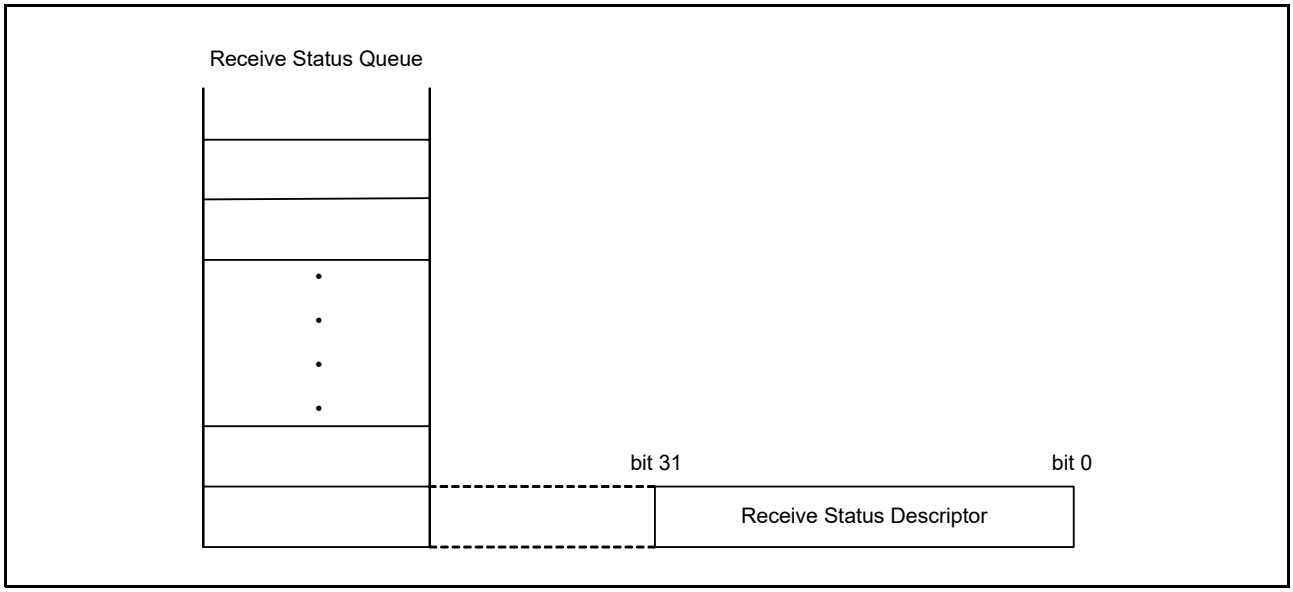
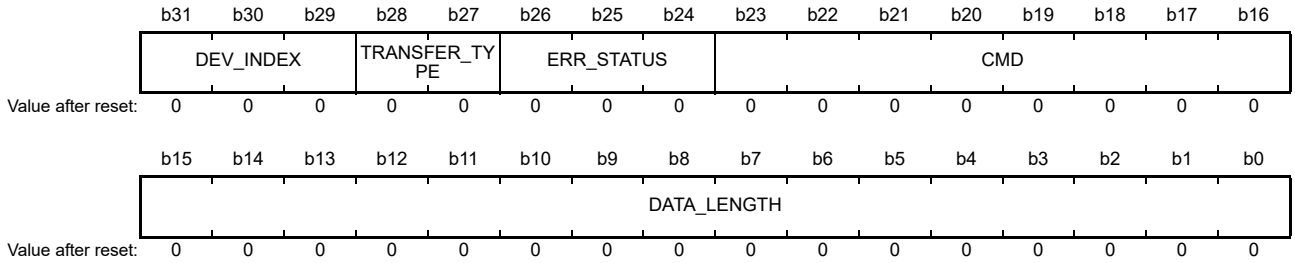


Figure 43.6 Receive Status Descriptor Data Structure

Details of the Receive Status Descriptor structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length	For Write Transfer: Received data length (in bytes) For Read Transfer: Transmitted data length (in bytes)	R
b23 to b16	CMD	Command Code	• I3C SDR private transfer or Legacy I^2C transfer: b23: Transfer type (0: Write/1: Read) b22 to b20: Reserved b19: Transfer mode (0: I3C SDR/1: Legacy I^2C) b18 to b16: Reserved • SDR CCC transfer: Indicates the CCC command code specified in the Command Descriptor of the transfer.	R
b26 to b24	ERR_STATUS	Error Status	b26 b24 0 0 0: SUCCESS: Transfer successful, no error 1 0 0: ADDR_HEADER: Address header error 1 0 1: NACK: Target NACKed 1 1 0: OVL: FIFO overflow/underflow 1 1 1: ABORTED: Aborted to Controller	R

Bit	Symbol	Bit Name	Description	R/W
b28, b27	TRANSFER_TYPE	Transfer Type	b28 b27 0 0: I3C SDR private transfer or Legacy I^2C transfer 0 1: I3C CCC	R
b31 to b29	DEV_INDEX	Device Index	Indicates the DAT table index (m of the ICTDATRm register) for the responded device with the transfer.	R

43.4 Operation

43.4.1 Transfer Methods

The relationship between the transfer method and the queue buffer/data buffer is shown in Table 43.8.

Table 43.8 Transfer Method and Size of Queue/Buffer

Protocol Mode	Transfer Method	Queue/Buffer	Size	Controller	Target	Secondary Controller
I3C	FIFO buffer transfer	Command Queue	64 bits × 4 stages	✓	✓	✓
		Response Queue	32 bits × 4 stages	✓	✓	✓
		Transmit Data Buffer	32 bits × 16 stages	✓	✓	✓
		Receive Data Buffer	32 bits × 16 stages	✓	✓	✓
		Receive Status Queue	32 bits × 2 stages	—	✓	✓
		IBI Status Queue	32 bits × 2 stages	✓	—	✓
		IBI Data Buffer	32 bits × 8 stages	✓	✓	✓
	High-priority FIFO buffer transfer	High-priority Command Queue	32 bits × 2 stages	✓	—	✓
		High-priority Response Queue	32 bits × 2 stages	✓	—	✓
		High-priority Transmit Data Buffer	32 bits × 2 stages	✓	—	✓
		High-priority Receive Data Buffer	32 bits × 2 stages	✓	—	✓
I^2C	Single buffer transfer	Transmit Data Buffer	1 byte	✓	✓	—
		Receive Data Buffer	1 byte	✓	✓	—

(1) FIFO Buffer Transfer

RI3C autonomously starts transfer when data and command are written.

Figure 43.7 shows transactions using FIFO buffer transfer. In the I3C bus protocol, transactions are initiated by the Controller. The Target transmits and receives data according to the command sent by the Controller.

- Controller transmission
  When data and a command descriptor is written to the ICDR and ICCQR registers, respectively, a write transaction is issued on the I3C bus.
- Controller reception
  Writing a command descriptor to the ICCQR register issues a read transaction on the I3C bus. The data sent by the Target is stored in the receive data buffer. Use the ICDR register to read the received data.
- Target transmission
  When the data requested by the read transaction issued by the Controller is written to the ICDR register, the data is transmitted.
- Target reception
  The data is stored in the receive data buffer by a write transaction issued by the Controller. Use the ICDR register to read the received data.

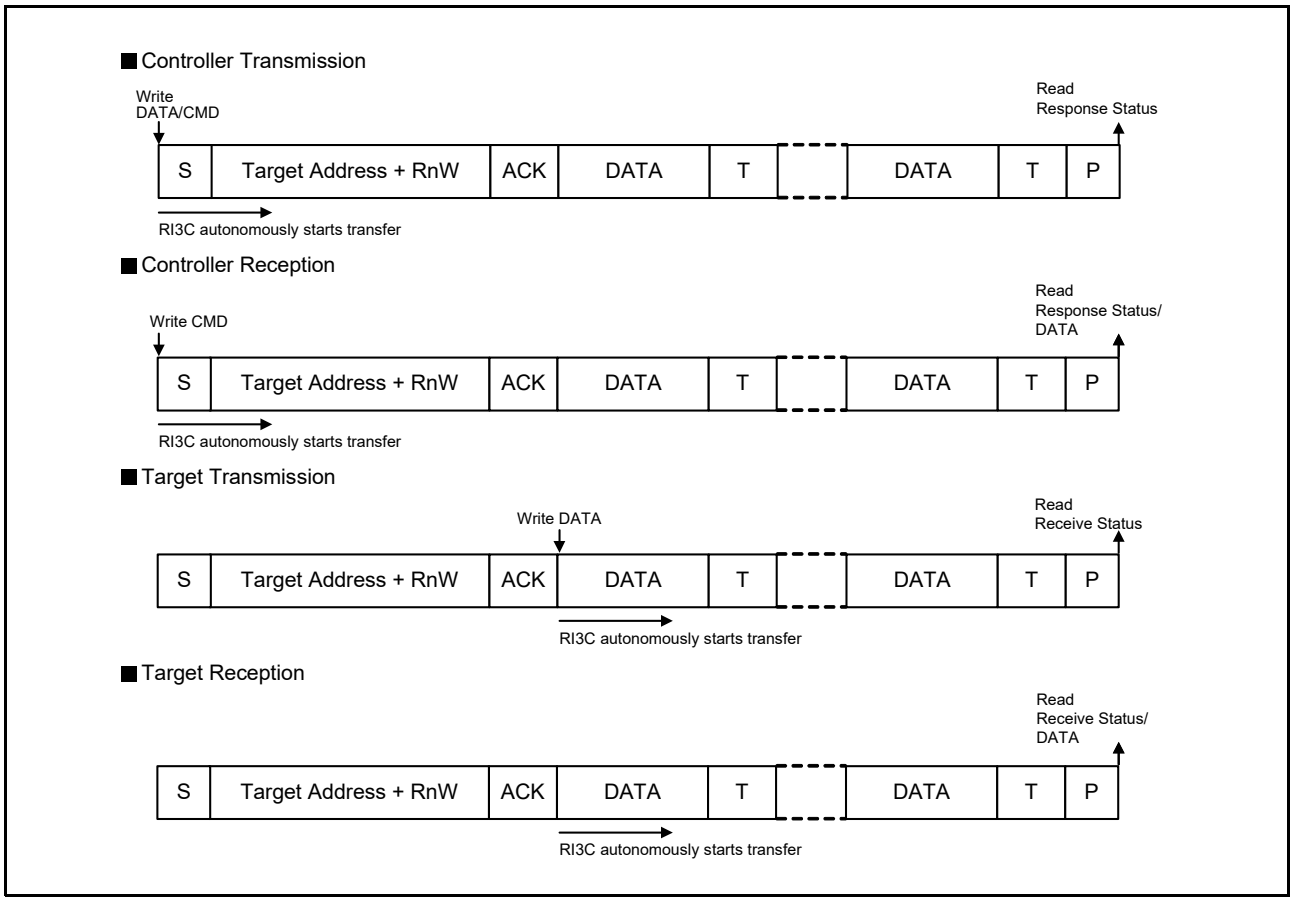


Figure 43.7 Data Handling with FIFO Buffer Transfer

(2) High-Priority FIFO Buffer Transfer

High-priority FIFO buffer transfers are only available in Controller mode, and take priority over normal FIFO buffer transfers. If a normal FIFO buffer transfer is already being executed, the high-priority FIFO buffer transfer is performed after waiting for the STOP condition for the currently executed transfer. The interrupted FIFO buffer transfer is resumed after the high-priority FIFO buffer transfer is completed.

The transactions using high-priority FIFO buffer transfer are the same as the Controller transmission and Controller reception in Figure 43.7.

(3) Single Buffer Transfer

In single buffer transfer, all operation such as issuing conditions, transferring data, and ACK/NACK responses are controlled by software. Figure 43.8 shows transactions using single buffer transfer.

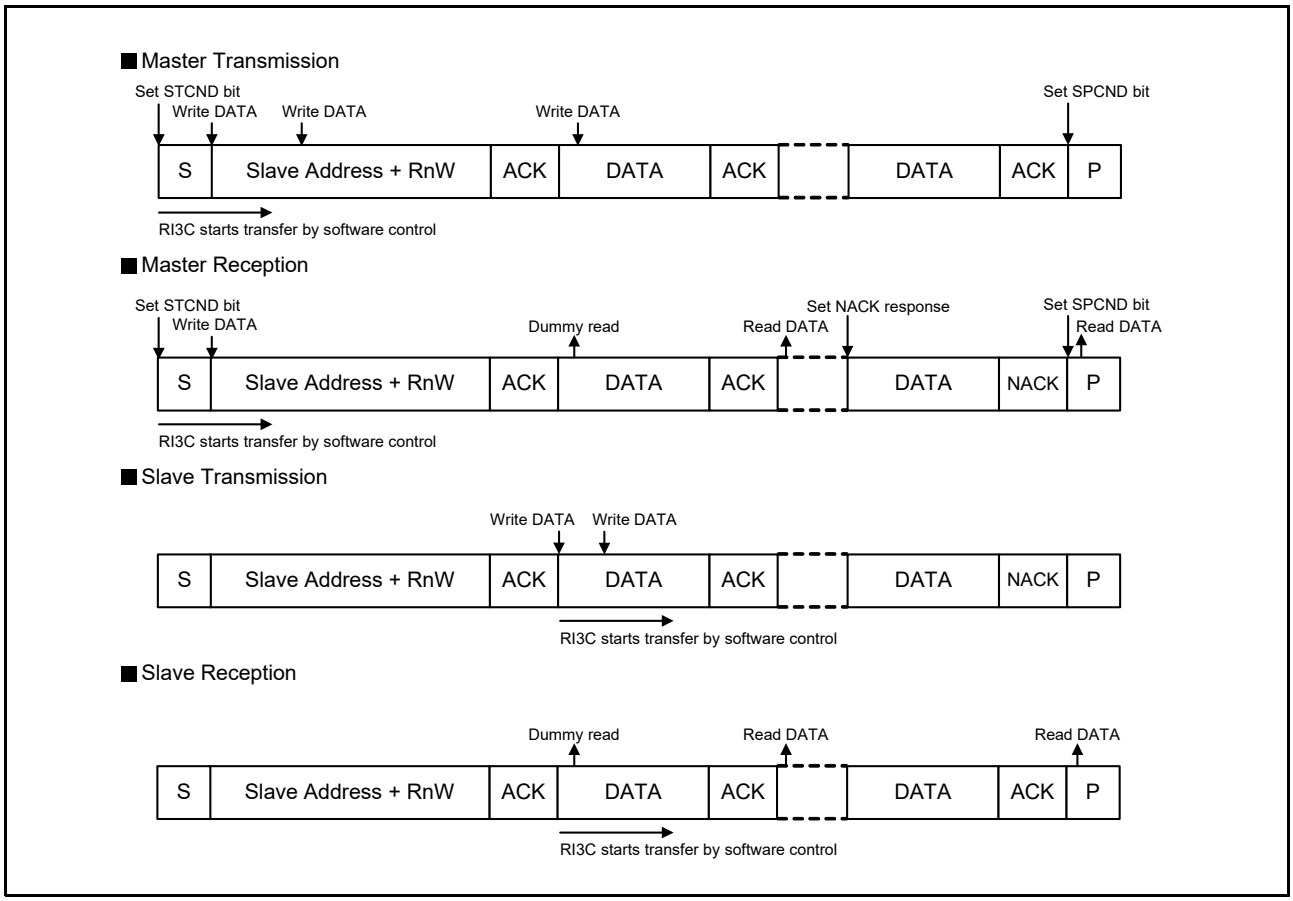


Figure 43.8 Data Handling with Single Buffer Transfer

43.4.2 I3C Protocol

43.4.2.1 Communication Protocol

(1) I3C Communication Data Format

Figure 43.9 through Figure 43.12 illustrate a typical communication for each of the supported I3C Protocols. While these diagrams do not exhaustively illustrate all possible RI3C communications, they do serve as useful introductions to the signaling and transmission formatting used in each supported I3C Protocol.

Figure 43.9 illustrates example communication using I3C Single Data Rate (SDR) mode with Broadcast Address (7Eh). It shows the Controller reading a byte of data from the Target at Address 2Bh in SDR Mode.
From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling the SDA line Low (in the Figure, pink fill means the Target is in control of the SDA line at this time). The Controller then issues a repeated START condition, then the Address of the Target (2Bh) it wants to read followed by RnW (1 for Read). The Controller then turns on a pull-up resistor and goes to Open Drain, allowing the Target to acknowledge by pulling the SDA line Low. At this point, the Controller continues to toggle the SCL line and release the SDA line, allowing the Target to drive SDA to send one byte of data (4Ah) followed by T. T = 1 informs the Controller that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Target drives SDA High until SCL goes High, at which time it releases SDA. The Controller has the option of holding SDA High with a weak pull-up, which signals to the Target that the Controller allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a repeated START condition), which would signal to the Target that the Controller has terminated the Read and is taking over.

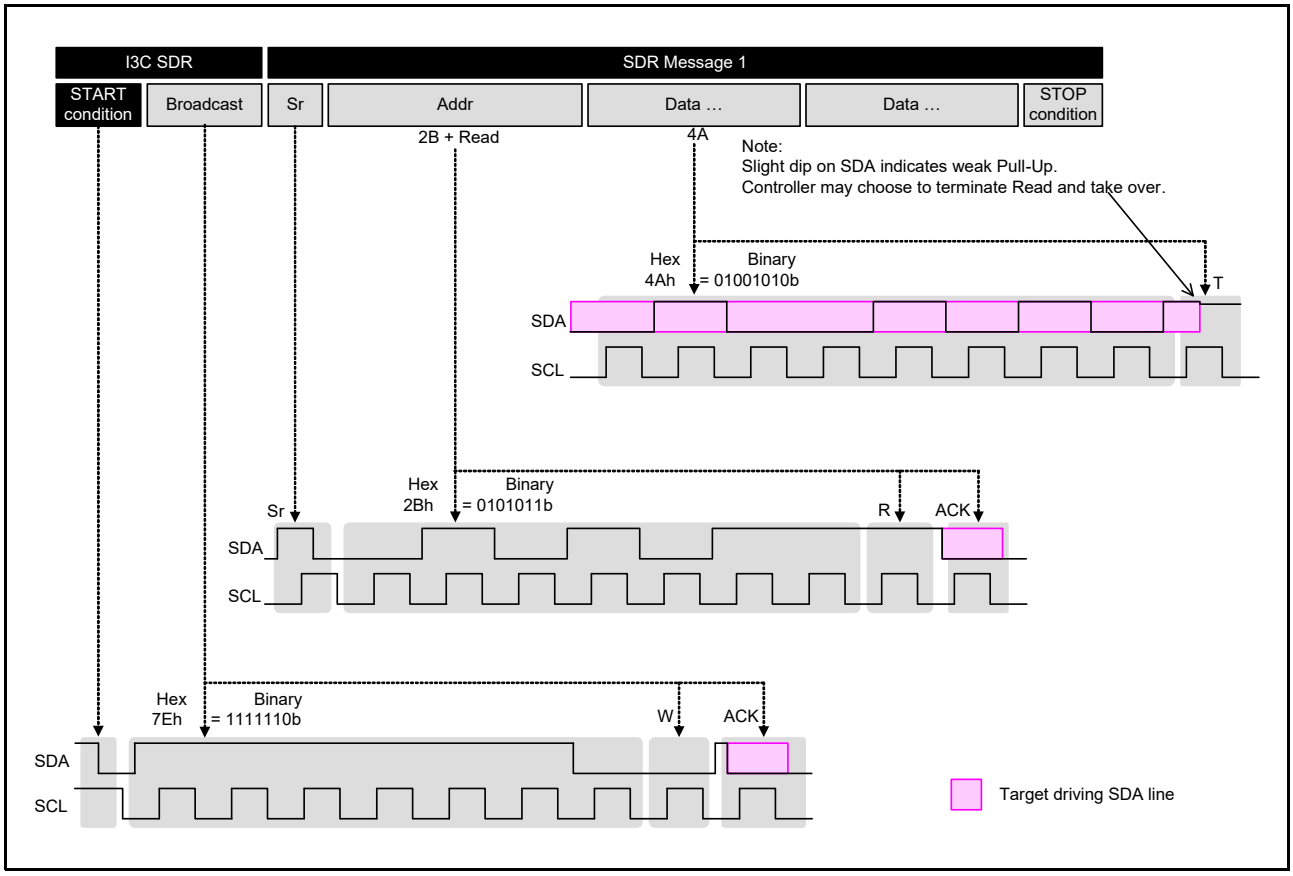


Figure 43.9 Example Communication Using I3C SDR Mode with Broadcast Address (7Eh)

Figure 43.10 illustrates example communication using I3C Single Data Rate (SDR) mode without Broadcast Address (7Eh). It shows the Controller reading a byte of data from the Target at Address 2Bh in SDR Mode.
From the Bus Free condition, the Controller issues a START condition, then the Address of the Target (2Bh) it wants to read followed by RnW (1 for Read). The Controller then turns on a pull-up resistor and goes to Open Drain, allowing the Target to acknowledge by pulling the SDA line Low. At this point, the Controller continues to toggle the SCL line and release the SDA line, allowing the Target to drive SDA to send one byte of data (4Ah) followed by T. T = 1 informs the Controller that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Target drives SDA High until SCL goes High, at which time it releases SDA. The Controller has the option of holding SDA High with a weak pull-up, which signals to the Target that the Controller allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a repeated START condition), which would signal to the Target that the Controller has terminated the Read and is taking over.

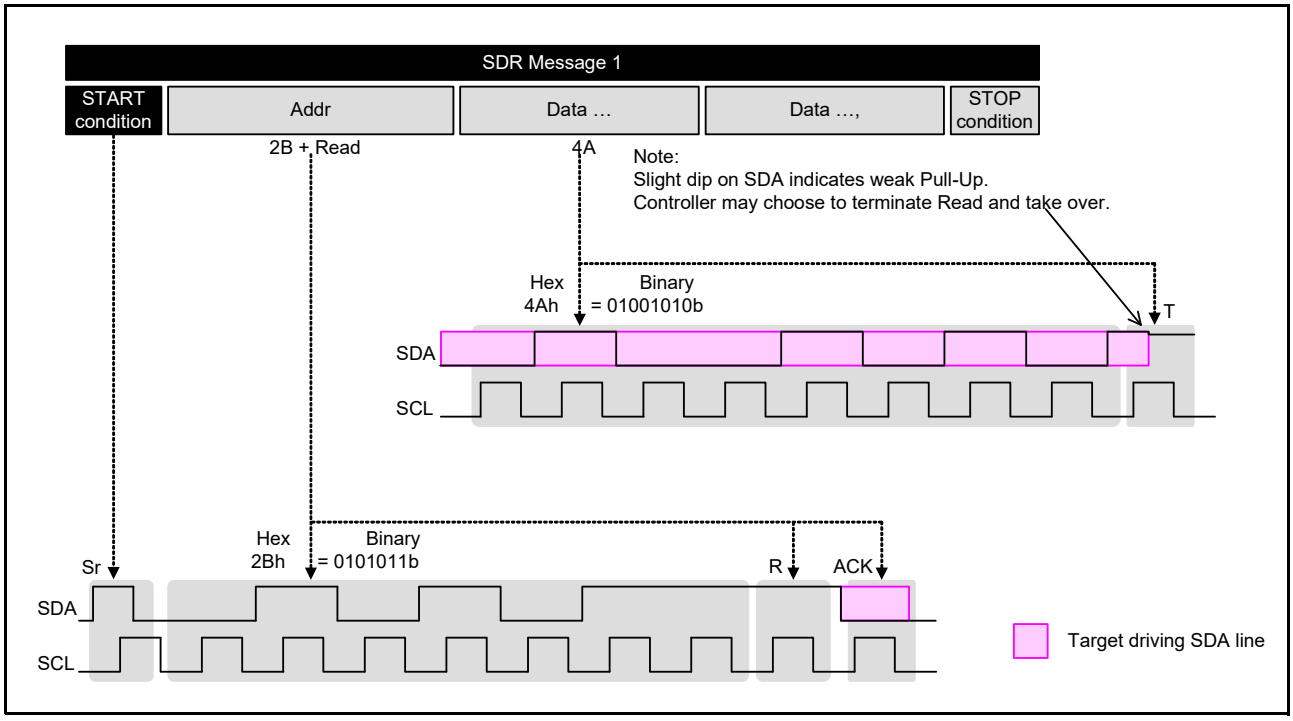


Figure 43.10 Example Communication Using I3C SDR Mode without Broadcast Address (7Eh)

Figure 43.11 shows the Controller issuing a Direct CCC to a single Target. This particular command (GETPID) reads the Provisioned ID of a Target.
From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling SDA Low (in the Figure, pink fill means the Targets are in control of SDA at this time). The Controller then issues the Direct Common Command Code for GETPID (8Dh) followed by parity bit T (odd parity = 1 for 8Dh) then the 7-bit Dynamic Address of the Target (chosen arbitrarily here to be 2Bh) followed by a RnW bit (1 for Read). Then the Controller turns on a pull-up resistor and goes to Open Drain, allowing the Target at Address 2Bh to ACK by pulling SDA Low, which tells the Controller that the Target acknowledges the command and will comply. (Alternatively, the Target may NACK by not pulling SDA Low, which would inform the Controller that the Target will not comply – in this case, that an error occurred.) Following the ACK the Target outputs its 48-bit PID one byte at a time, and then the Controller issues a repeated START condition (this part of the waveform sequence is not shown in the Figure).

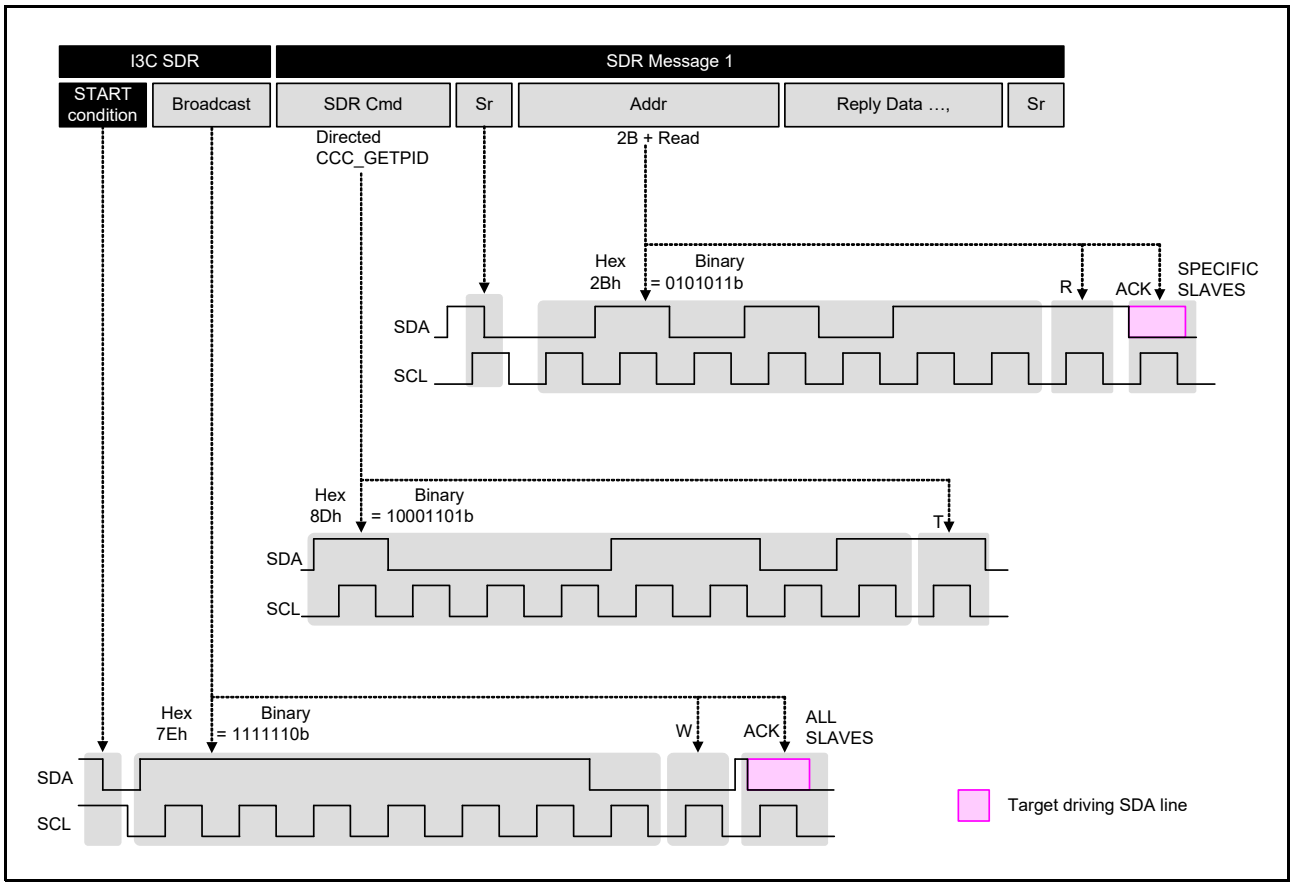


Figure 43.11 Example Communication Using I3C SDR Mode with Direct CCC

Figure 43.12 illustrates example SDR communication with a Broadcast CCC. The command used in this example sets the Maximum Read Length of all Targets to 43 bytes (002Bh).
From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling SDA Low (in the Figure, pink fill means the Targets are in control of SDA at this time). The Controller then issues the Broadcast Common Command Code for SETMRL (0Ah) followed by parity bit T (odd parity = 1 for 0Ah), and then 2 data bytes (MSB first) to define the maximum number of bytes that can be read from a Target in a single read operation. Each data byte is followed by a T bit (parity bit – odd parity). After this the Controller issues a repeated START condition.

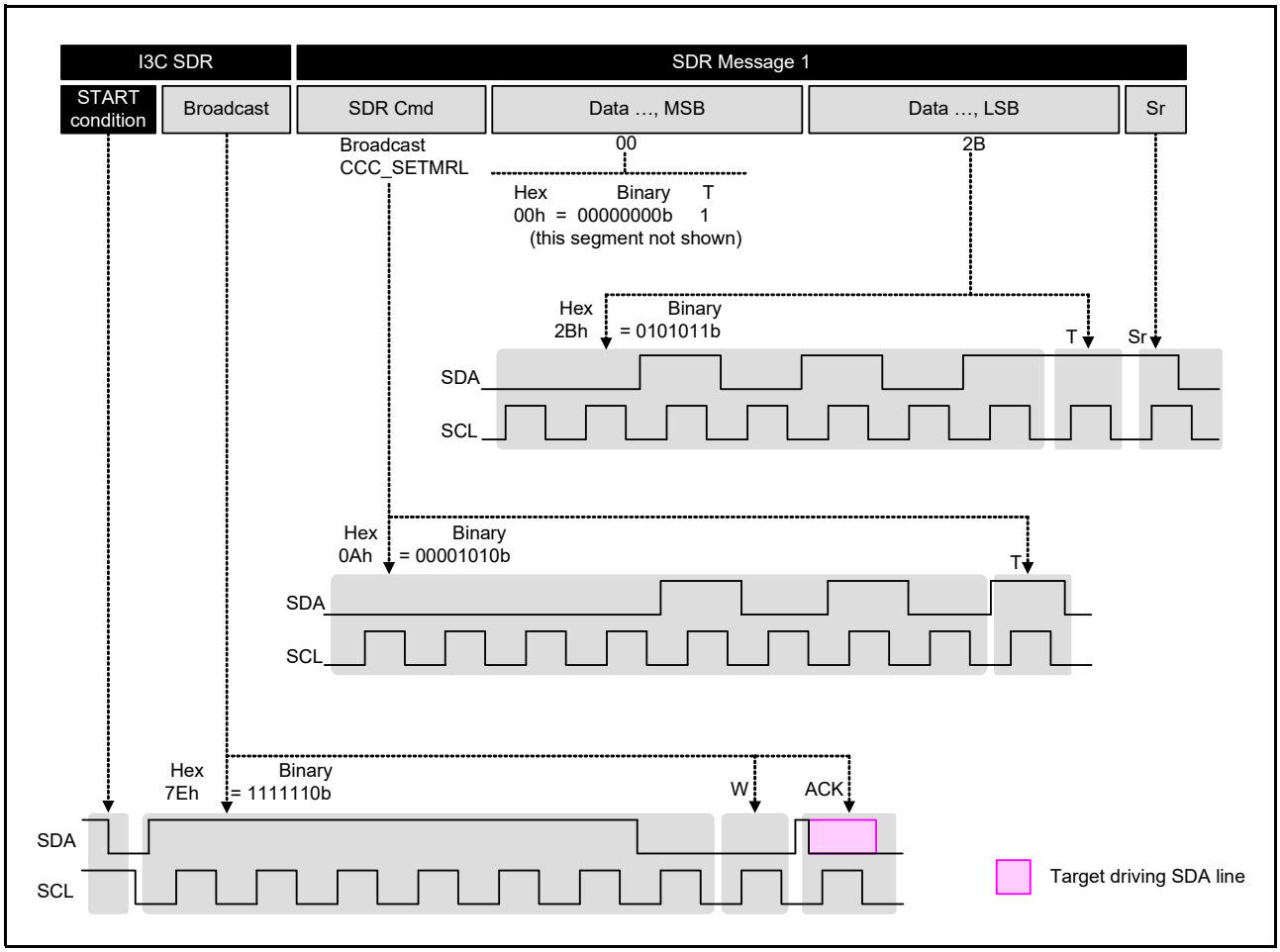


Figure 43.12 Example Communication Using I3C SDR Mode with Broadcast CCC

43.4.2.2 Bus Conditions

RI3C defines three distinct conditions in which the I3C bus shall be considered inactive: Bus Free, Bus Available, and Bus Idle (see Figure 43.13).

(1) Bus Free Condition

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBFTR register.

(2) Bus Available Condition (I3C protocol mode)

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBATR register.

A Target may only issue a START Request (For example, for an IBI, or for a CRR) after a Bus Available condition.

(3) Bus Idle Condition (I3C protocol mode)

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBITR register.

A Target may only issue a START Request after a Bus Idle condition.

Specifications are as follows. Bus Idle period needs to be the largest.

ICBFTR < ICBATR < ICBITR

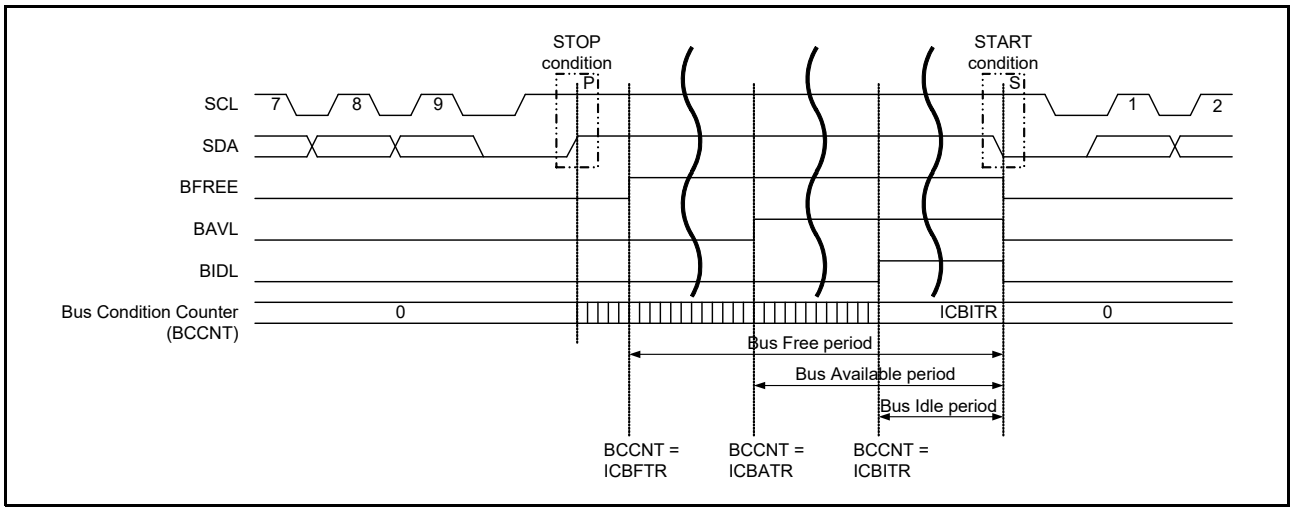


Figure 43.13 Bus Conditions

43.4.3 I²C Protocol

(1) I²C Communication Data Format

The I²C-bus format consists of 8-bit data and 1-bit acknowledgment. The frame following a START condition or repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is generated. Figure 43.14 shows the I²C-bus format, and Figure 43.15 shows the I²C-bus timing.

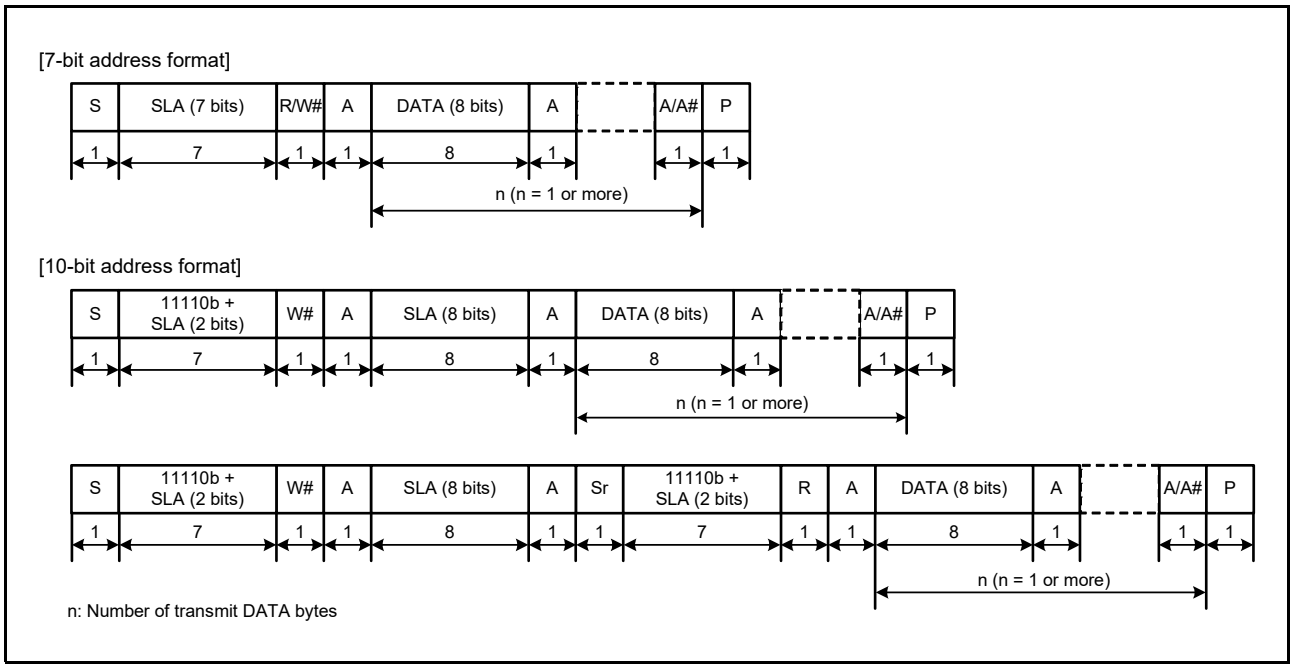


Figure 43.14 I²C-bus Format

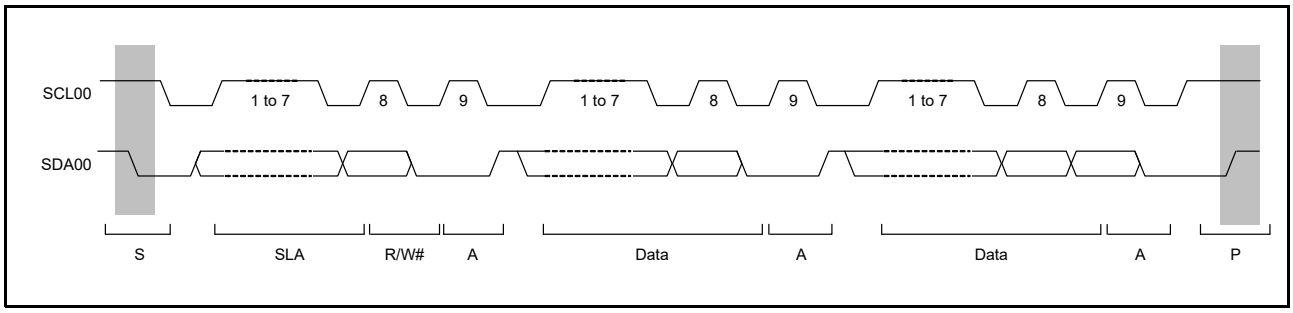


Figure 43.15 I²C-bus Timing (SLA = 7 Bits)

S: START condition. The master device drives the SDA00 line low from high while the SCL00 line is high.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.

A: Acknowledge. The receiver drives the SDA00 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDA00 line high.

Sr: Repeated START condition. The master device drives the SDA00 line low from high after the setup time has elapsed with the SCL00 line high.

DATA: Transmitted or received data

P: STOP condition. The master device drives the SDA00 line high from low while the SCL00 line is high.

43.4.4 Initial Settings

(1) I3C Protocol Mode

Before starting data transmission and reception, initialize the RI3C according to the procedure in Figure 43.16.

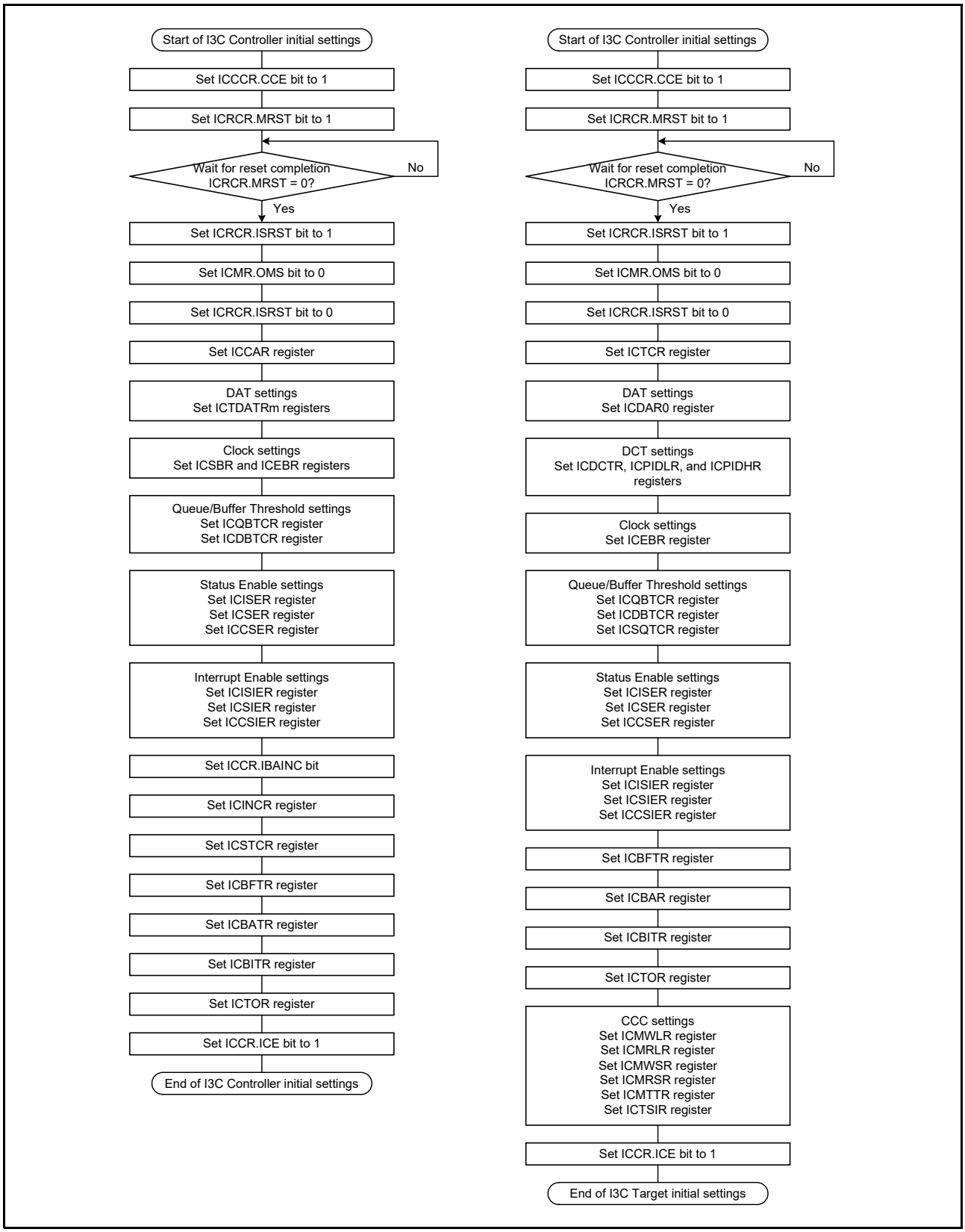
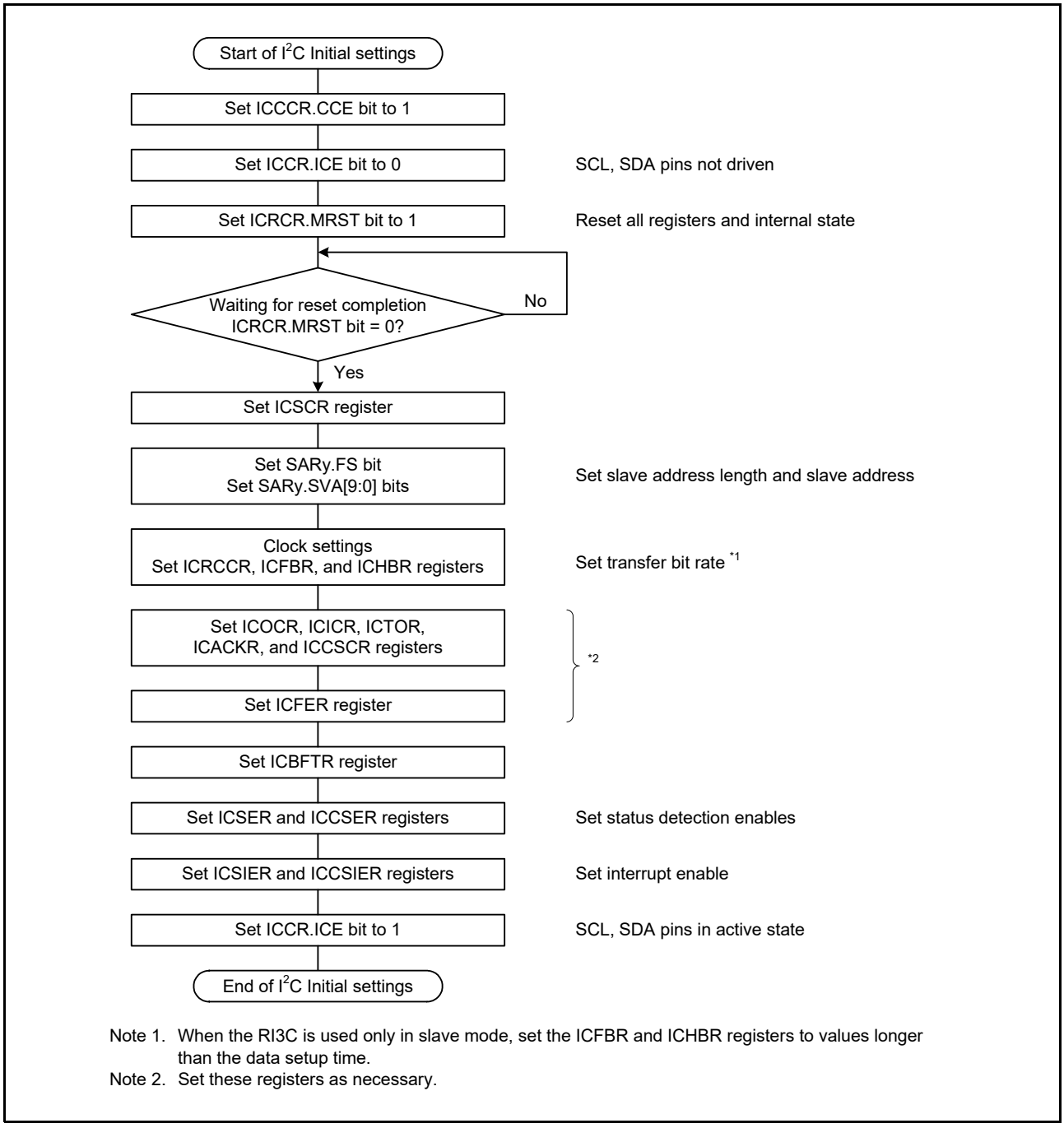


Figure 43.16 Example of RI3C Initialization Flowchart

(2) I2C Protocol Mode

Before starting data transmission and reception, initialize the RI3C according to the procedure in Figure 43.17.
First, Set the ICCR.ICE bit to 0 (SCL00, SDA00 pins not driven).
Next, Set the ICRCR.MRST bit to 1 (RI3C reset). This initialize the all registers and internal state. Refer to Table 43.16, Reset Control of Registers. Then, it waits for MRST to become 0.
After that, set SARy.FS bit, SARy.SVA[9:0] bits, ICFBR register, ICICR register, ICOCR register, ICTOR register, ICCSCR register, ICACKR register, ICFER register, and set the other registers as necessary (y = 0 to 2).



Note 1. When the RI3C is used only in slave mode, set the ICFBR and ICHBR registers to values longer than the data setup time.
Note 2. Set these registers as necessary.

Figure 43.17 Example of RI3C Initialization Flowchart (y = 0 to 2)

43.4.5 I3C Communication Flow

Figure 43.18 illustrates how I3C communication is initiated:

- All I3C communication occurs within a frame. The frame begins with a START condition, followed by one or more transfers, and a STOP condition.

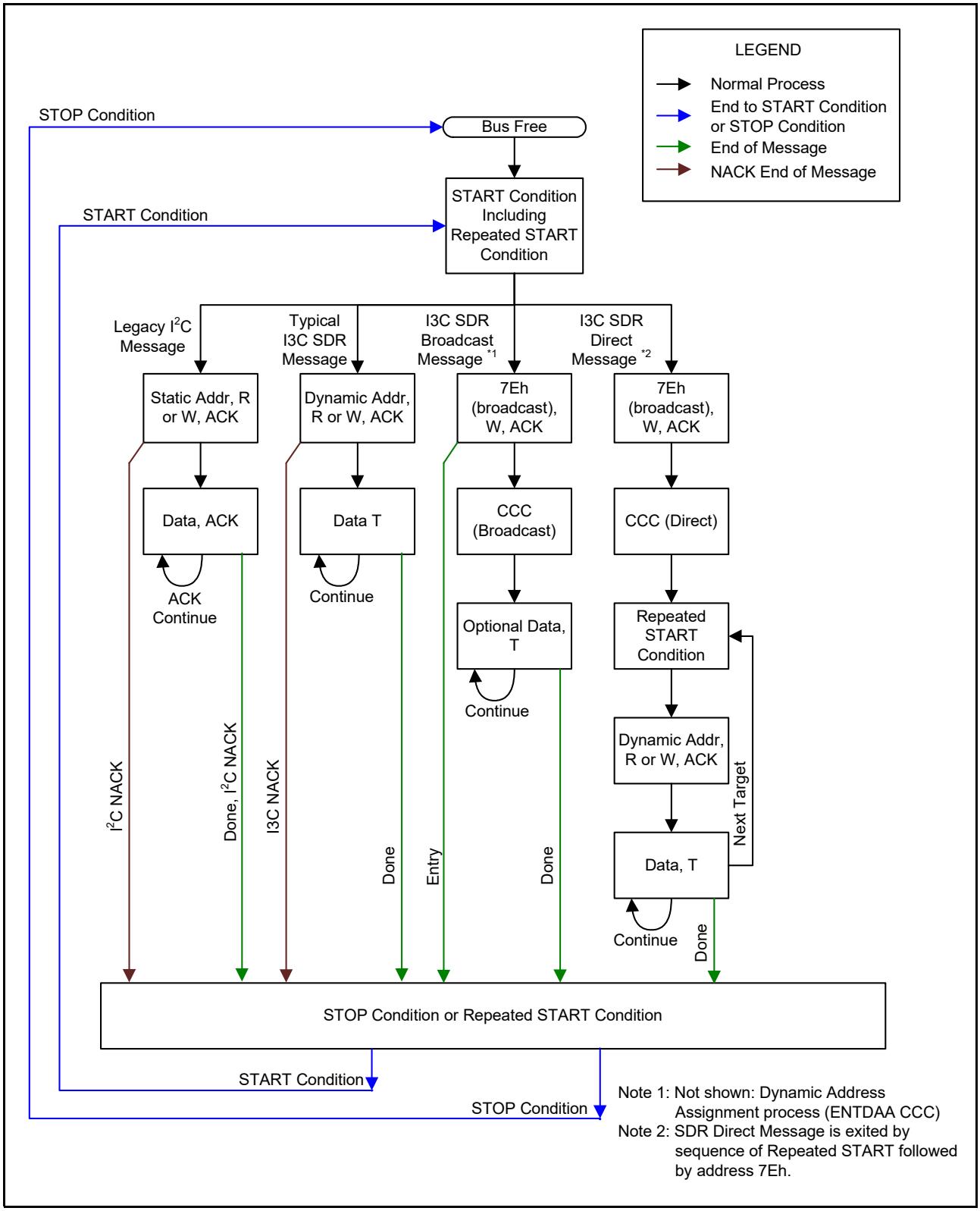


Figure 43.18 I3C Communication Flow

I3C is based on a frame encapsulation approach. A frame includes a data payload. The transfer protocol for the data payload is SDR. Frames are bordered by I^2C-like bus management.
The I3C frame always includes at least the START condition, the Header, the Data, and the STOP condition. The Header following a START condition allows for bus arbitration. The Controller uses the Header to address Target device(s). Target device(s) may use the Header Arbitration for multiple purposes: for IBI and for Secondary Controller functionality.
I3C allows only one Controller to have control of the I3C bus at a time. Mechanisms for handoff of the Controller role from one device to another device are provided.

43.4.6 I3C Controller Operation

43.4.6.1 Dynamic Address Assignment Procedure

After initializing RI3C, first execute Dynamic Address Assignment Procedure for I3C Target connected on the I3C bus. The following describes the procedure.

(1) Initial setting (refer to section 43.4.4, Initial Settings for details)

(2) Execute Dynamic Address Assignment with ENTDAA or SETDASA Common Command Code (CCC) for I3C Target set in DAT (ICTDATRm register).
Write Command Descriptor (address assignment command) to command queue via the ICCQR register.

(3) When Command Descriptor is written to command queue, Transaction is issued on I3C bus.

(4) When ENTDAA is specified for the CMD field of address assignment command:
Executes Dynamic Address Assignment for I3C Targets corresponding to the number of the ICTDATRm registers specified by the DEV_COUNT field starting with the ICTDATRm register indexed by the DEV_INDEX field of address assignment command.
When SETDASA is specified for the CMD field of address assignment command:
Execute Dynamic Address Assignment for I3C Target indicated by the ICTDATRm register indexed by the DEV_INDEX field of address assignment command.

(5) In case of ENTDAA, the Provisioned ID, BCR, DCR transmitted from I3C Target is stored in Receive Data Buffer (BCR is also automatically stored in the ICTDCTRm register).
Read the Provisioned ID, BCR, and DCR from the receive data buffer via the ICDR register with a receive data full interrupt.

(6) When execution of Dynamic Address Assignment is completed, issue STOP condition and store the Response Descriptor into the response queue.

(7) Read the Response Descriptor via the ICRQR register and check the status.

(8) Check whether the value of the DATA_LENGTH field of the Response Descriptor matches the value of the DEV_COUNT field of the address assignment command.

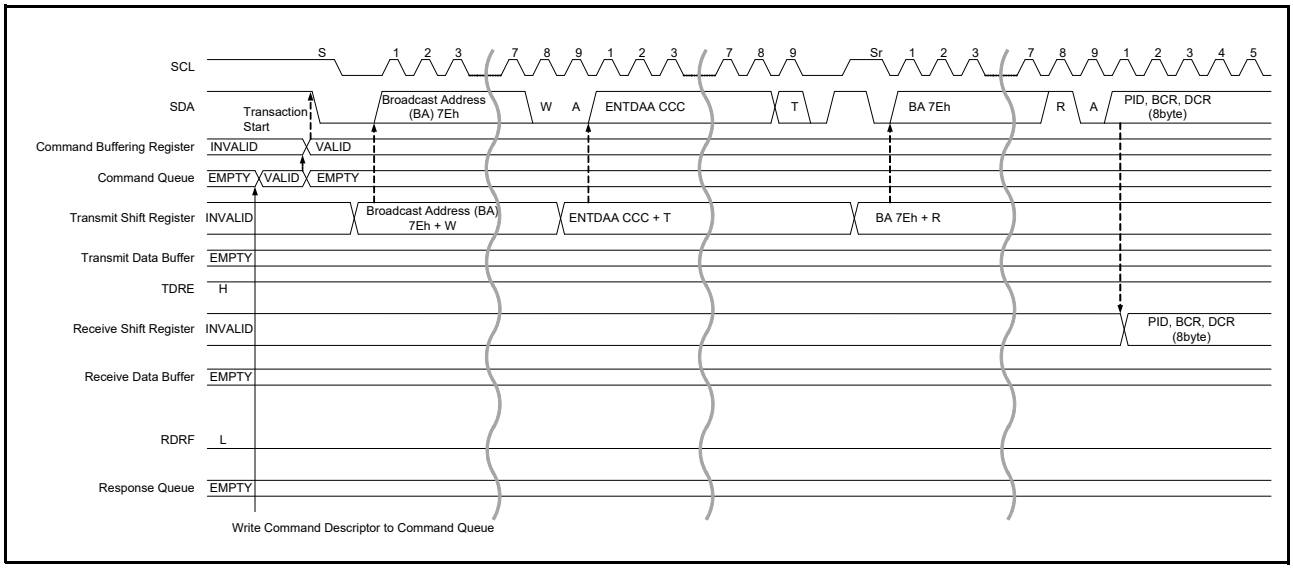


Figure 43.19 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (1/3)

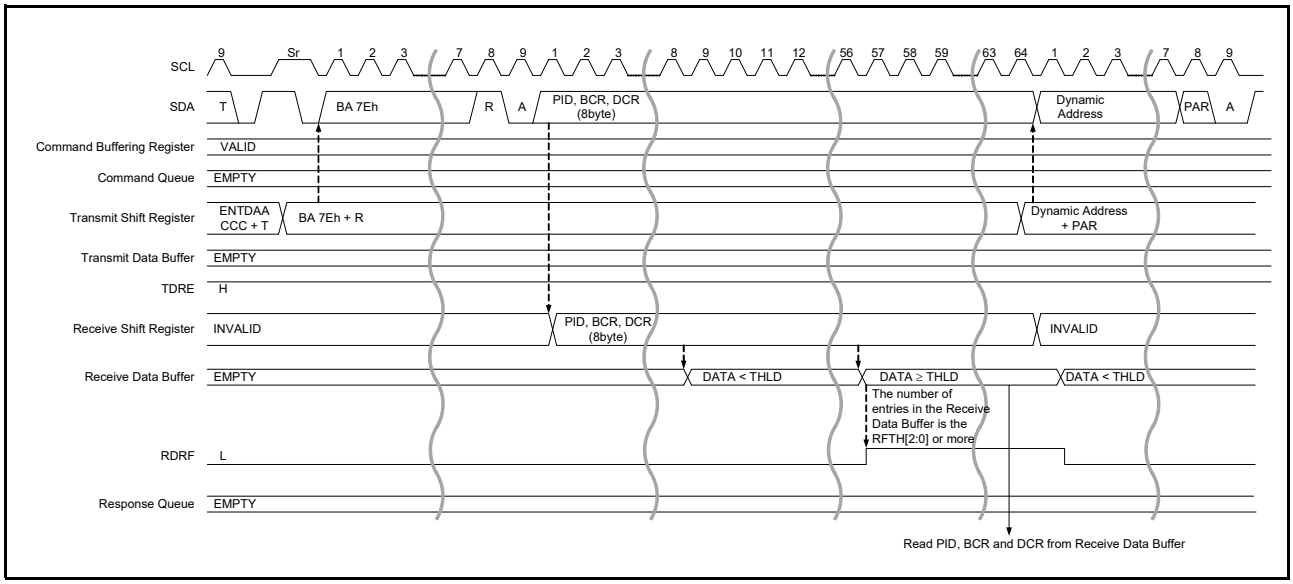


Figure 43.20 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (2/3)

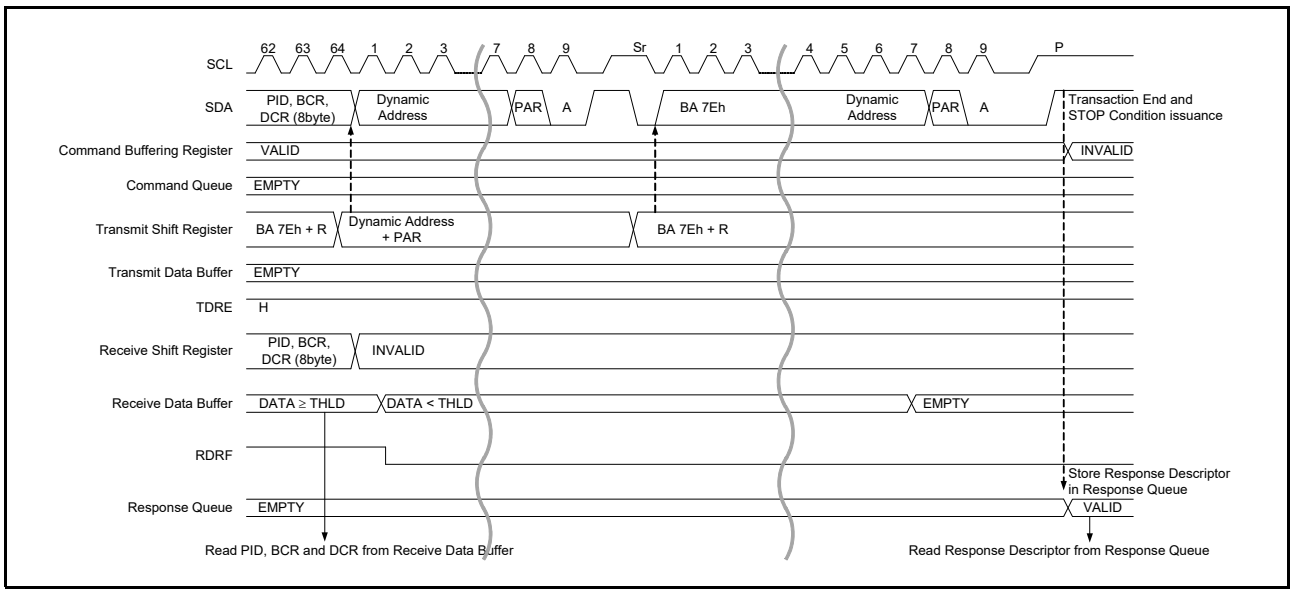


Figure 43.21 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (3/3)

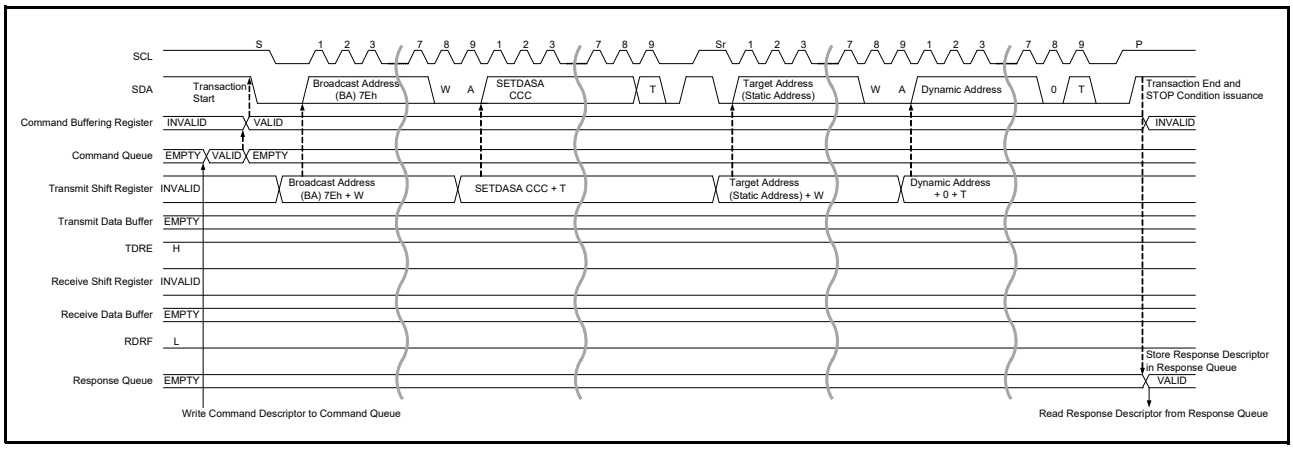


Figure 43.22 Dynamic Address Assignment Procedure (SETDASA CCC) Timing

43.4.6.2 SDR Write Transfer

(1) Write data for transmission to the Transmit Data Buffer via the ICDR register.
(2) Write a Command Descriptor (immediate data transfer command or regular data transfer command or combo transfer command) for Data Transfer to the command queue via the ICCQR register.
(3) When Command Descriptor is written to command queue, transaction is issued on I3C bus.
When NACK is received with the Address header, transaction of the same command is automatically issued according to the NACK Retry Count value (ICTDATRm.NACKRC[1:0]) of DAT.
(4) If data for transmission still remain, write data for transmission by a transmit data empty interrupt to the Transmit Data Buffer via the ICDR register.
(5) When data transmission for the number of bytes specified in the DATA_LENGTH field of the Command Descriptor is completed, the repeated START condition or STOP condition is issued and the Response Descriptor is stored in the response queue.
(6) Read the Response Descriptor via the ICRQR register and check the status.
(7) Check that the value of the DATA_LENGTH field of the Response Descriptor is 0.

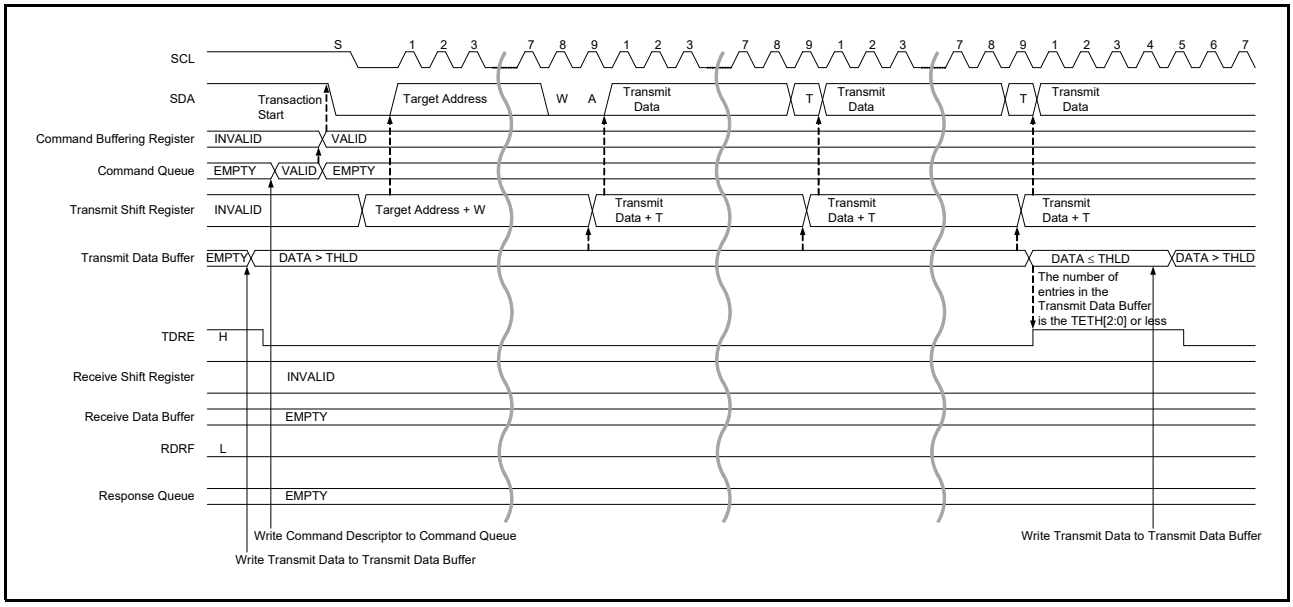


Figure 43.23 SDR Write Transfer Timing (1/2)

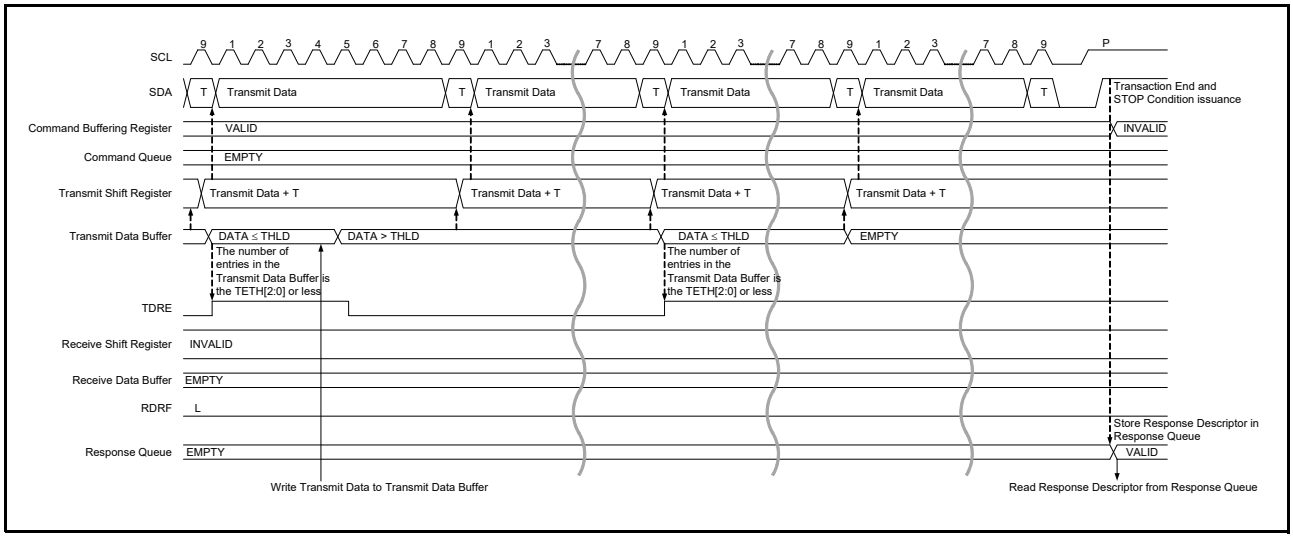


Figure 43.24 SDR Write Transfer Timing (2/2)

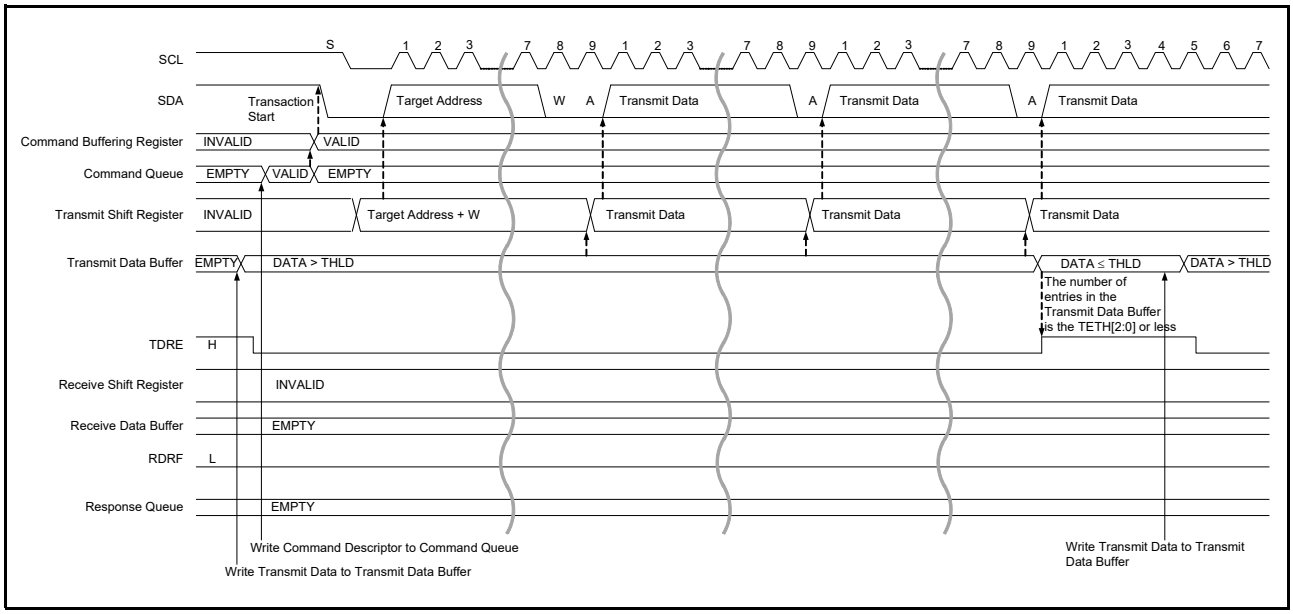


Figure 43.25 Legacy I²C Write Transfer Timing (1/2)

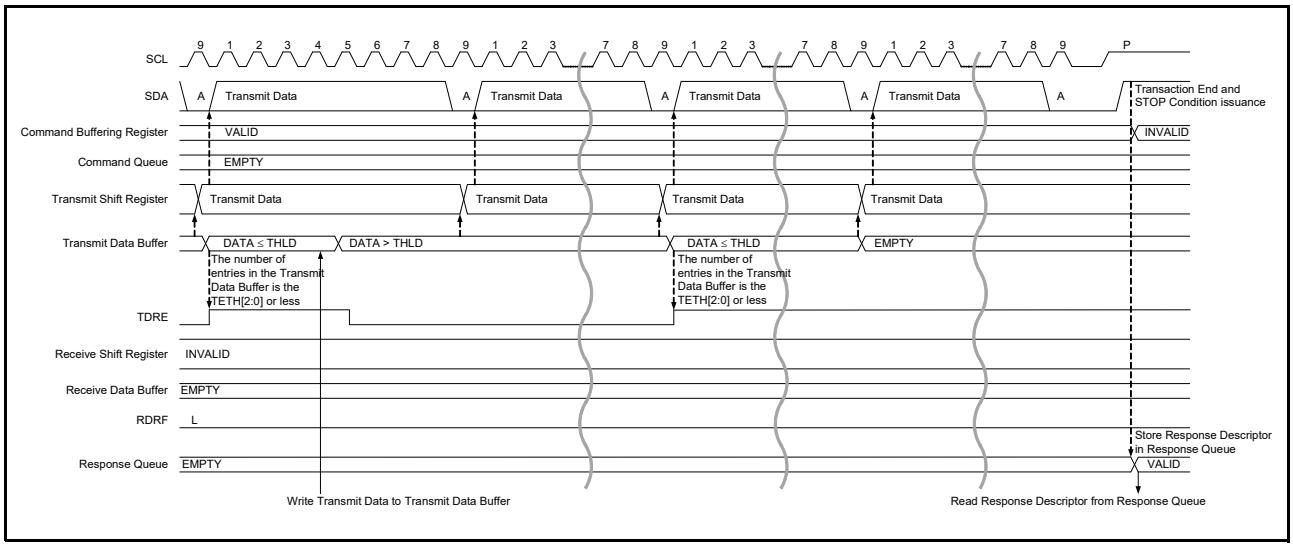


Figure 43.26 Legacy I²C Write Transfer Timing (2/2)

43.4.6.3 SDR Read Transfer

(1) Write a Command Descriptor (immediate data transfer command or regular data transfer command or combo transfer command) for Data Transfer to the command queue via the ICCQR register.

(2) When Command Descriptor is written to command queue, transaction is issued on I3C bus.
When NACK is received with the Address header, transaction of the same command is automatically issued according to the NACK Retry Count value (ICTDATRm.NACKRC[1:0]) of DAT.

(3) Data received from the I3C Target is stored in the Receive Data Buffer.

(4) In the receive data full interrupt, read the received data from the Receive Data Buffer via the ICDR register.

(5) SDR:
Detecting Low in T-bit or receiving Data for the number of bytes specified in the DATA_LENGTH field of Command Descriptor is completed, issue repeated START condition or STOP condition and store the Response Descriptor into the response queue.
Legacy I^2C Message:
When data reception for the number of bytes specified in the DATA_LENGTH field of Command Descriptor is completed, NACK is issued. After that, issue a repeated START condition or STOP condition and store the Response Descriptor into the response queue.

(6) Read the Response Descriptor via the ICRQR register and check the status.

(7) Check whether the value of the DATA_LENGTH field of the Response Descriptor matches the data length setting value of the Command Descriptor.

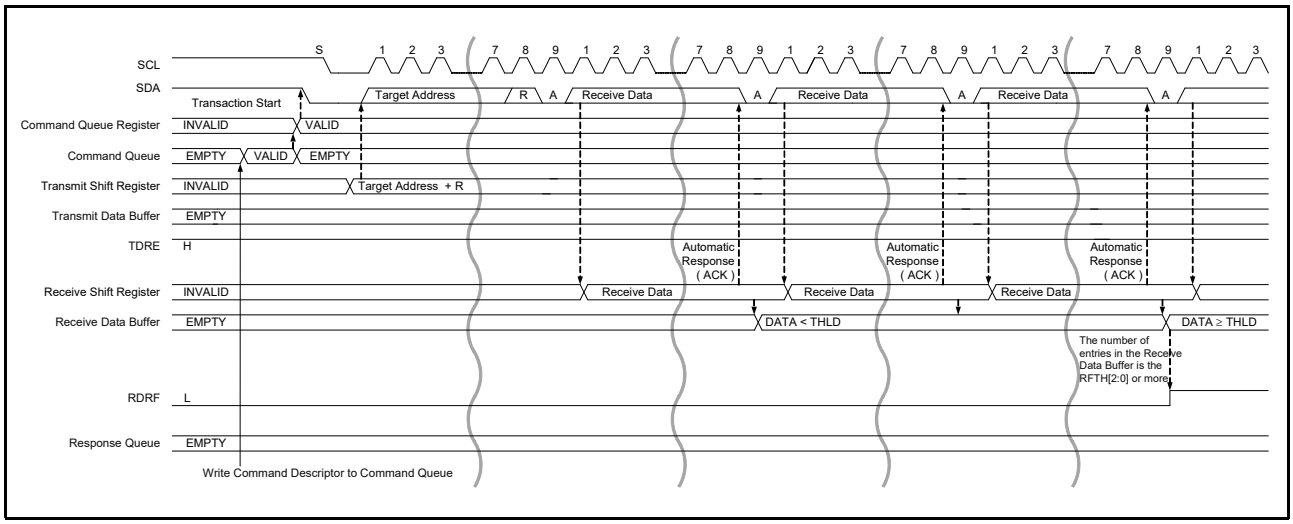


Figure 43.27 SDR Read Transfer Timing (1/2)

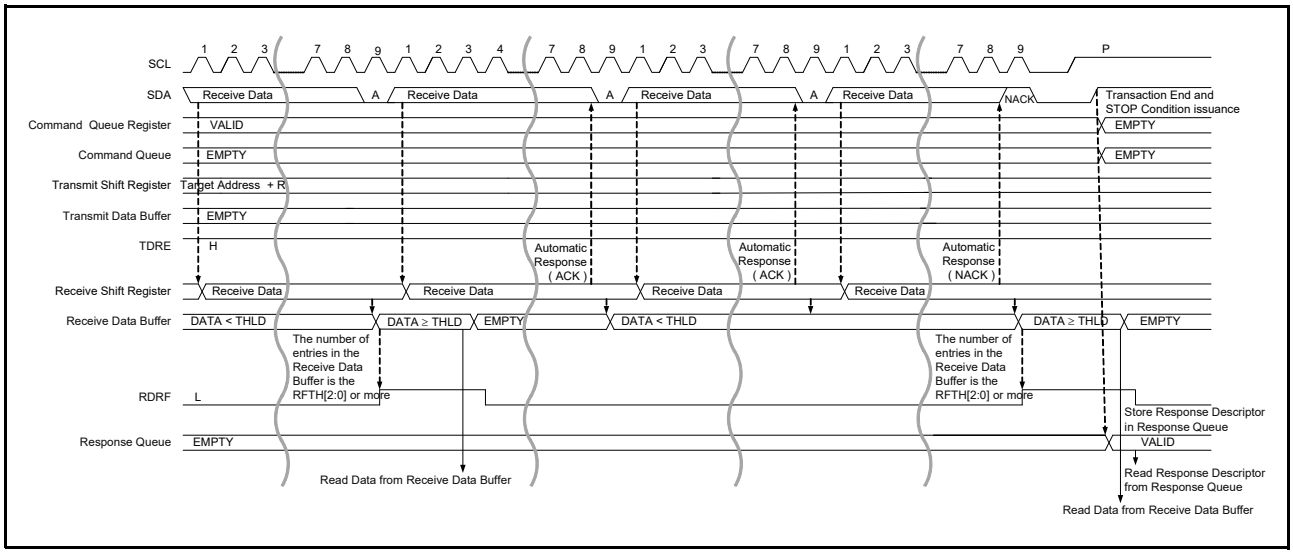


Figure 43.28 SDR Read Transfer Timing (2/2)

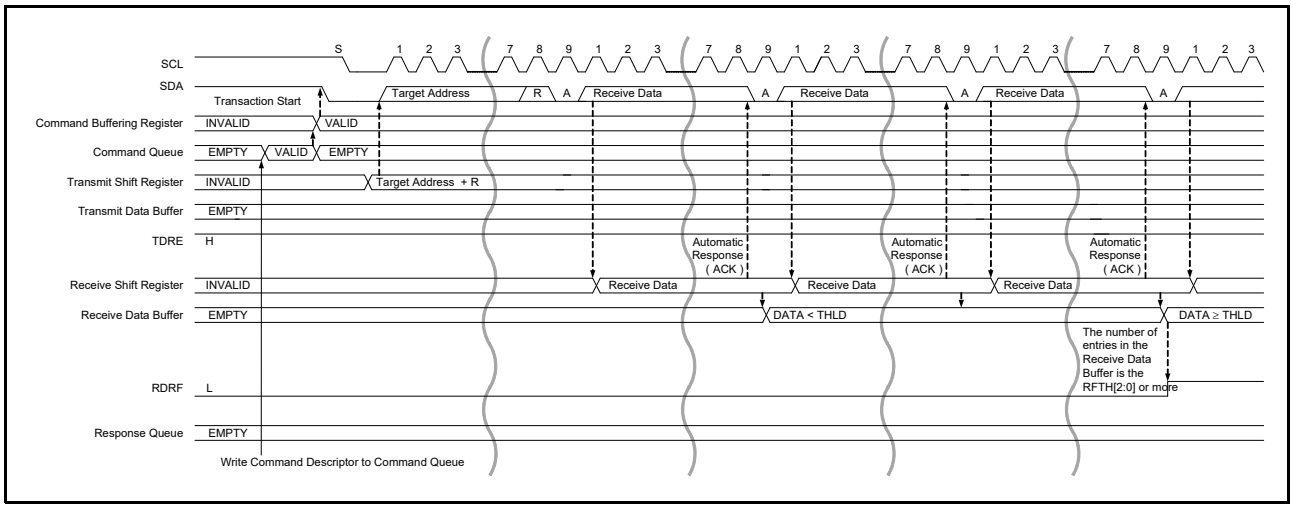


Figure 43.29 Legacy I²C Read Transfer Timing (1/2)

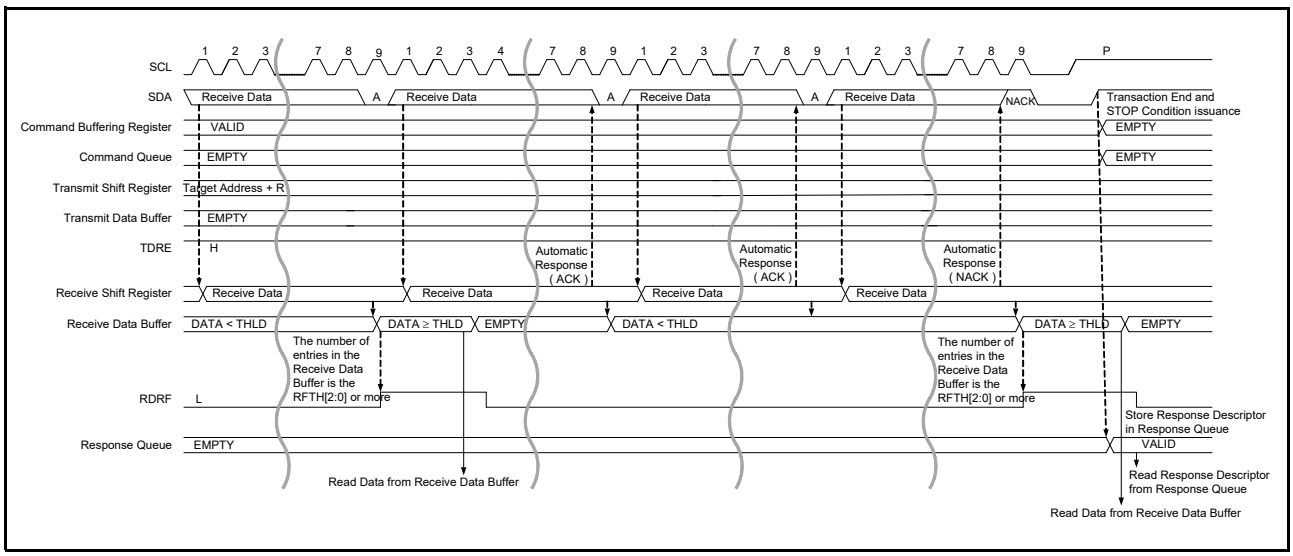


Figure 43.30 Legacy I²C Read Transfer Timing (2/2)

43.4.6.4 IBI Transfer

(1) Write Command Descriptor to the command queue and issue Transaction on I3C bus.
If START Request (SDA Low Drive) is issued from the Target device, RI3C drives SCL to Low and completes START condition.
Thereafter, the SCL is supplied and IBI Request is received.
(2) In Target Address and RnW field in the Address header, if losing Arbitration by issuing IBI from I3C Target, stop issuing Transaction.
(3) According to section 43.5.13, IBI (I3C Protocol Mode), detect IBI and process.
(4) In the IBI queue empty/full interrupt routine, read the IBI Status Descriptor from the IBI queue via the ICIQR register and check the status.
When detected a Target Interrupt Request and responded with ACK, Read the IBI Data for the Data Length indicated by the DATA_LENGTH field of the IBI Status Descriptor from the IBI queue via the ICIQR register.
(5) Restart issuing Transaction of Command of Step1.

An example of the processing procedure after detection of IBI is shown below.

Processing procedure for detecting Controller Role Request and transferring Controller role to Secondary Controller

(1) If the I3C Secondary Controller wins the Arbitration, issue a DEFTGTS CCC and notify Target information to Secondary Controller.
(2) Issue a GETACCCR CCC and complete CCC by a STOP condition.

Note: • After transferring Controller role to Secondary Controller, to get Controller role again, issue a Controller Role Request according to section 43.4.6.4, IBI Transfer.
• After Controller Role Request is accepted by the Active Controller, to get Controller role again at receiving the GETACCCR CCC and complete CCC by a STOP condition.

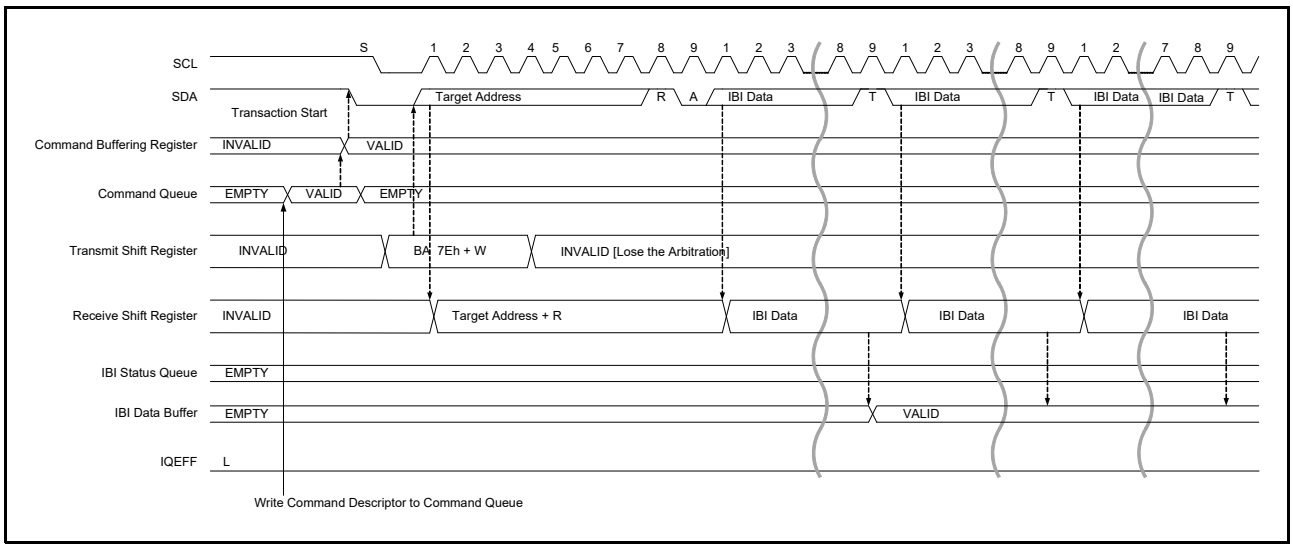


Figure 43.31 I3C Controller IBI Transfer Timing (1/2)

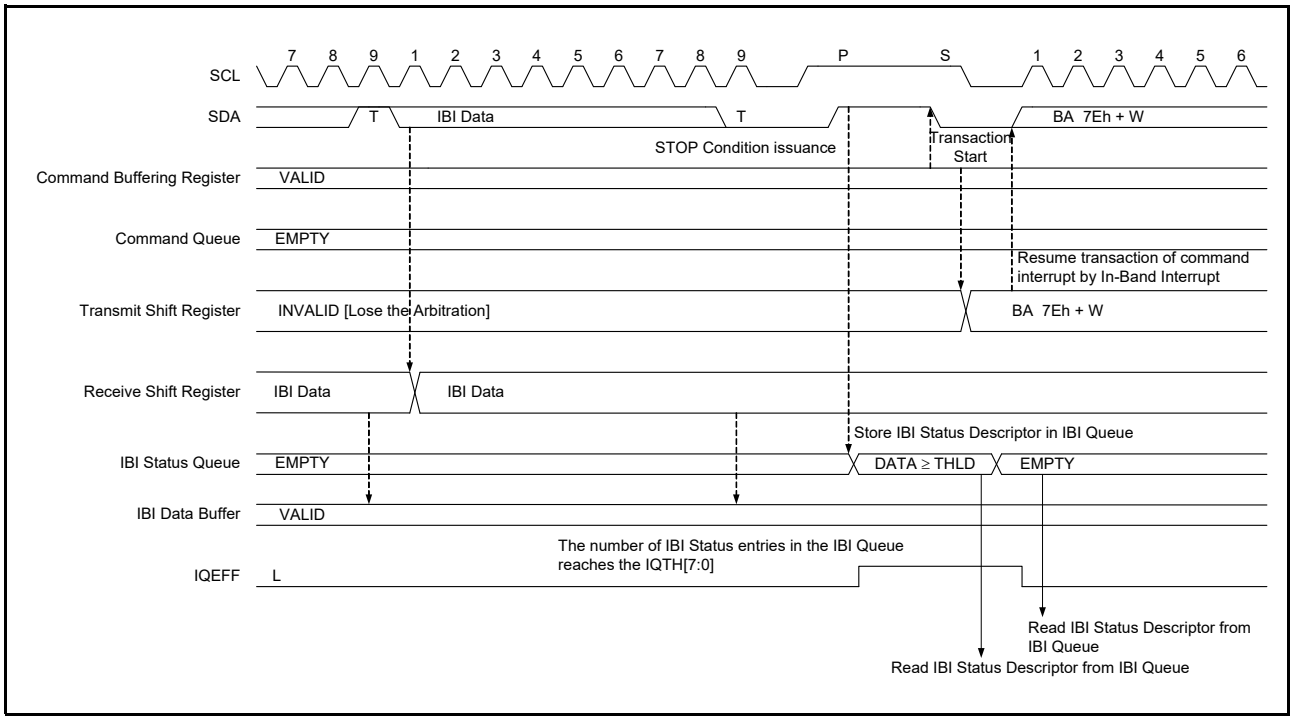
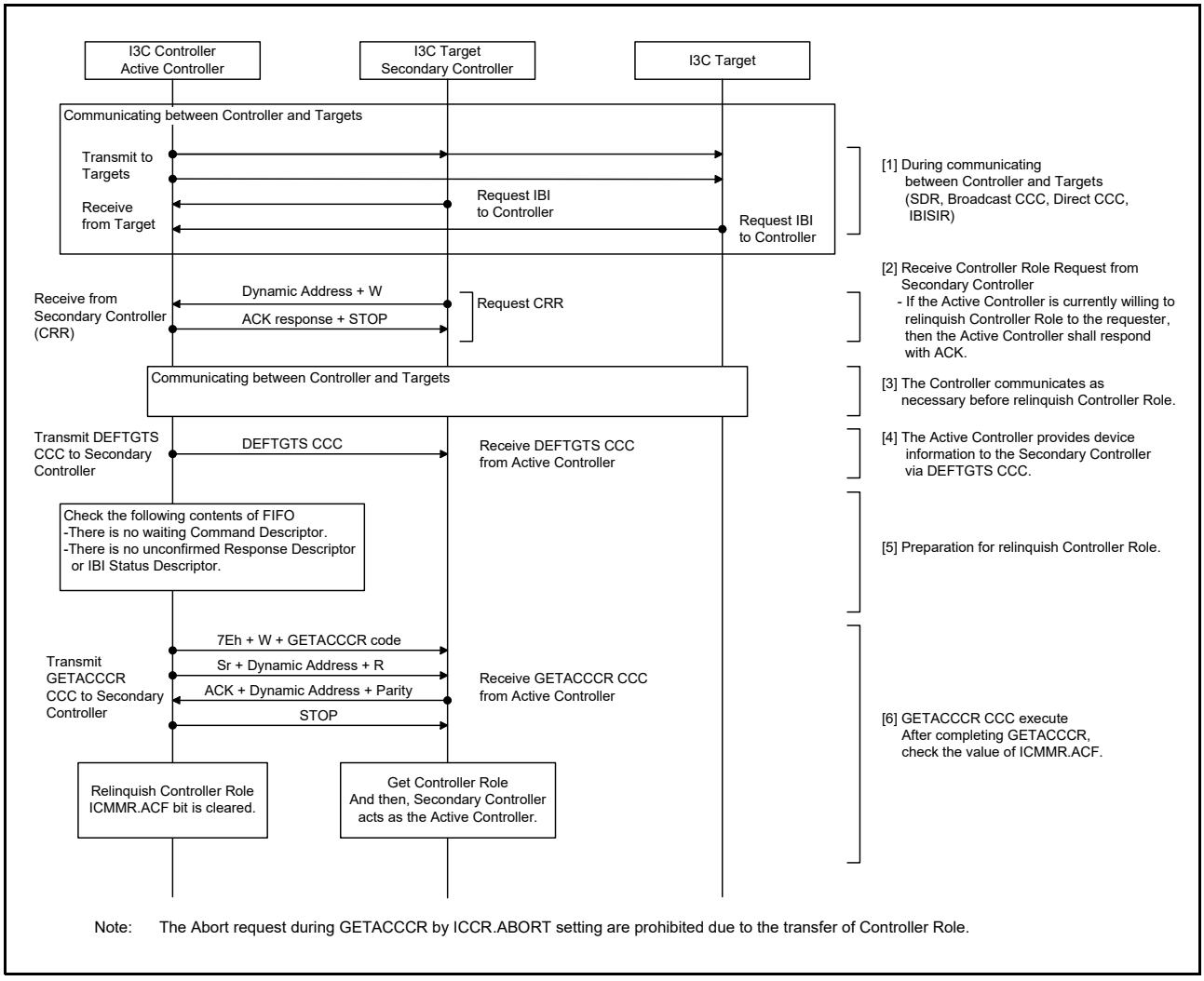


Figure 43.32 I3C Controller IBI Transfer Timing (2/2)

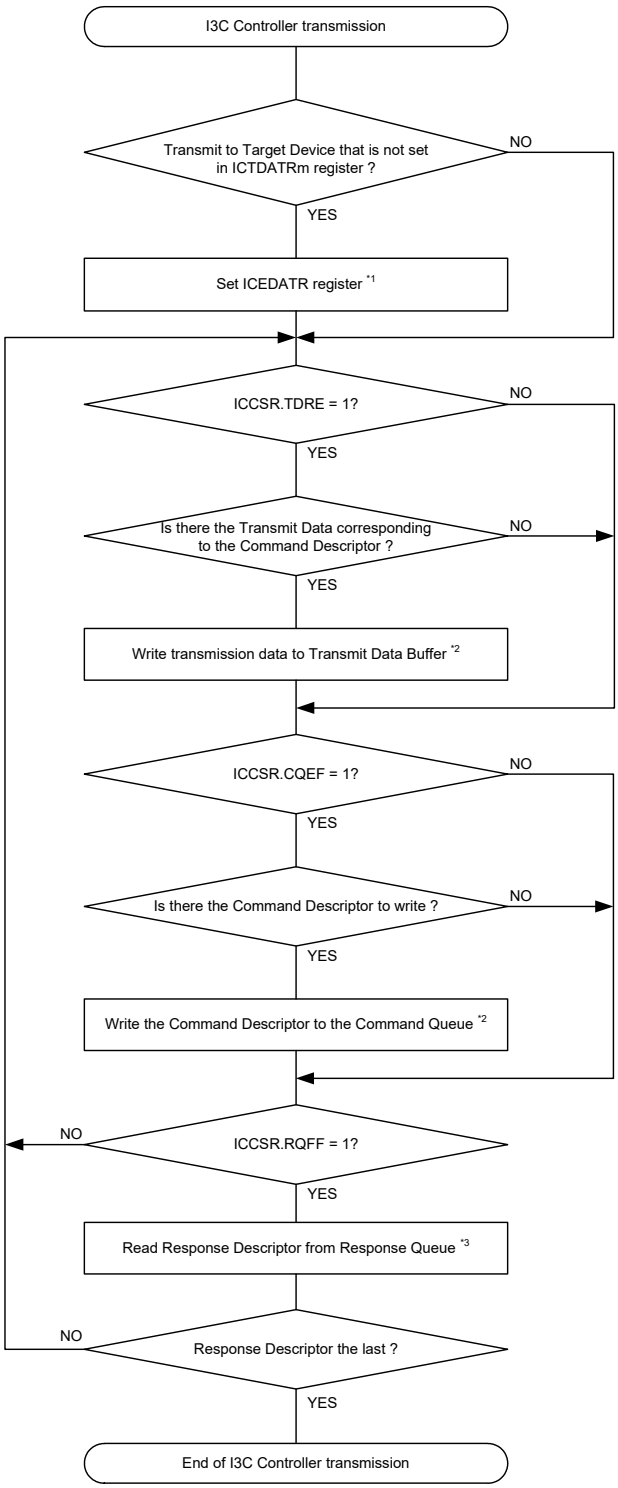


Note: The Abort request during GETACCCR by ICCR.ABORT setting are prohibited due to the transfer of Controller Role.

Figure 43.33 I3C Controller CRR Processing Flow

43.4.6.5 I3C Controller Transmission Flow (FIFO Buffer Transfer)

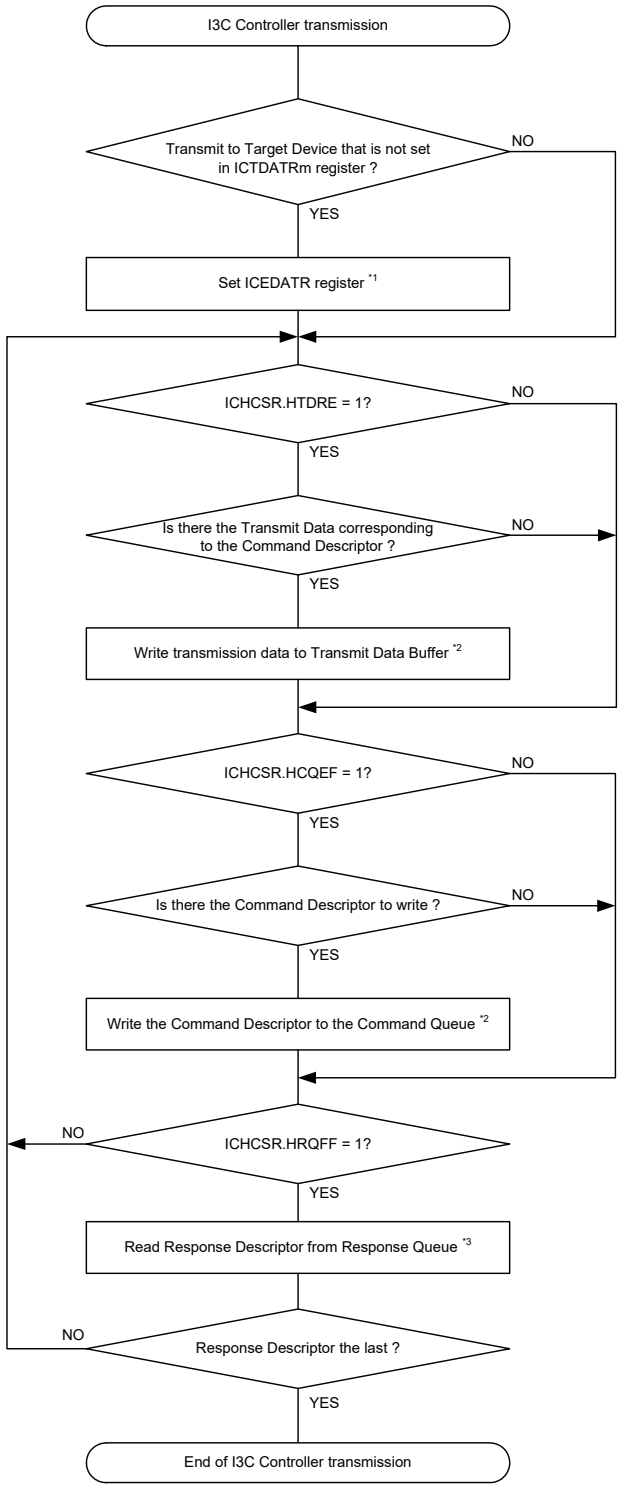
Controller transmission flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.34 Example of I3C Controller Transmission Flowchart (FIFO Buffer Transfer)

43.4.6.6 I3C Controller Transmission Flow (High-Priority FIFO Buffer Transfer)

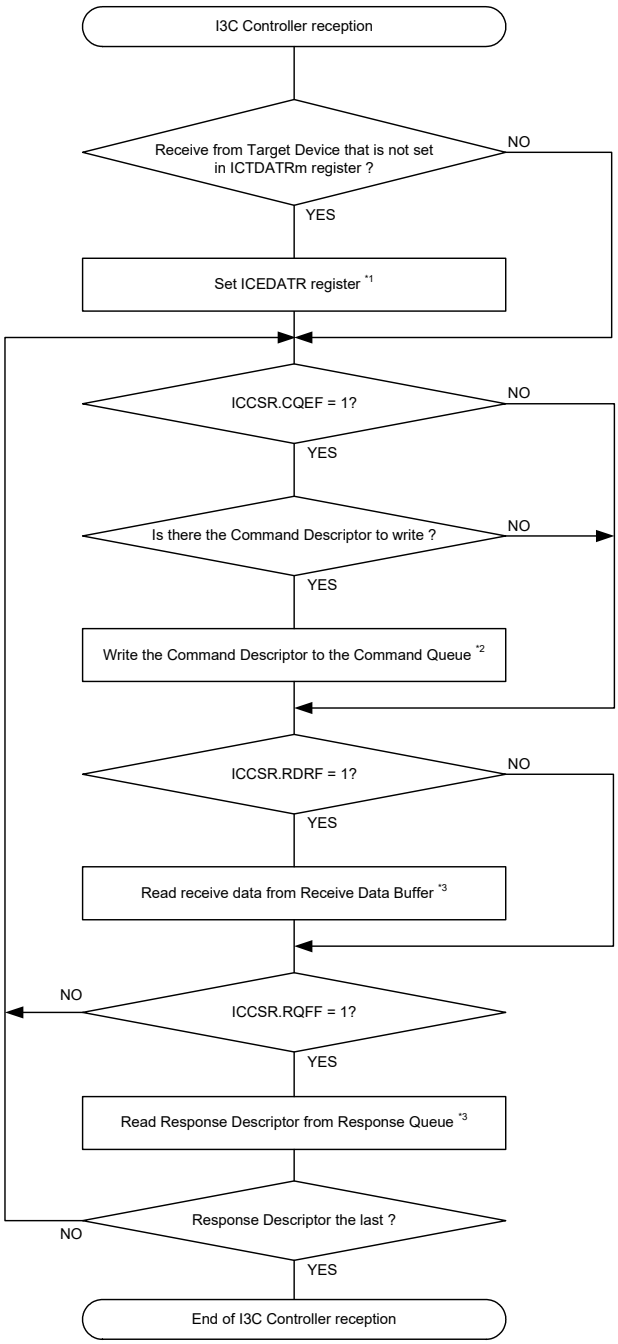


Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.35 Example of I3C Controller Transmission Flowchart (High-Priority FIFO Buffer Transfer)

43.4.6.7 I3C Controller Reception Flow (FIFO Buffer Transfer)

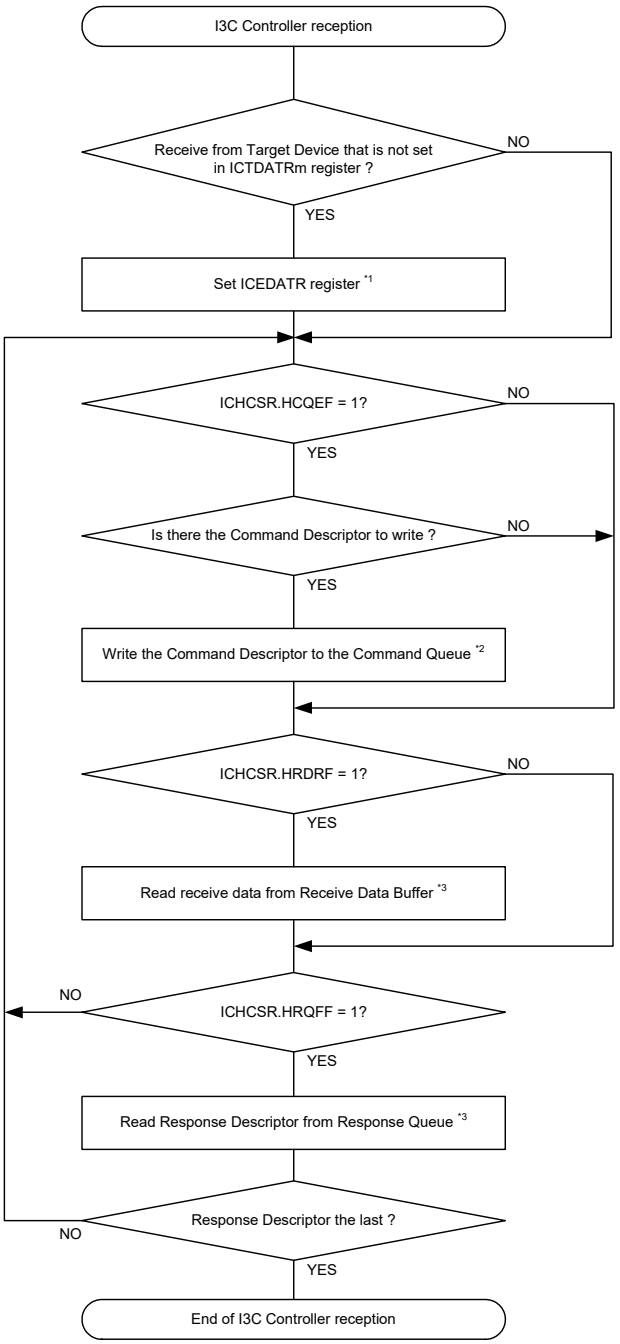
Controller reception flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.36 Example of I3C Controller Reception Flowchart (FIFO Buffer Transfer)

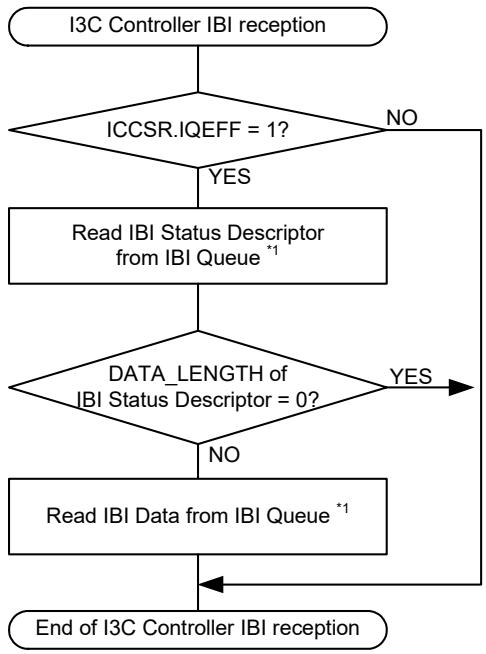
43.4.6.8 I3C Controller Reception Flow (High-Priority FIFO Buffer Transfer)



Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.37 Example of I3C Controller Reception Flowchart (High-Priority FIFO Buffer Transfer)

43.4.6.9 I3C Controller IBI Reception Flow



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.38 Example of I3C Controller IBI Reception Flowchart

43.4.7 I3C Target Operation

43.4.7.1 Dynamic Address Assignment Procedure

After initializing RI3C, the I3C Controller first performs Dynamic Address Assignment Procedure.
The operation of RI3C during the Dynamic Address Assignment Procedure by ENTDAA CCC is described below.

(1) Initial setting (For details, refer to section 43.4.4, Initial Settings)
(2) When ENTDAA CCC is received, RI3C transmits Provisioned ID (the ICPIDHR register and bits b15 to b0 of the ICPIDLR register), BCR (bits b15 to b8 of the ICDCTR register), DCR (bits b7 to b0 of the ICDCTR register) until a Dynamic Address is assigned. (For details, refer to In case of Broadcast CCC (ENTDAA) of 43.5.11 CCC detection function (I3C protocol mode).)
(3) When ENTDAA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in the receive status queue.
(4) Read Receive Status Descriptor via the ICSQR register and check the status.
(5) Read the data for the data length indicated by the DATA_LENGTH field of the Receive Status Descriptor from the Receive Data Buffer via the ICDR register.

Note: When multiple RI3C (I3C Target) are connected on the I3C bus assign Dynamic Addresses in the following order.
1. Set the ICPIDHR and ICPIDLR registers (6 bytes in total) of the RI3C (I3C Target) to a value (all 1 etc.) that has a lower priority than other Target devices by Dynamic Address Arbitration.
2. After setting the Static Address in the RI3C (I3C Target), assign the Dynamic Address using the SETDASA/ SETAASA command.
3. Assign a Dynamic Address to an I3C Target Device other than the RI3C (I3C Target) using the ENTDAA command.

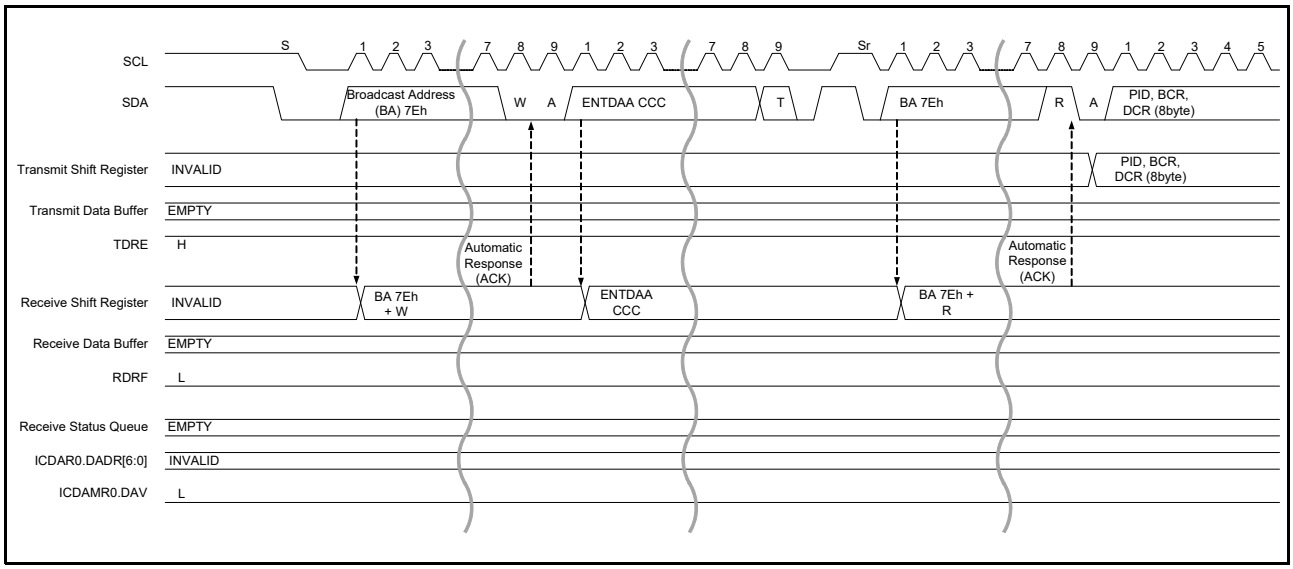


Figure 43.39 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (1/3)

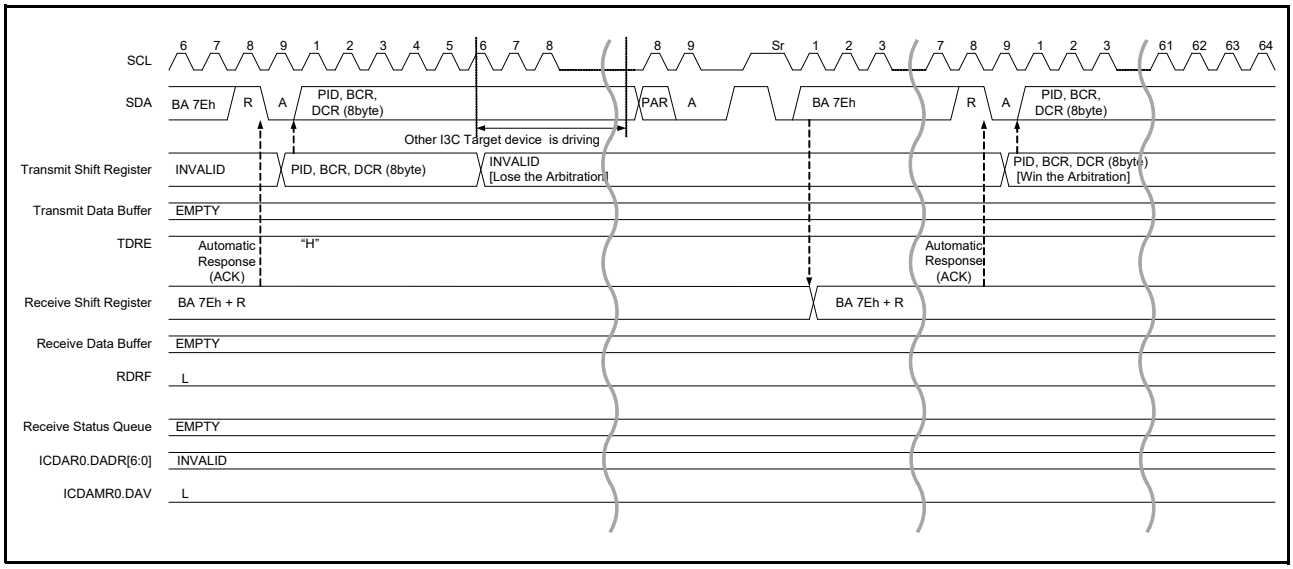


Figure 43.40 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (2/3)

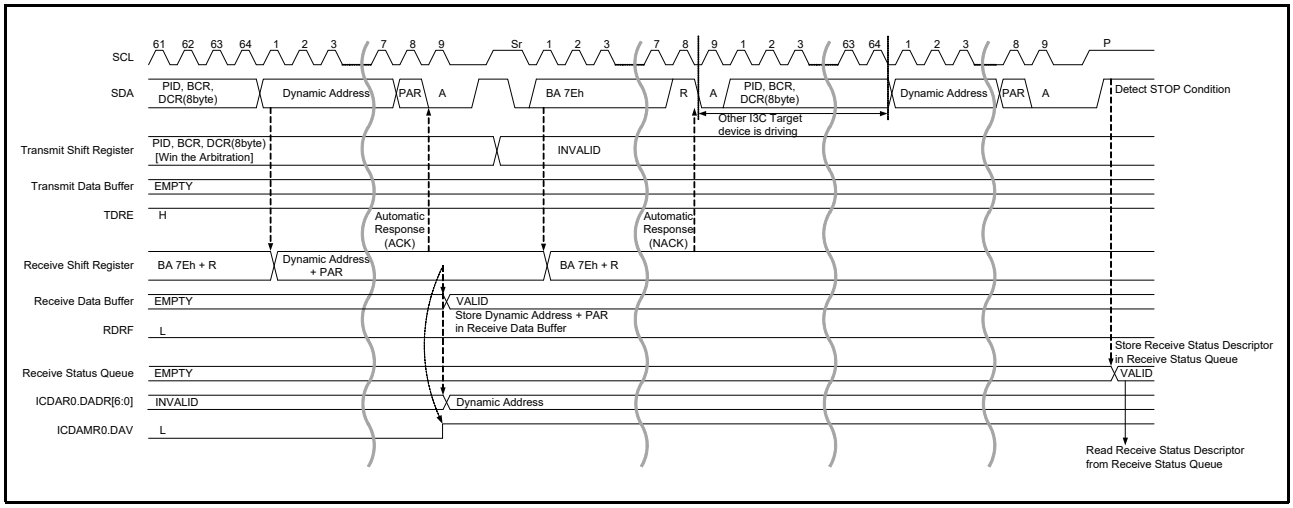


Figure 43.41 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (3/3)

When communicating with a Static Address until the Dynamic Address is assigned from the I3C Controller, by setting to the ICDARy.SADR[6:0] bits, the ICDAMRy.SAV flag is set to 1 and the Static Address Will be effective.

If the I3C Target has a Static Address and the I3C Controller executes the Dynamic Address Assignment Procedure, it is possible to assign a Dynamic Address with SETDASA CCC.

The operation of RI3C during SETDASA CCC Dynamic Address Assignment Procedure is described below.

(1) Initial setting (For details, refer to section 43.4.4, Initial Settings)

(2) When SETDASA CCC which agrees with its own Static Address is received, the ICDARy.DADR[6:0] bits are renewed and the ICDAMRy.DAV flag is set to 1. (For details, refer to In case of Direct Write CCC of 43.5.11 CCC detection function (I3C protocol mode).)

(3) When SETDASA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in receive status queue.

(4) Read Receive Status Descriptor via ICSQR register and check the status.

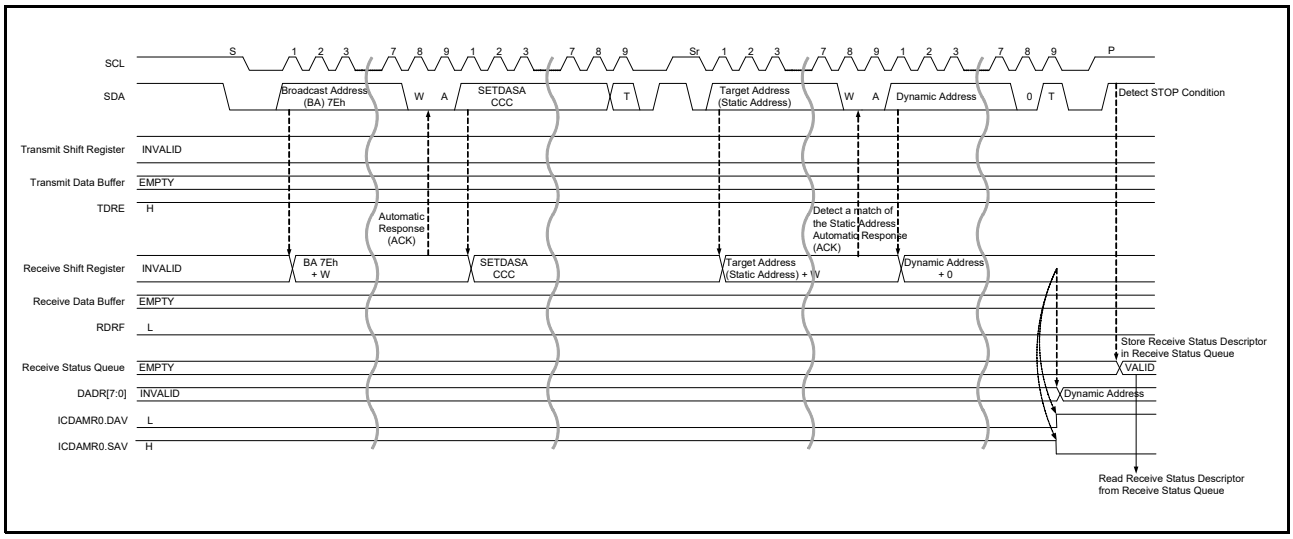


Figure 43.42 Dynamic Address Assignment Procedure (SETDASA CCC) Timing

43.4.7.2 SDR Write Transfer

(1) When Transaction is issued from the I3C Controller, it compares the Target Address in Address header with its own Target Address, and if it matches, RI3C responds with ACK.
When a Transaction is received, if the Receive Data Buffer is full, the I3C Target will respond with NACK in the Address header.
In preparation for retrying the I3C Controller, read the data from the Receive Data Buffer via the ICDR register, and empty the Receive Data Buffer.

(2) Data received from I3C Controller is stored in the Receive Data Buffer.

(3) In the receive data full interrupt, read the received data from the Receive Data Buffer via the ICDR register.

(4) When repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the receive status queue.

(5) Read Receive Status Descriptor via ICSQR register and check the status.

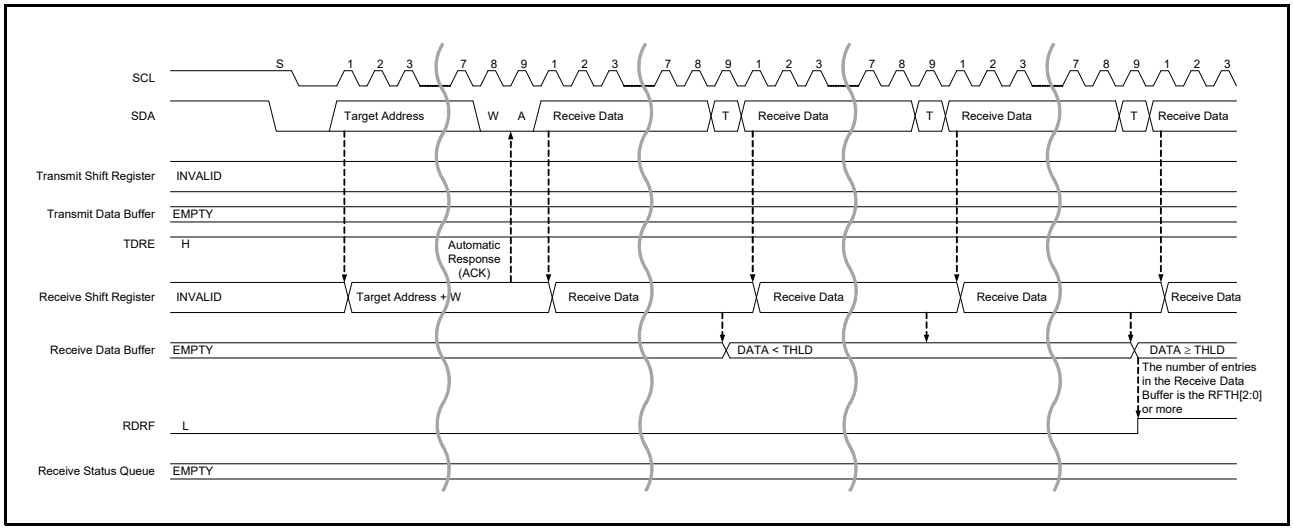


Figure 43.43 SDR Write Transfer Timing (1/2)

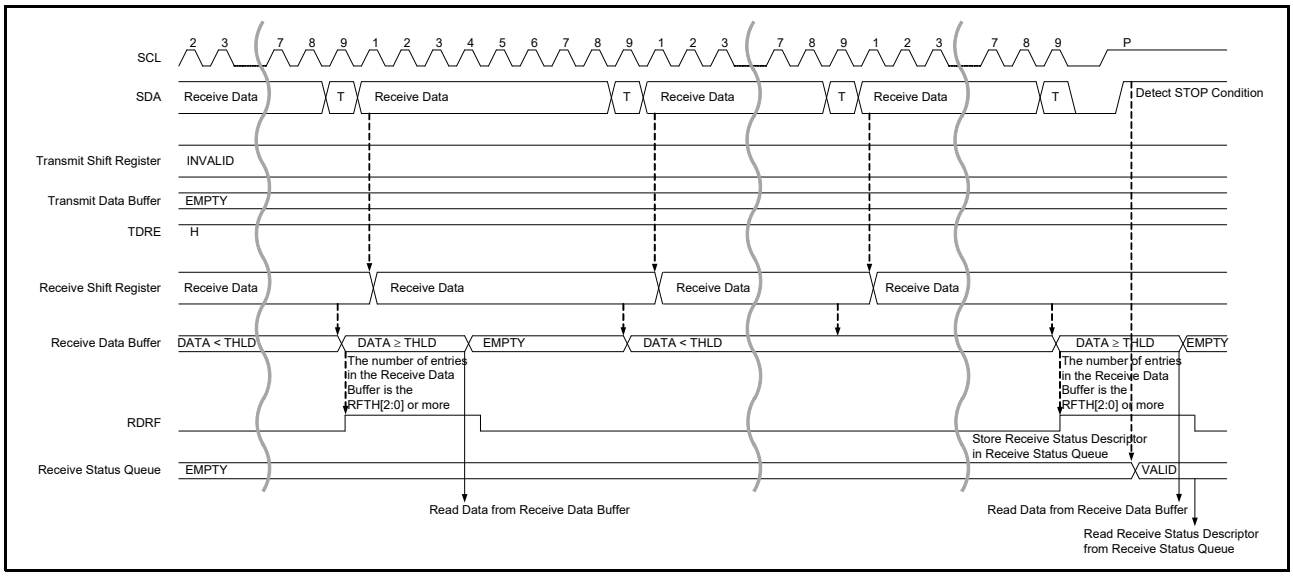


Figure 43.44 SDR Write Transfer Timing (2/2)

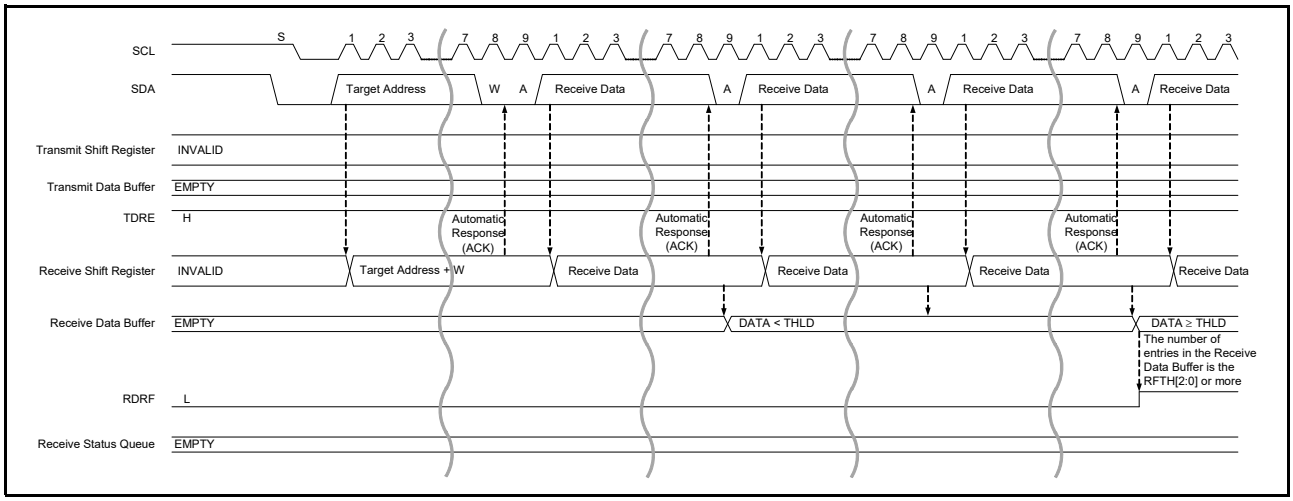


Figure 43.45 Legacy I²C Write Transfer Timing (1/2)

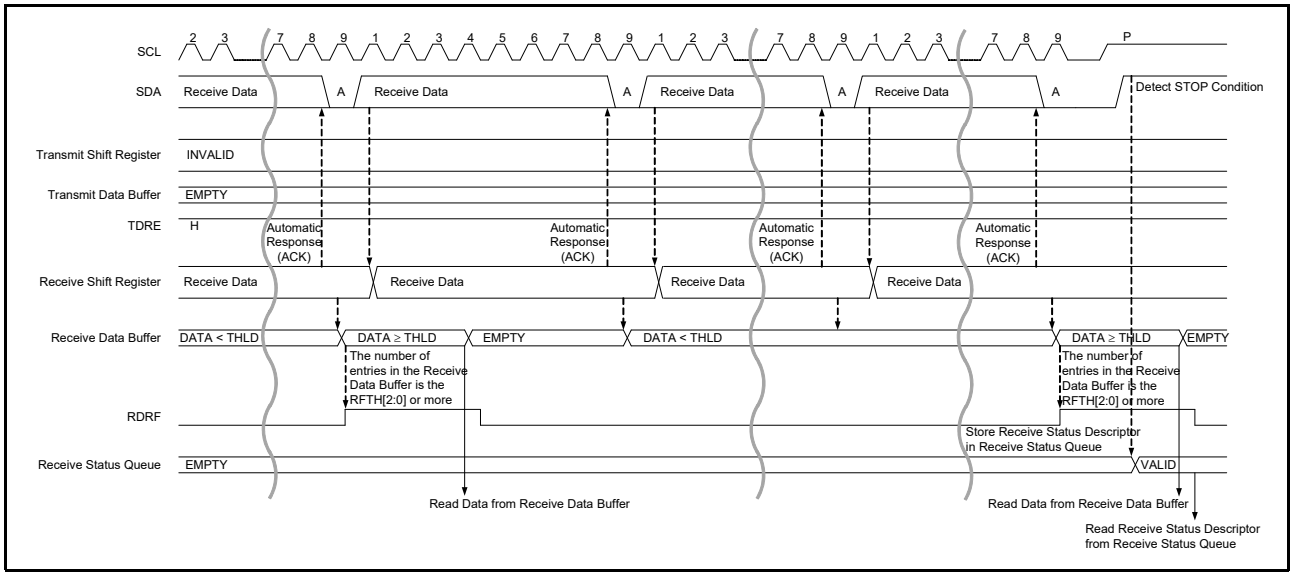


Figure 43.46 Legacy I²C Write Transfer Timing (2/2)

43.4.7.3 SDR Read Transfer

(1) Write the data requested from the I3C Controller to the Transmit Data Buffer via the ICDR register.

(2) When Transaction is issued from the I3C Controller, it compares the Target Address in Address header with its own Target Address, and if it matches, RI3C responds with ACK.
When a Transaction is received, if the Transmit Data Buffer is EMPTY, I3C Target responds with NACK with the Address header.
In preparation for retrying the I3C Controller, write data to the Transmit Data Buffer via the ICDR register.

(3) Transmit the data stored in the Transmit Data Buffer.

(4) If data to be transmitted still remains, write the data to be transmitted with a transmit data empty interrupt to the Transmit Data Buffer via the ICDR register.

(5) SDR:
When the transmission of the data stored in the Transmit Data Buffer is completed, Low is output to the T-bit following Data, and it is notified to the I3C Controller that it is the final data.
Legacy I^2C Message:
When NACK is detected, data transmission is terminated.

(6) When a repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the receive status queue.

(7) Read the Receive Status Descriptor via ICSQR and check the status.
If the data length does not match, set the ICRCR.ISRST bit to 1 and then reset the internal states of this module.
For details, refer to 43.5.16.6 Error Recovery Operation.

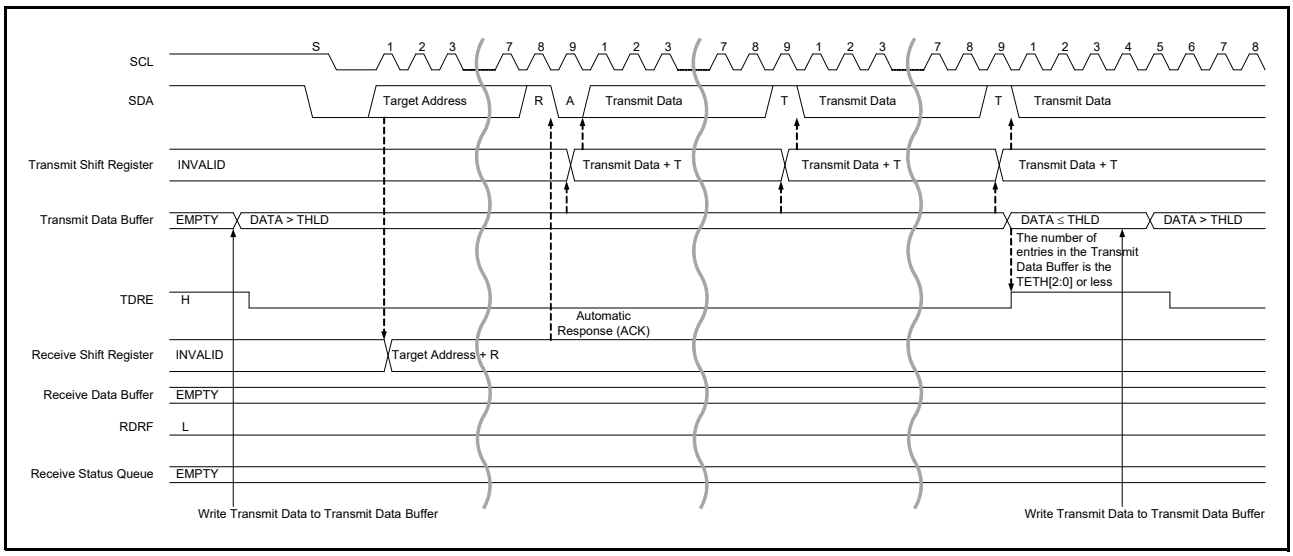


Figure 43.47 SDR Read Transfer Timing (1/2)

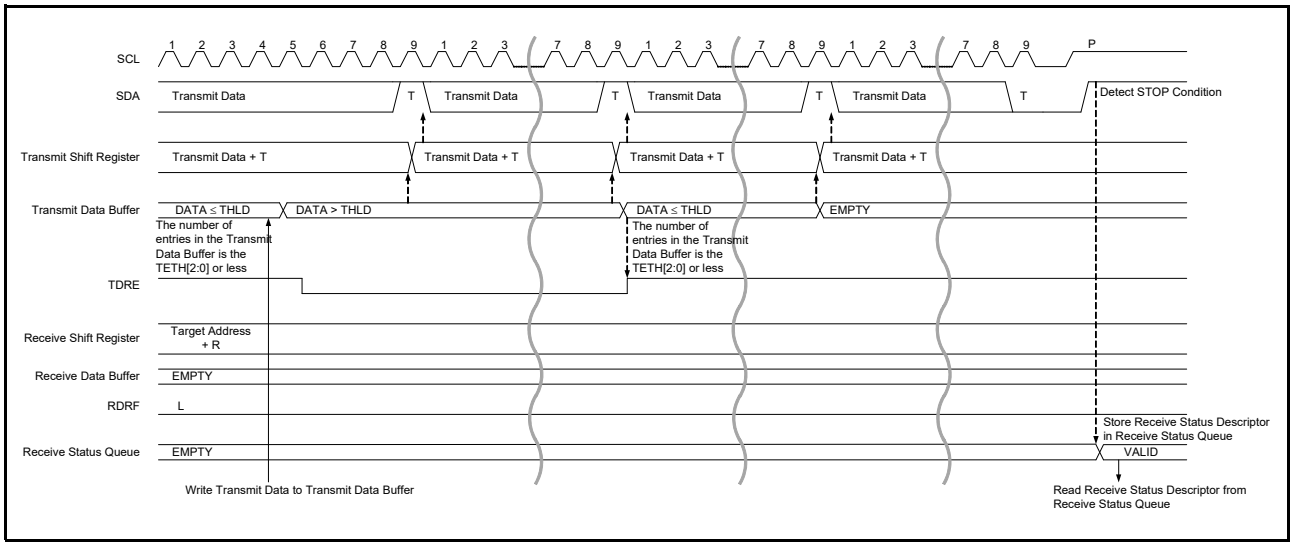


Figure 43.48 SDR Read Transfer Timing (2/2)

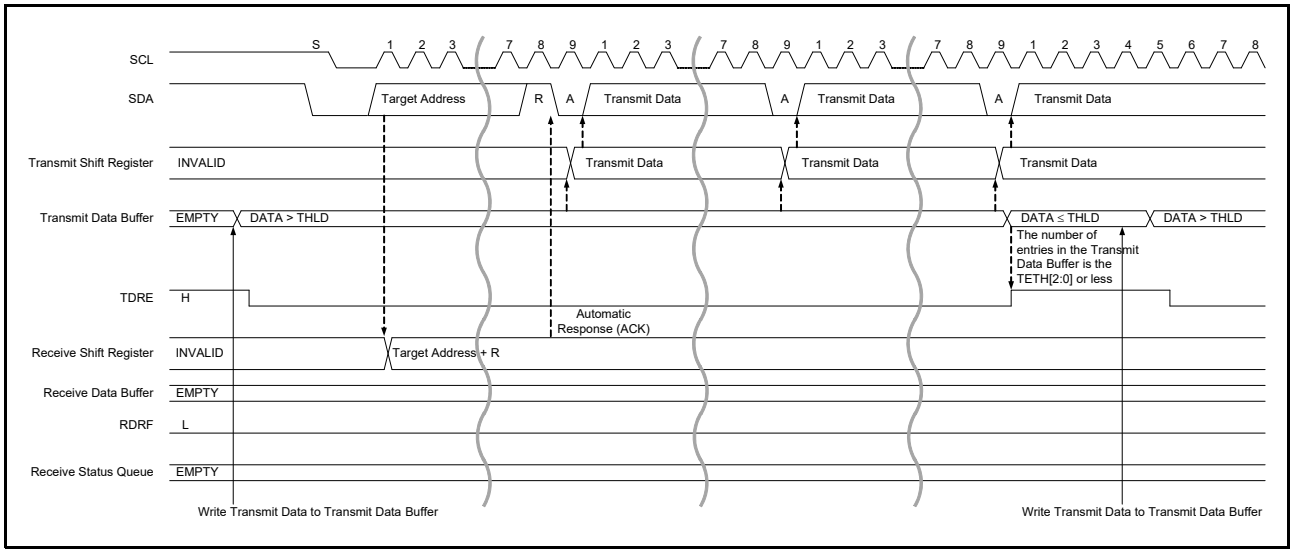


Figure 43.49 Legacy I²C Read Transfer Timing (1/2)

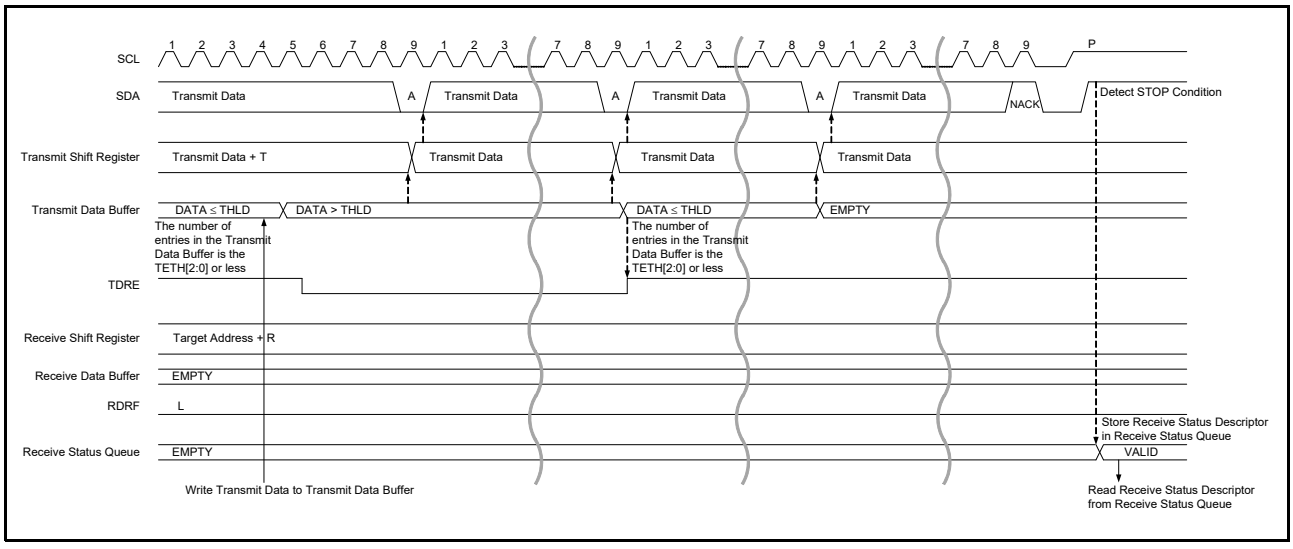


Figure 43.50 Legacy I²C Read Transfer Timing (2/2)

43.4.7.4 IBI Transfer

(1) When sending Target Interrupt Request.
When transmitting IBI Data, write IBI Data to the IBI Data Buffer via the ICIQR register.

(2) Write Command Descriptor (immediate data transfer command or regular data transfer command) to the command queue for IBI Transfer via the ICCQR register.

(3) When Command Descriptor is written to command queue, IBI Transaction is issued under the following conditions.

- Detect a START condition. (Does not apply a repeated START condition)
- If no START condition is forthcoming within the following bus condition, then this module issue a START Request by pulling the SDA line Low.
- (a) Target Interrupt Request, Controller Role Request: Bus Available

(4) In Target Address and RnW field in the Address header, if losing Arbitration by issuing a Transaction from I3C Controller, stop issuing Transaction.
When detecting repeated START condition or STOP condition, store the Response Descriptor into the response queue.

(5) When sending Target Interrupt Request:

- When IBI data for transmission still remain, write IBI data with an IBI queue empty/full interrupt to the IBI queue via the ICIQR register.
- When the transmission of IBI Data for the number of data length specified by the DATA_LENGTH field of the Command Descriptor is completed, output Low to the T-bit following IBI Data and notify the I3C Controller that it is the final IBI Data.

(6) When detecting repeated START condition or STOP condition, store the Response Descriptor into the response queue.

(7) Read the Response Descriptor from the response queue with the ICRQR register and check the status. If NACK is responded, repeat steps (1) to (7).

(8) When sending Target Interrupt Request:
Check that the value of the DATA_LENGTH field of the Response Descriptor is 0.

The CRR processing flow is shown in Figure 43.53.

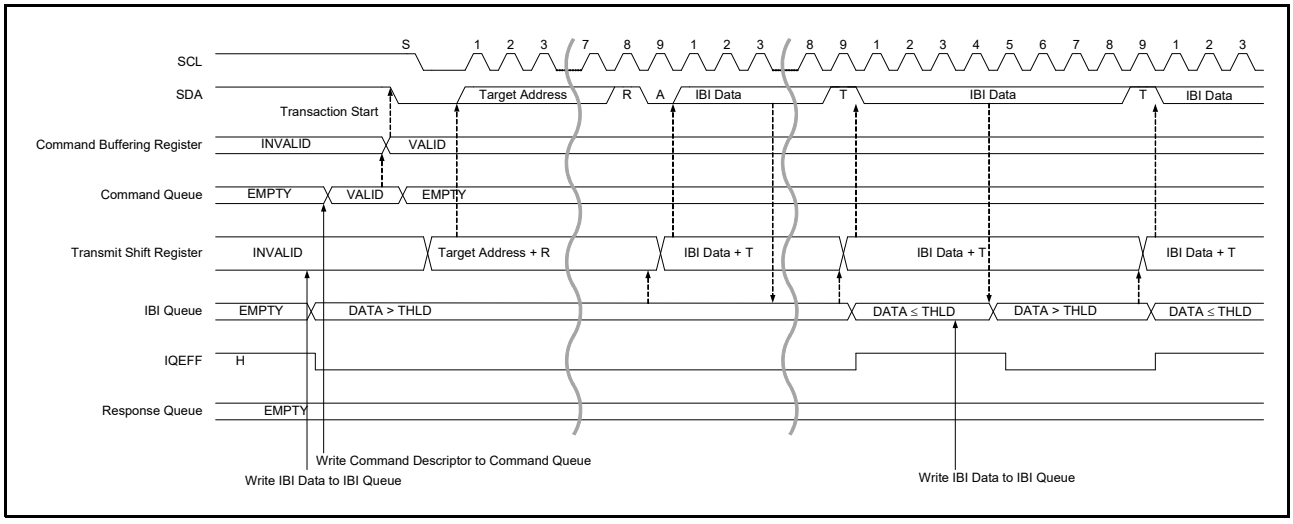


Figure 43.51 I3C Target IBI Transfer Timing (1/2)

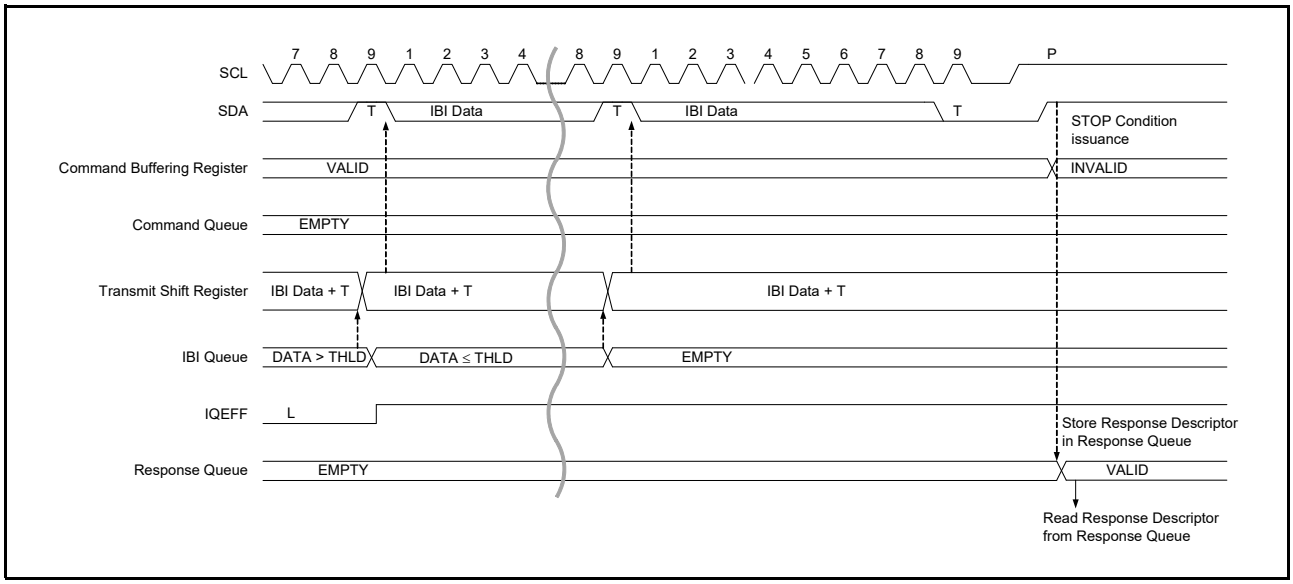


Figure 43.52 I3C Target IBI Transfer Timing (2/2)

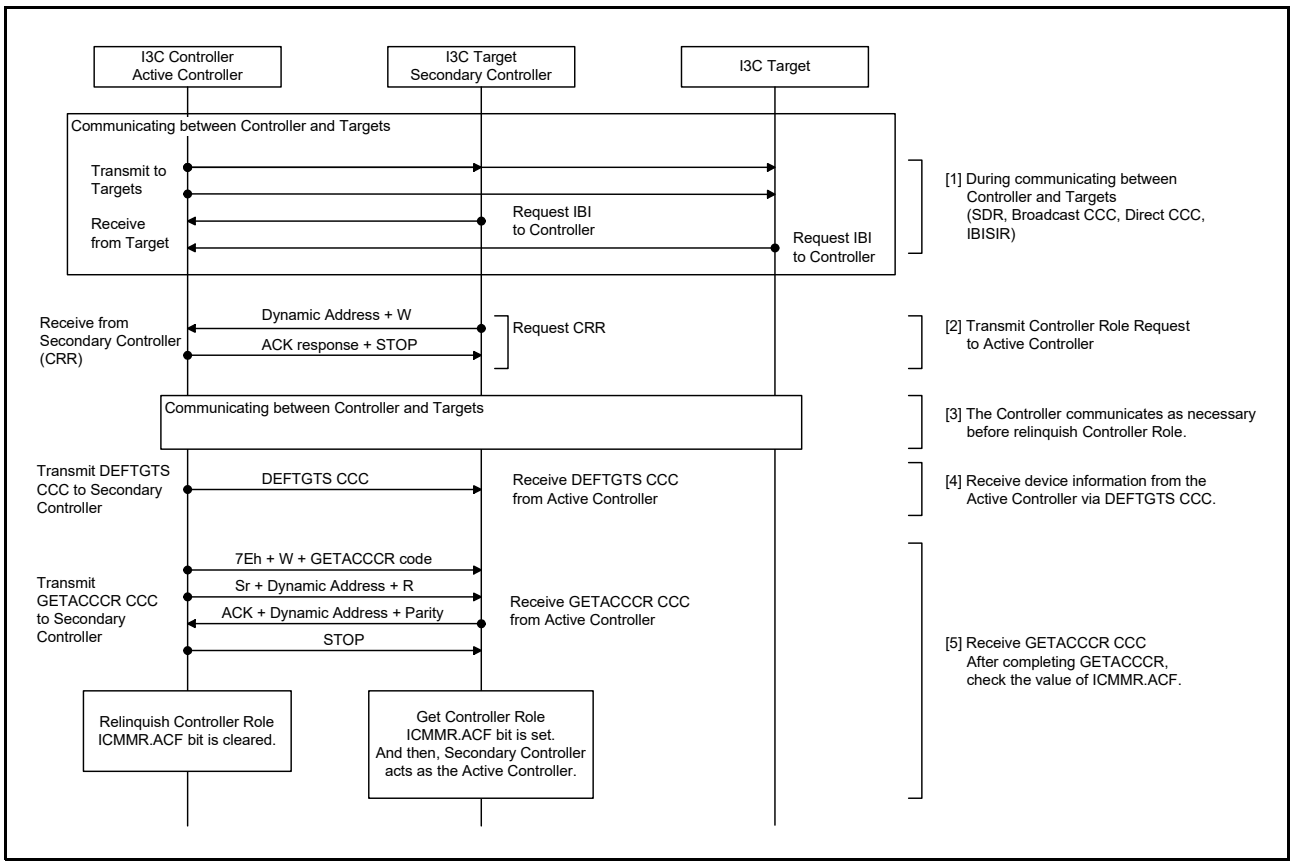
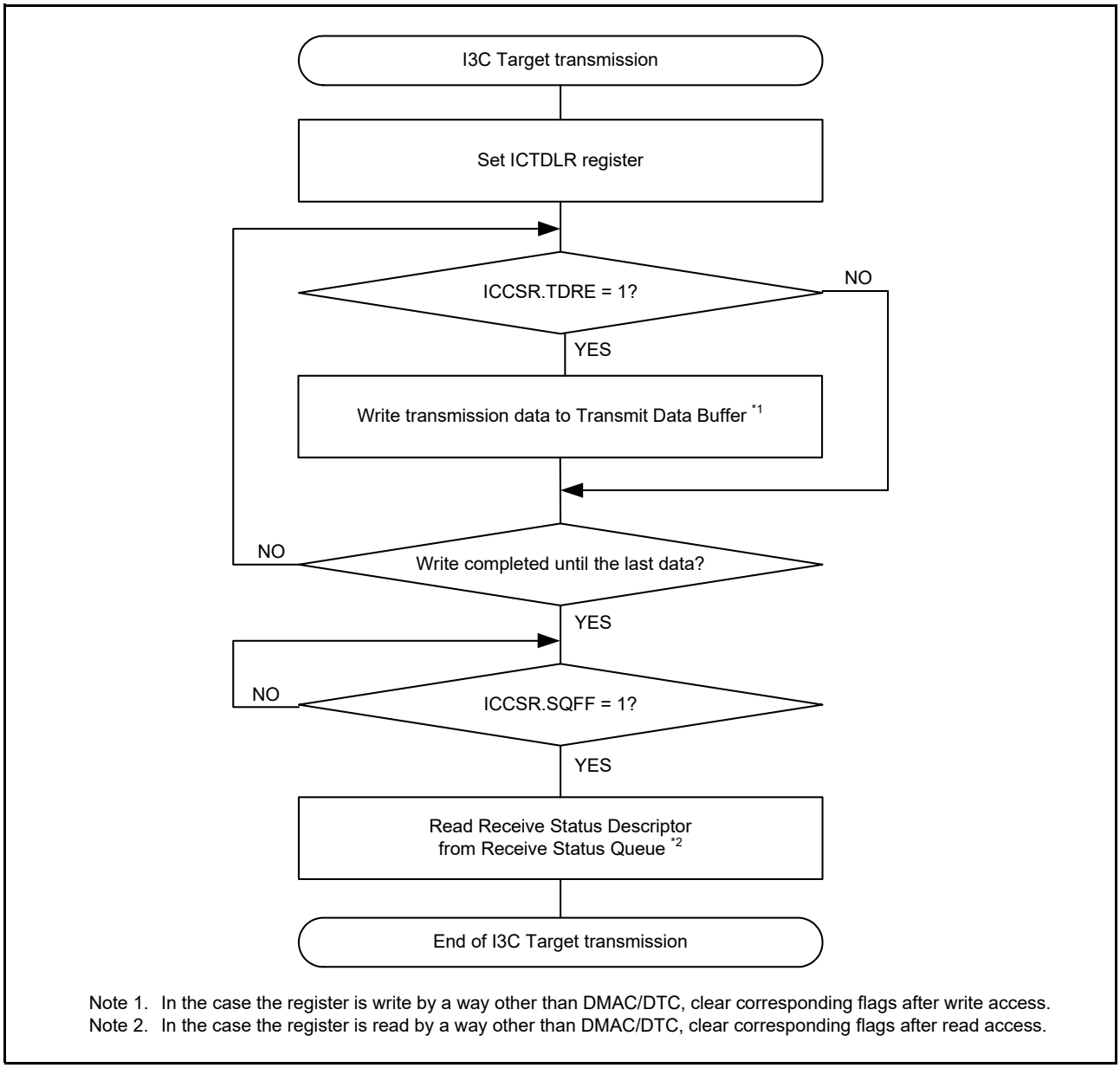


Figure 43.53 I3C Target CRR Processing Flow

43.4.7.5 I3C Target Transmission Flow (FIFO Buffer Transfer)

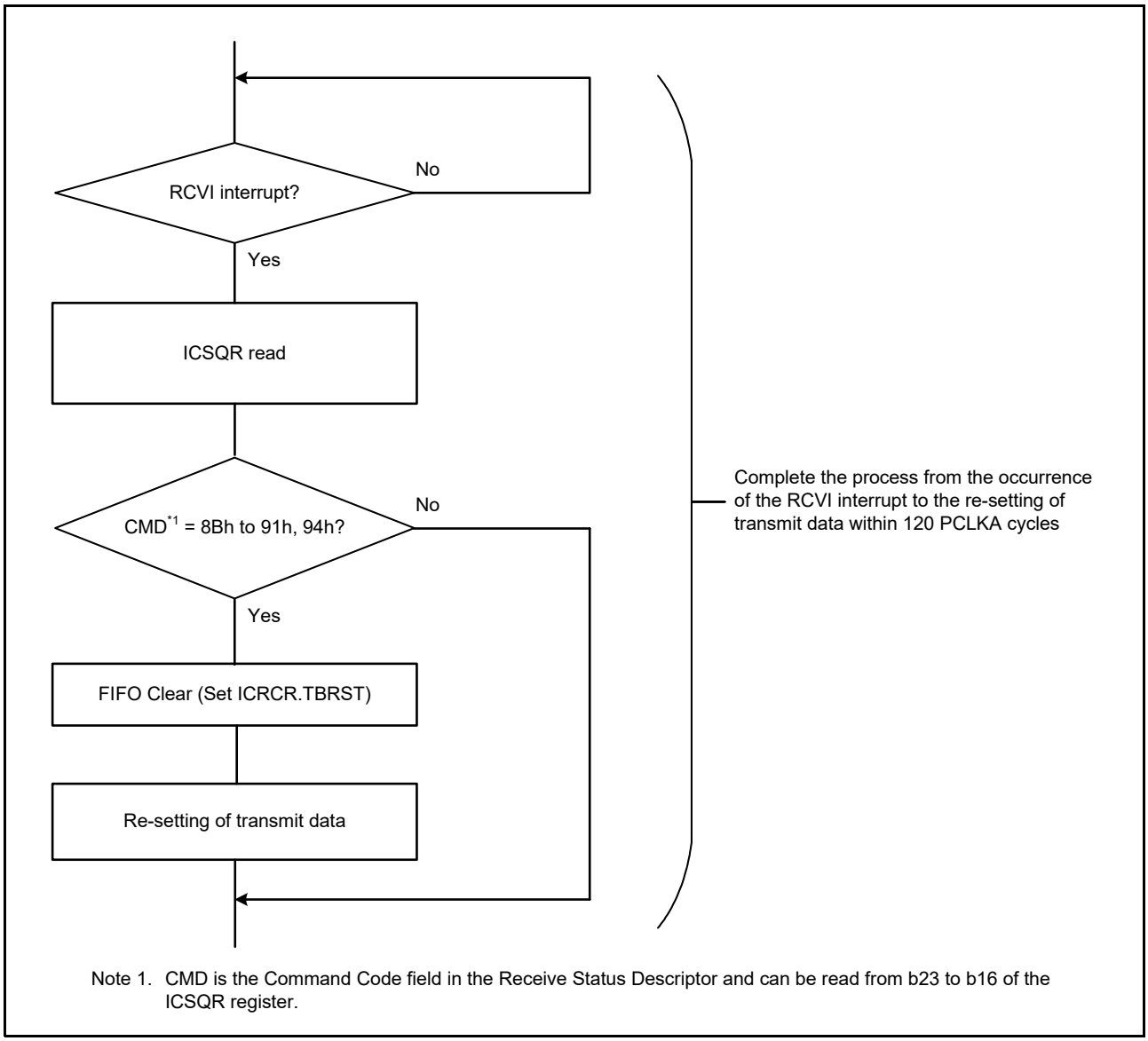
Target Transmission Flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.54 Example of I3C Target Transmission Flowchart (FIFO Buffer Transfer)

When using the RI3C as an I3C Target, if I3C Target receives GET CCC while data exists by writing from the ICDR register to the transmit buffer, follow the flow below.

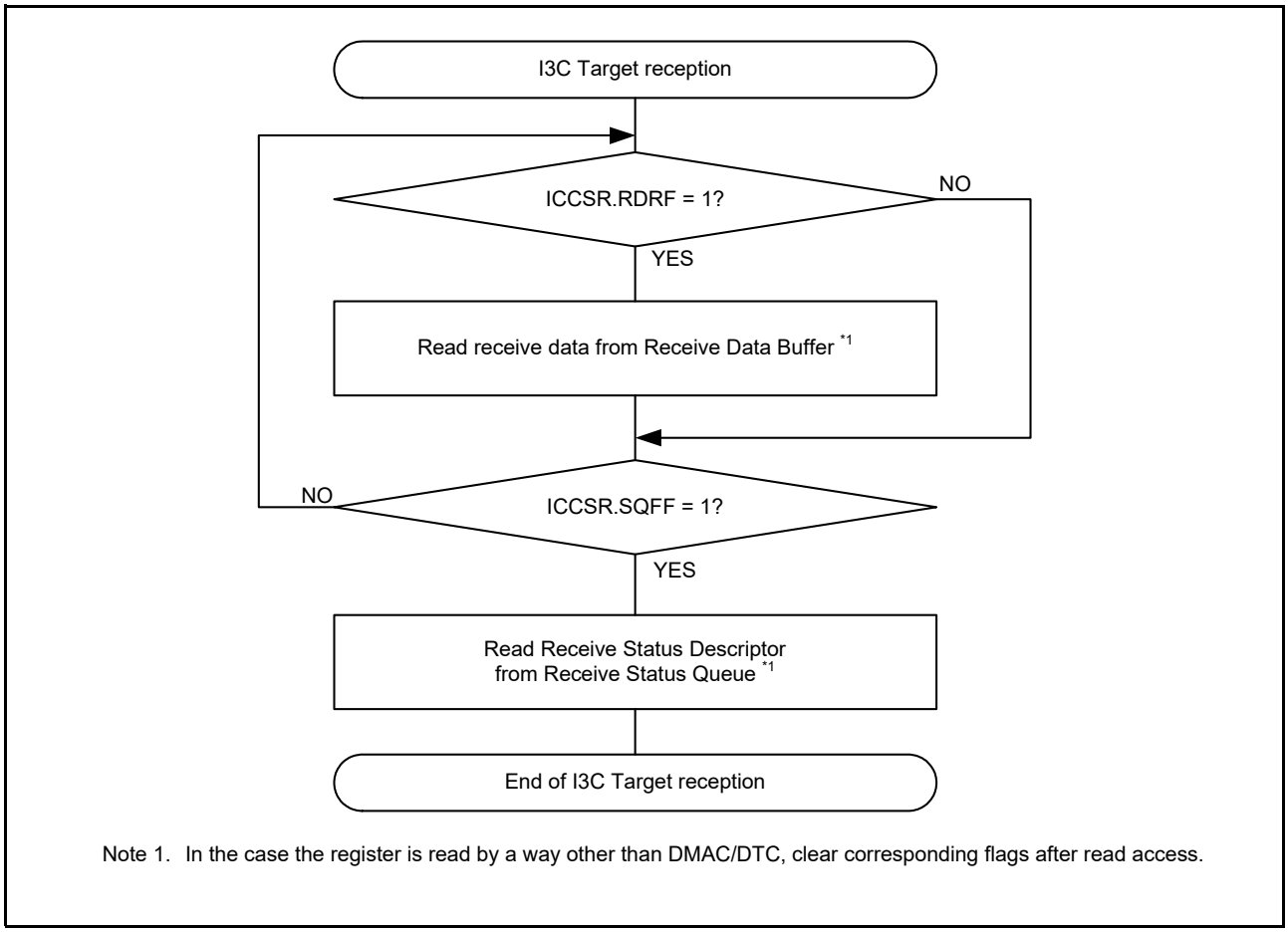


Note 1. CMD is the Command Code field in the Receive Status Descriptor and can be read from b23 to b16 of the ICSQR register.

Figure 43.55 I3C Target Receives GET CCC Command while Data Exists by Writing from the ICDR Register to the Transmit Buffer

43.4.7.6 I3C Target Reception Flow (FIFO Buffer Transfer)

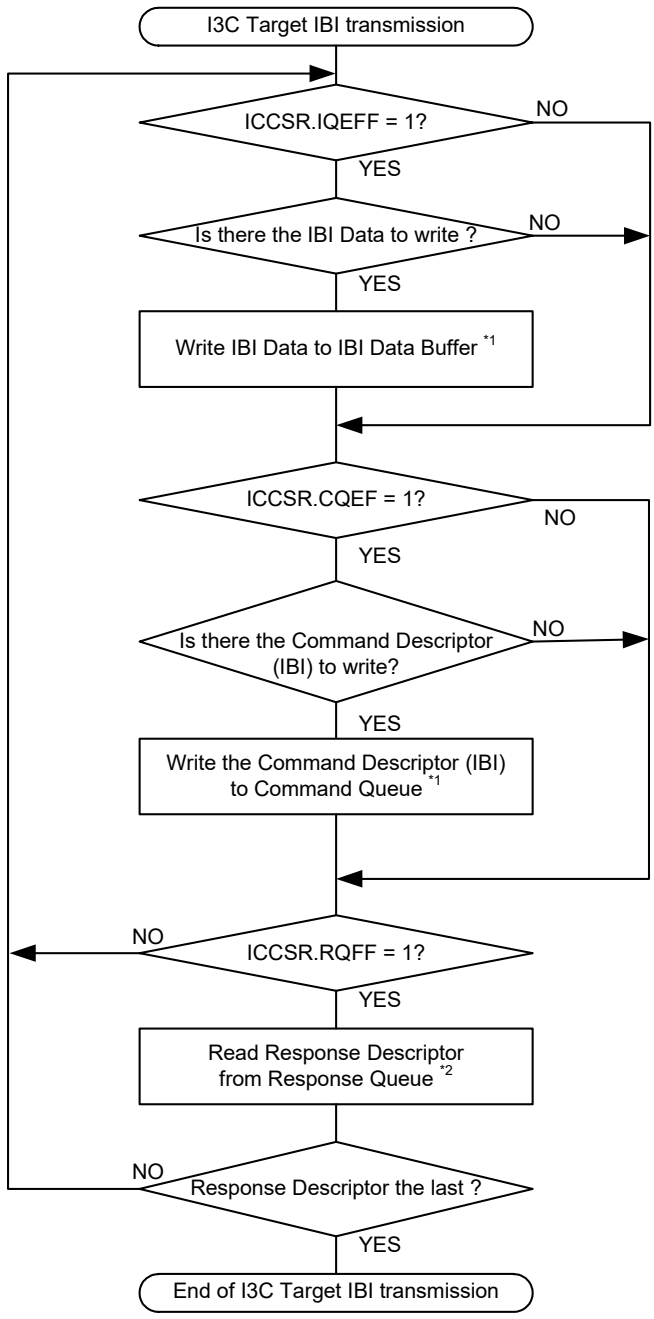
Target Reception Flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.56 Example of I3C Target Reception Flowchart (FIFO Buffer Transfer)

43.4.7.7 I3C Target IBI Transmission Flow



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 43.57 Example of I3C Target IBI Transmission Flowchart

43.4.8 I2C Master Operation

43.4.8.1 Master Transmit Operation

In master transmit operation, the RI3C outputs the SCL and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 43.58 shows an example of usage of master transmission and Figure 43.59 to Figure 43.61 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 43.4.4, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RI3C generates a START condition. At the same time, the BFREE flag is automatically set to 0, the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDA00 line have matched while the ICCGR.ST bit is 1, the RI3C recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.ACF and ICMMR.TRS flags are automatically set to 1, placing the RI3C in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS flag is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RI3C continues in master transmit mode. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCGR.SP bit to generate a STOP condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDR register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDR register.
(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RI3C automatically holds the SCL00 line low until the data for transmission are ready or a STOP condition is generated
(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCGR.SP bit to 1 (requests to generate a STOP condition). Upon receiving a STOP condition generation request, the RI3C generates the STOP condition.
(6) Upon detecting the STOP condition, the RI3C automatically sets the ICMMR.ACF and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, it automatically sets the ICCSR.TDRE flag to 0, ICSR2.TEND flag to 0 and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

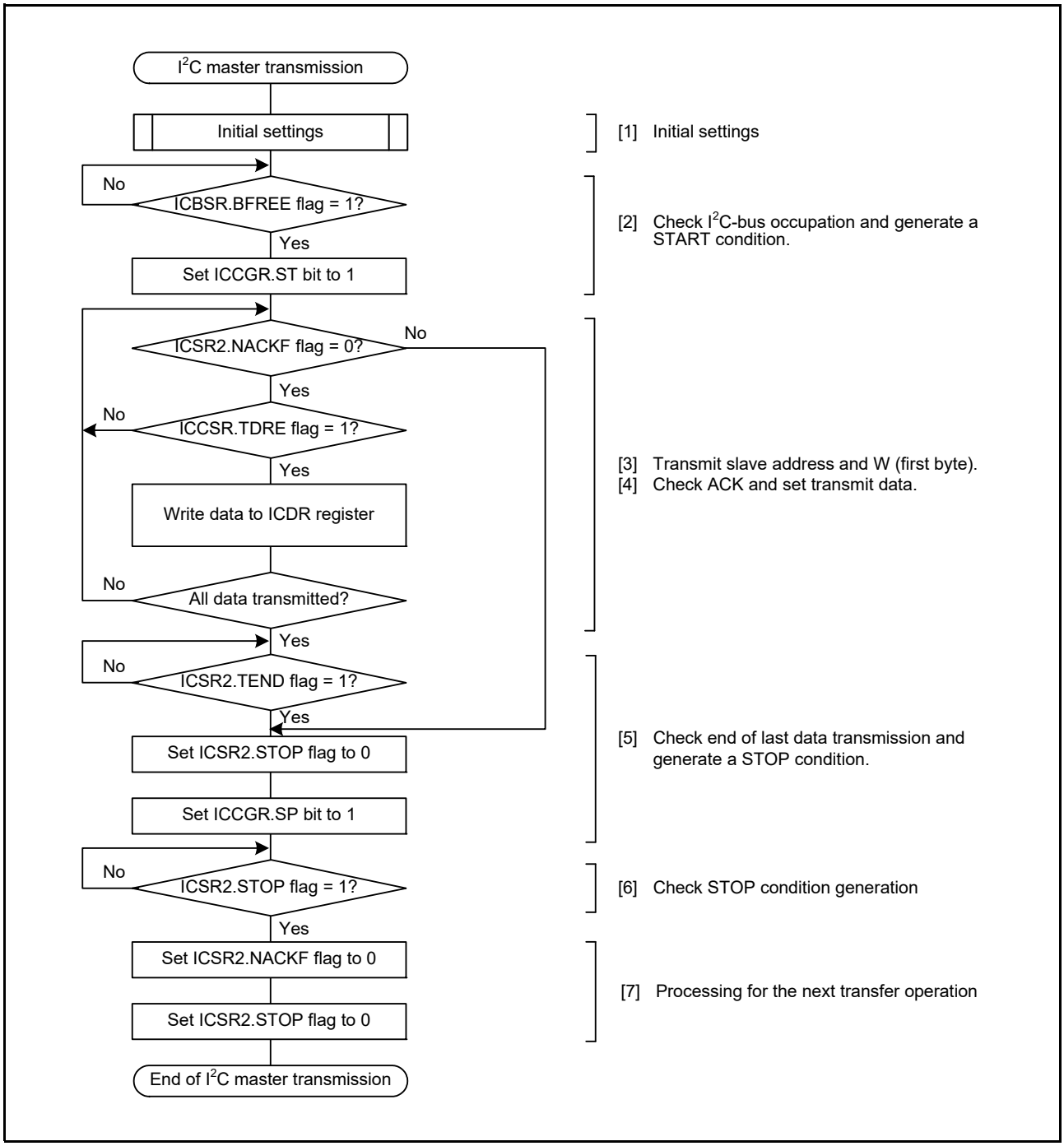


Figure 43.58 Example of I2C Master Transmission Flowchart

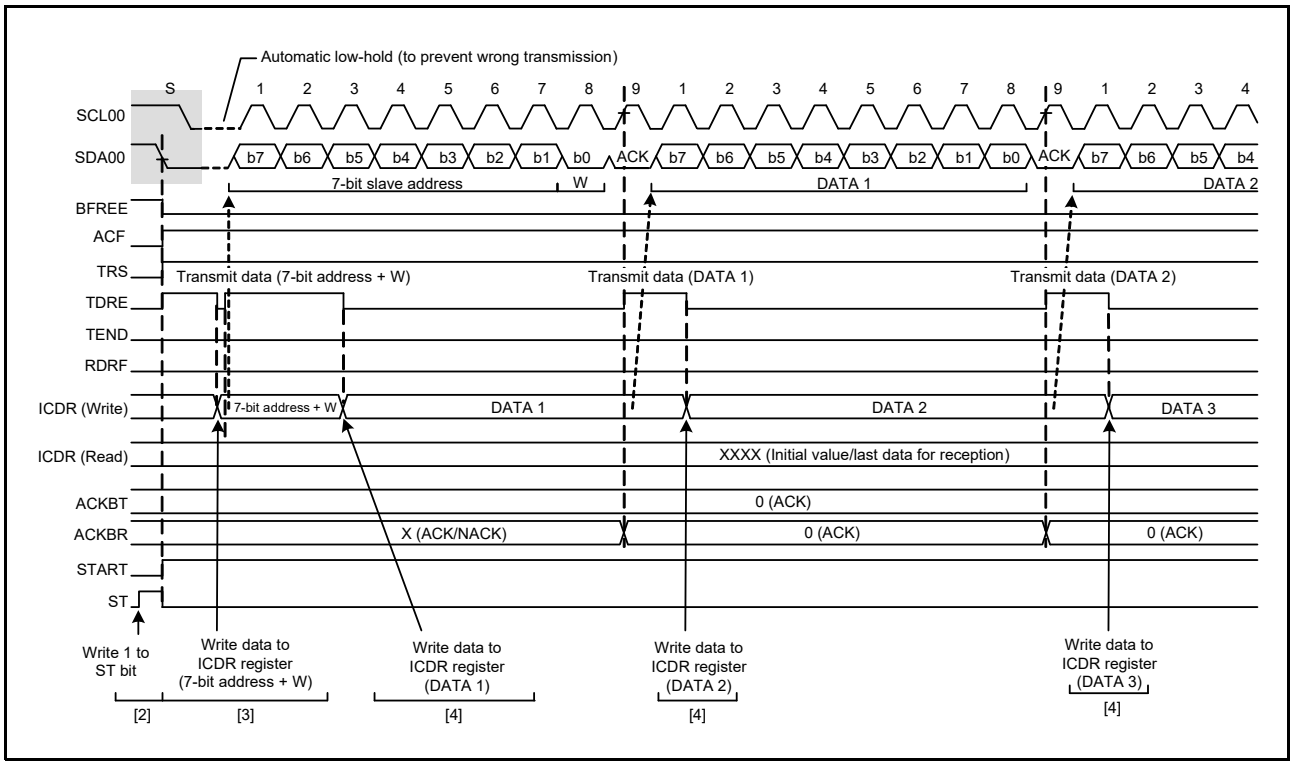


Figure 43.59 Master Transmit Operation Timing (1) (7-Bit Address Format)

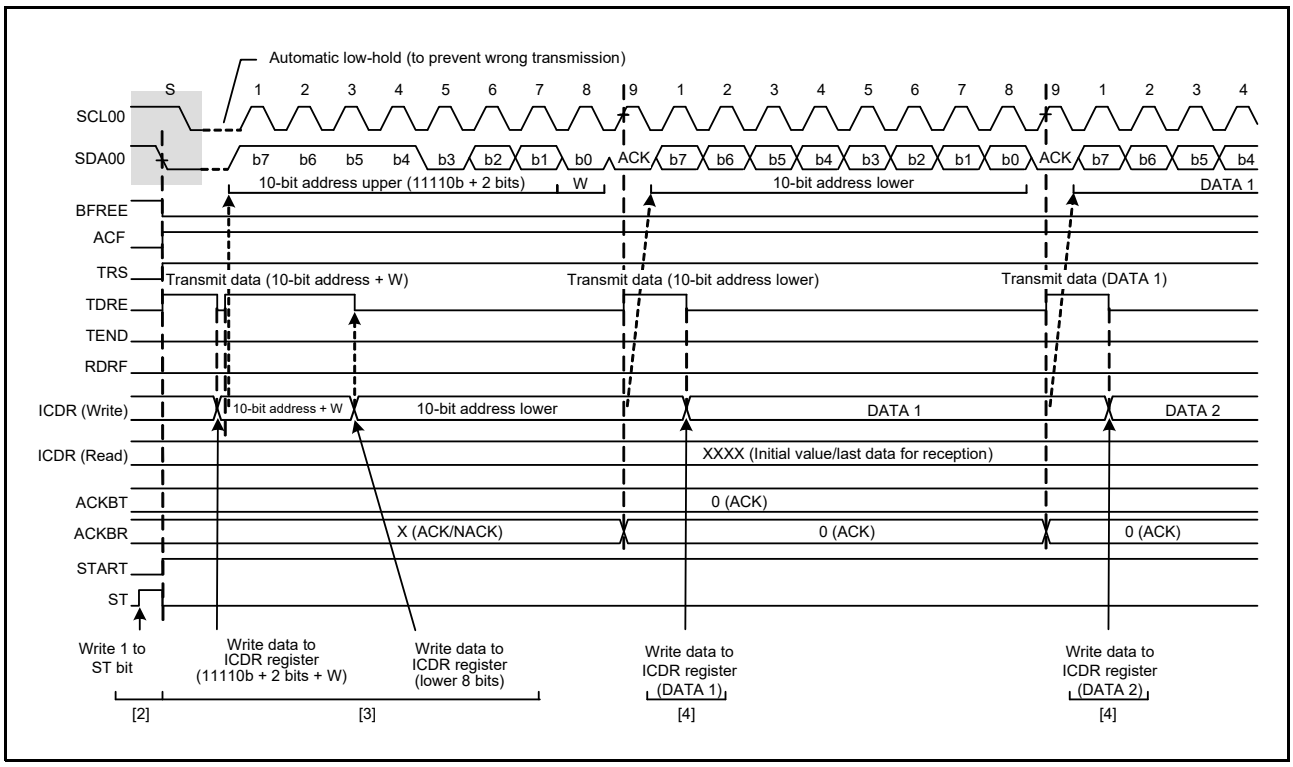


Figure 43.60 Master Transmit Operation Timing (2) (10-Bit Address Format)

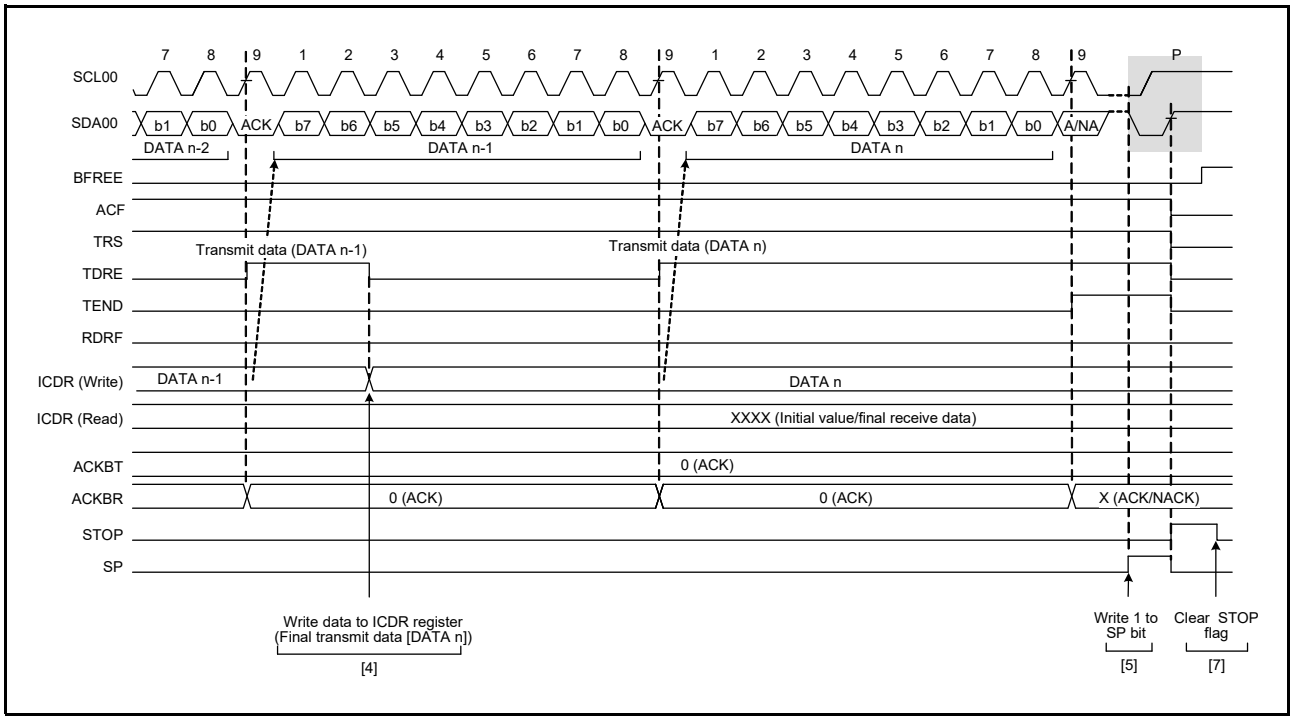


Figure 43.61 Master Transmit Operation Timing (3)

43.4.8.2 Master Receive Operation

In master receive operation, the RI3C as a master device outputs the SCL, receives data from the slave device, and returns acknowledgments. Since the RI3C must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 43.62 and Figure 43.63 show examples of usage of master reception and Figure 43.64 to Figure 43.66 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 43.4.4, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RI3C generates a START condition. When the RI3C detects the START condition, the ICBSR.BFREE flag is automatically set to 0 and the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the SDA00 line have matched while the ICCGR.ST bit is 1, the RI3C recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.ACF and ICMMR.TRS flags are automatically set to 1, placing the RI3C in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICMMR.TRS flag is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS flag is set to 0 on the rising edge of the ninth SCL, placing the RI3C in master receive mode. At this time, the ICCSR.TDRE flag is set to 0 and if the slave device acknowledges the ICCSR.RDRF flag is automatically set to 1.
If no slave device recognized the address or there was an error in communications, the ICSR2.NACKF flag becomes 1. In this case, write 1 to the ICCGR.SP bit to generate a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a repeated START condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RI3C in master receive mode.
(4) Dummy read ICDR register after confirming that the ICCSR.RDRF flag is 1; this makes the RI3C start output of the SCL and start data reception.
(5) After 1 byte of data has been received, the ICCSR.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICCSCR.WAITAE bit. Reading out ICDR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICACKR.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICCSCR.WAITRE bit to 1 (for wait insertion) before reading the ICDR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICACKR.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL00 line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a STOP condition is possible.
(6) When the ICCSCR.WAITAE bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICACKR.ACKBT bit to 1 (NACK).
(7) After reading out the byte before last from the ICDR register, if the value of the ICCSR.RDRF flag is confirmed to be 1, write 1 to the ICCGR.SP bit (requests to generate a STOP condition) and then read the last byte from ICDR register. When the ICCGR.SP bit is set to 1, the RI3C is released from the wait state and generates the STOP condition after low-level output in the ninth clock pulse is completed or the SCL00 line is released from the low-hold state.

(8) Upon detecting the STOP condition, the RI3C automatically sets the ICMMR.ACF and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

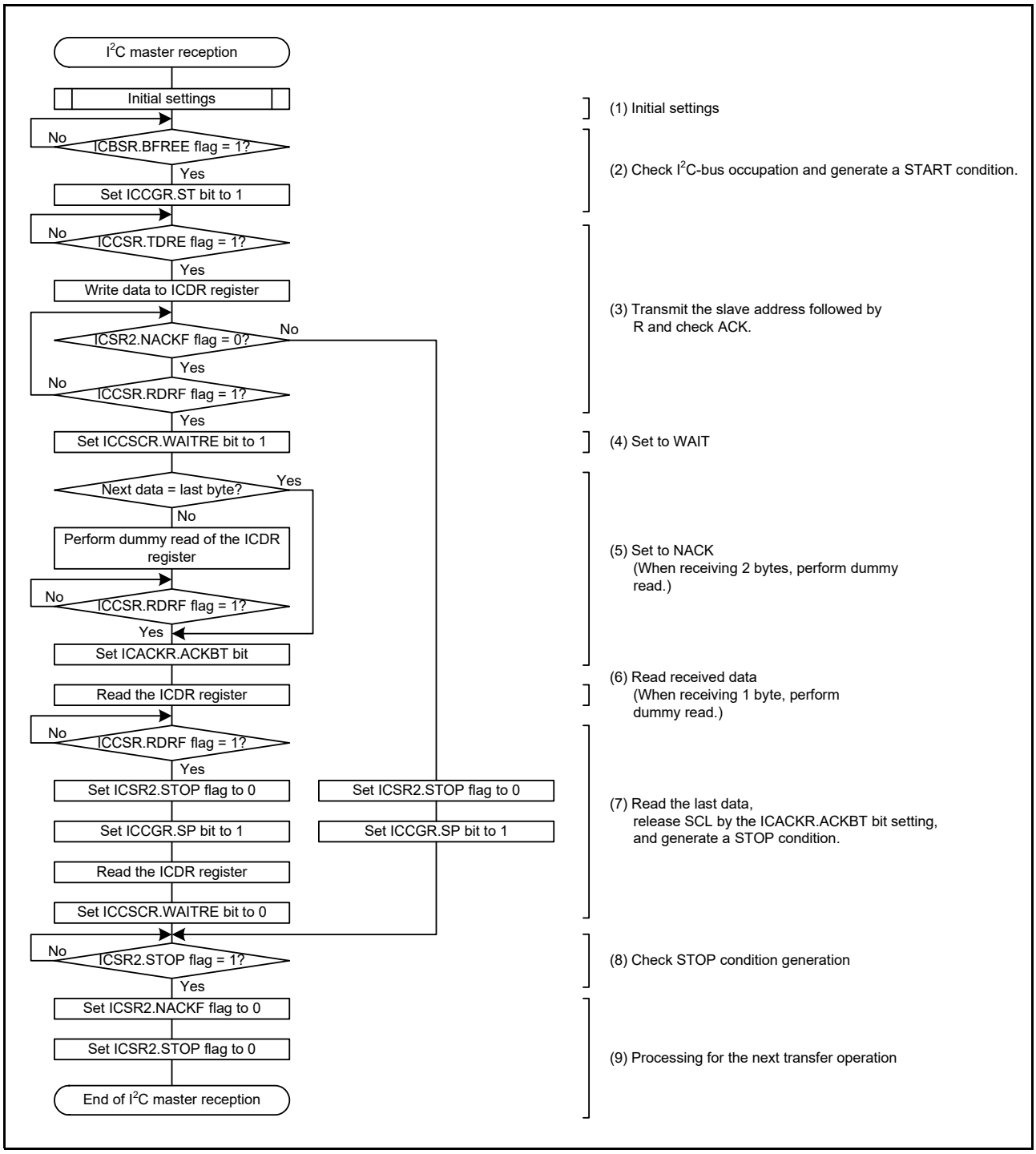


Figure 43.62 Example of I²C Master Reception Flowchart (7-Bit Address Format, 1 or 2 bytes)

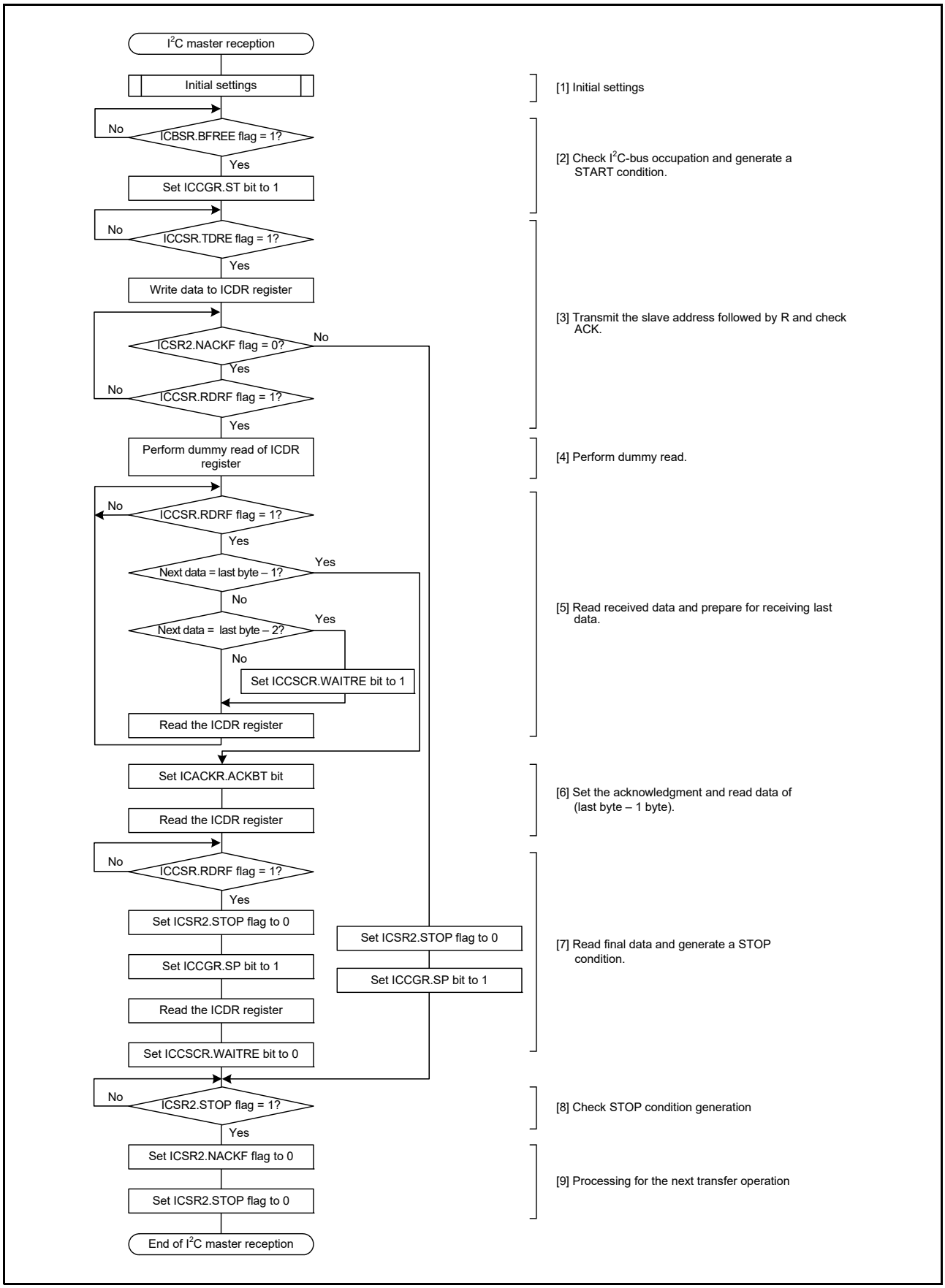


Figure 43.63 Example of I²C Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

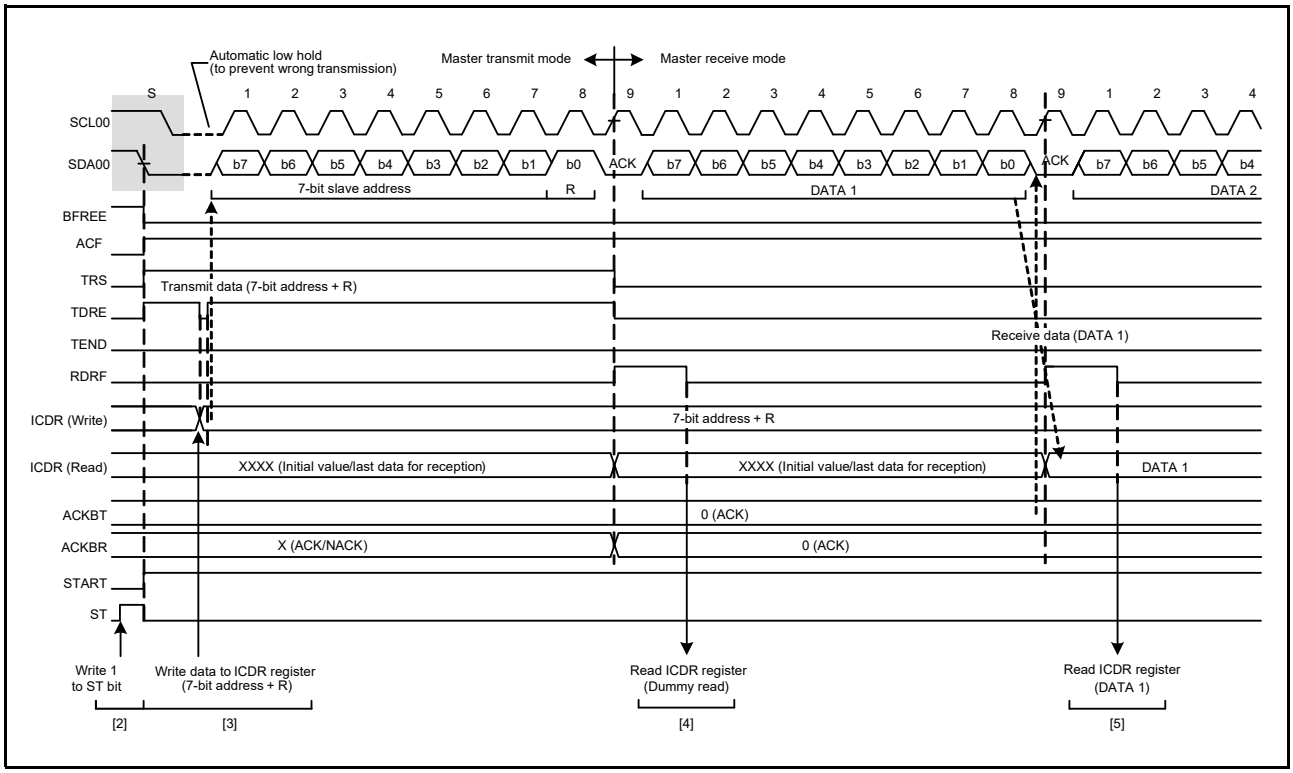


Figure 43.64 Master Receive Operation Timing (1) (7-Bit Address Format, When WAITAE = 0)

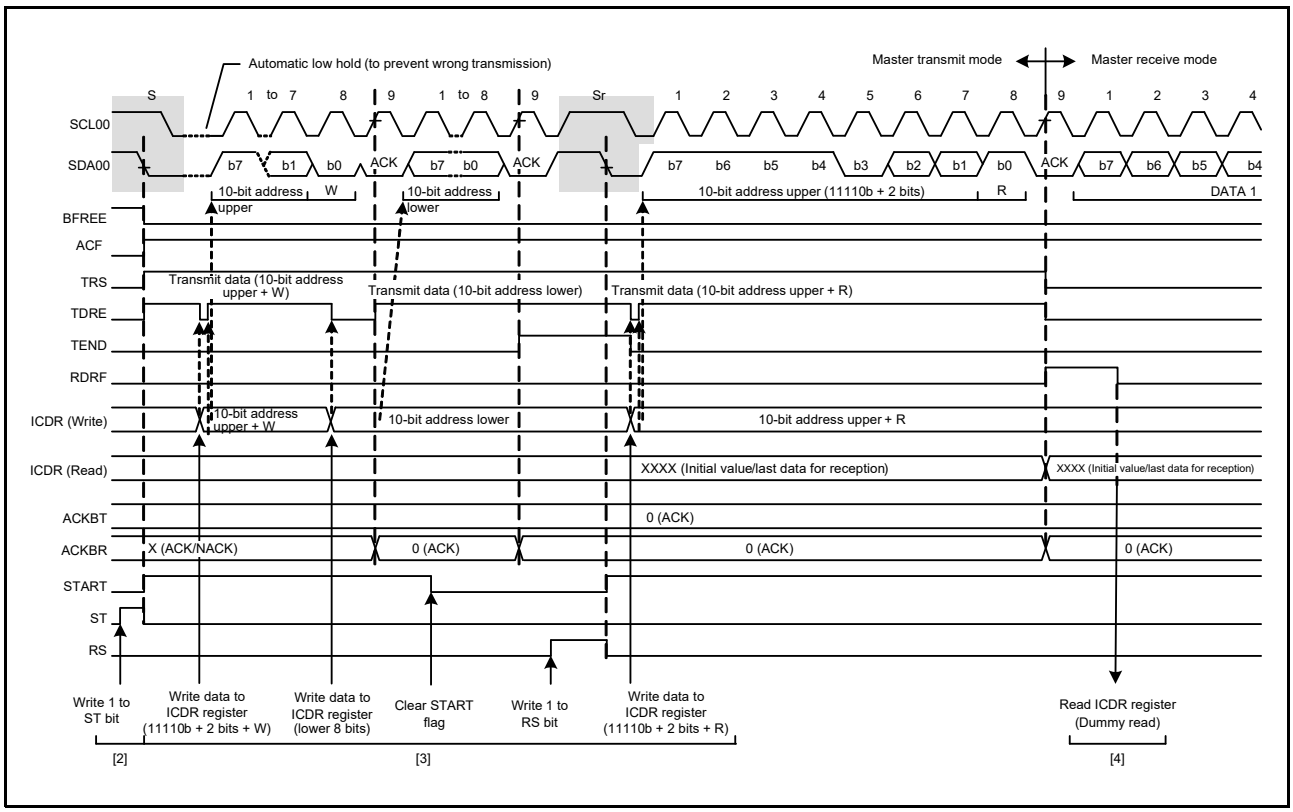


Figure 43.65 Master Receive Operation Timing (2) (10-Bit Address Format, When WAITAE = 0)

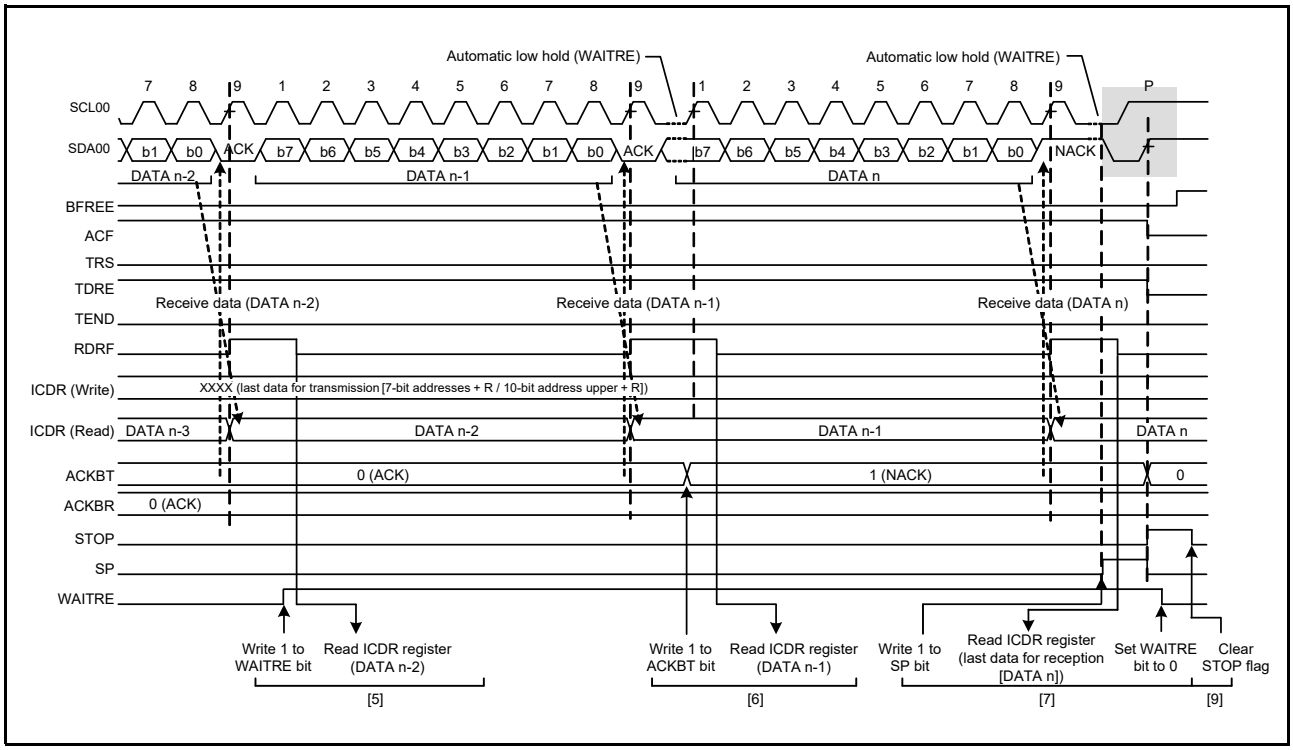


Figure 43.66 Master Receive Operation Timing (3) (When WAITAE = 0)

43.4.9 I2C Slave Operation

43.4.9.1 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RI3C transmits data as a slave device, and the master device returns acknowledgments.
Figure 43.67 shows an example of usage of slave transmission and Figure 43.68 and Figure 43.69 show the timing of operations in slave transmission.
The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 43.4.4, Initial Settings.
After initial settings, the RI3C will stay in the standby state until it receives a slave address that it matches.
(2) After receiving a matching slave address, the RI3C sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 1, the RI3C automatically places itself in slave transmit mode by setting both the ICMMR.TRS flag and the ICCSR.TDRE flag to 1.
(3) After the ICCSR.TDRE flag is confirmed to be 1, write the data for transmission to the ICDR register. At this time, if the RI3C receives no acknowledgment from the master device (receives an NACK signal) while the ICSER.NAKDE bit is 1, the RI3C suspends transfer of the next data.
(4) Wait until the ICSR2.TEND flag is set to 1 while the ICCSR.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDR register. When the ICSR2.NACKF flag or the TEND flag is 1, the RI3C drives the SCL00 line low on the falling edge of ninth SCL.
(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDR register to complete the processing. This releases the SCL00 line.
(6) Upon detecting the STOP condition, the RI3C automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2), flags ICCSR.TDRE and ICSR2.TEND, and the ICMMR.TRS flag to 0, and enters slave receive mode.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

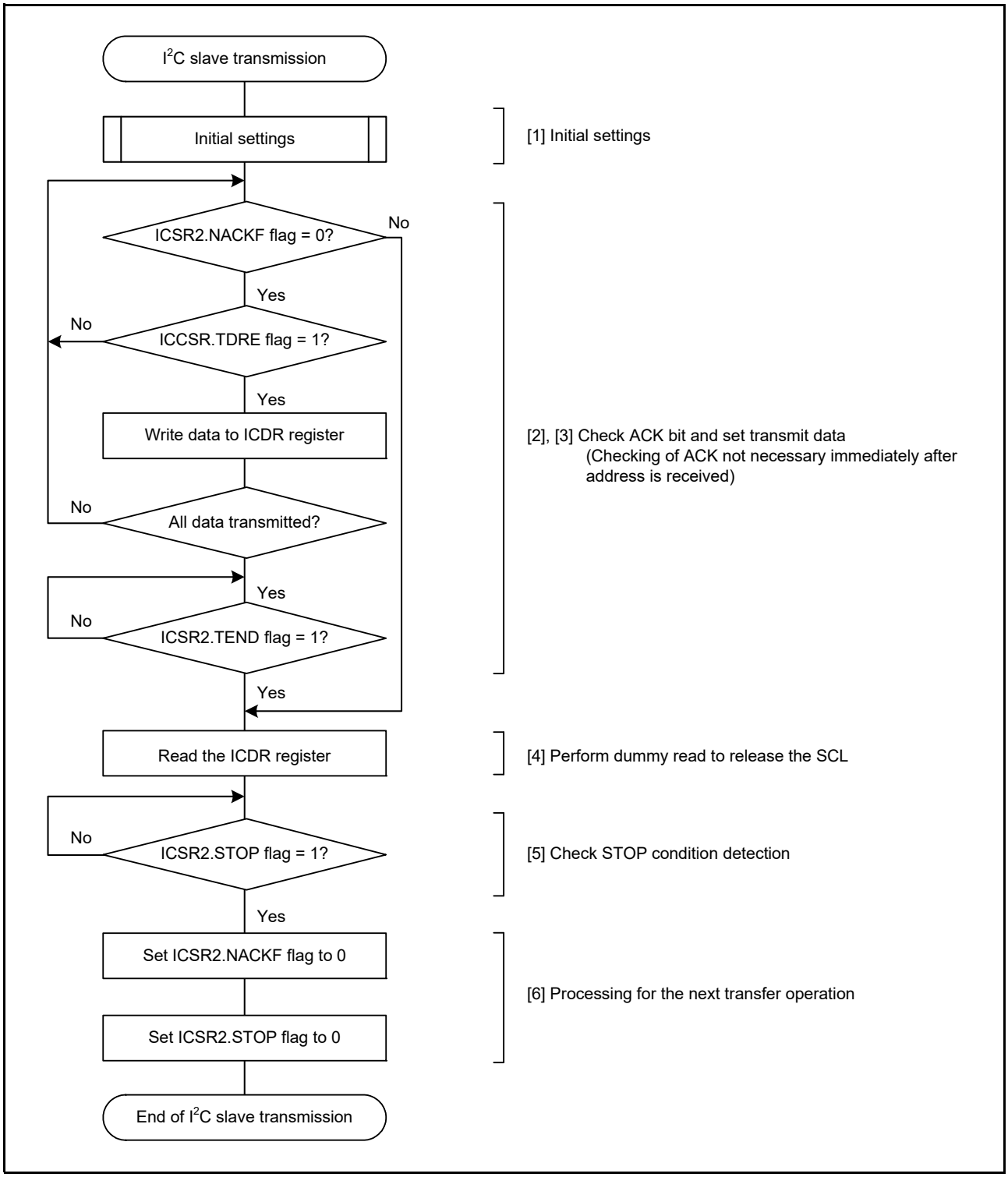


Figure 43.67 Example of I²C Slave Transmission Flowchart

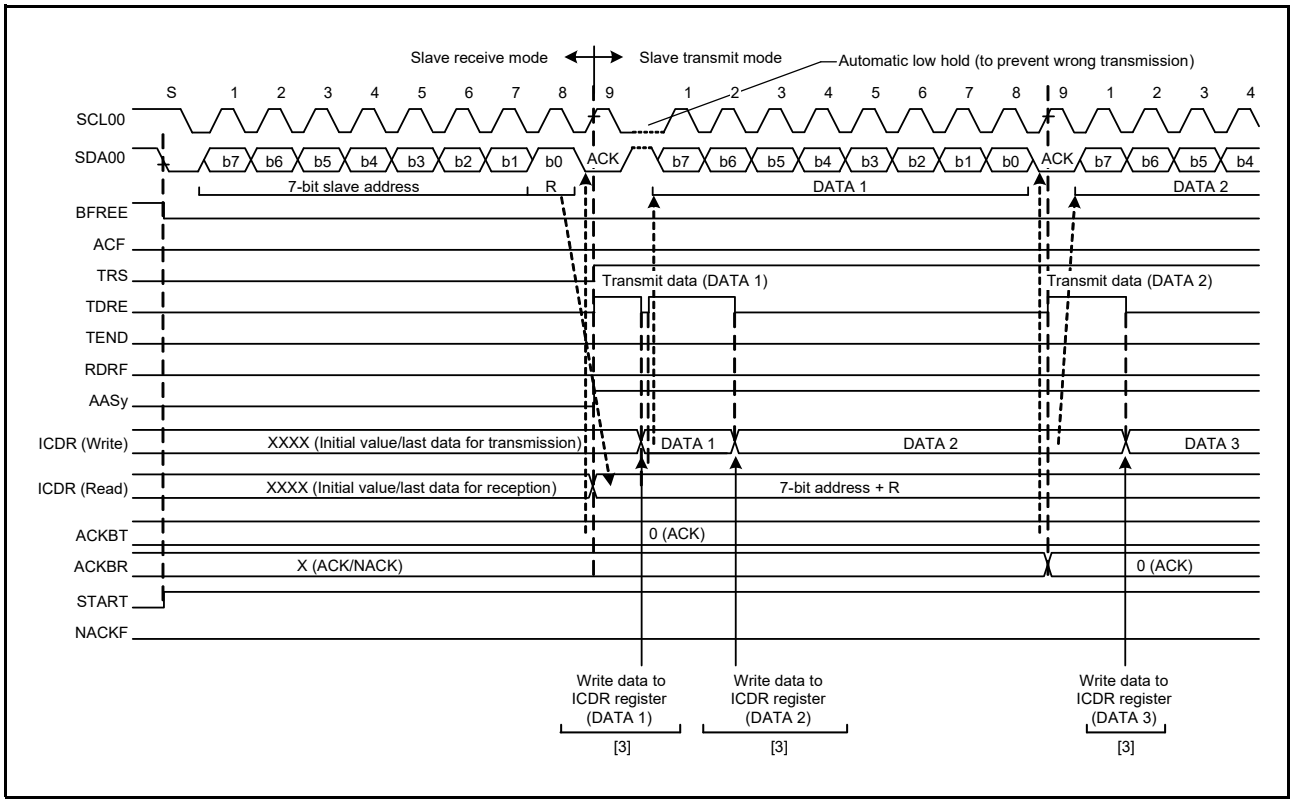


Figure 43.68 Slave Transmit Operation Timing (1) (7-Bit Address Format)

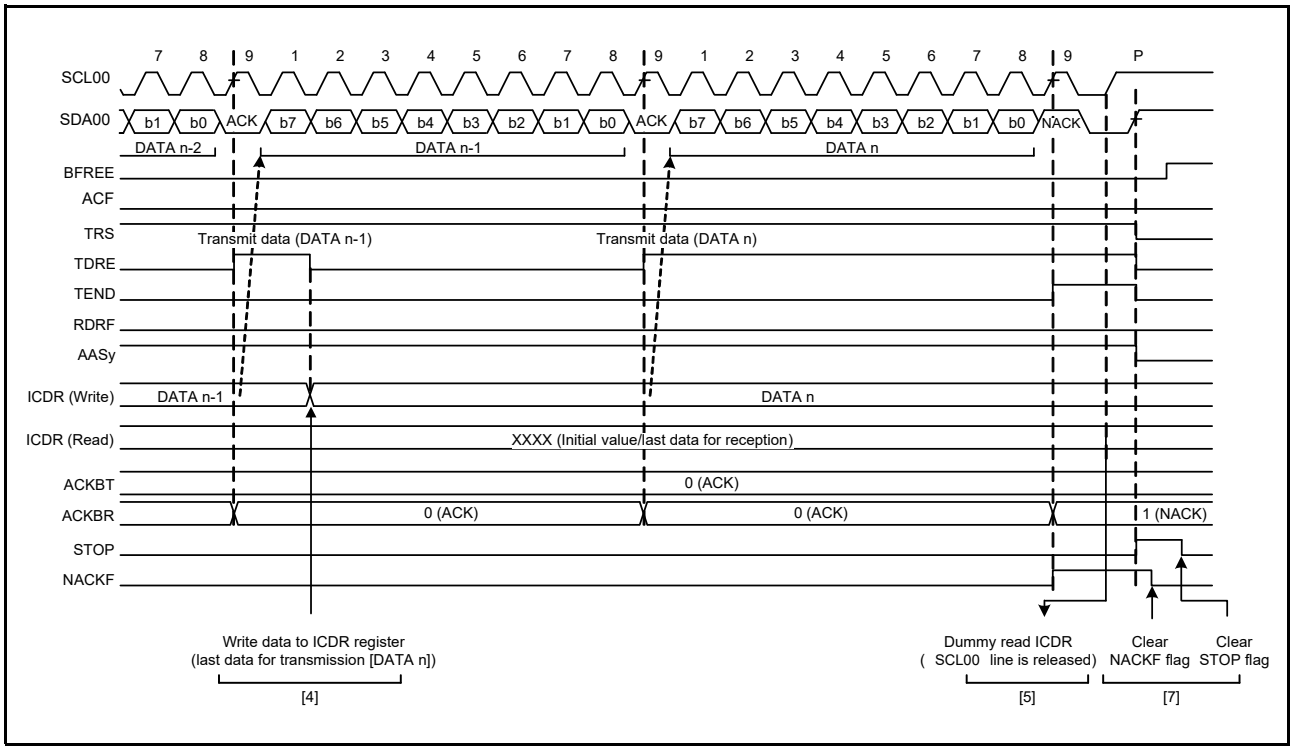


Figure 43.69 Slave Transmit Operation Timing (2)

43.4.9.2 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RI3C returns acknowledgments as a slave device.

Figure 43.70 shows an example of usage of slave reception and Figure 43.71 and Figure 43.72 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 43.4.4, Initial Settings. After initial settings, the RI3C will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RI3C sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 0, the RI3C continues to place itself in slave receive mode and sets the ICCSR.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICCSR.RDRF flag to be 1, dummy read ICDR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8-bits when the 10-bit address format is selected).

(4) When ICDR register is read, the RI3C automatically sets the ICCSR.RDRF flag to 0. If reading of ICDR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RI3C holds the SCL00 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDR register releases the SCL00 line from being held low.
When the ICSR2.STOP flag is 1 and the ICCSR.RDRF flag is also 1 or when all the data is completely received, read the ICDR register.

(5) Upon detecting the STOP condition, the RI3C automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

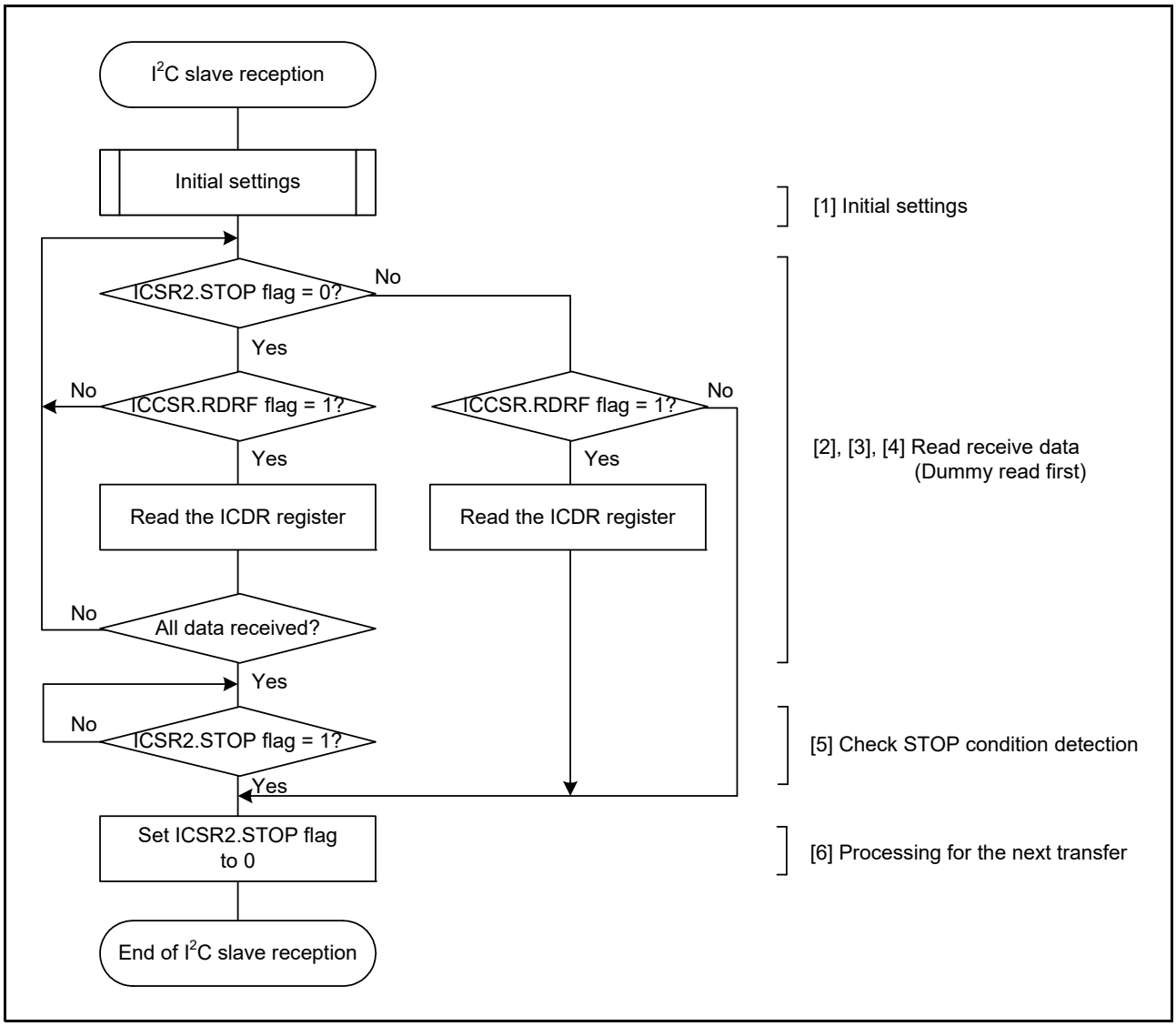


Figure 43.70 Example of I2C Slave Reception Flowchart

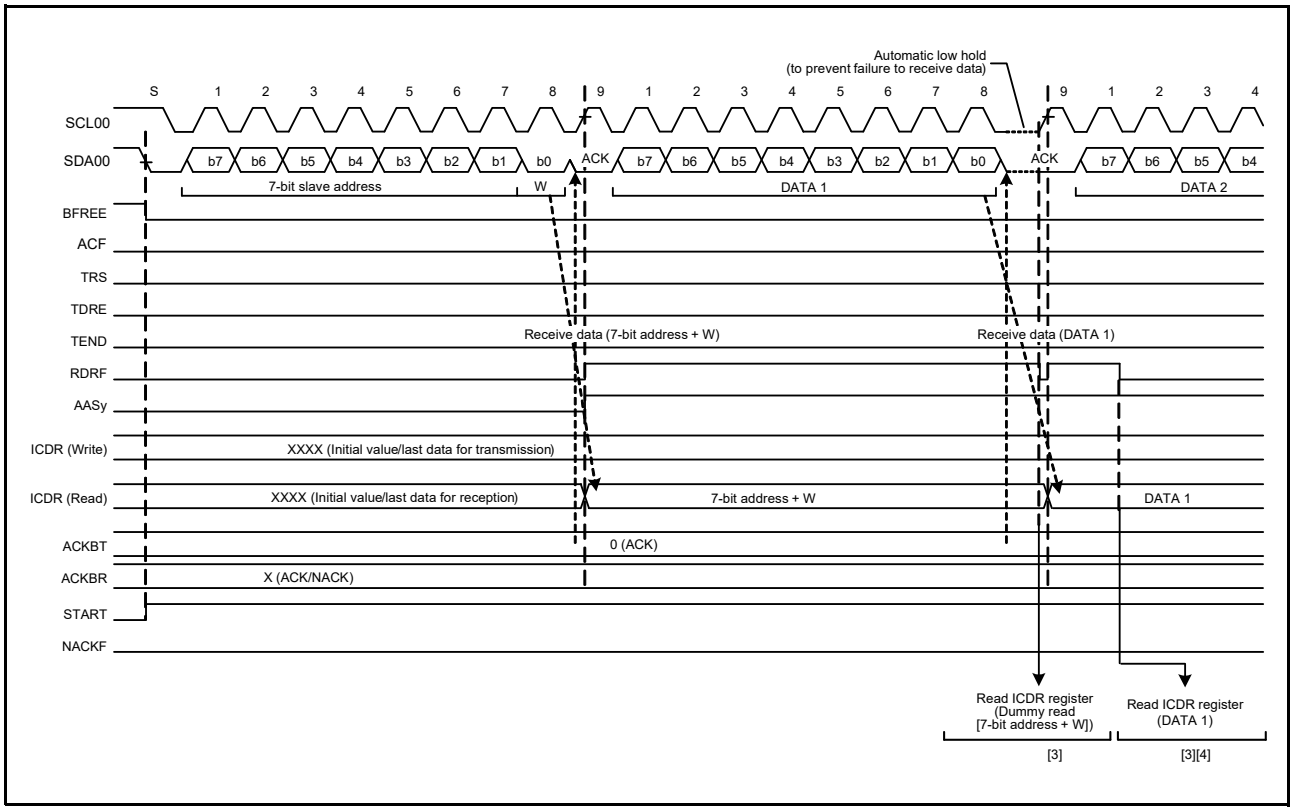


Figure 43.71 Slave Receive Operation Timing (1) (7-Bit Address Format, when WAITAE = 0)

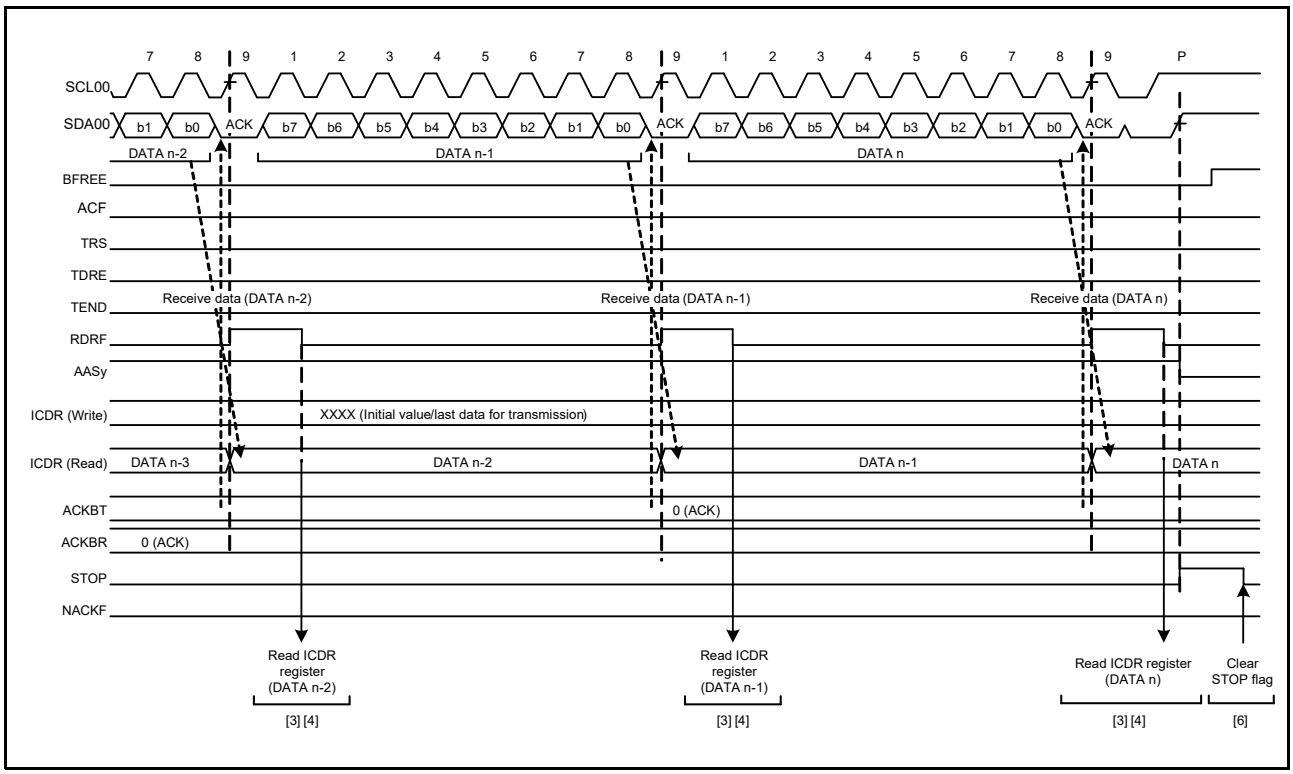


Figure 43.72 Slave Receive Operation Timing (2) (when WAITAE = 0)

43.5 Details of Function

43.5.1 SCL Synchronization Circuit (I2C Protocol Mode)

In generation of the SCL, the RI3C starts counting out the value for width at high level specified in the ICFBR.HIGH[7:0] bits when it detects a rising edge on the SCL00 line and drives the SCL00 line low once counting of the width at high level is complete. When the RI3C detects the falling edge of the SCL00 line, it starts counting out the width at low period specified in the ICFBR.LOW[7:0] bits, and then stops driving the SCL00 line (releases the line) once counting of the width at low level is complete. The SCL is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RI3C is equipped with a function (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL signals by monitoring the SCL00 line while in master mode.

When the RI3C has detected a rising edge on the SCL00 line and thus started counting out the width at high level specified in the ICFBR.HIGH[7:0] bits, and the level on the SCL00 line falls because an SCL signal is being generated by another master device, the RI3C stops counting when it detects the falling edge, drives the level on the SCL00 line low, and starts counting out the width at low level specified in the ICFBR.LOW[7:0] bits. When the RI3C finishes counting out the width at low level, it stops driving the SCL00 line to low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RI3C, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL00 line has been released. When the RI3C finishes outputting the low period of the SCL, the SCL00 line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

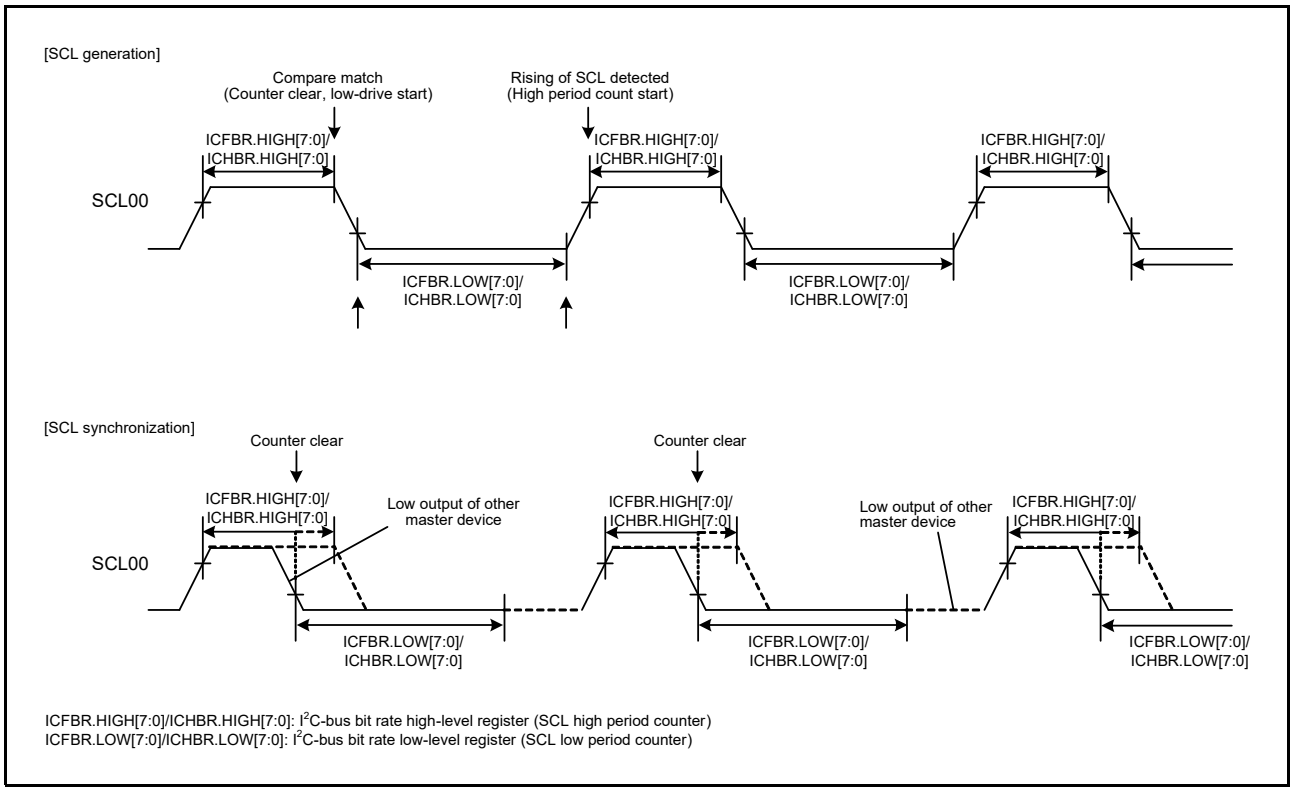


Figure 43.73 Generation and Synchronization of the SCL Signal

43.5.2 SDA Output Delay Function (I²C Protocol Mode)

The RI3C module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (generating of the START, repeated START, and STOP conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing from a falling edge of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) of data hold time specified by the SMBus specification.

The output delay facility is enabled by setting the ICOCR.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not 000b), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits with the count source selected by the ICOCR.DLCS bit (internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). When the counting of the delay cycles is completed, the RI3C output the SDA signal (generating of the START, repeated START, or STOP condition, data, or an ACK or NACK signal).

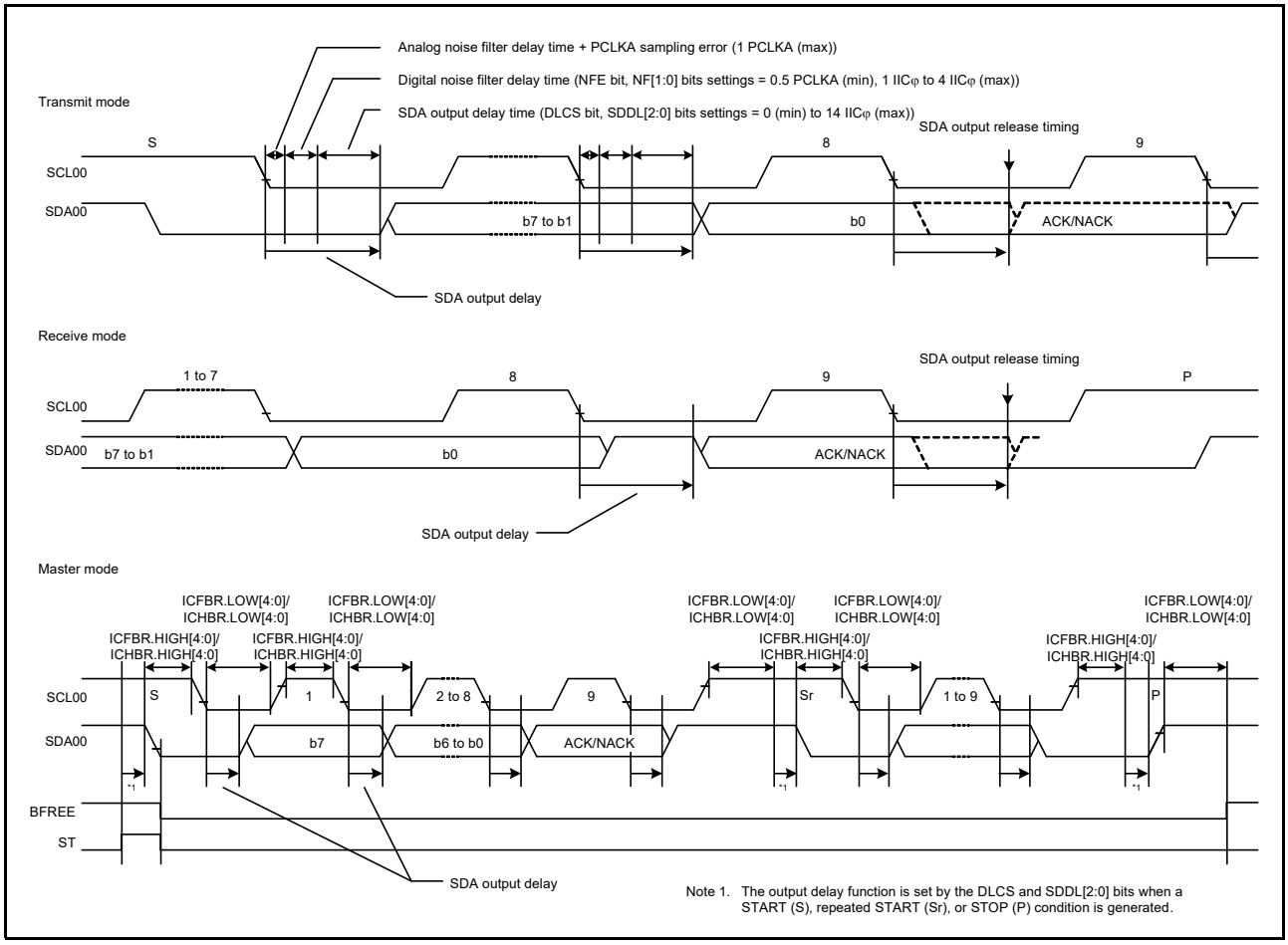


Figure 43.74 SDA Output Delay Function

43.5.3 Digital Noise Filters (I2C Protocol Mode)

The states of the SCL00 and SDA00 pins are conveyed to the internal circuitry through analog noise filter and digital noise filter. Figure 43.75 is a block diagram of the digital noise filter.

The on-chip digital noise filter of the RI3C consists of 16 flip-flop circuit stages connected in series and a match-detection circuit. When Hs-mode is selected, only the first four flip-flops are enabled.

The number of effective stages in the digital noise filter is selected by the ICICR.NF[3:0] bits (ICICR.NF[3:2] bits in Hs-mode). The selected number of effective stages determines the noise-filtering capability as a period from 1 to 16 IICφ cycles (one to four IICφ cycles in Hs-mode).

The input signal from the SCL00 pin (also the SDA00 pin) is sampled on rising edges of the IICφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the ICICR.NF[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (I3CCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with I3CCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, disable the digital noise filter (by setting the ICICR.NFE bit to 0) and use only the analog noise filter.

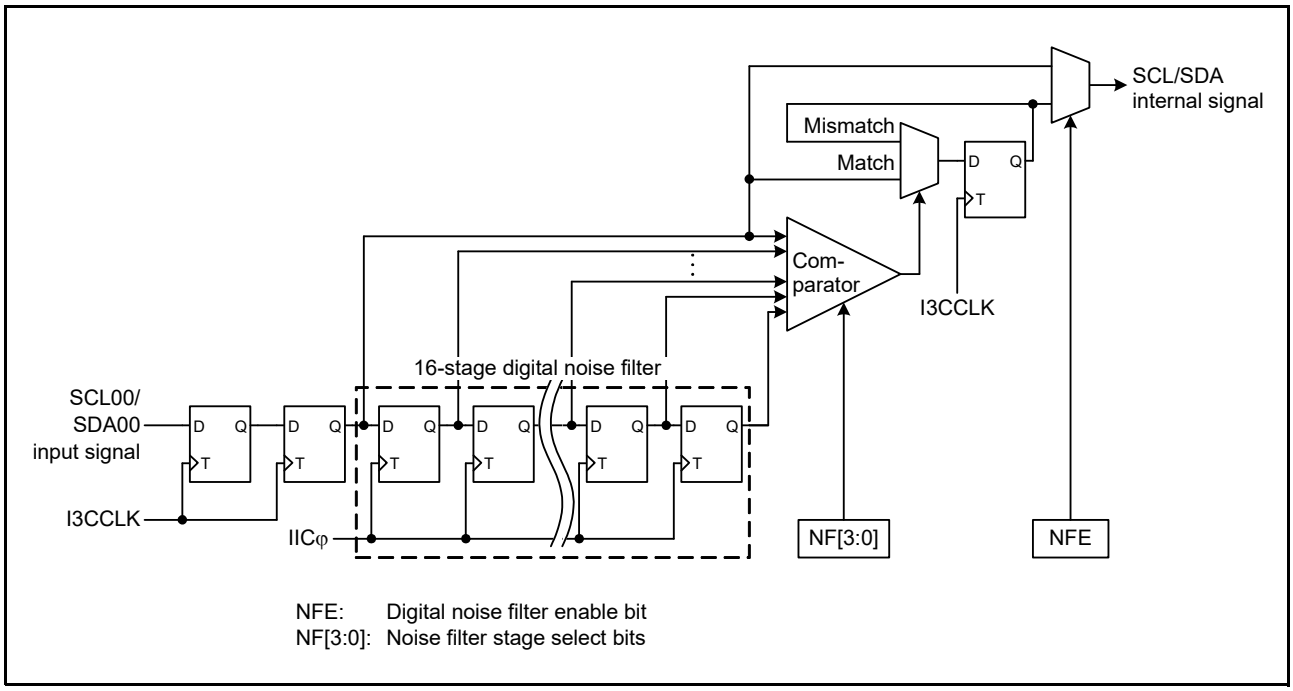


Figure 43.75 Block Diagram of Digital Noise Filter

43.5.4 Address Match Detection (I2C Protocol Mode)

The RI3C can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

43.5.4.1 Slave-Address Match Detection

The RI3C can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSCR.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in SAMRy register (y = 0 to 2) can be detected.

When the RI3C detects a match of the set slave address, the corresponding ICSSR.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL, and the ICCSR.RDRF flag or the ICCSR.TDRE flag is set to 1 according to the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 43.76 to Figure 43.78 show the AASy flag set timing in three cases.

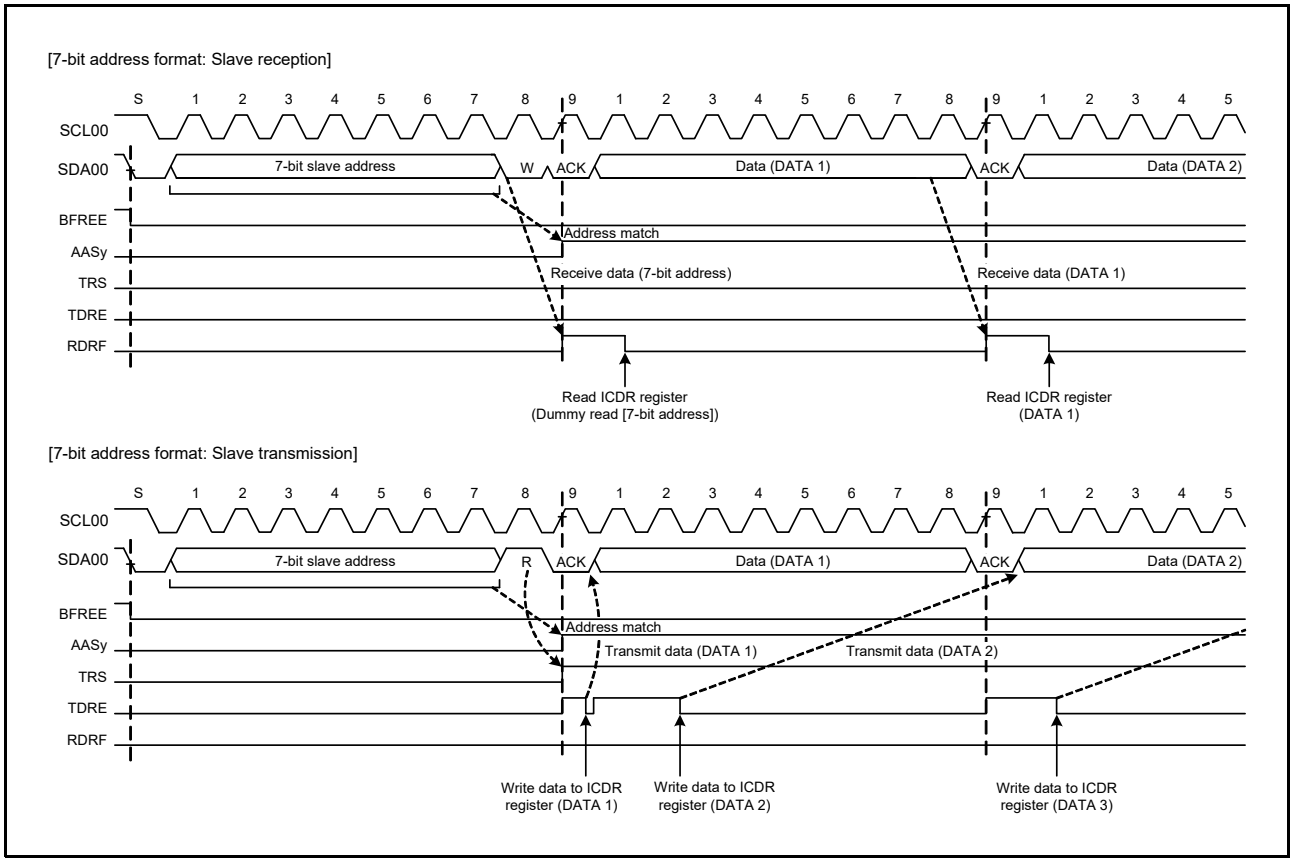


Figure 43.76 AASy Flag Set Timing with 7-Bit Address Format Selected

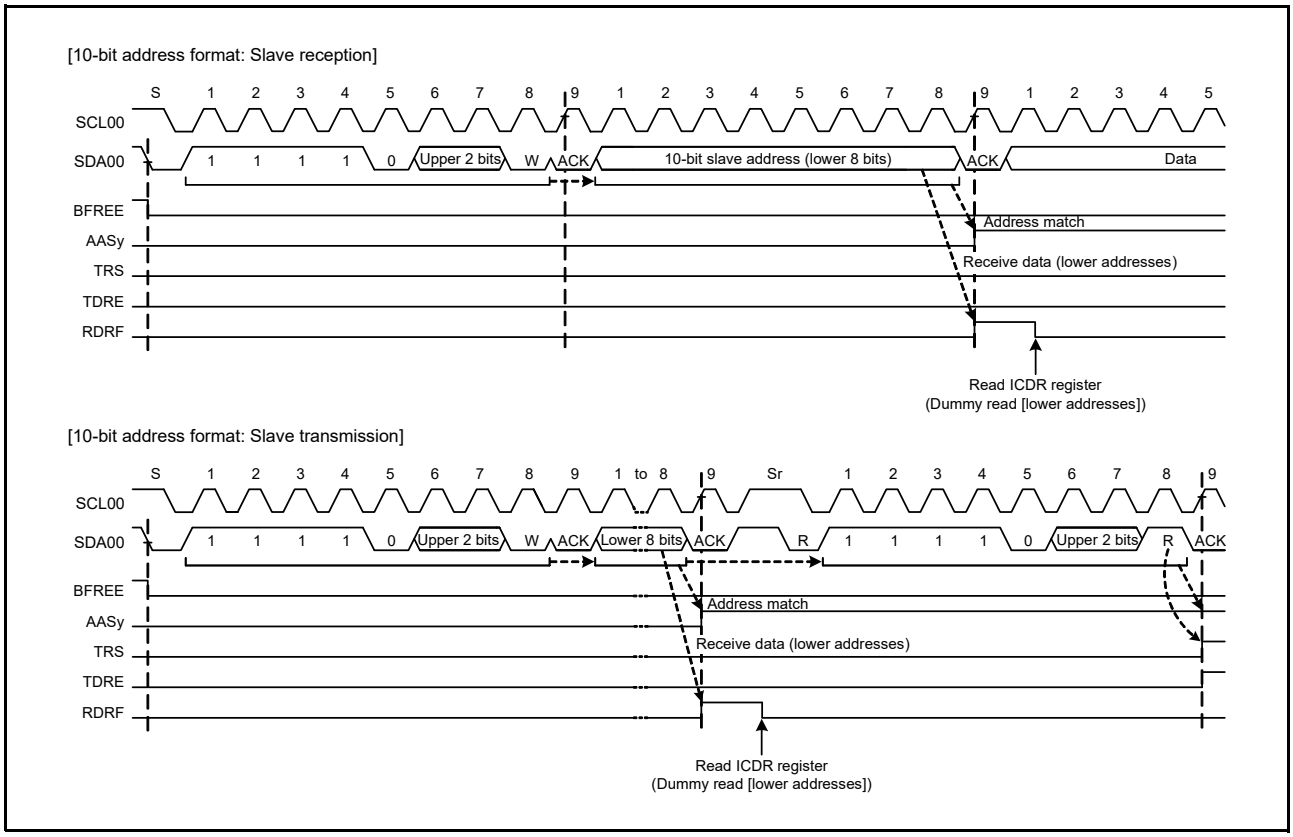


Figure 43.77 AASy Flag Set Timing with 10-Bit Address Format Selected

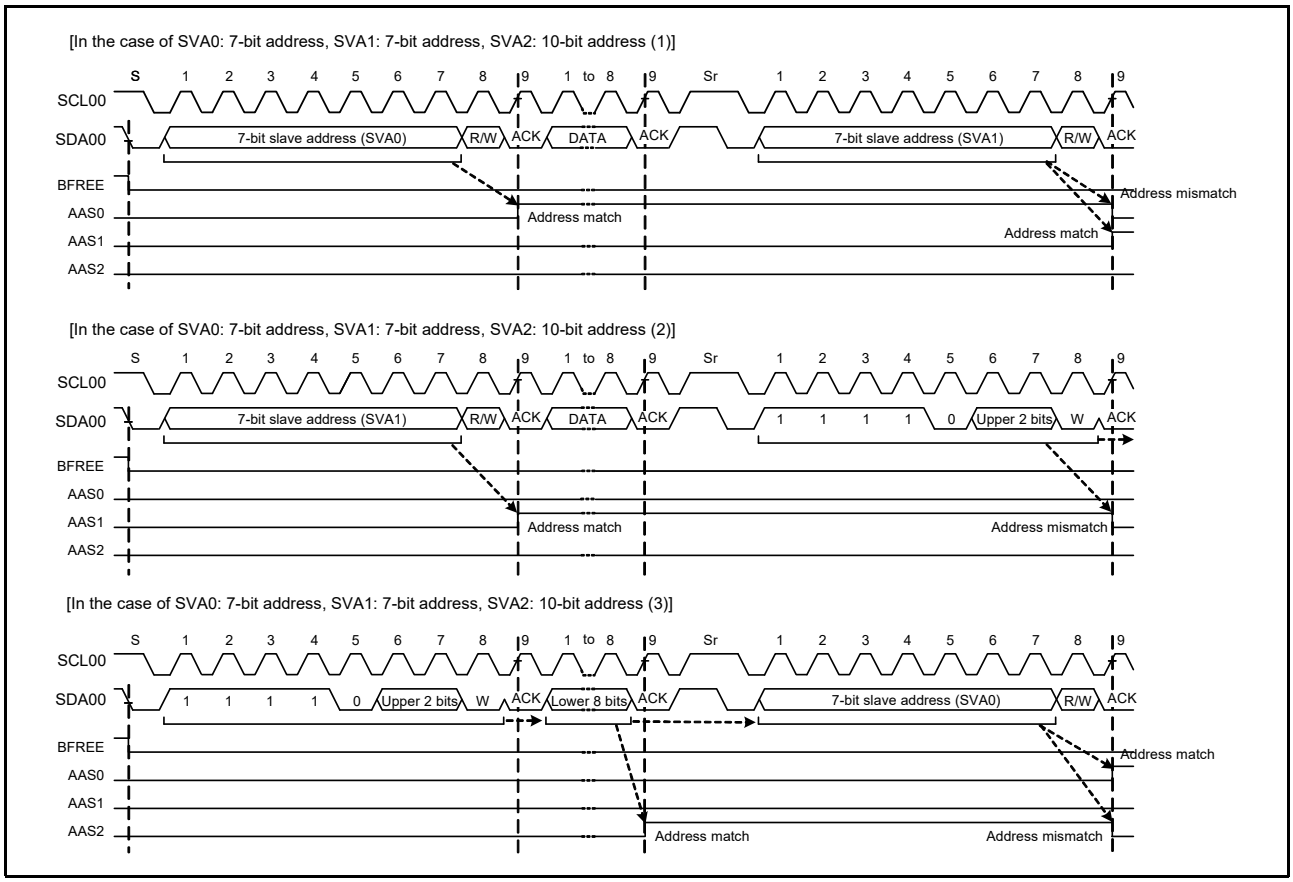


Figure 43.78 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

43.5.4.2 General Call Address Detection

The RI3C has a function for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSCR.GCAE bit to 1.

If the address following a START or repeated START condition is 0000 000b + 1 (read) (start byte), the RI3C recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RI3C detects the general call address, both the ICSSR.GCA flag and the ICCSR.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

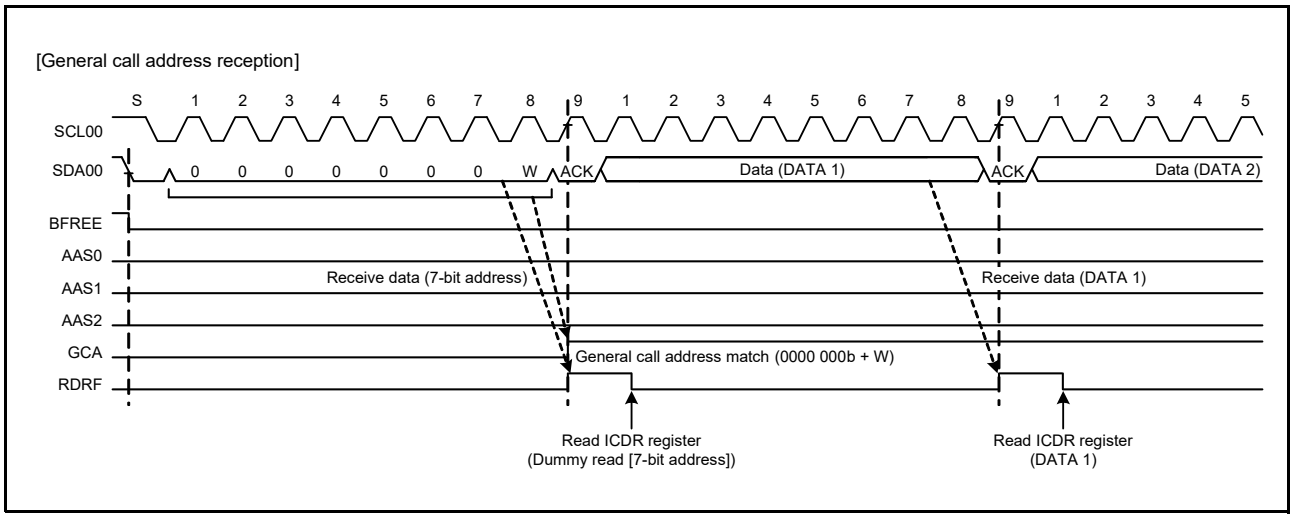


Figure 43.79 Timing of GCA Flag Setting during Reception of General Call Address

43.5.4.3 Device-ID Address Detection

The RI3C module has a function for detecting device-ID addresses conformant with the I^2C-bus specification.
When the RI3C receives 1111 100b as the first seven bits of the first byte following a START condition or repeated START condition while the ICSCR.DIDE bit is 1, the RI3C recognizes the address as a device ID address, sets the ICSSR.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RI3C sets the corresponding ICSSR.AASy flag (y = 0 to 2) to 1.
After that, when the first byte received after a START or repeated START condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RI3C does not compare the second and subsequent bytes and sets the ICCSR.TDRE flag to 1.
In the device-ID address detection function, the RI3C sets the DID flag to 0 if a match with the RI3C's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RI3C's own slave address and the detection of a repeated START condition. If the first byte after detection of a START or repeated START condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RI3C sets the DID flag to 1 and compares the second and subsequent bytes with the RI3C's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RI3C does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that ICCSR.TDRE flag is 1.
Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

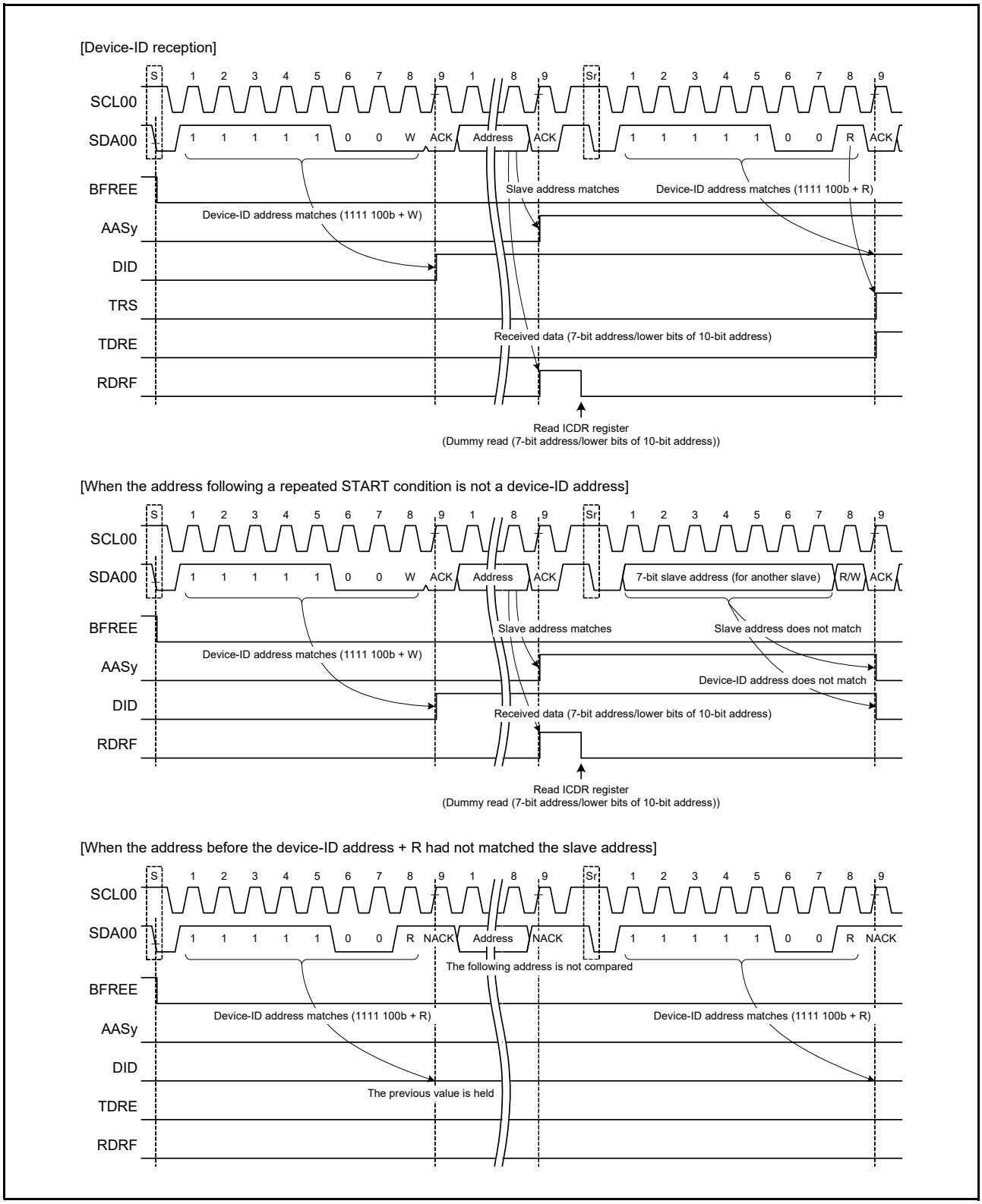


Figure 43.80 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

43.5.4.4 Hs-Mode Master Code Detection

The RI3C has a function to detect the Hs-mode master code (0000 1XXXb).
When the RI3C receives 0000 1XXXb as the first byte following a START condition while the ICSCR.HSMCE bit is 1, the RI3C recognizes it as Hs-mode master code and set the ICSSR.HSMC flag to 1 at the rising edge of the ninth SCL. The RI3C does not acknowledge the master code regardless of the value of the ICACKR.ACKBT bit (the SDA00 pin remains high).
The first byte following a repeated START condition after the NACK bit is recognized as the slave address and compared with the SAMRy.SVA[9:0] bits (y = 0 to 2). When they match, the corresponding ICSSR.AASy flag is set to 1 at the rising edge of the ninth SCL. Also, the ICCSR.RDRF or TDRE flag is set to 1 depending on the value of the R/W# bit. When a STOP condition is detected, the ICSSR.HSMC flag is set to 0.

Note: If the RI3C receives Hs-mode master code while the ICSCR.HSMCE bit is 0, it ignores all data patterns until a STOP condition is detected.

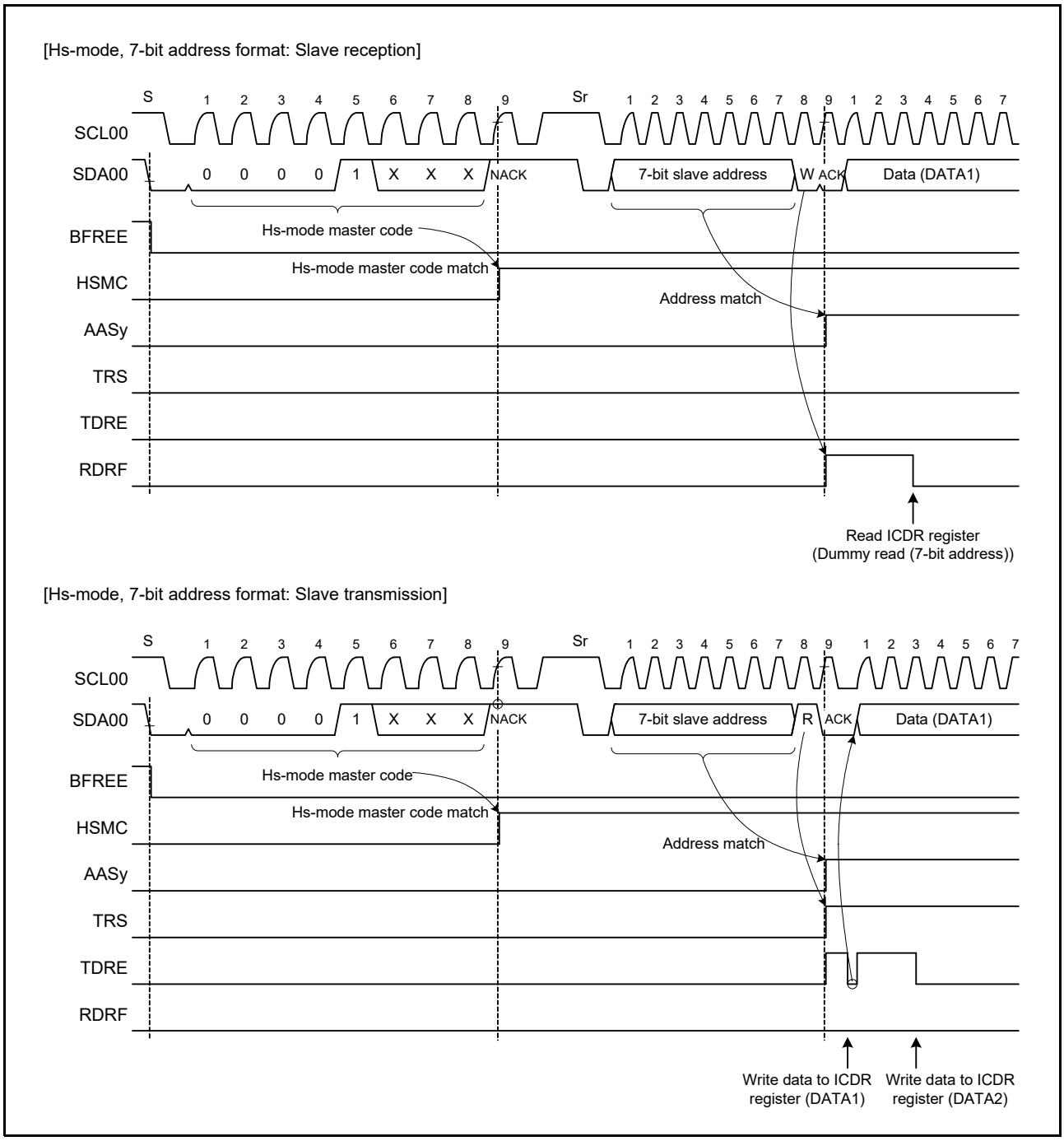


Figure 43.81 Set Timing of the HSMC and AASy Flags during Reception of Hs-Mode Master Code

43.5.4.5 Host Address Detection

The RI3C has a function to detect the host address while the SMBus is operating. When the ICSCR.HOAE bit is set to 1 while the ICFER.SMBS bit is 1, the RI3C can detect the host address (0001 000b) in slave receive mode (ICMMR.ACF and ICMMR.TRS flags = 00b).

When the RI3C detects the host address, the ICSSR.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICCSR.RDRF flag is set to 1 when the R/W# bit is 0 (write). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an R bit (R/W# bit = 1), the RI3C can also detect the host address. After the host address is detected, the RI3C operates in the same manner as normal slave operation.

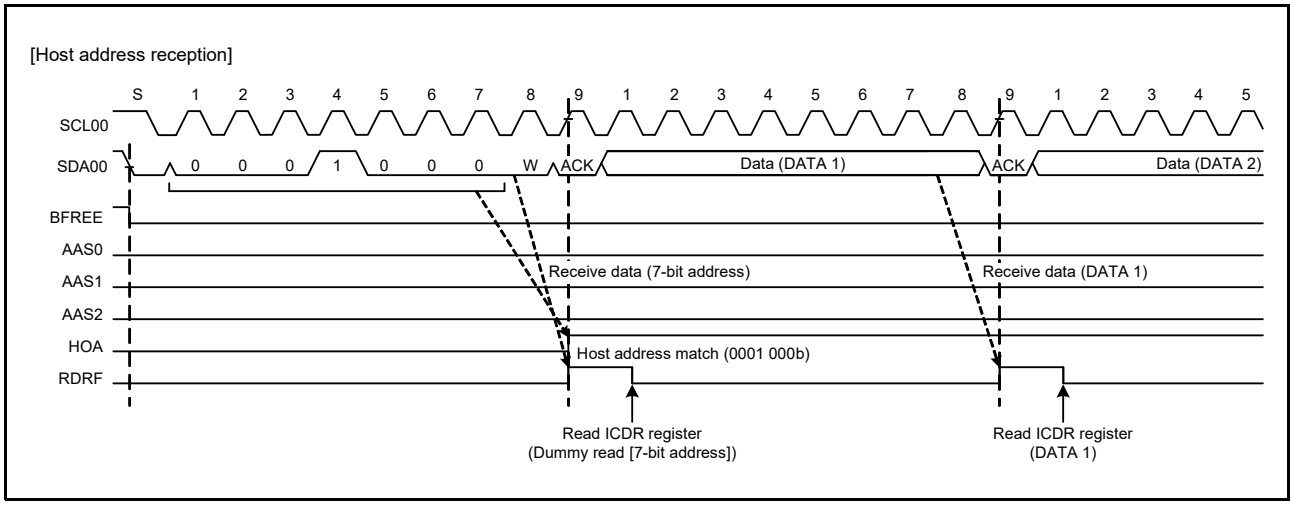


Figure 43.82 HOA Flag Set Timing during Reception of Host Address

43.5.5 Automatic Low-Hold Function for SCL (I^2C Protocol Mode)

43.5.5.1 Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the transmit data register (ICDR register) with the RI3C in transmission mode (ICMMR.TRS flag = 1), the SCL00 line is automatically held low over the periods shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a START condition or repeated START condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

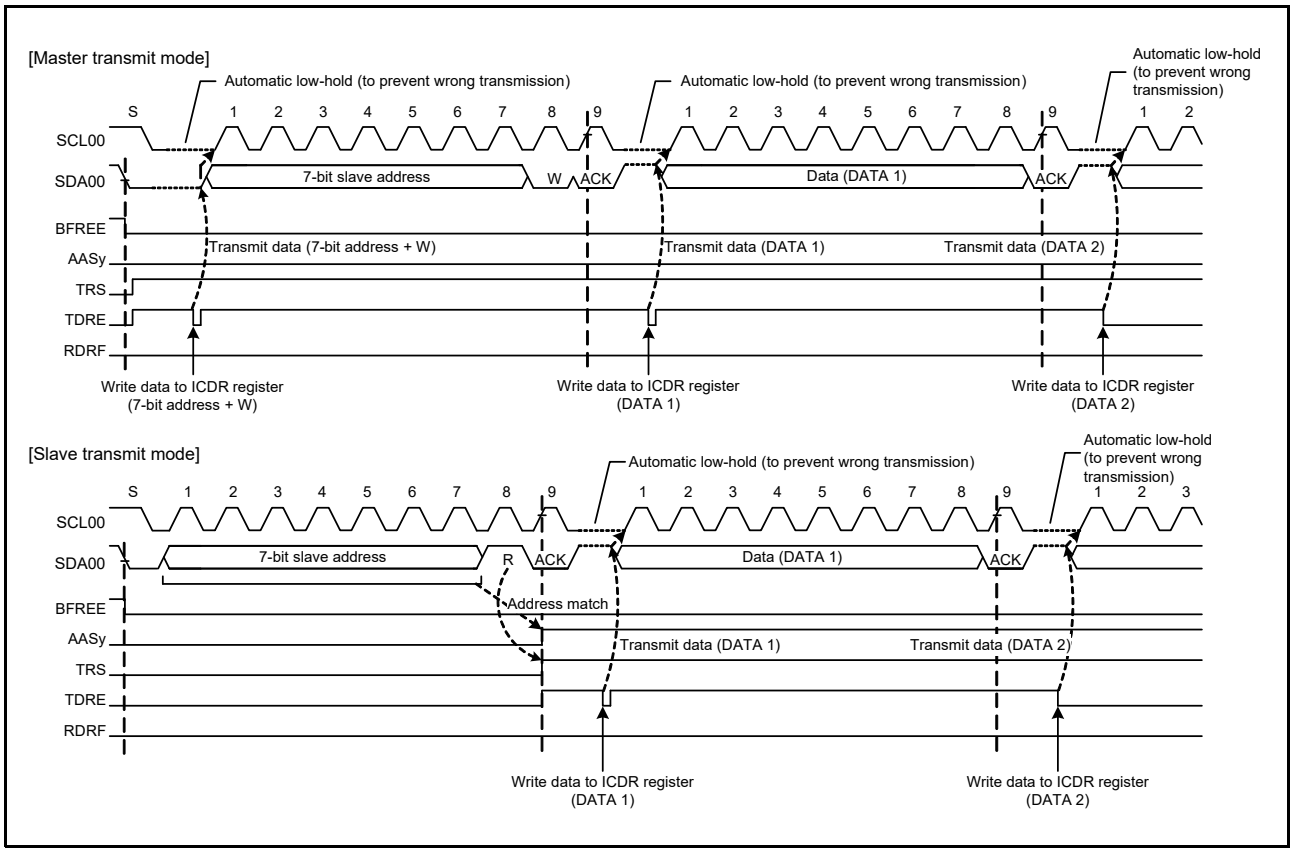


Figure 43.83 Automatic Low-Hold Operation in Transmit Mode

43.5.5.2 NACK Reception Transfer Suspension Function

The RI3C has a function to suspend transfer operation when NACK is received in transmit mode (ICMMR.TRS flag = 1).

This function is enabled when the ICSER.NAKDE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICCSR.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDA00 line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (ICSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, issue a repeated START or STOP condition, set the NACKF flag to 0, and then issue a START condition again.

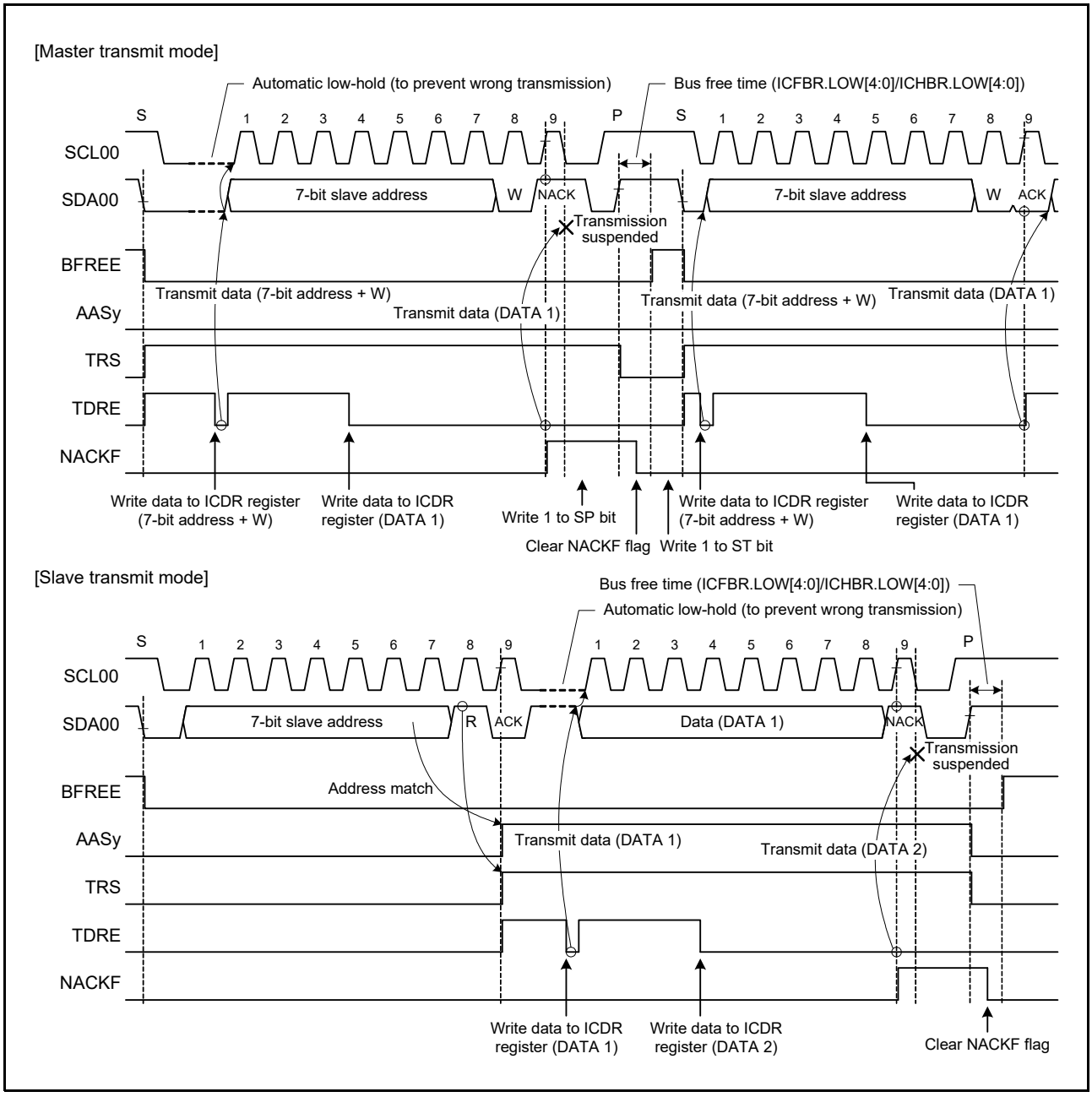


Figure 43.84 Suspension of Data Transmission When NACK is Received (NAKDE = 1)

43.5.5.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDR register) read is delayed for a period of one transfer frame or more with receive data full (ICCSR.RDRF flag = 1) in receive mode (ICMMR.TRS flag = 0), the RI3C holds the SCL00 line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RI3C's own slave address or a slave address for the other device is designated after a STOP condition is generated.
Sections in which the SCL00 line is held low can be selected with a combination of the ICCSCR.WAITRE and ICCSCR.WAITAE bits.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAITRE Bit

When the ICCSCR.WAITRE bit is set to 1, the RI3C performs 1-byte receive operation using the WAITRE bit function. Furthermore, when the ICCSCR.WAITAE bit is 0, the RI3C automatically sends the ICACKR.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCL00 line low at the falling edge of the ninth SCL using the WAITRE bit function. This low-hold is released by reading data from ICDR register, which enables bytewise receive operation.
The WAITRE bit function is enabled for receive frames after a match with the RI3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the WAITAE Bit

When the ICCSCR.WAITAE bit is set to 1, the RI3C performs 1-byte receive operation using the WAITAE bit function. When the WAITAE bit is set to 1, the ICCSR.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCL00 line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICACKR.ACKBT bit, but cannot be released by reading data from ICDR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The WAITAE bit function is enabled for receive frames after a match with the RI3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

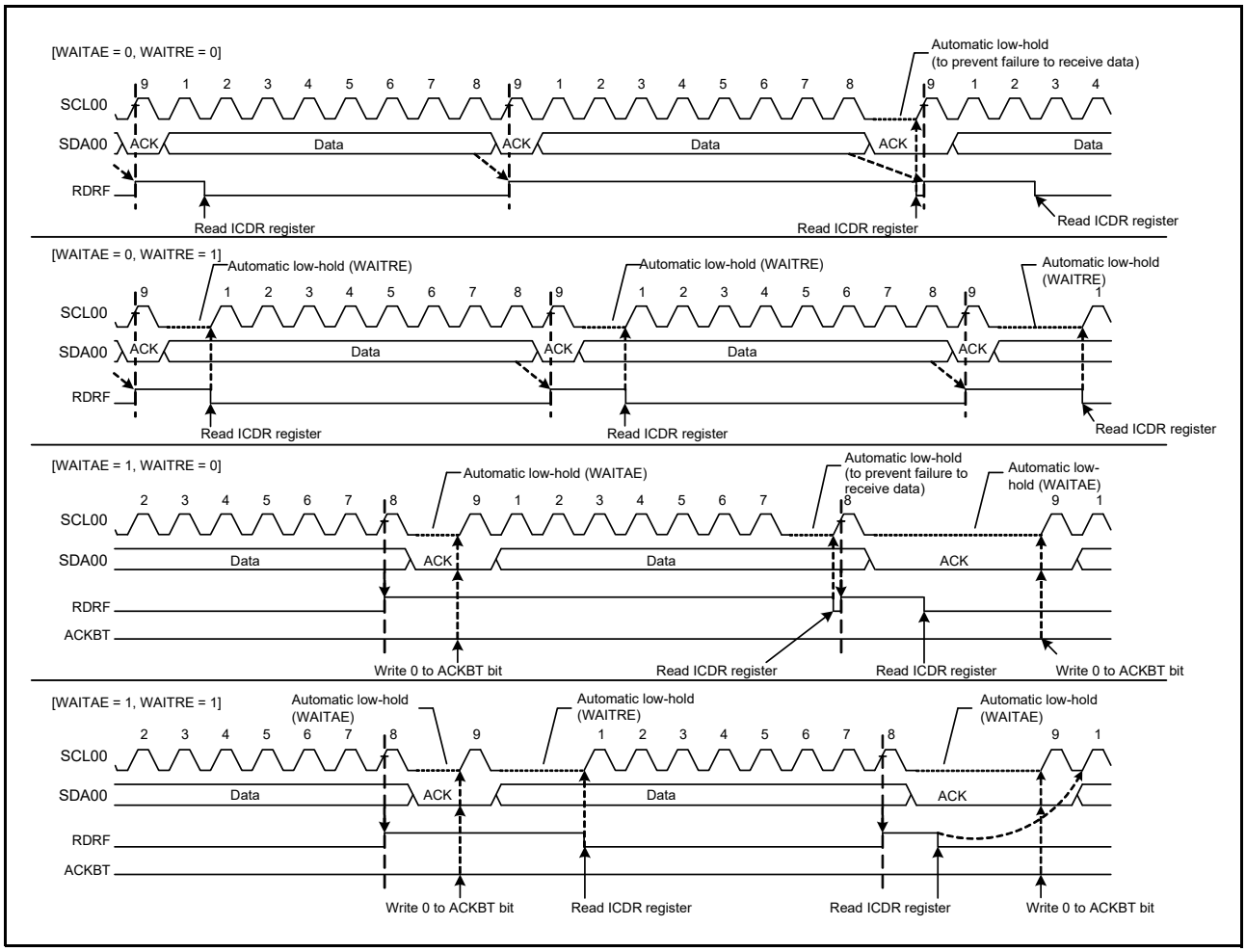


Figure 43.85 Automatic Low-Hold Operation in Receive Mode (Using WAITAE and WAITRE Bits)

43.5.6 Arbitration-Lost Detection Functions (I^2C Protocol Mode)

In addition to the normal arbitration-lost detection function specified in the I^2C-bus specification, the RI3C also has functions to prevent double generation of START conditions, to detect arbitration-lost during NACK transmission, and to detect arbitration-lost during slave transmission.

43.5.6.1 Master Arbitration-Lost Detection Function (MALE Bit)

The RI3C drives the SDA00 line low to generate a START condition. However, if the SDA00 line is already driven low by another master device that generates a START condition, the RI3C generates an arbitration-lost and gives a priority to the transfer by the other master device. Similarly, if the ICCGR.ST bit is set to 1 while the ICBSR.BFREE flag is 0 (bus busy), the RI3C generates an arbitration-lost and gives a priority to the transfer by the other master device. No START condition is generated in this case.
After a START condition is generated successfully, if the level of the transmit data including the address bits (i.e. the internal SDA output level) and the level on the SDA00 line do not match (the internal SDA output is high (the SDA00 pin is high-impedance) but low is detected on the SDA00 line), the RI3C generates an arbitration-lost.
If a master arbitration-lost is generated, the RI3C immediately transitions to slave receive mode. If a slave address including the general call address matches its own address at this time, the RI3C continues the slave operation.
The RI3C detects master arbitration-lost when one of the following conditions is met while the ICSER.ALE and ICFER.MALE bits are 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- When the internal SDA output level and the level on the SDA00 line do not match (START condition generation error) when a START condition is generated by setting the ICCGR.ST bit to 1 while the ICBSR.BFREE flag is 1.
- When the ICCGR.ST bit is set to 1 while the BFREE flag is 0 (START condition double generation error).
- When the transmit data excluding acknowledgment bit (internal SDA signal output level) and the level on the SDA00 line do not match in master transmit mode (the ICMMR.ACF and ICMMR.TRS flags are 1).

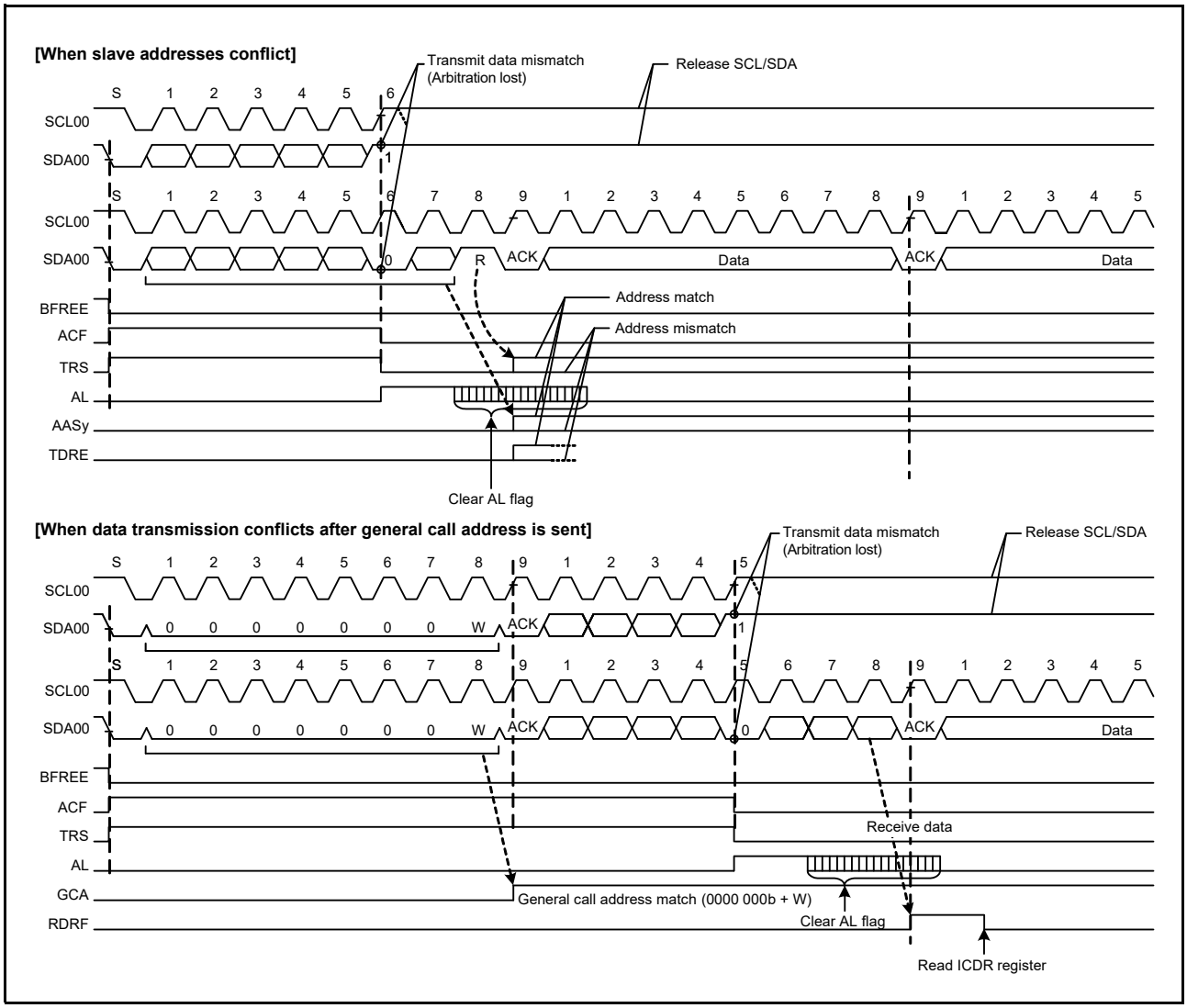


Figure 43.86 Examples of Master Arbitration-Lost Detection (MALE = 1)

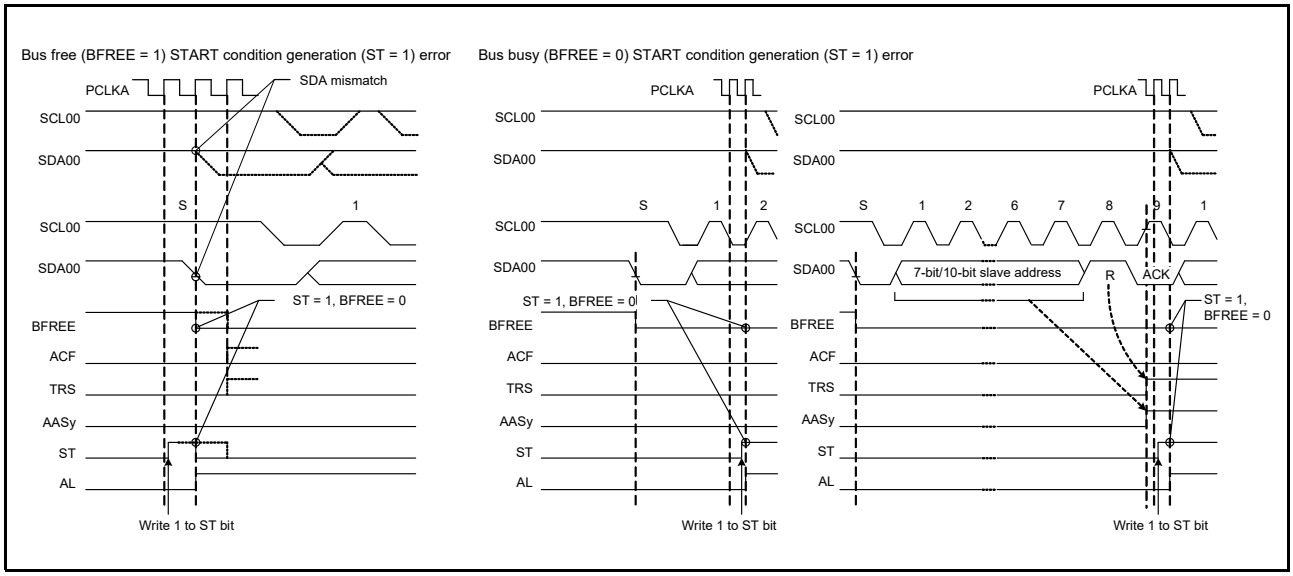


Figure 43.87 Arbitration-Lost Detection When a START Condition is Generated (MALE = 1)

43.5.6.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RI3C has a function to generate an arbitration-lost if the internal SDA output level and the level on the SDA00 line do not match (the internal SDA output is high (the SDA00 pin is high-impedance) but low is detected on the SDA00 line) during NACK transmission in receive mode. NACK transmission arbitration-lost is generated by a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflicts can occur when multiple master devices exchange common information through a single slave device.

Figure 43.88 shows an example of NACK transmission arbitration-lost detection when two master devices (master A and master B) and one slave device are connected to the bus.

In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device. If master A and master B access the slave device at the same time, arbitration-lost does not occur on either master A or master B when accessing the slave device because the slave addresses are the same. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends a NACK when it has received second byte, the last byte of data, from the slave device. On the other hand, master B sends an ACK because it has not yet received four bytes of data from the slave device. At this time, a conflict occurs between the NACK of master A and the ACK of master B.

If master A cannot detect the ACK sent by master B when such a conflict occurs, master A generates a STOP condition. This conflicts with the SCL output of master B and interferes with communication.

The RI3C can generate an arbitration-lost if it receives an ACK while sending an NACK. This prevents the generation of a STOP condition and prevents bus communication interference. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after the NACK transmission when the UDID (unique device identification) of the “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RI3C detects NACK transmission arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.NALE bits are 1 (NACK transmission arbitration-lost detection is enabled).

If the NACK transmission arbitration-lost occurs, the RI3C cancels the slave address matched state and enters slave receive mode.

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDA00 line (ACK is received) during transmission of NACK (the ICACKR.ACKBT bit is 1)

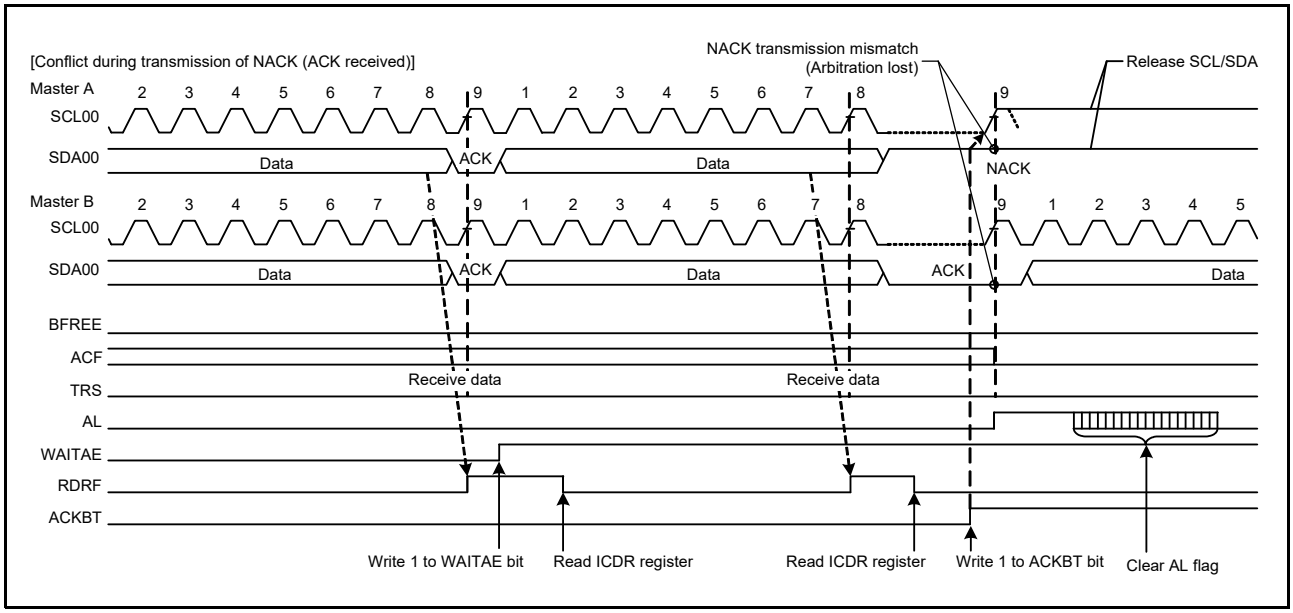


Figure 43.88 Example of NACK Transmission Arbitration-Lost Detection (NALE = 1)

43.5.6.3 Slave Arbitration-Lost Detection Function (SALE Bit)

The RI3C has a function to generate an arbitration-lost if the transmit data (i.e. the internal SDA output level) and the level on the SDA00 line do not match (the internal SDA output is high (the SDA00 pin is high-impedance) but low is detected on the SDA00 line) in slave transmit mode.
The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identification) of SMBus. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.
The RI3C detects slave arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.SALE bits are 1 (slave arbitration-lost detection enabled).
If the slave arbitration-lost occurs, the RI3C cancels the slave address matched state and enters slave receive mode.

Condition for detecting slave arbitration-lost

- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDA00 line in slave transmit mode (the ICMMR.ACF flag is 0 and the ICMMR.TRS flag is 1)

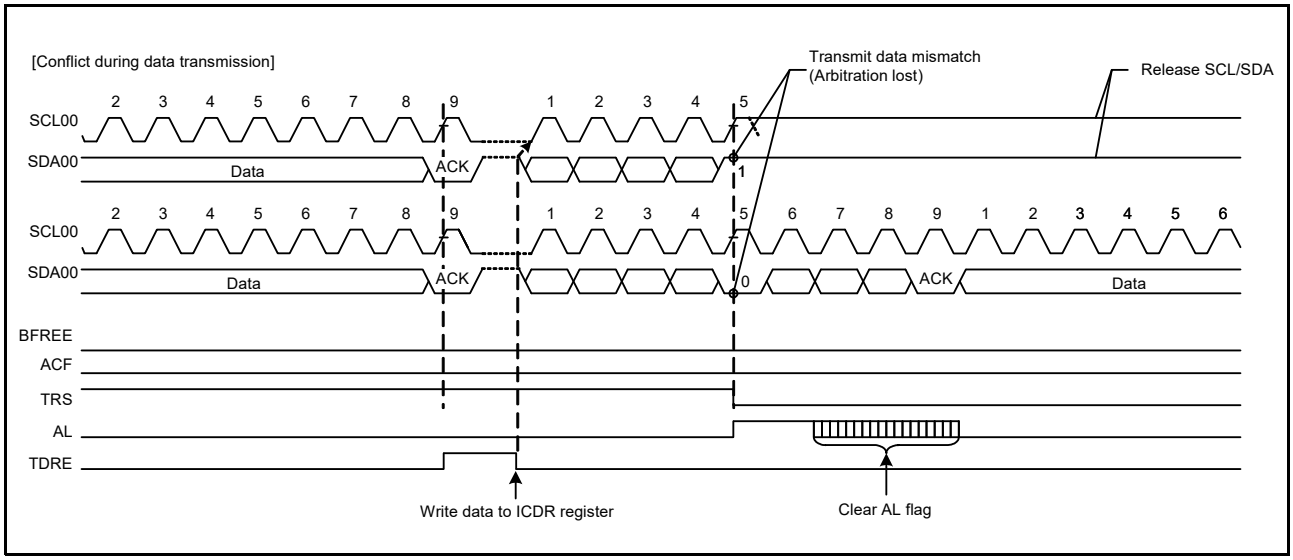


Figure 43.89 Example of Slave Arbitration-Lost Detection (SALE = 1)

43.5.7 START Condition/Repeated START Condition/STOP Condition Generating Function (I^2C Protocol Mode)

43.5.7.1 Generating a START Condition

The RI3C generates a START condition when the ICCGR.ST bit is set to 1.
When the ST bit is set to 1 (requests to generate a START condition) while the ICBSR.BFREE flag is set to 1 (bus free state), the RI3C generates a START condition.
When a START condition is generated normally, the RI3C automatically shifts to the master transmit mode.
A START condition is generated in the following sequence.

START condition generation

(1) Drive the SDA00 line low (high to low).
(2) Secure the START condition hold time with the time set in the ICFBR.HIGH[7:0] bits.
(3) Drive the SCL00 line low (high to low).
(4) Detect a low of the SCL00 line and secure the low period of SCL00 line with the time set in the ICFBR.LOW[7:0] bits.

43.5.7.2 Generating a Repeated START Condition

The RI3C generates a repeated START condition when the ICCGR.RS bit is set to 1.
When the RS bit is set to 1, generation of a repeated START condition is requested. When the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.ACF flag is 1 (master mode), the RI3C generates a repeated START condition.
A repeated START condition is generated in the following sequence.

Repeated START condition generation

(1) Release the SDA00 line.
(2) Secure the low period of SCL00 line with the time set in the ICFBR.LOW[7:0] bits.
(3) Release the SCL00 line (low to high).
(4) Detect a high of the SCL00 line and secure the repeated START condition setup time with the time set in the ICFBR.LOW[7:0] bits.
(5) Drive the SDA00 line low (high to low).
(6) Secure the repeated START condition hold time with the time set in the ICFBR.HIGH[7:0] bits.
(7) Drive the SCL00 line low (high to low).
(8) Detect a low of the SCL00 line and secure the low period of SCL00 line with the time set in the ICFBR.LOW[7:0] bits.

Note: When requesting generation of a repeated START condition, write the slave address to the ICDR register after confirming that the ICCGR.RS bit is 0. Data written while the ICCGR.RS bit is 1 is not transferred because it is considered to be before the repeated START condition is generated.

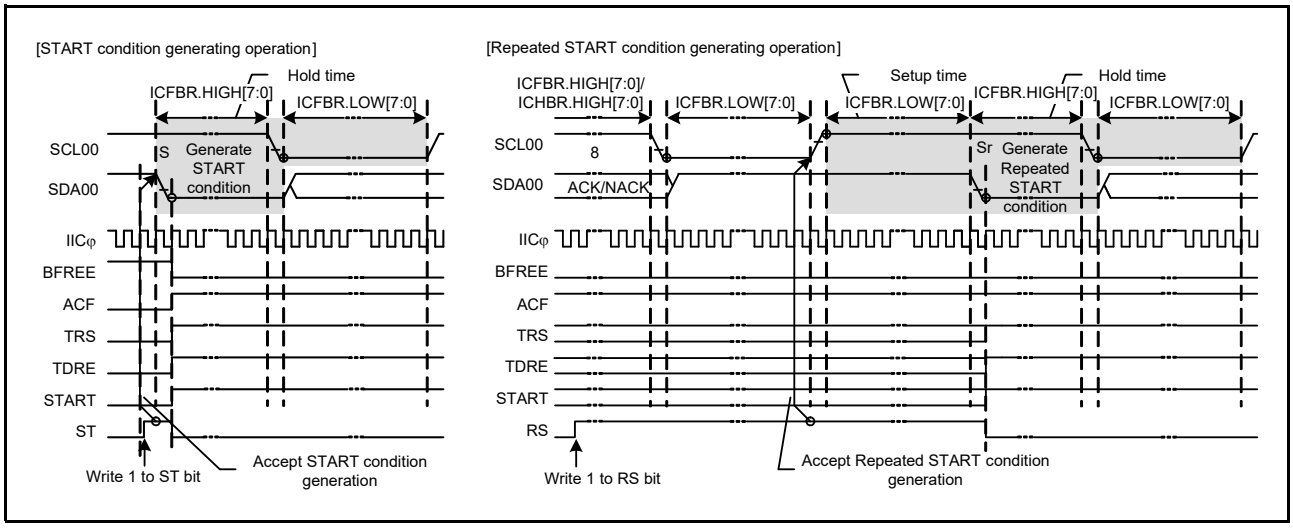


Figure 43.90 START Condition/Repeated START Condition Generating Timing (ST and RS Bits)

Figure 43.91 shows the operation timing when generating a repeated START condition after the master transmission.

Repeated START condition generation after the master transmission

(1) Initial setting. For details, refer to section 43.4.4, Initial Settings.

(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RI3C generates a START condition. At the same time, the BFREE flag is automatically set to 0, the ICSR2.START flag is automatically set to 1, and the ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDA00 line have matched while the ST bit is 1, the RI3C recognizes that generating of the START condition as requested by the ST bit has been successfully completed, and ICMMR.ACF and ICMMR.TRS flags are automatically set to 1, placing the RI3C in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.

(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDR register. Once the data for transmission are written to the ICDR register, the TDRE flag is automatically set to 0, the data are transferred from the ICDR register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS flag is automatically updated to select master transmit or master receive mode in accordance with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RI3C continues in master transmit mode. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to ICCGR.SP bit to generate a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDR register as the first address transmission. Then, as the second address transmission, write the lower 8 bits of the slave address to ICDR register.

(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RI3C automatically holds the SCL00 line low until the data for transmission are ready, a repeated START condition is generated, or a STOP condition is generated.

(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then, after check that the ICSR2.START flag is 1, set the ICSR2.START flag to 0.

(6) Set the ICCGR.RS bit to 1 (requests to generate a repeated START condition). Upon receiving the request, the RI3C generates a repeated START condition.

(7) After check that the ICSR2.START flag is 1, write the value for transmission (the slave address and the R/W# bit) to the ICDR register.

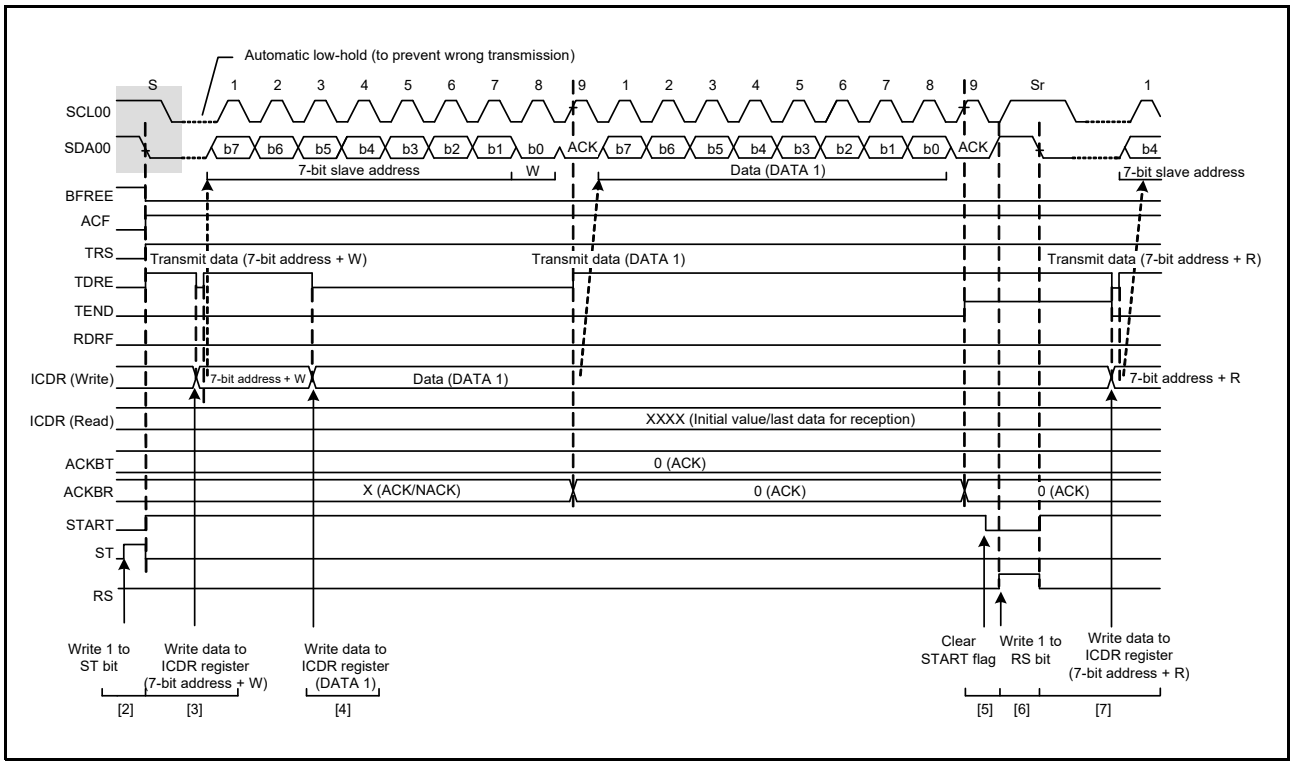


Figure 43.91 Repeated START Condition Generation Timing after the Master Transmission

43.5.7.3 Generating a STOP Condition

The RI3C generates a STOP condition when the ICCGR.SP bit is set to 1.
When the SP bit is set to 1, generation of a STOP condition is requested. When the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.ACF flag is 1 (master mode), the RI3C generates a STOP condition.
A STOP condition is generated in the following sequence.

STOP condition generation

(1) Drive the SDA00 line low (high to low).
(2) Secure the low period of SCL00 line with the time set in the ICFBR.LOW[7:0] bits.
(3) Release the SCL00 line (low to high).
(4) Detect a high of the SCL00 line and secure the STOP condition setup time with the time set in the ICFBR.HIGH[7:0] bits.
(5) Release the SDA00 line (low to high).
(6) Secure the bus free time with the time set in the ICFBR.LOW[7:0] bits.
(7) Set the BFREE flag to 1 (to release the bus).

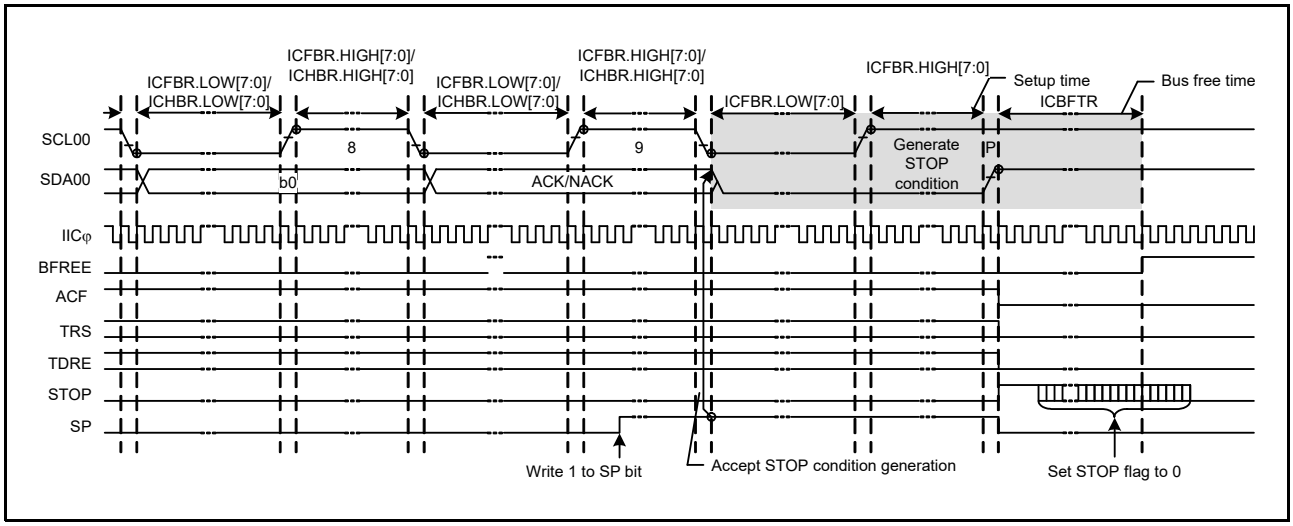


Figure 43.92 STOP Condition Generating Timing (SP Bit)

43.5.8 Bus Hanging (I2C Protocol Mode)

43.5.8.1 Timeout Function

The RI3C includes a timeout function for detecting when the SCL00 line has been stuck longer than the predetermined time. The RI3C can detect an abnormal bus state by monitoring that the SCL00 line is stuck low or high for a predetermined time.

The timeout function monitors the SCL00 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCL00 line changes (rising or falling), but continues to count unless the SCL00 line changes. If the internal counter overflows due to no SCL00 line change, the RI3C can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICSER.TMOE bit is 1. It detects a hung state that the SCL00 line is stuck low or high during the following conditions: (when ICTOR.TMOM[1:0] bits = 00b)

- The bus is busy (ICBSR.BFREE flag is 0) in master mode (ICMMR.ACF flag is 1).
- The RI3C's own slave address is detected (ICSSR register is not 0000 0000h) and the bus is busy (ICBSR.BFREE flag is 0) in slave mode (ICMMR.ACF flag is 0).
- The bus is free (ICBSR.BFREE flag is 1) while generation of a START condition is requested (ICCGR.ST bit is 1).

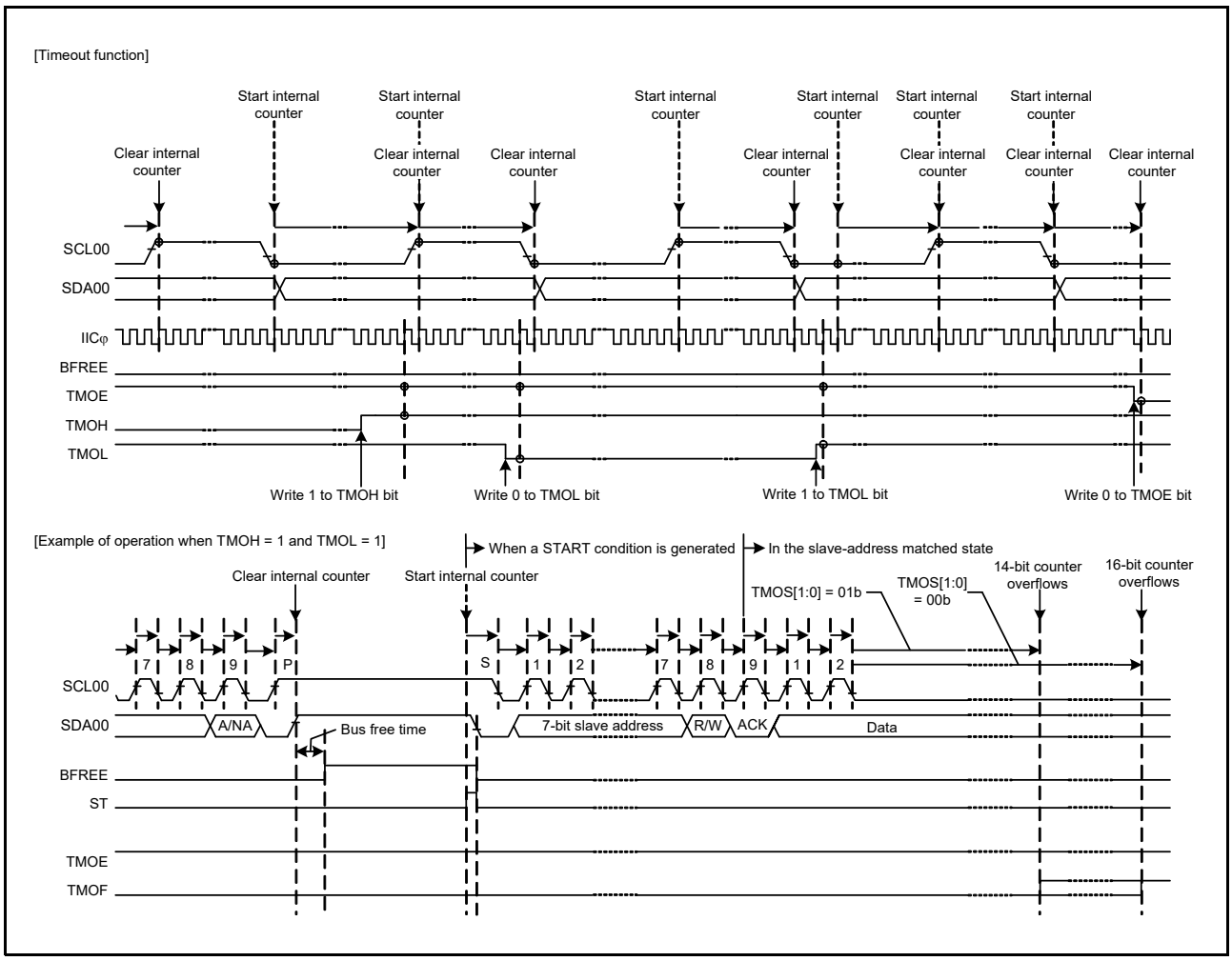


Figure 43.93 Timeout Function (TMOE, TMOS[1:0], TMOH, and TMOL Bits)

43.5.8.2 Additional SCL Output Function

In master mode, the RI3C module has a function for the output of additional SCL to release the SDA00 line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDA00 line from the state of being stuck low by including additional SCL output from the RI3C with single cycles of the SCL as the unit in the case of a bus error where the RI3C cannot generate a STOP condition because the slave device is holding the SDA00 line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICOCR.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICRCCR.CKS[2:0] bits and the HIGH[7:0] and LOW[7:0] bits in the ICFBR register (ICHBR register in Hs-mode) is output from the SCL00 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCL00 pin is held low when the ICBSR.BFREE flag is 0 and held high when the BFREE flag is 1. Consecutive additional clock pulses can be output by the software program writing 1 to the CLO bit after having read CLO = 0.
When the RI3C module is in master mode and the slave device is holding the SDA00 line low because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a STOP condition is not possible. The additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDA00 line from being held low, thus recovering the bus from an unusable state. Release of the SDA00 line by the slave device can be monitored by reading the ICIMR.SDAI flag After confirming release of the SDA00 line by the slave device, complete communications by regenerating the STOP condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled). If using this function with the ICFER.MALE bit set to 1, arbitration may be lost due to a mismatch between the value of the ICOCR.SDAO bit and the level of the SDA00 line.

Conditions for using the ICOCR.CLO bit

- When the bus is free (ICBSR.BFREE flag = 1) or in master mode (ICMMR.ACF flag = 1 and ICBSR.BFREE flag = 0)
- When the communication device does not hold the SCL00 line low

Figure 43.94 shows the operation timing of the additional SCL output function (CLO bit).

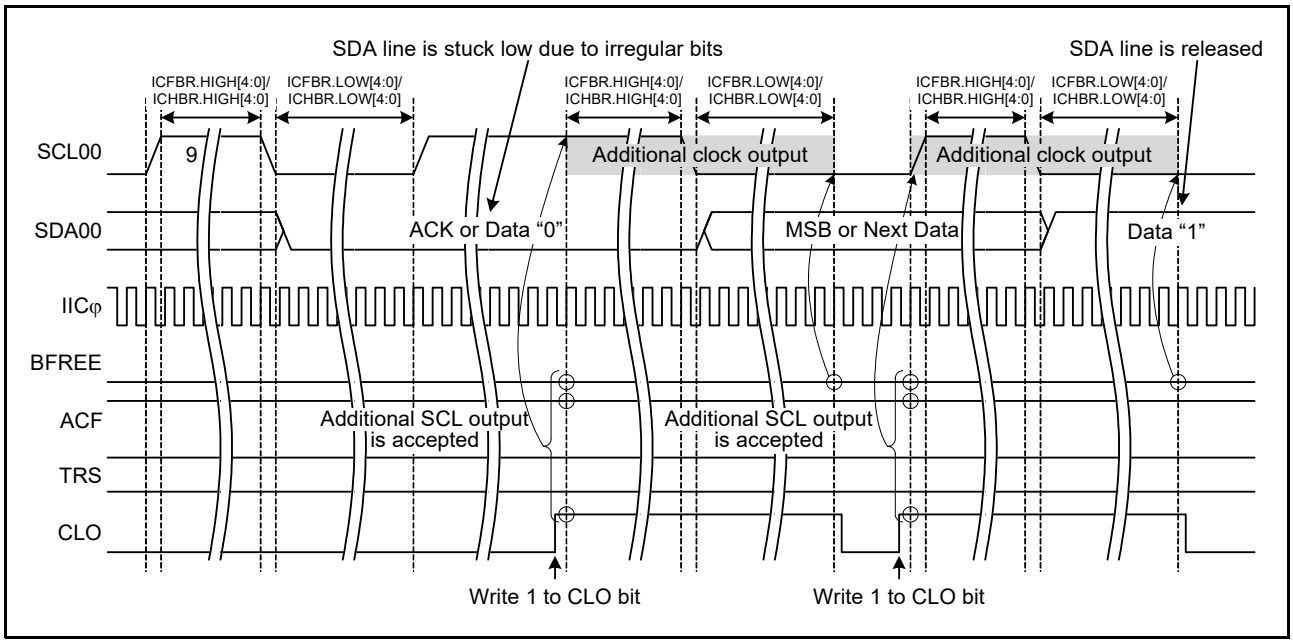


Figure 43.94 Additional SCL Output Function (CLO Bit)

43.5.9 Wake-Up Function (I2C Protocol Mode)

The RI3C includes a wake-up function to return the MCU from low power consumption states. The wake-up function enables data reception even when the operating clock (I3CCLK) is stopped, and is used to generate a wake-up interrupt when the received address matches its own slave address.
To receive data even when the operating clock is stopped, the wake-up unit must be set to asynchronous operation. After a wake-up interrupt occurs, return to synchronous operation so that communication can continue.
There are four wake-up response modes, as shown in Table 43.9.

Table 43.9 Wake-Up Response Modes

Response Mode	Response Timing	Response	SCL Status
Normal wake-up mode 1	Before returning to synchronous operation	ACK	Hold low after ACK response (at the ninth SCL falling edge), release after returning to synchronous operation.
Normal wake-up mode 2	After returning to synchronous operation	ACK	Hold low at the eighth SCL falling edge, release after returning to synchronous operation with ACK response.
Command recovery mode	Before returning to synchronous operation	ACK	Open.
EEP response mode	Before returning to synchronous operation	NACK	Open.

There are five wake-up interrupt sources.

- Host address detection (ICFER.SMBS bit = 1, ICSCR.HOAE bit = 1)
- General call address detection (ICSCR.GCAE bit = 1)
- Slave address 0 detection (ICSCR.SAR0E bit = 1, SAR0.FS bit = 0)
- Slave address 1 detection (ICSCR.SAR1E bit = 1, SAR1.FS bit = 0)
- Slave address 2 detection (ICSCR.SAR2E bit = 1, SAR2.FS bit = 0)

43.5.9.1 Normal Wake-Up Mode 1

In normal wake-up mode 1, the RI3C operates as follows after a wake-up interrupt occurs.

(1) Generates an ACK for the data byte that contains its own slave address.
(2) Holds the SCL line low at the ninth SCL falling edge.
(3) When 1 is written to the ICWUCR.WUFSYNE bit, it returns to synchronous operation and sets the ICWUSR.WUASYNF flag to 0.
(4) Releases the SCL line and returns to normal operation.

Note: When returning from low power consumption states due to wake-up interrupt, the setting of the ICCSCR.WAITRE bit is ignored.

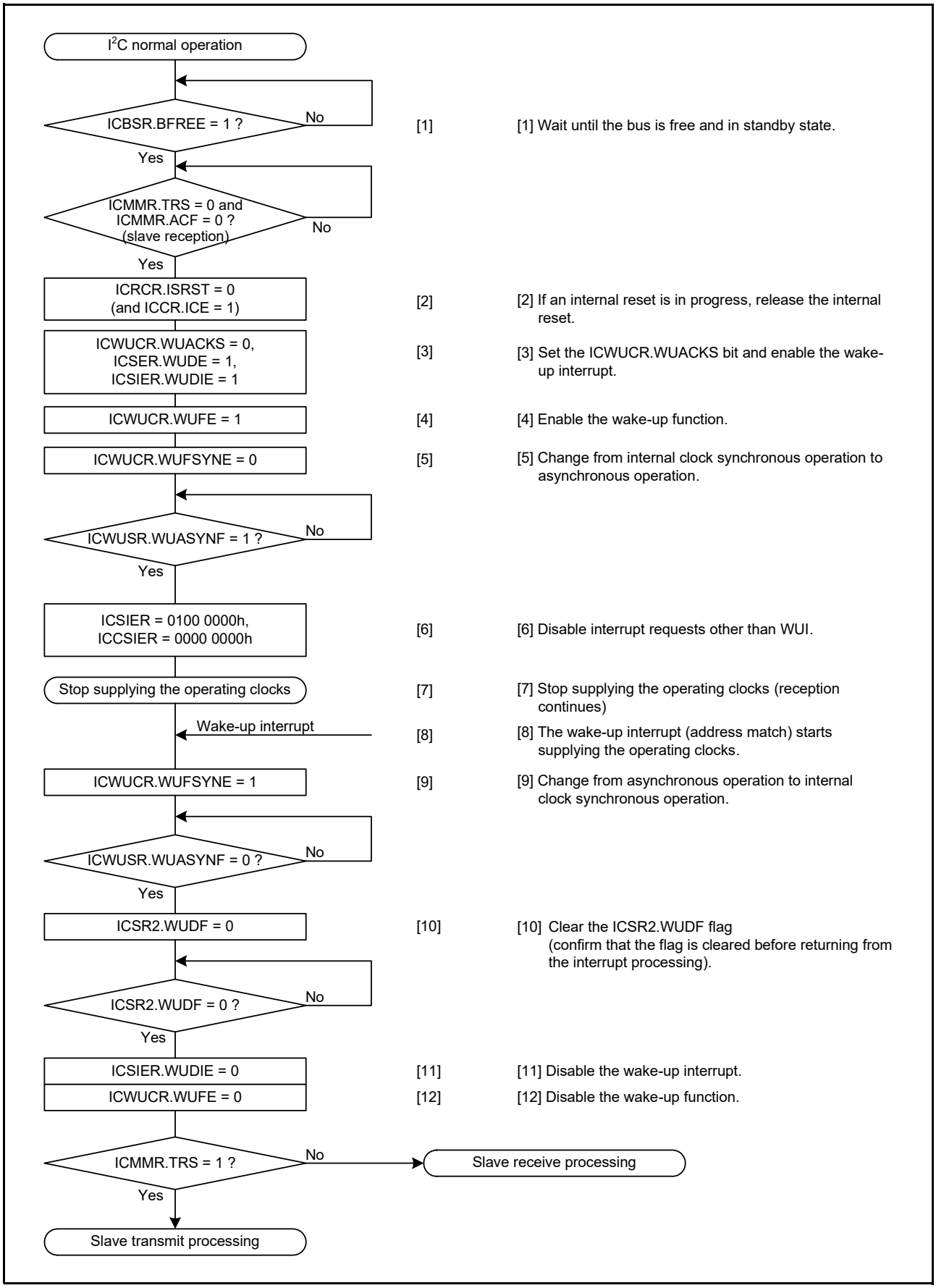


Figure 43.95 Example of Using Normal Wake-Up Mode 1

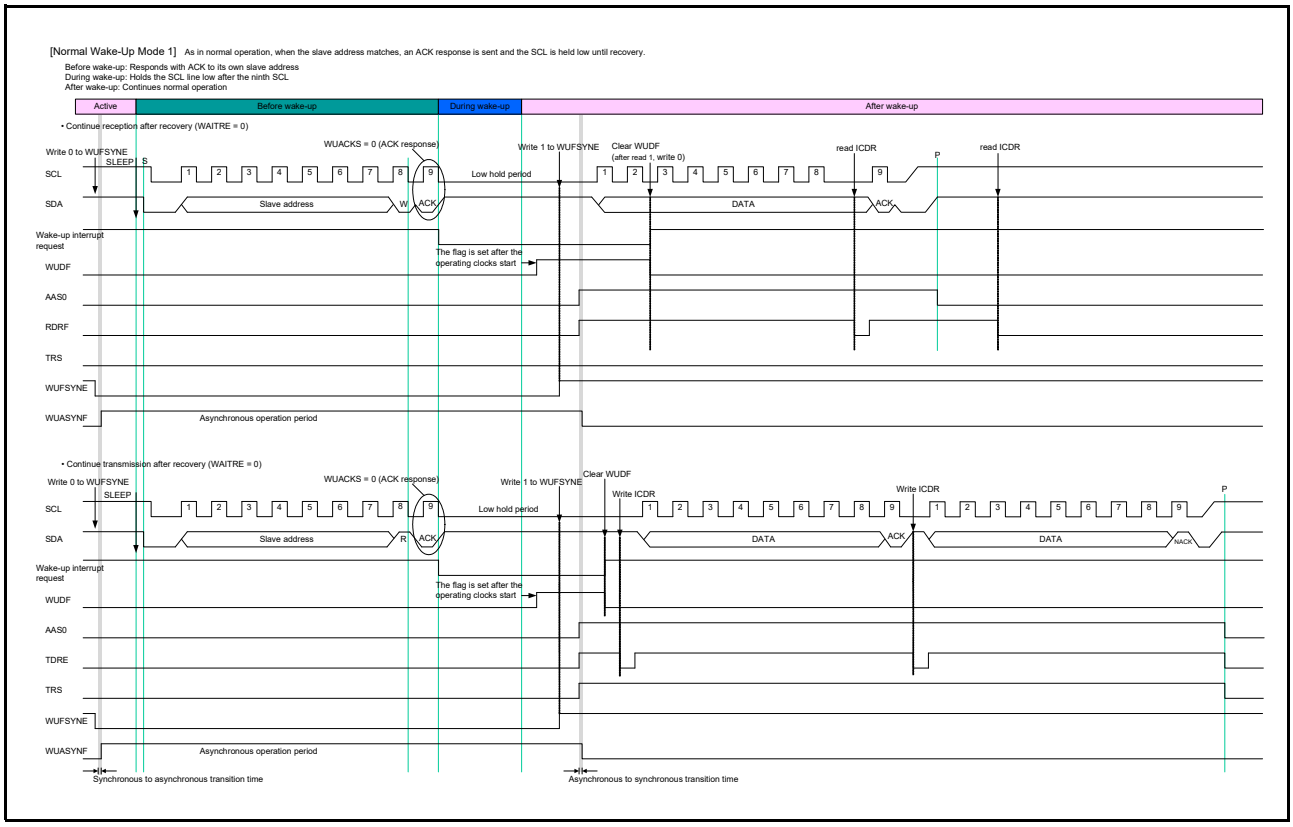


Figure 43.96 Operation Timing of Normal Wake-Up Mode 1

43.5.9.2 Normal Wake-Up Mode 2

In normal wake-up mode 2, the RI3C operates as follows after a wake-up interrupt occurs.

(1) Holds the SCL line low at the eighth SCL falling edge for the data byte that contains its own slave address.
(2) When 1 is written to the ICWUCR.WUFSYNE bit, it returns to synchronous operation and sets the ICWUSR.WUASYNF flag to 0.
(3) Generates an ACK and release the SCL line.
(4) Returns to normal operation.

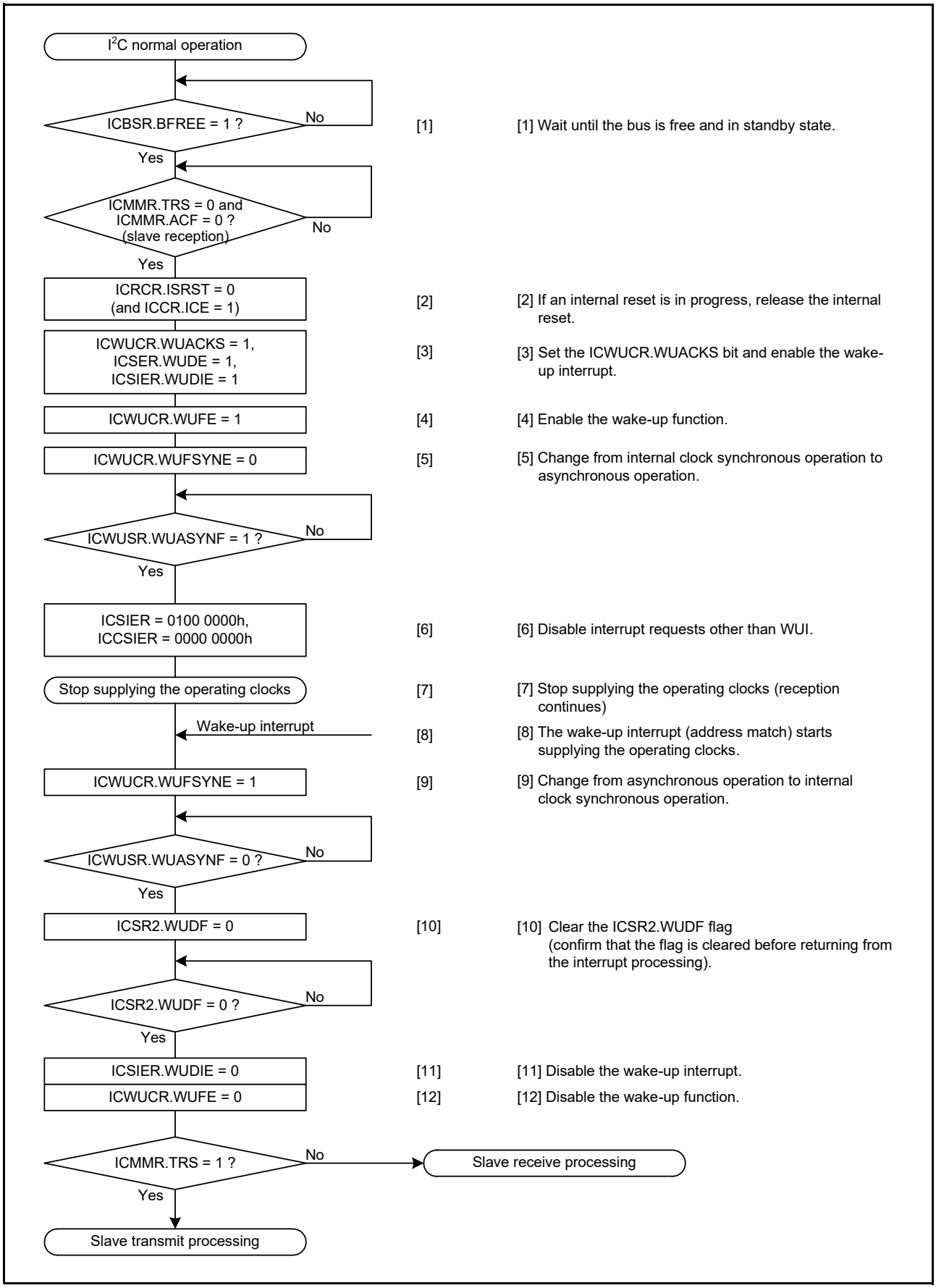


Figure 43.97 Example of Using Normal Wake-Up Mode 2

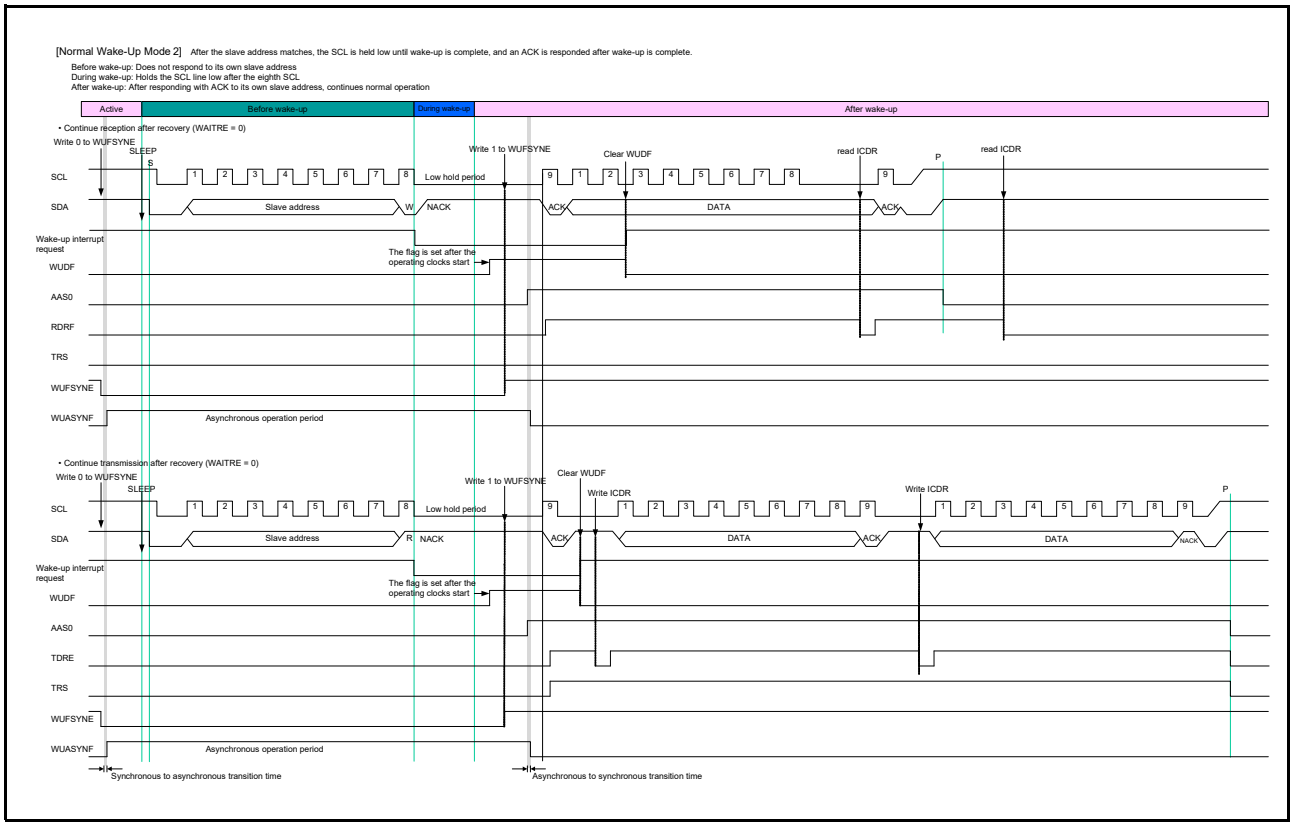


Figure 43.98 Operation Timing of Normal Wake-Up Mode 2

43.5.9.3 Command Recovery Mode and EEP Response Mode

In command recovery mode and EEP response mode, the SCL line is not held low during wake-up operations. As a result, other devices can use the bus during this period, but the RI3C cannot perform subsequent data transmission and reception.

In these modes, the ICRCR.ISRST bit is set to 1, so after returning from low power consumption states, set the ICRCR.MRST bit to 1 (module reset) and then perform the initial settings.

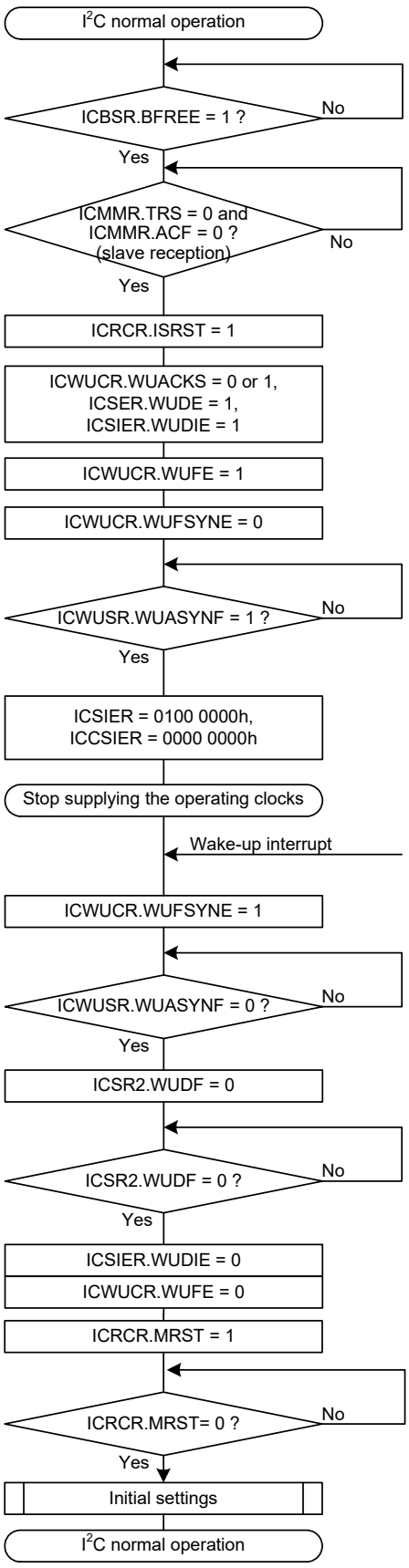


Figure 43.99 Example of Using Command Recovery Mode and EEP Response Mode

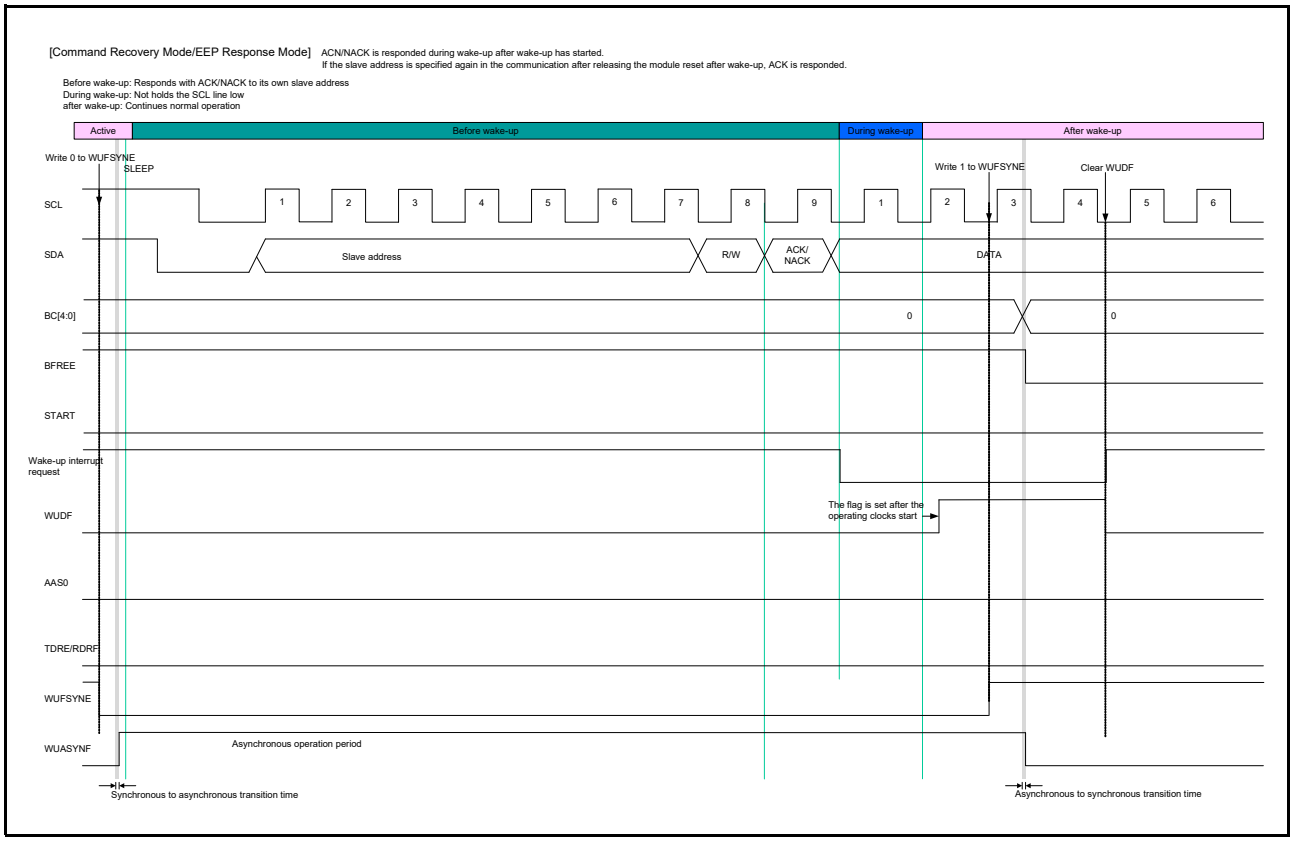


Figure 43.100 Operation Timing of Command Recovery Mode and EEP Response Mode

43.5.9.4 Notes on Using Wake-Up Function

Note the following points when using the wake-up function.

(1) Before setting the ICWUCR.WUFSYNE bit to 0 (asynchronous operation), set the following registers/bits.
   (a) Set the ICWUCR.WUFE, ICSER.WUDE, and ICSIER.WUDIE bits to 1, and the ICMMR.ACF and TRS flags to 0.
   (b) The wake-up function does not support device ID addresses and 10-bit address formats. Set the ICSCR.DIDE bit and SARy.FS bit (y = 0 to 2) to 0.
   (c) The timeout function cannot be used. Set the ICSER.TMOE bit to 0.

(2) When the wake-up detection circuit is disabled, the next reception may start if a START condition is detected at the timing when the ICWUCR.WUFSYNE bit is written to 0. In this case, the ICWUSR.WUASYNF flag becomes 1 after the communication is completed and a STOP condition is detected.

(3) After setting the ICWUCR.WUFSYNE bit to 0, do not rewrite the ICFER, ICCSCR, ICACKR, ICICR, ICTCR, and SARy registers until the ICWUSR.WUASYNF flag is set to 1. If rewritten, RI3C may malfunction.

(4) Do not access the RI3C registers except for the ICSIER register, ICCSIER register, and ICWUCR.WUFSYNE bit while the ICWUSR.WUASYNF flag is 1 (wake-up detection circuit enabled).

(5) When using normal wake-up mode 2, do not set the ICACKR.ACKBT bit to 1.

(6) If the ICWUCR.WUFSYNE bit is set to 1 (synchronous operation) when no wake-up condition is detected or when the bus is free, the wake-up detection circuit is disabled only after a STOP condition is detected.

43.5.10 SMBus Operation (I^2C Protocol Mode)

The RI3C is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICFER.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the ICRCCR.CKS[2:0] bits, the ICFBR.HIGH[7:0] bits, and the ICFBR.LOW[7:0] bits. In addition, determine the values of the ICOCR.DLCS bit and the ICOCR.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). When the RI3C is used only as a slave device, the transfer rate setting is not necessary, whereas the ICFBR.LOW[7:0] bits need to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers 0 to 2 (SAR0, SAR1, and SAR2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARy (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration lost detection function.

43.5.10.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with a timer such as TMR using a START condition detection interrupt (STI) and STOP condition detection interrupt (SPI) of the RI3C. The measured timeout period must be within cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus standard.
If the time measured with the timer exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the slave device must release the bus by writing 1 to the ICRCR.ISRST bit to perform an internal reset of the RI3C. When an internal reset is performed, the RI3C stops driving the bus for the SCL00 pin and SDA00 pin and make the SCL00/SDA00 pin outputs high-impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment bit to STOP condition

To measure timeout for master devices, measure these periods with a timer such as TMR using a START condition detection interrupt (STI), STOP condition detection interrupt (SPI), and transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RI3C. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus standard, and the total of all $T_{LOW:MEXT}$ from START condition to STOP condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICCSR.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICCSCR.WAITAE bit 0 until the byte just before reception of the final byte in master receive mode. While the WAITAE bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the timer exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus standard or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the master device must stop the transaction by generating a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDR register).

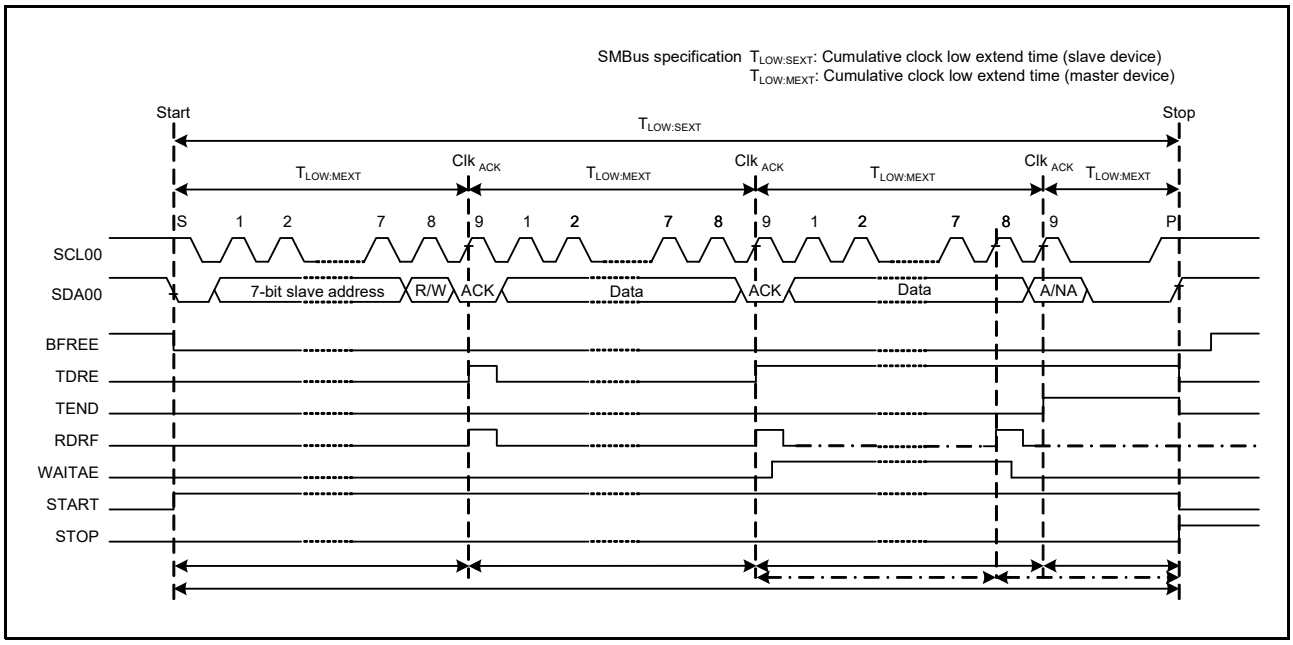


Figure 43.101 SMBus Timeout Measurement

43.5.10.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RI3C. For the CRC generating polynomials of the CRC calculator, refer to section 50, CRC Calculator (CRCA).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICCSCR.WAITAE bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCL00 line low at the falling edge of the eighth clock pulse.

43.5.10.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RI3C has a function for detecting the host address. To detect the host address as a slave address, set the ICFER.SMBS bit and the ICSCR.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

43.5.11 CCC detection function (I3C protocol mode)

In case of Broadcast CCC

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) In accordance with the CCC, the following data is stored. (Storage destination: refer to Table 43.10)
(5) Store the Receive Status Descriptor into the receive status queue.

In case of Broadcast CCC (ENTDAA)

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition.
(2) Respond with ACK.
(3) Receive ENTDAA.
(4) If receives Broadcast Address (7Eh) and RnW = 1 after repeated START condition.
(5) When the Dynamic Address is not assigned, ACK response is done.
(6) This Provisioned ID (the ICPIDHR register and bits b15 to b0 of the ICPIDLR register), BCR (bits b15 to b8 of the ICDCTR register), DCR (bits b7 to b0 of the ICDCTR register) are transmitted.
(7) When winning the arbitration in a transmission of the above Step 6, the Dynamic Address following that is received.
When losing arbitration in a transmission of the above Step 6, processing of Step 6 is repeated from Step 4.
(8) When parity of the Dynamic Address is valid, ACK response is done.
(9) When parity of the Dynamic Address is invalid, NACK replies, and repeat the process from Steps 4 to 7.
(10) ICDARy.DADR[6:0] bits are renewed and the ICDAMRy.DAV flag is set to 1.
(11) Upon detecting the STOP condition, Store the Receive Status Descriptor into the receive status queue.

In case of Direct Write CCC

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) Receive Dynamic Address and RnW = 0 after repeated START condition.
(5) Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, RI3C responds with ACK.
If they do not match, it responds with NACK and waits for repeated START condition or STOP condition.
(6) In accordance with the CCC, the following data is stored. (Storage destination: refer to Table 43.10)
(7) Store the Receive Status Descriptor into the receive status queue.

In case of Direct Read CCC

(1) It receives Broadcast Address (7Eh) and RnW = 1 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) Receive Dynamic Address and RnW = 1 after repeated START condition.
(5) Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, RI3C responds with ACK.
If they do not match, it responds with NACK and waits for repeated START condition or STOP condition.
(6) Respond from register according to CCC. (Responding CCC: refer to Table 43.10)
(7) Store the Receive Status Descriptor into the receive status queue.

Table 43.10 Common Command Code Operation

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
00h	Broadcast	ENEC	Yes	—	Register
01h	Broadcast	DISEC	Yes	—	Register
02h	Broadcast	ENTAS0	No	—	Register
03h	Broadcast	ENTAS1	No	—	Register
04h	Broadcast	ENTAS2	No	—	Register
05h	Broadcast	ENTAS3	No	—	Register
06h	Broadcast	RSTDAA	No	—	Register
07h	Broadcast	ENTDAA	Yes	Yes	Register
08h	Broadcast	DEFTGTS	Yes	—	FIFO
09h	Broadcast	SETMWL	Yes	—	Register
0Ah	Broadcast	SETMRL	Yes	—	Register
0Bh	Broadcast	ENTTM	Yes	—	Register
29h	Broadcast	SETAASA	No	—	Register
80h	Direct Write	ENEC	Yes	—	Register
81h	Direct Write	DISEC	Yes	—	Register
82h	Direct Write	ENTAS0	No	—	Register
83h	Direct Write	ENTAS1	No	—	Register
84h	Direct Write	ENTAS2	No	—	Register
85h	Direct Write	ENTAS3	No	—	Register
86h	Direct Write	RSTDAA	No	—	Register
87h	Direct Write	SETDASA	Yes	—	Register
88h	Direct Write	SETNEWDA	Yes	—	Register
89h	Direct Write	SETMWL	Yes	—	Register
8Ah	Direct Write	SETMRL	Yes	—	Register
8Bh	Direct Read	GETMWL	—	Yes	Register
8Ch	Direct Read	GETMRL	—	Yes	Register
8Dh	Direct Read	GETPID	—	Yes	Register
8Eh	Direct Read	GETBCR	—	Yes	Register
8Fh	Direct Read	GETDCR	—	Yes	Register
90h	Direct Read	GETSTATUS	—	Yes	Register
91h	Direct Read	GETACCCR	—	Yes	Register
94h	Direct Read	GETMXDS	—	Yes	Register

43.5.12 Clock Stalling (I3C Protocol Mode)

RI3C has the function of stalling the SCL during the SCL Low period.
The SCL stall control is described in the table below.

Table 43.11 I3C Clock Stalling

Clock Stalling Condition	Clock Stalling Control	Clock Stalling Period
I3C Transfer, ACK/NACK Phase	ICSTCR.APSE bit setting	During the count period of ICSTCR.STT[15:0] value
	Transmit FIFO Empty	Until data is written to the transmit FIFO
	Receive FIFO Full	Until data is read from the receive FIFO
I3C Write Data Transfer, Parity Bit	ICSTCR.PBSE bit setting	During the count period of ICSTCR.STT[15:0] value
	Transmit FIFO Empty	Until data is written to the transmit FIFO
I3C Read Transfer, Transition Bit	Receive FIFO Full	Until data is read from the receive FIFO
Assigned Address Phase	ICSTCR.AASE bit setting	During the count period of ICSTCR.STT[15:0] value

The following figure shows the stalling timing of each condition.

(1) I3C Transfer, ACK/NACK Phase

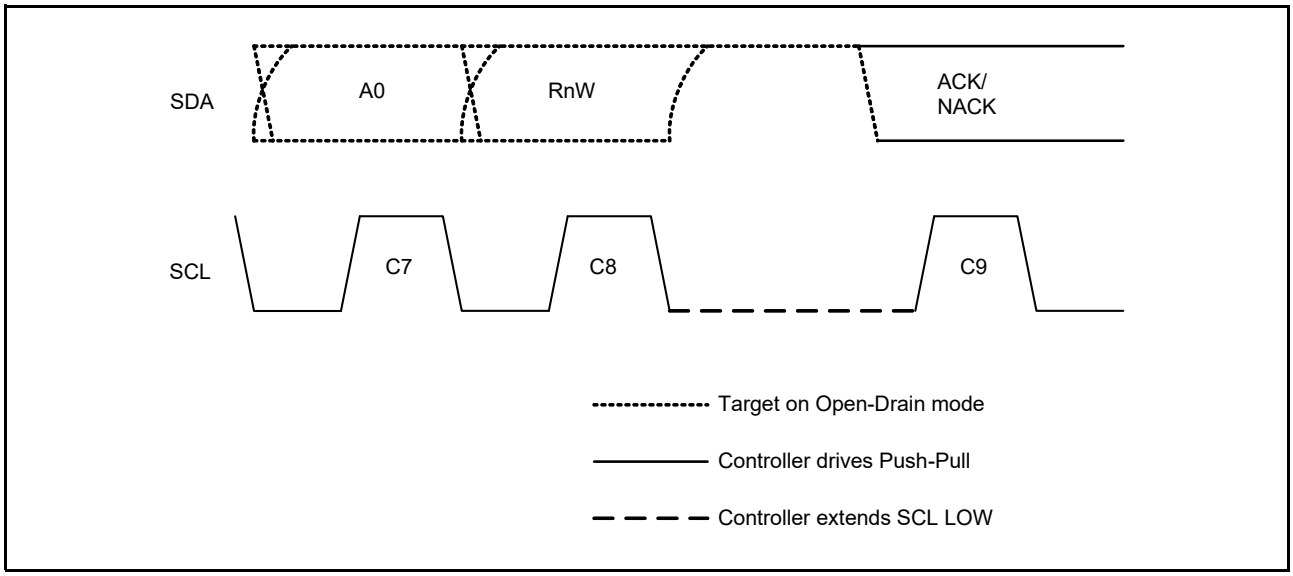


Figure 43.102 Controller Clock Stalling in ACK Phase

(2) I3C Write Data Transfer, Parity Bit

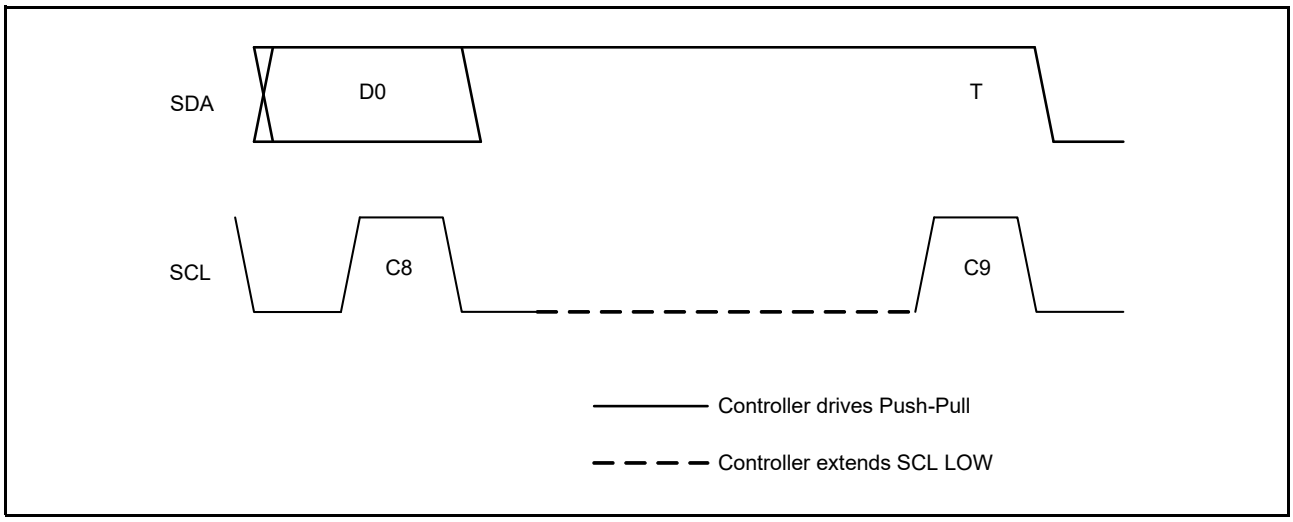


Figure 43.103 Controller Clock Stalling in Write Parity Bit

(3) I3C Read Transfer, Transition Bit

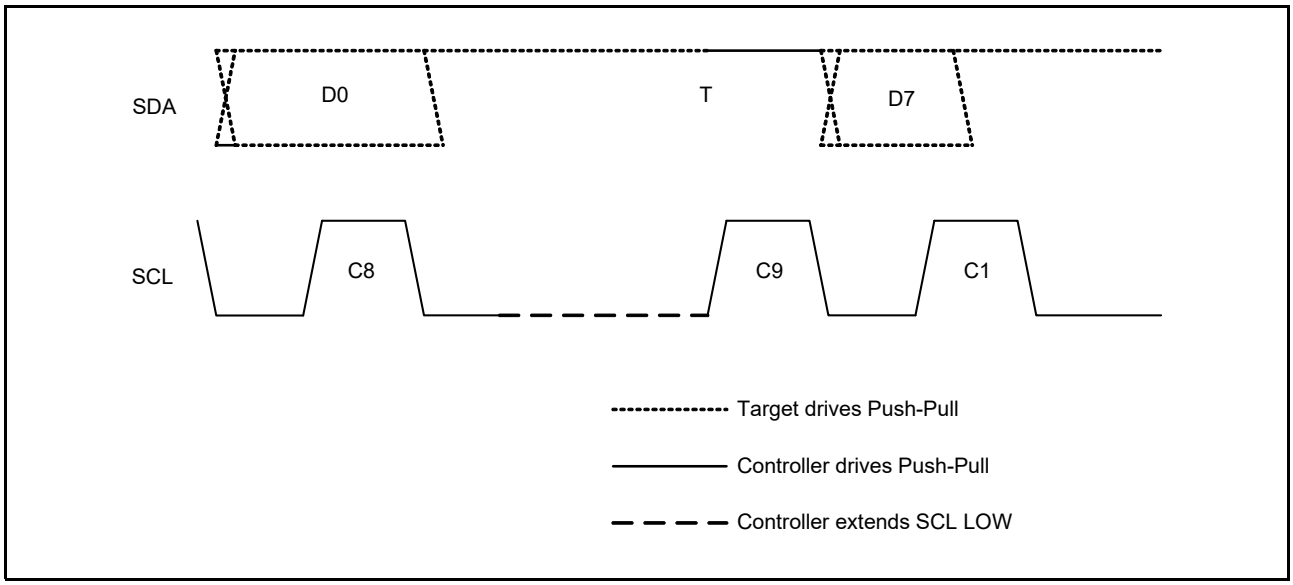


Figure 43.104 Controller Clock Stalling in T-bit Before Next Read Data

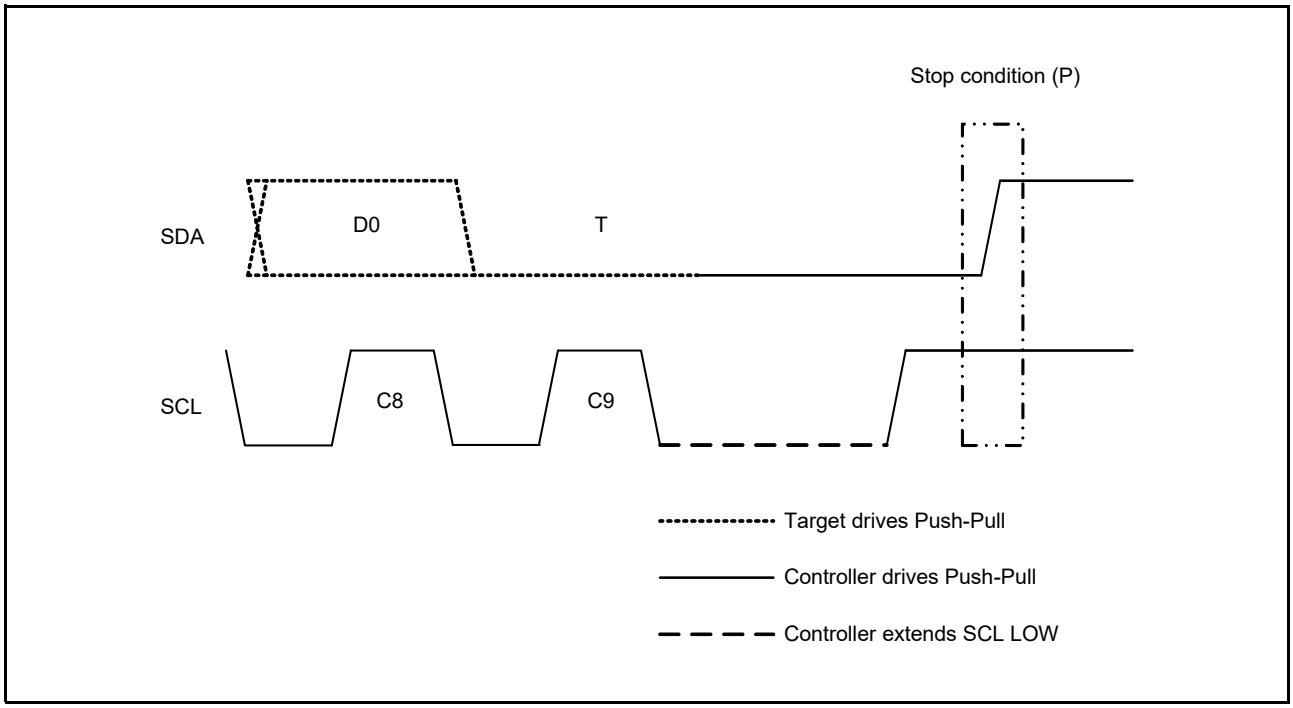


Figure 43.105 Controller Clock Stalling in T-bit Before STOP Condition

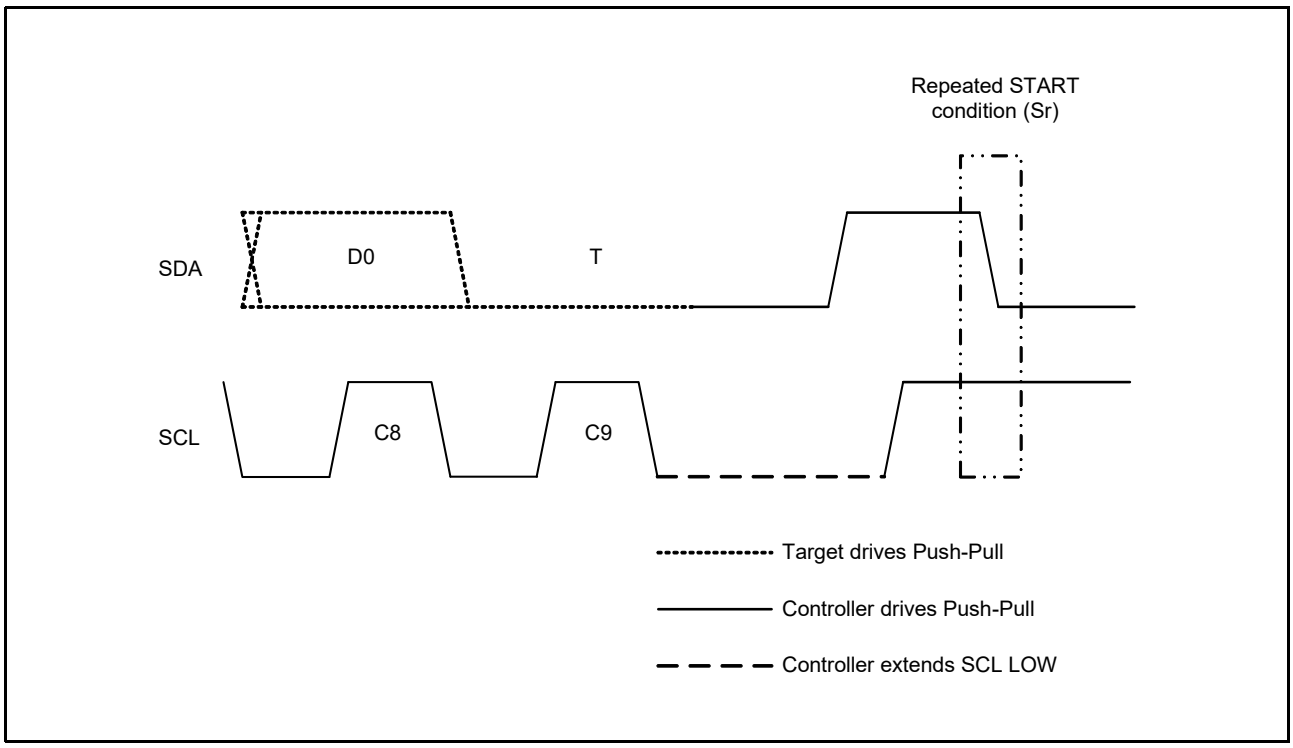


Figure 43.106 Controller Clock Stalling in Low T-bit before Repeated START Condition

(4) Dynamic Address Assignment, First Bit of Assigned Address

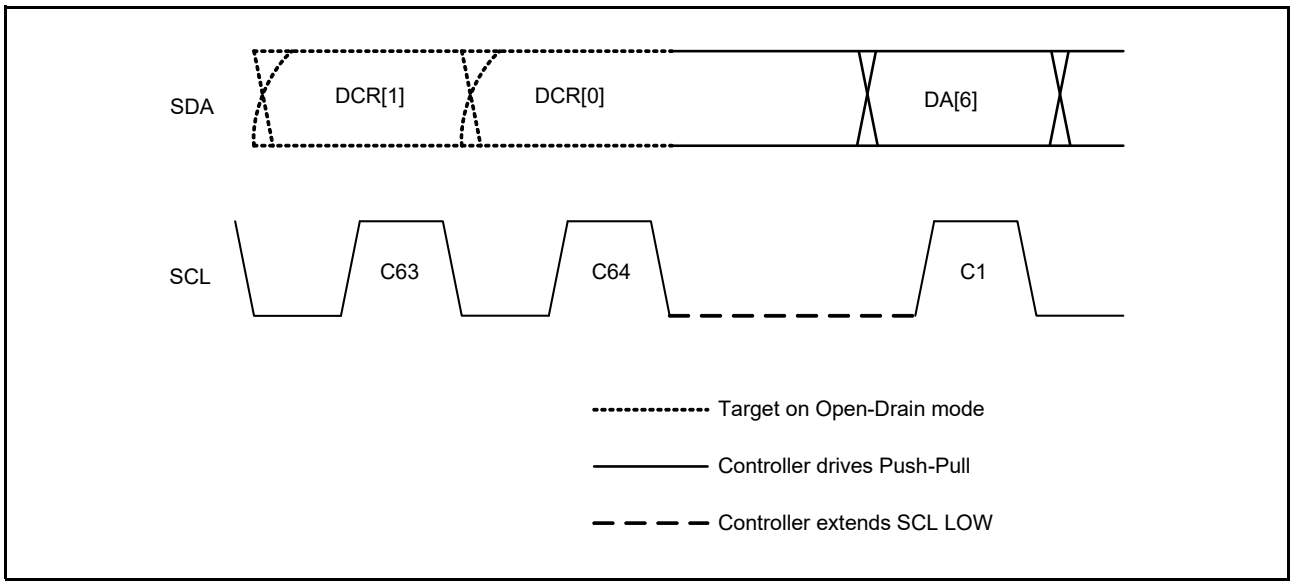


Figure 43.107 Controller Clock Stalling in Dynamic Address First Bit

43.5.13 IBI (I3C Protocol Mode)

RI3C detects IBI in the arbitrated Address header following a START condition (but not following a repeated START condition). If START Request (SDA Low Drive) is issued from Target Device, RI3C drives SCL low and completes START condition. After that, it supplies SCL and receives IBI Request.
The IBI to be detected is classified into the following three types.

- Target Interrupt Request
- Controller Role Request

The operation when detecting each IBI is described below.

43.5.13.1 Target Interrupt Request

(1) Detect Target Address with RnW bit High in Address header.
(2) Compare the detected Target Address with the DADR[7:0] bits in each DAT (the ICTDATRm register).
(3) When it does not match the ICTDATRm.DADR[7:0] bits:
• Responds with NACK, then issues a STOP condition.
When it matches the ICTDATRm.DADR[7:0] bits and the ICTDATRm.TIRRJ bit = 1:
• RI3C operates in the following order:
1. Responds with NACK.
2. Issues a repeated START condition, then automatically issues Direct DISEC CCC to the detected Target.
3. Issues a STOP condition.
When it matches the ICTDATRm.DADR[7:0] bits and the ICTDATRm.TIRRJ bit = 0:
• Responds with ACK.
(4) When the ICTDATRm.IBIPL bit is 0:
• Issues a STOP condition.
When the ICTDATRm.IBIPL bit is 1:
• RI3C operates in the following order:
1. Drives the SCL to receive the IBI Data from the Target following the ACK response and receives IBI Data.
2. It stores the received IBI Data into the IBI queue.
3. Each time IBI Data of the size set by the ICQBTCR.IDSS[7:0] bits is received, the IBI Status Descriptor is stored in the IBI queue.
(5) After detection of Low of T-bit following IBI Data, issues a STOP condition.
(6) After issuing the STOP condition,
NACK response:
• If the ICINCR.RTIRN bit is 0, the IBI Status Descriptor is not stored into the IBI queue.
• If the ICINCR.RTIRN bit is 1, the IBI Status Descriptor is stored into the IBI queue.
ACK response:
• Stores the IBI Status Descriptor into the IBI queue.

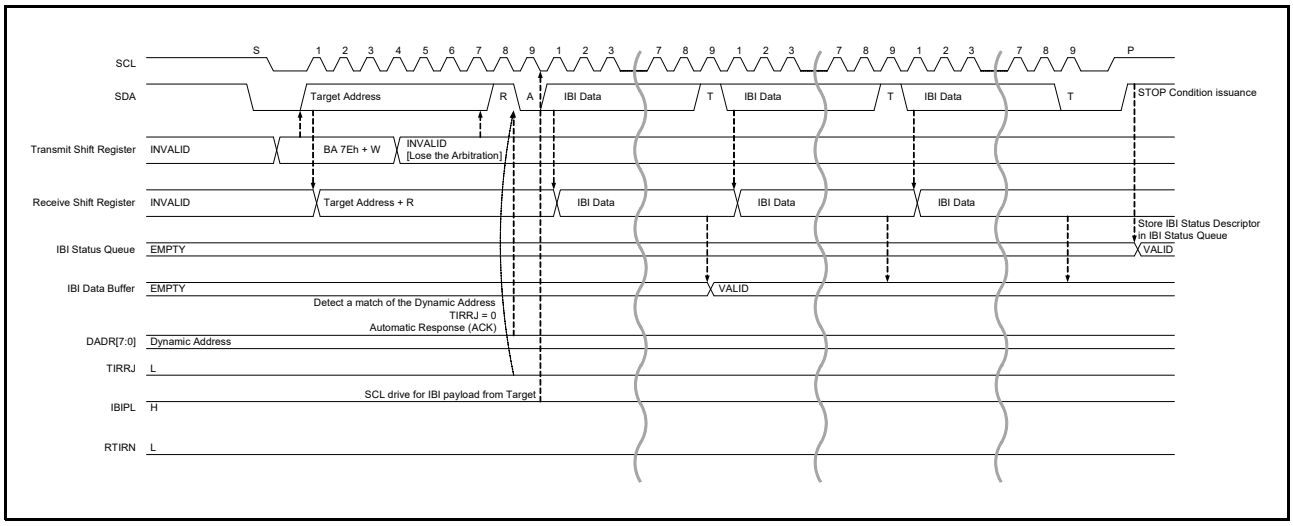


Figure 43.108 Target Interrupt Request: ACK and IBIPL = 1

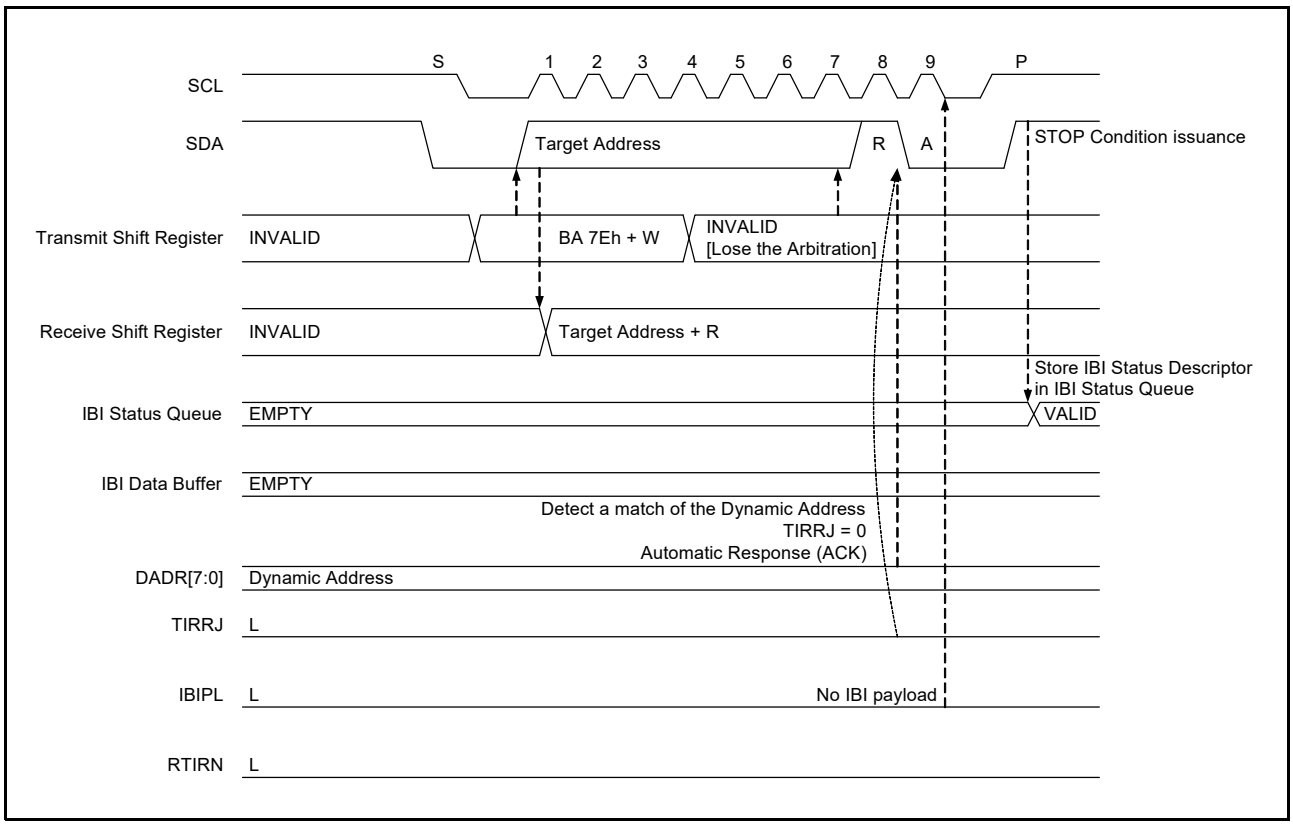


Figure 43.109 Target Interrupt Request: ACK and IBIPL = 0

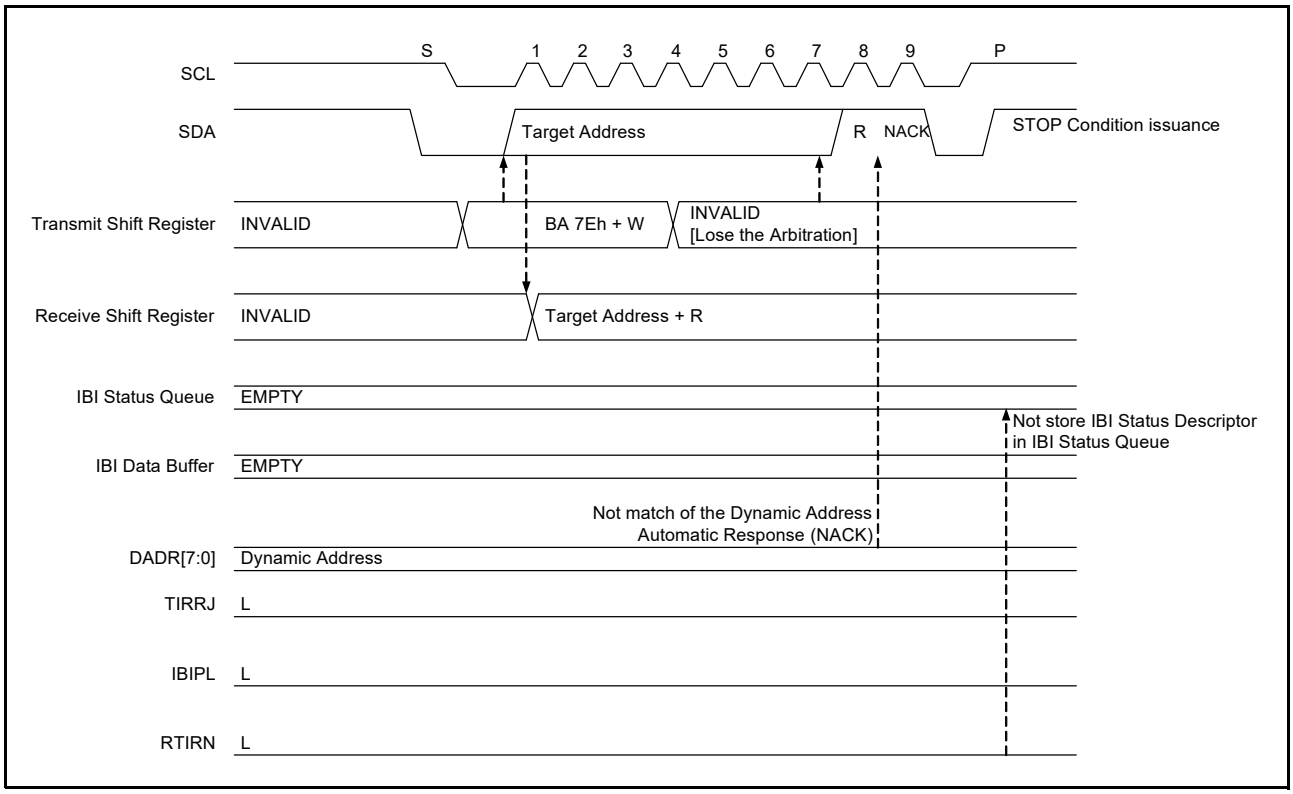


Figure 43.110 Target Interrupt Request: NACK (Not Match the DADR[7:0] Bits) and RTIRN = 0

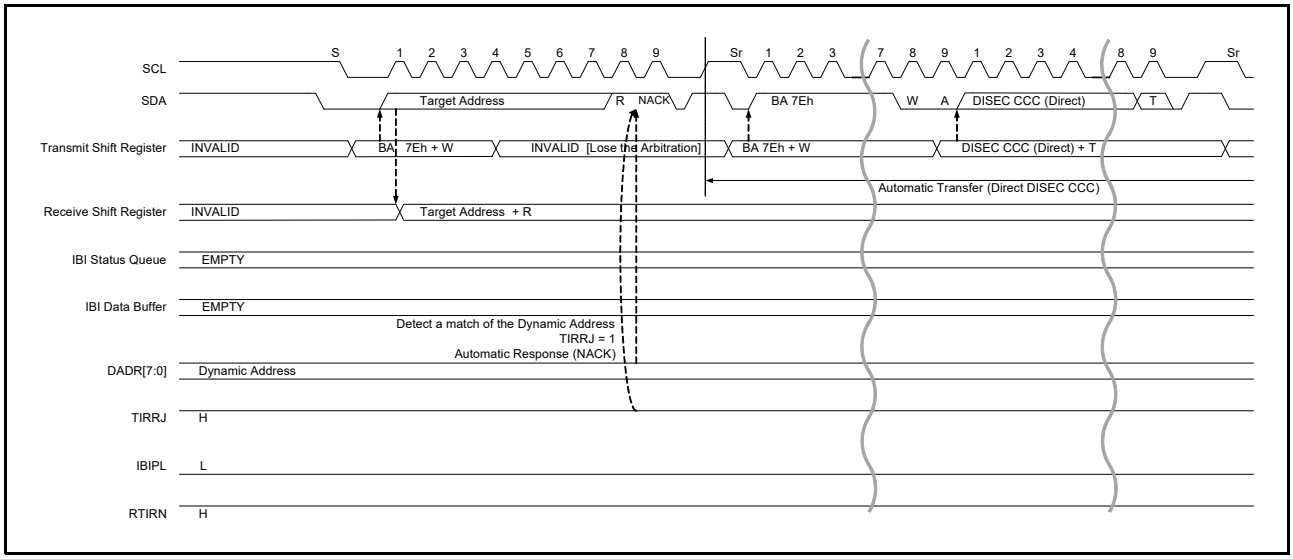


Figure 43.111 Target Interrupt Request: NACK (TIRRJ = 1) and RTIRN = 1 (1/2)

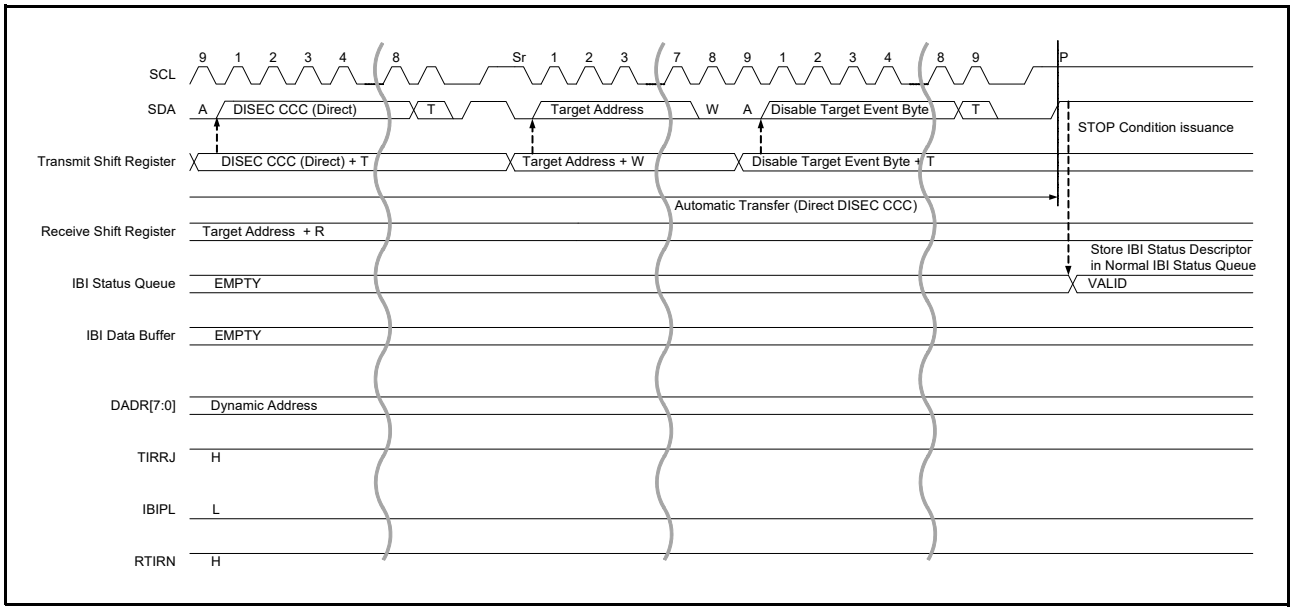


Figure 43.112 Target Interrupt Request: NACK (TIRRJ = 1) and RTIRN = 1 (2/2)

43.5.13.2 Controller Role Request

(1) Detect Target Address with RnW bit Low in Address header.

(2) Compare the detected Target Address with the DADR[7:0] bits in each DAT (the ICTDATRm register).

(3) When it does not match the ICTDATRm.DADR[7:0] bits:

• Responds with NACK, then issues a STOP condition.

When it matches the ICTDATRm.DADR[7:0] bits and the ICTDCTRm.ROLE[1:0] bits (BCR[7:6]) are “00b” (I3C Target):

• Responds with NACK, then issues a STOP condition.

When it matches the ICTDATRm.DADR[7:0] bits and the ICTDCTRm.ROLE[1:0] bits (BCR[7:6]) are “01b” (I3C Controller capable):

(a) When the ICTDATRm.CRRRJ bit is 1:

• RI3C operates in the following order.

1. Responds with NACK.
2. Issues a repeated START condition and automatically issues Direct DISEC CCC to the detected Target.
3. Issues a STOP condition.

(b) When the ICTDATRm.CRRRJ bit is 0:

• Responds with ACK, then issues a STOP condition.

(4) After issuing the STOP condition,

NACK response:

• If the ICINCR.RCRRN bit is 0, the IBI Status Descriptor is not stored into the IBI queue.

• If the ICINCR.RCRRN bit is 1, the IBI Status Descriptor is stored into the IBI queue.

ACK response:

• Stores the IBI Status Descriptor into the IBI queue.

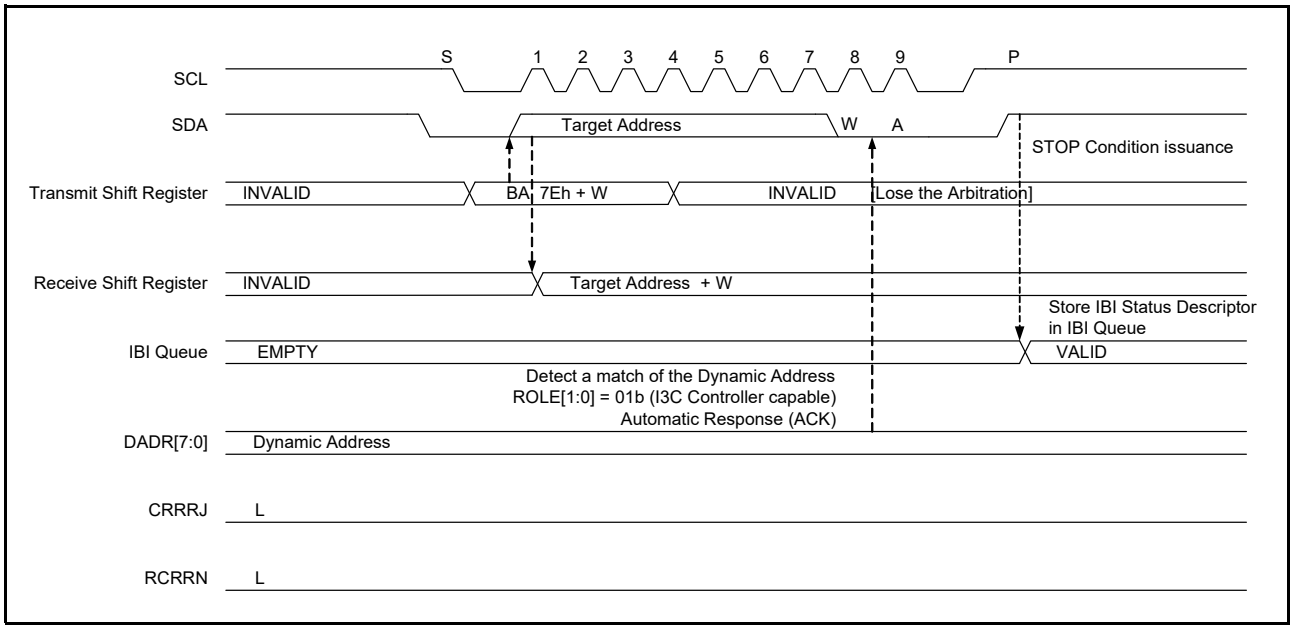


Figure 43.113 Controller Role Request: ACK

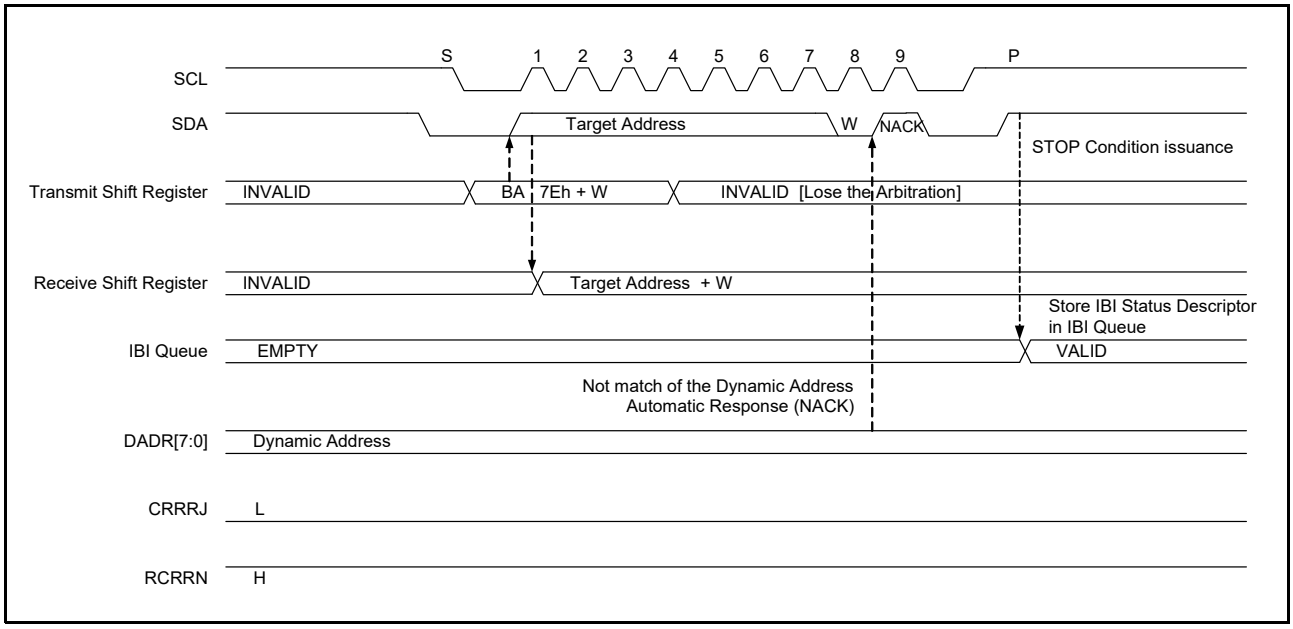


Figure 43.114 Controller Role Request: NACK (Not Match the DADR[7:0] Bits) and RCRRN = 1

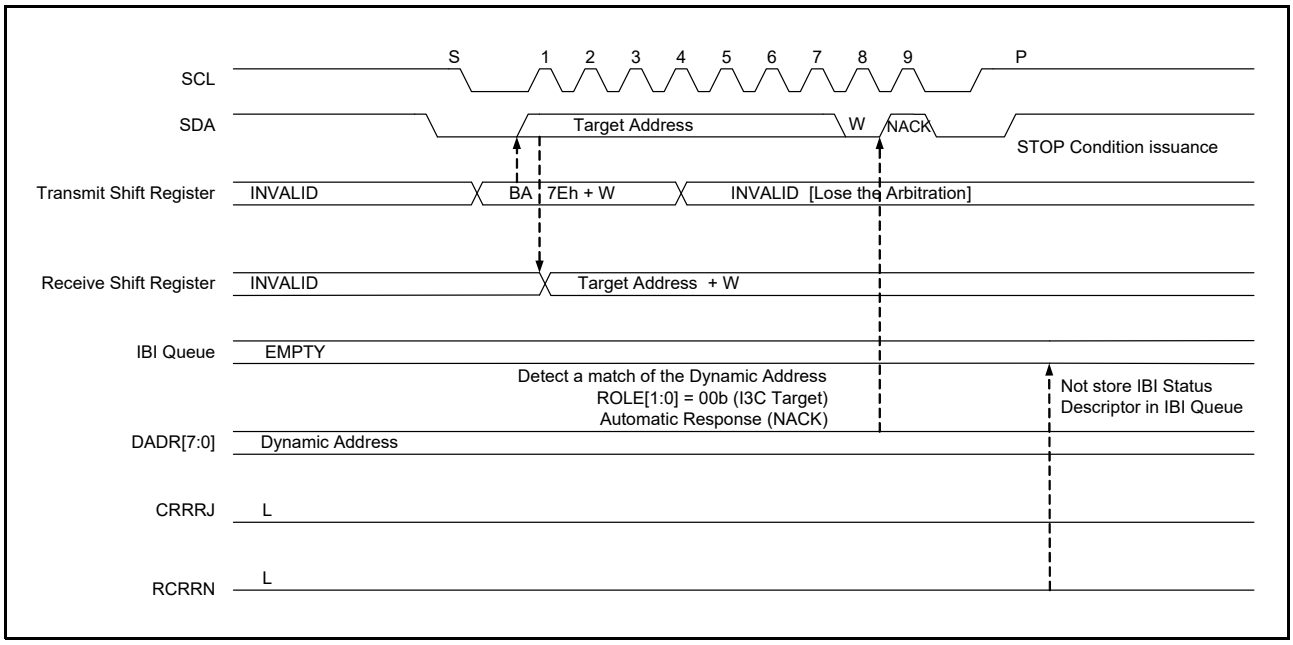


Figure 43.115 Controller Role Request: NACK (ROLE[1:0] = 00b (I3C Target)) and RCRRN = 0

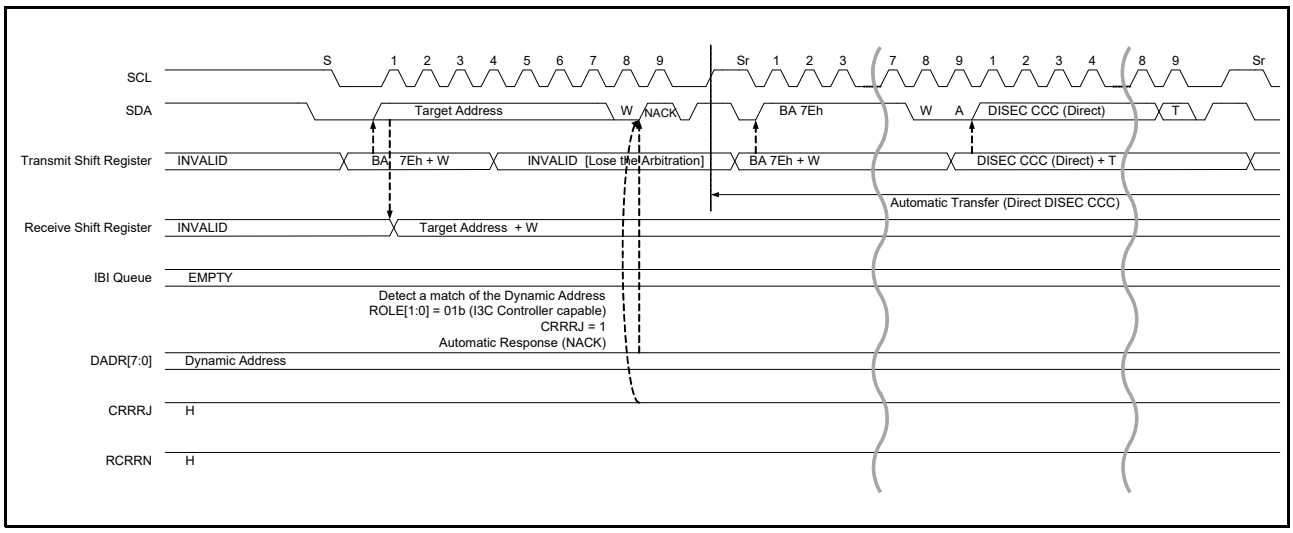


Figure 43.116 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (1/2)

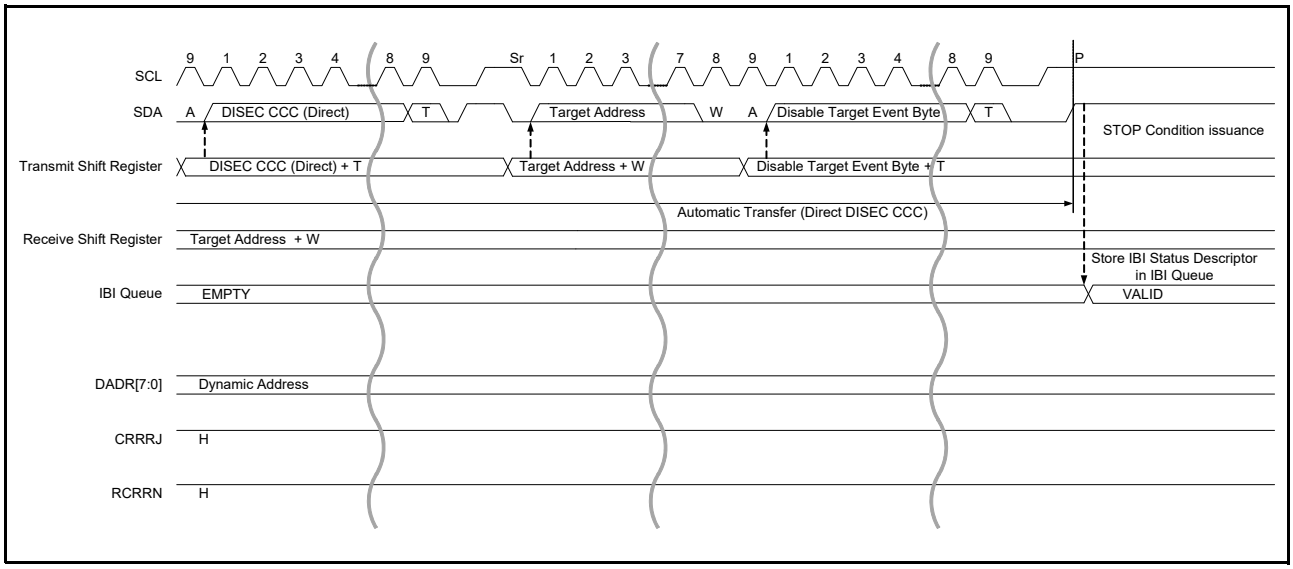


Figure 43.117 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (2/2)

43.5.14 Timing Control (I3C Protocol Mode)

Timing Control is a function that allows the Controller to efficiently read data from the Target by controlling and understanding the timing at which the Target device samples sensor values. For details, refer to Section 6.6, Timing Control in the MIPI I3C Specification Version 1.2.

RI3C supports the following three modes.

- Sync Mode
- Async Mode 0 (Asynchronous Basic Mode)
- Async Mode 1 (Asynchronous Advanced Mode)

43.5.14.1 Sync Mode

Whether the RI3C is the Controller or the Target, operate a counter with a period of T_Ph (T_Ph counter) using a timer such as GPTW, and configure the ELC so that the counter value can be captured with a synchronous timing event.

(1) I3C Controller

When the ICSYTCR.STOE bit is set to 1, a synchronous timing event is output when the Controller sends a SETXTIME CCC with the Sync Tick (ST) Sub-Command (ST Message).

Calculate the delay time from the start of counting of the T_Ph counter to the start of sending the ST message using the value of the T_Ph counter captured by the synchronous timing event, and send it in a SETXTIME CCC with the Delay Time (DT) Sub-Command (DT Message).

Send an ST Message and a DT Message each time the T_Ph counter overflows.

(2) I3C Target

When the ICSYTCR.STOE bit is set to 1, a synchronous timing event is output each time the Target detects a START condition. When an ST Message is received, the STOE bit is set to 0.

After confirming receipt of the ST Message in the Receive Status Descriptor, correct the T_Ph counter start timing based on the value of the T_Ph counter captured by the synchronous timing event and the delay time obtained from the subsequent DT Message.

43.5.14.2 Async Mode 0 (Asynchronous Basic Mode)

When performing timing control in Async Mode 0, set the ICASTCR register as necessary.

(1) I3C Controller

Async Mode 0 uses a 32-bit C_REF counter and a 16-bit C_C2 counter.

- C_REF counter*1
  When the ICASTCCR.ATCE bit is set to 1, the C_REF counter starts counting. The count value of the C_REF counter is captured on the first rising edge of SCL after the ACK response to IBI.
- C_C2 counter
  When the ICASTCCR.ATCE bit is 1, the C_C2 counter starts counting on the first rising edge of SCL after the ACK response to IBI. The count value of the C_C2 counter is captured on the first rising edge of SCL after the T bit of the Mandatory Byte.

When an IBI is received from a Target whose corresponding ICTDATRm.IBITSE bit (m = 0 to 7) is 1, the captured values of the C_REF counter and C_C2 counter are each stored as 64-bit data after the IBI Status Descriptor. The Mandatory Byte and subsequent data are stored after that. The value of the DATA_LENGTH field in the IBI Status Descriptor is increased by 16.

Note 1. If the length of the C_REF counter is insufficient at 32 bits, use the C_REF counter overflow event to count up a timer such as GPTW. The C_REF capture event can be used to capture the count value. To use these event outputs, set the ICASTCR.CREFOE bit to 1.

(2) I3C Target

Async Mode 0 uses a 16-bit T_C1 counter and an 8-bit T_C2 counter.

- T_C1 counter
  When the trigger specified by the ICASTCR.ATTRGS bit is input while the ICASTCCR.ATCE bit is set to 1, the T_C1 counter starts counting. The count value of the T_C1 counter is captured on the first rising edge of SCL after the ACK response to IBI.
- T_C2 counter
  When the ICASTCCR.ATCE bit is 1, the T_C2 counter starts counting on the first rising edge of SCL after the ACK response to IBI. The count value of the T_C2 counter is captured on the first rising edge of SCL after the T bit of the Mandatory Byte.

When the ICTSIR2.AM0E bit is 1 and the ITS bit in the Command Descriptor used to issue an IBI is 1, the captured values of the T_C1 and T_C2 counters are transmitted in the following order: T_C1 lower byte, T_C1 higher byte, and T_C2, following the Mandatory Byte. In this case, set the DATA_LENGTH field in the Command Descriptor to a value that is 3 bytes longer than the number of data to be transmitted.
When the T_C1 counter or T_C2 counter overflows, the captured values become FFFFh and FFh, respectively, and the ICTSIR2.OVF flag becomes 1.

43.5.14.3 Async Mode 1 (Asynchronous Advanced Mode)

When performing timing control in Async Mode 1, set the ICASTCR register as necessary.

(1) I3C Controller

Async Mode 1 uses a 32-bit C_REF counter, a 32-bit C_SyncCNT counter, and a 16-bit C_C2 counter.

- C_REF counter
  When the ICASTCCR.ATCE bit is set to 1, the C_REF counter starts counting. The count value of the C_REF

counter is captured on the first rising edge of SCL after the ACK response to IBI.

- C_SyncCNT counter
  When the ICASTCCR.ATCE bit is set to 1, the C_SyncCNT counter starts counting. The count value of the C_SyncCNT counter is captured each time an aBE (additional Bus Event: the falling edge of SCL after a START condition) is detected.
- C_C2 counter
  When the ICASTCCR.ATCE bit is 1, the C_C2 counter starts counting on the first rising edge of SCL after the ACK response to IBI. The count value of the C_C2 counter is captured on the first rising edge of SCL after the T bit of the Mandatory Byte.

When an IBI is received from a Target whose corresponding ICTDATRm.IBITSE bit (m = 0 to 7) is 1, the captured values of the C_REF counter and C_C2 counter are each stored as 64-bit data after the IBI Status Descriptor. The Mandatory Byte and subsequent data are stored after that. The value of the DATA_LENGTH field in the IBI Status Descriptor is increased by 16.
The captured value of the C_SyncCNT counter is stored in the ICCSCCR register. When the ICASTCR.ABEOE bit is 1, an additional bus event is output each time an aBE is detected. Use this event signal as a trigger to read the captured value of the C_SyncCNT counter from the ICCSCCR register and store it in memory.

(2) I3C Target

Async Mode 1 uses a 16-bit T_C1 counter, an 8-bit aBE_TICK counter, and an 8-bit T_C2 counter.

- T_C1 counter
  When the trigger specified by the ICASTCR.ATTRGS bit is input while the ICASTCCR.ATCE bit is set to 1, the T_C1 counter starts counting. The count value of the T_C1 counter is captured at the first aBE.
- aBE_TICK counter
  When the ICASTCCR.ATCE bit is 1, the aBE_TICK counter is incremented each time aBE is detected. The count value of the aBE_TICK counter is captured on the first rising edge of SCL after the ACK response to IBI.
  The counter value is cleared at the first aBE after the T_C1 counter trigger is input.
- T_C2 counter
  When the ICASTCCR.ATCE bit is 1, the T_C2 counter starts counting on the first rising edge of SCL after the ACK response to IBI. The count value of the T_C2 counter is captured on the first rising edge of SCL after the T bit of the Mandatory Byte.

When the ICTSIR2.AM1E bit is 1 and the ITS bit in the Command Descriptor used to issue an IBI is 1, the captured values of the T_C1, T_C2, and aBE_TICK counters are transmitted in the following order: T_C1 lower byte, T_C1 higher byte, T_C2, and aBE_TICK, following the Mandatory Byte. In this case, set the DATA_LENGTH field in the Command Descriptor to a value that is 4 bytes longer than the number of data to be transmitted.
When the T_C1 counter or T_C2 counter overflows, the captured values become FFFFh and FFh, respectively, and the ICTSIR2.OVF flag becomes 1.

43.5.15 Common Command Codes (CCC) (I3C Protocol Mode)

Command Code E0h to FEh Vendor Extension-Direct CCCs defined are not supported.
The MIPI reserved area and Vendor Extension area of Command Code are not supported.
Do not use an unsupported CCC when using this module with I3C Target.
If the I3C Controller must use an unsupported CCC, use the added CCC after using ENTASx CCC to put this module to Sleep mode.

43.5.16 Error Detection (I3C Protocol Mode)

43.5.16.1 SDR Error Detection and Recovery Methods for I3C Target Devices

The seven error types summarized in Table 43.12 are supported for all I3C Target devices. Each error type is further explained below the table.

Table 43.12 SDR Target Error Types

Error Type	Description	Error Detection Method	Error Recovery Method
TE0	Broadcast Address/W (= 7Eh/W) or Dynamic Address/RW	Detect any of the following: 3Eh/W 5Eh/W 6Eh/W 76h/W 7Ah/W 7Ch/W 7Fh/W 7Eh/R	Enable HDR Exit Detector and ignore all other patterns
TE1	CCC code	Parity check, using T-Bit	Enable HDR Exit Detector and neglect other patterns
TE2	Write data	Parity check, using T-Bit	Enable STOP condition detector and neglect other patterns
TE3	Assigned address during Dynamic Address arbitration	Parity check, using PAR Bit	Generate NACK (after PAR), then wait for another repeated START condition and 7Eh/R to re-transmit the Provisioned ID
TE4	7Eh/R after Sr during Dynamic Address arbitration	Detect any value other than 7Eh/R after Sr during Dynamic Address Arbitration	Generate NACK (after 7Eh/R), then enable STOP condition detector and ignore all other patterns
TE5	Transaction after detecting CCC	Detect illegally formatted CCC	Generate NACK (after Target Address), then enable STOP condition detector and ignore all other patterns
TE6 (optional)	Monitoring error	Target detects (through monitoring) that transmitted Data differs from what it intended to transmit (Does not apply during Dynamic Address arbitration)	Stop the transmission, then enable STOP condition detector and ignore all other patterns

43.5.16.2 SDR Error Detection and Recovery Methods for I3C Controller Devices

The two error types summarized in Table 43.13 are supported for all I3C Controller devices. Each error type is further explained below the table.

Table 43.13 SDR Controller Error Types

Error Type	Description	Error Detection Method	Error Recovery Method
CE0	Transaction after sending CCC	Detect illegally formatted CCC	Stop the transmission, then send STOP condition and retry the transmission.
CE1 (optional)	Monitoring error	Controller detects (through monitoring) transmitted data different from what it intended to transmit (does not apply during Dynamic Address arbitration)	Stop the transmission, then send STOP condition and retry the transmission.
CE2	No response to Broadcast Address (7Eh)	Controller detects NACK after Broadcast Address (7Eh) transmission	Upon detection of NACK, Controller transmits HDR Exit Pattern followed by STOP condition

43.5.16.3 Timeout Detection

RI3C includes a timeout function for detecting when the SCL line has been stuck longer than the predetermined time. RI3C can detect an abnormal bus state by monitoring that the SCL line is stuck low or high for a predetermined time. The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, RI3C can detect the timeout and report the bus hung state.

This timeout function is enabled when ICSER.TMOE = 1. It detects a hung state that the SCL line is stuck low or high during the following conditions: (When ICTOR.TMOM[1:0] = 00b)

- The bus is busy (ICBSR.BFREE = 0) in Controller mode (ICMMR.ACF = 1).
- The bus is busy (ICBSR.BFREE = 0) in Target mode (ICMMR.ACF = 0).

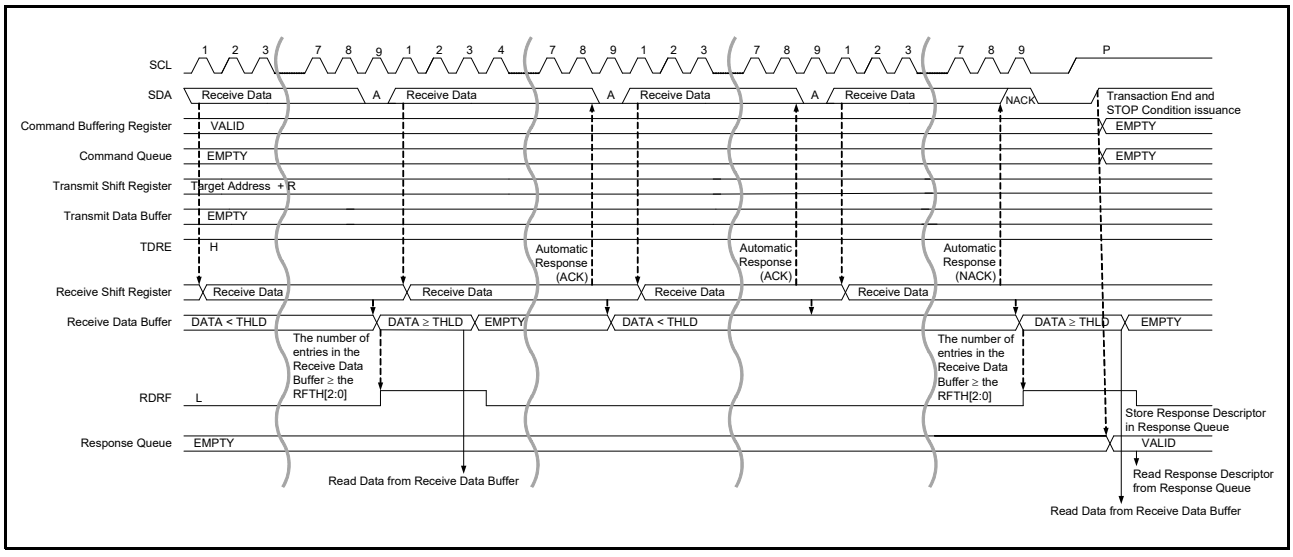


Figure 43.118 Timeout Detection (TMOE, TMOS[1:0], TMOH, and TMOL bits)

43.5.16.4 Resume Operation

RI3C enters the Halt state as a result of any type of error occurring in a transfer.
The error type is indicated by the field ERR_STATUS in Response Descriptor or Receive Status Descriptor. After RI3C has entered the Halt state, the user must write the value 1 to the RESUME bit to resume operation. RI3C shall auto-clear the RESUME bit once it has initiated the next Command transfer or detected the START condition.

43.5.16.5 Abort Operation

When the ICCR.ABORT bit is set to 1, RI3C relinquish control of the bus before completing the currently issued transfer. In response to an Abort request, RI3C issues the STOP condition on the bus after the complete data byte is transmitted or received. After RI3C has aborted, the user shall clear the ICCR.ABORT bit to allow operation on the bus.

Note: For Read transaction, received data are stored in the receive data buffer even if the ICCR.ABORT bit is set to 1.

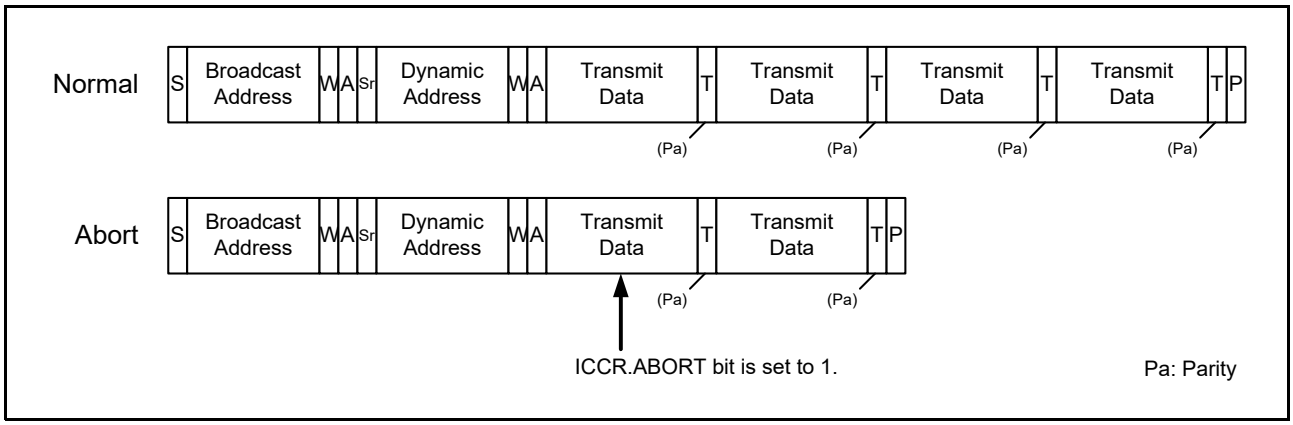


Figure 43.119 Abort Operation of SDR Write Transfer

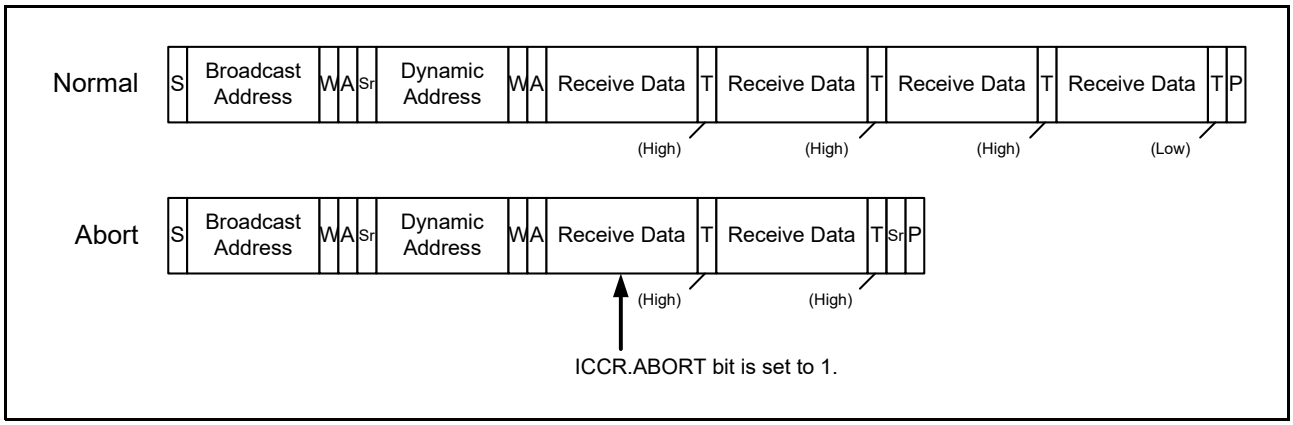


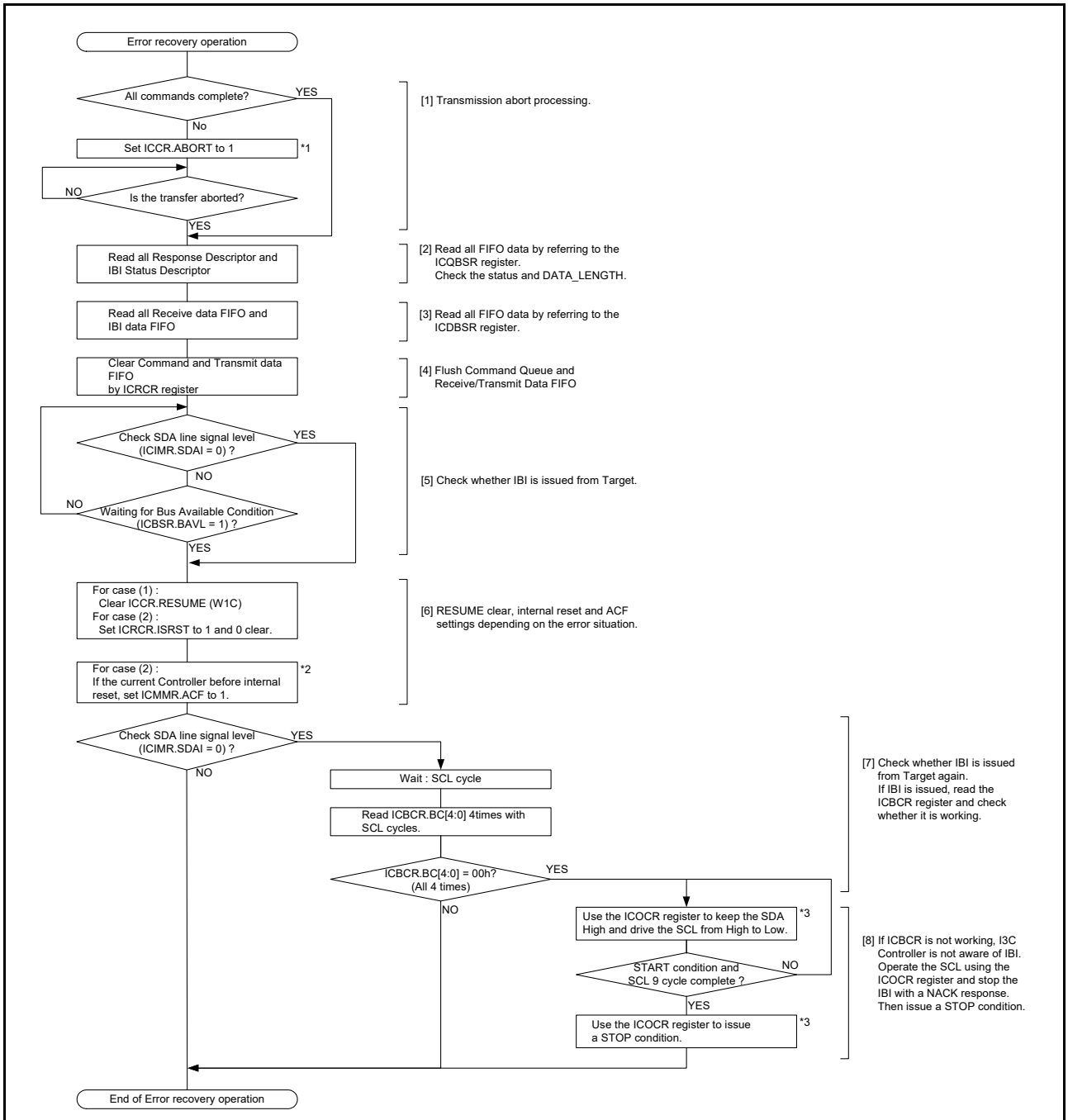
Figure 43.120 Abort Operation of SDR Read Transfer

43.5.16.6 Error Recovery Operation

When an error occurs, the ICISR.BERF, ICCSR.DTEF, and ICCSR.DTAF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled).
There is a possibility of communication error or internal module error.
The I3C Controller must perform an error recovery flow according to the following case:

- When DTEF is detected.

Figure 43.121 and Figure 43.122 show the error recovery flow.



Note 1. If the transfer is not aborted due to an error, set the ICCR.ABORT bit to 1 and then abort the transfer.
Note 2. When writing to the ACF flag, set ICMMR.WP bit to 1 at the same time.
Note 3. When writing to the SDAO and SCLO bit, set ICOCR.SOWP bit to 1 at the same time.

Figure 43.121 Example of Error Recovery Operation Flowchart for I3C Controller

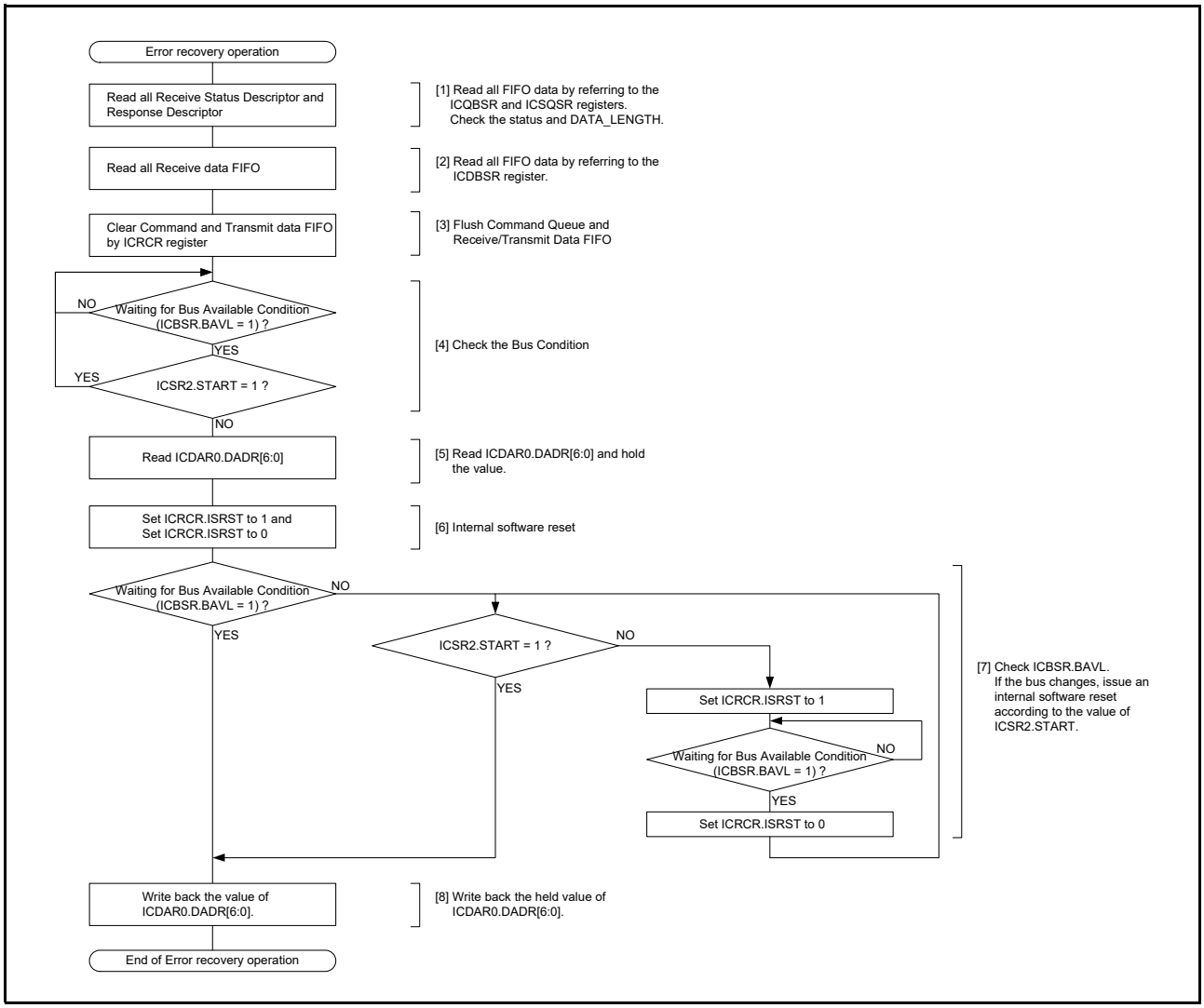


Figure 43.122 Example of Error Recovery Operation Flowchart for I3C Target

43.5.17 Wake-Up Function (I3C Protocol Mode)

The RI3C includes a wake-up function to return the MCU from low power consumption states. The wake-up interrupt source differs between controller mode and target mode.

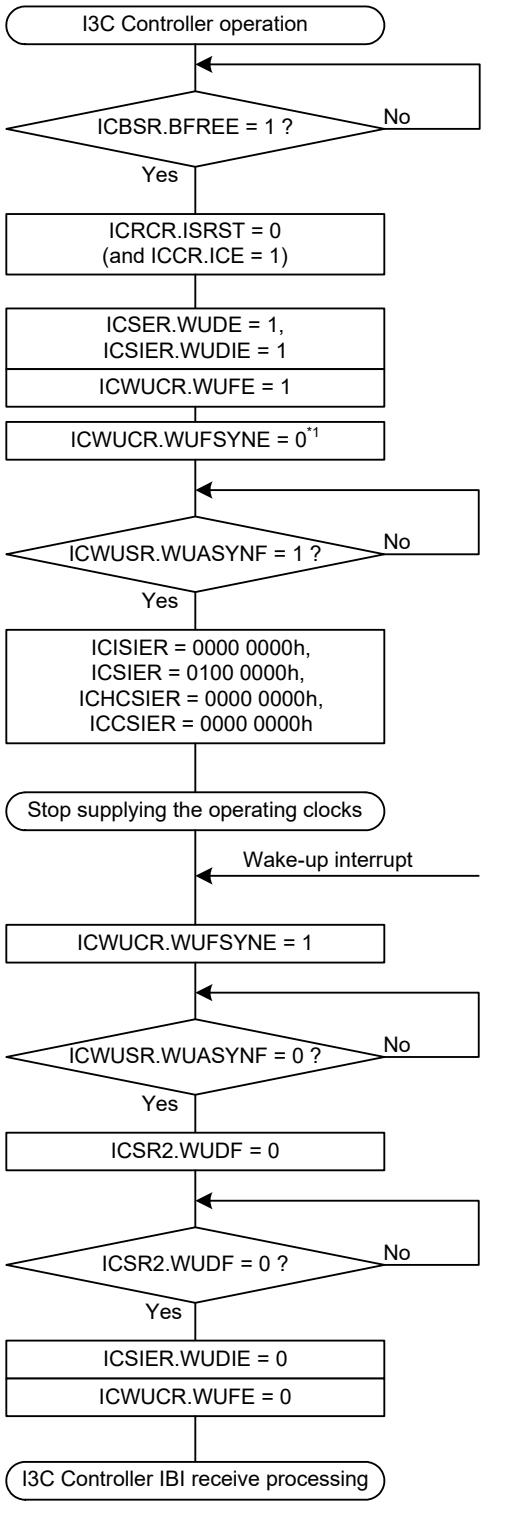
- I3C Controller mode: Low detection on the SDA line (IBI request from Target device)
- I3C Target mode: Detection of matches between the Target Address following Broadcast Address and its own Dynamic Address

43.5.17.1 I3C Controller Wake-Up Function

The wake-up detection circuit in I3C Controller mode operates as follows.

(1) Generates a wake-up interrupt when the SDA line is detected to be low.
(2) Holds the SCL line high until the wake-up function is disabled.
(3) When 1 is written to the ICWUCR.WUFSYNE bit, it sets the ICWUSR.WUASYNF flag to 0.
(4) Drives the SCL line low to complete the START condition.
(5) Supplies SCL and receives IBI from the Target device.

Set the ICWUCR.WUFE bit to 0 when the ICIMR.SDAI flag is 1 (SDA line is high).
If the low power consumption state is released by an interrupt other than a wake-up interrupt, disable the wake-up function as necessary.



[1] [1] Wait until the bus is free and in standby state.

[2] [2] If an internal reset is in progress, release the internal reset.

[3] [3] Enable the wake-up interrupt.

[4] [4] Enable the wake-up function.

[5] [5] Change from internal clock synchronous operation to asynchronous operation.*1

[6] [6] Disable interrupt requests other than WUI.

[7] [7] Stop supplying the operating clocks (reception continues)

[8] [8] The wake-up interrupt starts supplying the operating clocks.

[9] [9] Change from asynchronous operation to internal clock synchronous operation.

[10] [10] Clear the ICSR2.WUDF flag (confirm that the flag is cleared before returning from the interrupt processing).

[11] [11] Disable the wake-up interrupt.

[12] [12] Disable the wake-up function.

Note 1. When switching from internal clock synchronous operation to asynchronous operation, temporarily set the SDA pin to a general-purpose input port.

Figure 43.123 Example of Using I3C Controller Wake-Up

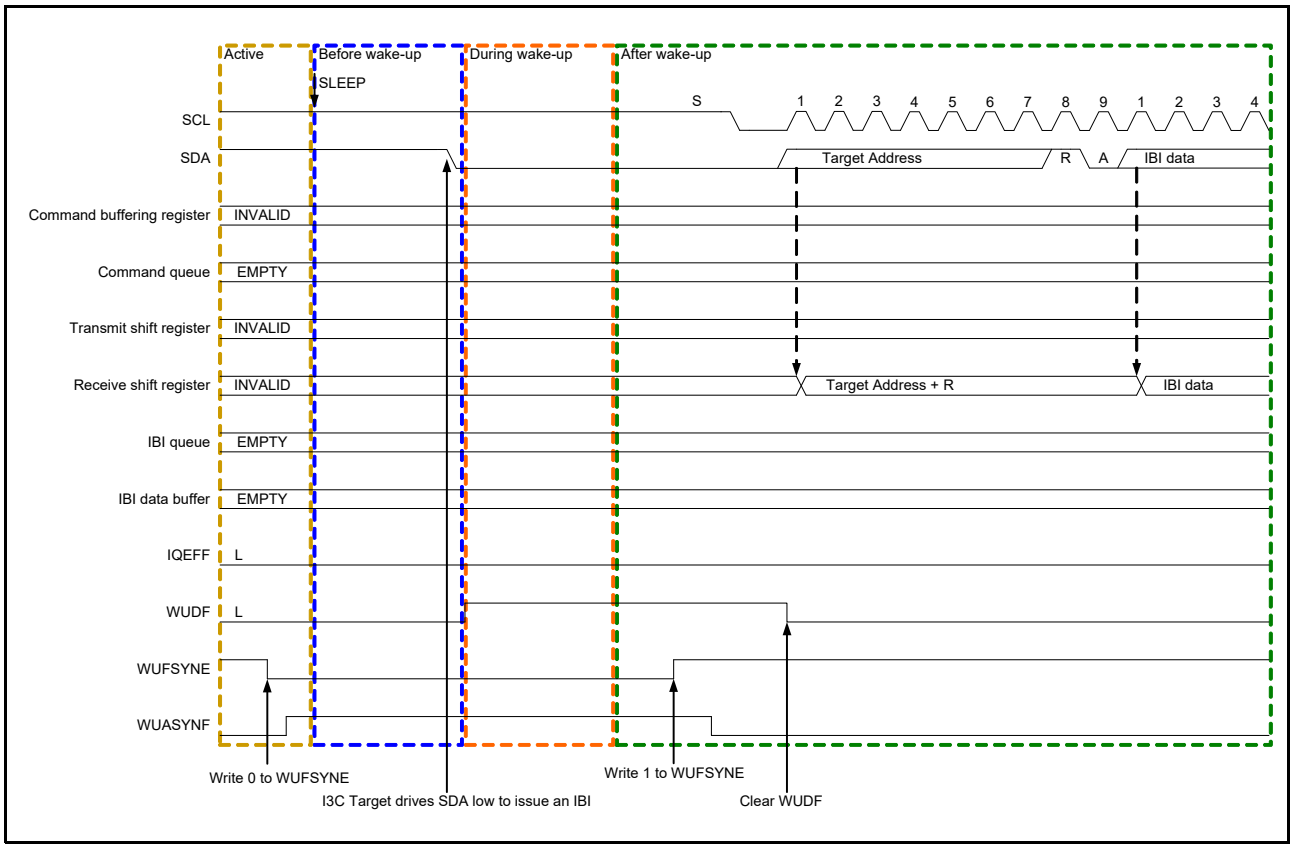


Figure 43.124 I3C Controller Wake-Up Timing

43.5.17.2 I3C Target Wake-Up Function

The wake-up detection circuit in I3C Target mode operates as follows.

(1) Generates an ACK when a Broadcast Address (7Eh) following a START or repeated START condition is detected.
(2) If the Target Address following the repeated START condition matches its own Dynamic Address, it responds with NACK and generates a wake-up interrupt.
(3) Responds with NACK to all data bytes until the wake-up function is disabled.
(4) When 1 is written to the ICWUCR.WUFSYNE bit, it sets the ICWUSR.WUASYNF flag to 0 at the next STOP condition.
(5) Returns to normal operation.

If the low power consumption state is released by an interrupt other than a wake-up interrupt, disable the wake-up function as necessary.

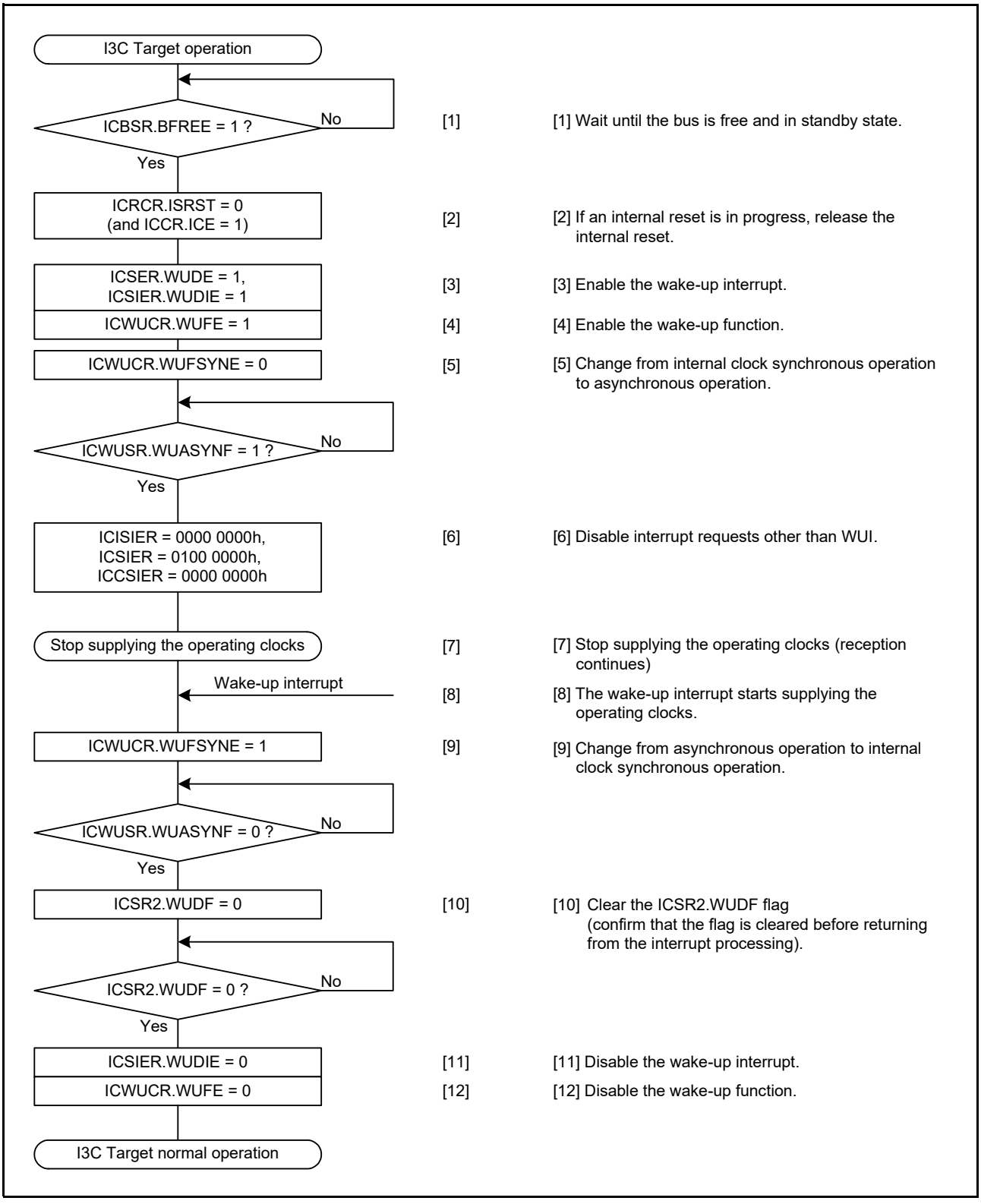


Figure 43.125 Example of Using I3C Target Wake-Up

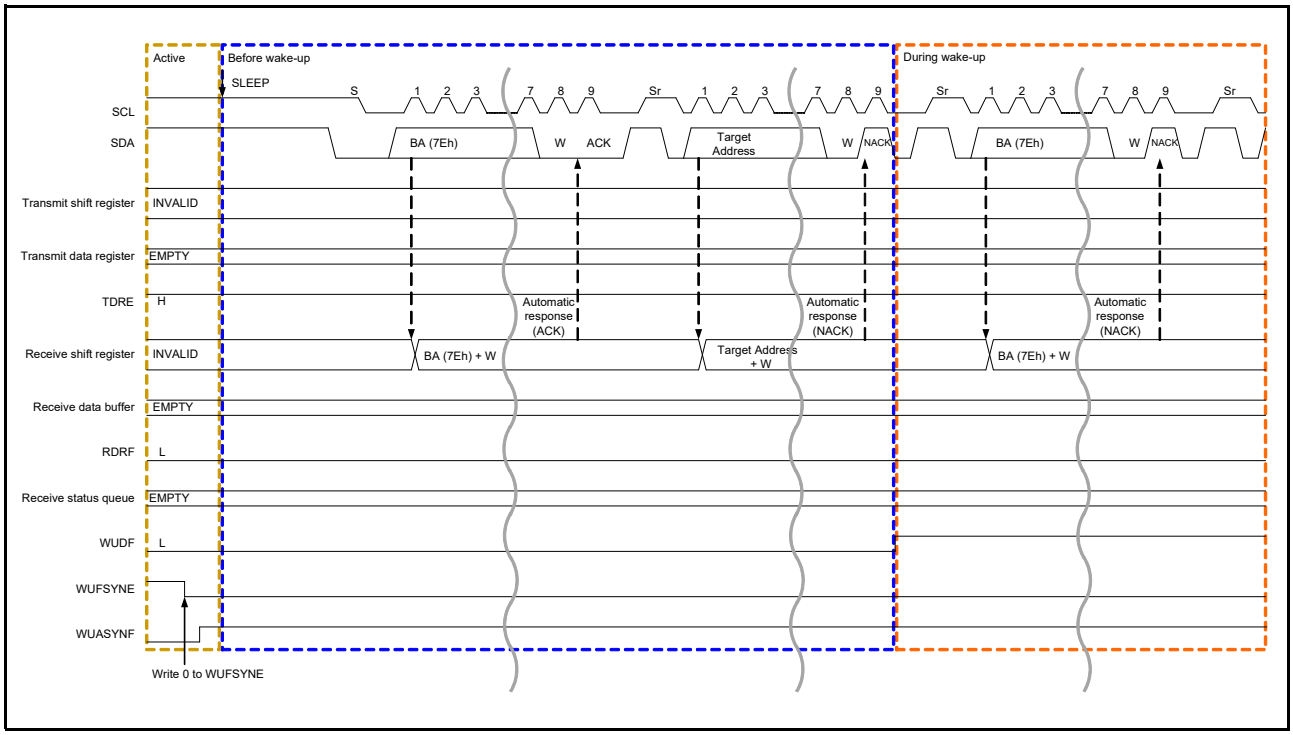


Figure 43.126 I3C Target Wake-Up Timing (1/2)

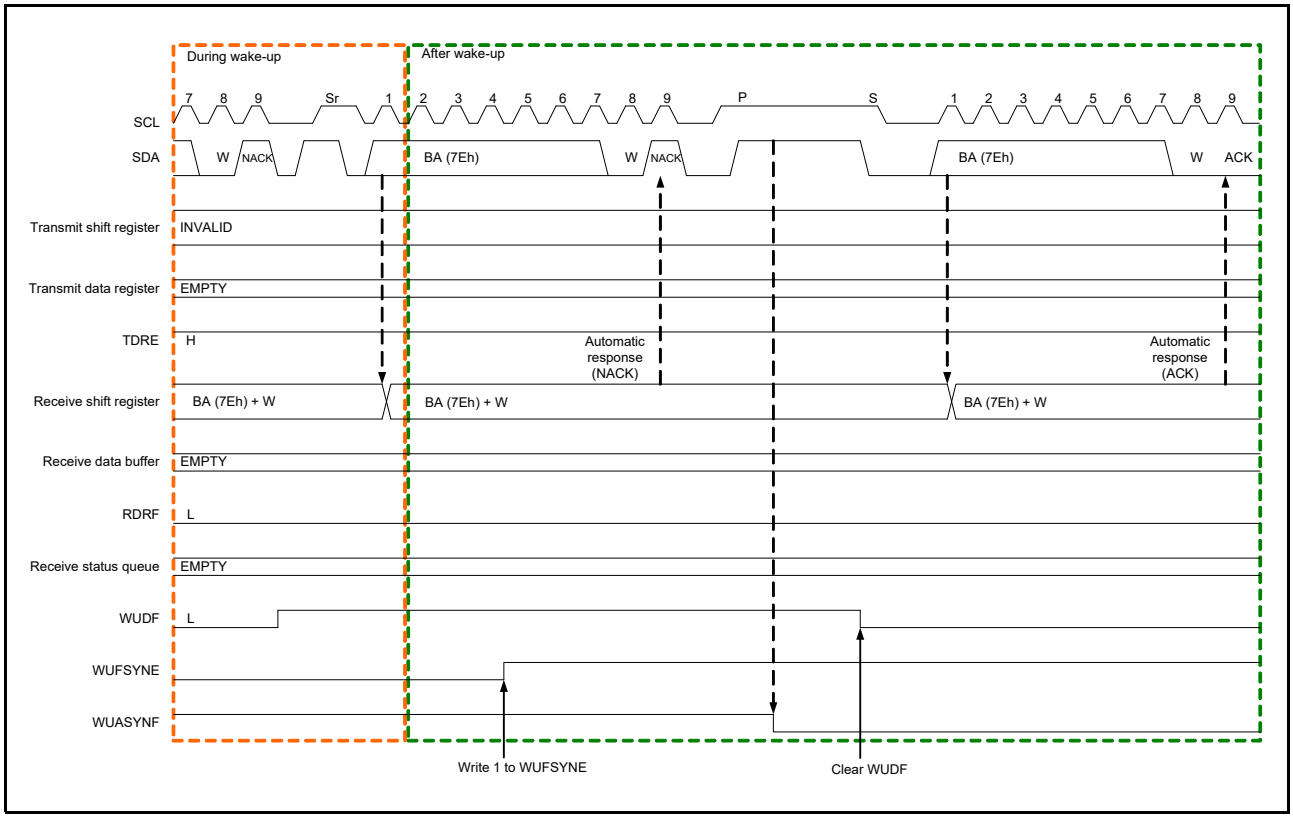


Figure 43.127 I3C Target Wake-Up Timing (2/2)

43.5.17.3 Notes on Using Wake-Up Function

When the wake-up function is enabled (ICWUCR.WUFE bit = 1), the timeout function cannot be used. Set the ICSER.TMOE bit to 0.

43.6 Interrupt Sources

(1) Interrupts in I3C Protocol Mode

The RI3C generates 12 types of interrupt requests, as shown in Table 43.14. The transmit data empty, receive data full, command queue empty, response queue full, IBI queue empty/full, and receive status queue full, high-priority transmit data empty, high-priority receive data full, high-priority command queue empty, and high-priority response queue full interrupt requests allow the DTC or DMAC to start data transfer.

Table 43.14 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Start DTC/DMA Transfer	Interrupt Generation Condition
TXI*1	Transmit data empty	ICCSR.TDRE	ICCSIER.TIE	Possible	TDRE = 1 and TIE = 1
RXI*2	Receive data full	ICCSR.RDRF	ICCSIER.RIE	Possible	RDRF = 1 and RIE = 1
CMDI*3	Command queue empty	ICCSR.CQEF	ICCSIER.CQEIE	Possible	CQEF = 1 and CQEIE = 1
RESPI*3	Response queue full	ICCSR.RQFF	ICCSIER.RQFIE	Possible	RQFF = 1 and RQFIE = 1
IBII*3	IBI queue empty/full	ICCSR.IQEFF	ICCSIER.IQEFIE	Possible	IQEFF = 1 and IQEFIE = 1
RCVI*3	Receive status queue full	ICCSR.SQFF	ICCSIER.SQFIE	Possible	SQFF = 1 and SQFIE = 1
HTXI*1	High-priority transmit data empty	ICHCSR.HTDRE	ICHCSIER.HTIE	Possible	HTDRE = 1 and HTIE = 1
HRXI*2	High-priority receive data full	ICHCSR.HRDRF	ICHCSIER.HRIE	Possible	HRDRF = 1 and HRIE = 1
HCMDI*3	High-priority command queue empty	ICHCSR.HCQEF	ICHCSIER.HCQEIE	Possible	HCQEF = 1 and HCQEIE = 1
HRESPI*3	High-priority response queue full	ICHCSR.HRQFF	ICHCSIER.HRQFIE	Possible	HRQFF = 1 and HRQFIE = 1
EEI	Communication error/ communication event	ICSR2.START	ICSIER.STIE	Not possible	START = 1 and STIE = 1
		ICSR2.STOP	ICSIER.SPIE		STOP = 1 and SPIE = 1
		ICSR2.HDRXDF	ICSIER.HDRXIE		HDRXDF = 1 and HDRXIE = 1
		ICSR2.TMOF	ICSIER.TMOIE		TMOF = 1 and TMOIE = 1
		ICISR.BERF	ICISIER.BERIE		BERF = 1 and BERIE = 1
		ICCSR.DTAF	ICCSIER.DTAIE		DTAF = 1 and DTAIE = 1
		ICCSR.DTEF	ICCSIER.DTEIE		DTEF = 1 and DTEIE = 1
		ICHCSR.HDTAF	ICHCSIER.HDTAIE		HDTAF = 1 and HDTAIE = 1
		ICHCSR.HDTEF	ICHCSIER.HDTEIE		HDTEF = 1 and HDTEIE = 1
WUI	Wake-up	ICSR2.WUDF	ICSIER.WUDIE	Not possible	WUDF = 1 and WUDIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.TDRE flag (a condition for TXI) is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.RDRF flag (a condition for RXI) is automatically set to 0 when data is read from the ICDR register.

Note 3. Because these interrupts are edge-detected interrupted, they do not require clearing. Furthermore, the each flag (a condition for interrupt) is automatically set to 0 when the corresponding queue is read or written to.

Clear each flag or mask the interrupt request during interrupt handling.

(2) Interrupts in I2C Protocol Mode

The RI3C generates five types of interrupt requests, as shown in Table 43.15. The transmit data empty and receive data full interrupt requests allow the DTC or DMAC to start data transfer.

Table 43.15 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Start DTC/DMA Transfer	Interrupt Generation Condition
TXI*1	Transmit data empty	ICCSR.TDRE	ICCSIER.TIE	Possible	TDRE = 1 and TIE = 1
RXI*2	Receive data full	ICCSR.RDRF	ICCSIER.RIE	Possible	RDRF = 1 and RIE = 1
TEI*3	Transmission end	ICSR2.TEND	ICSIER.TEIE	Not possible	TEND = 1 and TEIE = 1
EEI	Communication error/ communication event	ICSR2.START	ICSIER.STIE	Not possible	START = 1 and STIE = 1
		ICSR2.STOP	ICSIER.SPIE		STOP = 1 and SPIE = 1
		ICSR2.NACKF	ICSIER.NAKIE		NACKF = 1 and NAKIE = 1
		ICSR2.AL	ICSIER.ALIE		AL = 1 and ALIE = 1
		ICSR2.TMOF	ICSIER.TMOIE		TMOF = 1 and TMOIE = 1
WUI	Wake-up	ICSR2.WUDF	ICSIER.WUDIE	Not possible	WUDF = 1 and WUDIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.TDRE flag (a condition for TXI) is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.RDRF flag (a condition for RXI) is automatically set to 0 when data is read from the ICDR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Clear each flag or mask the interrupt request during interrupt handling.

43.6.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI or RXI interrupt is satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output to the ICU but retained internally (up to one request can be retained internally for each source).

When the IR flag becomes 0, the retained interrupt request is output to the ICU. The retained interrupt request is automatically cleared under normal conditions of usage.

The retained interrupt request can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICCSIER register.

43.7 Event Link Function

The RI3C outputs the corresponding event signal to the event link controller (ELC) when any of the following events occur.

(1) I3C Protocol Mode

- Transmit data empty
- Receive data full
- Command queue empty
- Response queue full
- IBI queue empty/full
- Receive status queue full
- High-priority transmit data empty
- High-priority receive data full
- High-priority command queue empty
- High-priority response queue full
- Communication error or communication event
- Synchronous timing event
- C_REF counter overflow event
- C_REF capture event
- Additional bus event

(2) I²C Protocol Mode

- Transmit data empty
- Receive data full
- Transmission end
- Communication error/communication event

43.7.1 Interrupt Handling and Event Linking

Each of the RI3C interrupts has an enable bit that controls whether the interrupt is enabled or disabled. When an interrupt request source occurs, an interrupt request signal is output to the ICU if the corresponding interrupt enable bit is 1.
On the other hand, the event signal is transmitted via the ELC to other modules when the corresponding interrupt request source occurs, regardless of the interrupt enable bit.
For details on interrupt sources, refer to Table 43.14 and Table 43.15.

43.8 Reset Description

Table 43.16 Reset Control of Registers (1/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICMR	OMS	To be reset	To be reset	Retained
ICCR	ICE	To be reset	To be reset	Retained
	RESUME	To be reset	To be reset	To be reset
	ABORT	To be reset	To be reset	Retained
	BAEXITMD	To be reset	To be reset	Retained
	IBAINC	To be reset	To be reset	Retained
ICCAR	DAV	To be reset	To be reset	Retained
	DADR[6:0]	To be reset	To be reset	Retained
ICRCR	ISRST	To be reset	To be reset	Retained
	HRBRST	To be reset	To be reset	To be reset
	HTBRST	To be reset	To be reset	To be reset
	HRQRST	To be reset	To be reset	To be reset
	HCQRST	To be reset	To be reset	To be reset
	SQRST	To be reset	To be reset	To be reset
	IQRST	To be reset	To be reset	To be reset
	RBRST	To be reset	To be reset	To be reset
	TBRST	To be reset	To be reset	To be reset
	RQRST	To be reset	To be reset	To be reset
	CQRST	To be reset	To be reset	To be reset
	MRST	To be reset	Retained	Retained
ICMMR	WP	To be reset	To be reset	To be reset
	TRS	To be reset	To be reset	To be reset
	ACF	To be reset	To be reset	Retained/To be reset*1
ICISR	BERF	To be reset	To be reset	To be reset
ICISER	BERDE	To be reset	To be reset	Retained
ICISIER	BERIE	To be reset	To be reset	Retained
ICDCTIR	INDEX[4:0]	To be reset	To be reset	To be reset
ICINCR	RTIRN	To be reset	To be reset	Retained
	RCRRN	To be reset	To be reset	Retained
ICFER	HSME	To be reset	To be reset	Retained
	FMPE	To be reset	To be reset	Retained
	SMBS	To be reset	To be reset	Retained
	SCLE	To be reset	To be reset	Retained
	SALE	To be reset	To be reset	Retained
	NALE	To be reset	To be reset	Retained
	MALE	To be reset	To be reset	Retained
ICTCR	TA0DE	To be reset	To be reset	Retained
	TA1DE	To be reset	To be reset	Retained
	TA2DE	To be reset	To be reset	Retained

Table 43.16 Reset Control of Registers (2/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICSCR	SAR2E	To be reset	To be reset	Retained
	SAR1E	To be reset	To be reset	Retained
	SAR0E	To be reset	To be reset	Retained
	HOAE	To be reset	To be reset	Retained
	DIDE	To be reset	To be reset	Retained
	HSMCE	To be reset	To be reset	Retained
	GCAE	To be reset	To be reset	Retained
ICRCCR	CKS[2:0]	To be reset	To be reset	Retained
ICSBR	ODDBL	To be reset	To be reset	Retained
	PPHW[5:0]	To be reset	To be reset	Retained
	PPLW[5:0]	To be reset	To be reset	Retained
	ODHW[7:0]	To be reset	To be reset	Retained
	ODLW[7:0]	To be reset	To be reset	Retained
ICFBR	DBL	To be reset	To be reset	Retained
	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICEBR	PPHW[5:0]	To be reset	To be reset	Retained
	PPLW[5:0]	To be reset	To be reset	Retained
	ODHW[7:0]	To be reset	To be reset	Retained
	ODLW[7:0]	To be reset	To be reset	Retained
ICHBR	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICBFTR		To be reset	To be reset	Retained
ICBATR		To be reset	To be reset	Retained
ICBITR		To be reset	To be reset	Retained
ICOCR	DLCS	To be reset	To be reset	Retained
	SDDL[2:0]	To be reset	To be reset	Retained
	CLO	To be reset	To be reset	Retained
	SOWP	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	Retained
	SDAO	To be reset	To be reset	Retained
ICICR	ANFE	To be reset	To be reset	Retained
	NFE	To be reset	To be reset	Retained
	NF[3:0]	To be reset	To be reset	Retained
ICTOR	TMOM[1:0]	To be reset	To be reset	Retained
	TMOH	To be reset	To be reset	Retained
	TMOL	To be reset	To be reset	Retained
	TMOS[1:0]	To be reset	To be reset	Retained
ICWUCR	WUFE	To be reset	To be reset	Retained
	WUFSYNE	To be reset	To be reset	Retained
	WUANFE	To be reset	To be reset	Retained
	WUACKS	To be reset	To be reset	Retained

Table 43.16 Reset Control of Registers (3/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICACKR	ACKWP	To be reset	To be reset	To be reset
	ACKBT	To be reset	To be reset	To be reset
	ACKBR	To be reset	To be reset	To be reset
ICCSCR	WAITRE	To be reset	To be reset	Retained
	WAITAE	To be reset	To be reset	Retained
ICSTCR	APSE	To be reset	To be reset	Retained
	PBSE	To be reset	To be reset	Retained
	AASE	To be reset	To be reset	Retained
	STT[15:0]	To be reset	To be reset	Retained
ICTDLR	DLGTH[15:0]	To be reset	To be reset	Retained
ICSYTCR	STOE	To be reset	To be reset	Retained
ICASTCR	CDIV[7:0]	To be reset	To be reset	Retained
	ABEOE	To be reset	To be reset	Retained
	CREFOE	To be reset	To be reset	Retained
	ATTRGS	To be reset	To be reset	Retained
ICASTTR	ATSTRG	To be reset	To be reset	To be reset
ICASTCCR	ATCE	To be reset	To be reset	Retained
ICCGR	SP	To be reset	To be reset	To be reset
	RS	To be reset	To be reset	To be reset
	ST	To be reset	To be reset	To be reset
ICCQR		To be reset	To be reset	To be reset
ICRQR		To be reset	To be reset	To be reset
ICDR		To be reset	To be reset	To be reset
ICIQR		To be reset	To be reset	To be reset
ICSQR		To be reset	To be reset	To be reset
ICHCQR		To be reset	To be reset	To be reset
ICHRQR		To be reset	To be reset	To be reset
ICHDR		To be reset	To be reset	To be reset
ICQBTCR	IQTH[7:0]	To be reset	To be reset	Retained
	IDSS[7:0]	To be reset	To be reset	Retained
	RQTH[7:0]	To be reset	To be reset	Retained
	CQTH[7:0]	To be reset	To be reset	Retained
ICDBTCR	RSTH[2:0]	To be reset	To be reset	Retained
	TSTH[2:0]	To be reset	To be reset	Retained
	RFTH[2:0]	To be reset	To be reset	Retained
	TETH[2:0]	To be reset	To be reset	Retained
ICSQTCR	SQTH[7:0]	To be reset	To be reset	Retained
ICHQBTCR	HRQTH[7:0]	To be reset	To be reset	Retained
	HCQTH[7:0]	To be reset	To be reset	Retained
ICHDBTCR	HRSTH[2:0]	To be reset	To be reset	Retained
	HTSTH[2:0]	To be reset	To be reset	Retained
	HRFTH[2:0]	To be reset	To be reset	Retained
	HTETH[2:0]	To be reset	To be reset	Retained

Table 43.16 Reset Control of Registers (4/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICSR2	WUDF	To be reset	To be reset	Retained
	TMOF	To be reset	To be reset	To be reset
	AL	To be reset	To be reset	To be reset
	TEND	To be reset	To be reset	To be reset
	NACKF	To be reset	To be reset	To be reset
	HDRXDF	To be reset	To be reset	To be reset
	STOP	To be reset	To be reset	To be reset
	START	To be reset	To be reset	To be reset
ICSER	WUDE	To be reset	To be reset	Retained
	TMOE	To be reset	To be reset	Retained
	ALE	To be reset	To be reset	Retained
	TEDE	To be reset	To be reset	Retained
	NAKDE	To be reset	To be reset	Retained
	HDRXDE	To be reset	To be reset	Retained
	SPDE	To be reset	To be reset	Retained
	STDE	To be reset	To be reset	Retained
ICSIER	WUDIE	To be reset	To be reset	Retained
	TMOIE	To be reset	To be reset	Retained
	ALIE	To be reset	To be reset	Retained
	TEIE	To be reset	To be reset	Retained
	NAKIE	To be reset	To be reset	Retained
	HDRXIE	To be reset	To be reset	Retained
	SPIE	To be reset	To be reset	Retained
	STIE	To be reset	To be reset	Retained
ICCSR	SQFF	To be reset	To be reset	To be reset
	DTEF	To be reset	To be reset	To be reset
	DTAF	To be reset	To be reset	To be reset
	RQFF	To be reset	To be reset	To be reset
	CQEF	To be reset	To be reset	To be reset
	IQEFF	To be reset	To be reset	To be reset
	RDRF	To be reset	To be reset	To be reset
	TDRE	To be reset	To be reset	To be reset
ICCSER	SQFDE	To be reset	To be reset	Retained
	DTEDE	To be reset	To be reset	Retained
	DTADE	To be reset	To be reset	Retained
	RQFDE	To be reset	To be reset	Retained
	CQEDE	To be reset	To be reset	Retained
	IQEFDE	To be reset	To be reset	Retained
	RDE	To be reset	To be reset	Retained
	TDE	To be reset	To be reset	Retained

Table 43.16 Reset Control of Registers (5/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICCSIER	SQFIE	To be reset	To be reset	Retained
	DTEIE	To be reset	To be reset	Retained
	DTAIE	To be reset	To be reset	Retained
	RQFIE	To be reset	To be reset	Retained
	CQEIE	To be reset	To be reset	Retained
	IQEFIE	To be reset	To be reset	Retained
	RIE	To be reset	To be reset	Retained
	TIE	To be reset	To be reset	Retained
ICHCSR	HDTEF	To be reset	To be reset	Retained
	HDTAF	To be reset	To be reset	Retained
	HRQFF	To be reset	To be reset	To be reset
	HCQEF	To be reset	To be reset	To be reset
	HRDRF	To be reset	To be reset	To be reset
	HTDRE	To be reset	To be reset	To be reset
ICHCSER	HDTEDE	To be reset	To be reset	Retained
	HDTADE	To be reset	To be reset	Retained
	HRQFDE	To be reset	To be reset	Retained
	HCQEDE	To be reset	To be reset	Retained
	HRDE	To be reset	To be reset	Retained
	HTDE	To be reset	To be reset	Retained
ICHCSIER	HDTEIE	To be reset	To be reset	Retained
	HDTAIE	To be reset	To be reset	Retained
	HRQFIE	To be reset	To be reset	Retained
	HCQEIE	To be reset	To be reset	Retained
	HRIE	To be reset	To be reset	Retained
	HTIE	To be reset	To be reset	Retained
ICBSR	BIDL	To be reset	To be reset	Retained
	BAVL	To be reset	To be reset	Retained
	BFREE	To be reset	To be reset	Retained
ICSSR	AAS2	To be reset	To be reset	To be reset
	AAS1	To be reset	To be reset	To be reset
	AAS0	To be reset	To be reset	To be reset
	HOA	To be reset	To be reset	To be reset
	DID	To be reset	To be reset	To be reset
	HSMC	To be reset	To be reset	To be reset
	GCA	To be reset	To be reset	To be reset
ICWUSR	WUASYNF	To be reset	To be reset	Retained
ICCSCCR		To be reset	To be reset	To be reset

Table 43.16 Reset Control of Registers (6/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICTDATRm (m = 0 to 7)	TYPE	To be reset	To be reset	Retained
	NACKRC[1:0]	To be reset	To be reset	Retained
	DADR[7:0]	To be reset	To be reset	Retained
	IBITSE	To be reset	To be reset	Retained
	CRRRJ	To be reset	To be reset	Retained
	TIRRJ	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	SADR[6:0]	To be reset	To be reset	Retained
ICEDATR	TYPE	To be reset	To be reset	Retained
	NACKRC[1:0]	To be reset	To be reset	Retained
	DADR[7:0]	To be reset	To be reset	Retained
	SADR[6:0]	To be reset	To be reset	Retained
ICDARy (y = 0 to 2)	DADR[6:0]	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	SADR[9:0]	To be reset	To be reset	Retained
SARy (y = 0 to 2)	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
ICTDCTRm (m = 0 to 7)	ROLE[1:0]	To be reset	To be reset	Retained
	ADVCAP	To be reset	To be reset	Retained
	VTSUP	To be reset	To be reset	Retained
	OFLC	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	IBIRQC	To be reset	To be reset	Retained
	LIMIT	To be reset	To be reset	Retained
ICDCTR	ROLE[1:0]	To be reset	To be reset	Retained
	ADVCAP	To be reset	To be reset	Retained
	VTSUP	To be reset	To be reset	Retained
	OFLC	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	IBIRQC	To be reset	To be reset	Retained
	LIMIT	To be reset	To be reset	Retained
	DCR[7:0]	To be reset	To be reset	Retained
ICPIDLR		To be reset	To be reset	Retained
ICPIDHR		To be reset	To be reset	Retained
ICDAMRy (y = 0 to 2)	DAV	To be reset	To be reset	Retained
	SAV	To be reset	To be reset	Retained
	TADR[9:0]	To be reset	To be reset	Retained
SAMRy (y = 0 to 2)	SAV	To be reset	To be reset	Retained
	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
ICTEVR	ENCR	To be reset	To be reset	Retained
	ENINT	To be reset	To be reset	Retained

Table 43.16 Reset Control of Registers (7/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICASR	ENTAS3	To be reset	To be reset	Retained
	ENTAS2	To be reset	To be reset	Retained
	ENTAS1	To be reset	To be reset	Retained
	ENTAS0	To be reset	To be reset	Retained
ICMWLR	MWL[15:0]	To be reset	To be reset	Retained
ICMRLR	IBIPL[7:0]	To be reset	To be reset	Retained
	MRL[15:0]	To be reset	To be reset	Retained
ICTMR	TM[7:0]	To be reset	To be reset	Retained
ICDSR	VRSV[7:0]	To be reset	To be reset	Retained
	CAS[1:0]	To be reset	To be reset	Retained
	PERR	To be reset	To be reset	Retained
	PNDINT[3:0]	To be reset	To be reset	To be reset
ICMWSR	MSWDR[2:0]	To be reset	To be reset	Retained
ICMRSR	TSCO[2:0]	To be reset	To be reset	Retained
	MSRDR[2:0]	To be reset	To be reset	Retained
ICMTTR	MRTTE	To be reset	To be reset	Retained
	MRTT[23:0]	To be reset	To be reset	Retained
ICTSIR	INAC[7:0]	To be reset	To be reset	Retained
	FREQ[7:0]	To be reset	To be reset	Retained
	SAM1	To be reset	To be reset	Retained
	SAM0	To be reset	To be reset	Retained
	SSM	To be reset	To be reset	Retained
ICTSIR2	OVF	To be reset	To be reset	To be reset
	AM1E	To be reset	To be reset	Retained
	AM0E	To be reset	To be reset	Retained
	SME	To be reset	To be reset	Retained
ICBCR	BC[4:0]	To be reset	To be reset	To be reset
ICQBSR	ISC[4:0]	To be reset	To be reset	To be reset
	IQFL[7:0]	To be reset	To be reset	To be reset
	RQFL[7:0]	To be reset	To be reset	To be reset
	CQFL[7:0]	To be reset	To be reset	To be reset
ICDBSR	RBFL[7:0]	To be reset	To be reset	To be reset
	TBFL[7:0]	To be reset	To be reset	To be reset
ICSQSR	SQFL[7:0]	To be reset	To be reset	To be reset
ICHQBSR	HRQFL[7:0]	To be reset	To be reset	To be reset
	HCQFL[7:0]	To be reset	To be reset	To be reset
ICHDBSR	HRBFL[7:0]	To be reset	To be reset	To be reset
	HTBFL[7:0]	To be reset	To be reset	To be reset
ICIMR	SDAO	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	To be reset
	SDAI	To be reset	To be reset	Retained
	SCLI	To be reset	To be reset	Retained
ICCECR	CE2C[7:0]	To be reset	To be reset	To be reset
ICTC1CR		To be reset	To be reset	To be reset

Table 43.16 Reset Control of Registers (8/8)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICTC2CR		To be reset	To be reset	To be reset
ICCCR	CCE	To be reset	Retained	Retained

Note 1. “Retained” in I3C protocol mode, “To be reset” in I^2C protocol mode.

43.9 Usage Notes

43.9.1 Settings of the Module Stop Function

RI3C operation can be disabled or enabled by module stop control register B (MSTPCRB).
After a reset, the RI3C is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

44. High-Speed I2C-bus Interface (RIICHS)

This MCU has two-channel I2C-bus interfaces (RIICHS1, RIICHS2).
The RIICHS module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.

44.1 Overview

Table 44.1 lists the specifications of the RIICHS and Figure 44.1 shows a block diagram of the RIICHS. Table 44.2 lists the I/O pins of the RIICHS.

Table 44.1 RIICHS Specifications (1/2)

Item	Specification
Operation mode	Four • Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Communication format/ Transfer rate	• I2C-bus format - Standard-mode (Sm): 0 to 100 kbps - Fast-mode (Fm): 0 to 400 kbps - Fast-mode Plus (Fm+): 0 to 1 Mbps - High-speed mode (Hs-mode): 0 to 3.4 Mbps • SMBus format: 10 to 100 kbps • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Serial clock (SCL)	• For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	START, repeated START, and STOP conditions are automatically generated. START conditions (including repeated START conditions) and STOP conditions are detectable.
Address detection	• Slave address (static address) (max 3 address) • 7-bit and 10-bit address formats are supported. • General call address • Hs-mode master code • Device ID • Host address
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function (clock stretch)	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	It is possible to delay the change timing of the output of the transmission data bits and the acknowledgment bit relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the START condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the START condition while the bus is busy is detectable (to prevent the generating of double START conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise filter	• Analog noise filter • Digital noise filter

Table 44.1 RIICHS Specifications (2/2)

Item	Specification
Interrupt source	Five sources: • Transmit data empty (TXI) • Receive data full (RXI) • Transmission end (TEI) • Communication error or communication event (EEI) - START condition detection (including repeated START condition) - STOP condition detection - NACK detection - Arbitration lost - Timeout detection • Wake-up (WUI) - Slave address match detection
Event link function (output)	• Transmit data empty • Receive data full • Transmission end • Communication error or communication event
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attributes can be set for each channel.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

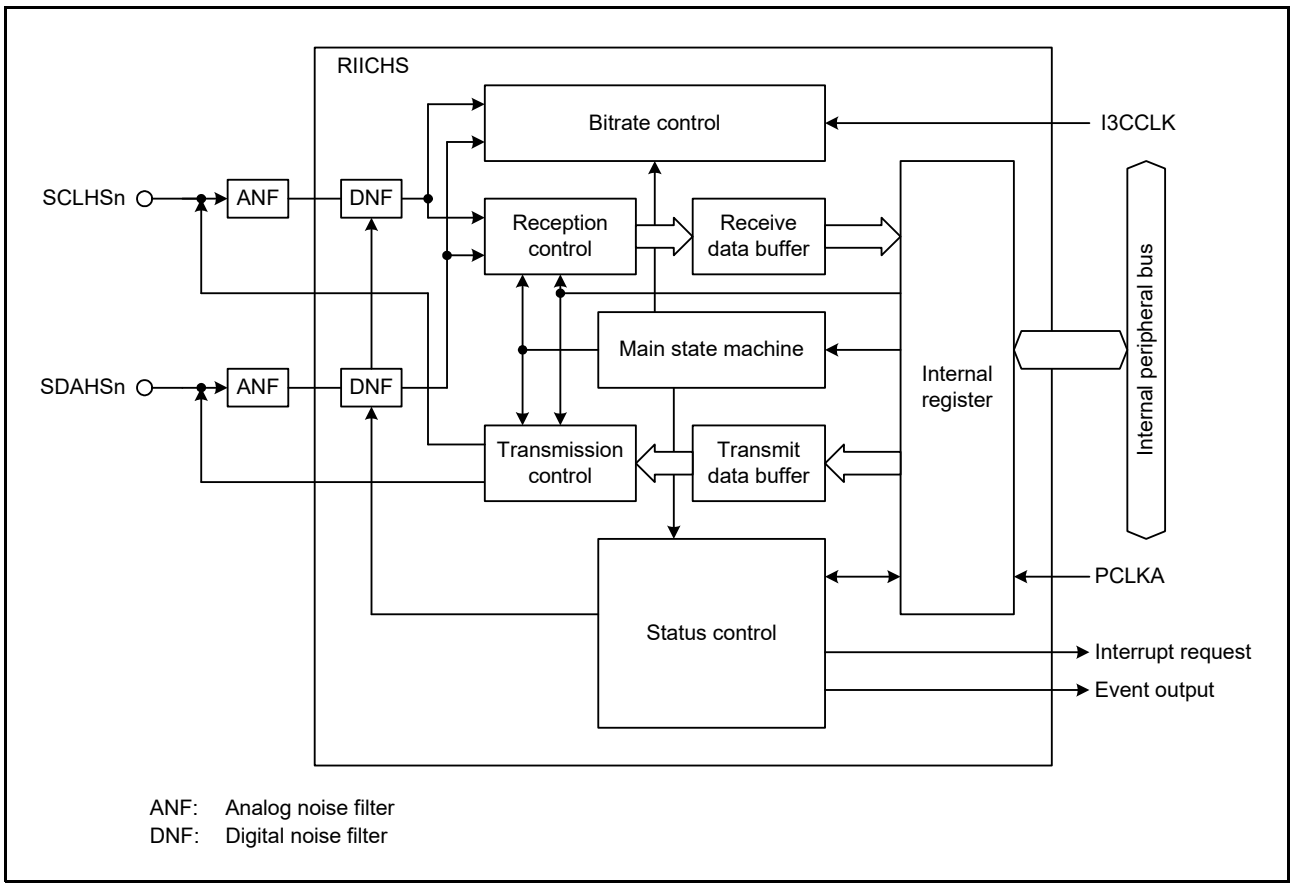


Figure 44.1 RIICHS Block Diagram (n = 1, 2)

Table 44.2 RIICHS Pin Configuration

Channel	Pin Name	I/O	Function
RIICHS1	SCLHS1	I/O	Serial clock I/O pin
	SDAHS1	I/O	Serial data I/O pin
RIICHS2	SCLHS2	I/O	Serial clock I/O pin
	SDAHS2	I/O	Serial data I/O pin

44.2 Register Descriptions

44.2.1 Control Register (ICCR)

Address(es): RIICHS1.ICCR 8745 F414h, RIICHS2.ICCR 8745 F814h,
RIICHS1_NS.ICCR 0745 F414h, RIICHS2_NS.ICCR 0745 F814h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	ICE	Bus Interface Enable	0: The I2C-bus operation is disabled. (SCLHSn and SDAHSn pins in inactive state) 1: The I2C-bus operation is enabled. (SCLHSn and SDAHSn pins in active state)	R/W

ICE Bit (Bus Interface Enable)

Enables or disables the operation on the I2C-bus.

When using RIICHS, set this bit to 1. When this bit is 1, the SCLHSn and SDAHSn pins are active.

Set this bit to 0 when RIICHS is not used. When the ICE bit is 0, the SCLHSn and SDAHSn pins are inactive.

When this bit is read as 0, it indicates that I2C-bus operation is prohibited.

44.2.2 Reset Control Register (ICRCR)

Address(es): RIICHS1.ICRCR 8745 F420h, RIICHS2.ICRCR 8745 F820h, RIICHS1_NS.ICRCR 0745 F420h, RIICHS2_NS.ICRCR 0745 F820h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRST	Module Reset*1	Writing 1 to this bit resets all registers and internal circuits. This bit automatically returns to 0 when the reset is complete.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ISRST	Internal Status Reset	0: Some registers and internal circuits are released from reset. 1: Some registers and internal circuits are reset.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not write any value to this register when this bit is 1.

ISRST Bit (Internal Status Reset)

When this bit is set to 1, some registers and internal circuits are reset. Refer to section 44.7, Reset Description for detail on the registers.

Set this bit to 0 to release from the reset.

44.2.3 Operating Mode Monitor Register (ICMMR)

Address(es): RIICHS1.ICMMR 8745 F424h, RIICHS2.ICMMR 8745 F824h,
RIICHS1_NS.ICMMR 0745 F424h, RIICHS2_NS.ICMMR 0745 F824h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MSWP	—	—	TRS	—	MST	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	MST	Master/Slave Mode Flag*1	0: Slave mode 1: Master mode	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	TRS	Transmit/Receive Mode Flag	0: Receive mode 1: Transmit mode	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	MSWP	MST Write Protect	0: Writing to the MST flag is disabled. 1: Writing to the MST flag is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When setting a value to the MST flag, set the MSWP bit to 1 at the same time.

MST Flag (Master/Slave Mode Flag)

This flag indicates master or slave mode.

The RIICHS is in slave mode when the MST flag is set to 0 and is in master mode when the flag is set to 1. Combination of this flag and the TRS flag indicates the operating mode of the RIICHS.

The value of the MST flag is automatically changed to 1 for master mode or 0 for slave mode by generating of a START condition and generating or detection of a STOP condition, etc. Although writing to the MST flag is possible when the MSWP bit is set to 1, writing to this flag is not necessary during normal usage.

[Setting conditions]

- When 1 is written to the MSWP bit and the MST flag at the same time.
- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).

[Clearing conditions]

- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.
- When 1 is written to the MSWP bit and 0 is written to the MST flag at the same time.
- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).

TRS Flag (Transmit/Receive Mode Flag)

This flag indicates transmit or receive mode.

The RIICHS is in receive mode when the TRS flag is set to 0 and is in transmit mode when the bit is set to 1.

Combination of this flag and the MST flag indicates the operating mode of the RIICHS.
The value of TRS flag is automatically changed to 1 for transmit mode or 0 for receive mode by generating or detection of a START condition and setting of the R/W# bit.
[Setting conditions]

- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).
- When a repeated START condition is generated normally according to the repeated START condition generation request (when a repeated START condition is detected with the ICCGR.RS bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in the ICSCR register, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended.
- In slave mode, a match between the received address and the address enabled in the ICSCR register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a repeated START condition is detected (a START condition is detected with ICBSR.BFREE flag = 0 and MST flag = 0).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to reset the RIICHS or the internal status.

MSWP Bit (MST Write Protect)

This bit is used to rewrite the MST flag.
When setting a value to the MST flag, set this bit to 1 at the same time.
This bit is automatically set to 0. The read value is 0.

44.2.4 Function Enable Register (ICFER)

Address(es): RIICHS1.ICFER 8745 F460h, RIICHS2.ICFER 8745 F860h, RIICHS1_NS.ICFER 0745 F460h, RIICHS2_NS.ICFER 0745 F860h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HSME	FMPE	—	SMBS	—	—	—	SCLE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICMMR.MST and ICMMR.TRS flags automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICMMR.MST and ICMMR.TRS flags automatically when arbitration is lost.)	R/W
b1	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b2	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b11 to b9	—	Reserved	These bits are read as 0 and cannot be modified.	R
b12	SMBS	SMBus/I2C-bus Selection	0: The I2C-bus select 1: The SMBus select	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	FMPE	Fast-mode Plus Enable	0: No Fm+ slope control circuit is used for the SCLHSn pin and SDAHSn pin. 1: An Fm+ slope control circuit is used for the SCLHSn pin and SDAHSn pin.	R/W
b15	HSME	Hs-mode Enable	0: High speed mode is disabled. 1: High speed mode is enabled.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit uses to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit uses to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit uses to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

SCLE Bit (SCL Synchronization Enable)

This bit uses to specify whether to synchronize the SCL output with the SCL input. Normally, set this bit to 1.
When the SCLE bit set to 0 (SCL synchronization is disabled), the RIICHS does not synchronize the SCL output with the SCL input. In this setting, the RIICHS outputs the clock with the transfer rate set in the ICBRH and ICBRL registers regardless of the SCLHSn line state. For this reason, if the load of the I2C bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not, it also affects the generation of a START condition, repeated START condition, and STOP condition, and the continuous output of additional SCL.
This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE Bit (Fast-mode Plus Enable)

This bit uses to specify whether to use a slope control circuit for Fast-mode Plus (Fm+).
When this bit set to 1, a slope control circuit conforming to the Fast-mode Plus (Fm+) slope control standard (tof) of the I2C-bus is selected. When this bit set to 0, a slope control circuit conforming to the Standard-mode (Sm) and Fast-mode (Fm) slope control standard (tof) of the I2C-bus is selected.
Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus (Fm+)) of the I2C-bus standard. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps (Sm), up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps). When performing Hs-mode communication, set this bit according to the mode when sending the Hs-mode master code.

HSME Bit (Hs-mode Enable)

This bit is used for communicating in Hs-mode.
When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.
After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from repeated START condition after receiving the NACK response.
It communicates at the bit rate set in the ICFBR register until the NACK response, and automatically switches from repeated START condition issuance after receiving the NACK response to the bit rate set in the ICHBR register.
Hs-mode continues until a STOP condition is detected.
When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in the ICFBR register.

Note: When this bit is set to 1, the ICSR2.NACKF flag will not be set even if a NACK response is received after sending the Hs-mode master code.

44.2.5 Slave Mode Control Register (ICSCR)

Address(es): RIICHS1.ICSCR 8745 F464h, RIICHS2.ICSCR 8745 F864h, RIICHS1_NS.ICSCR 0745 F464h, RIICHS2_NS.ICSCR 0745 F864h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	—	—	—	—	—	—	—	DIDE	HSMCE	—	—	—	—	GCAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GCAE	General Call Address Detection Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMCE	Hs-mode Master Code Detection Enable	0: Hs-mode master code detection is disabled. 1: Hs-mode master code detection is enabled.	R/W
b6	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	HOAE	Host Address Detection Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W
b16	SAR0E	Slave Address 0 Detection Enable	0: The value of the SAMR0.SVA[9:0] bits is disabled. 1: The value of the SAMR0.SVA[9:0] bits is enabled.	R/W
b17	SAR1E	Slave Address 1 Detection Enable	0: The value of the SAMR1.SVA[9:0] bits is disabled. 1: The value of the SAMR1.SVA[9:0] bits is enabled.	R/W
b18	SAR2E	Slave Address 2 Detection Enable	0: The value of the SAMR2.SVA[9:0] bits is disabled. 1: The value of the SAMR2.SVA[9:0] bits is enabled.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

GCAE Bit (General Call Address Detection Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIICHS recognizes the received slave address as the general call address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE Bit (Hs-mode Master Code Detection Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (0000 1XXXb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, RIICHS recognizes that the Hs-mode master code has been received.

The first byte after repeated START condition after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set in the SAMRy.SVA[9:0] bits (y = 0 to 2).

If the addresses match, the transmission/reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, RIICHS will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode master code.

When this bit is set to 1, set the ICCSCR.WAITAE bit to 0 and ICCSCR.WAITRE bit to 1.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a START condition or repeated START condition is detected.
When this bit is set to 1, if the received first frame matches the device ID, the RIICHS recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIICHS recognizes the second and the following frames as slave addresses and continues the receive operation.
When this bit is set to 0, the RIICHS ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.
For details on the device-ID address detection, refer to section 44.4.4.3, Device-ID Address Detection.

HOAE Bit (Host Address Detection Enable)

This bit uses to specify whether to ignore received host address (0001 000b) when the ICFER.SMBS bit is 1.
When this bit set to 1 while the ICFER.SMBS bit is 1, if the received slave address matches the host address, the RIICHS recognizes the received slave address as the host address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs the receive operation.
When the ICFER.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SARyE Bit (Slave Address y Detection Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in SAMRy.SVA[9:0] bits.
When this bit is set to 1, the slave address set in SAMRy.SVA[9:0] bits is enabled and is compared with the received slave address.
When this bit is set to 0, the slave address set in SAMRy.SVA[9:0] bits is disabled and is ignored even if it matches the received slave address.

44.2.6 Reference Clock Control Register (ICRCCR)

Address(es): RIICHS1.ICRCCR 8745 F470h, RIICHS2.ICRCCR 8745 F870h,
RIICHS1_NS.ICRCCR 0745 F470h, RIICHS2_NS.ICRCCR 0745 F870h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Internal Reference Clock Select	Selects the internal reference clock source (IICφ) for the RIICHS. b2 b0 0 0 0: I3CCLK/1 0 0 1: I3CCLK/2 0 1 0: I3CCLK/4 0 1 1: I3CCLK/8 1 0 0: I3CCLK/16 1 0 1: I3CCLK/32 1 1 0: I3CCLK/64 1 1 1: I3CCLK/128	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

44.2.7 F/S-Mode Bitrate Register (ICFBR)

Address(es): RIICHS1.ICFBR 8745 F474h, RIICHS2.ICFBR 8745 F874h,
RIICHS1_NS.ICFBR 0745 F474h, RIICHS2_NS.ICFBR 0745 F874h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DBL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HIGH[7:0]								LOW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b30 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	DBL	High/Low Width Doubling	0: Set the high and low widths to the set value. 1: Double the high and low widths of the set values.	R/W

This register is used to specify the bit rate other than Hs-mode.

LOW[7:0] Bits (Low Width Setting)

The LOW[7:0] bits are used to set the low period of SCL.
The RIICHS counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.
These bits also work to generate the data setup time for automatic SCL low-hold operation (refer to section 44.4.5, Automatic Low-Hold Function for SCL); when the RIICHS is used in slave mode, these bits need to be set to a value longer than the data setup time*1.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time ($t_{SU;DAT}$)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))
50 ns (up to 1 Mbps: Fast-mode Plus (Fm+))

HIGH[7:0] Bits (High Width Setting)

The HIGH[7:0] bits are used to set the high period of SCL.
These bits are valid in master mode. If the RIICHS is used only in I2C slave mode, these bits need not to set the high period.
The RIICHS counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = (LOW[7:0] + NF[3:0] +1) / IICφ
t_{HIGH} = (HIGH[7:0] + NF[3:0] +1) / IICφ
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I2C-bus standard from NXP Semiconductors.

Table 44.3 Example of the ICFBR Register Settings for Transfer Rate (I3CCLK = 120 MHz)

Transfer Rate (kbps)	CKS[2:0] Bits	HIGH[7:0] Bits	LOW[7:0] Bits	NFE Bit	NF[3:0] Bits
10	101b	122 (7Ah)	247 (F7h)	0	—
	101b	121 (79h)	246 (F6h)	1	0001b
50	011b	91 (5Bh)	188 (BCh)	0	—
	011b	90 (5Ah)	187 (BBh)	1	0001b
100	010b	108 (6Ch)	151 (97h)	0	—
	010b	107 (6Bh)	150 (96h)	1	0001b
400	000b	68 (44h)	158 (9Eh)	0	—
	000b	67 (43h)	157 (9Dh)	1	0001b
	001b	34 (22h)	78 (4Eh)	0	—
	001b	33 (21h)	77 (4Dh)	1	0001b
1000	000b	27 (1Bh)	63 (3Fh)	0	—
	000b	26 (1Ah)	62 (3Eh)	1	0001b
	001b	13 (0Dh)	31 (1Fh)	0	—
	001b	12 (0Ch)	30 (1Eh)	1	0001b

Rise time of SCL signal (t_r)
- 100 kbps or less (Sm): 1000 ns
- 400 kbps (Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

Fall time of SCL signal (t_f)
- 400 kbps (Sm/Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

44.2.8 Hs-Mode Bitrate Register (ICHBR)

Address(es): RIICHS1.ICHBR 8745 F478h, RIICHS2.ICHBR 8745 F878h,
RIICHS1_NS.ICHBR 0745 F478h, RIICHS2_NS.ICHBR 0745 F878h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HIGH[7:0]								LOW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to specify the bit rate in Hs-mode.

LOW[7:0] Bits (Low Width Setting)

The LOW[7:0] bits are used to set the low period of SCL.
The RIICHS counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.
These bits also work to generate the data setup time for automatic SCL low-hold operation (refer to section 44.4.5, Automatic Low-Hold Function for SCL); when the RIICHS is used in slave mode, these bits need to be set to a value longer than the data setup time*1.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time ($t_{SU;DAT}$)
10 ns (up to 3.4 Mbps: Hs-mode (Hs))

HIGH[7:0] Bits (High Width Setting)

The HIGH[7:0] bits are used to set the high period of SCL.
These bits are valid in master mode. If the RIICHS is used only in I2C slave mode, these bits need not to set the high period.
The RIICHS counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.
If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = (LOW[7:0] + NF[3:2] +1) / IICφ
t_{HIGH} = (HIGH[7:0] + NF[3:2] +1) / IICφ
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I2C-bus standard from NXP Semiconductors.

44.2.9 Bus Free Time Setting Register (ICBFTR)

Address(es): RIICHS1.ICBFTR 8745 F47Ch, RIICHS2.ICBFTR 8745 F87Ch, RIICHS1_NS.ICBFTR 0745 F47Ch, RIICHS2_NS.ICBFTR 0745 F87Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BFREE flag is set to 1. The upper 23 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using IICφ as the count source.
Set a value that satisfies the t_{BUF} (bus free time between a STOP and START condition) specified in the I2C-bus specification.

44.2.10 Output Signal Control Register (ICOCR)

Address(es): RIICHS1.ICOCR 8745 F488h, RIICHS2.ICOCR 8745 F888h,
RIICHS1_NS.ICOCR 0745 F488h, RIICHS2_NS.ICOCR 0745 F888h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	—	—	—	—	SDDL[2:0]			—	—	—	CLO	—	SOWP	SCLO	SDAO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAO	SDA Output Control	0: The RIICHS drives the SDAHSn pin low. 1: The RIICHS releases the SDAHSn pin. (SDAHSn line is pulled high by an external pull-up resistor.)	R/W
b1	SCLO	SCL Output Control	0: The RIICHS drives the SCLHSn pin low. 1: The RIICHS releases the SCLHSn pin. (SCLHSn line is pulled high by an external pull-up resistor.)	R/W
b2	SOWP	SCLO/SDAO Write Protect	0: Writing to the SCLO and SDAO bits is disabled. 1: Writing to the SCLO and SDAO bits is enabled (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (This bit automatically becomes 0 after output of one clock pulse.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	SDDL[2:0]	SDA Output Delay Setting	• When ICOCR.DLCS bit is 0 (IICφ) b10 b8 0 0 0: No output delay 0 0 1: 1 IICφ cycle 0 1 0: 2 IICφ cycles 0 1 1: 3 IICφ cycles 1 0 0: 4 IICφ cycles 1 0 1: 5 IICφ cycles 1 1 0: 6 IICφ cycles 1 1 1: 7 IICφ cycles • When ICOCR.DLCS bit is 1 (IICφ/2) b10 b8 0 0 0: No output delay 0 0 1: 1 or 2 IICφ cycles 0 1 0: 3 or 4 IICφ cycles 0 1 1: 5 or 6 IICφ cycles 1 0 0: 7 or 8 IICφ cycles 1 0 1: 9 or 10 IICφ cycles 1 1 0: 11 or 12 IICφ cycles 1 1 1: 13 or 14 IICφ cycles	R/W
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This setting (IICφ/2) is valid only when the SCL pin is low. When the SCL pin is high, this setting is invalid and the clock source becomes the internal reference clock (IICφ).

SDAO Bit (SDA Output Control) and SCLO Bit (SCL Output Control)

These bits are used to directly control the SDAHSn and SCLHSn signals output from the RIICHS.
When writing to these bits, also write 1 to the SOWP bit at the same time.
The result of setting these bits is input to the RIICHS via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.
Do not rewrite these bits during a START condition, STOP condition, repeated START condition, transmission, or reception. Operation after rewriting under the above conditions is not guaranteed.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 44.4.8.2, Additional SCL Output Function.

44.2.11 Input Signal Control Register (ICICR)

Address(es): RIICHS1.ICICR 8745 F48Ch, RIICHS2.ICICR 8745 F88Ch, RIICHS1_NS.ICICR 0745 F48Ch, RIICHS2_NS.ICICR 0745 F88Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ANFE	NFE	NF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	NF[3:0]	Noise Filter Stage Select	• Except in Hs-mode b3 b0 0 0 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 0 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 0 0 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). : : 1 1 1 1: Noise of up to 16 IICφ cycles is filtered out (16-stage filter). • In Hs-mode b3 b2 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to 4 IICφ cycles is filtered out (4-stage filter). The lower 2 bits (b1 and b0) are ignored.	R/W
b4	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b5	ANFE	Analog Noise Filter Enable	0: Analog noise filters are not used. 1: Analog noise filters are used.	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

NF[3:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.
For details on the digital noise filter function, refer to section 44.4.3, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLHSn line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – {1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) + 120 ns (pulse width suppressed by the analog noise filter, reference values)} or a greater value, the serial clock is regarded as noise by the noise filter function of the RIICHS, which may prevent the RIICHS from operating normally.

44.2.12 Timeout Control Register (ICTOR)

Address(es): RIICHS1.ICTOR 8745 F490h, RIICHS2.ICTOR 8745 F890h, RIICHS1_NS.ICTOR 0745 F490h, RIICHS2_NS.ICTOR 0745 F890h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TMOM[1:0]		TMOH	TMOL	—	—	TMOS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMOS[1:0]	Timeout Detection Time Select	b1 b0 0 0: Timeout counter is 16 bits (up to 65536 counts) 0 1: Timeout counter is 14 bits (up to 16384 counts) 1 0: Timeout counter is 8 bits (up to 256 counts) 1 1: Timeout counter is 6 bits (up to 64 counts)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TMOL	Timeout L Count Control	0: Count is disabled while the SCLHSn line is low. 1: Count is enabled while the SCLHSn line is low.	R/W
b5	TMOH	Timeout H Count Control	0: Count is disabled while the SCLHSn line is high. 1: Count is enabled while the SCLHSn line is high.	R/W
b7, b6	TMOM[1:0]	Timeout Detection Mode Select	b7 b6 0 0: Timeout function is enabled in any of the following conditions. • When the ICBSR.BFREE flag is 0 (bus busy) in master mode • When the ICBSR.BFREE flag is 0 (bus busy) when the received slave address matches its own slave address in slave mode • When the ICBSR.BFREE flag is 1 (bus free) while the ICCGR.ST is 1 (request to generate a START condition) 0 1: Timeout function is enabled during ICBSR.BFREE flag is 0 (bus busy). 1 0: Timeout function is enabled during ICBSR.BFREE flag is 1 (bus free). 1 1: Setting prohibited	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TMOS[1:0] Bits (Timeout Detection Time Select)

These bits are used to select for the timeout detection time when the timeout function is enabled (ICSER.TMOE bit = 1). While the SCLHSn line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.
For details on the timeout function, refer to section 44.4.8.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLHSn line is held low when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLHSn line is held high when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOM[1:0] Bits (Timeout Detection Mode Select)

These bits are used to set the timeout detection condition when the timeout function is enabled (ICSER.TMOE bit = 1).
When these bits are set to 00b, timeout detection is enabled during one of the following periods.

- When the bus is busy in master mode
- When RIICHS is the target of the communication in slave mode and the bus is busy
- When the bus is free while a START condition is requested

When these bits are set to 01b, timeout detection is enabled when the bus is busy.
When these bits are set to 10b, timeout detection is enabled when the bus is free.

44.2.13 Wake-Up Unit Control Register (ICWUCR)

Address(es): RIICHS1.ICWUCR 8745 F498h, RIICHS2.ICWUCR 8745 F898h, RIICHS1_NS.ICWUCR 0745 F498h, RIICHS2_NS.ICWUCR 0745 F898h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WUFE	WUFSYNE	—	WUANFE	—	—	—	WUACKS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	WUACKS	Wake-Up Acknowledge Select	The response mode can be selected in combination with the ICRCR.ISRST bit. Refer to Table 44.4 for details.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	WUANFE	Analog Noise Filter Enable	0: Analog noise filter for wake-up is disabled. 1: Analog noise filter for wake-up is enabled.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	WUFSYNE	Synchronous Operation Select	0: Asynchronous operation 1: Internal clock synchronous operation	R/W
b7	WUFE	Wake-Up Function Enable	0: Wake-Up function is disabled. 1: Wake-Up function is enabled.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

WUACKS Bit (Wake-Up Acknowledge Select)

This bit selects the ACK response when a wake-up condition is detected.

Table 44.4 Response Mode

ICRCR.ISRST Bit	WUACKS Bit	Response Mode	Description
0	0	Normal wake-up mode 1	Responds with an ACK and holds the SCL line low.
0	1	Normal wake-up mode 2*1	Holds the SCL line low after eighth clock pulse, and when the low power consumption mode is released, release the SCL line and responds with an ACK.
1	0	Command recovery mode	Responds with an ACK but does not hold the SCL line low.
1	1	EEP response mode	Responds with a NACK but does not hold the SCL line low.

Note 1. This mode cannot be used in Hs mode.

WUANFE Bit (Analog Noise Filter Enable)

This bit enables/disables the analog noise filter for the SCL and SDA input signals.

WUFSYNE Bit (Synchronous Operation Select)

This bit selects whether to synchronize the SCL and SDA lines with the internal clock or to use them as asynchronous. Asynchronous operation can be selected only when the WUFE bit is 1.

When this bit is set to 0, the ICWUSR.WUASYNF flag becomes 1 when bus-free is detected and the wake-up detection circuit is enabled. At this time, data reception is performed asynchronously with the internal clock.

When this bit is set to 1 after the wake-up condition is detected, the wake-up detection circuit is disabled and the normal reception circuit operates. If this bit is set to 1 when the wake-up condition is not detected, the wake-up detection circuit

is disabled when a STOP condition is detected.

44.2.14 Acknowledgment Bit Control Register (ICACKR)

Address(es): RIICHS1.ICACKR 8745 F4A0h, RIICHS2.ICACKR 8745 F8A0h,
RIICHS1_NS.ICACKR 0745 F4A0h, RIICHS2_NS.ICACKR 0745 F8A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACKWP	ACKBT	ACKBR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b1	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b2	ACKWP	ACKBT Write Protect	0: Writing to the ACKBT bit is disabled. 1: Writing to the ACKBT bit is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When setting a value to the ACKBT bit, set the ACKWP bit to 1 at the same time

ACKBR Bit (Received Acknowledge)

This bit is used to store the acknowledgment bit received from the receiver in transmit mode.

[Setting condition]

- When 1 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

ACKBT Bit (Transmit Acknowledge)

[Setting condition]

- When 1 is written to the ACKWP and ACKBT bits at the same time.

[Clearing conditions]

- When 1 is written to the ACKWP bit and 0 is written to the ACKBT bit at the same time.
- When a STOP condition generation is detected. (When a STOP condition is detected with the ICCGR.SP bit set to 1.)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

When setting a value to the ACKBT bit, set this bit to 1 at the same time.

This bit is automatically set to 0. The read value is 0.

44.2.15 Clock Stretch Control Register (ICCSCR)

Address(es): RIICHS1.ICCSCR 8745 F4A4h, RIICHS2.ICCSCR 8745 F8A4h,
RIICHS1_NS.ICCSCR 0745 F4A4h, RIICHS2_NS.ICCSCR 0745 F8A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WAITRE	WAITAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WAITAE	Acknowledgment Generation Wait Enable	0: The ICCSR.RDRF flag becomes 1 at the rising edge of the ninth SCL. (The SCLHSn line is not held low at the falling edge of the eighth clock pulse.) 1: The ICCSR.RDRF flag becomes 1 at the rising edge of the eighth SCL. (The SCLHSn line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ICACKR.ACKBT bit.	R/W
b1	WAITRE	Receive Data Read Wait Enable	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading ICDR register.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

WAITAE Bit (Acknowledgment Generation Wait Enable)

This bit is used to select the ICCSR.RDRF flag set timing in receive mode and also to select whether to hold the SCLHSn line low at the falling edge of the eighth SCL.

When the WAITAE bit is 0, the SCLHSn line is not held low at the falling edge of the eighth SCL, and the ICCSR.RDRF flag is set to 1 at the rising edge of the ninth SCL.

When the WAITAE bit is 1, the ICCSR.RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCLHSn line is held low at the falling edge of the eighth SCL. The low-hold of the SCLHSn line is released by writing a value to the ICACKR.ACKBT bit. After data is received with this setting, the SCLHSn line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ICACKR.ACKBT bit = 0) or NACK (ICACKR.ACKBT bit = 1) according to receive data.

WAITRE Bit (Receive Data Read Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the receive data buffer (ICDR register) is completely read each time single-byte data is received in receive mode.

When the WAITRE bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL low. When both the WAITAE and WAITRE bits are 0, continuous receive operation is enabled with the double buffer.

When the WAITRE bit is 1, the SCLHSn line is held low from the falling edge of the ninth SCL until the ICDR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: Read the ICDR register before setting the WAITRE bit to 0.

44.2.16 Condition Generation Request Register (ICCGR)

Address(es): RIICHS1.ICCGR 8745 F540h, RIICHS2.ICCGR 8745 F940h,
RIICHS1_NS.ICCGR 0745 F540h, RIICHS2_NS.ICCGR 0745 F940h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SP	RS	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ST	START Condition Generation Request	0: Do not request to generate a START condition. 1: Requests to generate a START condition.	R/W
b1	RS	Repeated START Condition Generation Request	0: Do not request to generate a repeated START condition. 1: Requests to generate a repeated START condition.	R/W
b2	SP	STOP Condition Generation Request	0: Do not request to generate a STOP condition. 1: Requests to generate a STOP condition.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

ST Bit (START Condition Generation Request)

This bit is used to request transition to master mode and generation of a START condition.
For details on the START condition generation, refer to section 44.4.7, START Condition/Repeated START Condition/STOP Condition Generating Function.
[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit.
- When a START condition has been generated (a START condition is detected).
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RIICHS reset or an internal reset.

Note: Set the ST bit to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 1 (bus free state).
Note that arbitration may be lost if the ST bit is set to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 0 (bus busy state).

RS Bit (Repeated START Condition Generation Request)

This bit is used to request that a repeated START condition be generated in master mode.
When this bit is set to 1 to request to generate a repeated START condition, a repeated START condition is generated when the BFREE flag is set to 0 (bus busy state) and the ICMMR.MST flag is set to 1 (master mode).
For details on the repeated START condition generation, refer to section 44.4.7, START Condition/Repeated START Condition/STOP Condition Generating Function.
[Setting condition]

- When 1 is written to the RS bit with the ICBSR.BFREE flag set to 0.

[Clearing conditions]

- When 0 is written to the RS bit.
- When a repeated START condition has been generated. (a repeated START condition is detected.)
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RIICHS reset or an internal reset.

Note: Do not set the RS bit to 1 while generating a STOP condition.
Note: If 1 (requests to generate a repeated START condition) is written to the RS bit in slave mode, the repeated START condition is not generated but the RS bit remains set to 1.
If the operating mode changes to master mode with the bit not being cleared, note that the repeated START condition may be generated.

SP Bit (STOP Condition Generation Request)

This bit is used to request that a STOP condition be generated in master mode.
When this bit is set to 1 to request to generate a STOP condition, a STOP condition is generated when the ICBSR.BFREE flag is set to 0 (bus busy state) and the ICMMR.MST flag is set to 1 (master mode).
For details on the STOP condition generation, refer to section 44.4.7, START Condition/Repeated START Condition/STOP Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with the ICBSR.BFREE flag set to 0 and the ICMMR.MST flag set to 1.

[Clearing conditions]

- When 0 is written to the SP bit.
- When a STOP condition has been generated. (a STOP condition is detected.)
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When a START condition and a repeated START condition are detected.
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply an RIICHS reset or an internal reset.

Note: When the ICBSR.BFREE flag is 1 (bus free state), writing 1 to the SP bit is ignored.
Note: Do not set the SP bit to 1 while a repeated START condition is being generated.

44.2.17 Transmit/Receive Data Register (ICDR)

Address(es): RIICHS1.ICDR 8745 F558h, RIICHS2.ICDR 8745 F958h,
RIICHS1_NS.ICDR 0745 F558h, RIICHS2_NS.ICDR 0745 F958h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICDR register is a 32-bit read/write register used to read data from the receive data buffer and write data to the transmit data buffer. This register can be accessed in 32-bit or 8-bit units. The upper 24 bits are reserved and should be written as 0 when data is written in 32-bit units.

For Read Operation:

When 1 byte of data has been received, the received data is transferred from the internal shift register to the ICDR register to enable the next data to be received.

The double-buffer structure of the shift register and the ICDR register allows continuous receive operation if the received data has been read from the ICDR register while the shift register is receiving data.

Read data from the ICDR register once when a receive data full interrupt (RXI) request is generated.

If the ICDR register receives the next receive data before the current data is read from the ICDR register (while the ICCSR.RDRF flag is 1), the RIICHS automatically holds the SCLHSn line low one cycle before the RDRF flag is set to 1 next.

For Write Operation:

When the ICDR register detects a space in the internal shift register, it transfers the transmit data that has been written to the ICDR register to the shift register and starts transmitting data in transmit mode.

The double-buffer structure of the ICDR register and the shift register allows continuous transmit operation if the next transmit data has been written to the ICDR register while the shift register data is being transmitted.

Write transmit data to the ICDR register once when a transmit data empty interrupt (TXI) request is generated.

44.2.18 Status Register 2 (ICSR2)

Address(es): RIICHS1.ICSR2 8745 F5D0h, RIICHS2.ICSR2 8745 F9D0h,
RIICHS1_NS.ICSR2 0745 F5D0h, RIICHS2_NS.ICSR2 0745 F9D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDF	—	—	—	TMOF	—	—	—	AL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEND	—	—	—	NACKF	—	—	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	START	START Condition Detection Flag	0: START condition is not detected. 1: START condition is detected.	R/(W) *1
b1	STOP	STOP Condition Detection Flag	0: STOP condition is not detected. 1: STOP condition is detected.	R/(W) *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NACKF	NACK Detection Flag	0: NACK is not detected 1: NACK is detected.	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEND	Transmission End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	AL	Arbitration Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDF	Wake-Up Condition Detection Flag	0: Wake-up condition is not detected. 1: Wake-up condition is detected.	R/(W) *1
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.

START Flag (START Condition Detection Flag)

[Setting condition]

- When a START condition (or a repeated START condition) is detected while the ICSER.STDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

STOP Flag (STOP Condition Detection Flag)

[Setting condition]

- When a STOP condition is detected while the ICSER.SPDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

NACKF Flag (NACK Detection Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.NAKDE bit is 1 (NACK detection is enabled).
  2. When ACK is not received (NACK is received) from the receiver in transmit mode.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TEND Flag (Transmission End Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.TEDE bit is 1 (transmission end detection is enabled).
  2. At the rising edge of the ninth SCL while the ICCSR.TDRE flag is 1. Excluding when sending an address.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When data is written to the ICDR register.
- When a STOP condition is detected.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

AL Flag (Arbitration Lost Flag)

[Setting condition]

When master arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.MALE bit = 1):

- When the internal SDA output state does not match the SDAHSn line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDAHSn line is driven low while the internal SDA output is high (the SDAHSn pin is in the high-impedance state)).
- The following 1 and 2 are satisfied.
  1. When the START condition is detected while the ICCGR.ST bit is 1.
  2. When the internal SDA output state does not match the SDAHSn line level.
- When the ICCGR.ST bit is set to 1 (requests to generate a START condition) while the ICBSR.BFREE flag is 0.

When NACK transmission arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.NALE bit = 1):

- When the internal SDA output state does not match the SDAHSn line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICSER.ALE bit = 1, ICFER.SALE bit = 1):

- When the internal SDA output state does not match the SDAHSn line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TMOF Flag (Timeout Detection Flag)

[Setting condition]

- All of the following 1 to 3 are satisfied.
  1. The ICSER.TMOE bit is 1 (timeout detection is enabled).
  2. When the master mode or the received slave address matches the slave address n in slave mode.
  3. When the SCLHSn line state remains unchanged for the period specified by ICTOR register.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

WUDF Flag (Wake-Up Condition Detection Flag)

This flag indicates that the condition for generating a wake-up interrupt has been met.

[Setting condition]

- When the received slave address matches any of the addresses permitted for detection by the ICSCR.SAR0E to SAR2E bits, the ICSCR.HOAE bit, and the ICSCR.GCAE bit while the ICWUCR.WUFE bit is 1 (wake-up function is enabled), the ICSER.WUDE bit is 1 (wake-up condition detection is enabled), and ICWUSR.WUASYNF flag is 1 (PCLKA and I3CCLK are stopped) (in slave mode).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1 while the ICWUSR.WUASYNF flag is 0 (PCLKA and I3CCLK are operating).

44.2.19 Status Detection Enable Register (ICSER)

Address(es): RIICHS1.ICSER 8745 F5D4h, RIICHS2.ICSER 8745 F9D4h,
RIICHS1_NS.ICSER 0745 F5D4h, RIICHS2_NS.ICSER 0745 F9D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDE	—	—	—	TMOE	—	—	—	ALE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEDE	—	—	—	NAKDE	—	—	SPDE	STDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STDE	START Condition Detection Enable	0: START condition detection is disabled. 1: START condition detection is enabled.	R/W
b1	SPDE	STOP Condition Detection Enable	0: STOP condition detection is disabled. 1: STOP condition detection is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NAKDE	NACK Detection Enable	0: NACK detection is disabled. 1: NACK detection is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEDE	Transmission End Detection Enable	0: Transmission end detection is disabled. 1: Transmission end detection is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALE	Arbitration Lost Detection Enable	0: Arbitration lost detection is disabled. 1: Arbitration lost detection is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOE	Timeout Detection Enable	0: Timeout detection is disabled. 1: Timeout detection is enabled.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDE	Wake-Up Condition Detection Enable	0: Wake-up condition detection is disabled. 1: Wake-up condition detection is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICSR2 register is set to 1.

44.2.20 Status Interrupt Enable Register (ICSIER)

Address(es): RIICHS1.ICSIER 8745 F5D8h, RIICHS2.ICSIER 8745 F9D8h,
RIICHS1_NS.ICSIER 0745 F5D8h, RIICHS2_NS.ICSIER 0745 F9D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	WUDIE	—	—	—	TMOIE	—	—	—	ALIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEIE	—	—	—	NAKIE	—	—	SPIE	STIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STIE	START Condition Detection Interrupt Enable	0: START condition detection interrupt is disabled. 1: START condition detection interrupt is enabled.	R/W
b1	SPIE	STOP Condition Detection Interrupt Enable	0: STOP condition detection interrupt is disabled. 1: STOP condition detection interrupt is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NAKIE	NACK Detection Interrupt Enable	0: NACK detection interrupt is disabled. 1: NACK detection interrupt is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEIE	Transmission End Interrupt Enable	0: Transmission end interrupt is disabled. 1: Transmission end interrupt is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALIE	Arbitration Lost Detection Interrupt Enable	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOIE	Timeout Detection Interrupt Enable	0: Timeout detection interrupt is disabled. 1: Timeout detection interrupt is enabled.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	WUDIE	Wake-Up Interrupt Enable	0: Wake-up interrupt is disabled. 1: Wake-up interrupt is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable interrupts for each status. When a flag in the ICSR2 register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is output.

44.2.21 Communication Status Register (ICCSR)

Address(es): RIICHS1.ICCSR 8745 F5E0h, RIICHS2.ICCSR 8745 F9E0h,
RIICHS1_NS.ICCSR 0745 F5E0h, RIICHS2_NS.ICCSR 0745 F9E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDRF	TDRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDRE	Transmit Data Empty Flag	0: Transmit data buffer contains transmit data. 1: Transmit data buffer contains no transmit data.	R
b1	RDRF	Receive Data Full Flag	0: Receive data buffer contains no receive data. 1: Receive data buffer contains receive data.	R
b31 to b2	—	Reserved	These bits are read as 0.	R

TDRE Flag (Transmit Data Empty Flag)

[Setting condition]

- When any of the following conditions is satisfied while the ICCSER.TDE bit is 1 (detection of transmit data empty is enabled).
  1. When data is transferred from transmit data buffer to shift register and transmit data buffer becomes empty.
  2. When the ICMMR.TRS flag is set to 1.
  3. When the received slave address matches while the TRS flag is 1.

[Clearing condition]

- When data is written to ICDR register.
- When the ICMMR.TRS flag is set to 0.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

Note: When the ICSR2.NACKF flag is set to 1 while the ICSER.NAKDE bit is 1, RIICHS aborts data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the shift register and the transmit data buffer becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

RDRF Flag (Receive Data Full Flag)

[Setting condition]

- When any of the following conditions is satisfied while the ICCSER.RDE bit is 1 (detection of receive data full is enabled).
  1. When receive data is transferred from shift register to receive data buffer.
     The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (selected in the ICCSCR.WAITAE bit).
  2. When the received slave address matches after a START (or repeated START) condition is detected with the ICMMR.TRS flag set to 0.

[Clearing condition]

- When data is read from ICDR register.

- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

44.2.22 Communication Status Detection Enable Register (ICCSER)

Address(es): RIICHS1.ICCSER 8745 F5E4h, RIICHS2.ICCSER 8745 F9E4h,
RIICHS1_NS.ICCSER 0745 F5E4h, RIICHS2_NS.ICCSER 0745 F9E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDE	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Transmit Data Empty Detection Enable	0: Detection of transmit data empty is disabled. 1: Detection of transmit data empty is enabled.	R/W
b1	RDE	Receive Data Full Detection Enable	0: Detection of receive data full is disabled. 1: Detection of receive data full is enabled.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICCSR register is set to 1.

44.2.23 Communication Status Interrupt Enable Register (ICCSIER)

Address(es): RIICHS1.ICCSIER 8745 F5E8h, RIICHS2.ICCSIER 8745 F9E8h,
RIICHS1_NS.ICCSIER 0745 F5E8h, RIICHS2_NS.ICCSIER 0745 F9E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIE	TIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b1	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable interrupts for each status. When a flag in the ICCSR register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is generated.

44.2.24 Bus Status Register (ICBSR)

Address(es): RIICHS1.ICBSR 8745 F610h, RIICHS2.ICBSR 8745 FA10h,
RIICHS1_NS.ICBSR 0745 F610h, RIICHS2_NS.ICBSR 0745 FA10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BFREE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFREE	Bus Free Flag	0: Have not detected bus free. 1: Have detected bus free.	R
b31 to b1	—	Reserved	These bits are read as 0.	R

BFREE Flag (Bus Free Flag)

This flag indicates whether the I2C-bus is occupied (bus busy) or released (bus free).

[Setting condition]

- When the number of cycles set in the ICBFTR register has elapsed when the SCLHSn and SDAHSn lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBFTR register has elapsed when the SCLHSn and SDAHSn lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When a START condition is detected.
- When the ICCR.ICE bit is set to 0.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

44.2.25 Slave Mode Status Register (ICSSR)

Address(es): RIICHS1.ICSSR 8745 F614h, RIICHS2.ICSSR 8745 FA14h, RIICHS1_NS.ICSSR 0745 F614h, RIICHS2_NS.ICSSR 0745 FA14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	—	—	—	—	—	—	—	DID	HSMC	—	—	—	—	GCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMC	Hs-mode Master Code Detection Flag	0: Hs-mode master code is not detected. 1: Hs-mode master code is detected.	R/(W) *1
b6	DID	Device-ID Address Detection Flag	0: Device-ID address is not detected. 1: Device-ID address is detected. • This bit is set to 1 when the first frame received immediately after a START condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1
b16	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b17	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b18	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.GCAE bit is 1 (general call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000b + 0 (write)).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

HSMC Flag (Hs-mode Master Code Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HSMCE bit is 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXXb) + 1 (NACK).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of device ID (1111 100b) + 0 (write).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected does not match a value of device ID (1111 100b).
  3. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2.

HOA Flag (Host Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HOAE bit is 1 (host address detection is enabled).
  2. When the received slave address matches the host address (0001 000b).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting condition]

For 7-bit address format (SAMRy.FS flag = 0):

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches the SAMRy.SVA[6:0] bits value.

For 10-bit address format (SAMRy.FS flag = 1):

- This flag is set to 1 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 1111 0b + SAMRy.SVA[9:8] bits and the following address matches the SAMRy.SVA[7:0] bits value.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.

For 7-bit address format (SAMRy.FS flag = 0):

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match SAMRy.SVA[6:0] bits value.

For 10-bit address format (SAMRy.FS flag = 1):

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match a value of 1111 0b + SAMRy.SVA[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 1111 0b + SAMRy.SVA[9:8] bits and the following address does not match the SAMRy.SVA[7:0] bits value.

44.2.26 Wake-Up Unit Status Register (ICWUSR)

Address(es): RIICHS1.ICWUSR 8745 F618h, RIICHS2.ICWUSR 8745 FA18h, RIICHS1_NS.ICWUSR 0745 F618h, RIICHS2_NS.ICWUSR 0745 FA18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WUASYNF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WUASYNF	Wake-Up Detection Circuit Status Flag	0: Wake-up detection circuit is disabled (PCLKA and I3CCLK are operating). 1: Wake-up detection circuit is enabled (PCLKA and I3CCLK are stopped).	R
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

WUASYNF Flag (Wake-Up Detection Circuit Status Flag)

This bit indicates whether the wake-up detection circuit is enabled or disabled.

[Setting condition]

- When the ICBSR.BFREE flag is set to 1 (bus free) while the ICWUCR.WUFE bit is 1 (wake-up function is enabled) and the WUFSYNE bit is 0 (asynchronous operation).

[Clearing condition]

- When the ICWUCR.WUFE bit is set to 0 (wake-up function is disabled).
- When a wake-up event is detected while the WUFE bit is 1, and then the ICWUCR.WUFSYNE bit is set to 1 (internal clock synchronous operation) while this flag is set to 1 (in slave mode).
- When a STOP condition is detected while the WUFE bit is 1, this flag is 1, and the WUFSYNE bit is 1 (in slave mode).

44.2.27 Slave Address Register y (SARy) (y = 0 to 2)

Address(es): RIICHS1.SAR0 8745 F6B0h, RIICHS1.SAR1 8745 F6B4h, RIICHS1.SAR2 8745 F6B8h, RIICHS2.SAR0 8745 FAB0h, RIICHS2.SAR1 8745 FAB4h, RIICHS2.SAR2 8745 FAB8h, RIICHS1_NS.SAR0 0745 F6B0h, RIICHS1_NS.SAR1 0745 F6B4h, RIICHS1_NS.SAR2 0745 F6B8h, RIICHS2_NS.SAR0 0745 FAB0h, RIICHS2_NS.SAR1 0745 FAB4h, RIICHS2_NS.SAR2 0745 FAB8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FS	SVA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SVA[9:0]	Slave Address	Set a slave address.	R/W
b10	FS	Address Format Select	0: 7-bit address format 1: 10-bit address format	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

FS Bit (Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address (SVA[9:0] bits).

When this bit is 0, the 7-bit address format is selected for slave address, the SVA[6:0] bit setting is valid, and the SVA[9:7] bit setting is ignored.

When this bit is 1, the 10-bit address format is selected for slave address and the SVA[9:0] bit setting is valid.

When the ICSCR.SARyE bit is 0, the setting of this bit is ignored.

44.2.28 Slave Address Monitor Register y (SAMRy) (y = 0 to 2)

Address(es): RIICHS1.SAMR0 8745 F730h, RIICHS1.SAMR1 8745 F734h, RIICHS1.SAMR2 8745 F738h, RIICHS2.SAMR0 8745 FB30h, RIICHS2.SAMR1 8745 FB34h, RIICHS2.SAMR2 8745 FB38h, RIICHS1_NS.SAMR0 0745 F730h, RIICHS1_NS.SAMR1 0745 F734h, RIICHS1_NS.SAMR2 0745 F738h, RIICHS2_NS.SAMR0 0745 FB30h, RIICHS2_NS.SAMR1 0745 FB34h, RIICHS2_NS.SAMR2 0745 FB38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SAV	—	—	FS	—	SVA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0.	R
b25 to b16	SVA[9:0]	Slave Address	Slave address is set.	R
b26	—	Reserved	This bit is read as 0.	R
b27	FS	Address Format Flag	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
b29, b28	—	Reserved	These bits are read as 0.	R
b30	SAV	Slave Address Valid Flag	0: Slave address is disabled. 1: Slave address is enabled.	R
b31	—	Reserved	This bit is read as 0.	R

SVA[9:0] Bits (Slave Address)

The SVA[9:0] bits indicate a valid slave address.

- When the SAV flag is 1 and the FS flag is 0:
  The SVA[9:7] bits are 000b.
  The SVA[6:0] bits are the SARy.SVA[6:0] bits.
- When the SAV flag is 1 and the FS flag is 1:
  The SVA[9:0] bits are the SARy.SVA[9:0] bits.

FS Flag (Address Format Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (detection of slave y is enabled).
  2. The SARy.FS bit is 1 (the address length is 10 bits).

[Clearing condition]

- [Setting condition] is not satisfied.

SAV Flag (Slave Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (detection of slave address y is enabled).
  2. The SARy.SVA[6:0] bits are not 0000000b when the FS flag is 0.
     The SARy.SVA[9:0] bits are not 0000000000b when the FS flag is 1.

[Clearing condition]

- [Setting condition] is not satisfied.

44.2.29 Bit Count Register (ICBCR)

Address(es): RIICHS1.ICBCR 8745 F780h, RIICHS2.ICBCR 8745 FB80h,
RIICHS1_NS.ICBCR 0745 F780h, RIICHS2_NS.ICBCR 0745 FB80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BC[4:0]	Bit Counter	These bits indicate the number of bits remaining to be transferred. Refer to Table 44.5 for details.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

BC[4:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCLHSn line.

Table 44.5 Counter Value and Number of Remaining Bits

	Master		Slave	
BC[4:0]	Address Phase	Data Phase	Address Phase	Data Phase
00h	2 or 1 bits	2 or 1 bits	3 to 1 bits	2 or 1 bits
01h	3 bits	3 bits	4 bits	3 bits
02h	4 bits	4 bits	5 bits	4 bits
03h	5 bits	5 bits	6 bits	5 bits
04h	6 bits	6 bits	7 bits	6 bits
05h	7 bits	7 bits	8 bits	7 bits
06h	8 bits	8 bits	9 bits	8 bits
07h	9 bits	9 bits	—	9 bits

44.2.30 Internal Status Monitor Register (ICIMR)

Address(es): RIICHS1.ICIMR 8745 F7CCh, RIICHS2.ICIMR 8745 FBCCh,
RIICHS1_NS.ICIMR 0745 F7CCh, RIICHS2_NS.ICIMR 0745 FBCCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SDAO	SCLO	SDAI	SCLI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SCLI	SCL Line Monitor Flag	0: SCLHSn line is low. 1: SCLHSn line is high.	R
b1	SDAI	SDA Line Monitor Flag	0: SDAHSn line is low. 1: SDAHSn line is high.	R
b2	SCLO	SCL Output Monitor Flag	0: The RIICHS drives the SCLHSn pin low. 1: The RIICHS releases the SCLHSn pin.	R
b3	SDAO	SDA Output Monitor Flag	0: The RIICHS drives the SDAHSn pin low. 1: The RIICHS releases the SDAHSn pin.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

SCLO Flag (SCL Output Monitor Flag) and SDAO Flag (SDA Output Monitor Flag)

When reading these bits, the state of signals output from the RIICHS can be read.

44.3 Operation

44.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The frame following a START condition or repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is generated. Figure 44.2 shows the I2C-bus format, and Figure 44.3 shows the I2C-bus timing.

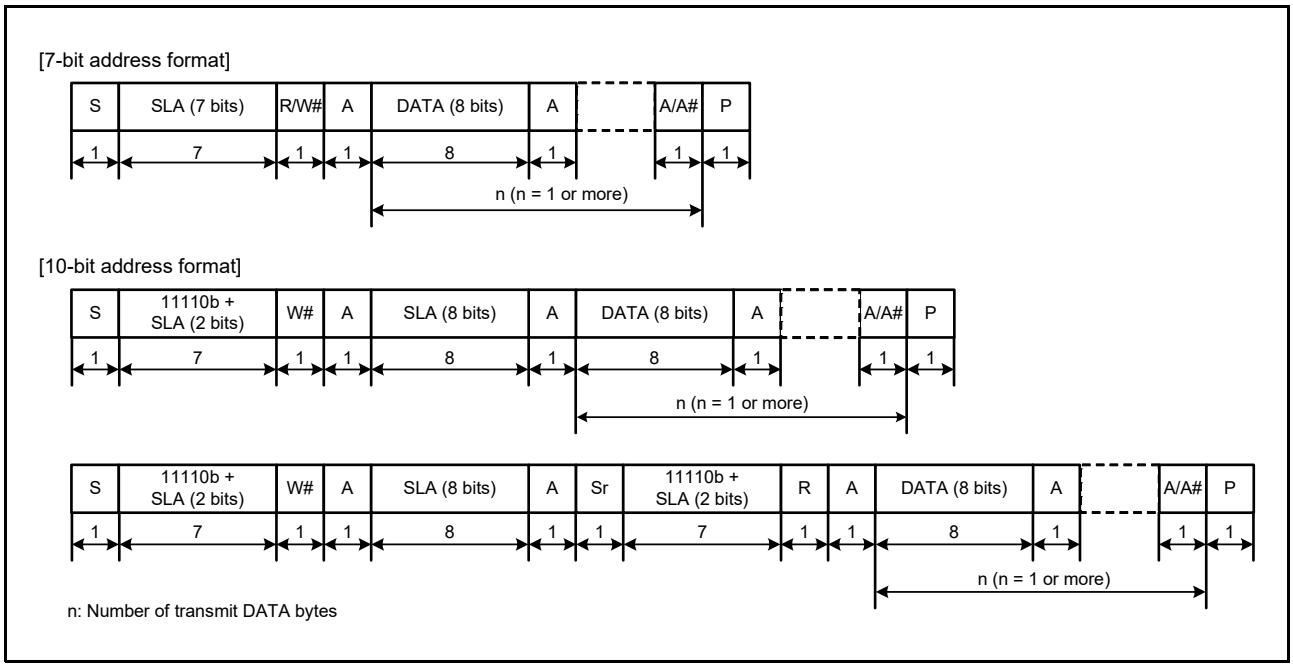


Figure 44.2 I2C-bus Format

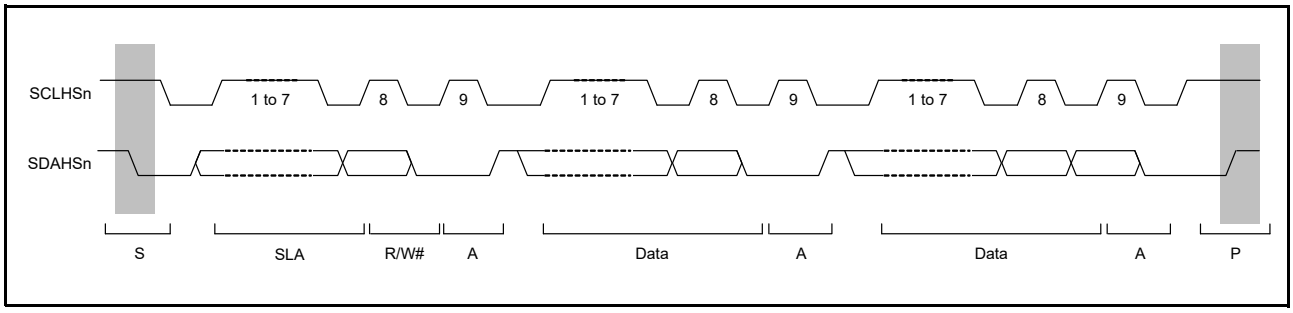


Figure 44.3 I2C-bus Timing (SLA = 7 Bits)

S: START condition. The master device drives the SDAHSn line low from high while the SCLHSn line is high.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.

A: Acknowledge. The receiver drives the SDAHSn line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDAHSn line high.

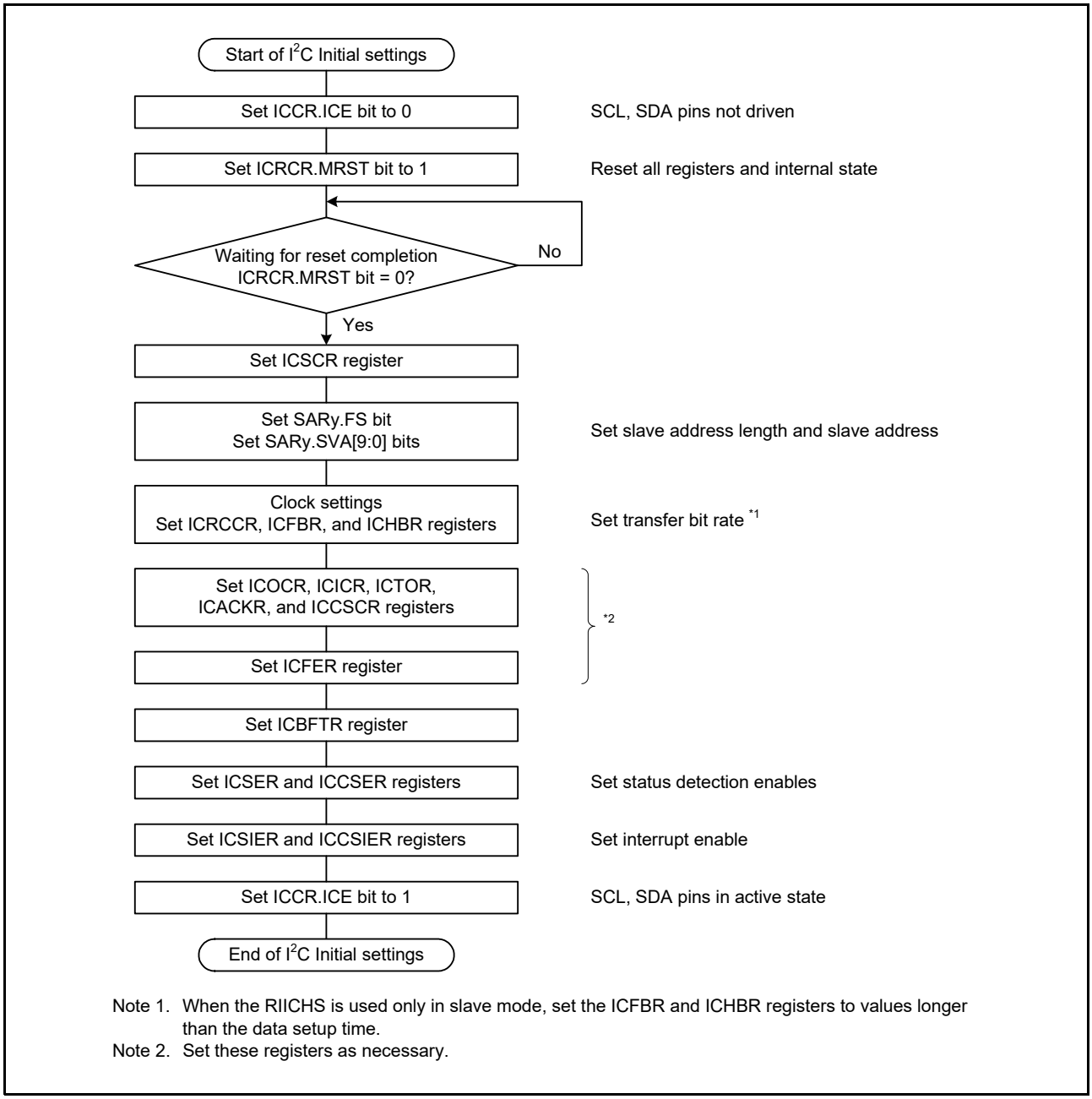
Sr: Repeated START condition. The master device drives the SDAHSn line low from high after the setup time has elapsed with the SCLHSn line high.

DATA: Transmitted or received data

P: STOP condition. The master device drives the SDAHSn line high from low while the SCLHSn line is high.

44.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIICHS according to the procedure in Figure 44.4.
First, Set the ICCR.ICE bit to 0 (SCLHSn, SDAHSn pins not driven).
Next, Set the ICRCR.MRST bit to 1 (RIICHS reset). This initialize the all registers and internal state. Refer to Table 44.8, Reset Control of Registers. Then, it waits for MRST to become 0.
After that, set SARy.FS bit, SARy.SVA[9:0] bits, ICFBR register, ICICR register, ICOCR register, ICTOR register, ICCSCR register, ICACKR register, ICFER register, and set the other registers as necessary (y = 0 to 2).



Note 1. When the RIICHS is used only in slave mode, set the ICFBR and ICHBR registers to values longer than the data setup time.
Note 2. Set these registers as necessary.

Figure 44.4 Example of RIICHS Initialization Flowchart (y = 0 to 2)

44.3.3 I2C Master Operation

44.3.3.1 Master Transmit Operation

In master transmit operation, the RIICHS outputs the SCL and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 44.5 shows an example of usage of master transmission and Figure 44.6 to Figure 44.8 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 44.3.2, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. At the same time, the BFREE flag is automatically set to 0, the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAHSn line have matched while the ICCGR.ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.MST and ICMMR.TRS flags are automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS flag is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIICHS continues in master transmit mode. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCGR.SP bit to generate a STOP condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDR register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDR register.
(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RIICHS automatically holds the SCLHSn line low until the data for transmission are ready or a STOP condition is generated
(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCGR.SP bit to 1 (requests to generate a STOP condition). Upon receiving a STOP condition generation request, the RIICHS generates the STOP condition.
(6) Upon detecting the STOP condition, the RIICHS automatically sets the ICMMR.MST and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, it automatically sets the ICCSR.TDRE flag to 0, ICSR2.TEND flag to 0 and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

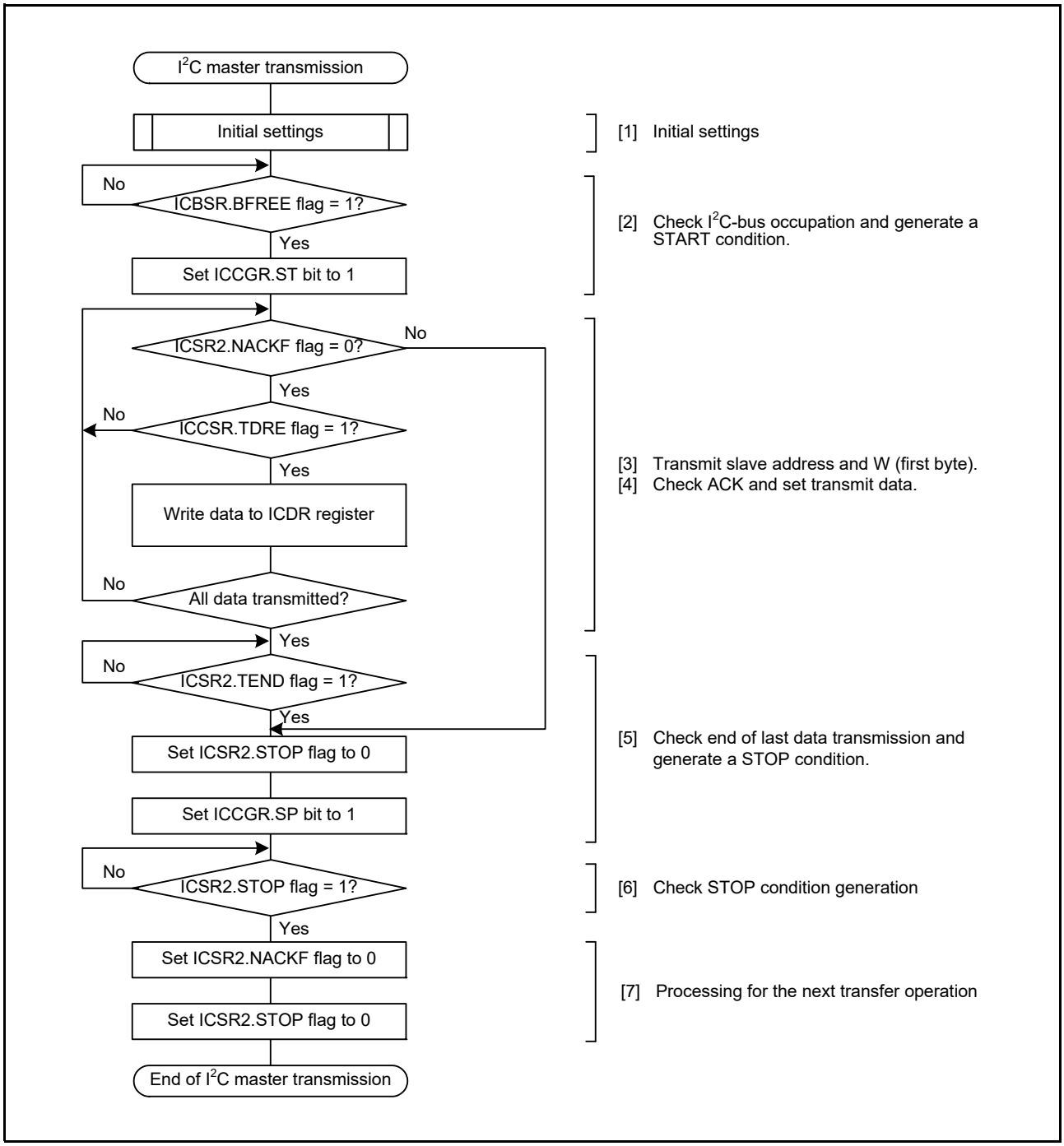


Figure 44.5 Example of I2C Master Transmission Flowchart

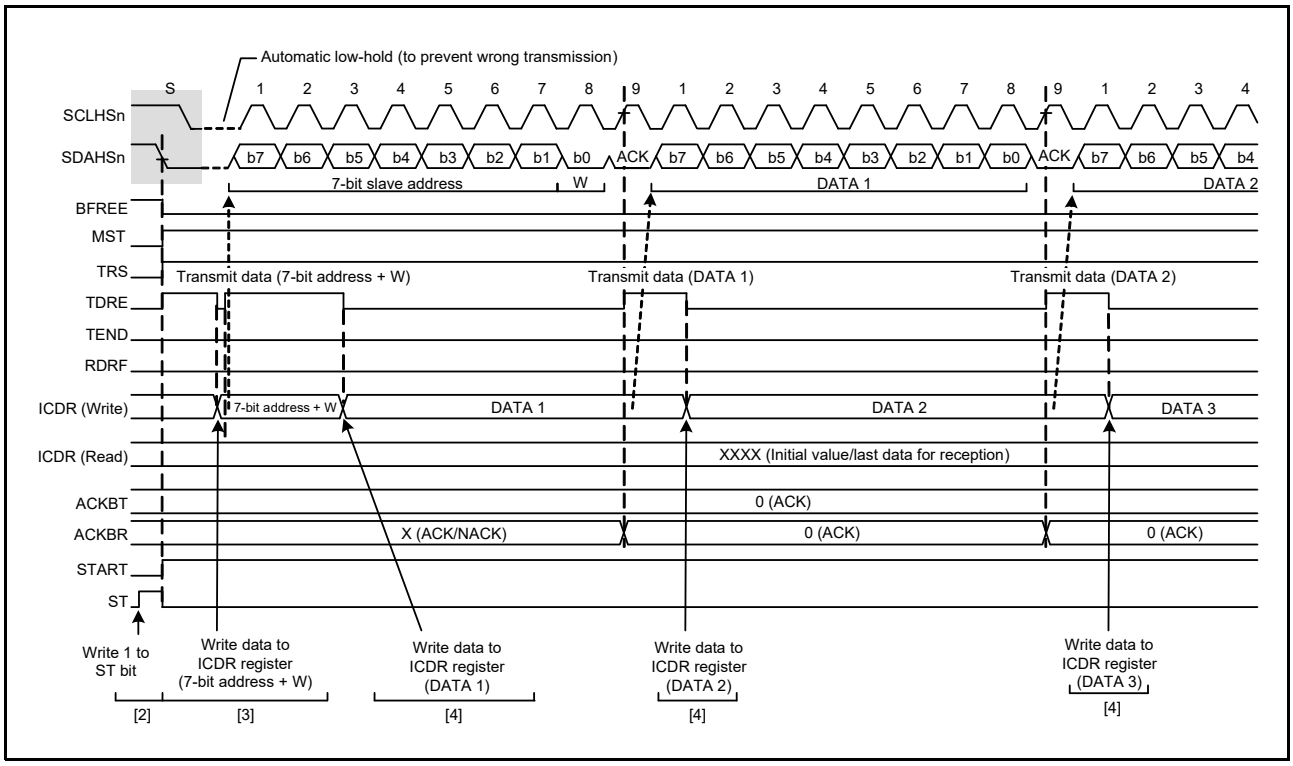


Figure 44.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

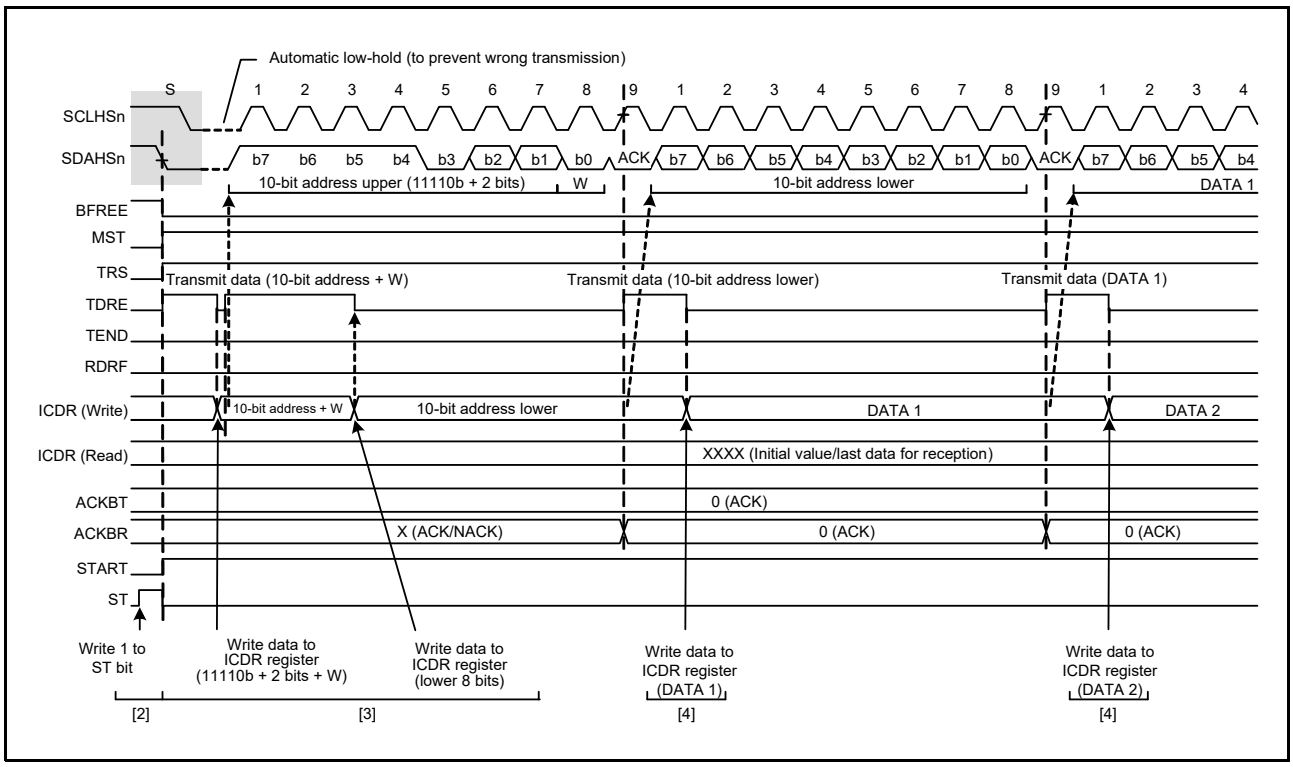


Figure 44.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

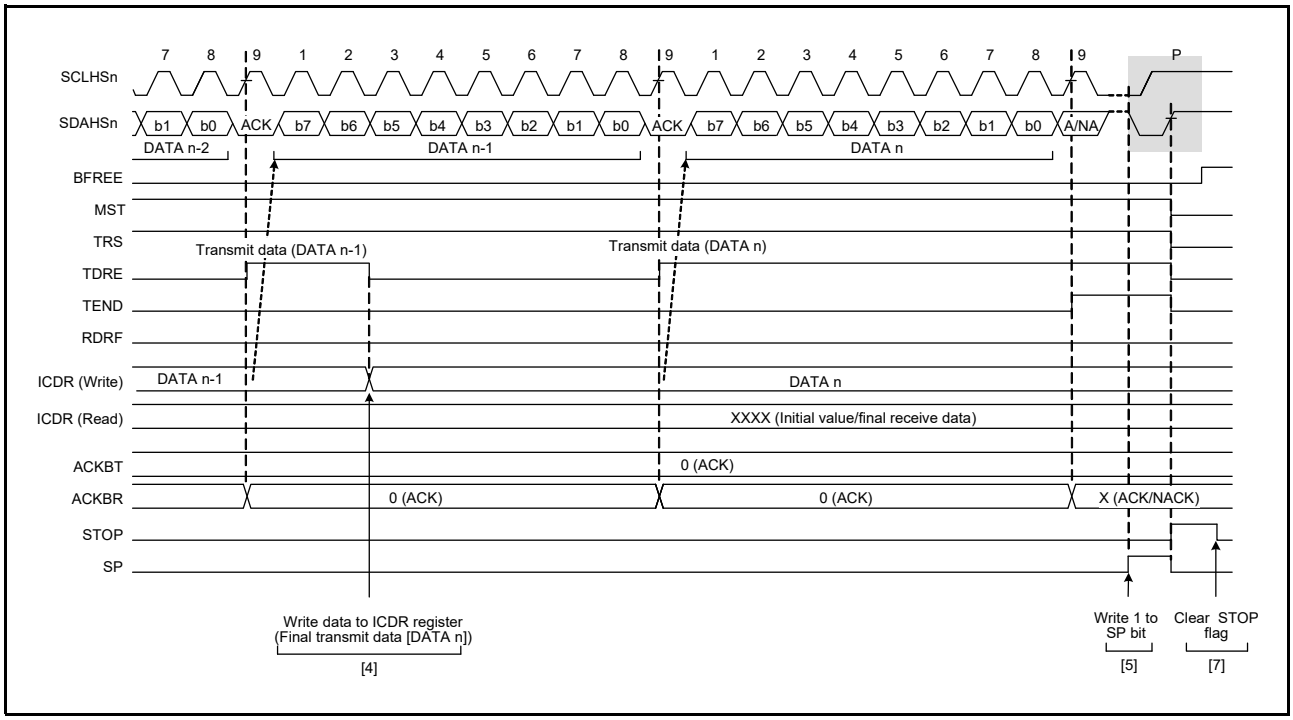


Figure 44.8 Master Transmit Operation Timing (3)

44.3.3.2 Master Receive Operation

In master receive operation, the RIICHS as a master device outputs the SCL, receives data from the slave device, and returns acknowledgments. Since the RIICHS must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 44.9 and Figure 44.10 show examples of usage of master reception and Figure 44.11 to Figure 44.13 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 44.3.2, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. When the RIICHS detects the START condition, the ICBSR.BFREE flag is automatically set to 0 and the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the SDAHSn line have matched while the ICCGR.ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.MST and ICMMR.TRS flags are automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICMMR.TRS flag is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS flag is set to 0 on the rising edge of the ninth SCL, placing the RIICHS in master receive mode. At this time, the ICCSR.TDRE flag is set to 0 and if the slave device acknowledges the ICCSR.RDRF flag is automatically set to 1.
If no slave device recognized the address or there was an error in communications, the ICSR2.NACKF flag becomes 1. In this case, write 1 to the ICCGR.SP bit to generate a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a repeated START condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIICHS in master receive mode.
(4) Dummy read ICDR register after confirming that the ICCSR.RDRF flag is 1; this makes the RIICHS start output of the SCL and start data reception.
(5) After 1 byte of data has been received, the ICCSR.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICCSCR.WAITAE bit. Reading out ICDR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICACKR.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICCSCR.WAITRE bit to 1 (for wait insertion) before reading the ICDR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICACKR.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCLHSn line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a STOP condition is possible.
(6) When the ICCSCR.WAITAE bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICACKR.ACKBT bit to 1 (NACK).
(7) After reading out the byte before last from the ICDR register, if the value of the ICCSR.RDRF flag is confirmed to be 1, write 1 to the ICCGR.SP bit (requests to generate a STOP condition) and then read the last byte from ICDR register. When the ICCGR.SP bit is set to 1, the RIICHS is released from the wait state and generates the STOP condition after low-level output in the ninth clock pulse is completed or the SCLHSn line is released from the low-hold state.

(8) Upon detecting the STOP condition, the RIICHS automatically sets the ICMMR.MST and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

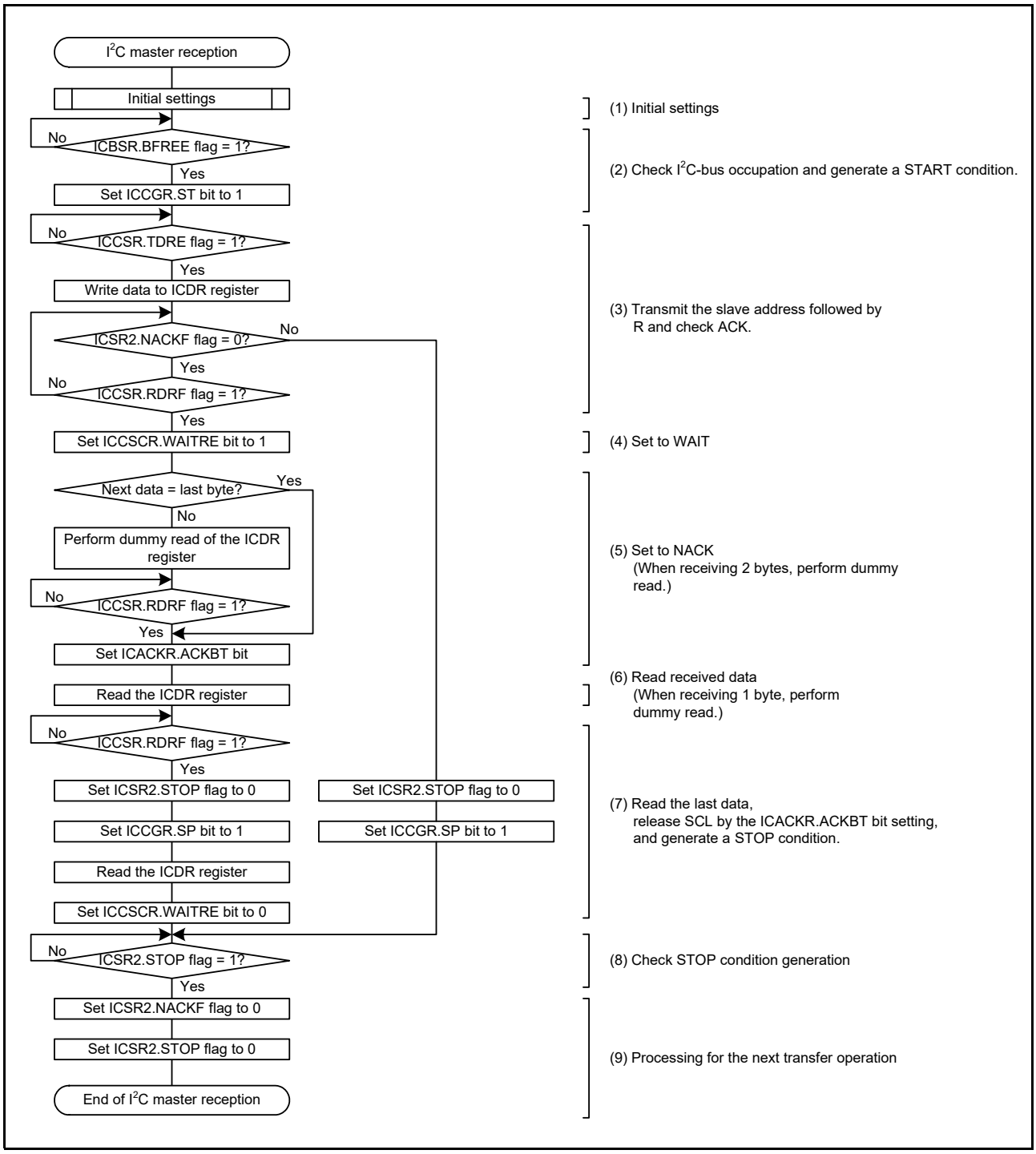


Figure 44.9 Example of I2C Master Reception Flowchart (7-Bit Address Format, 1 or 2 bytes)

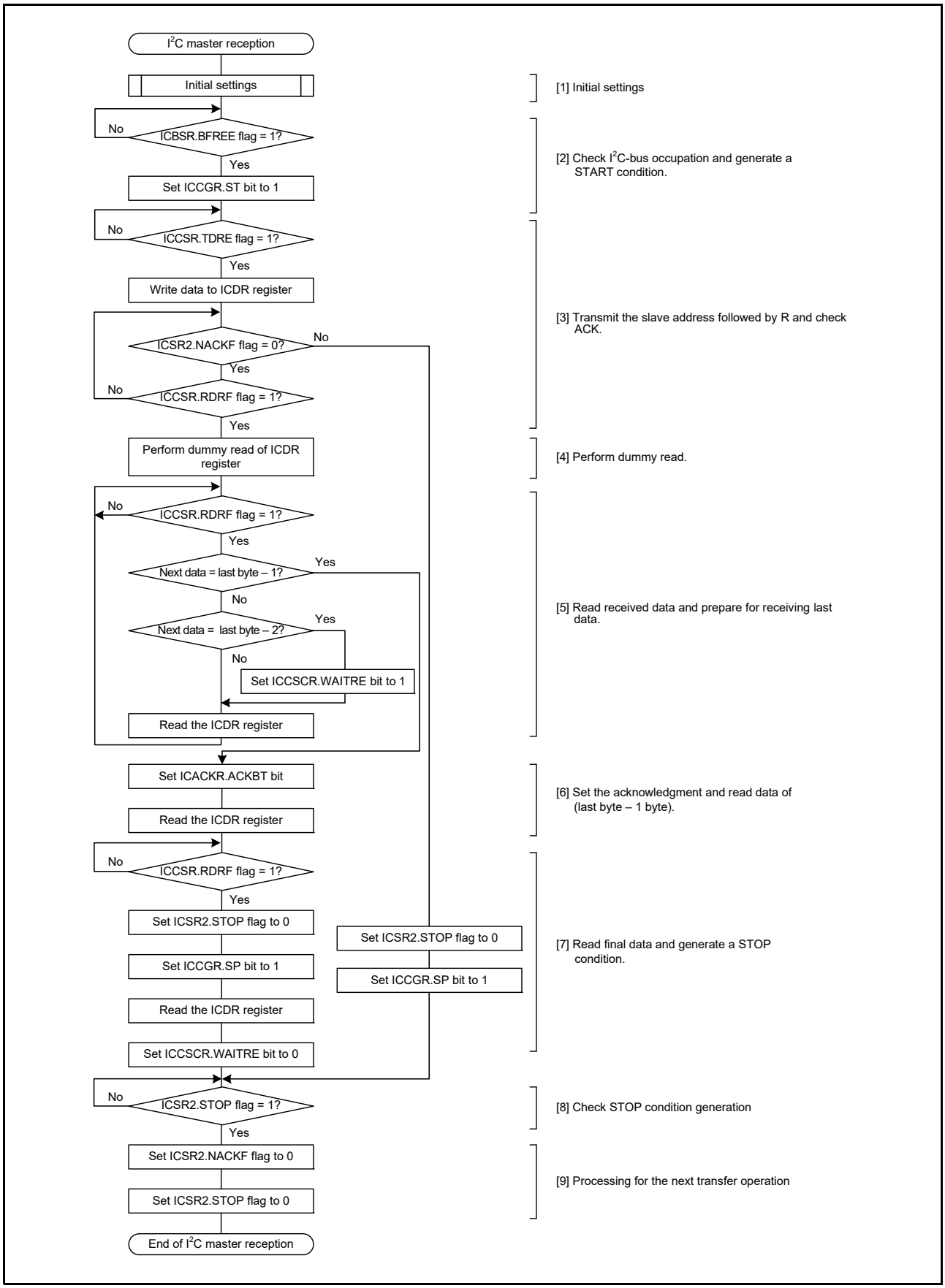


Figure 44.10 Example of I2C Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

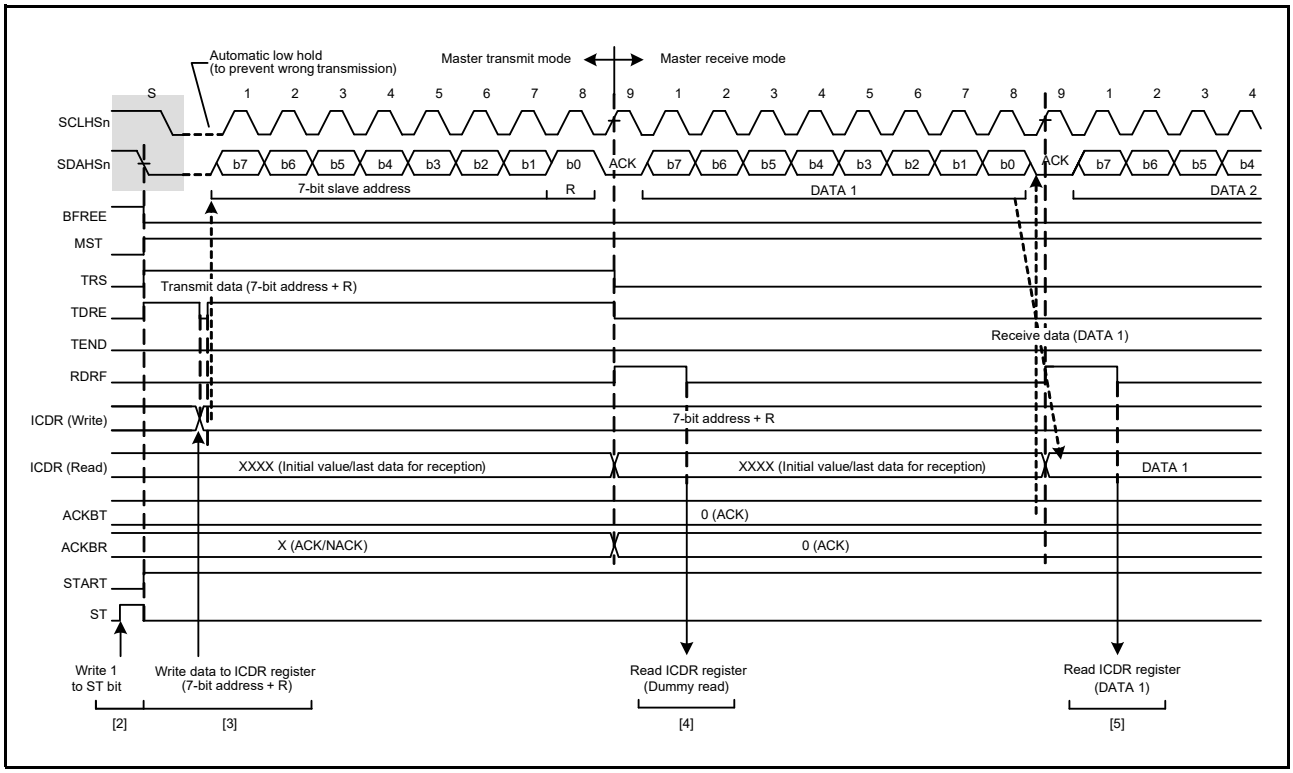


Figure 44.11 Master Receive Operation Timing (1) (7-Bit Address Format, When WAITAE = 0)

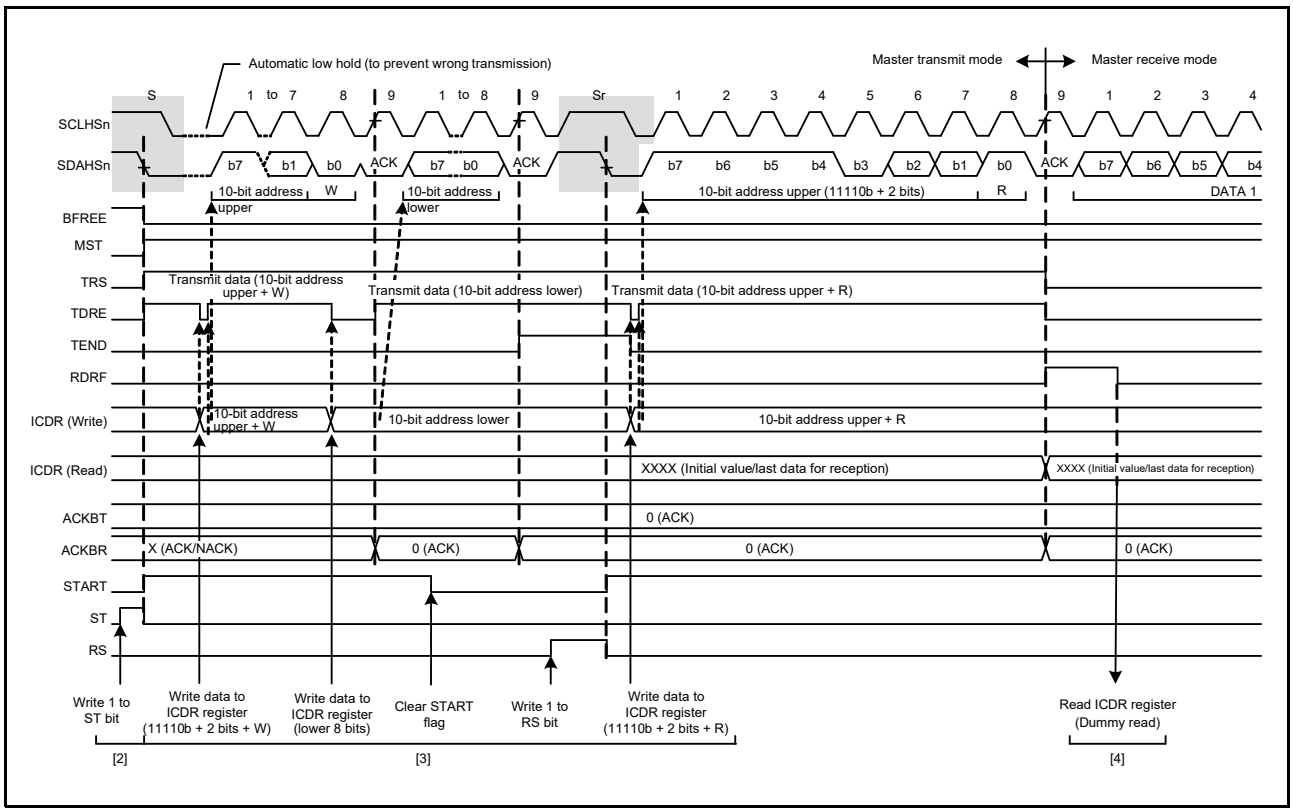


Figure 44.12 Master Receive Operation Timing (2) (10-Bit Address Format, When WAITAE = 0)

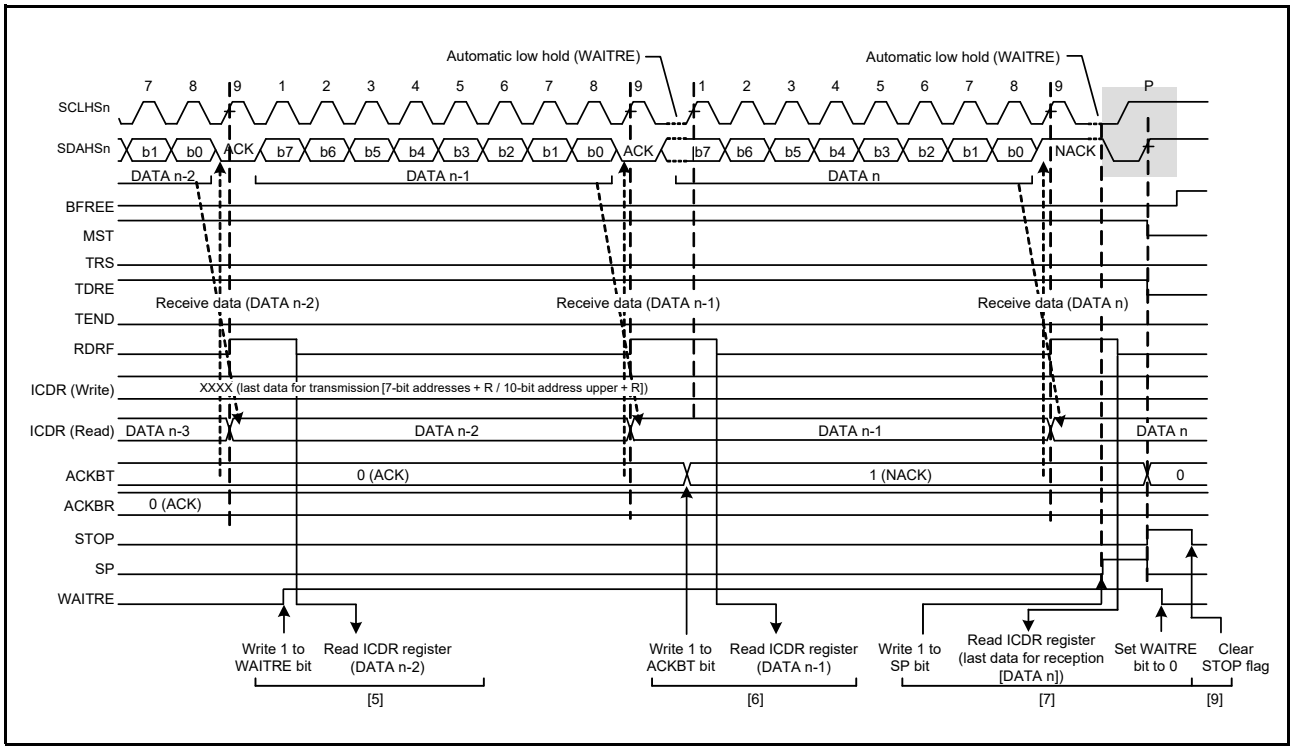


Figure 44.13 Master Receive Operation Timing (3) (When WAITAE = 0)

44.3.4 I2C Slave Operation

44.3.4.1 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIICHS transmits data as a slave device, and the master device returns acknowledgments.
Figure 44.14 shows an example of usage of slave transmission and Figure 44.15 and Figure 44.16 show the timing of operations in slave transmission.
The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 44.3.2, Initial Settings.
After initial settings, the RIICHS will stay in the standby state until it receives a slave address that it matches.
(2) After receiving a matching slave address, the RIICHS sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 1, the RIICHS automatically places itself in slave transmit mode by setting both the ICMMR.TRS flag and the ICCSR.TDRE flag to 1.
(3) After the ICCSR.TDRE flag is confirmed to be 1, write the data for transmission to the ICDR register. At this time, if the RIICHS receives no acknowledgment from the master device (receives an NACK signal) while the ICSER.NAKDE bit is 1, the RIICHS suspends transfer of the next data.
(4) Wait until the ICSR2.TEND flag is set to 1 while the ICCSR.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDR register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIICHS drives the SCLHSn line low on the falling edge of ninth SCL.
(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDR register to complete the processing. This releases the SCLHSn line.
(6) Upon detecting the STOP condition, the RIICHS automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2), flags ICCSR.TDRE and ICSR2.TEND, and the ICMMR.TRS flag to 0, and enters slave receive mode.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

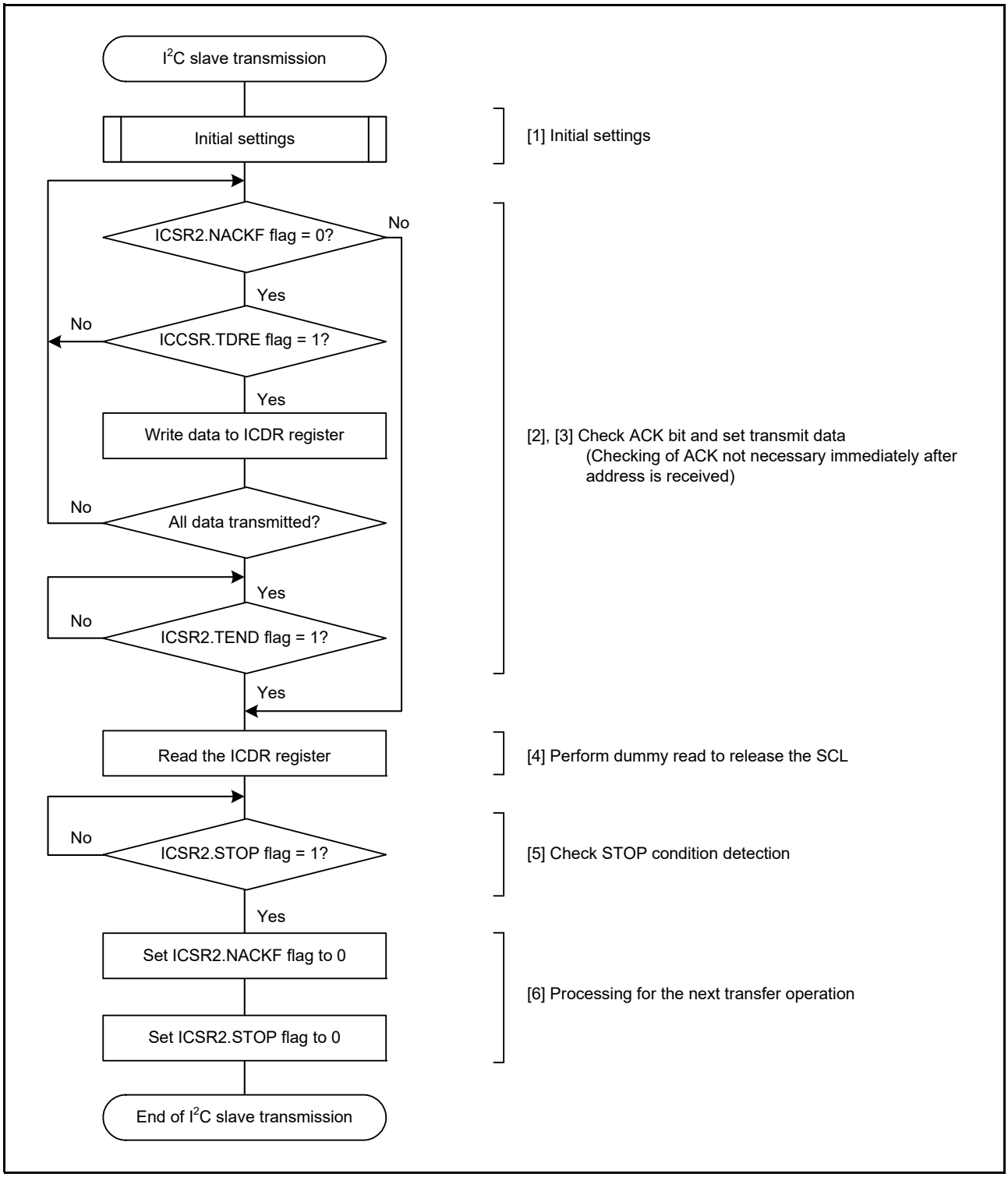


Figure 44.14 Example of I2C Slave Transmission Flowchart

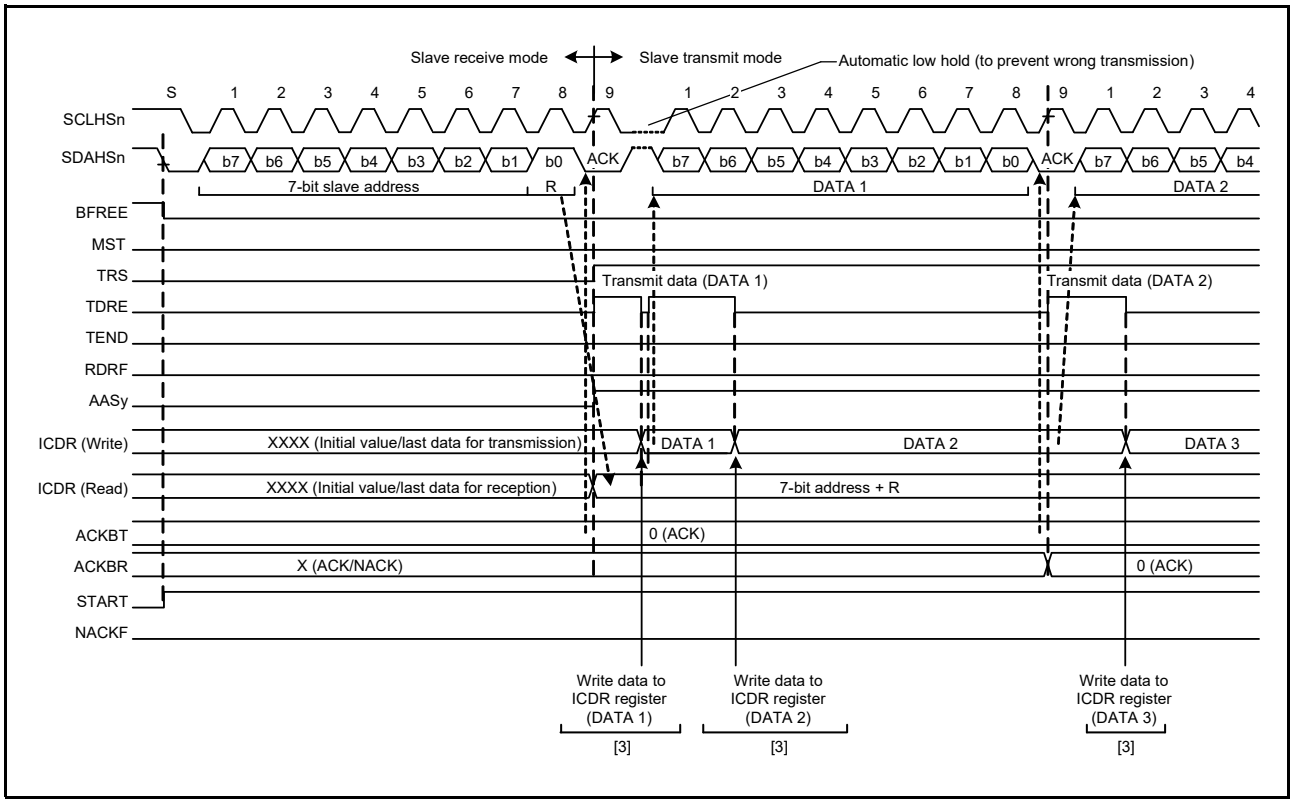


Figure 44.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

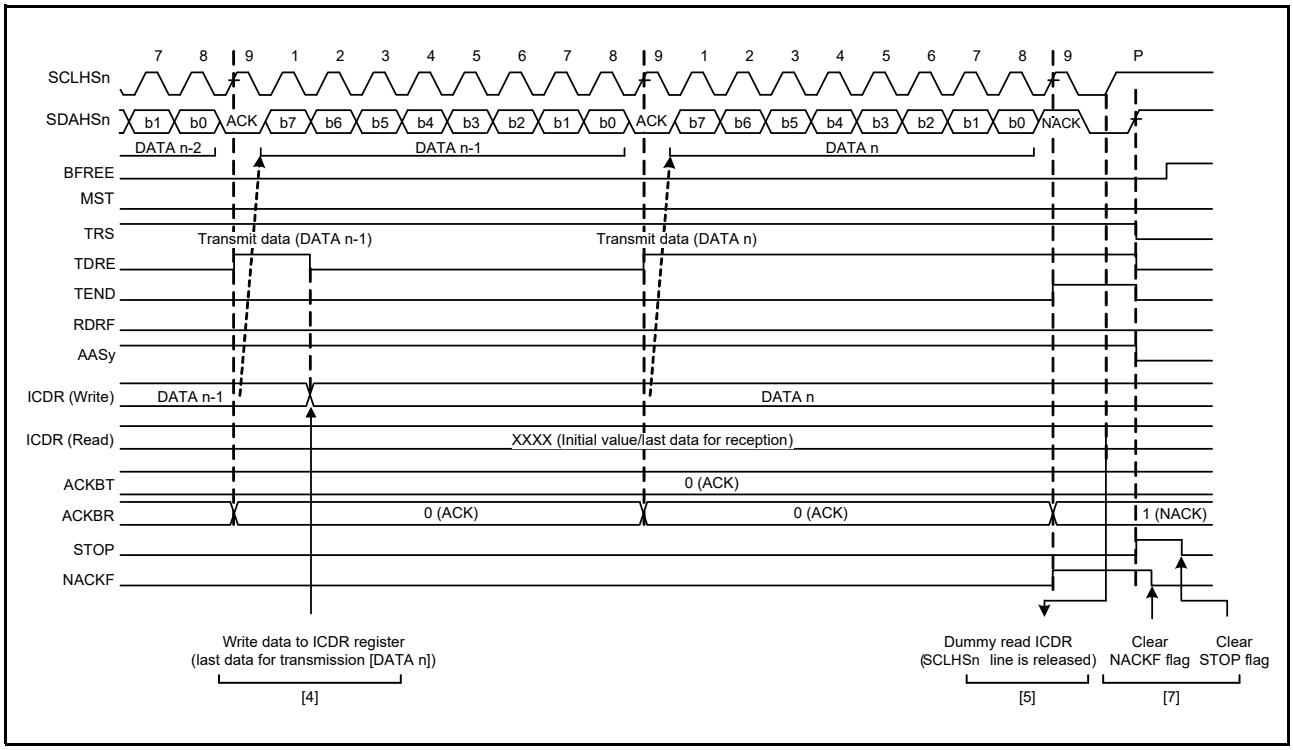


Figure 44.16 Slave Transmit Operation Timing (2)

44.3.4.2 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIICHS returns acknowledgments as a slave device.

Figure 44.17 shows an example of usage of slave reception and Figure 44.18 and Figure 44.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 44.3.2, Initial Settings. After initial settings, the RIICHS will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIICHS sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 0, the RIICHS continues to place itself in slave receive mode and sets the ICCSR.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICCSR.RDRF flag to be 1, dummy read ICDR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8-bits when the 10-bit address format is selected).

(4) When ICDR register is read, the RIICHS automatically sets the ICCSR.RDRF flag to 0. If reading of ICDR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIICHS holds the SCLHSn line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDR register releases the SCLHSn line from being held low.
When the ICSR2.STOP flag is 1 and the ICCSR.RDRF flag is also 1 or when all the data is completely received, read the ICDR register.

(5) Upon detecting the STOP condition, the RIICHS automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

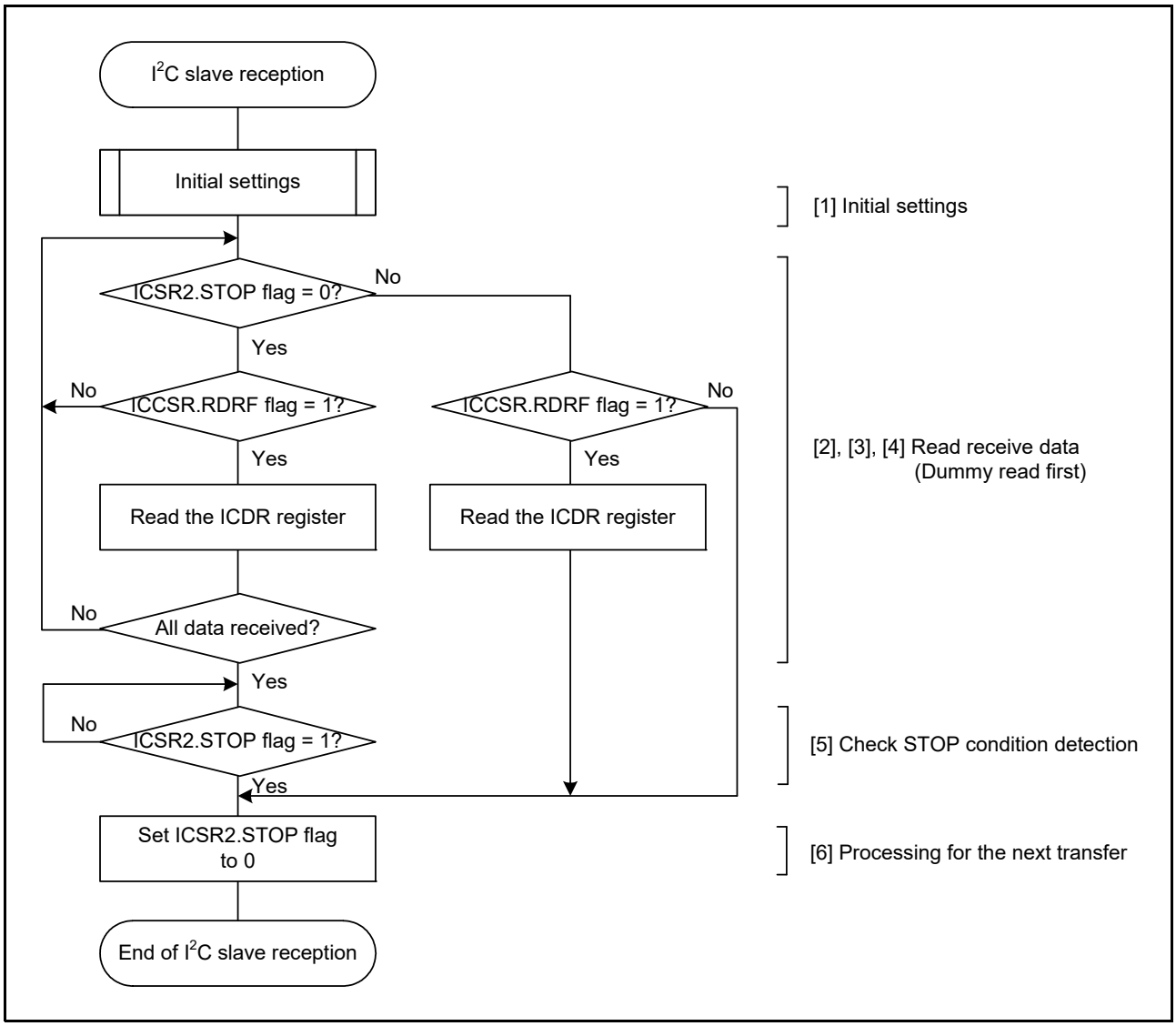


Figure 44.17 Example of I2C Slave Reception Flowchart

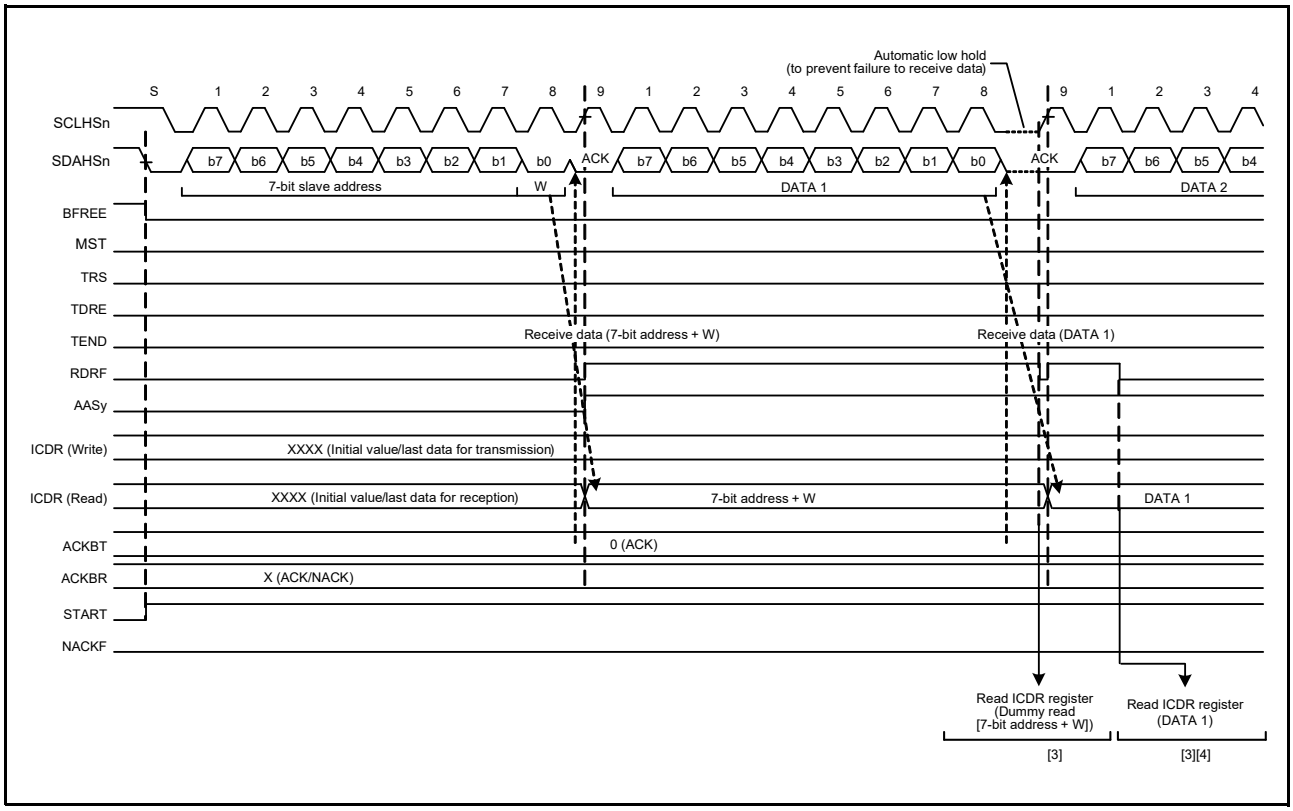


Figure 44.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when WAITAE = 0)

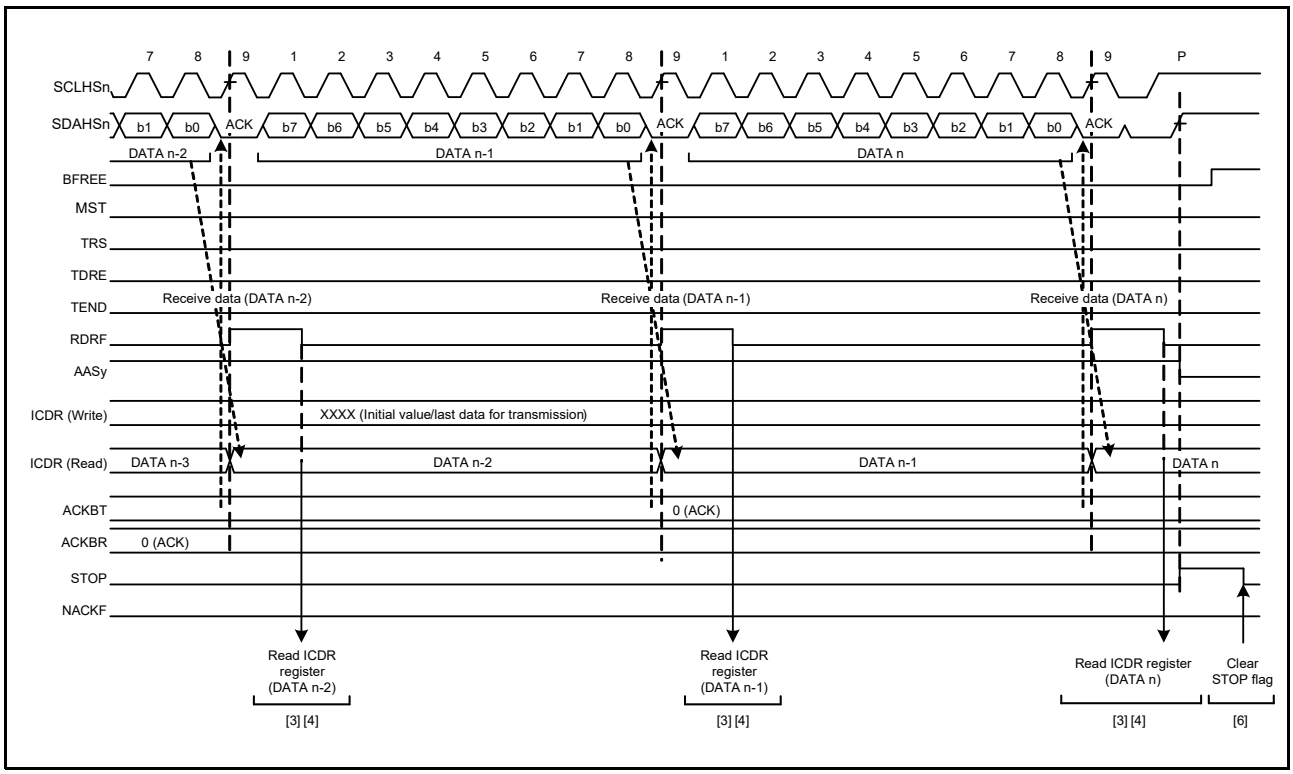


Figure 44.19 Slave Receive Operation Timing (2) (when WAITAE = 0)

44.4 Details of Function

44.4.1 SCL Synchronization Circuit

In generation of the SCL, the RIICHS starts counting out the value for width at high level specified in the ICFBR.HIGH[7:0] bits when it detects a rising edge on the SCLHSn line and drives the SCLHSn line low once counting of the width at high level is complete. When the RIICHS detects the falling edge of the SCLHSn line, it starts counting out the width at low period specified in the ICFBR.LOW[7:0] bits, and then stops driving the SCLHSn line (releases the line) once counting of the width at low level is complete. The SCL is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIICHS is equipped with a function (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL signals by monitoring the SCLHSn line while in master mode.

When the RIICHS has detected a rising edge on the SCLHSn line and thus started counting out the width at high level specified in the ICFBR.HIGH[7:0] bits, and the level on the SCLHSn line falls because an SCL signal is being generated by another master device, the RIICHS stops counting when it detects the falling edge, drives the level on the SCLHSn line low, and starts counting out the width at low level specified in the ICFBR.LOW[7:0] bits. When the RIICHS finishes counting out the width at low level, it stops driving the SCLHSn line to low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIICHS, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCLHSn line has been released. When the RIICHS finishes outputting the low period of the SCL, the SCLHSn line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

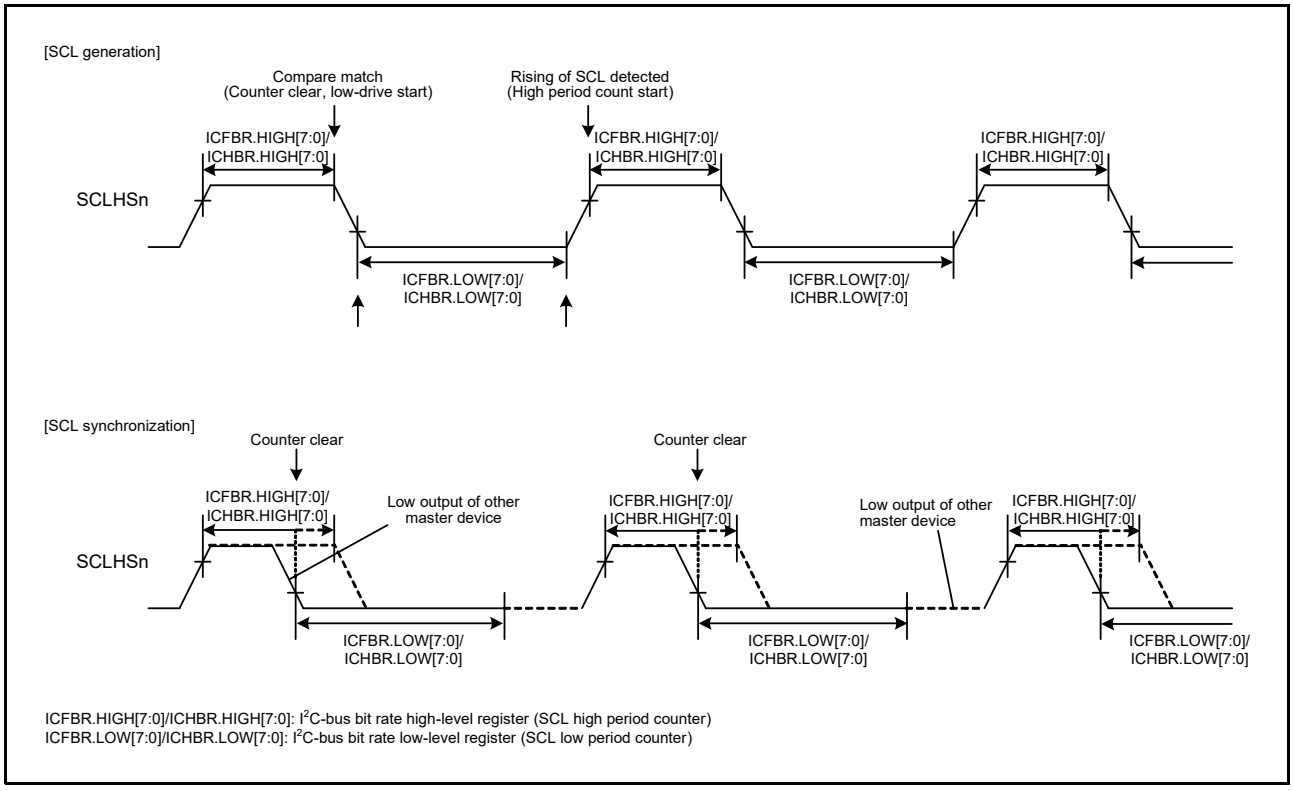


Figure 44.20 Generation and Synchronization of the SCL Signal

44.4.2 SDA Output Delay Function

The RIICHS module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (generating of the START, repeated START, and STOP conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing from a falling edge of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) of data hold time specified by the SMBus specification.

The output delay facility is enabled by setting the ICOCR.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not 000b), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits with the count source selected by the ICOCR.DLCS bit (internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). When the counting of the delay cycles is completed, the RIICHS output the SDA signal (generating of the START, repeated START, or STOP condition, data, or an ACK or NACK signal).

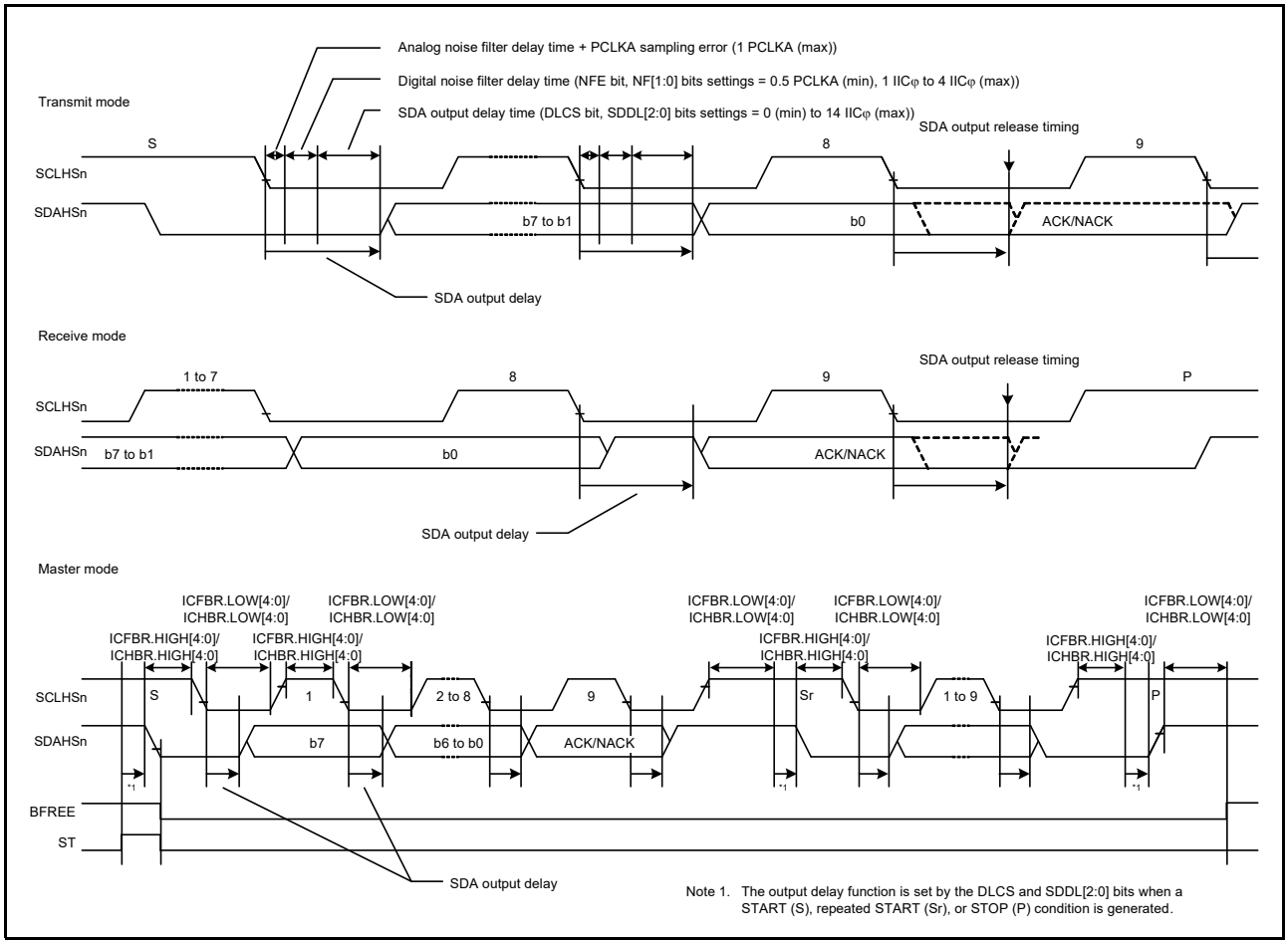


Figure 44.21 SDA Output Delay Function

44.4.3 Digital Noise Filters

The states of the SCLHSn and SDAHSn pins are conveyed to the internal circuitry through analog noise filter and digital noise filter. Figure 44.22 is a block diagram of the digital noise filter.

The on-chip digital noise filter of the RIICHS consists of 16 flip-flop circuit stages connected in series and a match-detection circuit. When Hs-mode is selected, only the first four flip-flops are enabled.

The number of effective stages in the digital noise filter is selected by the ICICR.NF[3:0] bits (ICICR.NF[3:2] bits in Hs-mode). The selected number of effective stages determines the noise-filtering capability as a period from 1 to 16 IICφ cycles (one to four IICφ cycles in Hs-mode).

The input signal from the SCLHSn pin (also the SDAHSn pin) is sampled on rising edges of the IICφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the ICICR.NF[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (I3CCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with I3CCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, disable the digital noise filter (by setting the ICICR.NFE bit to 0) and use only the analog noise filter.

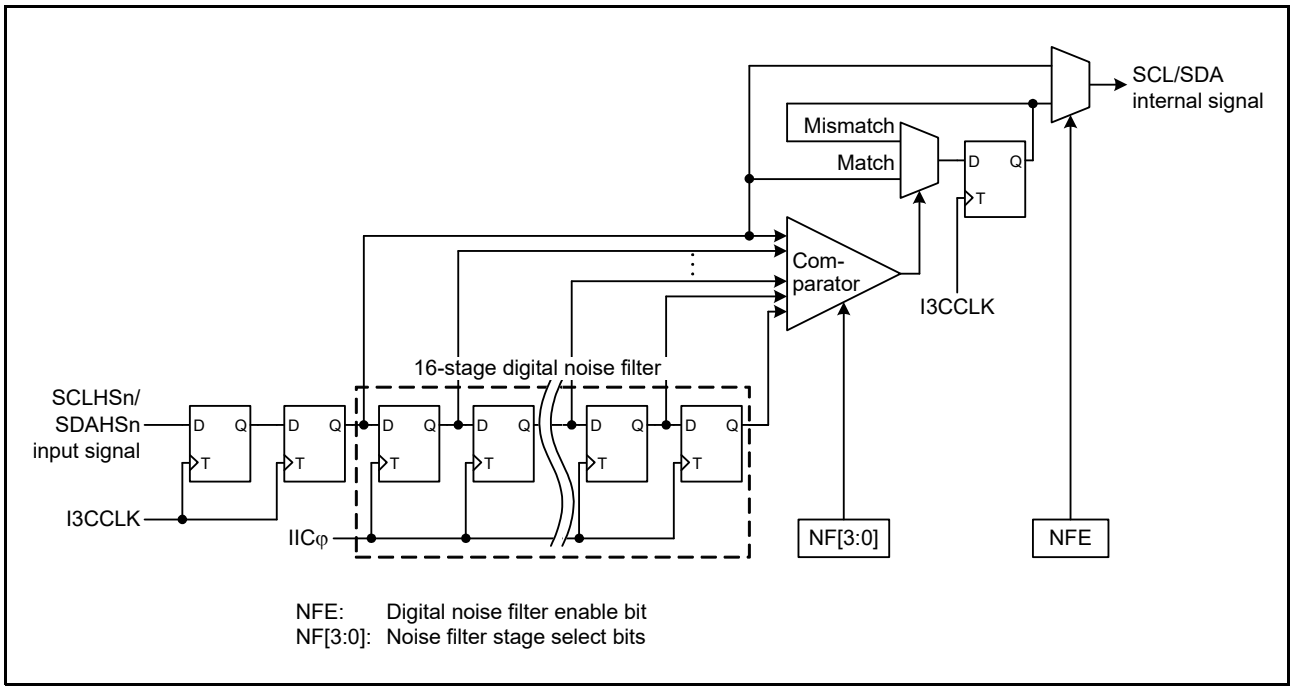


Figure 44.22 Block Diagram of Digital Noise Filter

44.4.4 Address Match Detection

The RIICHS can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

44.4.4.1 Slave-Address Match Detection

The RIICHS can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the ICSCR.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in SAMRy register (y = 0 to 2) can be detected.

When the RIICHS detects a match of the set slave address, the corresponding ICSSR.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL, and the ICCSR.RDRF flag or the ICCSR.TDRE flag is set to 1 according to the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 44.23 to Figure 44.25 show the AASy flag set timing in three cases.

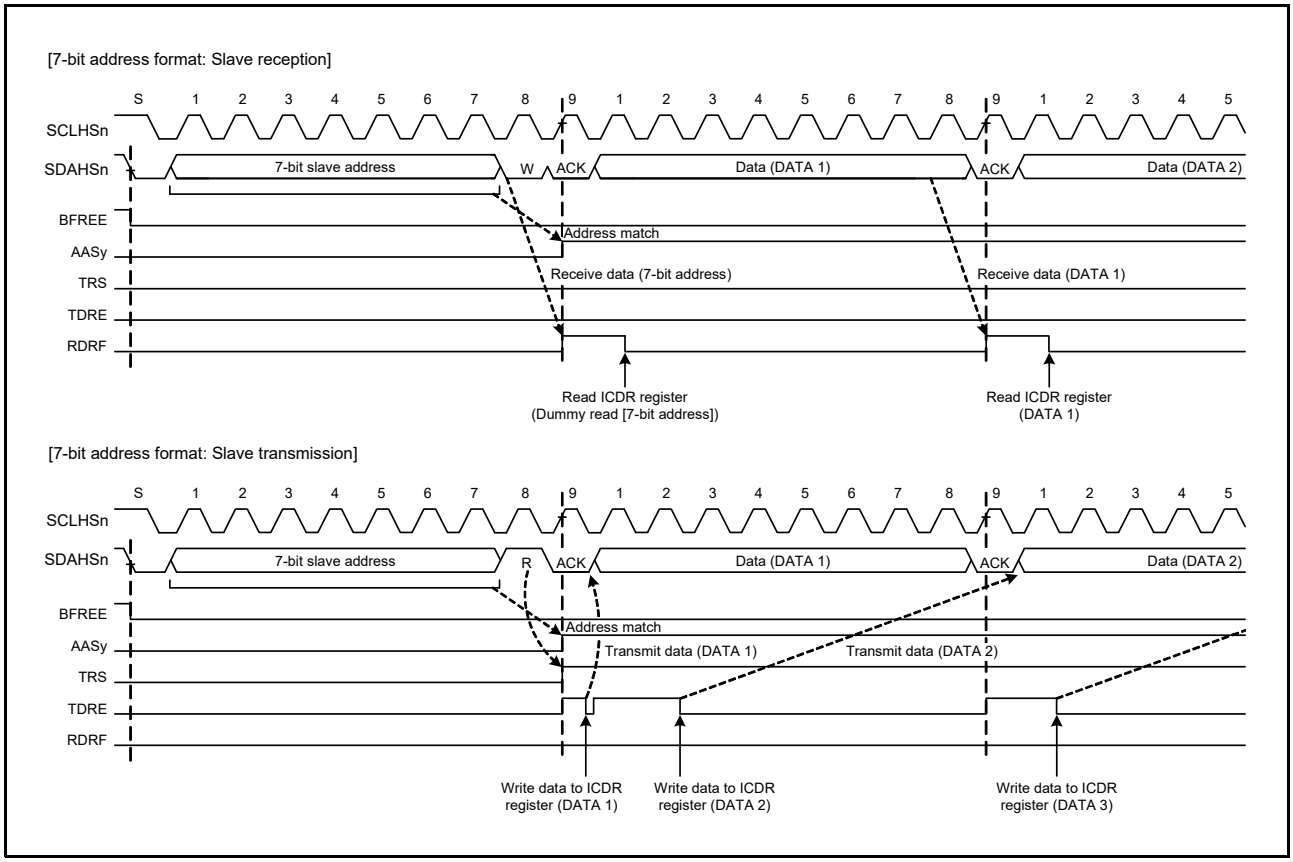


Figure 44.23 AASy Flag Set Timing with 7-Bit Address Format Selected

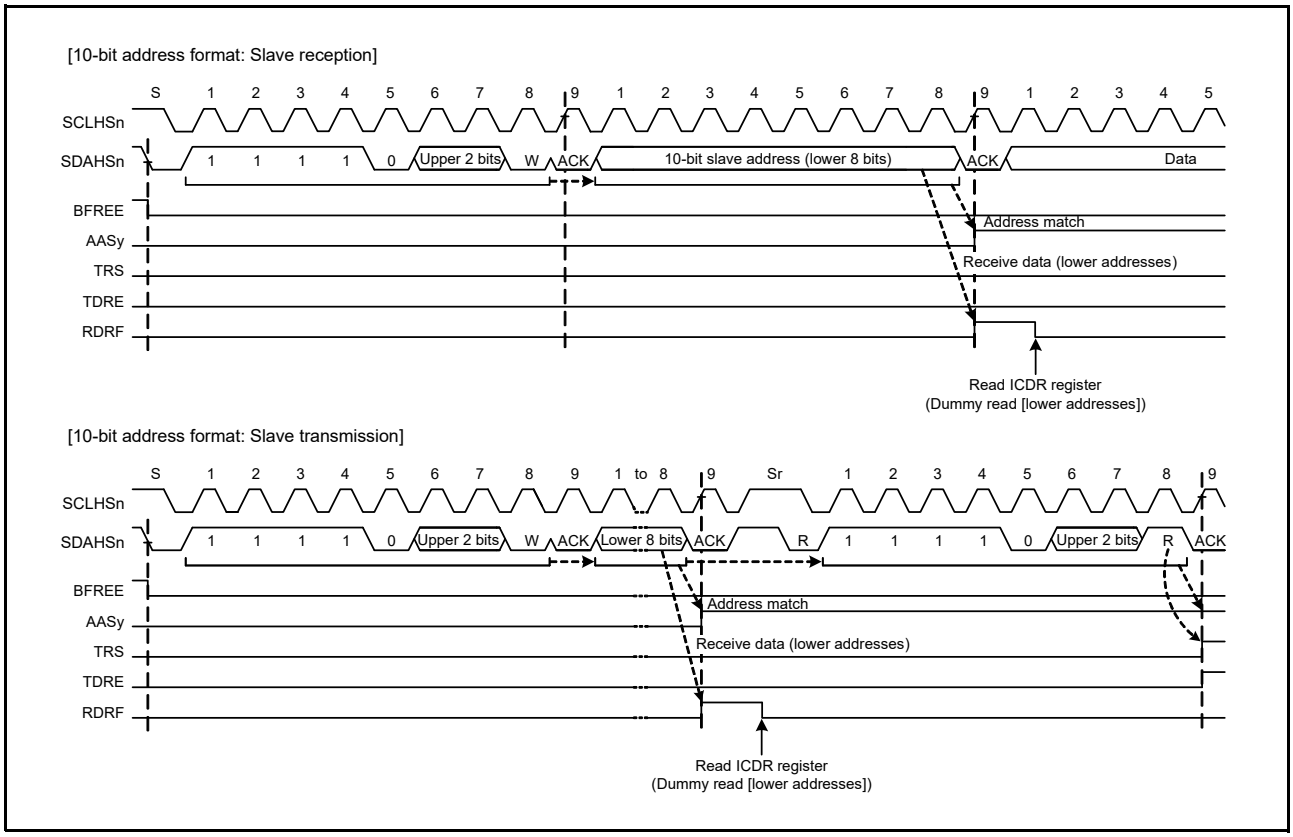


Figure 44.24 AASy Flag Set Timing with 10-Bit Address Format Selected

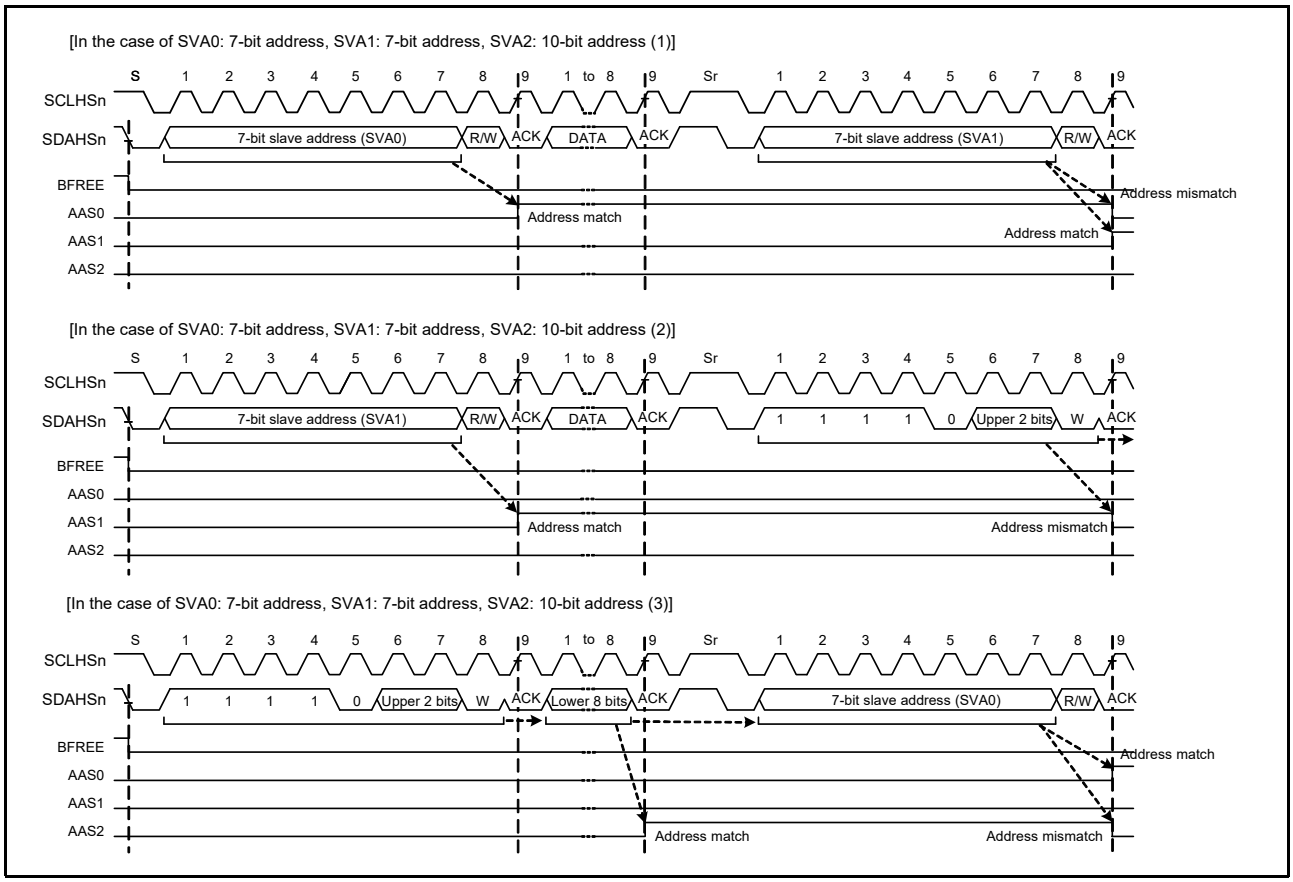


Figure 44.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

44.4.4.2 General Call Address Detection

The RIICHS has a function for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSCR.GCAE bit to 1.

If the address following a START or repeated START condition is 0000 000b + 1 (read) (start byte), the RIICHS recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIICHS detects the general call address, both the ICSSR.GCA flag and the ICCSR.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

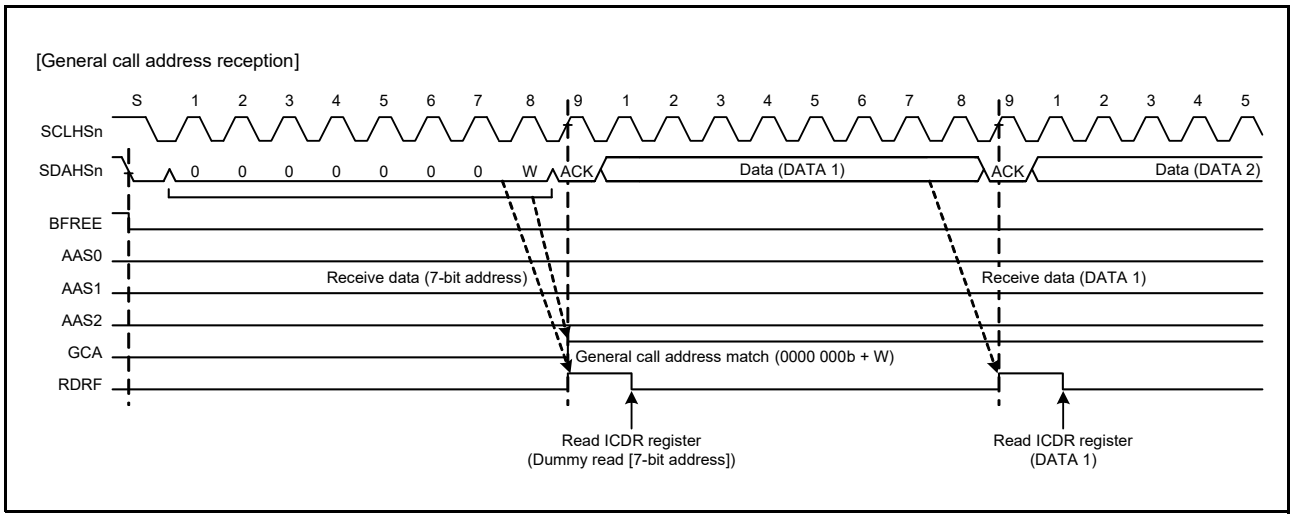


Figure 44.26 Timing of GCA Flag Setting during Reception of General Call Address

44.4.4.3 Device-ID Address Detection

The RIICHS module has a function for detecting device-ID addresses conformant with the I2C-bus specification.
When the RIICHS receives 1111 100b as the first seven bits of the first byte following a START condition or repeated START condition while the ICSCR.DIDE bit is 1, the RIICHS recognizes the address as a device ID address, sets the ICSSR.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIICHS sets the corresponding ICSSR.AASy flag (y = 0 to 2) to 1.
After that, when the first byte received after a START or repeated START condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIICHS does not compare the second and subsequent bytes and sets the ICCSR.TDRE flag to 1.
In the device-ID address detection function, the RIICHS sets the DID flag to 0 if a match with the RIICHS's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIICHS's own slave address and the detection of a repeated START condition. If the first byte after detection of a START or repeated START condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIICHS sets the DID flag to 1 and compares the second and subsequent bytes with the RIICHS's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIICHS does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that ICCSR.TDRE flag is 1.
Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

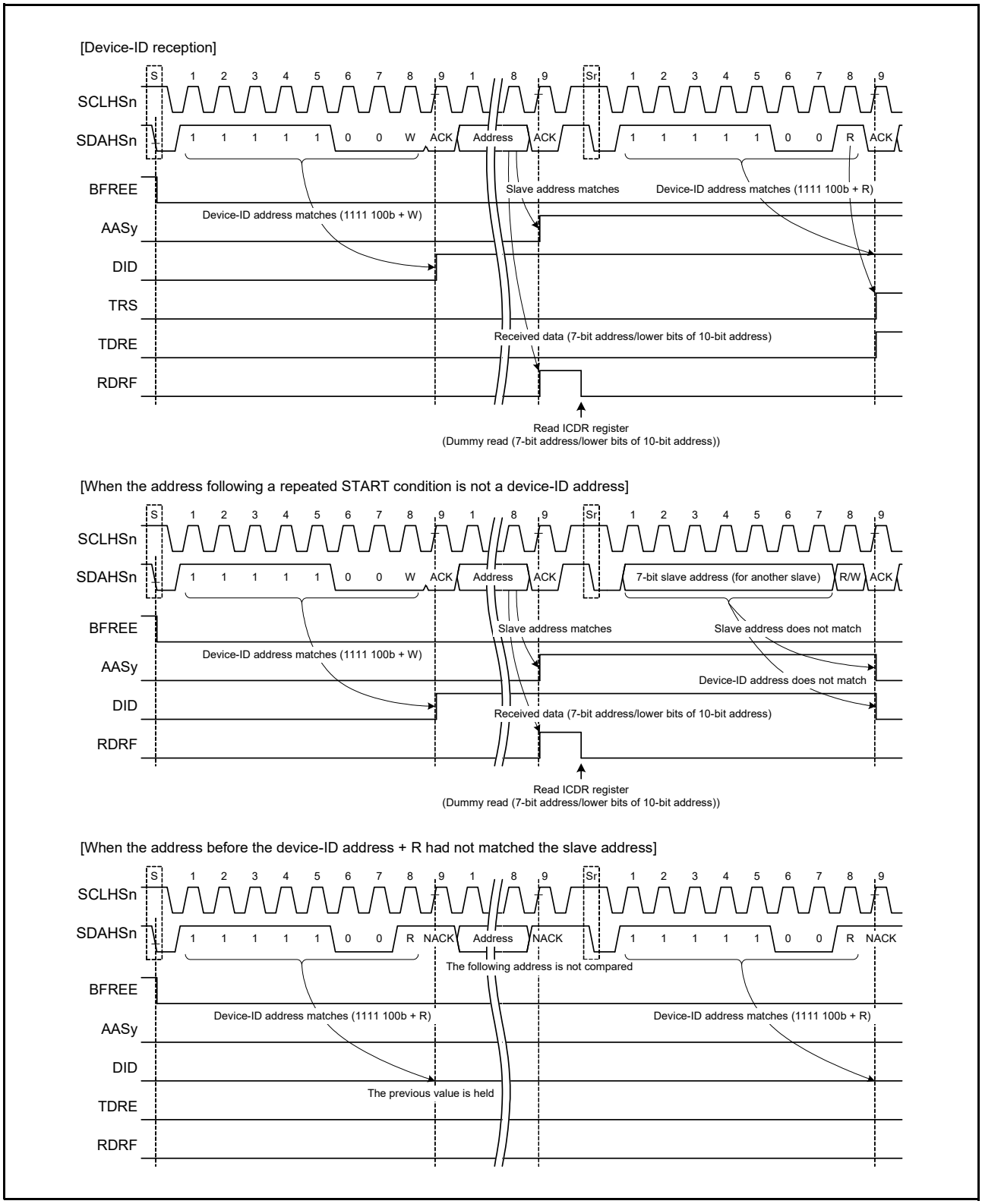


Figure 44.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

44.4.4.4 Hs-Mode Master Code Detection

The RIICHS has a function to detect the Hs-mode master code (0000 1XXXb).
When the RIICHS receives 0000 1XXXb as the first byte following a START condition while the ICSCR.HSMCE bit is 1, the RIICHS recognizes it as Hs-mode master code and set the ICSSR.HSMC flag to 1 at the rising edge of the ninth SCL. The RIICHS does not acknowledge the master code regardless of the value of the ICACKR.ACKBT bit (the SDAHSn pin remains high).
The first byte following a repeated START condition after the NACK bit is recognized as the slave address and compared with the SAMRy.SVA[9:0] bits (y = 0 to 2). When they match, the corresponding ICSSR.AASy flag is set to 1 at the rising edge of the ninth SCL. Also, the ICCSR.RDRF or TDRE flag is set to 1 depending on the value of the R/W# bit.
When a STOP condition is detected, the ICSSR.HSMC flag is set to 0.

Note: If the RIICHS receives Hs-mode master code while the ICSCR.HSMCE bit is 0, it ignores all data patterns until a STOP condition is detected.

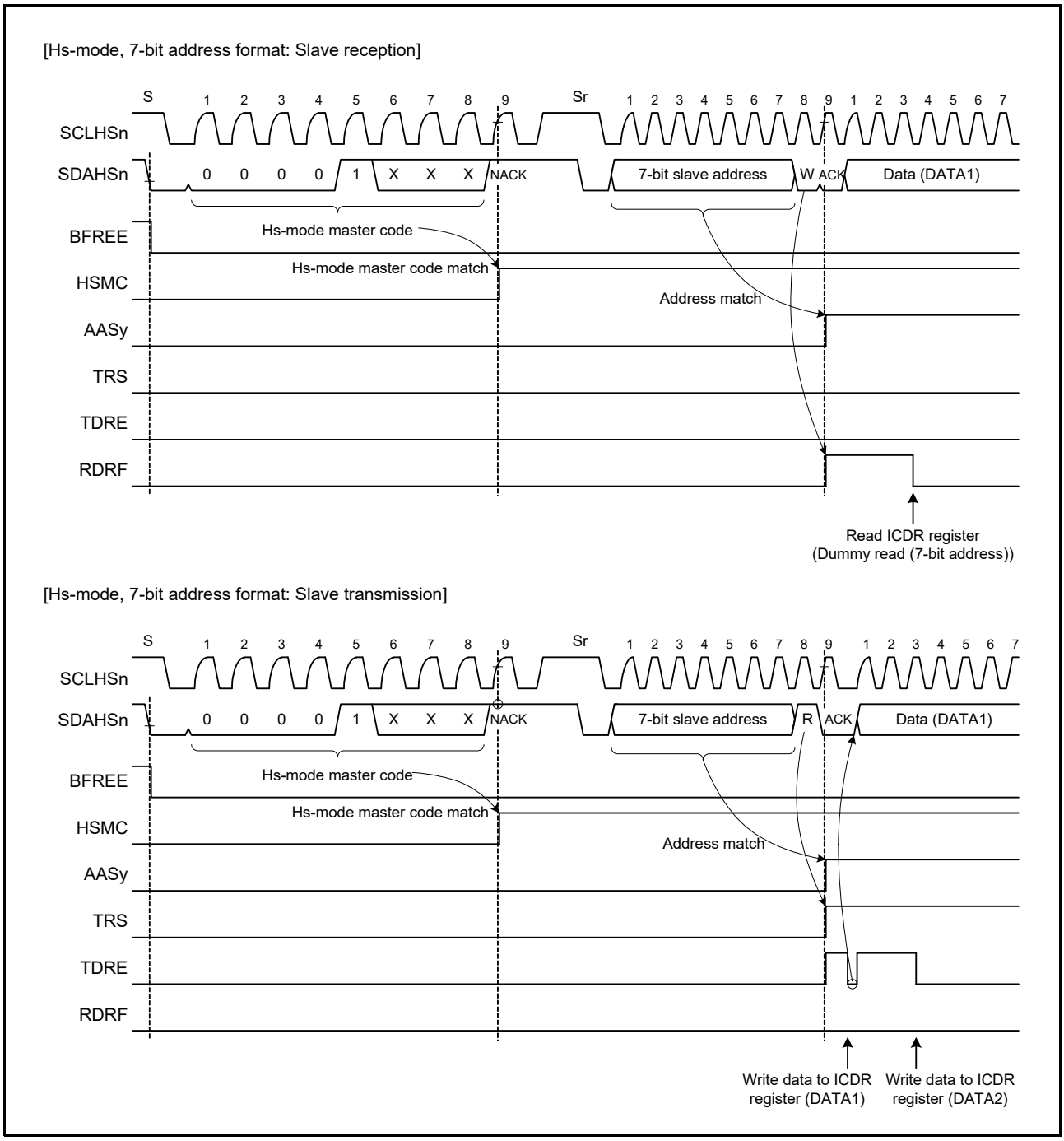


Figure 44.28 Set Timing of the HSMC and AASy Flags during Reception of Hs-Mode Master Code

44.4.4.5 Host Address Detection

The RIICHS has a function to detect the host address while the SMBus is operating. When the ICSCR.HOAE bit is set to 1 while the ICFER.SMBS bit is 1, the RIICHS can detect the host address (0001 000b) in slave receive mode (ICMMR.MST and ICMMR.TRS flags = 00b).

When the RIICHS detects the host address, the ICSSR.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICCSR.RDRF flag is set to 1 when the R/W# bit is 0 (write). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an R bit (R/W# bit = 1), the RIICHS can also detect the host address. After the host address is detected, the RIICHS operates in the same manner as normal slave operation.

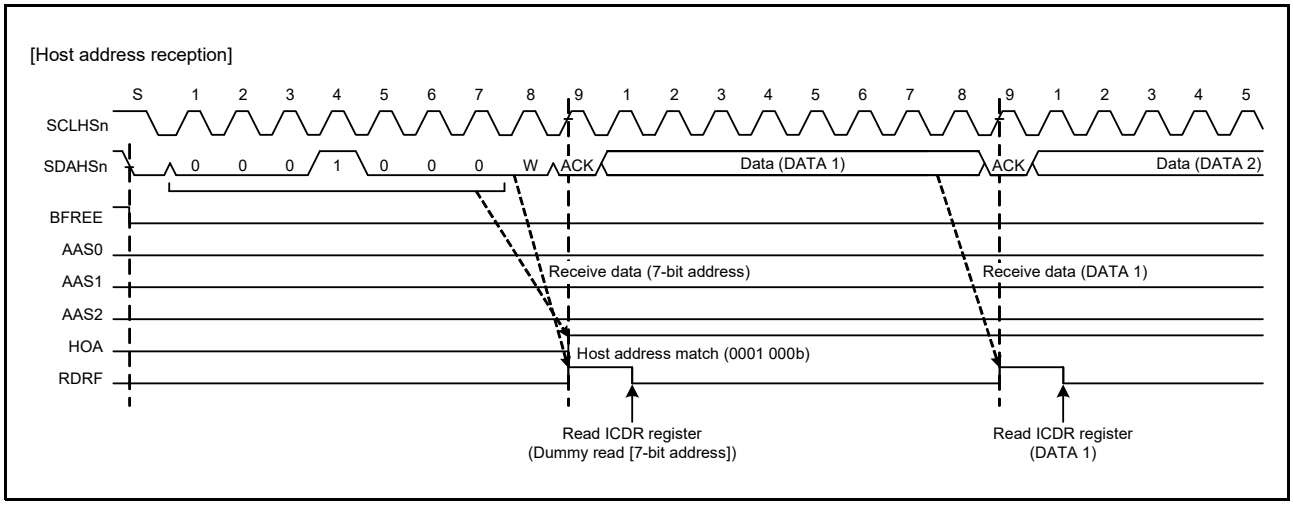


Figure 44.29 HOA Flag Set Timing during Reception of Host Address

44.4.5 Automatic Low-Hold Function for SCL

44.4.5.1 Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the transmit data register (ICDR register) with the RIICHS in transmission mode (ICMMR.TRS flag = 1), the SCLHSn line is automatically held low over the periods shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a START condition or repeated START condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

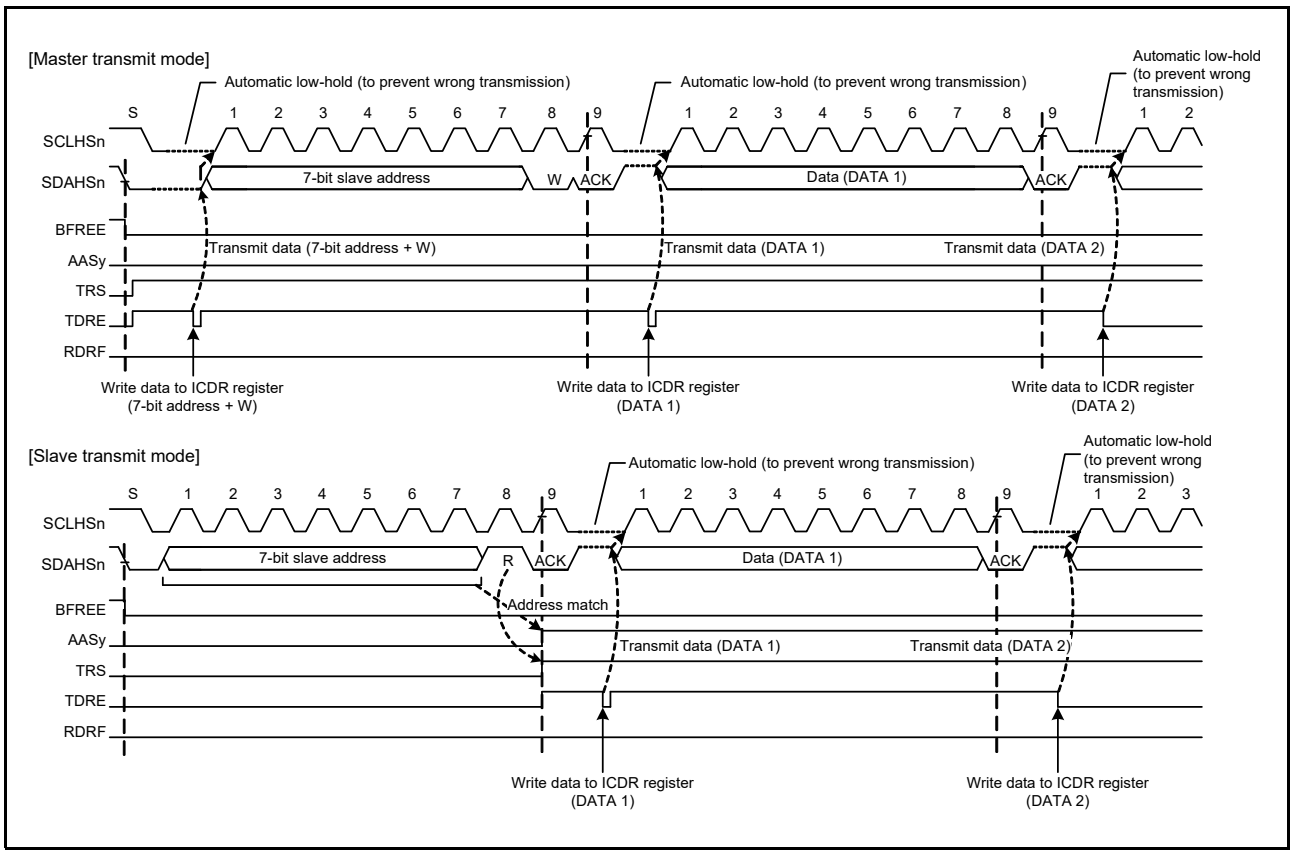


Figure 44.30 Automatic Low-Hold Operation in Transmit Mode

44.4.5.2 NACK Reception Transfer Suspension Function

The RIICHS has a function to suspend transfer operation when NACK is received in transmit mode (ICMMR.TRS flag = 1).

This function is enabled when the ICSER.NAKDE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICCSR.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDAHSn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (ICSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, issue a repeated START or STOP condition, set the NACKF flag to 0, and then issue a START condition again.

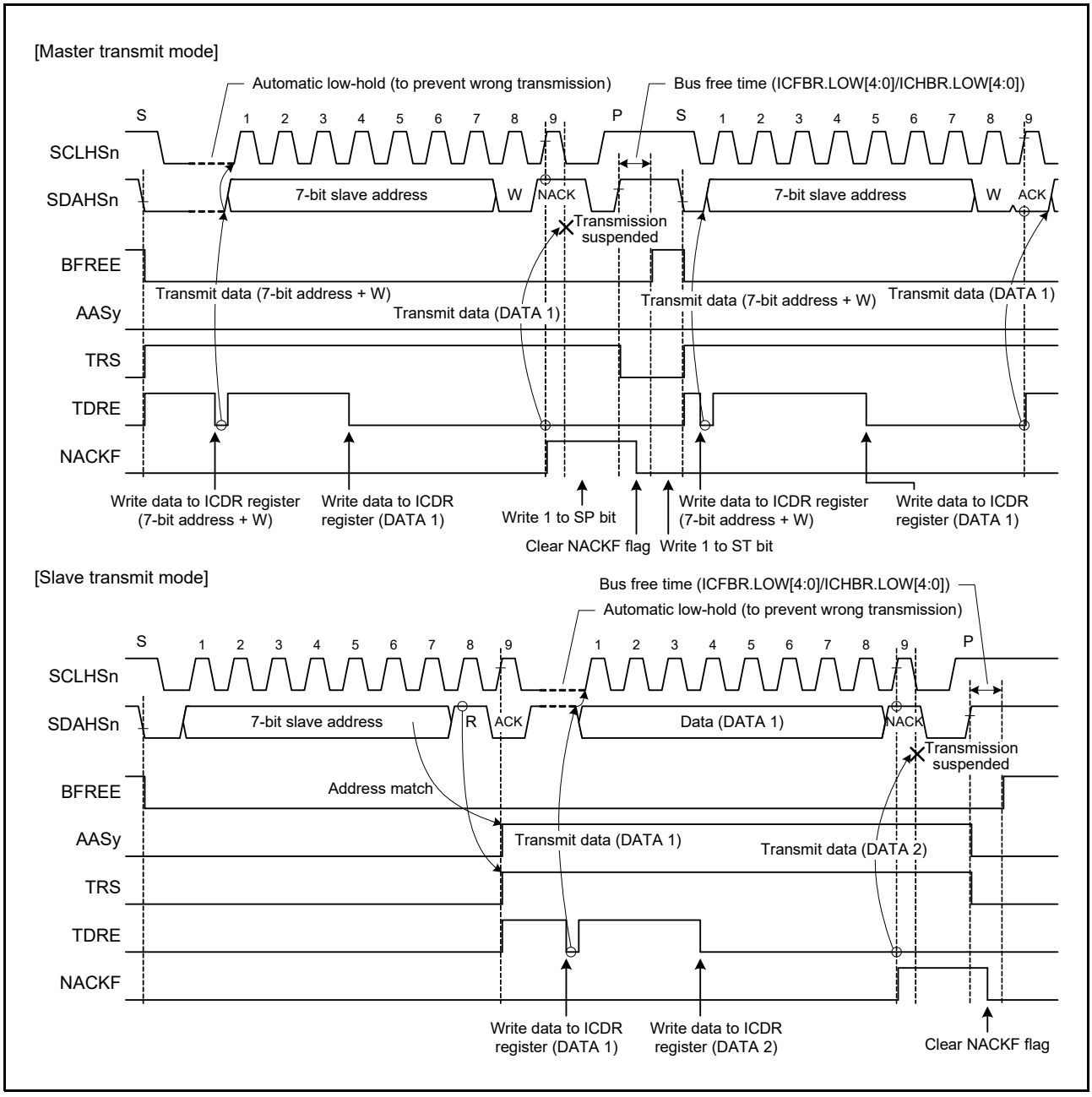


Figure 44.31 Suspension of Data Transmission When NACK is Received (NAKDE = 1)

44.4.5.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDR register) read is delayed for a period of one transfer frame or more with receive data full (ICCSR.RDRF flag = 1) in receive mode (ICMMR.TRS flag = 0), the RIICHS holds the SCLHSn line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIICHS's own slave address or a slave address for the other device is designated after a STOP condition is generated.
Sections in which the SCLHSn line is held low can be selected with a combination of the ICCSCR.WAITRE and ICCSCR.WAITAE bits.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAITRE Bit

When the ICCSCR.WAITRE bit is set to 1, the RIICHS performs 1-byte receive operation using the WAITRE bit function.
Furthermore, when the ICCSCR.WAITAE bit is 0, the RIICHS automatically sends the ICACKR.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCLHSn line low at the falling edge of the ninth SCL using the WAITRE bit function. This low-hold is released by reading data from ICDR register, which enables bytewise receive operation.
The WAITRE bit function is enabled for receive frames after a match with the RIICHS's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the WAITAE Bit

When the ICCSCR.WAITAE bit is set to 1, the RIICHS performs 1-byte receive operation using the WAITAE bit function.
When the WAITAE bit is set to 1, the ICCSR.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCLHSn line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICACKR.ACKBT bit, but cannot be released by reading data from ICDR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The WAITAE bit function is enabled for receive frames after a match with the RIICHS's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

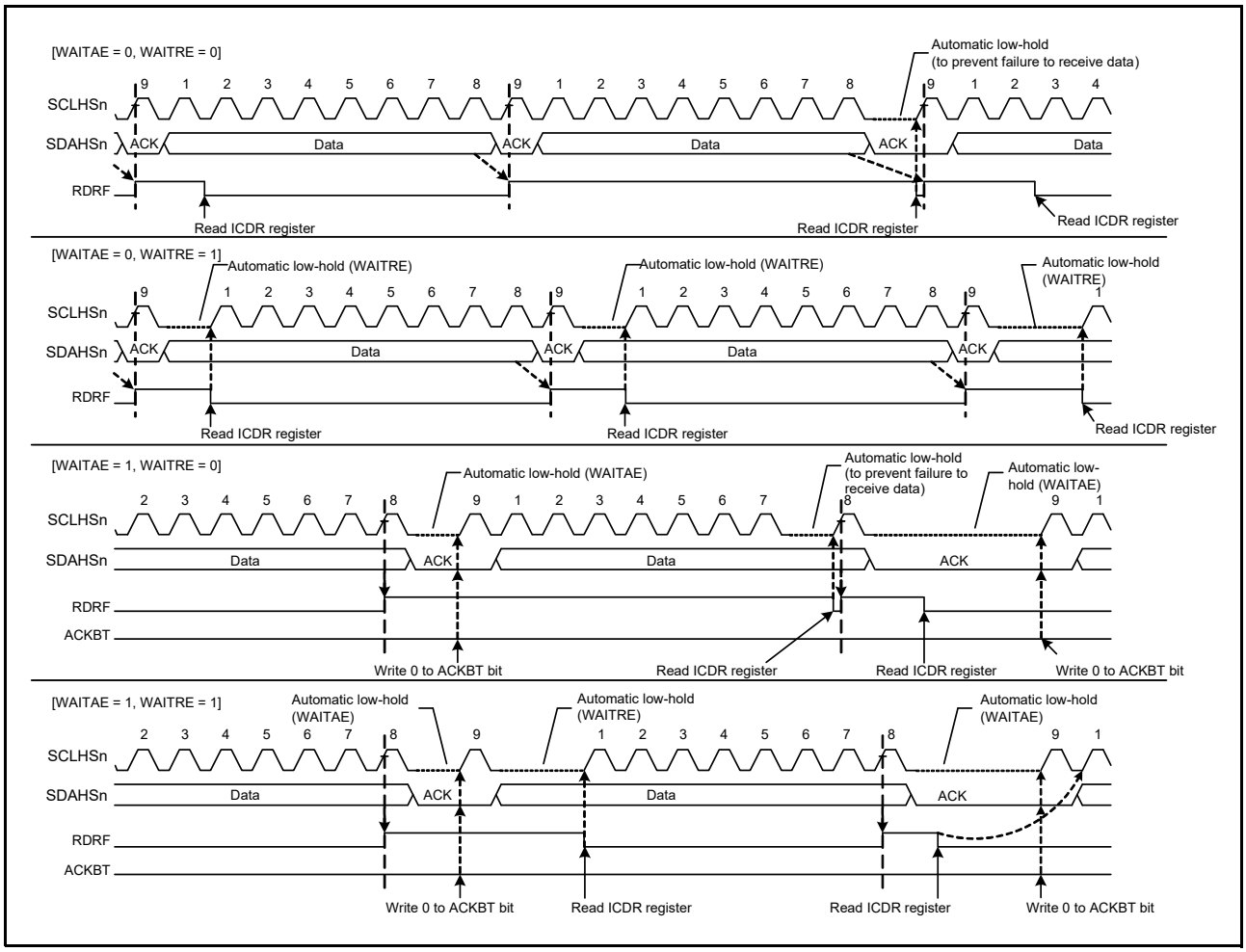


Figure 44.32 Automatic Low-Hold Operation in Receive Mode (Using WAITAE and WAITRE Bits)

44.4.6 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function specified in the I2C-bus specification, the RIICHS also has functions to prevent double generation of START conditions, to detect arbitration-lost during NACK transmission, and to detect arbitration-lost during slave transmission.

44.4.6.1 Master Arbitration-Lost Detection Function (MALE Bit)

The RIICHS drives the SDAHSn line low to generate a START condition. However, if the SDAHSn line is already driven low by another master device that generates a START condition, the RIICHS generates an arbitration-lost and gives a priority to the transfer by the other master device. Similarly, if the ICCGR.ST bit is set to 1 while the ICBSR.BFREE flag is 0 (bus busy), the RIICHS generates an arbitration-lost and gives a priority to the transfer by the other master device. No START condition is generated in this case.
After a START condition is generated successfully, if the level of the transmit data including the address bits (i.e. the internal SDA output level) and the level on the SDAHSn line do not match (the internal SDA output is high (the SDAHSn pin is high-impedance) but low is detected on the SDAHSn line), the RIICHS generates an arbitration-lost.
If a master arbitration-lost is generated, the RIICHS immediately transitions to slave receive mode. If a slave address including the general call address matches its own address at this time, the RIICHS continues the slave operation.
The RIICHS detects master arbitration-lost when one of the following conditions is met while the ICSER.ALE and ICFER.MALE bits are 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- When the internal SDA output level and the level on the SDAHSn line do not match (START condition generation error) when a START condition is generated by setting the ICCGR.ST bit to 1 while the ICBSR.BFREE flag is 1.
- When the ICCGR.ST bit is set to 1 while the BFREE flag is 0 (START condition double generation error).
- When the transmit data excluding acknowledgment bit (internal SDA signal output level) and the level on the SDAHSn line do not match in master transmit mode (the ICMMR.MST and ICMMR.TRS flags are 1).

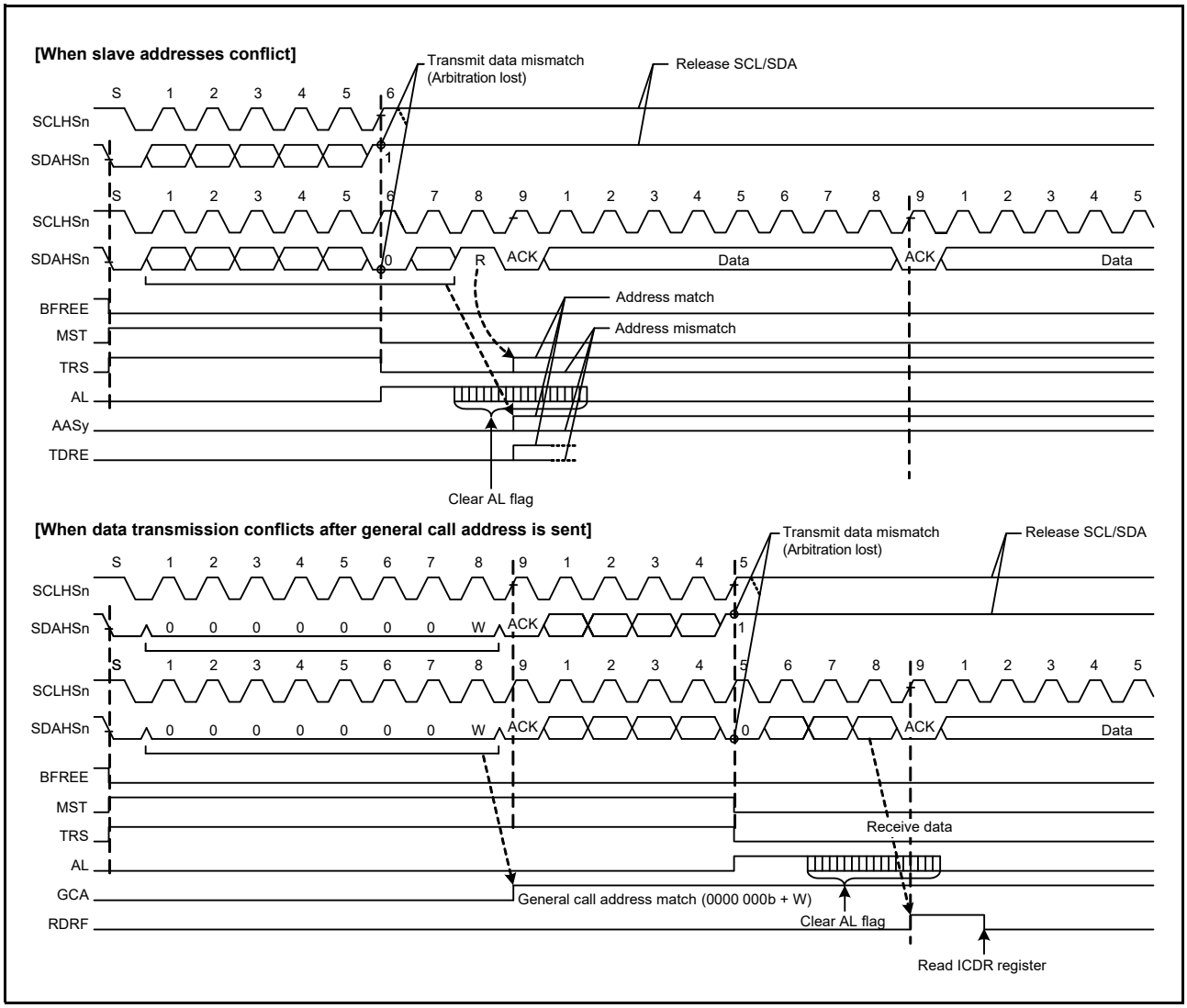


Figure 44.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

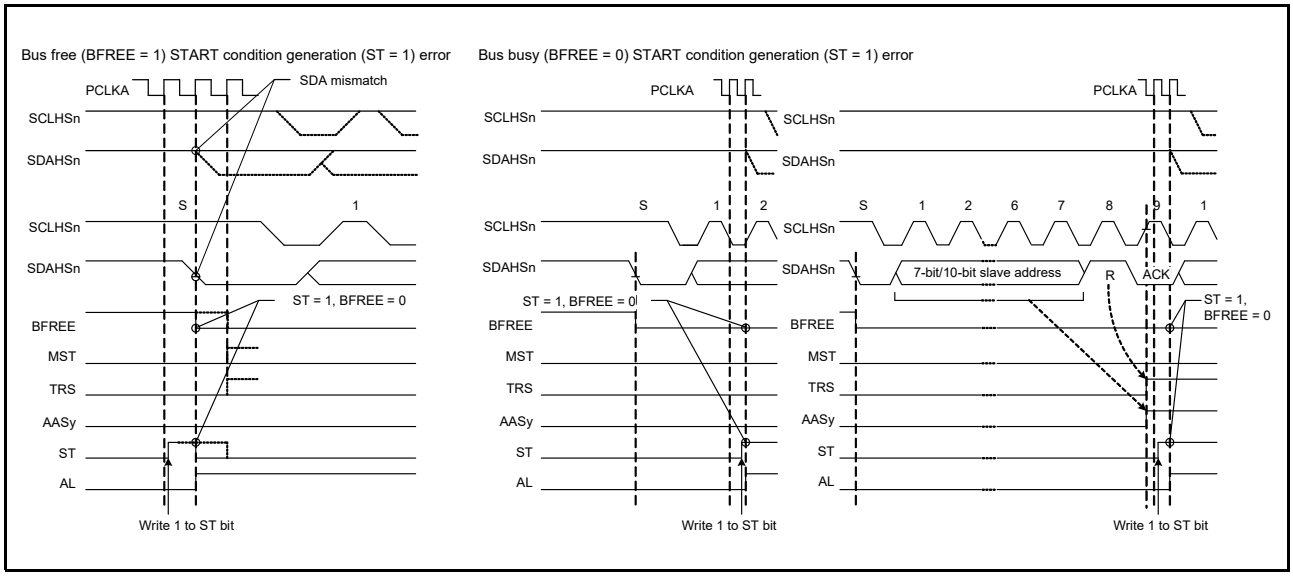


Figure 44.34 Arbitration-Lost Detection When a START Condition is Generated (MALE = 1)

44.4.6.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RIICHS has a function to generate an arbitration-lost if the internal SDA output level and the level on the SDAHSn line do not match (the internal SDA output is high (the SDAHSn pin is high-impedance) but low is detected on the SDAHSn line) during NACK transmission in receive mode. NACK transmission arbitration-lost is generated by a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflicts can occur when multiple master devices exchange common information through a single slave device.

Figure 44.35 shows an example of NACK transmission arbitration-lost detection when two master devices (master A and master B) and one slave device are connected to the bus.

In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device. If master A and master B access the slave device at the same time, arbitration-lost does not occur on either master A or master B when accessing the slave device because the slave addresses are the same. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends a NACK when it has received second byte, the last byte of data, from the slave device. On the other hand, master B sends an ACK because it has not yet received four bytes of data from the slave device. At this time, a conflict occurs between the NACK of master A and the ACK of master B.

If master A cannot detect the ACK sent by master B when such a conflict occurs, master A generates a STOP condition. This conflicts with the SCL output of master B and interferes with communication.

The RIICHS can generate an arbitration-lost if it receives an ACK while sending an NACK. This prevents the generation of a STOP condition and prevents bus communication interference. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after the NACK transmission when the UDID (unique device identification) of the “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RIICHS detects NACK transmission arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.NALE bits are 1 (NACK transmission arbitration-lost detection is enabled).

If the NACK transmission arbitration-lost occurs, the RIICHS cancels the slave address matched state and enters slave receive mode.

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDAHSn line (ACK is received) during transmission of NACK (the ICACKR.ACKBT bit is 1)

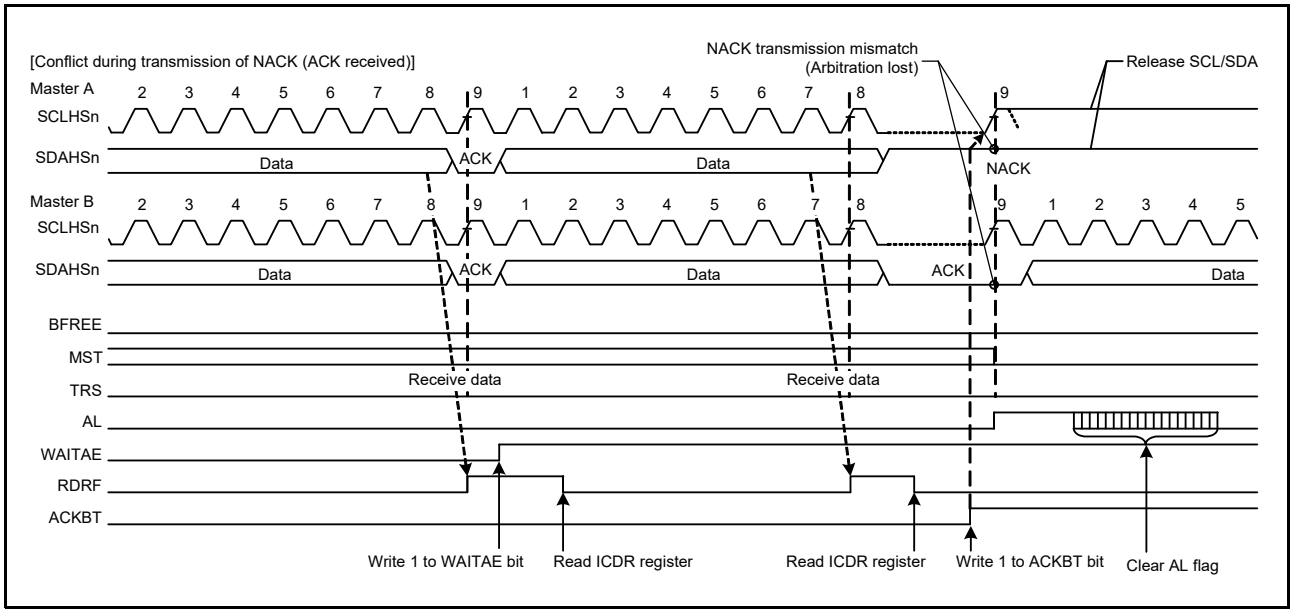


Figure 44.35 Example of NACK Transmission Arbitration-Lost Detection (NALE = 1)

44.4.6.3 Slave Arbitration-Lost Detection Function (SALE Bit)

The RIICHS has a function to generate an arbitration-lost if the transmit data (i.e. the internal SDA output level) and the level on the SDAHSn line do not match (the internal SDA output is high (the SDAHSn pin is high-impedance) but low is detected on the SDAHSn line) in slave transmit mode.

The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identification) of SMBus. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.

The RIICHS detects slave arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.SALE bits are 1 (slave arbitration-lost detection enabled).

If the slave arbitration-lost occurs, the RIICHS cancels the slave address matched state and enters slave receive mode.

Condition for detecting slave arbitration-lost

- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDAHSn line in slave transmit mode (the ICMMR.MST flag is 0 and the ICMMR.TRS flag is 1)

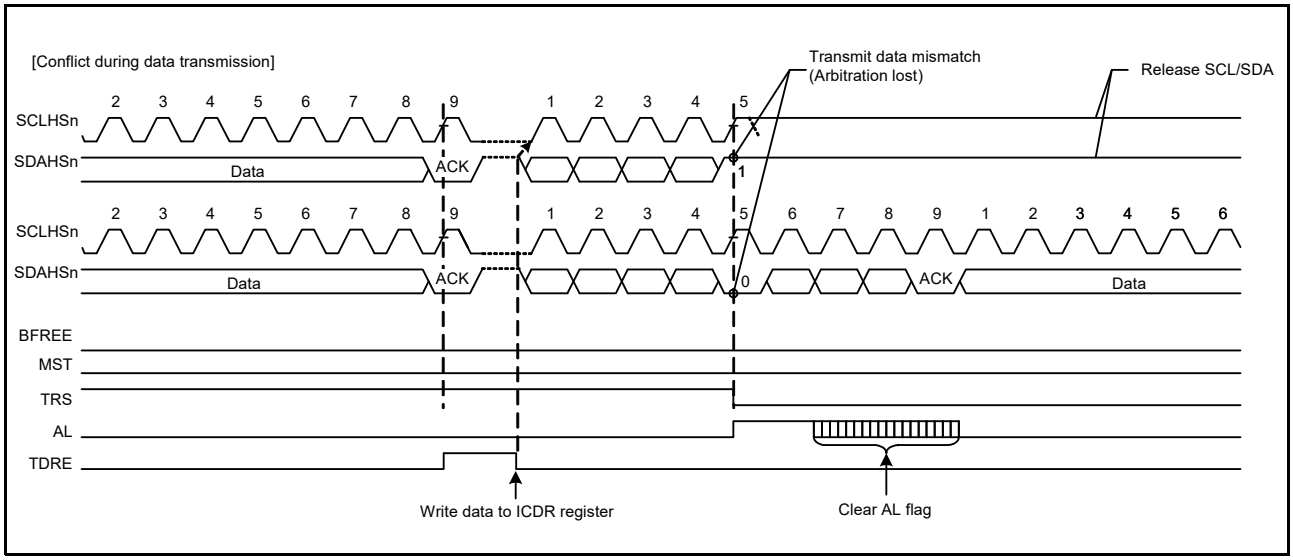


Figure 44.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

44.4.7 START Condition/Repeated START Condition/STOP Condition Generating Function

44.4.7.1 Generating a START Condition

The RIICHS generates a START condition when the ICCGR.ST bit is set to 1.
When the ST bit is set to 1 (requests to generate a START condition) while the ICBSR.BFREE flag is set to 1 (bus free state), the RIICHS generates a START condition.
When a START condition is generated normally, the RIICHS automatically shifts to the master transmit mode.
A START condition is generated in the following sequence.

START condition generation

(1) Drive the SDAHSn line low (high to low).
(2) Secure the START condition hold time with the time set in the ICFBR.HIGH[7:0] bits.
(3) Drive the SCLHSn line low (high to low).
(4) Detect a low of the SCLHSn line and secure the low period of SCLHSn line with the time set in the ICFBR.LOW[7:0] bits.

44.4.7.2 Generating a Repeated START Condition

The RIICHS generates a repeated START condition when the ICCGR.RS bit is set to 1.
When the RS bit is set to 1, generation of a repeated START condition is requested. When the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.MST flag is 1 (master mode), the RIICHS generates a repeated START condition.
A repeated START condition is generated in the following sequence.

Repeated START condition generation

(1) Release the SDAHSn line.
(2) Secure the low period of SCLHSn line with the time set in the ICFBR.LOW[7:0] bits.
(3) Release the SCLHSn line (low to high).
(4) Detect a high of the SCLHSn line and secure the repeated START condition setup time with the time set in the ICFBR.LOW[7:0] bits.
(5) Drive the SDAHSn line low (high to low).
(6) Secure the repeated START condition hold time with the time set in the ICFBR.HIGH[7:0] bits.
(7) Drive the SCLHSn line low (high to low).
(8) Detect a low of the SCLHSn line and secure the low period of SCLHSn line with the time set in the ICFBR.LOW[7:0] bits.

Note: When requesting generation of a repeated START condition, write the slave address to the ICDR register after confirming that the ICCGR.RS bit is 0. Data written while the ICCGR.RS bit is 1 is not transferred because it is considered to be before the repeated START condition is generated.

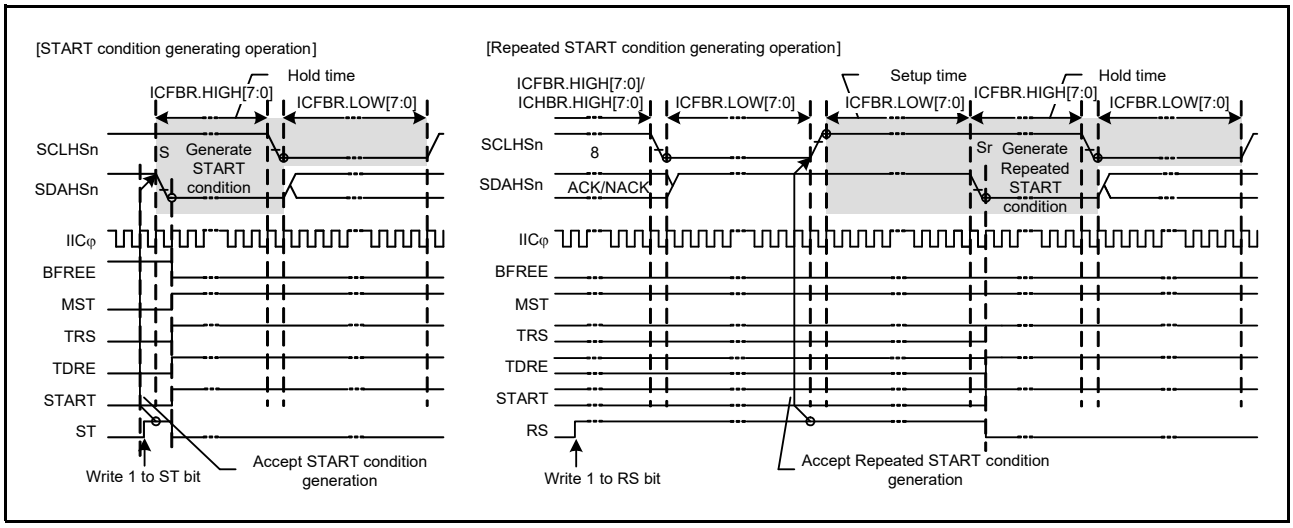


Figure 44.37 START Condition/Repeated START Condition Generating Timing (ST and RS Bits)

Figure 44.38 shows the operation timing when generating a repeated START condition after the master transmission.

Repeated START condition generation after the master transmission

(1) Initial setting. For details, refer to section 44.3.2, Initial Settings.

(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. At the same time, the BFREE flag is automatically set to 0, the ICSR2.START flag is automatically set to 1, and the ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAHSn line have matched while the ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ST bit has been successfully completed, and ICMMR.MST and ICMMR.TRS flags are automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.

(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDR register. Once the data for transmission are written to the ICDR register, the TDRE flag is automatically set to 0, the data are transferred from the ICDR register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS flag is automatically updated to select master transmit or master receive mode in accordance with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIICHS continues in master transmit mode. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to ICCGR.SP bit to generate a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDR register as the first address transmission. Then, as the second address transmission, write the lower 8 bits of the slave address to ICDR register.

(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RIICHS automatically holds the SCLHSn line low until the data for transmission are ready, a repeated START condition is generated, or a STOP condition is generated.

(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then, after check that the ICSR2.START flag is 1, set the ICSR2.START flag to 0.

(6) Set the ICCGR.RS bit to 1 (requests to generate a repeated START condition). Upon receiving the request, the RIICHS generates a repeated START condition.

(7) After check that the ICSR2.START flag is 1, write the value for transmission (the slave address and the R/W# bit) to the ICDR register.

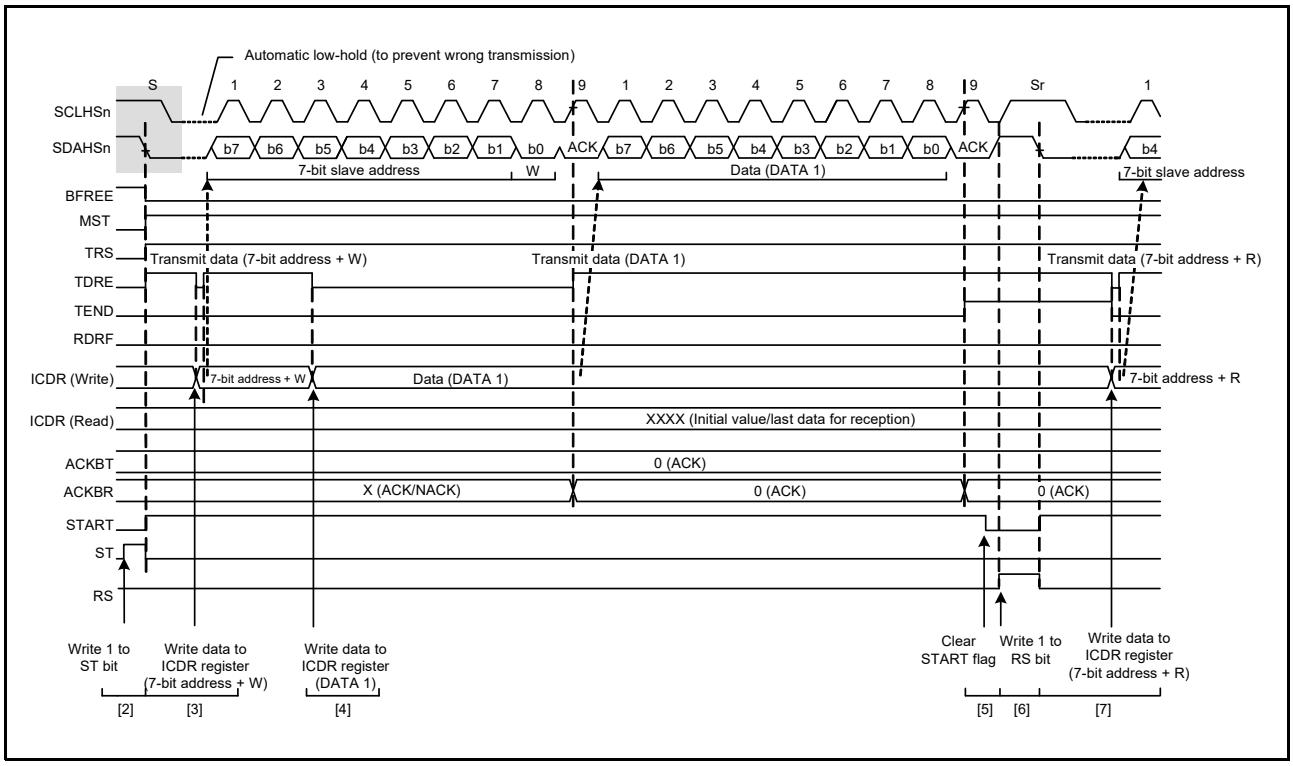


Figure 44.38 Repeated START Condition Generation Timing after the Master Transmission

44.4.7.3 Generating a STOP Condition

The RIICHS generates a STOP condition when the ICCGR.SP bit is set to 1.
When the SP bit is set to 1, generation of a STOP condition is requested. When the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.MST flag is 1 (master mode), the RIICHS generates a STOP condition.
A STOP condition is generated in the following sequence.

STOP condition generation

(1) Drive the SDAHSn line low (high to low).
(2) Secure the low period of SCLHSn line with the time set in the ICFBR.LOW[7:0] bits.
(3) Release the SCLHSn line (low to high).
(4) Detect a high of the SCLHSn line and secure the STOP condition setup time with the time set in the ICFBR.HIGH[7:0] bits.
(5) Release the SDAHSn line (low to high).
(6) Secure the bus free time with the time set in the ICFBR.LOW[7:0] bits.
(7) Set the BFREE flag to 1 (to release the bus).

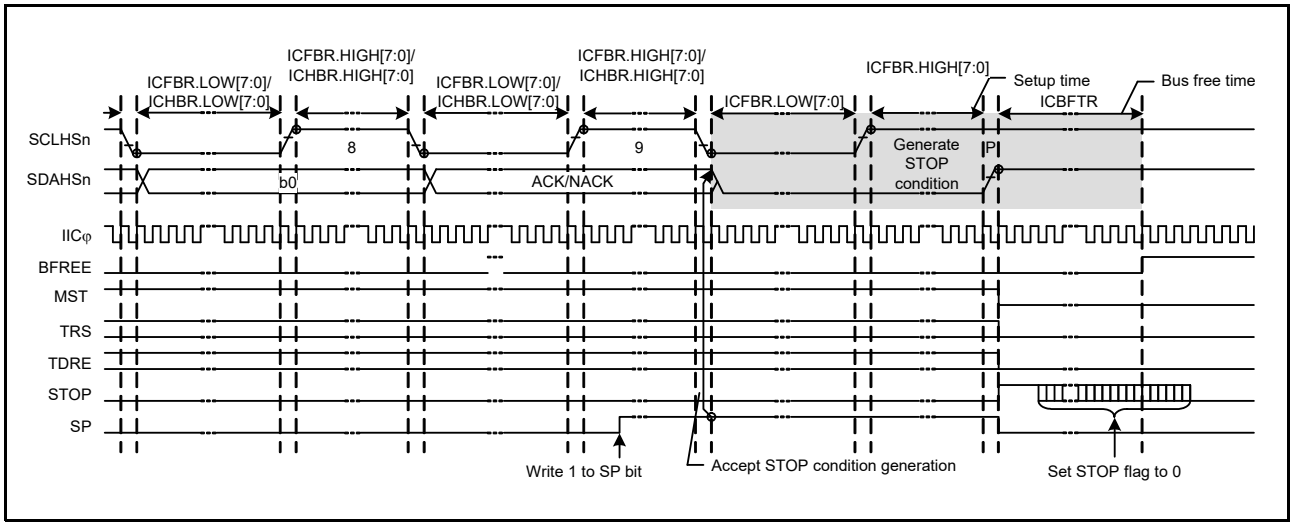


Figure 44.39 STOP Condition Generating Timing (SP Bit)

44.4.8 Bus Hanging

44.4.8.1 Timeout Function

The RIICHS includes a timeout function for detecting when the SCLHSn line has been stuck longer than the predetermined time. The RIICHS can detect an abnormal bus state by monitoring that the SCLHSn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLHSn line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCLHSn line changes (rising or falling), but continues to count unless the SCLHSn line changes. If the internal counter overflows due to no SCLHSn line change, the RIICHS can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICSER.TMOE bit is 1. It detects a hung state that the SCLHSn line is stuck low or high during the following conditions: (when ICTOR.TMOM[1:0] bits = 00b)

- The bus is busy (ICBSR.BFREE flag is 0) in master mode (ICMMR.MST flag is 1).
- The RIICHS's own slave address is detected (ICSSR register is not 0000 0000h) and the bus is busy (ICBSR.BFREE flag is 0) in slave mode (ICMMR.MST flag is 0).
- The bus is free (ICBSR.BFREE flag is 1) while generation of a START condition is requested (ICCGR.ST bit is 1).

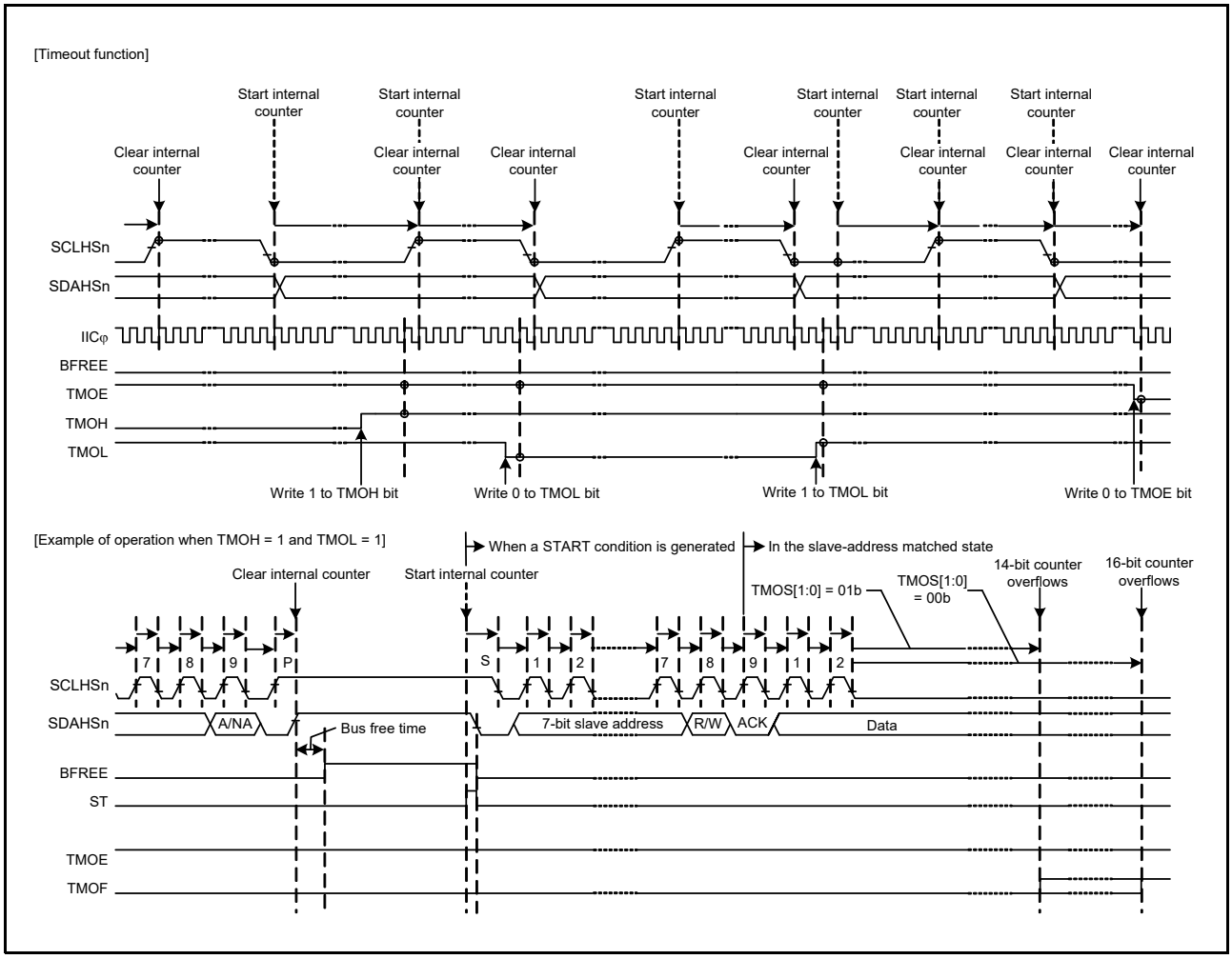


Figure 44.40 Timeout Function (TMOE, TMOS[1:0], TMOH, and TMOL Bits)

44.4.8.2 Additional SCL Output Function

In master mode, the RIICHS module has a function for the output of additional SCL to release the SDAHSn line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDAHSn line from the state of being stuck low by including additional SCL output from the RIICHS with single cycles of the SCL as the unit in the case of a bus error where the RIICHS cannot generate a STOP condition because the slave device is holding the SDAHSn line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICOCR.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICRCCR.CKS[2:0] bits and the HIGH[7:0] and LOW[7:0] bits in the ICFBR register (ICHBR register in Hs-mode) is output from the SCLHSn pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCLHSn pin is held low when the ICBSR.BFREE flag is 0 and held high when the BFREE flag is 1. Consecutive additional clock pulses can be output by the software program writing 1 to the CLO bit after having read CLO = 0.
When the RIICHS module is in master mode and the slave device is holding the SDAHSn line low because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a STOP condition is not possible. The additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDAHSn line from being held low, thus recovering the bus from an unusable state. Release of the SDAHSn line by the slave device can be monitored by reading the ICIMR.SDAI flag After confirming release of the SDAHSn line by the slave device, complete communications by regenerating the STOP condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled). If using this function with the ICFER.MALE bit set to 1, arbitration may be lost due to a mismatch between the value of the ICOCR.SDAO bit and the level of the SDAHSn line.

Conditions for using the ICOCR.CLO bit

- When the bus is free (ICBSR.BFREE flag = 1) or in master mode (ICMMR.MST flag = 1 and ICBSR.BFREE flag = 0)
- When the communication device does not hold the SCLHSn line low

Figure 44.41 shows the operation timing of the additional SCL output function (CLO bit).

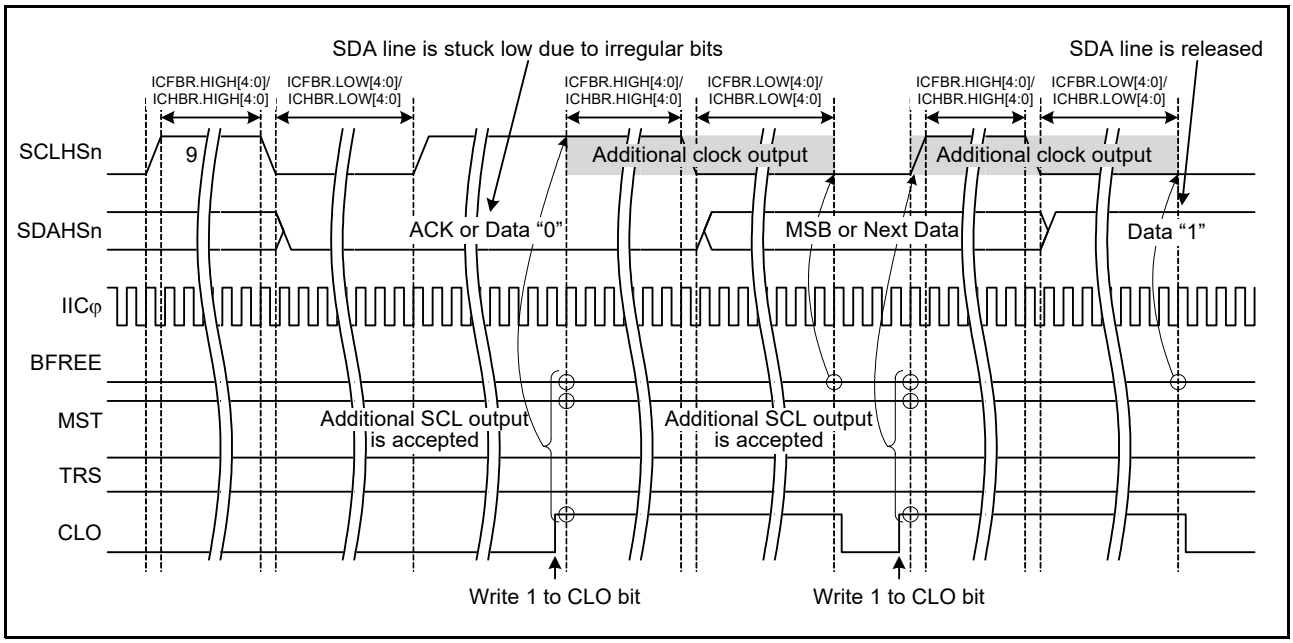


Figure 44.41 Additional SCL Output Function (CLO Bit)

44.4.9 Wake-Up Function

The RIICHS includes a wake-up function to return the MCU from low power consumption states. The wake-up function enables data reception even when the operating clock (I3CCLK) is stopped, and is used to generate a wake-up interrupt when the received address matches its own slave address.
To receive data even when the operating clock is stopped, the wake-up unit must be set to asynchronous operation. After a wake-up interrupt occurs, return to synchronous operation so that communication can continue.
There are four wake-up response modes, as shown in Table 44.6.

Table 44.6 Wake-Up Response Modes

Response Mode	Response Timing	Response	SCL Status
Normal wake-up mode 1	Before returning to synchronous operation	ACK	Hold low after ACK response (at the ninth SCL falling edge), release after returning to synchronous operation.
Normal wake-up mode 2	After returning to synchronous operation	ACK	Hold low at the eighth SCL falling edge, release after returning to synchronous operation with ACK response.
Command recovery mode	Before returning to synchronous operation	ACK	Open.
EEP response mode	Before returning to synchronous operation	NACK	Open.

There are five wake-up interrupt sources.

- Host address detection (ICFER.SMBS bit = 1, ICSCR.HOAE bit = 1)
- General call address detection (ICSCR.GCAE bit = 1)
- Slave address 0 detection (ICSCR.SAR0E bit = 1, SAR0.FS bit = 0)
- Slave address 1 detection (ICSCR.SAR1E bit = 1, SAR1.FS bit = 0)
- Slave address 2 detection (ICSCR.SAR2E bit = 1, SAR2.FS bit = 0)

44.4.9.1 Normal Wake-Up Mode 1

In normal wake-up mode 1, the RIICHS operates as follows after a wake-up interrupt occurs.

(1) Generates an ACK for the data byte that contains its own slave address.
(2) Holds the SCL line low at the ninth SCL falling edge.
(3) When 1 is written to the ICWUCR.WUFSYNE bit, it returns to synchronous operation and sets the ICWUSR.WUASYNF flag to 0.
(4) Releases the SCL line and returns to normal operation.

Note: When returning from low power consumption states due to wake-up interrupt, the setting of the ICCSCR.WAITRE bit is ignored.

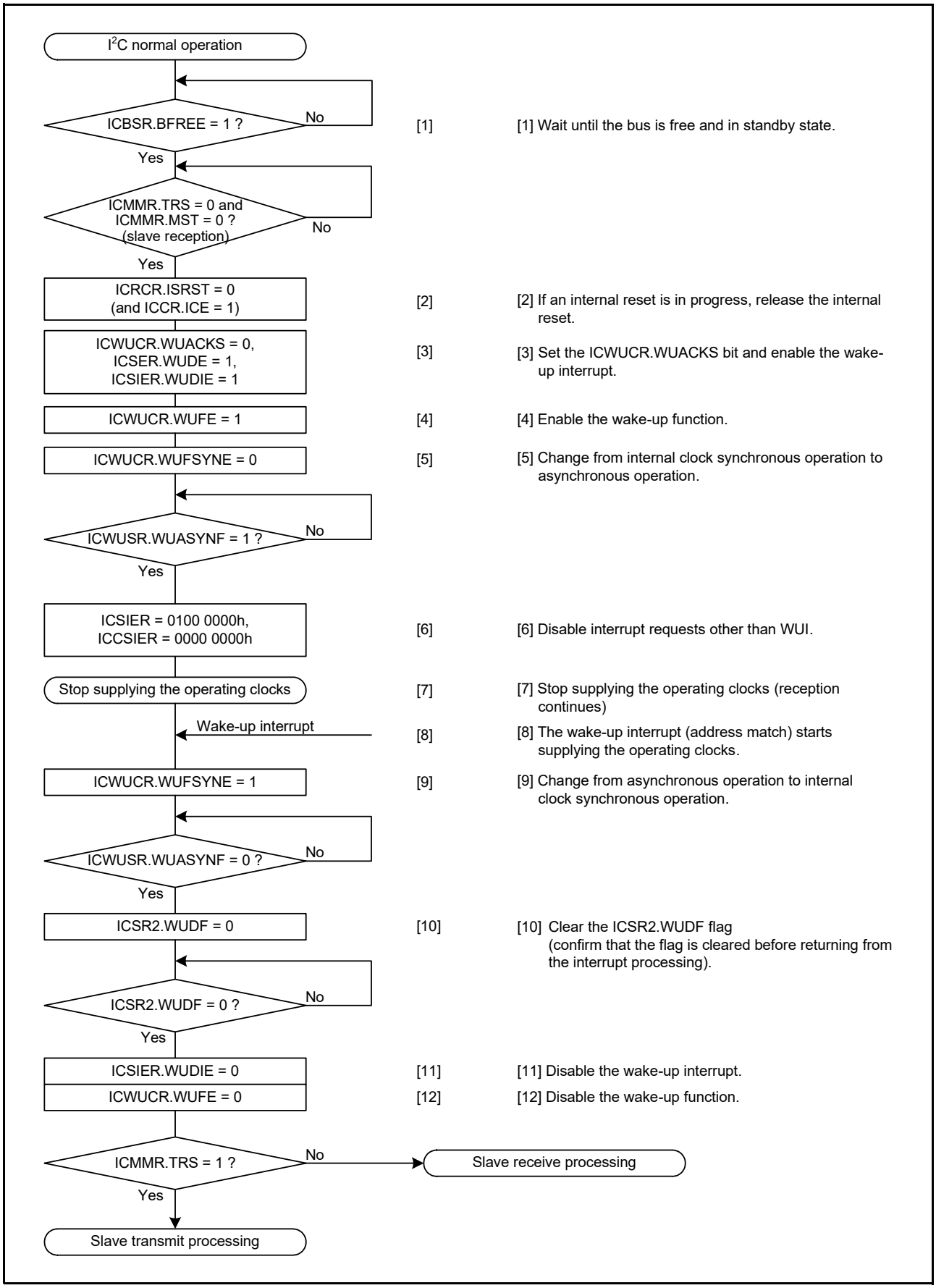


Figure 44.42 Example of Using Normal Wake-Up Mode 1

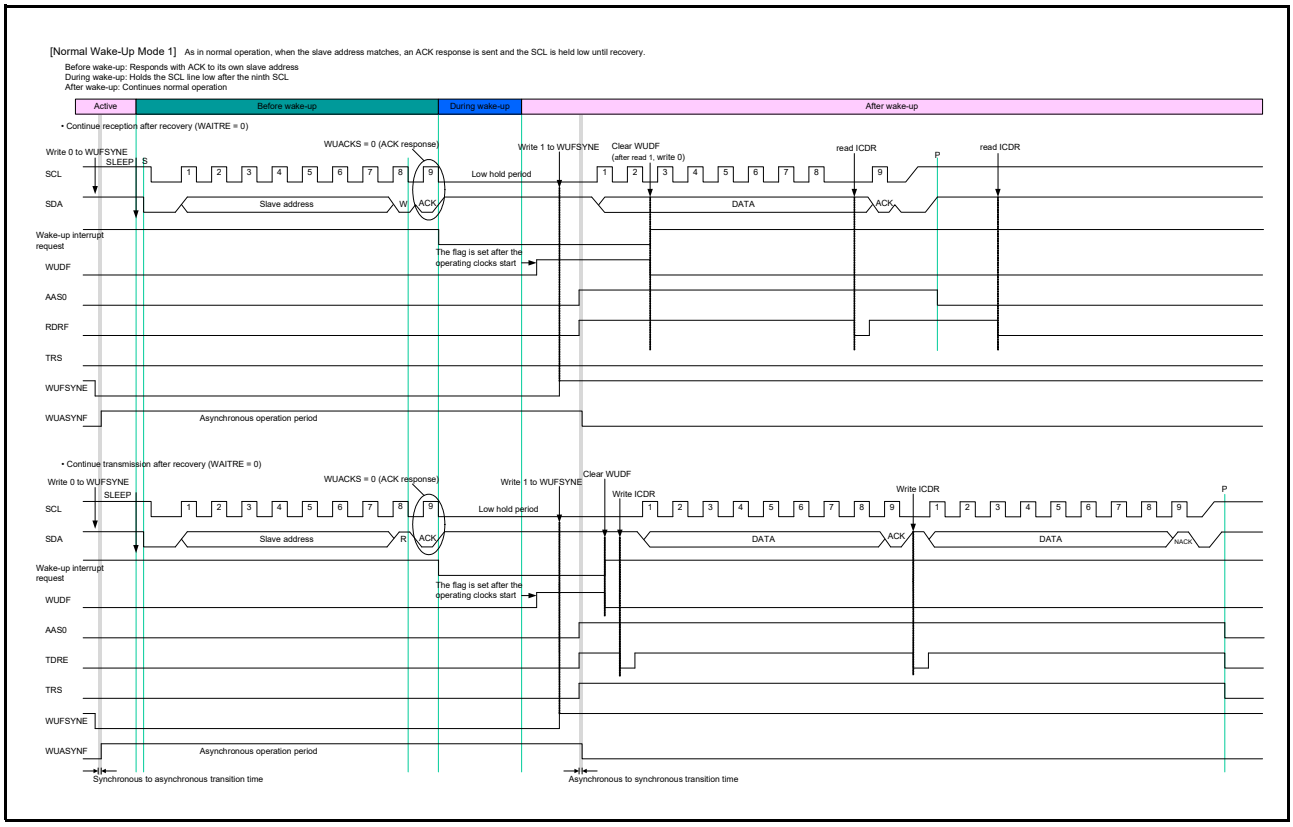


Figure 44.43 Operation Timing of Normal Wake-Up Mode 1

44.4.9.2 Normal Wake-Up Mode 2

In normal wake-up mode 2, the RIICHS operates as follows after a wake-up interrupt occurs.

(1) Holds the SCL line low at the eighth SCL falling edge for the data byte that contains its own slave address.
(2) When 1 is written to the ICWUCR.WUFSYNE bit, it returns to synchronous operation and sets the ICWUSR.WUASYNF flag to 0.
(3) Generates an ACK and release the SCL line.
(4) Returns to normal operation.

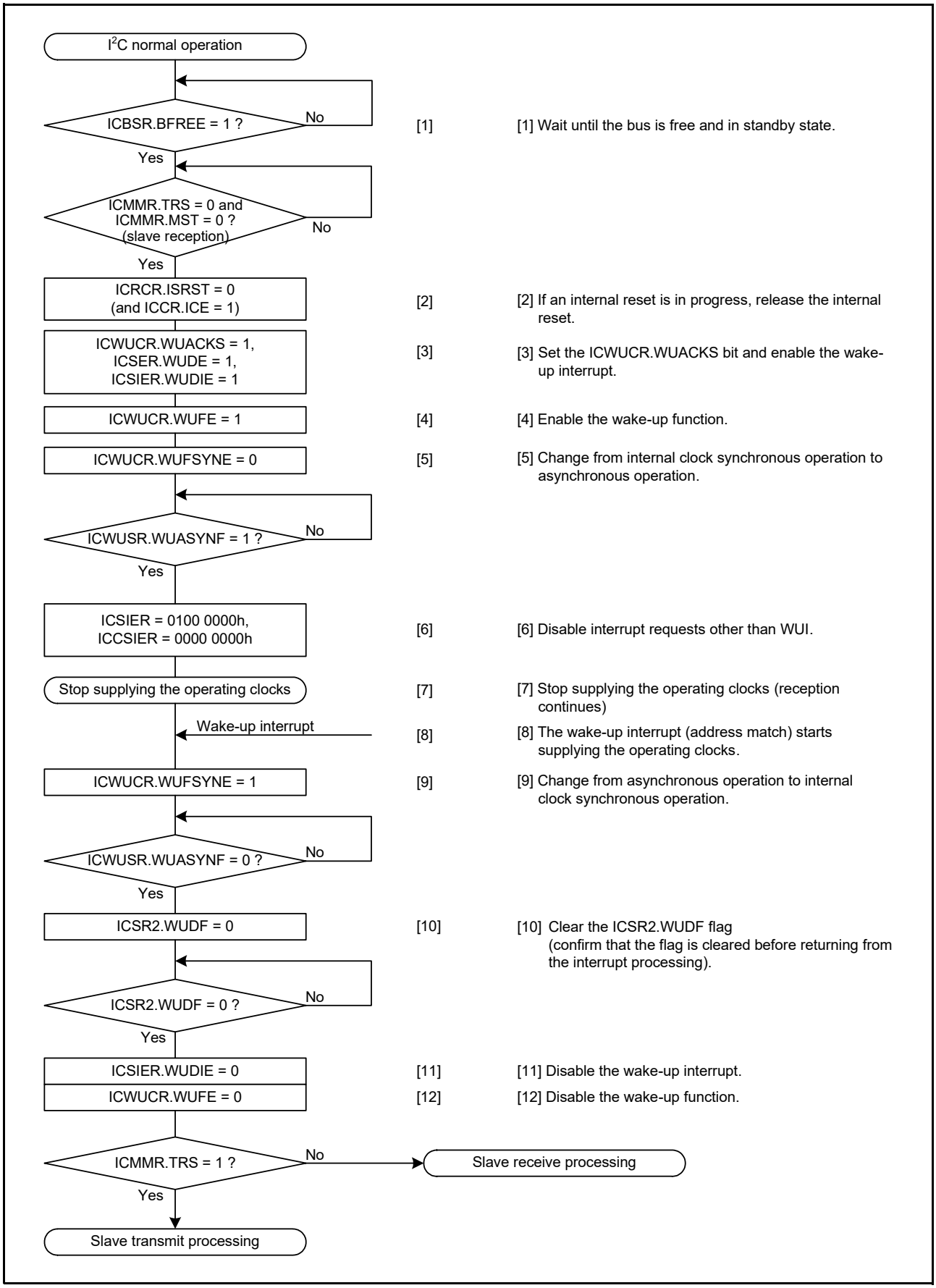


Figure 44.44 Example of Using Normal Wake-Up Mode 2

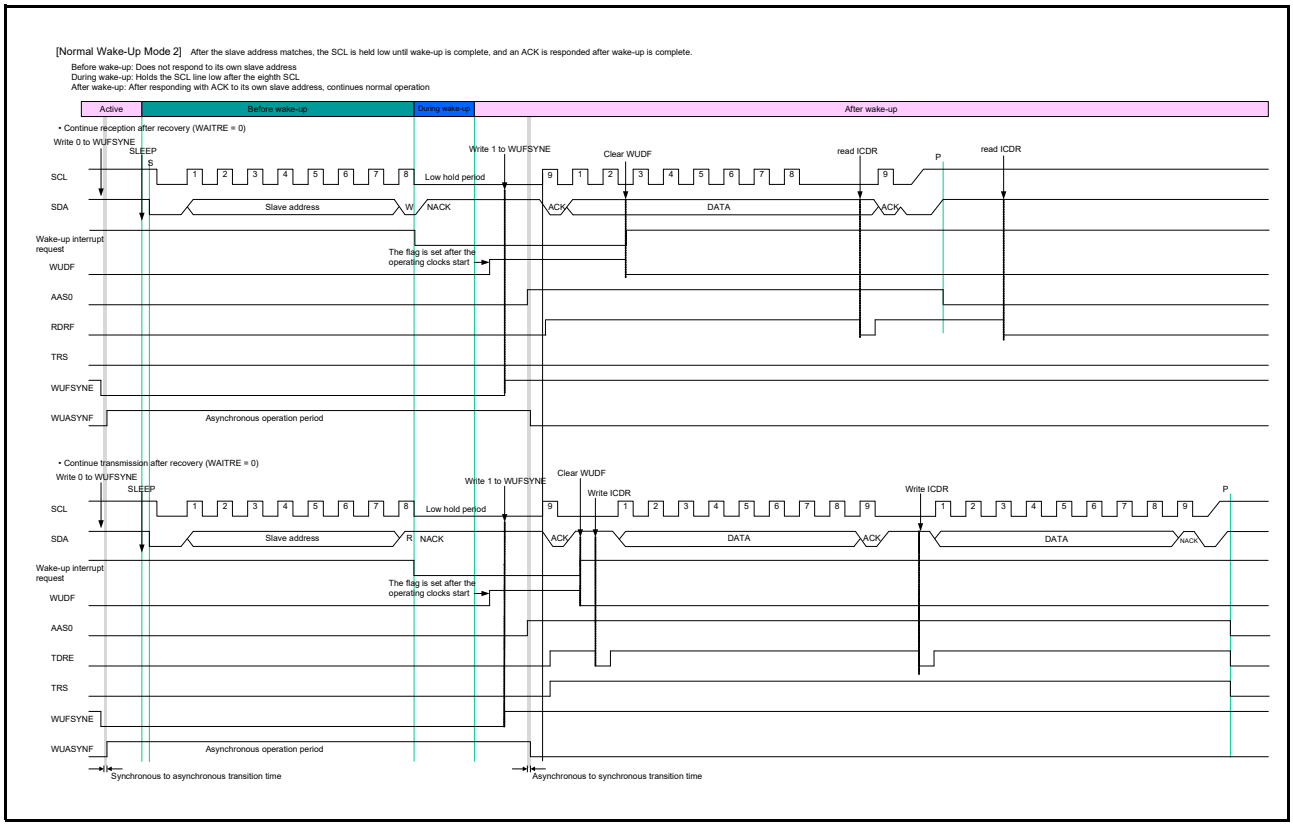


Figure 44.45 Operation Timing of Normal Wake-Up Mode 2

44.4.9.3 Command Recovery Mode and EEP Response Mode

In command recovery mode and EEP response mode, the SCL line is not held low during wake-up operations. As a result, other devices can use the bus during this period, but the RIICHS cannot perform subsequent data transmission and reception.

In these modes, the ICRCR.ISRST bit is set to 1, so after returning from low power consumption states, set the ICRCR.MRST bit to 1 (module reset) and then perform the initial settings.

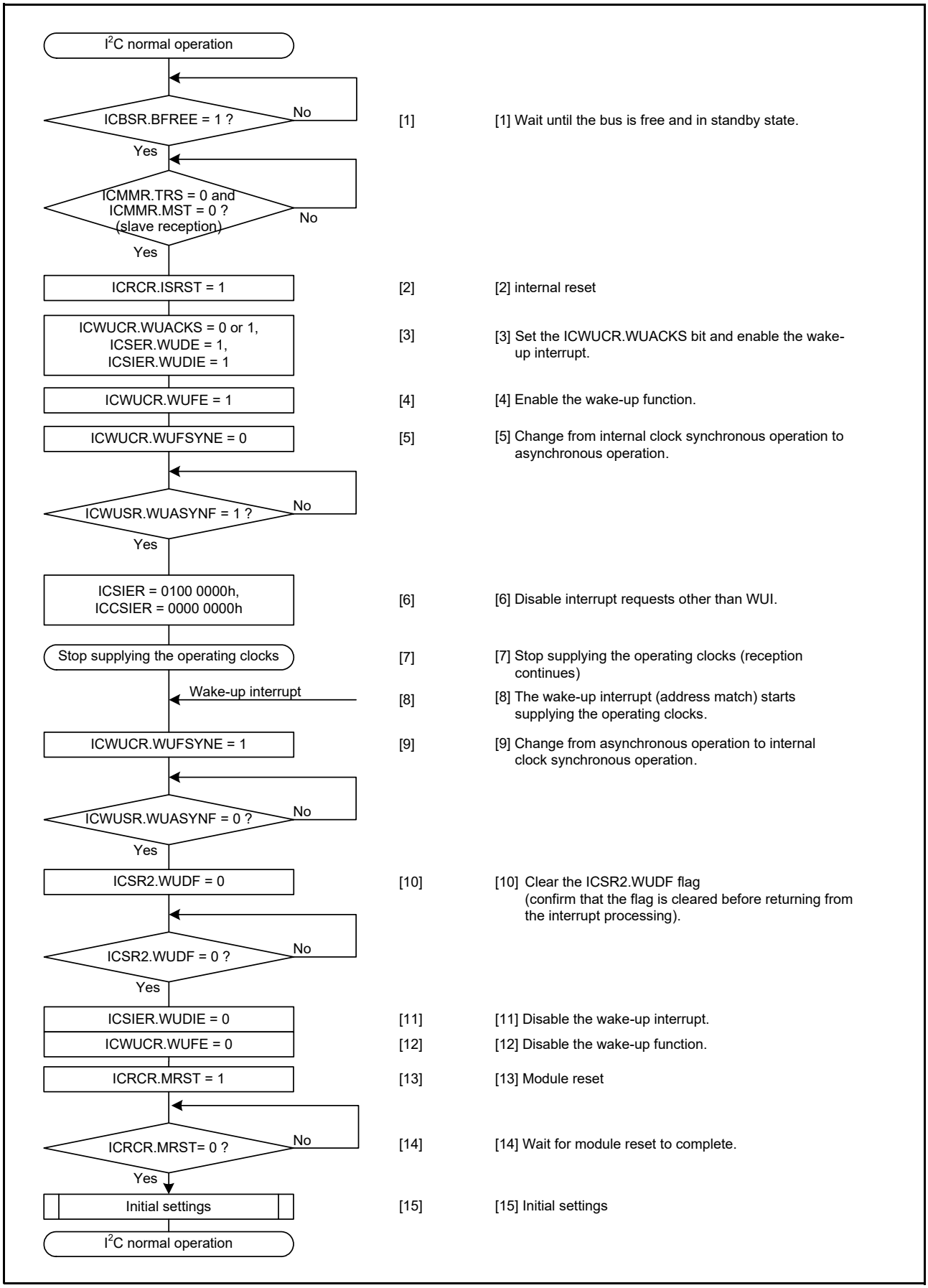


Figure 44.46 Example of Using Command Recovery Mode and EEP Response Mode

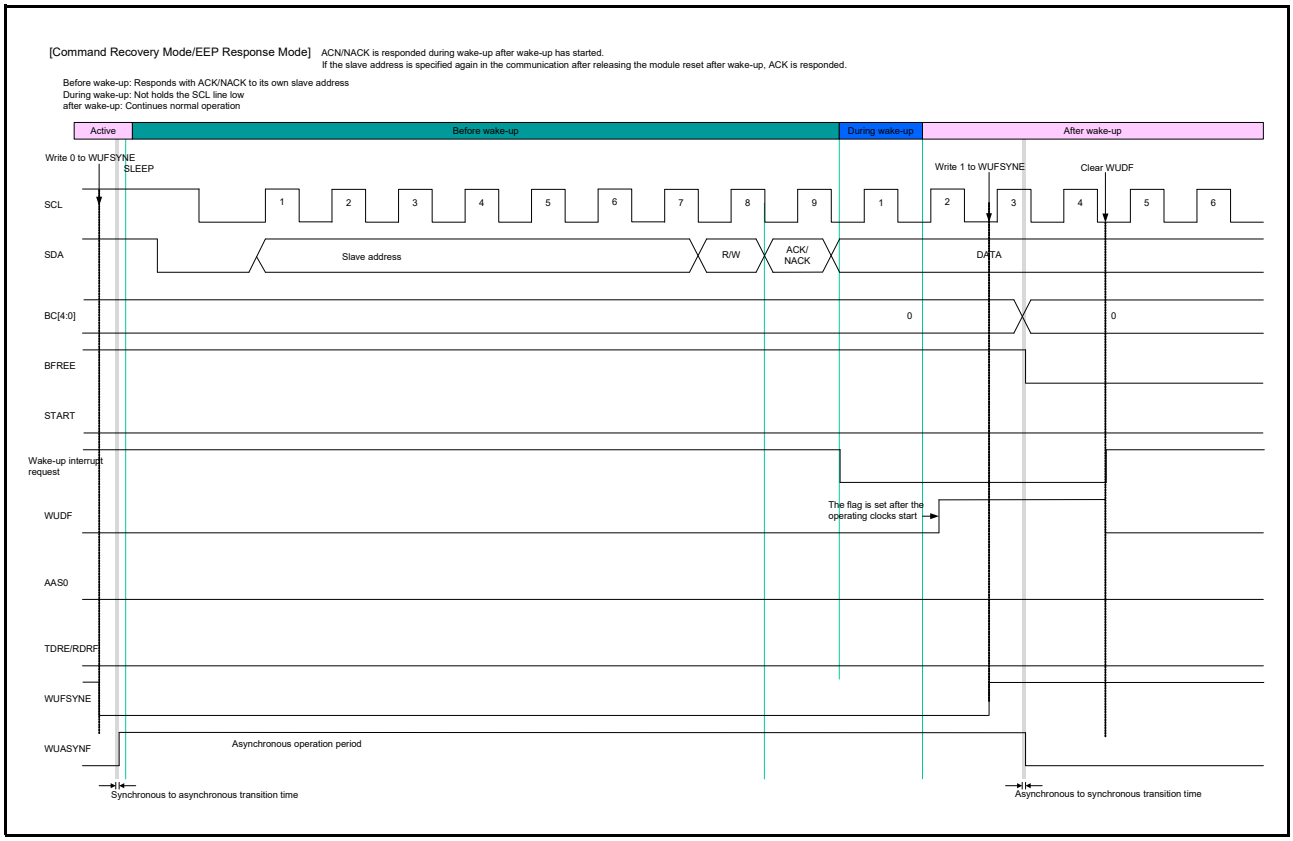


Figure 44.47 Operation Timing of Command Recovery Mode and EEP Response Mode

44.4.9.4 Notes on Using Wake-Up Function

Note the following points when using the wake-up function.

(1) Before setting the ICWUCR.WUFSYNE bit to 0 (asynchronous operation), set the following registers/bits.
   (a) Set the ICWUCR.WUFE, ICSER.WUDE, and ICSIER.WUDIE bits to 1, and the ICMMR.MST and TRS flags to 0.
   (b) The wake-up function does not support device ID addresses and 10-bit address formats. Set the ICSCR.DIDE bit and SARy.FS bit (y = 0 to 2) to 0.
   (c) The timeout function cannot be used. Set the ICSER.TMOE bit to 0.

(2) When the wake-up detection circuit is disabled, the next reception may start if a START condition is detected at the timing when the ICWUCR.WUFSYNE bit is written to 0. In this case, the ICWUSR.WUASYNF flag becomes 1 after the communication is completed and a STOP condition is detected.

(3) After setting the ICWUCR.WUFSYNE bit to 0, do not rewrite the ICFER, ICCSCR, ICACKR, ICICR, ICTCR, and SARy registers until the ICWUSR.WUASYNF flag is set to 1. If rewritten, RIICHS may malfunction.

(4) Do not access the RIICHS registers except for the ICSIER register, ICCSIER register, and ICWUCR.WUFSYNE bit while the ICWUSR.WUASYNF flag is 1 (wake-up detection circuit enabled).

(5) When using normal wake-up mode 2, do not set the ICACKR.ACKBT bit to 1.

(6) If the ICWUCR.WUFSYNE bit is set to 1 (synchronous operation) when no wake-up condition is detected or when the bus is free, the wake-up detection circuit is disabled only after a STOP condition is detected.

44.4.10 SMBus Operation

The RIICHS is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICFER.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the ICRCCR.CKS[2:0] bits, the ICFBR.HIGH[7:0] bits, and the ICFBR.LOW[7:0] bits. In addition, determine the values of the ICOCR.DLCS bit and the ICOCR.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). When the RIICHS is used only as a slave device, the transfer rate setting is not necessary, whereas the ICFBR.LOW[7:0] bits need to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers 0 to 2 (SAR0, SAR1, and SAR2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARy (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration lost detection function.

44.4.10.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with a timer such as TMR using a START condition detection interrupt (STI) and STOP condition detection interrupt (SPI) of the RIICHS. The measured timeout period must be within cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus standard.
If the time measured with the timer exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the slave device must release the bus by writing 1 to the ICRCR.ISRST bit to perform an internal reset of the RIICHS. When an internal reset is performed, the RIICHS stops driving the bus for the SCLHSn pin and SDAHSn pin and make the SCLHSn/SDAHSn pin outputs high-impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment bit to STOP condition

To measure timeout for master devices, measure these periods with a timer such as TMR using a START condition detection interrupt (STI), STOP condition detection interrupt (SPI), and transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RIICHS. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus standard, and the total of all $T_{LOW:MEXT}$ from START condition to STOP condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICCSR.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICCSCR.WAITAE bit 0 until the byte just before reception of the final byte in master receive mode. While the WAITAE bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the timer exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus standard or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the master device must stop the transaction by generating a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDR register).

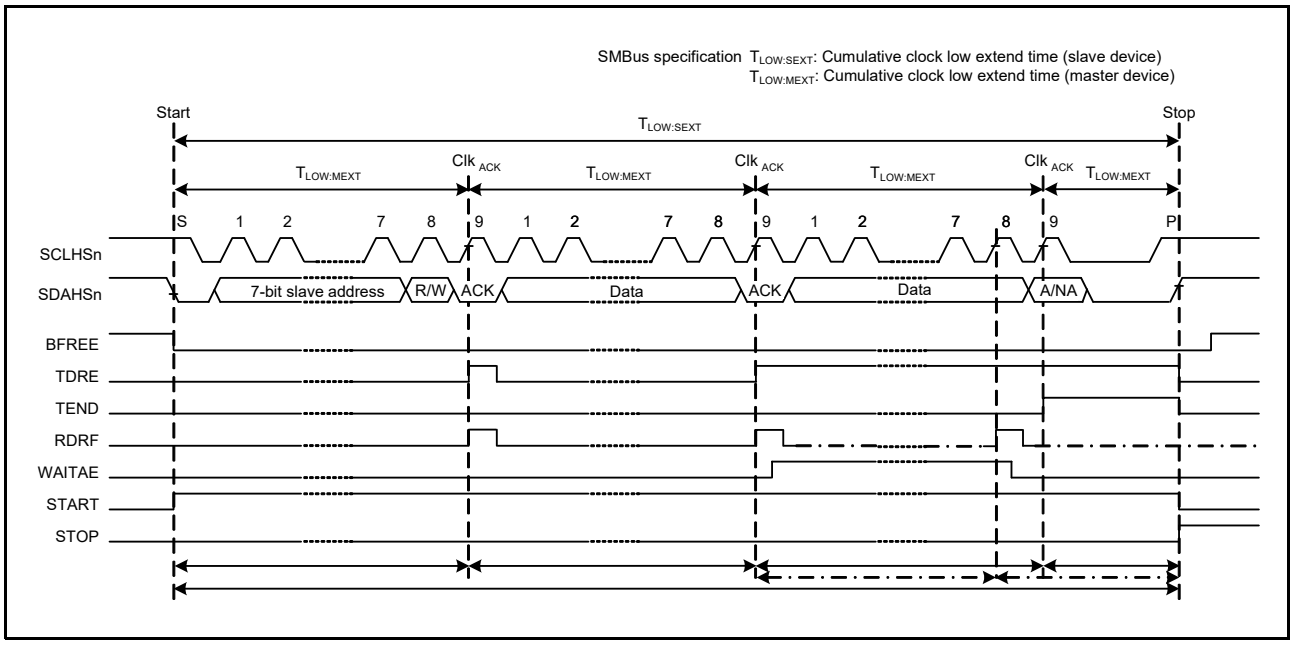


Figure 44.48 SMBus Timeout Measurement

44.4.10.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIICHS. For the CRC generating polynomials of the CRC calculator, refer to section 50, CRC Calculator (CRCA).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICCSCR.WAITAE bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCLHSn line low at the falling edge of the eighth clock pulse.

44.4.10.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIICHS has a function for detecting the host address. To detect the host address as a slave address, set the ICFER.SMBS bit and the ICSCR.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

44.5 Interrupt Sources

The RIICHS generates five types of interrupt requests, as shown in Table 44.7. The transmit data empty and receive data full interrupt requests allow the DTC or DMAC to start data transfer.

Table 44.7 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Start DTC/DMA Transfer	Interrupt Generation Condition
TXI*1	Transmit data empty	ICCSR.TDRE	ICCSIER.TIE	Possible	TDRE = 1 and TIE = 1
RXI*2	Receive data full	ICCSR.RDRF	ICCSIER.RIE	Possible	RDRF = 1 and RIE = 1
TEI*3	Transmission end	ICSR2.TEND	ICSIER.TEIE	Not possible	TEND = 1 and TEIE = 1
EEI	Communication error/ communication event	ICSR2.START	ICSIER.STIE	Not possible	START = 1 and STIE = 1
		ICSR2.STOP	ICSIER.SPIE		STOP = 1 and SPIE = 1
		ICSR2.NACKF	ICSIER.NAKIE		NACKF = 1 and NAKIE = 1
		ICSR2.AL	ICSIER.ALIE		AL = 1 and ALIE = 1
		ICSR2.TMOF	ICSIER.TMOIE		TMOF = 1 and TMOIE = 1
WUI	Wake-up	ICSR2.WUDF	ICSIER.WUDIE	Not possible	WUDF = 1 and WUDIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.TDRE flag (a condition for TXI) is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.RDRF flag (a condition for RXI) is automatically set to 0 when data is read from the ICDR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Clear each flag or mask the interrupt request during interrupt handling.

44.5.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI or RXI interrupt is satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output to the ICU but retained internally (up to one request can be retained internally for each source).

When the IR flag becomes 0, the retained interrupt request is output to the ICU. The retained interrupt request is automatically cleared under normal conditions of usage.

The retained interrupt request can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICCSIER register.

44.6 Event Link Function

The RIICHS outputs the corresponding event signal to the event link controller (ELC) when any of the following events occur.

- Transmit data empty
- Receive data full
- Transmission end
- Communication error/communication event

44.6.1 Interrupt Handling and Event Linking

Each of the RIICHS interrupts has an enable bit that controls whether the interrupt is enabled or disabled. When an interrupt request source occurs, an interrupt request signal is output to the ICU if the corresponding interrupt enable bit is 1.
On the other hand, the event signal is transmitted via the ELC to other modules when the corresponding interrupt request source occurs, regardless of the interrupt enable bit.
For details on interrupt sources, refer to Table 44.7.

44.7 Reset Description

Table 44.8 Reset Control of Registers (1/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICCR	ICE	To be reset	To be reset	Retained
ICRCR	ISRST	To be reset	To be reset	Retained
	MRST	To be reset	Retained	Retained
ICMMR	MSWP	To be reset	To be reset	To be reset
	TRS	To be reset	To be reset	To be reset
	MST	To be reset	To be reset	To be reset
ICFER	HSME	To be reset	To be reset	Retained
	FMPE	To be reset	To be reset	Retained
	SMBS	To be reset	To be reset	Retained
	SCLE	To be reset	To be reset	Retained
	SALE	To be reset	To be reset	Retained
	NALE	To be reset	To be reset	Retained
	MALE	To be reset	To be reset	Retained
ICSCR	SAR2E	To be reset	To be reset	Retained
	SAR1E	To be reset	To be reset	Retained
	SAR0E	To be reset	To be reset	Retained
	HOAE	To be reset	To be reset	Retained
	DIDE	To be reset	To be reset	Retained
	HSMCE	To be reset	To be reset	Retained
	GCAE	To be reset	To be reset	Retained
ICRCCR	CKS[2:0]	To be reset	To be reset	Retained
ICFBR	DBL	To be reset	To be reset	Retained
	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICHBR	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICBFTR		To be reset	To be reset	Retained
ICOCR	DLCS	To be reset	To be reset	Retained
	SDDL[2:0]	To be reset	To be reset	Retained
	CLO	To be reset	To be reset	Retained
	SOWP	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	Retained
	SDAO	To be reset	To be reset	Retained
ICICR	ANFE	To be reset	To be reset	Retained
	NFE	To be reset	To be reset	Retained
	NF[3:0]	To be reset	To be reset	Retained
ICTOR	TMOM[1:0]	To be reset	To be reset	Retained
	TMOH	To be reset	To be reset	Retained
	TMOL	To be reset	To be reset	Retained
	TMOS[1:0]	To be reset	To be reset	Retained

Table 44.8 Reset Control of Registers (2/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICWUCR	WUFE	To be reset	To be reset	Retained
	WUFSYNE	To be reset	To be reset	Retained
	WUANFE	To be reset	To be reset	Retained
	WUACKS	To be reset	To be reset	Retained
ICACKR	ACKWP	To be reset	To be reset	To be reset
	ACKBT	To be reset	To be reset	To be reset
	ACKBR	To be reset	To be reset	To be reset
ICCSCR	WAITRE	To be reset	To be reset	Retained
	WAITAE	To be reset	To be reset	Retained
ICCGR	SP	To be reset	To be reset	To be reset
	RS	To be reset	To be reset	To be reset
	ST	To be reset	To be reset	To be reset
ICDR		To be reset	To be reset	To be reset
ICSR2	WUDF	To be reset	To be reset	Retained
	TMOF	To be reset	To be reset	To be reset
	AL	To be reset	To be reset	To be reset
	TEND	To be reset	To be reset	To be reset
	NACKF	To be reset	To be reset	To be reset
	STOP	To be reset	To be reset	To be reset
	START	To be reset	To be reset	To be reset
ICSER	WUDE	To be reset	To be reset	Retained
	TMOE	To be reset	To be reset	Retained
	ALE	To be reset	To be reset	Retained
	TEDE	To be reset	To be reset	Retained
	NAKDE	To be reset	To be reset	Retained
	SPDE	To be reset	To be reset	Retained
	STDE	To be reset	To be reset	Retained
ICSIER	WUDIE	To be reset	To be reset	Retained
	TMOIE	To be reset	To be reset	Retained
	ALIE	To be reset	To be reset	Retained
	TEIE	To be reset	To be reset	Retained
	NAKIE	To be reset	To be reset	Retained
	SPIE	To be reset	To be reset	Retained
	STIE	To be reset	To be reset	Retained
ICCSR	RDRF	To be reset	To be reset	To be reset
	TDRE	To be reset	To be reset	To be reset
ICCSER	RDE	To be reset	To be reset	Retained
	TDE	To be reset	To be reset	Retained
ICCSIER	RIE	To be reset	To be reset	Retained
	TIE	To be reset	To be reset	Retained
ICBSR	BFREE	To be reset	To be reset	Retained

Table 44.8 Reset Control of Registers (3/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICSSR	AAS2	To be reset	To be reset	To be reset
	AAS1	To be reset	To be reset	To be reset
	AAS0	To be reset	To be reset	To be reset
	HOA	To be reset	To be reset	To be reset
	DID	To be reset	To be reset	To be reset
	HSMC	To be reset	To be reset	To be reset
	GCA	To be reset	To be reset	To be reset
ICWUSR	WUASYNF	To be reset	To be reset	Retained
SARy (y = 0 to 2)	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
SAMRy (y = 0 to 2)	SAV	To be reset	To be reset	Retained
	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
ICBCR	BC[4:0]	To be reset	To be reset	To be reset
ICIMR	SDAO	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	To be reset
	SDAI	To be reset	To be reset	Retained
	SCLI	To be reset	To be reset	Retained

44.8 Usage Notes

44.8.1 Settings of the Module Stop Function

RIICHS operation can be disabled or enabled by module stop control register B (MSTPCRB).
After a reset, the RIICHS is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

45. CAN FD Module with Gateway Functionality (CANFDG)

45.1 Overview

This MCU incorporates a single CAN FD module with gateway functionality (CANFDG0: CANFD0, CANFD1), each featuring two built-in CAN FD (Controller Area Network with Flexible Data Rate) controllers compliant with the ISO 11898-1:2015 standard.

Table 45.1 lists the specifications of the CANFDG module, and Figure 45.1 shows a block diagram of the CANFDG module.

Table 45.1 CANFDG Module Specifications

Item	Description
Protocol	ISO 11898-1:2015 compliant
Number of channels	Two channels
Gateway functionality	CAN 2.0 ↔ CAN 2.0 CAN 2.0 ↔ CAN FD (only when the payload size is 8 bytes)*1 CAN FD ↔ CAN FD*1
Data transfer rate	Arbitration phase: up to 1 Mbps Data phase: up to 8 Mbps individually for each CAN channel*1, *2
Operating frequency*3	Register block: up to 150 MHz (PCLKA) Message buffer RAM: up to 300 MHz (PCLKE)
Operating clock for data link layer (DLL clock)	Up to 80 MHz (either CANFDMCLK or CANFDCLK can be selected)
Frame types	Classic CAN (CAN 2.0) • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID) • Remote frame in base format (11-bit ID) • Remote frame in extended format (29-bit ID) CAN FD*1 • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID)
Data length	Classic CAN: 0 to 8 bytes CAN FD*1: 0 to 8, 12, 16, 20, 24, 32, 48, or 64 bytes
Message buffers	• 32 receive message buffers, shared among all CAN channels • 16 transmit message buffers per channel • Four transmit queue per channel Messages can be automatically transferred to the transmit queue.
FIFOs	Variable size FIFO buffers • Eight receive FIFOs • Six common FIFOs, each of which can be configured as a receive FIFO, transmit FIFO, or gateway FIFO
Automatic transmission interval adjustment	Available when the common FIFO is configured as a transmit FIFO or a gateway FIFO The interval between messages sent from the FIFO can be adjusted.
Acceptance filter	Filterable in the following fields: • IDE bit (base format, extended format, or both) • ID field • RTR bit (data frame or remote frame) (only for Classic CAN) • DLC field (data length) Protection function when the payload size is exceeded Acceptance filter list (AFL) entries can be updated during communication.
Software support	Label information is automatically added to received messages
Timer	Transmission and reception timestamp function
Power down function	Module start/stop function for CAN node (CH_SLEEP and GL_SLEEP mode) Transition to module stop state is possible.
RAM	RAM with ECC protection
Bus traffic measurement	CAN bus traffic measurement of each channel is possible
Protect Zone filter	Security attribute can be set for each module.*4

Note 1. This is only available for products that support the CAN FD protocol.
Note 2. The bit rate for communications depends on the board design and external environment. Determine it following sufficient evaluation.
Note 3. The frequency ratio of PCLKE and PCLKA should be 2 : 1. Also, the frequency of PCLKA should be equal to or higher than that of the DLL clock.
Note 4. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

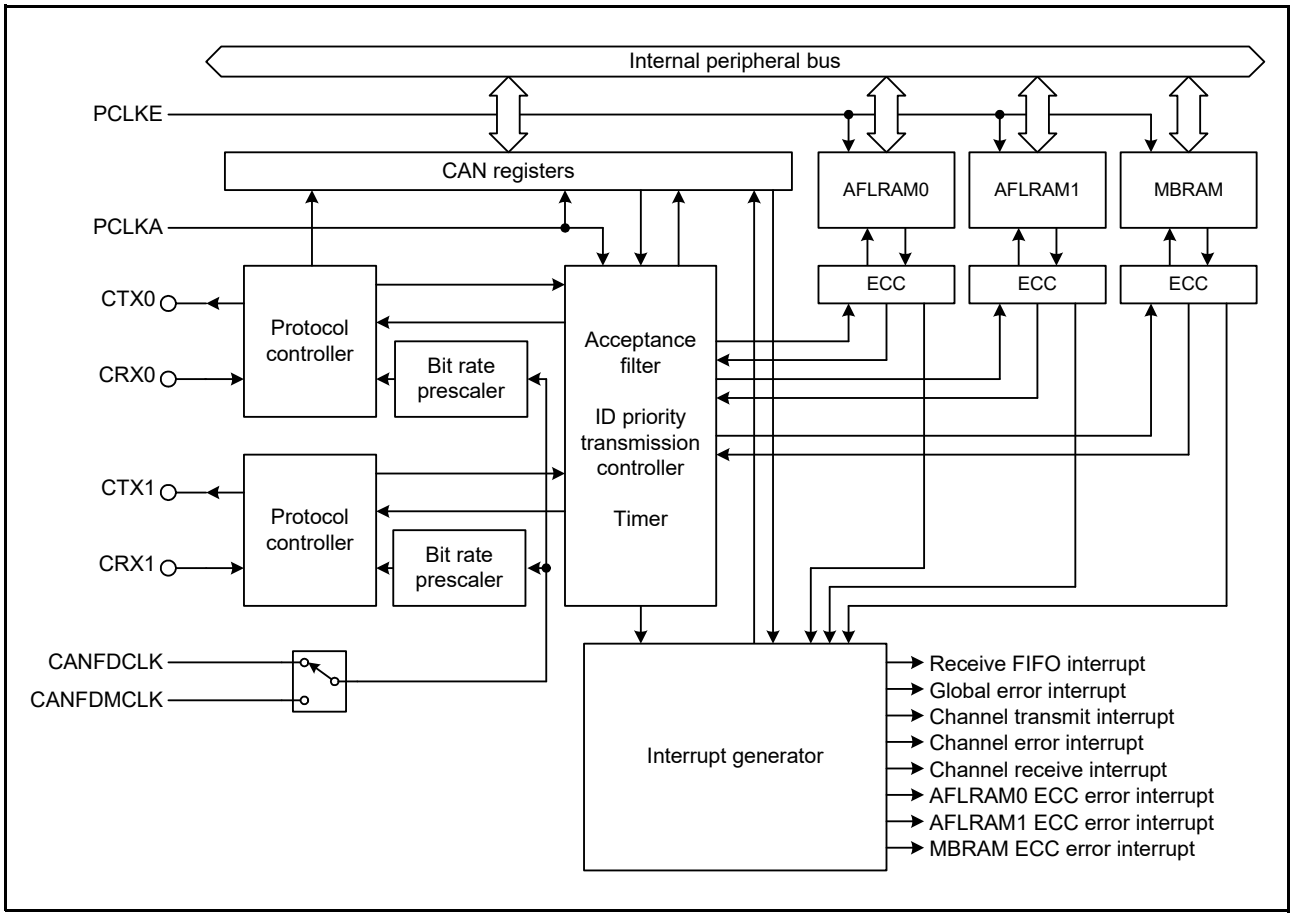


Figure 45.1 CANFDG Module Block Diagram

- CRXi, CTXi (i = 0, 1)
  I/O pins of the CAN FD controller
- Protocol controller
  Handles CAN FD protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.
- Acceptance filter list RAM (AFLRAM0, AFLRAM1)
  Dedicated RAMs to storing the acceptance filter list entries. Each entry contains comparison information for filtering received messages and a destination for messages that match the filter. To accelerate the acceptance filter processing, two acceptance filter list RAMs are provided.
- Message buffer RAM (MBRAM)
  Used for message buffers or FIFOs for transmitting and receiving messages. Each entry has an individual identifier, data length code, a data field, a message pointer for upper layer application, and a timestamp.
- Acceptance filter
  Performs filtering of received messages. The entries configured in the acceptance filter list are used for the filtering process.

- Timers
  Two timers: one used for the receive timestamp function and the other for adjusting the message transmission interval from the transmit FIFO.

Table 45.2 lists the pin configuration of the CANFDG module.

Table 45.2 CANFDG Module Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Receive data input
CTX0	Output	Transmit data output
CRX1	Input	Receive data input
CTX1	Output	Transmit data output

45.2 Register Descriptions

45.2.1 Nominal Bit Rate Configuration Register (NBCR)

Address(es): CANFD0.NBCR 8748 0000h, CANFD1.NBCR 8748 0010h,
CANFD0_NS.NBCR 0748 0000h, CANFD1_NS.NBCR 0748 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG2[6:0]							TSEG1[7:0]								SJW[6]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SJW[5:0]						BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	BRP[9:0]	Bit Rate Prescaler Setting	000h: No division 001h: Divided by 2 : : 3FEh: Divided by 1023 3FFh: Divided by 1024	R/W
b16 to b10	SJW[6:0]	Resynchronization Jump Width Control	00h: 1 Tq 01h: 2 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W
b24 to b17	TSEG1[7:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : FEh: 255 Tq FFh: 256 Tq	R/W
b31 to b25	TSEG2[6:0]	Time Segment 2 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W

This register is used to set the nominal bit rate during transmission and reception of the corresponding channel.
The value cannot be changed in CH_OPERATION or CH_SLEEP mode. Rewrite this register in CH_RESET or CH_HALT mode.
For details on the setting values, refer to section 45.4.1.2, Bit Timing.

BRP[9:0] Bits (Bit Rate Prescaler Setting)

These bits are used to define a period of 1 Tq (Time Quantum) that is the basis for CAN communication. Set the division ratio for the operating clock of the data link layer (DLL clock) selected by the GCFG.DLLCS bit. If the set value is n, the bit rate prescaler divides the DLL clock by n + 1.

SJW[6:0] Bits (Resynchronization Jump Width Control)

These bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 128 Tq can be set. Set a value less than or equal to that of the TSEG2[6:0] bits.

TSEG1[7:0] Bits (Time Segment 1 Control)

These bits are used to specify the total value (TSEG1) of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a Tq value. A value from 2 to 256 Tq can be set.

TSEG2[6:0] Bits (Time Segment 2 Control)

These bits are used to specify the value (TSEG2) of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 128 Tq can be set. Set a value less than that of the TSEG1[7:0] bits.

45.2.2 Channel Control Register (CHCR)

Address(es): CANFD0.CHCR 8748 0004h, CANFD1.CHCR 8748 0014h,
CANFD0_NS.CHCR 0748 0004h, CANFD1_NS.CHCR 0748 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ROME	BFT	—	—	—	CTMS[1:0]		CTME	EDM	BOM[1:0]		—	TDCVIE	SCOVIE	ECOVIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Channel Mode Control	b1 b0 0 0: Requests transition to CH_OPERATION mode 0 1: Requests transition to CH_RESET mode 1 0: Requests transition to CH_HALT mode 1 1: Keep current mode	R/W
b2	SLPRQ	CH_SLEEP Mode Request	0: Requests release from CH_SLEEP mode 1: Requests transition to CH_SLEEP mode	R/W
b3	RTBO	Forced Recovery from Bus-Off*1	0: Forced recovery from bus-off state is disabled. 1: Forced recovery from bus-off state is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BEIE	Bus Error Interrupt Enable*2	0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.	R/W
b9	EWIE	Error Warning Interrupt Enable*2	0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.	R/W
b10	EPIE	Error Passive Interrupt Enable*2	0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.	R/W
b11	BOEIE	Bus-Off Entry Interrupt Enable*2	0: Bus-off entry interrupt is disabled. 1: Bus-off entry interrupt is enabled.	R/W
b12	BORIE	Bus-Off Recovery Interrupt Enable*2	0: Bus-off recovery interrupt is disabled. 1: Bus-off recovery interrupt is enabled.	R/W
b13	OLIE	Overload Interrupt Enable*2	0: Overload interrupt is disabled. 1: Overload interrupt is enabled.	R/W
b14	BLIE	Bus Lock Interrupt Enable*2	0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.	R/W
b15	ALIE	Arbitration Lost Interrupt Enable*2	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b16	TAIE	Transmission Abort Interrupt Enable*2	0: Transmission abort interrupt is disabled. 1: Transmission abort interrupt is enabled.	R/W
b17	ECOVIE	Error Occurrence Counter Overflow Interrupt Enable*2	0: Error occurrence counter overflow interrupt is disabled. 1: Error occurrence counter overflow interrupt is enabled.	R/W
b18	SCOVIE	Success Occurrence Counter Overflow Interrupt Enable*2	0: Success occurrence counter overflow interrupt is disabled. 1: Success occurrence counter overflow interrupt is enabled.	R/W
b19	TDCVIE	Transceiver Delay Compensation Violation Interrupt Enable*2, *3	0: Transceiver delay compensation violation interrupt is disabled. 1: Transceiver delay compensation violation interrupt is enabled.	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22, b21	BOM[1:0]	Bus-Off Recovery Mode Select*2	b22 b21 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Automatically enters CH_HALT mode at bus-off entry 1 0: Automatically enters CH_HALT mode at bus-off end 1 1: Enters CH_HALT mode by software (during bus-off recovery)	R/W
b23	EDM	Error Display Mode Select*4	0: Only the first error detected is indicated. 1: All detected errors are indicated.	R/W
b24	CTME	Channel Test Mode Enable*5	0: Channel test mode is disabled. 1: Channel test mode is enabled.	R/W
b26, b25	CTMS[1:0]	Channel Test Mode Select*5	b26 b25 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loop-back mode) 1 1: Self-test mode 1 (internal loop-back mode)	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	BFT	Bit Flip Test*5	0: First bit of received data stream is not inverted. 1: First bit of received data stream is inverted.	R/W
b31	ROME	Restricted Operation Mode Enable *3, *5	0: Limited operation mode is disabled. 1: Limited operation mode is enabled.	R/W

Note 1. Set this bit in CH_OPERATION mode.
Note 2. Rewrite these bits in CH_RESET mode.
Note 3. Do not set this bit to 1 in Classic Only mode.
Note 4. Set this bit in CH_RESET or CH_HALT mode.
Note 5. Set these bits in CH_HALT mode.

This register controls the modes of the corresponding channel. It is also used to enable interrupt generation when an error is detected on the CAN bus and to set the test mode.
In CH_SLEEP mode, values other than the SLPRQ bit cannot be changed.

MDC[1:0] Bits (Channel Mode Control)

The MDC[1:0] bits are used to specify the mode of the CAN channel. For the CAN mode transition, refer to section 45.3.2, Channel Modes.
When the CANFDG module is in GL_HALT mode, these bits can only be set to 10b (CH_HALT mode) or 01b (CH_RESET mode).
These bits automatically become 10b when the mode is changed to CH_HALT mode by the setting of the CHCR.BOM[1:0] bits.
When writing to these bits and transition to CH_HALT mode (at the entry to bus-off state when the CHCR.BOM[1:0] bits are 01b, and at the end of bus-off state when the CHCR.BOM[1:0] bits are 10b) occur at the same time, priority is given to writing from the CPU. The value of these bits are automatically updated only if the above event occurs when these bits are 00b (CH_OPERATION mode).

SLPRQ Bit (CH_SLEEP Mode Request)

This bit is used to control the transition to CH_SLEEP mode and the return from CH_SLEEP mode.
Setting this bit to 1 in CH_RESET mode requests the channel to transition to CH_SLEEP mode. Setting this bit to 0 in CH_SLEEP mode requests the channel to transition to CH_RESET mode. These bits cannot be changed in other modes.

RTBO Bit (Forced Recovery from Bus-Off)

This bit is used to forcibly recover from the bus-off state. Use this bit only when the CHCR.BOM[1:0] bits are 00b.
This bit is automatically set to 0 after being set to 1. The read value is always 0.
When this bit is set to 1 during bus-off state, the channel transitions from bus-off state to integrating state within 1 bit time. Also the CHSR.REC[7:0] and TEC[7:0] bits are set to 00h and the CHSR.BOST flag is set to 0. Other registers and

bits do not change. In this case, even if the bus-off recovery interrupt is enabled, the bus-off recovery interrupt does not generated.
If this bit is set to 1 except in the bus-off state, nothing occurs.

BEIE Bit (Bus Error Interrupt Enable)

When the CHESR.BEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EWIE Bit (Error Warning Interrupt Enable)

When the CHESR.EWDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EPIE Bit (Error Passive Interrupt Enable)

When the CHESR.EPDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the CHESR.BOEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the CHESR.BORDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

OLIE Bit (Overload Interrupt Enable)

When the CHESR.OLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BLIE Bit (Bus Lock Interrupt Enable)

When the CHESR.BLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

ALIE Bit (Arbitration Lost Interrupt Enable)

When the CHESR.ALDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TAIE Bit (Transmission Abort Interrupt Enable)

When the transmission from the transmit message buffer is successfully aborted while this bit is 1, a channel transmit interrupt request is generated.

ECOVIE Bit (Error Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.ECOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

SCOVIE Bit (Success Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.SCOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TDCVIE Bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the FDSTS.TDCV flag is set to 1 while this bit is 1, a channel error interrupt request is generated. Do not set this bit to 1 in Classic only mode.

BOM[1:0] Bits (Bus-Off Recovery Mode Select)

These bits control the recovery timing from the bus-off state.

EDM Bit (Error Display Mode Select)

This bit controls the indication mode of the error flags (b14 to b8) in the CHESR register.
When this bit is 0, only the flag corresponding to the first error detected becomes 1. If multiple errors are detected at the

same time, all corresponding flags become 1. The other flags do not become 1 until all the flags in b14 to b8 have been cleared.
When this bit is 1, the error flags are updated each time an error is detected.

CTME Bit (Channel Test Mode Enable)

This bit is used to enable the channel test mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

CTMS[1:0] Bits (Channel Test Mode Select)

These bits are used to select the test mode.
When the channel transitions to CH_RESET mode, these bits become 00b.

BFT Bit (Bit Flip Test)

This bit is used to check the CRC generator in the protocol controller.
Setting this bit to 1 inverts the first bit of the message data stream (ID bit) so that the result of the internally generated CRC does not match the received CRC value.
Note that as a result of bit inversion, a stuff error may be detected instead of a CRC error. Refer to the bit stuffing rule when using this function.
The CRC value generated internally can be checked in the following bits.

- The CHESR.CRC15[14:0] bits (for Classical CAN frames)
- The FDCRC.CRC21[20:0] bits (for CAN FD frames)

When using the BFT bit, it is necessary to have another CAN nodes transmit a reference message.

Note: Since the transmit and receive modes share the same CRC generator, there is no need to test for CRC errors in transmit mode separately.

Bit flip test mode is enabled when both the BFT and CTME bits are 1 and the CTMS[1:0] bits are 00b (basic test mode).
Using this function on the transmitter node causes a bit error or arbitration lost.
When the channel transitions to CH_RESET mode, this bit becomes 0.

ROME Bit (Restricted Operation Mode Enable)

When both this bit and the CTME bit are 1, the limited operation mode is enabled. Use this mode only in basic test mode (CTMS[1:0] bits = 00b). Also, do not set this bit to 1 in Classic only mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

45.2.3 Channel Status Register (CHSR)

Address(es): CANFD0.CHSR 8748 0008h, CANFD1.CHSR 8748 0018h,
CANFD0_NS.CHSR 0748 0008h, CANFD1_NS.CHSR 0748 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RESI	CRDY	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	CH_RESET Status Flag	0: Not in CH_RESET mode 1: In CH_RESET mode	R
b1	HLTST	CH_HALT Status Flag	0: Not in CH_HALT mode 1: In CH_HALT mode	R
b2	SLPST	CH_SLEEP Status Flag	0: Not in CH_SLEEP mode 1: In CH_SLEEP mode	R
b3	EPST	Error Passive Status Flag	0: Not in error passive state 1: In error passive state	R
b4	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b5	TRMST	Transmit Status Flag	0: Channel is not transmitting 1: Transmission in progress	R
b6	RECST	Receive Status Flag	0: Channel is not receiving 1: Reception in progress	R
b7	CRDY	Communication Ready Flag	0: Channel is not ready for communication 1: Channel is ready for communication	R
b8	RESI	Receive ESI Flag*1	0: No message with ESI flag set to 1 was received. 1: At least 1 message with ESI flag set to 1 was received.	R/(W) *2
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	REC[7:0]	Reception Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
b31 to b24	TEC[7:0]	Transmission Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the mode, error and transmission or reception status of the related CAN channel together with its reception and transmission error count values.

RSTST Flag (CH_RESET Status Flag)

The RSTST flag indicates whether the related CAN channel is in CH_RESET mode.
This flag is automatically set to 1 when the related CAN channel enters CH_RESET mode, and is automatically set to 0 when the related CAN channel exits CH_RESET mode. When the mode is changed from CH_RESET mode to CH_SLEEP mode, the RSTST flag remains 1.

HLTST Flag (CH_HALT Status Flag)

The HLTST flag indicates whether the related CAN channel is in CH_HALT mode.
This flag is automatically set to 1 when the related CAN channel enters CH_HALT mode, and is automatically set to 0 when the related CAN channel exits CH_HALT mode.

SLPST Flag (CH_SLEEP Status Flag)

The SLPST flag indicates whether the related CAN channel is in CH_SLEEP mode.
This flag is automatically set to 1 when the related CAN channel enters CH_SLEEP mode, and is automatically set to 0 when the related CAN channel exits CH_SLEEP mode.

EPST Flag (Error Passive Status Flag)

The EPST flag indicates whether the related CAN channel has entered the error passive state.
This flag is automatically set to 1 when the value of the REC[7:0] or TEC[7:0] bits exceeds 127.
This flag is automatically set to 0 when the related CAN channel exits the error passive state or enters CH_RESET mode.

BOST Flag (Bus-Off Status Flag)

The BOST flag indicates whether the related CAN channel has entered the bus-off state.
This flag is automatically set to 1 when the value of the TEC[7:0] bits exceeds 255 and the related CAN channel is in the bus-off state.
This flag is automatically set to 0 when the related CAN channel exits bus-off state.

TRMST Flag (Transmit Status Flag)

The TRMST flag indicates whether the related CAN channel is transmitting a message.
This flag is automatically set to 1 when the related CAN channel is operating as a transmitter node or is in the bus-off state, and is automatically set to 0 when the related CAN channel is in the idle state or starts operating as a receiver node.

RECST Flag (Receive Status Flag)

The RECST flag indicates whether the related CAN channel is receiving a message.
This flag is automatically set to 1 when the related CAN channel is operating as a receiver node, and is automatically set to 0 when the related CAN channel is in the idle state or starts operating as a transmitter node.

CRDY Flag (Communication Ready Flag)

The CRDY flag indicates whether the related CAN channel is ready for communication.
This flag is automatically set to 1 when the related CAN channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting CH_RESET or CH_HALT mode.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET or CH_HALT mode.

Note: This flag is 1 in the bus-off state.

RESI Flag (Receive ESI Flag)

The RESI flag is set to 1 when the ESI bit is sampled recessively for a received message without any error. When in loop-back or mirror mode, the self-transmitted messages are considered reception messages.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is cleared by writing 0 to it. This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only this flag to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

REC[7:0] Bits (Reception Error Count)

The REC[7:0] bits indicate the value of the reception error counter.
The value in bus-off state is indeterminate.
These bits are automatically set to 00h when the CANFDG module enters GL_RESET or the CAN channel enters CH_RESET mode.

TEC[7:0] Bits (Transmission Error Count)

The TEC[7:0] bits indicate the value of the transmission error counter.
These bits are automatically set to 00h when CANFDG module enters GL_RESET or CAN channel enters CH_RESET mode.

45.2.4 Channel Error Status Register (CHESR)

Address(es): CANFD0.CHESR 8748 000Ch, CANFD1.CHESR 8748 001Ch,
CANFD0_NS.CHESR 0748 000Ch, CANFD1_NS.CHESR 0748 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CRC15[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADEDF	B0EDF	B1EDF	CEDF	AEDF	FEDF	SEDF	ALDF	BLDF	OLDF	BORDF	BOEDF	EPDF	EWDF	BEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEDF	Bus Error Detect Flag*1	0: Bus error is not detected 1: Bus error is detected	R/(W) *2
b1	EWDF	Error Warning Detect Flag*1	0: Error warning is not detected 1: Error warning is detected	R/(W) *2
b2	EPDF	Error Passive Detect Flag*1	0: Error passive is not detected 1: Error passive is detected	R/(W) *2
b3	BOEDF	Bus-Off Entry Detect Flag*1	0: Bus-off entry is not detected 1: Bus-off entry is detected	R/(W) *2
b4	BORDF	Bus-Off Recovery Detect Flag*1	0: Bus-off recovery is not detected 1: Bus-off recovery is detected	R/(W) *2
b5	OLDF	Overload Detect Flag*1	0: Overload is not detected 1: Overload is detected	R/(W) *2
b6	BLDF	Bus Lock Detect Flag*1	0: Bus lock is not detected 1: Bus lock is detected	R/(W) *2
b7	ALDF	Arbitration Lost Detect Flag*1	0: Arbitration lost is not detected 1: Arbitration lost is detected	R/(W) *2
b8	SEDF	Stuff Error Detect Flag*1	0: Stuff error is not detected 1: Stuff error is detected	R/(W) *2
b9	FEDF	Form Error Detect Flag*1	0: Form error is not detected 1: Form error is detected	R/(W) *2
b10	AEDF	Acknowledge Error Detect Flag*1	0: Acknowledge error is not detected 1: Acknowledge error is detected	R/(W) *2
b11	CEDF	CRC Error Detect Flag*1	0: CRC error is not detected 1: CRC error is detected	R/(W) *2
b12	B1EDF	Bit 1 Error Detect Flag*1	0: Bit 1 error is not detected 1: Bit 1 error is detected	R/(W) *2
b13	B0EDF	Bit 0 Error Detect Flag*1	0: Bit 0 error is not detected 1: Bit 0 error is detected	R/(W) *2
b14	ADEDF	ACK Delimiter Error Detect Flag*1	0: Acknowledge delimiter error is not detected 1: Acknowledge delimiter error is detected	R/(W) *2
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30-b16	CRC15[14:0]	CRC_15 Test	These bits show the CRC_15 value calculated for the CAN2.0 CAN frame.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of various error conditions detectable regardless of the interrupt enable/disable setting of the Channel Control Register (CHCR). It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for the conditions under which each error occurs.

Only a single bit can be cleared at a time. Do not use the bit clear instruction to clear the flag. Set only the flag to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

BEDF Flag (Bus Error Detect Flag)

The BEDF flag indicates a detection of an error state, flagged by b14 to b8 in this register.
This flag is automatically set to 1 when a bus error is detected, and is automatically set to 0 when the related CAN channel enters CH_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

EWDF Flag (Error Warning Detect Flag)

The EWDF flag indicates whether an error warning condition has been detected for the CAN channel.
This flag is automatically set to 1 when the value of either the CHSR.TEC[7:0] or REC[7:0] bits exceeds 95.
The setting of this flag only occurs when the value of the TEC[7:0] or REC[7:0] bits initially exceeds 95. Therefore, if the TEC[7:0] or REC[7:0] bits remains > 95 and the EWDF flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 96 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 95 to a value > 95.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the related CAN channel enters CH_RESET mode.

EPDF Flag (Error Passive Detect Flag)

The EPDF flag indicates a detection of a CAN channel error passive state.
This flag is automatically set to 1 when the CAN error state becomes error passive state.
The setting of this flag only occurs when the value of either the CHSR.TEC[7:0] or REC[7:0] bits initially exceeds 127. Therefore, if the value of either the TEC[7:0] or REC[7:0] bits remains > 127 and the flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 128 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 127 to a value > 127.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the related CAN channel enters CH_RESET mode.

BOEDF Flag (Bus-Off Entry Detect Flag)

The BOEDF flag indicates a detection of a CAN channel bus-off entry state.
This flag is automatically set to 1 when the CAN error state enters the bus-off state.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.

BORDF Flag (Bus-Off Recovery Detect Flag)

The BORDF flag indicates a detection of a CAN channel bus-off recovery state.
This flag is automatically set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CHCR.BOM[1:0] bits are 00b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 10b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 11b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs.

The flag is not set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CH_RESET mode is requested
- When setting the CHCR.RTBO bit to 1 (the CAN channel returns to error active)
- When the CHCR.BOM[1:0] bits are 01b

- When the CHCR.BOM[1:0] bits are 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set to 1.

OLDF Flag (Overload Detect Flag)

The OLDF flag indicates a detection of a CAN channel overload state.
This flag is automatically set to 1 when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode.

BLDF Flag (Bus Lock Detect Flag)

The BLDF flag indicates a detection of a CAN channel bus lock condition.
This flag is automatically set to 1 when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in CH_OPERATION mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode.

ALDF Flag (Arbitration Lost Detect Flag)

The ALDF flag indicates a detection of a CAN channel bus arbitration lost condition.
The flag is automatically set to 1 when an arbitration lost condition is detected on the CAN bus while the CAN channel is in CH_OPERATION mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode.

SEDF Flag (Stuff Error Detect Flag)

The SEDF flag indicates a detection of a CAN stuff error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a stuff error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

FEDF Flag (Form Error Detect Flag)

The FEDF flag indicates a detection of a CAN form error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

AEDF Flag (Acknowledge Error Detect Flag)

The AEDF flag indicates a detection of a CAN acknowledge error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.

This flag is automatically set to 1 when an acknowledge error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CEDF Flag (CRC Error Detect Flag)

The CEDF flag indicates a detection of a CAN CRC error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a CRC error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B1EDF Flag (Bit 1 Error Detect Flag)

The B1EDF flag indicates a detection of a recessive bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B0EDF Flag (Bit 0 Error Detect Flag)

The B0EDF flag indicates a detection of a dominant bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

ADEDF Flag (ACK Delimiter Error Detect Flag)

The ADEDF flag indicates a detection of an acknowledge delimiter bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected during the acknowledge delimiter state of frame transmission. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the related CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CRC15[14:0] Bits (CRC_15 Test)

The CRC15[14:0] bits indicate the calculated CRC_15 value when the CHCR.CTME bit is 1 (channel test mode enabled). If the CHCR.CTME bit is 0, then these bits are always read as 0000h.
The CRC_15 value which is read from these bits show the CAN2.0 CRC value calculated by the CAN channel logic.
The value of the CRC15[14:0] bits is updated in the first bit of the CRC field of the Classical CAN frame.
These bits are automatically set to 0000h when the related CAN channel enters CH_RESET mode.

45.2.5 Data Bit Rate Configuration Register (DBCR)

Address(es): CANFD0.DBCR 8748 1400h, CANFD1.DBCR 8748 1420h,
CANFD0_NS.DBCR 0748 1400h, CANFD1_NS.DBCR 0748 1420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SJW[3:0]				—	—	—	—	TSEG2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSEG1[4:0]					BRP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BRP[7:0]	Bit Rate Prescaler Setting	00h: No division 01h: Divided by 2 : : FEh: Divided by 255 FFh: Divided by 256	R/W
b12 to b8	TSEG1[4:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : 1Eh: 31 Tq 1Fh: 32 Tq	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	TSEG2[3:0]	Time Segment 2 Control	0h: Setting prohibited 1h: 2 Tq : : Eh: 15 Tq Fh: 16 Tq	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	SJW[3:0]	Resynchronization Jump Width Control	0h: 1 Tq 1h: 2 Tq : : Fh: 16 Tq	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register configures the transmission/reception data bit rate parameters of the corresponding channel.
In Classic only mode, there is no need to configure this register.

BRP[7:0] Bits (Bit Rate Prescaler Setting)

The BRP[7:0] bits define the number of the DLL clock contained in 1 Tq (Time Quantum).

TSEG1[4:0] Bits (Time Segment 1 Control)

The TSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 Tq can be set.
The TSEG1[4:0] bits are also used to set the propagation time segment.
Do not write any other value to these bits. Refer to section 45.4.1.2, Bit Timing for more details.

TSEG2[3:0] Bits (Time Segment 2 Control)

The TSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A

value from 2 to 16 Tq can be set.
Do not write any other value to these bits.

SJW[3:0] Bits (Resynchronization Jump Width Control)

The SJW[3:0] bits set the resynchronization jump width. A value from 1 to 16 Tq can be set.

45.2.6 CAN FD Configuration Register (FDCFG)

Address(es): CANFD0.FDCFG 8748 1404h, CANFD1.FDCFG 8748 1424h,
CANFD0_NS.FDCFG 0748 1404h, CANFD1_NS.FDCFG 0748 1424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CLOE	REFE	FDOE	—	GWBRS	GWFDF	GWEN	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TESI	TDCE	SSPC	—	—	—	—	—	ECC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ECC[2:0]	Error Occurrence Counter Configuration*2	b2 b0 0 0 0: All CAN transmitter or receiver frames 0 0 1: All CAN transmitter frames 0 1 0: All CAN receiver frames 0 1 1: Setting prohibited 1 0 0: Only transmitter or receiver CAN FD data-phase (high-speed bits) 1 0 1: Only transmitter CAN FD data-phase (high-speed bits) 1 1 0: Only receiver CAN FD data-phase (high-speed bits) 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SSPC	Secondary Sample Point Configuration*2	0: Measured + offset 1: Offset-only	R/W
b9	TDCE	Transceiver Delay Compensation Enable*2	0: Transceiver delay compensation is disabled 1: Transceiver delay compensation is enabled	R/W
b10	TESI	Transmit ESI Configuration*2	0: The ESI flag in the transmission frame reflects the error status of the node itself. 1: The ESI flag in the transmission frame reflects the ESI bit in the message buffer if the node is not error passive, and the error status of the node itself If the node is error passive.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TDCO[7:0]	Transceiver Delay Compensation Offset*2	Sets the offset value for the transceiver delay compensation	R/W
b24	GWEN	Gateway Function Enable*3	0: Gateway function is disabled 1: Gateway function is enabled	R/W
b25	GWFDF	Gateway FDF Configuration*3	0: Forwarded messages are translated into Classical CAN frames. 1: Forwarded messages are translated into CAN FD frame.	R/W
b26	GWBRS	Gateway BRS Configuration*3	0: Messages are forwarded with the BRS bit set to 0. 1: Messages are forwarded with the BRS bit set to 1.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b28	FDOE	FD Only Mode Enable*3	0: FD only mode is disabled 1: FD only mode is enabled	R/W
b29	REFE	Receive Edge Filter Enable*3	0: Reception edge filter is disabled 1: Reception edge filter is enabled	R/W
b30	CLOE	Classic Only Mode Enable*3, *4	0: Classic only mode is disabled 1: Classic only mode is enabled	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The value after reset is 0 for products that support the CAN FD protocol, and 1 for products that support only CAN 2.0 protocol.

Note 2. Set these bits in CH_RESET or CH_HALT mode.
Note 3. Set these bits in CH_RESET mode.
Note 4. This bit can only be written for products that support the CAN FD protocol. For products that support only CAN 2.0 protocol, this bit is reserved and fixed to 1.

This register configures which communication direction (transmitter/receiver) errors are counted.

ECC[2:0] Bits (Error Occurrence Counter Configuration)

The ECC[2:0] bits select which type of CAN frame the protocol errors should be counted for.

SSPC Bit (Secondary Sample Point Configuration)

The SSPC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN channel. If the bit is 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.
Do not set this bit to 1 in Classic only mode.

TDCE Bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN channel.
Do not set this bit to 1 in Classic only mode.

TESI Bit (Transmit ESI Configuration)

The TESI bit selects whether to reflect the error status of the node itself or the value of the ESI bit in the message buffer (CFBn.HF2.ESI bit or TMBn.HF2.ESI bit) in the ESI flag of the transmit message.
When the gateway function is used, the ESI flag in the CAN FD message to be forwarded must be sent with the same value as the original message, regardless of the status of the gateway, bridge, or router.
Do not set this bit to 1 in Classic only mode.

TDCO[7:0] Bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the offset of the secondary sample point. How this value is used, depends on the SSPC setting. If the SSPC bit is 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in the TDCO[7:0] bits, rounded down to the nearest integer number of Tq. Otherwise, the result is equal to the value in the TDCO[7:0] bits. Refer to section 45.4.1.5, Transceiver Delay Compensation for details.
The actual offset value is interpreted as TDCO[7:0] + 1. For example, if 4 is set in TDCO[7:0], the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.
Do not set to these bits in Classic only mode.

GWEN Bit (Gateway Function Enable)

When the GWEN bit is set to 1, the gateway function is enabled.
Configure the common FIFO as gateway FIFO mode (CFCRn.MODE[1:0] bits = 10b) or configure the transmit queue as gateway mode.
The FDF and BRS bits of the message to be forwarded can be specified by the GWFDF and GWBRS bits.

Example:

When the GWEN bit is 1 and the GWFDF bit is 1 on channel 1:
If a Classical CAN frame is received on channel 0 and forwarded to channel 1, this CAN frame is sent as a CAN FD frame by the GWFDF bit.

Table 45.3 shows the received message and converted message corresponding to each bit setting.

Table 45.3 Bit Settings and Message Conversion

Received Message			Value of the GWFDF bit	Forwarded Message		
Frame type	DLC	BRS		DLC	BRS	Frame type
CAN2.0	0 to 8	N/A	1	No changes	Value of the GWBRS bit	CAN FD
CAN2.0	9 to 15	N/A	1	8	Value of the GWBRS bit	CAN FD
CAN FD	0 to 8	Any value	1	No changes	Value of the GWBRS bit	CAN FD
CAN FD	9 to 15	Any value	1	No changes	Value of the GWBRS bit	CAN FD
CAN2.0	0 to 8	N/A	0	No changes	N/A	CAN2.0
CAN2.0	9 to 15	N/A	0	No changes	N/A	CAN2.0
CAN FD	0 to 8	Any value	0	No changes	N/A	CAN2.0
CAN FD	9 to 15	Any value	0	8	N/A	CAN2.0

Note that for different frame types, the payload size is limited to 8 bytes. If the frame type of the received message and the forwarded message are the same, the DLC value remains unchanged.
If a CAN FD frame with a payload size greater than 8 bytes is routed to a Classical CAN node, only the first 8 bytes of data are forwarded and the remaining data is discarded.
When routing a Classical CAN frame with the DLC value greater than 8, the DLC value is changed to 8 at the destination CAN FD node.
Do not route remote frames when the GWEN bit is set to 1.
When the FDOE bit for the destination node is set to 1 (FD only mode is enabled), set both the GWEN and GWFDF bits to 1.
When the CLOE bit for the destination node is set to 1 (Classic only mode is enabled), set the GWEN bit to 1 and the GWFDF bit to 0.

GWFDF Bit (Gateway FDF Configuration)

When the GWEN bit is set to 1, the FDF bit of the transmitted frame is replaced by the value of the GWFDF bit.
Do not set this bit to 1 in Classic only mode.

GWBRS Bit (Gateway BRS Configuration)

When the GWEN bit is set to 1, the BRS bit of the transmitted frame is replaced by the value of the GWBRS bit.
In Classical CAN frames, the BRS bit is not used.
Do not set this bit to 1 in Classic only mode.

FDOE Bit (FD Only Mode Enable)

The FDOE bit enables the transmission and reception of CAN FD frames only. If enabled, communication in Classical CAN frame format is disabled. The value of the FDF bit in the message buffer (CFBn.HF2.FDF bit or TMBn.HF2.FDF bit) is arbitrary because transmission of Classical CAN frames is not possible.
If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and responds with error frames. If a Classical CAN frame is configured for transmission, the FDF bit is transmitted recessive, therefore a CAN FD frame is transmitted. If the data length code (DLC) is configured to be 9 bytes or more, the remaining data bytes are padded with CCh.
Do not set the FDOE and CLOE bits to 1 simultaneously.

REFE Bit (Receive Edge Filter Enable)

The REFE bit enables the reception edge filter during the integrating state. When this bit is 1, two or more consecutive dominant Tq are required to detect an edge for hard synchronization.
Do not set this bit to 1 in Classic only mode.

CLOE Bit (Classic Only Mode Enable)

The CLOE bit enables the Classic only mode. If this bit is set to 1, the protocol controller can only transmit Classical CAN frames and responds to CAN FD frames with a form error or CRC error.
Do not set the CLOE and FDOE bits to 1 simultaneously.

Table 45.4 Operation Mode Configuration

CLOE bit	FDOE bit	Operation mode
0	0	CAN FD mode
0	1	FD only mode
1	0	Classic only mode
1	1	Setting prohibited

45.2.7 CAN FD Control Register (FDCTR)

Address(es): CANFD0.FDCTR 8748 1408h, CANFD1.FDCTR 8748 1428h,
CANFD0_NS.FDCTR 0748 1408h, CANFD1_NS.FDCTR 0748 1428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCCL	ECCL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECCL	Error Occurrence Counter Clear	When 1 is written to this bit, the error occurrence counter is cleared. This bit is read as 0.	R/W
b1	SCCL	Success Occurrence Counter Clear	When 1 is written to this bit, the success occurrence counter is cleared. This bit is read as 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the error and success occurrence counters.

ECCL Bit (Error Occurrence Counter Clear)

The ECCL bit is used to clear the error occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel enters CH_RESET mode.

SCCL Bit (Success Occurrence Counter Clear)

The SCCL bit is used to clear the success occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel enters CH_RESET mode.

45.2.8 CAN FD Status Register (FDSTS)

Address(es): CANFD0.FDSTS 8748 140Ch, CANFD1.FDSTS 8748 142Ch,
CANFD0_NS.FDSTS 0748 140Ch, CANFD1_NS.FDSTS 0748 142Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SC[7:0]								EC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDCV	—	—	—	—	—	SCOV	ECOV	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TDCR[7:0]	Transceiver Delay Compensation Result	Indicates the transceiver delay compensation result when the transceiver delay has been measured	R
b8	ECOV	Error Occurrence Counter Overflow Flag*1	0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/(W) *2
b9	SCOV	Success Occurrence Counter Overflow Flag*1	0: Success occurrence counter has not overflowed 1: Success occurrence counter has overflowed	R/(W) *2
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TDCV	Transceiver Delay Compensation Violation Flag*1	0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/(W) *2
b23 to b16	EC[7:0]	Error Occurrence Counter	These bits show the error occurrence counter value.	R
b31 to b24	SC[7:0]	Success Occurrence Counter	These bits show the success occurrence counter value.	R

Note 1. Set this flag to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] Bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.
The measured delay is in cycles of the DLL clock. The result depends on the configuration of the FDCFG.SSPC bit and the offset value in the FDCFG.TDCO[7:0] bits. Refer to section 45.4.1.5, Transceiver Delay Compensation for details.
The TDCR[7:0] bits are updated at the falling edge between the FDF bit and res bit when the FDCFG.SSPC bit is 0 and the FDCFG.TDCE bit is 1 (transceiver delay compensation is enabled).
These bits are automatically set to 00h when the CAN channel enters CH_RESET mode.

ECOV Flag (Error Occurrence Counter Overflow Flag)

The ECOV flag indicates whether the CAN channel error occurrence counter has overflowed.
This flag is set to 1 if a CAN bus error specified in the FDCFG.ECC[2:0] bits is detected when the EC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

SCOV Flag (Success Occurrence Counter Overflow Flag)

The SCOV flag indicates whether the CAN channel success occurrence counter has overflowed.
This flag is set to 1 if a successful message reception or successful message transmission occurs when the SC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

TDCV Flag (Transceiver Delay Compensation Violation Flag)

The CANFDG module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.
The transceiver delay has some variations depending on the physical parameters such as temperature. The TDCR[7:0] bits are updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCV flag captures this violation.
This flag is set to 1 when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times – 2 DLL clocks) and the internal bit overruns.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

EC[7:0] Bits (Error Occurrence Counter)

The EC[7:0] bits are used together with the SC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
This higher error rate can be detected depending on the configuration of the FDCFG.ECC[2:0] bits.
The EC[7:0] bits are set only by CANFDG module logic. These bits are cleared by writing 1 to the FDCTR.ECCL bit.
These bits are updated when an error occurs, according to the configuration of the FDCFG.ECC[2:0] bits. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel enters CH_RESET mode.

SC[7:0] Bits (Success Occurrence Counter)

The SC[7:0] bits are used together with the EC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
The SC[7:0] bits are set only by CANFDG module logic. These bits are cleared by writing 1 to the FDCTR.SCCL bit.
These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel enters CH_RESET mode.

Note: In loop-back mode, the counter is incremented twice.

45.2.9 CAN FD CRC Register (FDCRC)

Address(es): CANFD0.FDCRC 8748 1410h, CANFD1.FDCRC 8748 1430h,
CANFD0_NS.FDCRC 0748 1410h, CANFD1_NS.FDCRC 0748 1430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SBC[3:0]				—	—	—	CRC21[20:16]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CRC21[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b20 to b0	CRC21[20:0]	CRC_21 Test	These bits show the CRC_17 value or CRC_21 value calculated for the CAN FD frame.	R
b23 to b21	—	Reserved	These bits are read as 0.	R
b27 to b24	SBC[3:0]	Stuff Bit Counter	These bits shows the stuff bit count (Mod 8) for the CAN FD frame.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

This register holds the CRC value calculated for the CAN FD frame.

CRC21[20:0] Bits (CRC_21 Test)

The calculated CRC_17 value or CRC_21 value can be read from these bits when the CHCR.CTME bit is 1 (channel test mode).
When the CHCR.CTME bit is 0, the CRC21[20:0] bits are read as 000000h.
The CRC21[20:0] bits is updated in the first bit of the CRC field of the CAN FD frame.
When the CRC_17 field is used, the CRC21[20:17] bits are read as 0.
These bits are automatically set to 000000h when the CAN channel enters CH_RESET mode.

SBC[3:0] Bits (Stuff Bit Counter)

The SBC[3:0] bits contain the stuff count value of the CAN FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN FD frame when the CHCR.CTME bit is enabled in SBC[3:1]. SBC[0] is the parity bit.
When the CHCR.CTME bit is 0, the SBC[3:0] bits are always read as 0000b.
The value of the SBC[3:0] bits is updated in the first bit of CRC field of the CAN FD frame. These bits are automatically set to 0000b when the CAN channel enters CH_RESET mode.

45.2.10 Global Configuration Register (GCFG)

Address(es): CANFDG0.GCFG 8748 0084h, CANFDG0_NS.GCFG 0748 0084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ITP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSBTCS[2:0]			TSCS	TSP[3:0]				—	—	OMRC	DLLCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TPRI	Transmission Priority Setting	0: ID priority 1: Message buffer number priority	R/W
b1	DCE	DLC Check Enable	0: DLC check is disabled 1: DLC check is enabled	R/W
b2	DRE	DLC Replacement Enable	0: DLC replacement is disabled 1: DLC replacement is enabled	R/W
b3	MME	Mirror Mode Enable	0: Mirror mode is disabled 1: Mirror mode is enabled	R/W
b4	DLLCS	DLL Clock Select	0: CANFDCLK 1: CANFDMCLK	R/W
b5	OMRC	Payload-Overflowed Message Reception Configuration	0: The message is discarded. 1: The message payload is cut to fit the specified message size	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TSP[3:0]	Timestamp Counter Prescaler Setting	b11 b8 0 0 0 0: No division 0 0 0 1: Divide-by-2 (= 2^1) 0 0 1 0: Divide-by-4 (= 2^2) 0 0 1 1: Divide-by-8 (= 2^3) : : 1 1 0 1: Divide-by-8192 (= 2^{13}) 1 1 1 0: Divide-by-16384 (= 2^{14}) 1 1 1 1: Divide-by-32768 (= 2^{15})	R/W
b12	TSCS	Timestamp Counter Source Select	0: Clock source for timestamp counter is peripheral module clock (PCLKA) 1: Clock source for timestamp counter is bit time clock	R/W
b15 to b13	TSBTCS[2:0]	Bit Time Clock Select	b15 b13 0 0 0: Select the bit time clock of channel 0 0 0 1: Select the bit time clock of channel 1 Settings other than above are prohibited.	R/W
b31 to b16	ITP[15:0]	Interval Timer Prescaler Setting	FIFO interval timer prescaler value. Set the divided value for the peripheral module clock (PCLKA)	R/W

This register is used to select the transmission priority to be used for all the transmit message buffers and the clock source for the CAN protocol engine. The GCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI Bit (Transmission Priority Setting)

The TPRI bit selects the transmission priority for the CAN channel.

This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode. Do not select message buffer number priority when using the transmit queue.

DCE Bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

DRE Bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE bit is 1, the CANFDG module stores the configured value (AFLn.PTR0.DLC[3:0]) of the DLC in the destination receive message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination receive message buffer or FIFO buffer is unchanged.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

MME Bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

DLLCS Bit (DLL Clock Select)

The DLLCS bit selects the clock source for CAN communication.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only in GL_RESET mode.

OMRC Bit (Payload-Overflowed Message Reception Configuration)

The OMRC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size (RMCR.PLS[2:0], RFCRn.PLS[2:0], and CFCRn.PLS[2:0]). The received message payload is always compared with the available message payload size in the message buffer.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only in GL_RESET mode.
When this bit is set to 1 and payload overflow occurs, the DLC value is stored in the receive message buffer or FIFO buffer unchanged.

TSP[3:0] Bits (Timestamp Counter Prescaler Setting)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.
These bits cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

TSCS Bit (Timestamp Counter Source Select)

The TSCS bit allows the selection of the clock source for the timestamp counter.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode. Additionally, do not set this bit to 1 when CAN FD communication is used.

Note: The bit time clock varies depending on the nominal bit rate and data bit rate configuration.

TSBTCS[2:0] Bits (Bit Time Clock Select)

The TSBTCS[2:0] bits select which channel's bit time clock to use when the bit time clock is selected as the clock source for the timestamp counter (TSCS bit = 1).
These bits cannot be written in GL_SLEEP mode. Write to these bits only in GL_RESET mode.

ITP[15:0] Bits (Interval Timer Prescaler Setting)

The ITP[15:0] bits allow the definition of a reference clock for the FIFO interval timer clock source. When these bits are 0000h, the timer is disabled.
These bits cannot be written in GL_SLEEP mode. Write to these bits only in GL_RESET mode.

45.2.11 Global Control Register (GCR)

Address(es): CANFDG0.GCR 8748 0088h, CANFDG0_NS.GCR 0748 0088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFOWIE	TQMLIE	—	—	POIE	THLIE	MLIE	DEIE	—	—	—	—	—	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Global Mode Control	b1 b0 0 0: Request transition to GL_OPERATION mode 0 1: Request transition to GL_RESET mode 1 0: Request transition to GL_HALT mode 1 1: Keep current value	R/W
b2	SLPRQ	GL_SLEEP Mode Request	0: Request to release GL_SLEEP 1: Request transition to GL_SLEEP	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DEIE	DLC Error Interrupt Enable	0: DLC error interrupt is disabled 1: DLC error interrupt is enabled	R/W
b9	MLIE	Message Lost Interrupt Enable	0: Message lost interrupt is disabled 1: Message lost interrupt is enabled	R/W
b10	THLIE	Transmission History Entry Lost Interrupt Enable	0: Transmission history entry lost interrupt is disabled 1: Transmission history entry lost interrupt is enabled	R/W
b11	POIE	Payload Overflow Interrupt Enable	0: Payload overflow interrupt is disabled 1: Payload overflow interrupt is enabled	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	TQMLIE	Transmit Queue Message Lost Interrupt Enable	0: Transmit queue message lost interrupt is disabled 1: Transmit queue message lost interrupt is enabled	R/W
b15	CFOWIE	Common FIFO Message Overwrite Interrupt Enable	0: Common FIFO message overwrite interrupt is disabled 1: Common FIFO message overwrite interrupt is enabled	R/W
b16	TSCR	Timestamp Counter Reset	When 1 is written to this bit, the timestamp counter is reset. This bit is read as 0.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the Global mode of the CANFDG module and the timestamp function. The register also enables and disables the global error interrupts.

MDC[1:0] Bits (Global Mode Control)

The MDC[1:0] bits are used to specify the modes of the CANFDG module. For the mode transition of the CANFDG module, refer to section 45.3.1, Global Modes.

These bits cannot be written in GL_SLEEP mode.

To request the CANFDG module to transition to GL_SLEEP mode, first set these bits to 01b to enter GL_RESET mode, then set the GCR.SLPRQ bit to 1.

SLPRQ Bit (GL_SLEEP Mode Request)

The SLPRQ bit is used to control the transition to GL_SLEEP mode and the return from GL_SLEEP mode.

When this bit is set to 1, the transition to CH_SLEEP mode is also requested for all CAN channels.
This bit can only be written in GL_RESET or GL_SLEEP mode.

DEIE Bit (DLC Error Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.
This bit cannot be written in GL_SLEEP mode.

MLIE Bit (Message Lost Interrupt Enable)

When the MLIE bit is 1, an interrupt is generated if a message lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

THLIE Bit (Transmission History Entry Lost Interrupt Enable)

When the THLIE bit is 1, an interrupt is generated if a transmission history entry lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

POIE Bit (Payload Overflow Interrupt Enable)

When the POIE bit is 1, an interrupt is generated when a message payload overflow condition occurs.
This bit cannot be written in GL_SLEEP mode.

TQMLIE Bit (Transmit Queue Message Lost Interrupt Enable)

When the TQMLIE bit is 1, an interrupt is generated when a transmit queue message lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

CFOWIE Bit (Common FIFO Message Overwrite Interrupt Enable)

When the CFOWIE bit is 1 and the common FIFO is configured for gateway FIFO mode, an interrupt is generated when a message in the common FIFO is overwritten.
This bit cannot be written in GL_SLEEP mode.

TSCR Bit (Timestamp Counter Reset)

When the TSCR bit is 1, the Timestamp Counter Register (TSCR) is reset to 00000000h.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit in GL_RESET mode.
This bit is automatically set to 0.

45.2.12 Global Status Register (GSR)

Address(es): CANFDG0.GSR 8748 008Ch, CANFDG0_NS.GSR 0748 008Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RAMST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	GL_RESET Status Flag	0: Not in GL_RESET mode 1: In GL_RESET mode	R
b1	HLTST	GL_HALT Status Flag	0: Not in GL_HALT mode 1: In GL_HALT mode	R
b2	SLPST	GL_SLEEP Status Flag	0: Not in GL_SLEEP mode 1: In GL_SLEEP mode	R
b3	RAMST	RAM Initialization Status Flag	0: RAM initialization is completed 1: RAM initialization is in progress	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register indicates the global status of the CANFDG module.

RSTST Flag (GL_RESET Status Flag)

The RSTST flag indicates the status of the CANFDG module in GL_RESET mode.
This flag is automatically set to 1 when the CANFDG module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains 1. This flag is automatically set to 0 when the CANFDG module enters GL_HALT or GL_OPERATION mode.

HLTST Flag (GL_HALT Status Flag)

The HLTST flag indicates the status of the CANFDG module in GL_HALT mode.
This flag is automatically set to 1 when the CANFDG module enters GL_HALT mode. This flag is automatically set to 0 when the CANFDG module exits the GL_HALT mode.

SLPST Flag (GL_SLEEP Status Flag)

The SLPST flag indicates the status of the CANFDG module in GL_SLEEP mode.
This flag is automatically set to 1 when the CANFDG module enters GL_SLEEP mode. This flag is automatically set to 0 when the CANFDG module exits the GL_SLEEP mode.

RAMST Flag (RAM Initialization Status Flag)

The RAMST flag indicates the status of the CANFDG module's RAM initialization.
This flag is automatically set to 1 when the CANFDG module enters GL_SLEEP mode by resetting the MCU. This flag is automatically set to 0 when the CANFDG module completed RAM initialization.

45.2.13 Global Error Status Register (GESR)

Address(es): CANFDG0.GESR 8748 0090h, CANFDG0_NS.GESR 0748 0090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEDF1	EEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CFOWDF	TQMLDF	—	—	PODF	THLDF	MLDF	DEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DEDF	DLC Error Detect Flag	0: DLC error is not detected 1: DLC error is detected	R/(W) *1
b1	MLDF	Message Lost Detect Flag	0: Message lost error is not detected 1: Message lost error is detected	R
b2	THLDF	Transmission History Entry Lost Detect Flag	0: Transmission history entry lost is not detected 1: Transmission history entry lost is detected	R
b3	PODF	Payload Overflow Detect Flag	0: Payload overflow is not detected 1: Payload overflow is detected	R/(W) *1
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b6	TQMLDF	Transmit Queue Message Lost Detect Flag	0: Transmit queue message lost is not detected 1: Transmit queue message lost is detected	R
b7	CFOWDF	Common FIFO Message Overwrite Detect Flag	0: Message overwrite is not detected 1: Message overwrite is detected	R
b9, b8	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EEDF0	Channel 0 ECC Error Detect Flag	0: ECC error is not detected during transmission scan for channel 0 1: ECC error is detected during transmission scan for channel 0	R/(W) *1
b17	EEDF1	Channel 1 ECC Error Detect Flag	0: ECC error is not detected during transmission scan for channel 1 1: ECC error is detected during transmission scan for channel 1	R/(W) *1
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the detection of global errors.

DEDF Flag (DLC Error Detect Flag)

The DEDF flag indicates the error status of the DLC.

This flag cannot be written in GL_SLEEP or GL_RESET mode.

Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

This flag is automatically set to 1 when a DLC error is detected in a received frame.

This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

The flag is cleared by writing 0 to it. This flag is automatically set to 0 when the CANFDG module enters GL_RESET mode.

MLDF Flag (Message Lost Detect Flag)

The MLDF flag indicates the status of the message lost error.
This flag is automatically set to 1 when a FIFO message lost error is detected.
This flag is automatically set to 0 when:

- All FIFO message lost flags (RFSRn.LOST, CFSRn.LOST) are cleared
- The CANFDG module enters GL_RESET mode.

THLDF Flag (Transmission History Entry Lost Detect Flag)

The THLDF flag indicates the status of the transmission history entry lost error.
This flag is automatically set to 1 when a transmission history entry lost error is detected.
This flag is automatically set to 0 when:

- The transmission history lost flag (THSR.LOST) is cleared
- The CANFDG module enters GL_RESET mode.

PODF Flag (Payload Overflow Detect Flag)

The PODF flag is automatically set to 1 when a message payload overflow is detected on any channel.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 0 when the CANFDG module enters GL_RESET mode.

TQMLDF Flag (Transmit Queue Message Lost Detect Flag)

The TQMLDF flag is automatically set to 1 when a message lost is detected in any transmit queue.
This flag is automatically set to 0 when:

- All transmit queue message lost flags (TQSR0.LOST, TQSR1.LOST, TQSR2.LOST, TQSR3.LOST) are cleared
- The CANFDG module enters GL_RESET mode.

CFOWDF Flag (Common FIFO Message Overwrite Detect Flag)

The CFOWDF flag is automatically set to 1 when a message overwrite is detected in any of the common FIFOs while the common FIFO is in gateway FIFO mode.
This flag is automatically set to 0 when:

- All the common FIFO message overwrite flags (CFOWSR.CFOWFn) are cleared
- The CANFDG module enters GL_RESET mode.

EEDFi Flag (Channel i ECC Error Detect Flag) (i = 0, 1)

The EEDFi flag indicates whether an ECC error has occurred on channel i.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode.

45.2.14 Transmit Interrupt Status Register (TISR)

Address(es): CANFDG0.TISR 8748 1300h, CANFDG0_NS.TISR 0748 1300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CFOTIF1	TQOTIF1	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	CFOTIF0	TQOTIF0	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSIF0	Channel 0 Transmission Successful Interrupt Monitor Flag	0: Channel 0 transmission successful interrupt is not generated. 1: Channel 0 transmission successful interrupt is generated.	R
b1	TAIF0	Channel 0 Transmission Abort Interrupt Monitor Flag	0: Channel 0 transmission abort interrupt is not generated. 1: Channel 0 transmission abort interrupt is generated.	R
b2	TQIF0	Channel 0 Transmit Queue Interrupt Monitor Flag	0: Channel 0 transmit queue interrupt is not generated. 1: Channel 0 transmit queue interrupt is generated.	R
b3	CFTIF0	Channel 0 Common FIFO Transmission Interrupt Monitor Flag	0: Channel 0 common FIFO transmission interrupt is not generated. 1: Channel 0 common FIFO transmission interrupt is generated.	R
b4	THIF0	Channel 0 Transmission History Interrupt Monitor Flag	0: Channel 0 transmission history interrupt is not generated. 1: Channel 0 transmission history interrupt is generated.	R
b5	TQOTIF0	Channel 0 Transmit Queue One Frame Transmission Interrupt Monitor Flag	0: Channel 0 transmit queue one frame transmission interrupt is not generated. 1: Channel 0 transmit queue one frame transmission interrupt is generated.	R
b6	CFOTIF0	Channel 0 Common FIFO One Frame Transmission Interrupt Monitor Flag	0: Channel 0 common FIFO one frame transmission interrupt is not generated. 1: Channel 0 common FIFO one frame transmission interrupt is generated.	R
b7	—	Reserved	These bits are read as 0.	R
b8	TSIF1	Channel 1 Transmission Successful Interrupt Monitor Flag	0: Channel 1 transmission successful interrupt is not generated. 1: Channel 1 transmission successful interrupt is generated.	R
b9	TAIF1	Channel 1 Transmission Abort Interrupt Monitor Flag	0: Channel 1 transmission abort interrupt is not generated. 1: Channel 1 transmission abort interrupt is generated.	R
b10	TQIF1	Channel 1 Transmit Queue Interrupt Monitor Flag	0: Channel 1 transmit queue interrupt is not generated. 1: Channel 1 transmit queue interrupt is generated.	R
b11	CFTIF1	Channel 1 Common FIFO Transmission Interrupt Monitor Flag	0: Channel 1 common FIFO transmission interrupt is not generated. 1: Channel 1 common FIFO transmission interrupt is generated.	R
b12	THIF1	Channel 1 Transmission History Interrupt Monitor Flag	0: Channel 1 transmission history interrupt is not generated. 1: Channel 1 transmission history interrupt is generated.	R
b13	TQOTIF1	Channel 1 Transmit Queue One Frame Transmission Interrupt Monitor Flag	0: Channel 1 transmit queue one frame transmission interrupt is not generated. 1: Channel 1 transmit queue one frame transmission interrupt is generated.	R
b14	CFOTIF1	Channel 1 Common FIFO One Frame Transmission Interrupt Monitor Flag	0: Channel 1 common FIFO one frame transmission interrupt is not generated. 1: Channel 1 common FIFO one frame transmission interrupt is generated.	R

Bit	Symbol	Bit Name	Description	R/W
b31 to b15	—	Reserved	These bits are read as 0.	R

This register indicates the detection of transmit specific interrupts.

TSIF0 Flag (Channel 0 Transmission Successful Interrupt Monitor Flag)

The TSIF0 flag becomes 1 if transmission from the transmit message buffer n of channel 0 is successful when the transmit message buffer n interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TMIER0.TMIEn bits = 0)
- When the transmission result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TAIF0 Flag (Channel 0 Transmission Abort Interrupt Monitor Flag)

The TAIF0 flag becomes 1 if transmission from the transmit message buffer n of channel 0 is aborted when the transmission abort interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CHCR.TAIE bit = 0)
- When the transmission result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TQIF0 Flag (Channel 0 Transmit Queue Interrupt Monitor Flag)

The TQIF0 flag becomes 1 if the transmit queue interrupt flag of channel 0 is set to 1 when the transmit queue interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQIE, TQCR1.TQIE, TQCR2.TQIE, or TQCR3.TQIE bit = 0)
- When the transmit queue interrupt flag (TQSR0.TQIF, TQSR1.TQIF, TQSR2.TQIF, or TQSR3.TQIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFTIF0 Flag (Channel 0 Common FIFO Transmission Interrupt Monitor Flag)

The CFTIF0 flag becomes 1 if the common FIFO transmit interrupt flag (CFSRn.CFTIF (n = 0 to 5)) of channel 0 is set to 1 when the common FIFO transmit interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CFCRn.CFTIE bit = 0)
- When the common transmit FIFO interrupt flag (CFSRn.CFTIF) is cleared
- When in GL_RESET or CH_RESET mode.

THIF0 Flag (Channel 0 Transmission History Interrupt Monitor Flag)

The THIF0 flag becomes 1 if the transmission history interrupt flag (THSR.THIF) of channel 0 is set to 1 when the transmission history interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (THCR.THIE bit = 0)
- When the transmission history interrupt flag (THSR.THIF) is cleared
- When in GL_RESET or CH_RESET mode.

TQOTIF0 Flag (Channel 0 Transmit Queue One Frame Transmission Interrupt Monitor Flag)

The TQOTIF0 flag becomes 1 if the transmit queue one frame transmission interrupt flag of channel 0 is set to 1 when the transmit queue one frame transmission interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQOTIE, TQCR1.TQOTIE, TQCR2.TQOTIE, or TQCR3.TQOTIE bit = 0)
- When the transmit queue one frame transmission interrupt flag (TQSR0.TQOTIF, TQSR1.TQOTIF, TQSR2.TQOTIF, or TQSR3.TQOTIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFOTIF0 Flag (Channel 0 Common FIFO One Frame Transmission Interrupt Monitor Flag)

The CFOTIF0 flag becomes 1 if the common FIFO one frame transmission interrupt flag of channel 0 is set to 1 when the common FIFO one frame transmission interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CFECRn.CFOTIE bit = 0)
- When the common FIFO one frame transmission interrupt flag (CFSRn.CFOTIF) is cleared
- When in GL_RESET or CH_RESET mode.

TSIF1 Flag (Channel 1 Transmission Successful Interrupt Monitor Flag)

The TSIF1 flag becomes 1 if transmission from the transmit message buffer n of channel 1 is successful when the transmit message buffer n interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TMIER0.TMIEn bits = 0)
- When the transmission result flags (TMSRn.TXRF[1:0]) of transmit message buffer n are cleared
- When in GL_RESET or CH_RESET mode.

TAIF1 Flag (Channel 1 Transmission Abort Interrupt Monitor Flag)

The TAIF1 flag becomes 1 if transmission from the transmit message buffer n of channel 1 is aborted when the transmission abort interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CHCR.TAIE bit = 0)
- When the transmission result flags (TMSRn.TXRF[1:0]) of transmit message buffer n are cleared
- When in GL_RESET or CH_RESET mode.

TQIF1 Flag (Channel 1 Transmit Queue Interrupt Monitor Flag)

The TQIF1 flag becomes 1 if the transmit queue interrupt flag of channel 1 is set to 1 when the transmit queue interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQIE, TQCR1.TQIE, TQCR2.TQIE, or TQCR3.TQIE bit = 0)
- When the transmit queue interrupt flag (TQSR0.TQIF, TQSR1.TQIF, TQSR2.TQIF, or TQSR3.TQIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFTIF1 Flag (Channel 1 Common FIFO Transmission Interrupt Monitor Flag)

The CFTIF1 flag becomes 1 if the common FIFO transmit interrupt flag (CFSRn.CFTIF (n = 0 to 5)) of channel 1 is set to 1 when the common FIFO transmit interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CFCRn.CFTIE bit = 0)

- When the common transmit FIFO interrupt flag (CFSRn.CFTIF) is cleared
- When in GL_RESET or CH_RESET mode.

THIF1 Flag (Channel 1 Transmission History Interrupt Monitor Flag)

The THIF1 flag becomes 1 if the transmission history interrupt flag (THSR.THIF) of channel 1 is set to 1 when the transmission history interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (THCR.THIE bit = 0)
- When the transmission history interrupt flag (THSR.THIF) is cleared
- When in GL_RESET or CH_RESET mode.

TQOTIF1 Flag (Channel 1 Transmit Queue One Frame Transmission Interrupt Monitor Flag)

The TQOTIF1 flag becomes 1 if the transmit queue one frame transmission interrupt flag of channel 1 is set to 1 when the transmit queue one frame transmission interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQOTIE, TQCR1.TQOTIE, TQCR2.TQOTIE, or TQCR3.TQOTIE bit = 0)
- When the transmit queue one frame transmission interrupt flag (TQSR0.TQOTIF, TQSR1.TQOTIF, TQSR2.TQOTIF, or TQSR3.TQOTIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFOTIF1 Flag (Channel 1 Common FIFO One Frame Transmission Interrupt Monitor Flag)

The CFOTIF1 flag becomes 1 if the common FIFO one frame transmission interrupt flag of channel 1 is set to 1 when the common FIFO one frame transmission interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CFECRn.CFOTIE bit = 0)
- When the common FIFO one frame transmission interrupt flag (CFSRn.CFOTIF) is cleared
- When in GL_RESET or CH_RESET mode.

45.2.15 Timestamp Counter Register (TSCR)

Address(es): CANFDG0.TSCR 8748 0094h, CANFDG0_NS.TSCR 0748 0094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is a read-only register that stores a timestamp based on the selected configuration.
The timestamp value is stored in the TSCR register based on the configuration of the GCFG.TSCS bit, TSBTCS[2:0] bits, and TSP[3:0] bits. The accuracy of the timestamp counter cannot be guaranteed when transitioning to GL_HALT state.
This register is automatically set to 00000000h in GL_RESET mode.

45.2.16 Acceptance Filter List Control Register (AFCR)

Address(es): CANFDG0.AFCR 8748 0098h, CANFDG0_NS.AFCR 0748 0098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	AFLWE	—	—	—	—	PAGE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PAGE[3:0]	Access Page Setting	Set the page number of the Acceptance Filter List, a value between 0 and 7.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	AFLWE	AFL Write Enable	0: Writing data to the Acceptance Filter List is disabled 1: Writing data to the Acceptance Filter List is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to select the Acceptance Filter List page for reading or writing entries into the Acceptance Filter List.

PAGE[3:0] Bits (Access Page Setting)

The PAGE[3:0] bits are used to select the page number to access the desired RAM area of the Acceptance Filter List.
One Acceptance Filter List page consists of 16 Acceptance Filter List entries.
Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.
These bits cannot be written in GL_SLEEP mode.

AFLWE Bit (AFL Write Enable)

Setting the AFLWE bit to 0 after configuration of the Acceptance Filter List prevents writing to the Acceptance Filter List.
Data can be read from the Acceptance Filter List independent of the status of this bit.
This bit cannot be written in GL_SLEEP mode.
Setting this bit to 1 enables write access for the Acceptance Filter List.

45.2.17 Acceptance Filter List Configuration Register (AFCFG)

Address(es): CANFDG0.AFCFG 8748 009Ch, CANFDG0_NS.AFCFG 0748 009Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	RN0[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RN1[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	RN1[8:0]	Number of Rules Setting for Channel 1	Number of rules in the Acceptance Filter List for Channel 1	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24 to b16	RN0[8:0]	Number of Rules Setting for Channel 0	Number of rules in the Acceptance Filter List for Channel 0	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to define the number of rules for entry in the Acceptance Filter List.
The maximum number of available entries in the Acceptance Filter List is 64 per channel.

RN1[8:0] Bits (Number of Rules Setting for Channel 1)

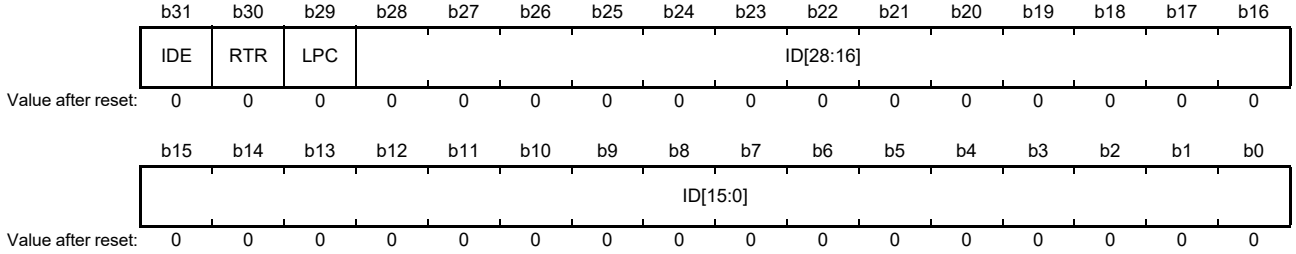
The RN1[8:0] bits define the number of rules in the Acceptance Filter List for channel 1. These bits can be set to a value of 64 rules or less.
These bits can only be written in GL_RESET mode.

RN0[8:0] Bits (Number of Rules Setting for Channel 0)

The RN0[8:0] bits define the number of rules in the Acceptance Filter List for channel 0. These bits can be set to a value of 64 rules or less.
These bits can only be written in GL_RESET mode.

45.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15)

Address(es): CANFDG0.AFL0.IDR 8748 1800h, CANFDG0.AFL1.IDR 8748 1810h, CANFDG0.AFL2.IDR 8748 1820h, CANFDG0.AFL3.IDR 8748 1830h, CANFDG0.AFL4.IDR 8748 1840h, CANFDG0.AFL5.IDR 8748 1850h, CANFDG0.AFL6.IDR 8748 1860h, CANFDG0.AFL7.IDR 8748 1870h, CANFDG0.AFL8.IDR 8748 1880h, CANFDG0.AFL9.IDR 8748 1890h, CANFDG0.AFL10.IDR 8748 18A0h, CANFDG0.AFL11.IDR 8748 18B0h, CANFDG0.AFL12.IDR 8748 18C0h, CANFDG0.AFL13.IDR 8748 18D0h, CANFDG0.AFL14.IDR 8748 18E0h, CANFDG0.AFL15.IDR 8748 18F0h,
CANFDG0_NS.AFL0.IDR 0748 1800h, CANFDG0_NS.AFL1.IDR 0748 1810h, CANFDG0_NS.AFL2.IDR 0748 1820h, CANFDG0_NS.AFL3.IDR 0748 1830h, CANFDG0_NS.AFL4.IDR 0748 1840h, CANFDG0_NS.AFL5.IDR 0748 1850h, CANFDG0_NS.AFL6.IDR 0748 1860h, CANFDG0_NS.AFL7.IDR 0748 1870h, CANFDG0_NS.AFL8.IDR 0748 1880h, CANFDG0_NS.AFL9.IDR 0748 1890h, CANFDG0_NS.AFL10.IDR 0748 18A0h, CANFDG0_NS.AFL11.IDR 0748 18B0h, CANFDG0_NS.AFL12.IDR 0748 18C0h, CANFDG0_NS.AFL13.IDR 0748 18D0h, CANFDG0_NS.AFL14.IDR 0748 18E0h, CANFDG0_NS.AFL15.IDR 0748 18F0h



Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	ID Field	ID part of the Acceptance Filter List entry	R/W
b29	LPC	Loop-Back Configuration	0: Perform normal processing of received messages 1: Only accept its own transmitted messages	R/W
b30	RTR	RTR	0: Data frame 1: Remote frame	R/W
b31	IDE	IDE	0: Standard ID 1: Extended ID	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register is used to configure the ID field in the rule entry of the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

ID[28:0] Bits (ID Field)

The ID[28:0] bits indicate the CAN identifier (ID) field of each entry in the Acceptance Filter List.
The acceptance filter process compares this field against the ID of a received message. For alignment of these bits in base and extended formats, refer to section 45.2.78, Identifier Bits Alignment.

LPC Bit (Loop-Back Configuration)

The LPC bit specifies the processing of received messages in loop-back mode and mirror mode for each entry in the Acceptance Filter List. For normal received messages, set this bit to 0.
Entries with this bit set to 1 only match its own transmitted messages in loop-back mode or mirror mode. Entries with this bit set to 0 match all messages transmitted by other nodes, and its own transmitted messages in loop-back mode.
For details, refer to section 45.5.8, Loop-Back Configuration.

RTR Bit (RTR)

The RTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received message.

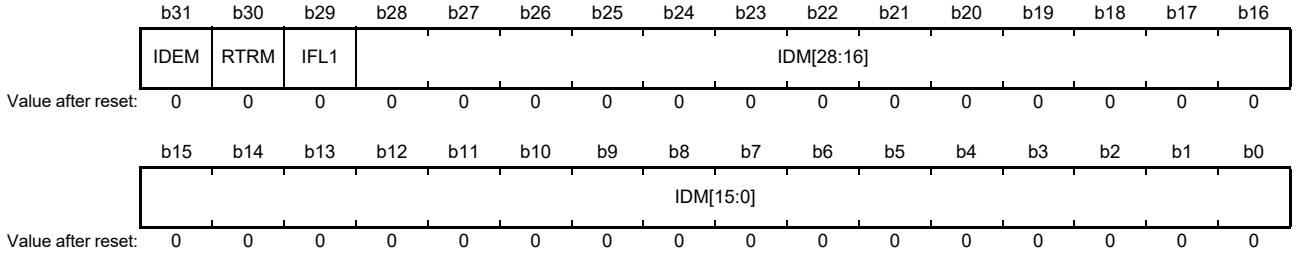
IDE Bit (IDE)

The IDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Acceptance

Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received message.

45.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15)

Address(es): CANFDG0.AFL0.MASK 8748 1804h, CANFDG0.AFL1.MASK 8748 1814h, CANFDG0.AFL2.MASK 8748 1824h, CANFDG0.AFL3.MASK 8748 1834h, CANFDG0.AFL4.MASK 8748 1844h, CANFDG0.AFL5.MASK 8748 1854h, CANFDG0.AFL6.MASK 8748 1864h, CANFDG0.AFL7.MASK 8748 1874h, CANFDG0.AFL8.MASK 8748 1884h, CANFDG0.AFL9.MASK 8748 1894h, CANFDG0.AFL10.MASK 8748 18A4h, CANFDG0.AFL11.MASK 8748 18B4h, CANFDG0.AFL12.MASK 8748 18C4h, CANFDG0.AFL13.MASK 8748 18D4h, CANFDG0.AFL14.MASK 8748 18E4h, CANFDG0.AFL15.MASK 8748 18F4h,
CANFDG0_NS.AFL0.MASK 0748 1804h, CANFDG0_NS.AFL1.MASK 0748 1814h, CANFDG0_NS.AFL2.MASK 0748 1824h, CANFDG0_NS.AFL3.MASK 0748 1834h, CANFDG0_NS.AFL4.MASK 0748 1844h, CANFDG0_NS.AFL5.MASK 0748 1854h, CANFDG0_NS.AFL6.MASK 0748 1864h, CANFDG0_NS.AFL7.MASK 0748 1874h, CANFDG0_NS.AFL8.MASK 0748 1884h, CANFDG0_NS.AFL9.MASK 0748 1894h, CANFDG0_NS.AFL10.MASK 0748 18A4h, CANFDG0_NS.AFL11.MASK 0748 18B4h, CANFDG0_NS.AFL12.MASK 0748 18C4h, CANFDG0_NS.AFL13.MASK 0748 18D4h, CANFDG0_NS.AFL14.MASK 0748 18E4h, CANFDG0_NS.AFL15.MASK 0748 18F4h



Bit	Symbol	Bit Name	Description	R/W
b28 to b0	IDM[28:0]	ID Mask Field	0: Corresponding ID bit is not used for ID matching 1: Corresponding ID bit is used for ID matching	R/W
b29	IFL1	Information Label 1	Set an information label 1 to be attached to a received message	R/W
b30	RTRM	RTR Mask	0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
b31	IDEM	IDE Mask	0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register is used to configure the mask field of each rule for entries in the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

IDM[28:0] Bits (ID Mask Field)

The IDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Acceptance Filter List entry.

IFL1 Bit (Information Label 1)

The IFL1 bit is the upper bit of a 2-bit information label to be attached to a received message accepted by the associated entry in the Acceptance Filter List.
This bit is stored in the upper bit of the Information Label Field (RMBn.HF2.IFL[1], RFBn.HF2.IFL[1], CFBn.HF2.IFL[1]) of the storage location of a received message. This bit is also stored in THACR1n.IFL[1] if the Gateway function is enabled and the THCRn.THRGC bit is set to 1.

RTRM Bit (RTR Mask)

The RTRM bit is the RTR mask bit for each entry in the Acceptance Filter List.

IDEM Bit (IDE Mask)

The IDEM bit is the IDE mask bit for each entry in the Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the IDE bit of the received message.

- If the IDE bit of the received message is 0, the standard ID comparison takes place.
- If the IDE bit of the received message is 1, the extended ID comparison takes place.

45.2.20 Acceptance Filter List n Pointer Register 0 (AFLn.PTR0) (n = 0 to 15)

Address(es): CANFDG0.AFL0.PTR0 8748 1808h, CANFDG0.AFL1.PTR0 8748 1818h, CANFDG0.AFL2.PTR0 8748 1828h, CANFDG0.AFL3.PTR0 8748 1838h, CANFDG0.AFL4.PTR0 8748 1848h, CANFDG0.AFL5.PTR0 8748 1858h, CANFDG0.AFL6.PTR0 8748 1868h, CANFDG0.AFL7.PTR0 8748 1878h, CANFDG0.AFL8.PTR0 8748 1888h, CANFDG0.AFL9.PTR0 8748 1898h, CANFDG0.AFL10.PTR0 8748 18A8h, CANFDG0.AFL11.PTR0 8748 18B8h, CANFDG0.AFL12.PTR0 8748 18C8h, CANFDG0.AFL13.PTR0 8748 18D8h, CANFDG0.AFL14.PTR0 8748 18E8h, CANFDG0.AFL15.PTR0 8748 18F8h,
CANFDG0_NS.AFL0.PTR0 0748 1808h, CANFDG0_NS.AFL1.PTR0 0748 1818h, CANFDG0_NS.AFL2.PTR0 0748 1828h, CANFDG0_NS.AFL3.PTR0 0748 1838h, CANFDG0_NS.AFL4.PTR0 0748 1848h, CANFDG0_NS.AFL5.PTR0 0748 1858h, CANFDG0_NS.AFL6.PTR0 0748 1868h, CANFDG0_NS.AFL7.PTR0 0748 1878h, CANFDG0_NS.AFL8.PTR0 0748 1888h, CANFDG0_NS.AFL9.PTR0 0748 1898h, CANFDG0_NS.AFL10.PTR0 0748 18A8h, CANFDG0_NS.AFL11.PTR0 0748 18B8h, CANFDG0_NS.AFL12.PTR0 0748 18C8h, CANFDG0_NS.AFL13.PTR0 0748 18D8h, CANFDG0_NS.AFL14.PTR0 0748 18E8h, CANFDG0_NS.AFL15.PTR0 0748 18F8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMBE	DMB[6:0]							IFL0	RDS2	RDS1	RDS0	DLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	DLC Field	Set a minimum number of data bytes to be accepted.	R/W
b4	RDS0	Routing Destination Select 0	0: The routing destination is the common FIFO 0. 1: The routing destination is the transmit queue 0.	R/W
b5	RDS1	Routing Destination Select 1	0: The routing destination is the common FIFO 1. 1: The routing destination is the transmit queue 1.	R/W
b6	RDS2	Routing Destination Select 2	0: The routing destination is the common FIFO 2. 1: The routing destination is the transmit queue 2.	R/W
b7	IFL0	Information Label 0	Set an information label 0 to be attached to a received message.	R/W
b14 to b8	DMB[6:0]	Destination Message Buffer Setting	Set the receive message buffer number for storing received messages.	R/W
b15	DMBE	Destination Message Buffer Setting Enable	0: The setting of the DMB[6:0] bits is disabled. 1: The setting of the DMB[6:0] bits is enabled.	R/W
b31 to b16	PTR[15:0]	Pointer	Set a 16-bit pointer to be attached to a received message	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register is used to configure the data length code (DLC), software pointer, and destination message buffer for each rule entry in the Acceptance Filter List.

When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

DLC[3:0] Bits (DLC Field)

The DLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Acceptance Filter List is equal to or higher than the DLC value configured for this associated Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0000b.

RDSj Bit (Routing Destination Select j) (j = 0 to 2)

The RDSj bit is used to select whether the destination for routing is the common FIFO or the transmit queue.
When the RDSj bit is set to 1, setting the corresponding AFLn.PTR1.CFmE bit (m = 0 to 5) to 1 selects the transmit queue. When the RDSj bit is set to 0, setting the corresponding AFLn.PTR1.CFmE bit to 1 selects the common FIFO.

IFL0 Bit (Information Label 0)

The IFL0 bit is the lower bit of a 2-bit information label that can be attached to a received message accepted by the Acceptance Filter List entry.
This bit is stored in the lower bit of the Information Label Field (RMBn.HF2.IFL[0], RFBn.HF2.IFL[0], CFBn.HF2.IFL[0]) of the storage location of a received message. If the gateway function is enabled and the THCRn.THRGC bit is 1, it is also stored in THACR1n.IFL[0].

DMB[6:0] Bits (Destination Message Buffer Setting)

The DMB[6:0] bits allow the configuration of a message buffer as the destination for a received message that passes the acceptance check of the Acceptance Filter List entry. Configure the destination message buffer number.
Write the RMCR.NMB[7:0] bits to configure the number of receive message buffers. The value to be entered in the DMB[6:0] bits should only be between 0000000b and ‘NMB[7:0] – 1’.
If the RMCR.NMB[7:0] bits are 00000000b, set the DMBE bit to 0.

DMBE Bit (Destination Message Buffer Setting Enable)

The DMBE bit allows the enabling or disabling of the receive message buffer as the destination message buffer for a received message that passes the acceptance check of the Acceptance Filter List entry.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the Acceptance Filter List entry. The pointer is added during message storage in the message buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

45.2.21 Acceptance Filter List n Pointer Register 1 (AFLn.PTR1) (n = 0 to 15)

Address(es): CANFDG0.AFL0.PTR1 8748 180Ch, CANFDG0.AFL1.PTR1 8748 181Ch, CANFDG0.AFL2.PTR1 8748 182Ch, CANFDG0.AFL3.PTR1 8748 183Ch, CANFDG0.AFL4.PTR1 8748 184Ch, CANFDG0.AFL5.PTR1 8748 185Ch, CANFDG0.AFL6.PTR1 8748 186Ch, CANFDG0.AFL7.PTR1 8748 187Ch, CANFDG0.AFL8.PTR1 8748 188Ch, CANFDG0.AFL9.PTR1 8748 189Ch, CANFDG0.AFL10.PTR1 8748 18ACh, CANFDG0.AFL11.PTR1 8748 18BCh, CANFDG0.AFL12.PTR1 8748 18CCh, CANFDG0.AFL13.PTR1 8748 18DCh, CANFDG0.AFL14.PTR1 8748 18ECh, CANFDG0.AFL15.PTR1 8748 18FCh,
CANFDG0_NS.AFL0.PTR1 0748 180Ch, CANFDG0_NS.AFL1.PTR1 0748 181Ch, CANFDG0_NS.AFL2.PTR1 0748 182Ch, CANFDG0_NS.AFL3.PTR1 0748 183Ch, CANFDG0_NS.AFL4.PTR1 0748 184Ch, CANFDG0_NS.AFL5.PTR1 0748 185Ch, CANFDG0_NS.AFL6.PTR1 0748 186Ch, CANFDG0_NS.AFL7.PTR1 0748 187Ch, CANFDG0_NS.AFL8.PTR1 0748 188Ch, CANFDG0_NS.AFL9.PTR1 0748 189Ch, CANFDG0_NS.AFL10.PTR1 0748 18ACh, CANFDG0_NS.AFL11.PTR1 0748 18BCh, CANFDG0_NS.AFL12.PTR1 0748 18CCh, CANFDG0_NS.AFL13.PTR1 0748 18DCh, CANFDG0_NS.AFL14.PTR1 0748 18ECh, CANFDG0_NS.AFL15.PTR1 0748 18FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CF5E	CF4E	CF3E	CF2E	CF1E	CF0E	RF7E	RF6E	RF5E	RF4E	RF3E	RF2E	RF1E	RF0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RF0E	Receive FIFO 0 Destination Enable	0: Do not select receive FIFO 0 as the message storage destination 1: Selects receive FIFO 0 as the message storage destination	R/W
b1	RF1E	Receive FIFO 1 Destination Enable	0: Do not select receive FIFO 1 as the message storage destination 1: Selects receive FIFO 1 as the message storage destination	R/W
b2	RF2E	Receive FIFO 2 Destination Enable	0: Do not select receive FIFO 2 as the message storage destination 1: Selects receive FIFO 2 as the message storage destination	R/W
b3	RF3E	Receive FIFO 3 Destination Enable	0: Do not select receive FIFO 3 as the message storage destination 1: Selects receive FIFO 3 as the message storage destination	R/W
b4	RF4E	Receive FIFO 4 Destination Enable	0: Do not select receive FIFO 4 as the message storage destination 1: Selects receive FIFO 4 as the message storage destination	R/W
b5	RF5E	Receive FIFO 5 Destination Enable	0: Do not select receive FIFO 5 as the message storage destination 1: Selects receive FIFO 5 as the message storage destination	R/W
b6	RF6E	Receive FIFO 6 Destination Enable	0: Do not select receive FIFO 6 as the message storage destination 1: Selects receive FIFO 6 as the message storage destination	R/W
b7	RF7E	Receive FIFO 7 Destination Enable	0: Do not select receive FIFO 7 as the message storage destination 1: Selects receive FIFO 7 as the message storage destination	R/W
b8	CF0E	Common FIFO 0/Channel 0 Transmit Queue 0 Destination Enable	0: Do not select common FIFO 0 and channel 0 transmit queue 0 as the message storage destination 1: Selects common FIFO 0 or channel 0 transmit queue 0 as the message storage destination	R/W
b9	CF1E	Common FIFO 1/Channel 0 Transmit Queue 1 Destination Enable	0: Do not select common FIFO 1 and channel 0 transmit queue 1 as the message storage destination 1: Selects common FIFO 1 or channel 0 transmit queue 1 as the message storage destination	R/W
b10	CF2E	Common FIFO 2/Channel 0 Transmit Queue 2 Destination Enable	0: Do not select common FIFO 2 and channel 0 transmit queue 2 as the message storage destination 1: Selects common FIFO 2 or channel 0 transmit queue 2 as the message storage destination	R/W

Bit	Symbol	Bit Name	Description	R/W
b11	CF3E	Common FIFO 3/Channel 1 Transmit Queue 0 Destination Enable	0: Do not select common FIFO 3 and channel 1 transmit queue 0 as the message storage destination 1: Selects common FIFO 3 or channel 1 transmit queue 0 as the message storage destination	R/W
b12	CF4E	Common FIFO 4/Channel 1 Transmit Queue 1 Destination Enable	0: Do not select common FIFO 4 and channel 1 transmit queue 1 as the message storage destination 1: Selects common FIFO 4 or channel 1 transmit queue 1 as the message storage destination	R/W
b13	CF5E	Common FIFO 5/Channel 1 Transmit Queue 2 Destination Enable	0: Do not select common FIFO 5 and channel 1 transmit queue 2 as the message storage destination 1: Selects common FIFO 5 or channel 1 transmit queue 2 as the message storage destination	R/W
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register is used to configure the destination FIFO buffer for each rule entry in the Acceptance Filter List.
Up to eight storage destinations can be specified for received messages using any of the following combinations:

- Eight FIFO buffers
- Seven FIFO buffers and one receive message buffer
- Eight transmit queues
- Seven transmit queues and one receive message buffer
- Total of eight FIFO buffers and transmit queues

When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

RFmE Bit (Receive FIFO m Destination Enable) (m = 0 to 7)

The RFmE bit allows the configuration of the receive FIFO m as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

CFmE Bit (Common FIFO m/Channel i Transmit Queue j Destination Enable) (m = i × 3 + j; i = 0, 1; j = 0 to 2)

The CFmE bit allows the configuration of the common FIFO m or transmit queue j of channel i as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.
When specifying a common FIFO as the storage destination, the specified common FIFO must be configured as a receive FIFO or gateway FIFO.
When specifying a transmit queue as the storage destination, the specified transmission queue buffer must be in gateway mode.

45.2.22 Receive Message Buffer Configuration Register (RMCR)

Address(es): CANFDG0.RMCR 8748 00ACh, CANFDG0_NS.RMCR 0748 00ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PLS[2:0]			NMB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	NMB[7:0]	Number of Message Buffer Setting	Configure the number of receive message buffers	R/W
b10 to b8	PLS[2:0]	Payload Size Setting	b10 b8 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the total number of receive message buffers allocated to a channel.

NMB[7:0] Bits (Number of Message Buffer Setting)

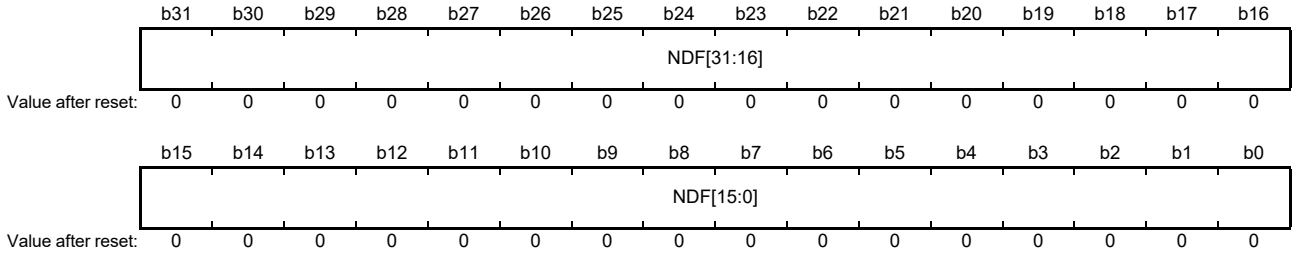
The NMB[7:0] bits are used to configure the number of receive message buffers.
These bits can only be written in GL_RESET mode.
Set a value between 0 and 32. A value of 0 indicates that no receive message buffer is allocated.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits are used to configure the message buffer payload size.
These bits can only be written in GL_RESET mode.

45.2.23 Receive Message Buffer New Data Register 0 (RMNDR0)

Address(es): CANFDG0.RMNDR0 8748 00B0h, CANFDG0_NS.RMNDR0 0748 00B0h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	NDF[31:0]	New Data Flag	0: New data is not stored in corresponding receive message buffer 1: New data is stored in corresponding receive message buffer	R/(W) *1

Note 1. Writing 1 has no effect. These flags are cleared by writing 0 to them.

This register specifies the new data storage status of the receive message buffers.
The bit number of this register corresponds to the buffer number of the receive message buffer.

NDF[31:0] Flags (New Data Flag)

The NDF[31:0] flags indicate that the new data is stored in the corresponding receive message buffer. The NDF[0] flag corresponds to receive message buffer 0.
These flags are automatically set to 1 when storage of new messages are in the corresponding receive message buffer. When RMCR.PLS[2:0] = 000b (payload size is 8 bytes), the time required to store a message is 6 PCLKA cycles. When RMCR.PLS[2:0] > 000b, the time increases by 1 PCLKA cycle for every 4 bytes of payload size (20 PCLKA cycles for 64 bytes).
Do not write to these flags in GL_RESET or GL_SLEEP mode.
These flags are automatically set to 0 when the CANFDG module enters GL_RESET mode.
These flags cannot be cleared when message storage in the corresponding receive message buffer is in progress.
Do not use the bit clear instruction to clear the flags. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

45.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0 to 7)

Address(es): CANFDG0.RFCR0 8748 00C0h, CANFDG0.RFCR1 8748 00C4h, CANFDG0.RFCR2 8748 00C8h, CANFDG0.RFCR3 8748 00CCh, CANFDG0.RFCR4 8748 00D0h, CANFDG0.RFCR5 8748 00D4h, CANFDG0.RFCR6 8748 00D8h, CANFDG0.RFCR7 8748 00DCh, CANFDG0_NS.RFCR0 0748 00C0h, CANFDG0_NS.RFCR1 0748 00C4h, CANFDG0_NS.RFCR2 0748 00C8h, CANFDG0_NS.RFCR3 0748 00CCh, CANFDG0_NS.RFCR4 0748 00D0h, CANFDG0_NS.RFCR5 0748 00D4h, CANFDG0_NS.RFCR6 0748 00D8h, CANFDG0_NS.RFCR7 0748 00DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFITH[2:0]			RFIM	—	FDS[2:0]			—	PLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	RFIE	Receive FIFO Interrupt Enable	0: FIFO interrupt generation is disabled 1: FIFO interrupt generation is enabled	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	FDS[2:0]	FIFO Depth Setting	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	RFIM	Receive FIFO Interrupt Mode Setting	0: Interrupt is generated when the number of messages stored in the receive FIFO reaches the value of the RFITH[2:0] bits from a value smaller than the RFITH[2:0] bits 1: Interrupt is generated at the end of every received message storage	R/W
b15 to b13	RFITH[2:0]	Receive FIFO Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt generated when FIFO is 1/8 full 0 0 1: Interrupt generated when FIFO is 1/4 full 0 1 0: Interrupt generated when FIFO is 3/8 full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8 full 1 0 1: Interrupt generated when FIFO is 3/4 full 1 1 0: Interrupt generated when FIFO is 7/8 full 1 1 1: Interrupt generated when FIFO is full	R/W
b16	RFFIE	Receive FIFO Full Interrupt Enable	0: FIFO full interrupt generation is disabled 1: FIFO full interrupt generation is enabled	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

These registers are used to configure and control the eight receive FIFOs.

RFE Bit (Receive FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the receive FIFO is cleared to empty.
This bit can only be written in GL_HALT or GL_OPERATION mode.
This bit can be set to 1 only when the FIFO depth is 4 or greater (FDS[2:0] ≥ 001b).
Set the RFE bit to 1 with a separate write access to the RFCRn register, after all the other flags in the RFCRn register are set.
This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode.

RFIE Bit (Receive FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.
This bit cannot be written in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM.
This is the maximum number of bytes which can be received by this FIFO.
These bits can only be written in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits select the depth of the FIFO in units of the number of messages. If the FIFO depth is set to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

RFIM Bit (Receive FIFO Interrupt Mode Setting)

The RFIM bit selects the interrupt generation condition for the FIFO.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

RFITH[2:0] Bits (Receive FIFO Interrupt Threshold Setting)

The RFITH[2:0] bits select the counter value of the FIFO for generation of receive FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the RFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 45.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

RFFIE Bit (Receive FIFO Full Interrupt Enable)

When the RFFIE bit is set to 1, an interrupt is generated when the receive FIFO becomes full.
This bit cannot be written in GL_SLEEP mode.
This bit can be used as follows:

- Generate an interrupt at any stage (setting of the RFITH[2:0] bits)
- Generate an interrupt when FIFO is full

These interrupt notifications allow you to manage the received data in the FIFO.

45.2.25 Receive FIFO n Status Register (RFSRn) (n = 0 to 7)

Address(es): CANFDG0.RFSR0 8748 00E0h, CANFDG0.RFSR1 8748 00E4h, CANFDG0.RFSR2 8748 00E8h, CANFDG0.RFSR3 8748 00ECh, CANFDG0.RFSR4 8748 00F0h, CANFDG0.RFSR5 8748 00F4h, CANFDG0.RFSR6 8748 00F8h, CANFDG0.RFSR7 8748 00FCh, CANFDG0_NS.RFSR0 0748 00E0h, CANFDG0_NS.RFSR1 0748 00E4h, CANFDG0_NS.RFSR2 0748 00E8h, CANFDG0_NS.RFSR3 0748 00ECh, CANFDG0_NS.RFSR4 0748 00F0h, CANFDG0_NS.RFSR5 0748 00F4h, CANFDG0_NS.RFSR6 0748 00F8h, CANFDG0_NS.RFSR7 0748 00FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FLVL[7:0]								—	—	—	—	RFIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Receive FIFO Empty Flag	0: Message in receive FIFO 1: No message in receive FIFO (empty)	R
b1	FULL	Receive FIFO Full Flag	0: Receive FIFO not full 1: Receive FIFO full	R
b2	LOST	Message Lost Flag	0: Receive FIFO message lost has not occurred 1: Receive FIFO message lost has occurred	R/(W) *1
b3	RFIF	Receive FIFO Interrupt Flag	0: Receive FIFO interrupt condition is not satisfied 1: Receive FIFO interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FLVL[7:0]	Receive FIFO Fill Level	Indicate the number of messages stored in receive FIFO	R
b16	RFFIF	Receive FIFO Full Interrupt Flag	0: FIFO full interrupt condition is not satisfied 1: FIFO full interrupt condition is satisfied	R/(W) *1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

These registers show the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Receive FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The FLVL[7:0] flags are 00000000b
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFDG module enters GL_RESET mode.

The EMPTY flag is automatically set to 0 when the first message is stored in the receive FIFO.

FULL Flag (Receive FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO buffer matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO buffer is less than the configured FIFO depth
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFDG module enters GL_RESET mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage when the FIFO buffer is already full. This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode.

This flag can only be written in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

RFIF Flag (Receive FIFO Interrupt Flag)

The RFIF flag is automatically set to 1 when the configured interrupt condition is satisfied. This flag is not automatically cleared when the receive FIFO is disabled.
The flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode.

This flag can only be written in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

FLVL[7:0] Bits (Receive FIFO Fill Level)

The FLVL[7:0] bits indicate the number of messages stored in the receive FIFO that can be read by the CPU. These bits are automatically set to 00000000b when the FIFO is disabled or when the CANFDG module enters GL_RESET mode.

RFFIF Flag (Receive FIFO Full Interrupt Flag)

The RFFIF flag is automatically set to 1 when the FIFO full interrupt condition is satisfied. This flag is not automatically cleared when the receive FIFO is disabled.
The flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode.

This flag can only be written in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

45.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0 to 7)

Address(es): CANFDG0.RFPCR0 8748 0100h, CANFDG0.RFPCR1 8748 0104h, CANFDG0.RFPCR2 8748 0108h, CANFDG0.RFPCR3 8748 010Ch, CANFDG0.RFPCR4 8748 0110h, CANFDG0.RFPCR5 8748 0114h, CANFDG0.RFPCR6 8748 0118h, CANFDG0.RFPCR7 8748 011Ch, CANFDG0_NS.RFPCR0 0748 0100h, CANFDG0_NS.RFPCR1 0748 0104h, CANFDG0_NS.RFPCR2 0748 0108h, CANFDG0_NS.RFPCR3 0748 010Ch, CANFDG0_NS.RFPCR4 0748 0110h, CANFDG0_NS.RFPCR5 0748 0114h, CANFDG0_NS.RFPCR6 0748 0118h, CANFDG0_NS.RFPCR7 0748 011Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read pointer of the corresponding receive FIFOs.

When the value 000000FFh is written to this register, the pointer of the corresponding receive FIFO is moved to the next FIFO entry. Write to this register only when the corresponding receive FIFO is enabled and not empty.

The read value from this register is always 00000000h.

This register can only be written in GL_HALT or GL_OPERATION mode.

Do not write to this register when DTC/DMA transfer is enabled (DTCR.RFDTEn bit = 1).

45.2.27 Common FIFO n Configuration Register (CFCRn) (n = 0 to 5)

Address(es): CANFDG0.CFCR0 8748 0120h, CANFDG0.CFCR1 8748 0124h, CANFDG0.CFCR2 8748 0128h, CANFDG0.CFCR3 8748 012Ch, CANFDG0.CFCR4 8748 0130h, CANFDG0.CFCR5 8748 0134h,
CANFDG0_NS.CFCR0 0748 0120h, CANFDG0_NS.CFCR1 0748 0124h, CANFDG0_NS.CFCR2 0748 0128h, CANFDG0_NS.CFCR3 0748 012Ch, CANFDG0_NS.CFCR4 0748 0130h, CANFDG0_NS.CFCR5 0748 0134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TINT[7:0]								FDS[2:0]			LTM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFITH[2:0]			CFIM	ITR	ITCS	MODE[1:0]		—	PLS[2:0]			—	CFTIE	CFRIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFE	Common FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	CFRIE	Common FIFO Receive Interrupt Enable	0: FIFO receive interrupt generation is disabled 1: FIFO receive interrupt generation is enabled	R/W
b2	CFTIE	Common FIFO Transmit Interrupt Enable	0: FIFO transmit interrupt generation is disabled 1: FIFO transmit interrupt generation is enabled	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0: 8 bytes 0 1: 12 bytes 1 0: 16 bytes 1 1: 20 bytes 0 0: 24 bytes 0 1: 32 bytes 1 0: 48 bytes 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9, b8	MODE[1:0]	Operation Mode Setting	b9 b8 0 0: Receive FIFO mode 0 1: Transmit FIFO mode 1 0: Gateway FIFO mode 1 1: Setting prohibited	R/W
b10	ITCS	Interval Timer Clock Source Select	0: Reference clock (× 1 / × 10 period) 1: Bit time clock	R/W
b11	ITR	Interval Timer Resolution Select	0: Reference clock period × 1 1: Reference clock period × 10	R/W
b12	CFIM	Common FIFO Interrupt Mode Setting	In receive FIFO mode, or in gateway FIFO mode with receive interrupt: 0: Reception interrupt generated when the number of messages stored in the common FIFO reaches CFITH[2:0] value from a lower value 1: Reception interrupt generated at the end of every received message storage In transmit FIFO mode, or in gateway FIFO mode with transmit interrupt: 0: Transmission interrupt generated when common FIFO transmits the last message successfully 1: Transmission interrupt generated for every successfully transmitted message	R/W

Bit	Symbol	Bit Name	Description	R/W
b15 to b13	CFITH[2:0]	Common FIFO Receive Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt is generated when FIFO is 1/8 full 0 0 1: Interrupt is generated when FIFO is 1/4 full 0 1 0: Interrupt is generated when FIFO is 3/8 full 0 1 1: Interrupt is generated when FIFO is 1/2 full 1 0 0: Interrupt is generated when FIFO is 5/8 full 1 0 1: Interrupt is generated when FIFO is 3/4 full 1 1 0: Interrupt is generated when FIFO is 7/8 full 1 1 1: Interrupt is generated when FIFO is full	R/W
b20 to b16	LTM[4:0]	Linked Transmit Message Buffer Select	Transmission scan link position of the corresponding channel	R/W
b23 to b21	FDS[2:0]	FIFO Depth Setting	b23 b21 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages	R/W
b31 to b24	TINT[7:0]	Transmission Interval Setting	Delay the start of transmission from the FIFO if configured in transmit mode, delay is a multiple of basic interval timer clock source unit	R/W

CFE Bit (Common FIFO Enable)

Setting the CFE bit to 1 enables the FIFO. The FIFO is disabled when this bit is set to 0.
This bit can also be used, by clearing it, to abort transmission from common FIFO when configured in transmit FIFO mode or gateway FIFO mode, or to stop reception into the common FIFO in receive FIFO mode.
This bit can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the corresponding CAN channel is not in CH_RESET mode.
This bit can only be set to 1 when the FIFO depth is 4 or greater (FDS[2:0] ≥ 001b).
Set the CFE bit to 1 with a separate write access to the CFCRn register, after setting all the other bits in this register.
This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit is automatically set to 0 when the corresponding CAN channel enters CH_RESET mode.

CFRIE Bit (Common FIFO Receive Interrupt Enable)

The CFRIE bit enables or disables generation of common FIFO receive interrupt. If this flag is 1, common FIFO receive interrupt is generated when the common FIFO receive interrupt flag is set to 1 after reception of a frame in the corresponding FIFO buffer.
Do not write to this bit in GL_SLEEP mode.

CFTIE Bit (Common FIFO Transmit Interrupt Enable)

The CFTIE bit enables or disables generation of common FIFO transmit interrupt. If this flag is 1, common FIFO transmit interrupt is generated when the common FIFO transmit interrupt flag is set to 1 after transmission of a frame from the corresponding FIFO buffer.
Do not write to this bit in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, refer to section 45.6, FIFO Buffers and Message Buffer Configuration.
These bits can only be written in GL_RESET mode.

MODE[1:0] Bits (Operation Mode Setting)

The MODE[1:0] bits are used to select the operation mode of the common FIFO. When the MCU reset is applied, all the common FIFO are configured in receive FIFO mode.
These bits cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to these bits only in GL_RESET mode.

ITCS Bit (Interval Timer Clock Source Select)

The ITCS bit is used to select the clock source for the transmission interval timer.
Do not write to this bit in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.
Do not write 1 to this bit when CAN FD communication is used.

Note: The bit time clock can vary depending on the nominal bit rate and data bit rate configuration.

ITR Bit (Interval Timer Resolution Select)

The ITR bit is used to select the resolution of the reference clock for the transmission interval timer.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit when the CFE bit is set to 1.

CFIM Bit (Common FIFO Interrupt Mode Setting)

The CFIM bit is used to select the condition for the generation of common FIFO interrupts.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

CFITH[2:0] Bits (Common FIFO Receive Interrupt Threshold Setting)

The CFITH[2:0] bits are used to select the message counter value for the generation of common FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the CFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 45.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

LTM[4:0] Bits (Linked Transmit Message Buffer Select)

The LTM[4:0] bits are used to select the number of the transmit message buffer that links common FIFO configured in transmit FIFO mode or gateway FIFO mode, for transmission scanning. Set these bits to a value between 0 and 7. Setting these bits to 0 selects transmit message buffer 32, and setting them to 7 selects transmit message buffer 39.
These bits cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to these bits only in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits are used to select the FIFO depth in units of the number of messages. If the FIFO depth is configured to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

TINT[7:0] Bits (Transmission Interval Setting)

The TINT[7:0] bits are used to select the delay in the start of transmission for all messages transmitted from the common FIFO configured in transmit FIFO mode or gateway FIFO mode. The delay is an integral multiple of the clock source cycle of the interval timer (reference clock period × 1, reference clock period × 10, or bit-time clock period).

These bits cannot be written in GL_SLEEP mode.
Do not write to these bits when the CFE bit is set to 1.
When the GCFG.ITP[15:0] bits are set to 0000h, set the TINT[7:0] bits to 00h.

45.2.28 Common FIFO n Extended Configuration Register (CFECRn) (n = 0 to 5)

Address(es): CANFDG0.CFECR0 8748 0180h, CANFDG0.CFECR1 8748 0184h, CANFDG0.CFECR2 8748 0188h, CANFDG0.CFECR3 8748 018Ch, CANFDG0.CFECR4 8748 0190h, CANFDG0.CFECR5 8748 0194h,
CANFDG0_NS.CFECR0 0748 0180h, CANFDG0_NS.CFECR1 0748 0184h, CANFDG0_NS.CFECR2 0748 0188h,
CANFDG0_NS.CFECR3 0748 018Ch, CANFDG0_NS.CFECR4 0748 0190h, CANFDG0_NS.CFECR5 0748 0194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFBME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFOWM	—	—	—	—	—	CFOTIE	CFORIE	CFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFFIE	Common FIFO Full Interrupt Enable	0: FIFO full interrupt generation is disabled 1: FIFO full interrupt generation is enabled	R/W
b1	CFORIE	Common FIFO One Frame Reception Interrupt Enable	0: One frame reception interrupt generation is disabled 1: One frame reception interrupt generation is enabled	R/W
b2	CFOTIE	Common FIFO One Frame Transmit Interrupt Enable	0: One frame transmit interrupt generation is disabled 1: One frame transmit interrupt generation is enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CFOWM	Common FIFO Message Overwrite Mode Select	0: When FIFO is full, the received messages are discarded 1: When FIFO is full, the received messages are overwritten	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	CFBME	Common FIFO Buffering Mode Enable	0: Transmission from common FIFO is disabled 1: Transmission from common FIFO is enabled	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the operation of the common FIFO.

CFFIE Bit (Common FIFO Full Interrupt Enable)

When the CFFIE bit is 1, an interrupt occurs if the common FIFO full interrupt flag becomes 1 after a frame is received.
This bit cannot be written in GL_SLEEP mode.

CFORIE Bit (Common FIFO One Frame Reception Interrupt Enable)

The CFORIE bit enables or disables the common FIFO one frame reception interrupt.
When this bit is set to 1, an interrupt occurs if the common FIFO one frame reception interrupt flag becomes 1 after a frame is received in the corresponding FIFO buffer.
This bit cannot be written in GL_SLEEP mode.

CFOTIE Bit (Common FIFO One Frame Transmit Interrupt Enable)

The CFOTIE bit enables or disables the common FIFO one frame transmit interrupt.

When this bit is set to 1, an interrupt occurs if the common FIFO one frame transmit interrupt flag becomes 1 after a frame in the corresponding FIFO buffer is transmitted.
This bit cannot be written in GL_SLEEP mode.

CFOWM Bit (Common FIFO Message Overwrite Mode Select)

The CFOWM bit is used to select whether to overwrite received messages when there is no space available in the FIFO. When this bit is 0, the received message is discarded when there is no space available in the FIFO. When this bit is 1, the message already stored in the FIFO is overwritten with the new received message.
This bit cannot be written in GL_SLEEP mode.
The CFOWM bit should be set to 1 only when the common FIFO is in gateway mode.
When the CFCRn.CFE bit is 1, do not change the CFOWM bit.

CFBME Bit (Common FIFO Buffering Mode Enable)

When the CFBME bit is 0, messages are sent from the FIFO. When the bit is 1, messages are not sent from the FIFO.
This bit cannot be written in GL_SLEEP mode.
When the CFCRn.CFE bit is 1, do not change the CFBME bit from 0 to 1.

45.2.29 Common FIFO n Status Register (CFSRn) (n = 0 to 5)

Address(es): CANFDG0.CFSR0 8748 01E0h, CANFDG0.CFSR1 8748 01E4h, CANFDG0.CFSR2 8748 01E8h, CANFDG0.CFSR3 8748 01ECh, CANFDG0.CFSR4 8748 01F0h, CANFDG0.CFSR5 8748 01F4h,
CANFDG0_NS.CFSR0 0748 01E0h, CANFDG0_NS.CFSR1 0748 01E4h, CANFDG0_NS.CFSR2 0748 01E8h, CANFDG0_NS.CFSR3 0748 01ECh, CANFDG0_NS.CFSR4 0748 01F0h, CANFDG0_NS.CFSR5 0748 01F4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	CFOWF	—	—	—	—	—	CFOTIF	CFORIF	CFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FLVL[7:0]								—	—	—	CFTIF	CFRIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Common FIFO Empty Flag	0: Message in common FIFO 1: No message in common FIFO (empty)	R
b1	FULL	Common FIFO Full Flag	0: Common FIFO is not full 1: Common FIFO is full	R
b2	LOST	Message Lost Flag	0: No message lost in common FIFO 1: Common FIFO message lost	R/(W) *1
b3	CFRIF	Common FIFO Receive Interrupt Flag	0: Common FIFO receive interrupt condition is not satisfied 1: Common FIFO receive interrupt condition is satisfied	R/(W) *1
b4	CFTIF	Common FIFO Transmit Interrupt Flag	0: Common FIFO transmit interrupt condition is not satisfied 1: Common FIFO transmit interrupt condition is satisfied	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FLVL[7:0]	Common FIFO Fill Level	Indicate the number of messages stored in common FIFO	R
b16	CFFIF	Common FIFO Full Interrupt Flag	0: Common FIFO full interrupt condition is not satisfied 1: Common FIFO full interrupt condition is satisfied	R/(W) *1
b17	CFORIF	Common FIFO One Frame Reception Interrupt Flag	0: Common FIFO one frame reception interrupt condition is not satisfied 1: Common FIFO one frame reception interrupt condition is satisfied	R/(W) *1
b18	CFOTIF	Common FIFO One Frame Transmit Interrupt Flag	0: Common FIFO one frame transmit interrupt condition is not satisfied 1: Common FIFO one frame transmit interrupt condition is satisfied	R/(W) *1
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	CFOWF	Common FIFO Message Overwrite Flag	0: No message overwrite occurred in common FIFO 1: Message overwrite occurred in common FIFO	R/(W) *1
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Common FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The CPU has read all messages from the FIFO buffer in receive FIFO mode

- All messages have been transmitted from the FIFO buffer in transmit FIFO mode or gateway FIFO mode
- The common FIFO is disabled by setting the CFCRn.CFE bit to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

The EMPTY flag is automatically set to 0 when:

- The first reception message is stored in the FIFO buffer in receive FIFO mode
- The first message to be transmitted is stored in the FIFO buffer in transmit FIFO mode or gateway FIFO mode.

FULL Flag (Common FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO matches the configured FIFO depth.
The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO buffer becomes less than the configured FIFO depth
- The common FIFO is disabled by setting the CFCRn.CFE bit to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage of a new message when FIFO buffer is already full in receive FIFO mode or gateway FIFO mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the corresponding CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
The LOST flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

CFRIF Flag (Common FIFO Receive Interrupt Flag)

The CFRIF flag is set to 1 when the configured interrupt condition is satisfied in receive FIFO mode or gateway FIFO mode.
The CFRIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the corresponding CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFRIF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode.
- The CAN channel enters CH_RESET mode in gateway FIFO mode.

CFTIF Flag (Common FIFO Transmit Interrupt Flag)

The CFTIF flag is set to 1 when the configured interrupt condition is satisfied in transmit FIFO mode or gateway FIFO mode.
The CFTIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the corresponding CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFTIF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

FLVL[7:0] Bits (Common FIFO Fill Level)

The FLVL[7:0] bits indicate the following:

- The number of messages stored by the CPU and waiting to be transmitted in transmit FIFO mode
- The number of messages stored by the CANFDG module and available for the CPU to read in receive FIFO mode.
- The number of messages stored by the CANFDG module in gateway FIFO mode

The FLVL[7:0] bits are automatically set to 00000000b when:

- The common FIFO is disabled by setting the CFCRn.CFE bit to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

CFFIF Flag (Common FIFO Full Interrupt Flag)

The CFFIF flag is set to 1 when the configured common FIFO full interrupt condition is satisfied in receive FIFO mode or gateway FIFO mode.
This flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFFIF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

CFORIF Flag (Common FIFO One Frame Reception Interrupt Flag)

The CFORIF flag is set to 1 when the configured common FIFO one frame reception interrupt condition is satisfied in receive FIFO mode or gateway FIFO mode.
This flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET

mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFORIF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in gateway FIFO mode.

CFOTIF Flag (Common FIFO One Frame Transmit Interrupt Flag)

The CFOTIF flag is set to 1 when the configured common FIFO one frame transmit interrupt condition is satisfied in transmit FIFO mode or gateway FIFO mode.
This flag is not cleared automatically if the common FIFO is disabled.
This flag is not affected by the value of the CFCRn.CFIM bit.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFOTIF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

CFOWF Flag (Common FIFO Message Overwrite Flag)

In gateway FIFO mode, when the FIFO is full and the CFECRn.CFOWM bit = 1, the CFOWF flag is set to 1 each time a stored message is overwritten with a new message.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode or gateway FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFOWF flag is set to 0:

- By writing 0 to it
- When the CANFDG module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode or gateway FIFO mode.

45.2.30 Common FIFO n Pointer Control Register (CFPCRn) (n = 0 to 5)

Address(es): CANFDG0.CFPCR0 8748 0240h, CANFDG0.CFPCR1 8748 0244h, CANFDG0.CFPCR2 8748 0248h, CANFDG0.CFPCR3 8748 024Ch, CANFDG0.CFPCR4 8748 0250h, CANFDG0.CFPCR5 8748 0254h, CANFDG0_NS.CFPCR0 0748 0240h, CANFDG0_NS.CFPCR1 0748 0244h, CANFDG0_NS.CFPCR2 0748 0248h, CANFDG0_NS.CFPCR3 0748 024Ch, CANFDG0_NS.CFPCR4 0748 0250h, CANFDG0_NS.CFPCR5 0748 0254h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read or write pointer of the corresponding common FIFO.
When the value 000000FFh is written into this register, the read pointer of the corresponding common FIFO in receive FIFO mode, or the write pointer of the corresponding common FIFO in transmit FIFO mode moves to the next FIFO entry.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
In gateway FIFO mode, writing is prohibited.
Write to this register only when:

- The common FIFO is enabled and is not empty in receive FIFO mode
- The common FIFO is enabled and is not full in transmit FIFO mode.

Do not write to the CFPCRn register when DTC/DMA transfer is enabled (DTCR.CFDTEn bit = 1).

45.2.31 FIFO Empty Status Register (FESR)

Address(es): CANFDG0.FESR 8748 02A0h, CANFDG0_NS.FESR 0748 02A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFEMP5	CFEMP4	CFEMP3	CFEMP2	CFEMP1	CFEMP0	RFEMP7	RFEMP6	RFEMP5	RFEMP4	RFEMP3	RFEMP2	RFEMP1	RFEMP0
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RFEMP0	Receive FIFO 0 Empty Flag	0: Message in corresponding receive FIFO 0 1: No message in corresponding receive FIFO 0 (empty)	R
b1	RFEMP1	Receive FIFO 1 Empty Flag	0: Message in corresponding receive FIFO 1 1: No message in corresponding receive FIFO 1 (empty)	R
b2	RFEMP2	Receive FIFO 2 Empty Flag	0: Message in corresponding receive FIFO 2 1: No message in corresponding receive FIFO 2 (empty)	R
b3	RFEMP3	Receive FIFO 3 Empty Flag	0: Message in corresponding receive FIFO 3 1: No message in corresponding receive FIFO 3 (empty)	R
b4	RFEMP4	Receive FIFO 4 Empty Flag	0: Message in corresponding receive FIFO 4 1: No message in corresponding receive FIFO 4 (empty)	R
b5	RFEMP5	Receive FIFO 5 Empty Flag	0: Message in corresponding receive FIFO 5 1: No message in corresponding receive FIFO 5 (empty)	R
b6	RFEMP6	Receive FIFO 6 Empty Flag	0: Message in corresponding receive FIFO 6 1: No message in corresponding receive FIFO 6 (empty)	R
b7	RFEMP7	Receive FIFO 7 Empty Flag	0: Message in corresponding receive FIFO 7 1: No message in corresponding receive FIFO 7 (empty)	R
b8	CFEMP0	Common FIFO 0 Empty Flag	0: Message in corresponding common FIFO 0 1: No message in corresponding common FIFO 0 (empty)	R
b9	CFEMP1	Common FIFO 1 Empty Flag	0: Message in corresponding common FIFO 1 1: No message in corresponding common FIFO 1 (empty)	R
b10	CFEMP2	Common FIFO 2 Empty Flag	0: Message in corresponding common FIFO 2 1: No message in corresponding common FIFO 2 (empty)	R
b11	CFEMP3	Common FIFO 3 Empty Flag	0: Message in corresponding common FIFO 3 1: No message in corresponding common FIFO 3 (empty)	R
b12	CFEMP4	Common FIFO 4 Empty Flag	0: Message in corresponding common FIFO 4 1: No message in corresponding common FIFO 4 (empty)	R
b13	CFEMP5	Common FIFO 5 Empty Flag	0: Message in corresponding common FIFO 5 1: No message in corresponding common FIFO 5 (empty)	R
b31 to b14	—	Reserved	These bits are read as 0.	R

This register shows the status of the empty flags of the FIFO buffers.

RFEMPn Flag (Receive FIFO n Empty Flag) (n = 0 to 7)

The RFEMPn flag is set to 1 when the RFSRn.EMPTY flag is set to 1, and is set to 0 when the RFSRn.EMPTY flag is set to 0.

The RFEMPn flag is set to 1 when the CANFDG module enters GL_RESET mode.

CFEMPn Flag (Common FIFO n Empty Flag) (n = 0 to 5)

The CFEMPn flag is set to 1 when the CFSRn.EMPTY flag is set to 1, and is set to 0 when the CFSRn.EMPTY flag is set to 0.

The CFEMPn flag is set to 1 when the CANFDG module enters GL_RESET mode.

45.2.32 FIFO Full Status Register (FFSR)

Address(es): CANFDG0.FFSR 8748 02A4h, CANFDG0_NS.FFSR 0748 02A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFFUL5	CFFUL4	CFFUL3	CFFUL2	CFFUL1	CFFUL0	RFFUL7	RFFUL6	RFFUL5	RFFUL4	RFFUL3	RFFUL2	RFFUL1	RFFUL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFFUL0	Receive FIFO 0 Full Flag	0: Corresponding receive FIFO 0 is not full 1: Corresponding receive FIFO 0 is full	R
b1	RFFUL1	Receive FIFO 1 Full Flag	0: Corresponding receive FIFO 1 is not full 1: Corresponding receive FIFO 1 is full	R
b2	RFFUL2	Receive FIFO 2 Full Flag	0: Corresponding receive FIFO 2 is not full 1: Corresponding receive FIFO 2 is full	R
b3	RFFUL3	Receive FIFO 3 Full Flag	0: Corresponding receive FIFO 3 is not full 1: Corresponding receive FIFO 3 is full	R
b4	RFFUL4	Receive FIFO 4 Full Flag	0: Corresponding receive FIFO 4 is not full 1: Corresponding receive FIFO 4 is full	R
b5	RFFUL5	Receive FIFO 5 Full Flag	0: Corresponding receive FIFO 5 is not full 1: Corresponding receive FIFO 5 is full	R
b6	RFFUL6	Receive FIFO 6 Full Flag	0: Corresponding receive FIFO 6 is not full 1: Corresponding receive FIFO 6 is full	R
b7	RFFUL7	Receive FIFO 7 Full Flag	0: Corresponding receive FIFO 7 is not full 1: Corresponding receive FIFO 7 is full	R
b8	CFFUL0	Common FIFO 0 Full Flag	0: Common FIFO 0 is not full 1: Common FIFO 0 is full	R
b9	CFFUL1	Common FIFO 1 Full Flag	0: Common FIFO 1 is not full 1: Common FIFO 1 is full	R
b10	CFFUL2	Common FIFO 2 Full Flag	0: Common FIFO 2 is not full 1: Common FIFO 2 is full	R
b11	CFFUL3	Common FIFO 3 Full Flag	0: Common FIFO 3 is not full 1: Common FIFO 3 is full	R
b12	CFFUL4	Common FIFO 4 Full Flag	0: Common FIFO 4 is not full 1: Common FIFO 4 is full	R
b13	CFFUL5	Common FIFO 5 Full Flag	0: Common FIFO 5 is not full 1: Common FIFO 5 is full	R
b31 to b14	—	Reserved	These bits are read as 0.	R

This register shows the status of the full flags of the FIFO buffers.

RFFULn Flag (Receive FIFO n Full Flag) (n = 0 to 7)

The RFFULn flag is set to 1 when the RFSRn.FULL flag is set to 1, and is set to 0 when the RFSRn.FULL flag is set to 0.

The RFFULn flag is set to 0 when the CANFDG module enters GL_RESET mode.

CFFULn Flag (Common FIFO n Full Flag) (n = 0 to 5)

The CFFULn flag is set to 1 when the CFSRn.FULL flag is set to 1 is set to 0 when the CFSRn.FULL flag is set to 0.
The CFFULn flag is set to 0 when the CANFDG module enters GL_RESET mode.

45.2.33 FIFO Message Lost Status Register (FMLSR)

Address(es): CANFDG0.FMLSR 8748 02A8h, CANFDG0_NS.FMLSR 0748 02A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFML5	CFML4	CFML3	CFML2	CFML1	CFML0	RFML7	RFML6	RFML5	RFML4	RFML3	RFML2	RFML1	RFML0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFML0	Receive FIFO 0 Message Lost Flag	0: No message lost has occurred in receive FIFO 0 1: Message lost has occurred in receive FIFO 0	R
b1	RFML1	Receive FIFO 1 Message Lost Flag	0: No message lost has occurred in receive FIFO 1 1: Message lost has occurred in receive FIFO 1	R
b2	RFML2	Receive FIFO 2 Message Lost Flag	0: No message lost has occurred in receive FIFO 2 1: Message lost has occurred in receive FIFO 2	R
b3	RFML3	Receive FIFO 3 Message Lost Flag	0: No message lost has occurred in receive FIFO 3 1: Message lost has occurred in receive FIFO 3	R
b4	RFML4	Receive FIFO 4 Message Lost Flag	0: No message lost has occurred in receive FIFO 4 1: Message lost has occurred in receive FIFO 4	R
b5	RFML5	Receive FIFO 5 Message Lost Flag	0: No message lost has occurred in receive FIFO 5 1: Message lost has occurred in receive FIFO 5	R
b6	RFML6	Receive FIFO 6 Message Lost Flag	0: No message lost has occurred in receive FIFO 6 1: Message lost has occurred in receive FIFO 6	R
b7	RFML7	Receive FIFO 7 Message Lost Flag	0: No message lost has occurred in receive FIFO 7 1: Message lost has occurred in receive FIFO 7	R
b8	CFML0	Common FIFO 0 Message Lost Flag	0: No message lost has occurred in common FIFO 0 1: Message lost has occurred in common FIFO 0	R
b9	CFML1	Common FIFO 1 Message Lost Flag	0: No message lost has occurred in common FIFO 1 1: Message lost has occurred in common FIFO 1	R
b10	CFML2	Common FIFO 2 Message Lost Flag	0: No message lost has occurred in common FIFO 2 1: Message lost has occurred in common FIFO 2	R
b11	CFML3	Common FIFO 3 Message Lost Flag	0: No message lost has occurred in common FIFO 3 1: Message lost has occurred in common FIFO 3	R
b12	CFML4	Common FIFO 4 Message Lost Flag	0: No message lost has occurred in common FIFO 4 1: Message lost has occurred in common FIFO 4	R
b13	CFML5	Common FIFO 5 Message Lost Flag	0: No message lost has occurred in common FIFO 5 1: Message lost has occurred in common FIFO 5	R
b31 to b14	—	Reserved	These bits are read as 0.	R

This register shows the status of the message lost flags of the FIFO buffers.

RFMLn Flag (Receive FIFO n Message Lost Flag) (n = 0 to 7)

The RFMLn flag is set to 1 when the RFSRn.LOST flag is set to 1, and is set to 0 when the RFSRn.LOST flag is set to 0. The RFMLn flag is cleared when the CANFDG module enters GL_RESET mode.

CFMLn Flag (Common FIFO n Message Lost Flag) (n = 0 to 5)

The CFMLn flag is set to 1 when the CFSRn.LOST flag is set to 1, and is set to 0 when the CFSRn.LOST flag is set to 0. The CFMLn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.34 Receive FIFO Interrupt Status Register (RFISR)

Address(es): CANFDG0.RFISR 8748 02ACh, CANFDG0_NS.RFISR 0748 02ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	RFFIF7	RFFIF6	RFFIF5	RFFIF4	RFFIF3	RFFIF2	RFFIF1	RFFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RFIF7	RFIF6	RFIF5	RFIF4	RFIF3	RFIF2	RFIF1	RFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFIF0	Receive FIFO 0 Interrupt Flag	0: Receive FIFO 0 interrupt condition is not satisfied 1: Receive FIFO 0 interrupt condition is satisfied	R
b1	RFIF1	Receive FIFO 1 Interrupt Flag	0: Receive FIFO 1 interrupt condition is not satisfied 1: Receive FIFO 1 interrupt condition is satisfied	R
b2	RFIF2	Receive FIFO 2 Interrupt Flag	0: Receive FIFO 2 interrupt condition is not satisfied 1: Receive FIFO 2 interrupt condition is satisfied	R
b3	RFIF3	Receive FIFO 3 Interrupt Flag	0: Receive FIFO 3 interrupt condition is not satisfied 1: Receive FIFO 3 interrupt condition is satisfied	R
b4	RFIF4	Receive FIFO 4 Interrupt Flag	0: Receive FIFO 4 interrupt condition is not satisfied 1: Receive FIFO 4 interrupt condition is satisfied	R
b5	RFIF5	Receive FIFO 5 Interrupt Flag	0: Receive FIFO 5 interrupt condition is not satisfied 1: Receive FIFO 5 interrupt condition is satisfied	R
b6	RFIF6	Receive FIFO 6 Interrupt Flag	0: Receive FIFO 6 interrupt condition is not satisfied 1: Receive FIFO 6 interrupt condition is satisfied	R
b7	RFIF7	Receive FIFO 7 Interrupt Flag	0: Receive FIFO 7 interrupt condition is not satisfied 1: Receive FIFO 7 interrupt condition is satisfied	R
b15 to b8	—	Reserved	These bits are read as 0.	R
b16	RFFIF0	Receive FIFO 0 Full Interrupt Flag	0: Receive FIFO 0 full interrupt condition is not satisfied 1: Receive FIFO 0 full interrupt condition is satisfied	R
b17	RFFIF1	Receive FIFO 1 Full Interrupt Flag	0: Receive FIFO 1 full interrupt condition is not satisfied 1: Receive FIFO 1 full interrupt condition is satisfied	R
b18	RFFIF2	Receive FIFO 2 Full Interrupt Flag	0: Receive FIFO 2 full interrupt condition is not satisfied 1: Receive FIFO 2 full interrupt condition is satisfied	R
b19	RFFIF3	Receive FIFO 3 Full Interrupt Flag	0: Receive FIFO 3 full interrupt condition is not satisfied 1: Receive FIFO 3 full interrupt condition is satisfied	R
b20	RFFIF4	Receive FIFO 4 Full Interrupt Flag	0: Receive FIFO 4 full interrupt condition is not satisfied 1: Receive FIFO 4 full interrupt condition is satisfied	R
b21	RFFIF5	Receive FIFO 5 Full Interrupt Flag	0: Receive FIFO 5 full interrupt condition is not satisfied 1: Receive FIFO 5 full interrupt condition is satisfied	R
b22	RFFIF6	Receive FIFO 6 Full Interrupt Flag	0: Receive FIFO 6 full interrupt condition is not satisfied 1: Receive FIFO 6 full interrupt condition is satisfied	R
b23	RFFIF7	Receive FIFO 7 Full Interrupt Flag	0: Receive FIFO 7 full interrupt condition is not satisfied 1: Receive FIFO 7 full interrupt condition is satisfied	R
b31 to b24	—	Reserved	These bits are read as 0.	R

This register shows the status of the interrupt flags of the receive FIFOs.

RFIFn Flag (Receive FIFO n Interrupt Flag) (n = 0 to 7)

The RFIFn flag is set to 1 when the RFSRn.RFIF flag is set to 1, and is set to 0 when the RFSRn.RFIF flag is set to 0.

The RFIFn flag is cleared when the CANFDG module enters GL_RESET mode.

RFFIFn Flag (Receive FIFO n Full Interrupt Flag) (n = 0 to 7)

The RFFIFn flag is set to 1 when the RFSRn.RFFIF flag is set to 1, and is set to 0 when the RFSRn.RFFIF flag is set to 0. The RFFIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.35 Common FIFO Receive Interrupt Status Register (CFRISR)

Address(es): CANFDG0.CFRISR 8748 02B0h, CANFDG0_NS.CFRISR 0748 02B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	CFRIF5	CFRIF4	CFRIF3	CFRIF2	CFRIF1	CFRIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFRIF0	Common FIFO 0 Receive Interrupt Flag	0: Common FIFO 0 receive interrupt condition is not satisfied 1: Common FIFO 0 receive interrupt condition is satisfied	R
b1	CFRIF1	Common FIFO 1 Receive Interrupt Flag	0: Common FIFO 1 receive interrupt condition is not satisfied 1: Common FIFO 1 receive interrupt condition is satisfied	R
b2	CFRIF2	Common FIFO 2 Receive Interrupt Flag	0: Common FIFO 2 receive interrupt condition is not satisfied 1: Common FIFO 2 receive interrupt condition is satisfied	R
b3	CFRIF3	Common FIFO 3 Receive Interrupt Flag	0: Common FIFO 3 receive interrupt condition is not satisfied 1: Common FIFO 3 receive interrupt condition is satisfied	R
b4	CFRIF4	Common FIFO 4 Receive Interrupt Flag	0: Common FIFO 4 receive interrupt condition is not satisfied 1: Common FIFO 4 receive interrupt condition is satisfied	R
b5	CFRIF5	Common FIFO 5 Receive Interrupt Flag	0: Common FIFO 5 receive interrupt condition is not satisfied 1: Common FIFO 5 receive interrupt condition is satisfied	R
b31 to b6	—	Reserved	These bits are read as 0.	R

This register shows the status of the receive interrupt flags of the common FIFOs.

CFRIFn Flag (Common FIFO n Receive Interrupt Flag) (n = 0 to 5)

The CFRIFn flag is set to 1 when the CFSRn.CFRIF flag is set to 1, and is set to 0 when the CFSRn.CFRIF flag is set to 0.

The CFRIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.36 Common FIFO Transmit Interrupt Status Register (CFTISR)

Address(es): CANFDG0.CFTISR 8748 02B4h, CANFDG0_NS.CFTISR 0748 02B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	CFTIF5	CFTIF4	CFTIF3	CFTIF2	CFTIF1	CFTIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFTIF0	Common FIFO 0 Transmit Interrupt Flag	0: Common FIFO 0 transmit interrupt condition is not satisfied 1: Common FIFO 0 transmit interrupt condition is satisfied	R
b1	CFTIF1	Common FIFO 1 Transmit Interrupt Flag	0: Common FIFO 1 transmit interrupt condition is not satisfied 1: Common FIFO 1 transmit interrupt condition is satisfied	R
b2	CFTIF2	Common FIFO 2 Transmit Interrupt Flag	0: Common FIFO 2 transmit interrupt condition is not satisfied 1: Common FIFO 2 transmit interrupt condition is satisfied	R
b3	CFTIF3	Common FIFO 3 Transmit Interrupt Flag	0: Common FIFO 3 transmit interrupt condition is not satisfied 1: Common FIFO 3 transmit interrupt condition is satisfied	R
b4	CFTIF4	Common FIFO 4 Transmit Interrupt Flag	0: Common FIFO 4 transmit interrupt condition is not satisfied 1: Common FIFO 4 transmit interrupt condition is satisfied	R
b5	CFTIF5	Common FIFO 5 Transmit Interrupt Flag	0: Common FIFO 5 transmit interrupt condition is not satisfied 1: Common FIFO 5 transmit interrupt condition is satisfied	R
b31 to b6	—	Reserved	These bits are read as 0.	R

This register shows the status of the transmit interrupt flags of the common FIFOs.

CFTIFn Flag (Common FIFO n Transmit Interrupt Flag) (n = 0 to 5)

The CFTIFn flag is set to 1 when the CFSRn.CFTIF flag is set to 1, and is set to 0 when the CFSRn.CFTIF flag is set to 0.

The CFTIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.37 FIFO Full Interrupt Status Register (FFISR)

Address(es): CANFDG0.FFISR 8748 02C4h, CANFDG0_NS.FFISR 0748 02C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFFIF5	CFFIF4	CFFIF3	CFFIF2	CFFIF1	CFFIF0	RFFIF7	RFFIF6	RFFIF5	RFFIF4	RFFIF3	RFFIF2	RFFIF1	RFFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFFIF0	Receive FIFO 0 Full Interrupt Flag	0: Receive FIFO 0 full interrupt condition is not satisfied 1: Receive FIFO 0 full interrupt condition is satisfied	R
b1	RFFIF1	Receive FIFO 1 Full Interrupt Flag	0: Receive FIFO 1 full interrupt condition is not satisfied 1: Receive FIFO 1 full interrupt condition is satisfied	R
b2	RFFIF2	Receive FIFO 2 Full Interrupt Flag	0: Receive FIFO 2 full interrupt condition is not satisfied 1: Receive FIFO 2 full interrupt condition is satisfied	R
b3	RFFIF3	Receive FIFO 3 Full Interrupt Flag	0: Receive FIFO 3 full interrupt condition is not satisfied 1: Receive FIFO 3 full interrupt condition is satisfied	R
b4	RFFIF4	Receive FIFO 4 Full Interrupt Flag	0: Receive FIFO 4 full interrupt condition is not satisfied 1: Receive FIFO 4 full interrupt condition is satisfied	R
b5	RFFIF5	Receive FIFO 5 Full Interrupt Flag	0: Receive FIFO 5 full interrupt condition is not satisfied 1: Receive FIFO 5 full interrupt condition is satisfied	R
b6	RFFIF6	Receive FIFO 6 Full Interrupt Flag	0: Receive FIFO 6 full interrupt condition is not satisfied 1: Receive FIFO 6 full interrupt condition is satisfied	R
b7	RFFIF7	Receive FIFO 7 Full Interrupt Flag	0: Receive FIFO 7 full interrupt condition is not satisfied 1: Receive FIFO 7 full interrupt condition is satisfied	R
b8	CFFIF0	Common FIFO 0 Full Interrupt Flag	0: Common FIFO 0 full interrupt condition is not satisfied 1: Common FIFO 0 full interrupt condition is satisfied	R
b9	CFFIF1	Common FIFO 1 Full Interrupt Flag	0: Common FIFO 1 full interrupt condition is not satisfied 1: Common FIFO 1 full interrupt condition is satisfied	R
b10	CFFIF2	Common FIFO 2 Full Interrupt Flag	0: Common FIFO 2 full interrupt condition is not satisfied 1: Common FIFO 2 full interrupt condition is satisfied	R
b11	CFFIF3	Common FIFO 3 Full Interrupt Flag	0: Common FIFO 3 full interrupt condition is not satisfied 1: Common FIFO 3 full interrupt condition is satisfied	R
b12	CFFIF4	Common FIFO 4 Full Interrupt Flag	0: Common FIFO 4 full interrupt condition is not satisfied 1: Common FIFO 4 full interrupt condition is satisfied	R
b13	CFFIF5	Common FIFO 5 Full Interrupt Flag	0: Common FIFO 5 full interrupt condition is not satisfied 1: Common FIFO 5 full interrupt condition is satisfied	R
b31 to b14	—	Reserved	These bits are read as 0.	R

This register shows the status of the FIFO full interrupt flags of the common FIFOs.

RFFIFn Flag (Receive FIFO n Full Interrupt Flag) (n = 0 to 7)

The RFFIFn flag is set to 1 when the RFSRn.RFFIF flag is set to 1, and is set to 0 when the RFSRn.RFFIF flag is set to 0. The RFFIFn flag is cleared when the CANFDG module enters GL_RESET mode.

CFFIFn Flag (Common FIFO n Full Interrupt Flag) (n = 0 to 5)

The CFFIFn flag is set to 1 when the CFSRn.CFFIF flag is set to 1, and is set to 0 when the CFSRn.CFFIF flag is set to 0. The CFFIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.38 Common FIFO Message Overwrite Status Register (CFOWSR)

Address(es): CANFDG0.CFOWSR 8748 02C0h, CANFDG0_NS.CFOWSR 0748 02C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	CFOWF5	CFOWF4	CFOWF3	CFOWF2	CFOWF1	CFOWF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFOWF0	Common FIFO 0 Message Overwrite Flag	0: Common FIFO 0 message overwrite condition is not satisfied 1: Common FIFO 0 message overwrite condition is satisfied	R
b1	CFOWF1	Common FIFO 1 Message Overwrite Flag	0: Common FIFO 1 message overwrite condition is not satisfied 1: Common FIFO 1 message overwrite condition is satisfied	R
b2	CFOWF2	Common FIFO 2 Message Overwrite Flag	0: Common FIFO 2 message overwrite condition is not satisfied 1: Common FIFO 2 message overwrite condition is satisfied	R
b3	CFOWF3	Common FIFO 3 Message Overwrite Flag	0: Common FIFO 3 message overwrite condition is not satisfied 1: Common FIFO 3 message overwrite condition is satisfied	R
b4	CFOWF4	Common FIFO 4 Message Overwrite Flag	0: Common FIFO 4 message overwrite condition is not satisfied 1: Common FIFO 4 message overwrite condition is satisfied	R
b5	CFOWF5	Common FIFO 5 Message Overwrite Flag	0: Common FIFO 5 message overwrite condition is not satisfied 1: Common FIFO 5 message overwrite condition is satisfied	R
b31 to b6	—	Reserved	These bits are read as 0.	R

This register shows the status of the message overwrite flags of the common FIFOs. This register is valid only in gateway FIFO mode.

CFOWFn Flag (Common FIFO n Message Overwrite Flag) (n = 0 to 5)

The CFOWFn flag is set to 1 when the CFSRn.CFOWF flag is set to 1, and is set to 0 when the CFSRn.CFOWF flag is set to 0.

The CFOWFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.39 Common FIFO One Frame Receive Interrupt Status Register (CFORISR)

Address(es): CANFDG0.CFORISR 8748 02B8h, CANFDG0_NS.CFORISR 0748 02B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	CFORIF5	CFORIF4	CFORIF3	CFORIF2	CFORIF1	CFORIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFORIF0	Common FIFO 0 One Frame Receive Interrupt Flag	0: Common FIFO 0 one frame receive interrupt condition is not satisfied 1: Common FIFO 0 one frame receive interrupt condition is satisfied	R
b1	CFORIF1	Common FIFO 1 One Frame Receive Interrupt Flag	0: Common FIFO 1 one frame receive interrupt condition is not satisfied 1: Common FIFO 1 one frame receive interrupt condition is satisfied	R
b2	CFORIF2	Common FIFO 2 One Frame Receive Interrupt Flag	0: Common FIFO 2 one frame receive interrupt condition is not satisfied 1: Common FIFO 2 one frame receive interrupt condition is satisfied	R
b3	CFORIF3	Common FIFO 3 One Frame Receive Interrupt Flag	0: Common FIFO 3 one frame receive interrupt condition is not satisfied 1: Common FIFO 3 one frame receive interrupt condition is satisfied	R
b4	CFORIF4	Common FIFO 4 One Frame Receive Interrupt Flag	0: Common FIFO 4 one frame receive interrupt condition is not satisfied 1: Common FIFO 4 one frame receive interrupt condition is satisfied	R
b5	CFORIF5	Common FIFO 5 One Frame Receive Interrupt Flag	0: Common FIFO 5 one frame receive interrupt condition is not satisfied 1: Common FIFO 5 one frame receive interrupt condition is satisfied	R
b31 to b6	—	Reserved	These bits are read as 0.	R

This register shows the status of the one frame receive interrupt flags of the common FIFOs.

CFORIFn Flag (Common FIFO n One Frame Receive Interrupt Flag) (n = 0 to 5)

The CFORIFn flag is set to 1 when the CFSRn.CFORIF flag is set to 1, and is set to 0 when the CFSRn.CFORIF flag is set to 0.

The CFORIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.40 Common FIFO One Frame Transmit Interrupt Status Register (CFOTISR)

Address(es): CANFDG0.CFOTISR 8748 02BCh, CANFDG0_NS.CFOTISR 0748 02BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	CFOTIF5	CFOTIF4	CFOTIF3	CFOTIF2	CFOTIF1	CFOTIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFOTIF0	Common FIFO 0 One Frame Transmit Interrupt Flag	0: Common FIFO 0 one frame transmit interrupt condition is not satisfied 1: Common FIFO 0 one frame transmit interrupt condition is satisfied	R
b1	CFOTIF1	Common FIFO 1 One Frame Transmit Interrupt Flag	0: Common FIFO 1 one frame transmit interrupt condition is not satisfied 1: Common FIFO 1 one frame transmit interrupt condition is satisfied	R
b2	CFOTIF2	Common FIFO 2 One Frame Transmit Interrupt Flag	0: Common FIFO 2 one frame transmit interrupt condition is not satisfied 1: Common FIFO 2 one frame transmit interrupt condition is satisfied	R
b3	CFOTIF3	Common FIFO 3 One Frame Transmit Interrupt Flag	0: Common FIFO 3 one frame transmit interrupt condition is not satisfied 1: Common FIFO 3 one frame transmit interrupt condition is satisfied	R
b4	CFOTIF4	Common FIFO 4 One Frame Transmit Interrupt Flag	0: Common FIFO 4 one frame transmit interrupt condition is not satisfied 1: Common FIFO 4 one frame transmit interrupt condition is satisfied	R
b5	CFOTIF5	Common FIFO 5 One Frame Transmit Interrupt Flag	0: Common FIFO 5 one frame transmit interrupt condition is not satisfied 1: Common FIFO 5 one frame transmit interrupt condition is satisfied	R
b31 to b6	—	Reserved	These bits are read as 0.	R

This register shows the status of the one frame transmit interrupt flags of the common FIFOs.

CFOTIFn Flag (Common FIFO n One Frame Transmit Interrupt Flag) (n = 0 to 5)

The CFOTIFn flag is set to 1 when the CFSRn.CFOTIF flag is set to 1, and is set to 0 when the CFSRn.CFOTIF flag is set to 0.

The CFOTIFn flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.41 DMA Transfer Control Register (DTCR)

Address(es): CANFDG0.DTCR 8748 1330h, CANFDG0_NS.DTCR 0748 1330h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CFDTE1	CFDTE0	RFDTE7	RFDTE6	RFDTE5	RFDTE4	RFDTE3	RFDTE2	RFDTE1	RFDTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTE0	Receive FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b1	RFDTE1	Receive FIFO 1 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b2	RFDTE2	Receive FIFO 2 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b3	RFDTE3	Receive FIFO 3 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b4	RFDTE4	Receive FIFO 4 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b5	RFDTE5	Receive FIFO 5 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b6	RFDTE6	Receive FIFO 6 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b7	RFDTE7	Receive FIFO 7 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b8	CFDTE0	Channel 0 Common FIFO DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b9	CFDTE1	Channel 1 Common FIFO DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of DTC/DMA transfer operation.

RFDTEn Bit (Receive FIFO n DMA Transfer Enable) (n = 0 to 7)

The RFDTEn bit enables or disables DTC/DMA transfer request for receive FIFO n.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.

CFDTE0 Bit (Channel 0 Common FIFO DMA Transfer Enable)

The CFDTE0 bit enables or disables DTC/DMA transfer request for common FIFO 0 of channel 0.
Receive DTC/DMA transfers using common FIFOs can only be performed on common FIFO 0 of each channel.
Do not enable DTC/DMA transfer for the common FIFO 0 that is configured as transmit FIFO or gateway FIFO.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.

CFDTE1 Bit (Channel 1 Common FIFO DMA Transfer Enable)

The CFDTE1 bit enables or disables DTC/DMA transfer request for common FIFO 0 of channel 1.

Receive DTC/DMA transfers using common FIFOs can only be performed on common FIFO 0 of each channel (common FIFO 3 for channel 1).
Do not enable DTC/DMA transfer for the common FIFO 3 that is configured as transmit FIFO or gateway FIFO.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.

45.2.42 DMA Transfer Status Register (DTSR)

Address(es): CANFDG0.DTSR 8748 1334h, CANFDG0_NS.DTSR 0748 1334h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CFDTS1	CFDTS0	RFDTS7	RFDTS6	RFDTS5	RFDTS4	RFDTS3	RFDTS2	RFDTS1	RFDTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTS0	Receive FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b1	RFDTS1	Receive FIFO 1 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b2	RFDTS2	Receive FIFO 2 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b3	RFDTS3	Receive FIFO 3 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b4	RFDTS4	Receive FIFO 4 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b5	RFDTS5	Receive FIFO 5 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b6	RFDTS6	Receive FIFO 6 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b7	RFDTS7	Receive FIFO 7 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b8	CFDTS0	Channel 0 Common FIFO DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b9	CFDTS1	Channel 1 Common FIFO DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b31 to b10	—	Reserved	These bits are read as 0.	R

This register shows the status of the DTC/DMA transfer.

RFDTSn Flag (Receive FIFO n DMA Transfer Status Flag) (n = 0 to 7)

The RFDTSn flag is automatically set to 1 when the DTCR.RFDTEn bit is set to 1 and the receive FIFO 0 is not empty.
This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTEn bit is set to 0 or the receive FIFO n is empty.
When the DTCR.RFDTEn bit is set to 0 while DTC/DMA transfer for the receive FIFO n is in progress, the RFDTSn flag becomes 0 when the DTC/DMA transfer is completed.
This flag is set to 0 when the CANFDG module enters GL_RESET mode.

CFDTS0 Flag (Channel 0 Common FIFO DMA Transfer Status Flag)

The CFDTS0 flag is automatically set to 1 when the DTCR.CFDTE0 bit is set to 1 and the common FIFO 0 of channel 0 is not empty.
This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.CFDTE0 bit is set to 0 or the common FIFO 0 of channel 0 is empty.
When the DTCR.CFDTE0 bit is set to 0 while DTC/DMA transfer for the common FIFO 0 of channel 0 is in progress,

the CFDTS0 flag becomes 0 when the DTC/DMA transfer is complete.
This flag is set to 0 when the CANFDG module enters GL_RESET mode.

CFDTS1 Flag (Channel 1 Common FIFO DMA Transfer Status Flag)

The CFDTS1 flag is automatically set to 1 when the DTCR.CFDTE1 bit is set to 1 and the common FIFO 0 of channel 1 (common FIFO 3) is not empty.
This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.CFDTE1 bit is set to 0 or the common FIFO 3 is empty.
When the DTCR.CFDTE1 bit is set to 0 while DTC/DMA transfer for the common FIFO 3 is in progress, the CFDTS1 flag becomes 0 when the DTC/DMA transfer is complete.
This flag is set to 0 when the CANFDG module enters GL_RESET mode.

45.2.43 Transmit DMA Transfer Control Register (TDTCR)

Address(es): CANFDG0.TDTCR 8748 1340h, CANFDG0_NS.TDTCR 0748 1340h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTE1	CFDTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TQDTE13	TQDTE03	—	—	—	—	—	—	TQDTE10	TQDTE00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQDTE00	Channel 0 Transmit Queue 0 DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b1	TQDTE10	Channel 1 Transmit Queue 0 DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TQDTE03	Channel 0 Transmit Queue 3 DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b9	TQDTE13	Channel 1 Transmit Queue 3 DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	CFDTE0	Channel 0 Common FIFO DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b17	CFDTE1	Channel 1 Common FIFO DMA Transfer Enable	0: DTC/DMA transmit transfer request is disabled 1: DTC/DMA transmit transfer request is enabled	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of DTC/DMA transmit transfer operations.

TQDTEi0 Bit (Channel i Transmit Queue 0 DMA Transfer Enable) (i = 0, 1)

The TQDTEi0 bit enables or disables a DTC/DMA transmit transfer request for transmit queue 0.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.

TQDTEi3 Bit (Channel i Transmit Queue 3 DMA Transfer Enable) (i = 0, 1)

The TQDTEi3 bit enables or disables a DTC/DMA transmit transfer request for transmit queue 3.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.

CFDTE0 Bit (Channel 0 Common FIFO DMA Transfer Enable)

The CFDTE0 bit enables or disables a DTC/DMA transmit transfer request for common FIFO 2 of channel 0.
DTC/DMA transmit transfers using common FIFO can only be performed using common FIFO 2 of each channel.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.
When common FIFO 2 is configured as receive FIFO mode or gateway FIFO mode, do not set this bit to 1.

CFDTE1 Bit (Channel 1 Common FIFO DMA Transfer Enable)

The CFDTE1 bit enables or disables a DTC/DMA transmit transfer request for common FIFO 2 of channel 1.
DTC/DMA transmit transfers using common FIFO can only be performed using common FIFO 2 of each channel (common FIFO 5 for channel 1).
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFDG module enters GL_RESET mode.
When common FIFO 5 is configured as receive FIFO mode or gateway FIFO mode, do not set this bit to 1.

45.2.44 Transmit DMA Transfer Status Register (TDTSR)

Address(es): CANFDG0.TDTSR 8748 1344h, CANFDG0_NS.TDTSR 0748 1344h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTS1	CFDTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TQDTS13	TQDTS03	—	—	—	—	—	—	TQDTS10	TQDTS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQDTS00	Channel 0 Transmit Queue 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b1	TQDTS10	Channel 1 Transmit Queue 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	TQDTS03	Channel 0 Transmit Queue 3 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b9	TQDTS13	Channel 1 Transmit Queue 3 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b16	CFDTS0	Channel 0 Common FIFO DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b17	CFDTS1	Channel 1 Common FIFO DMA Transfer Status Flag	0: DTC/DMA transfer is stopped 1: DTC/DMA transfer is enabled	R
b31 to b18	—	Reserved	These bits are read as 0.	R

This register shows the status of the DTC/DMA transfer. The same value as the TDTCR register can be read.

TQDTSi0 Flag (Channel i Transmit Queue 0 DMA Transfer Status Flag) (i = 0, 1)

The TQDTSi0 flag is set to 1 when the TDTCR.TQDTEi0 bit is set to 1, and is set to 0 when the TDTCR.TQDTEi0 bit is set to 0.

The TQDTSi0 flag is cleared when the CANFDG module enters GL_RESET mode.

TQDTSi3 Flag (Channel i Transmit Queue 3 DMA Transfer Status Flag) (i = 0, 1)

The TQDTSi3 flag is set to 1 when the TDTCR.TQDTEi3 bit is set to 1, and is set to 0 when the TDTCR.TQDTEi3 bit is set to 0.

The TQDTSi3 flag is cleared when the CANFDG module enters GL_RESET mode.

CFDTSi Flag (Channel i Common FIFO DMA Transfer Status Flag) (i = 0, 1)

The CFDTSi flag is set to 1 when the TDTCR.CFDTEi bit is set to 1, and is set to 0 when the TDTCR.CFDTEi bit is set to 0.

The CFDTSi flag is cleared when the CANFDG module enters GL_RESET mode.

45.2.45 Receive Interrupt Status Register (RISR)

Address(es): CANFD0.RISR 8748 1350h, CANFD1.RISR 8748 1354h,
CANFD0_NS.RISR 0748 1350h, CANFD1_NS.RISR 0748 1354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CFORIF2	CFORIF1	CFORIF0	—	CFFIF2	CFFIF1	CFFIF0	—	—	—	—	—	CFRIF2	CFRIF1	CFRIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TQORIF2	TQORIF1	TQORIF0	—	—	—	—	—	TQFIF2	TQFIF1	TQFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQFIF0	Transmit Queue 0 Full Interrupt Monitor Flag	0: Transmit queue 0 full interrupt condition is not satisfied 1: Transmit queue 0 full interrupt condition is satisfied	R
b1	TQFIF1	Transmit Queue 1 Full Interrupt Monitor Flag	0: Transmit queue 1 full interrupt condition is not satisfied 1: Transmit queue 1 full interrupt condition is satisfied	R
b2	TQFIF2	Transmit Queue 2 Full Interrupt Monitor Flag	0: Transmit queue 2 full interrupt condition is not satisfied 1: Transmit queue 2 full interrupt condition is satisfied	R
b7 to b3	—	Reserved	These bits are read as 0.	R
b8	TQORIF0	Transmit Queue 0 One Frame Receive Interrupt Monitor Flag	0: Transmit queue 0 one frame receive interrupt condition is not satisfied 1: Transmit queue 0 one frame receive interrupt condition is satisfied	R
b9	TQORIF1	Transmit Queue 1 One Frame Receive Interrupt Monitor Flag	0: Transmit queue 1 one frame receive interrupt condition is not satisfied 1: Transmit queue 1 one frame receive interrupt condition is satisfied	R
b10	TQORIF2	Transmit Queue 2 One Frame Receive Interrupt Monitor Flag	0: Transmit queue 2 one frame receive interrupt condition is not satisfied 1: Transmit queue 2 one frame receive interrupt condition is satisfied	R
b15 to b11	—	Reserved	These bits are read as 0.	R
b16	CFRIF0	Common FIFO 0 Receive Interrupt Monitor Flag	0: Common FIFO 0 receive interrupt condition is not satisfied 1: Common FIFO 0 receive interrupt condition is satisfied	R
b17	CFRIF1	Common FIFO 1 Receive Interrupt Monitor Flag	0: Common FIFO 1 receive interrupt condition is not satisfied 1: Common FIFO 1 receive interrupt condition is satisfied	R
b18	CFRIF2	Common FIFO 2 Receive Interrupt Monitor Flag	0: Common FIFO 2 receive interrupt condition is not satisfied 1: Common FIFO 2 receive interrupt condition is satisfied	R
b23 to b19	—	Reserved	These bits are read as 0.	R
b24	CFFIF0	Common FIFO 0 Full Interrupt Monitor Flag	0: Common FIFO 0 full interrupt condition is not satisfied 1: Common FIFO 0 full interrupt condition is satisfied	R
b25	CFFIF1	Common FIFO 1 Full Interrupt Monitor Flag	0: Common FIFO 1 full interrupt condition is not satisfied 1: Common FIFO 1 full interrupt condition is satisfied	R
b26	CFFIF2	Common FIFO 2 Full Interrupt Monitor Flag	0: Common FIFO 2 full interrupt condition is not satisfied 1: Common FIFO 2 full interrupt condition is satisfied	R
b27	—	Reserved	This bit is read as 0.	R
b28	CFORIF0	Common FIFO 0 One Frame Receive Interrupt Monitor Flag	0: Common FIFO 0 one frame receive interrupt condition is not satisfied 1: Common FIFO 0 one frame receive interrupt condition is satisfied	R

Bit	Symbol	Bit Name	Description	R/W
b29	CFORIF1	Common FIFO 1 One Frame Receive Interrupt Monitor Flag	0: Common FIFO 1 one frame receive interrupt condition is not satisfied 1: Common FIFO 1 one frame receive interrupt condition is satisfied	R
b30	CFORIF2	Common FIFO 2 One Frame Receive Interrupt Monitor Flag	0: Common FIFO 2 one frame receive interrupt condition is not satisfied 1: Common FIFO 2 one frame receive interrupt condition is satisfied	R
b31	—	Reserved	This bit is read as 0.	R

TQFIFm Flag (Transmit Queue m Full Interrupt Monitor Flag) (m = 0 to 2)

The TQFIFm flag is automatically set to 1 when the transmit queue full interrupt flag of the corresponding channel becomes 1 while interrupts are enabled.
The TQFIFm flag is set to 0:

- When interrupts are disabled
- When the transmit queue result status flag is cleared
- When the CANFDG module enters GL_RESET mode or the CAN channel enters CH_RESET mode

TQORIFm Flag (Transmit Queue m One Frame Receive Interrupt Monitor Flag) (m = 0 to 2)

The TQORIFm flag is automatically set to 1 when the transmit queue one frame receive interrupt flag of the corresponding channel becomes 1 while interrupts are enabled.
The TQORIFm flag is set to 0:

- When interrupts are disabled
- When the transmit queue result status flag is cleared
- When the CANFDG module enters GL_RESET mode or the CAN channel enters CH_RESET mode

CFRIFm Flag (Common FIFO m Receive Interrupt Monitor Flag) (m = 0 to 2)

The CFRIFm flag is automatically set to 1 when the common FIFO receive interrupt flag of the corresponding channel becomes 1 while interrupts are enabled.
The CFRIFm flag is set to 0:

- When interrupts are disabled
- When the common FIFO reception result status flag is cleared
- When the CANFDG module enters GL_RESET mode or the CAN channel enters CH_RESET mode

CFFIFm Flag (Common FIFO m Full Interrupt Monitor Flag) (m = 0 to 2)

The CFFIFm flag is automatically set to 1 when the common FIFO full interrupt flag of the corresponding channel becomes 1 while interrupts are enabled.
The CFFIFm flag is set to 0:

- When interrupts are disabled
- When the common FIFO reception result status flag is cleared
- When the CANFDG module enters GL_RESET mode or the CAN channel enters CH_RESET mode

CFORIFm Flag (Common FIFO m One Frame Receive Interrupt Monitor Flag) (m = 0 to 2)

The CFORIFm flag is automatically set to 1 when the common FIFO one frame receive interrupt flag of the corresponding channel becomes 1 while interrupts are enabled.
The CFORIFm flag is set to 0:

- When interrupts are disabled
- When the common FIFO reception result status flag is cleared
- When the CANFDG module enters GL_RESET mode or the CAN channel enters CH_RESET mode

45.2.46 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Address(es): CANFDG0.TMCR0 8748 02D0h to CANFDG0.TMCR7 8748 02D7h,
CANFDG0.TMCR32 8748 02F0h to CANFDG0.TMCR39 8748 02F7h,
CANFDG0.TMCR64 8748 0310h to CANFDG0.TMCR71 8748 0317h,
CANFDG0.TMCR96 8748 0330h to CANFDG0.TMCR103 8748 0337h,
CANFDG0_NS.TMCR0 0748 02D0h to CANFDG0_NS.TMCR7 0748 02D7h,
CANFDG0_NS.TMCR32 0748 02F0h to CANFDG0_NS.TMCR39 0748 02F7h,
CANFDG0_NS.TMCR64 0748 0310h to CANFDG0_NS.TMCR71 0748 0317h,
CANFDG0_NS.TMCR96 0748 0330h to CANFDG0_NS.TMCR103 0748 0337h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ONESHOT	TARQ	TXRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQ	Transmission Request	0: Message transmission is not requested 1: Message transmission is requested	R/W
b1	TARQ	Transmission Abort Request	0: Message transmission abort is not requested 1: Message transmission abort is requested	R/W
b2	ONESHOT	One-shot Transmission Enable	0: One-shot transmission is disabled 1: One-shot transmission is enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmit message buffer functions.

TXRQ Bit (Transmission Request)

When the TXRQ bit is set to 1, the CANFDG module attempts to transmit the message stored in the corresponding message buffer.
This bit can only be written in CH_HALT or CH_OPERATION mode.
This bit cannot be set to 1 if the corresponding transmit message buffer is linked to the common FIFO configured in transmit FIFO or gateway FIFO mode, or is a part of the transmit queue.
This bit can be set to 1 only when the TMSRn.TXRF[1:0] flags are set to 00b.
This bit cannot be directly cleared by a CPU write access. The TXRQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort, requested by the TARQ bit
- When a CAN bus error or arbitration lost is detected while the ONESHOT bit is set to 1
- When the CANFDG module enters GL_RESET or the CAN channel enters CH_RESET mode.

TARQ Bit (Transmission Abort Request)

When the TARQ bit is set to 1, the CANFDG module attempts to abort transmission of the frame stored in the corresponding message buffer.
In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame will be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (pin for reception) before it starts transmission from the selected message buffer.
This bit can only be written in CH_HALT or CH_OPERATION mode. This bit can be set to 1 only when the TXRQ bit is set to 1.
This bit cannot be set to 0 by a CPU write access. This bit is set to 0 if writing 1 by CPU and clearing by the CAN channel occur at the same time.
The TARQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort
- When a CAN bus error or arbitration lost is detected
- When the CANFDG module enters GL_RESET or the CAN channel enters CH_RESET mode.

ONESHOT Bit (One-shot Transmission Enable)

When the ONESHOT bit is set to 1, the CANFDG module attempts to transmit the message only once.
If the transmission is successful, the TMSRn.TXRF[1:0] flags are set to 10b or 11b. If a transmission fails due to a bus error or an arbitration lost detection, the transmission is automatically aborted and TMSRn.TXRF[1:0] flags are set to 01b.
The ONESHOT bit remains 1 if the transmission has completed successfully or aborted due to a CAN bus error or an arbitration lost detection.
This bit can only be written in CH_HALT or CH_OPERATION mode.
Set this bit to 1 at the same time as the TXRQ bit. To set this bit to 0, write 0.
If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.
The ONESHOT bit is automatically set to 0 when the CANFDG module enters GL_RESET or the CAN channel enters CH_RESET mode.

45.2.47 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Address(es): CANFDG0.TMSR0 8748 07D0h to CANFDG0.TMSR7 8748 07D7h,
CANFDG0.TMSR32 8748 07F0h to CANFDG0.TMSR39 8748 07F7h,
CANFDG0.TMSR64 8748 0810h to CANFDG0.TMSR71 8748 0817h,
CANFDG0.TMSR96 8748 0830h to CANFDG0.TMSR103 8748 0837h,
CANFDG0_NS.TMSR0 0748 07D0h to CANFDG0_NS.TMSR7 0748 07D7h,
CANFDG0_NS.TMSR32 0748 07F0h to CANFDG0_NS.TMSR39 0748 07F7h,
CANFDG0_NS.TMSR64 0748 0810h to CANFDG0_NS.TMSR71 0748 0817h,
CANFDG0_NS.TMSR96 0748 0830h to CANFDG0_NS.TMSR103 0748 0837h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TARQS	TXRQS	TXRF[1:0]		TXSF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXSF	Transmission Status Flag	0: No transmission is in progress 1: Transmission is in progress	R
b2, b1	TXRF[1:0]	Transmission Result Flag	b2 b1 0 0: No result (transmission is not requested or in progress) 0 1: Transmission was aborted. 1 0: Transmission was successful and transmission abort is not requested. 1 1: Transmission was successful and transmission abort is requested.	R/W
b3	TXRQS	Transmission Request Status Flag	0: Transmission is not requested 1: Transmission is requested	R
b4	TARQS	Transmission Abort Request Status Flag	0: Transmission abort is not requested 1: Transmission abort is requested	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register shows the status of the transmission and transmission abort for the transmit message buffers.

TXSF Flag (Transmission Status Flag)

The TXSF flag is automatically set to 1 at the start of the transmission from the corresponding transmit message buffer. This flag is automatically set to 0 when:

- Transmission stops
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

TXRF[1:0] Flags (Transmission Result Flag)

The TXRF[1:0] flags show the transmission result for the transmit message buffer.
These flags can only be written in CH_HALT or CH_OPERATION mode.
Each bit of these flags is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The TXRF[1:0] flags are automatically set to 00b when the CANFDG module enters GL_RESET or the CAN channel enters CH_RESET mode.

TXRQS Flag (Transmission Request Status Flag)

The TXRQS flag reflects the value of the TMCRn.TXRQ bit.
The TXRQS flag is set to 1 when the TMCRn.TXRQ bit is set to 1. This flag is set to 0 when the TMCRn.TXRQ bit is set to 0.

TARQS Flag (Transmission Abort Request Status Flag)

The TARQS flag reflects the value of the TMCRn.TARQ bit.

The TARQS flag is set to 1 when the TMCRn.TARQ bit is set to 1. This flag is set to 0 when the TMCRn.TARQ bit is set to 0.

45.2.48 Transmit Message Buffer Transmission Request Status Register m (TMTRSRm) (m = 0 to 3)

Address(es): CANFDG0.TMTRSR0 8748 0CD0h, CANFDG0.TMTRSR1 8748 0CD4h, CANFDG0.TMTRSR2 8748 0CD8h, CANFDG0.TMTRSR3 8748 0CDCh,
CANFDG0_NS.TMTRSR0 0748 0CD0h, CANFDG0_NS.TMTRSR1 0748 0CD4h, CANFDG0_NS.TMTRSR2 0748 0CD8h, CANFDG0_NS.TMTRSR3 0748 0CDCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TXRQS7	TXRQS6	TXRQS5	TXRQS4	TXRQS3	TXRQS2	TXRQS1	TXRQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQS0	Transmit Message Buffer n Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer n, 1: Transmission is requested for the transmit message buffer n,	R
b1	TXRQS1			R
b2	TXRQS2			R
b3	TXRQS3			R
b4	TXRQS4			R
b5	TXRQS5			R
b6	TXRQS6			R
b7	TXRQS7			R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: For the transmit message buffer numbers corresponding to each flag, refer to Table 45.5.

This register shows the status of the transmission request in each transmit message buffer.

TXRQSj Flag (Transmit Message Buffer n Transmission Request Status Flag) (j = 0 to 7; n = m × 32 + j)

The TXRQSj flag indicates status of the TMCRn.TXRQ bit.

Each flag is set to 1 only when the TMCRn.TXRQ bit is set to 1 and the message buffer does not belong to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TXRQ bit is set to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

Table 45.5 TXRQSj Flag and the Transmit Message Buffer Number

Channel	Register Number (m)	Bit Number (j)	Transmit Message Buffer Number (n)
Channel 0	0	0	0
		1	1
		:	:
		7	7
	1	0	32
		1	33
		:	:
		7	39
Channel 1	2	0	64
		1	65
		:	:
		7	71
	3	0	96
		1	97
		:	:
		7	103

45.2.49 Transmit Message Buffer Transmission Abort Request Status Register m (TMARSRm) (m = 0 to 3)

Address(es): CANFDG0.TMARSR0 8748 0D70h, CANFDG0.TMARSR1 8748 0D74h, CANFDG0.TMARSR2 8748 0D78h, CANFDG0.TMARSR3 8748 0D7Ch,
CANFDG0_NS.TMARSR0 0748 0D70h, CANFDG0_NS.TMARSR1 0748 0D74h, CANFDG0_NS.TMARSR2 0748 0D78h, CANFDG0_NS.TMARSR3 0748 0D7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TARQS7	TARQS6	TARQS5	TARQS4	TARQS3	TARQS2	TARQS1	TARQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TARQS0	Transmit Message Buffer n Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer n, 1: Transmission abort is requested for the transmit message buffer n,	R
b1	TARQS1			R
b2	TARQS2			R
b3	TARQS3			R
b4	TARQS4			R
b5	TARQS5			R
b6	TARQS6			R
b7	TARQS7			R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: For the transmit message buffer numbers corresponding to each flag, refer to Table 45.5.

This register shows the status of the transmission abort request in each transmit message buffer.

TARQSj Flag (Transmit Message Buffer n Transmission Abort Request Status Flag)
(j = 0 to 7; n = m × 32 + j)

The TARQSj flag indicates status of the TMCRn.TARQ bit.
Each flag is set to 1 when the TMCRn.TARQ bit is set to 1 or the message buffer belongs to a transmit queue.
Each flag is automatically set to 0 when:

- The TMCRn.TARQ bit is set to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

45.2.50 Transmit Message Buffer Transmission Completion Status Register m (TMTCSRm) (m = 0 to 3)

Address(es): CANFDG0.TMTCSR0 8748 0E10h, CANFDG0.TMTCSR1 8748 0E14h, CANFDG0.TMTCSR2 8748 0E18h, CANFDG0.TMTCSR3 8748 0E1Ch,
CANFDG0_NS.TMTCSR0 0748 0E10h, CANFDG0_NS.TMTCSR1 0748 0E14h, CANFDG0_NS.TMTCSR2 0748 0E18h, CANFDG0_NS.TMTCSR3 0748 0E1Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TXCF7	TXCF6	TXCF5	TXCF4	TXCF3	TXCF2	TXCF1	TXCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXCF0	Transmit Message Buffer n Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer n, 1: Transmission is completed for the transmit message buffer n,	R
b1	TXCF1			R
b2	TXCF2			R
b3	TXCF3			R
b4	TXCF4			R
b5	TXCF5			R
b6	TXCF6			R
b7	TXCF7			R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: For the transmit message buffer numbers corresponding to each flag, refer to Table 45.5.

This register shows the transmission completion status for each transmit message buffer.

TXCFj Flag (Transmit Message Buffer n Transmission Completion Status) (j = 0 to 7; n = m × 32 + j)

The TXCFj flag indicates the transmission completion status for the transmit message buffer n.
Each flag is automatically set to 1 when the TMSRn.TXRF[1] bit is set to 1.
Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1] bit is set to 0
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

45.2.51 Transmit Message Buffer Transmission Abort Status Register m (TMTASRm) (m = 0 to 3)

Address(es): CANFDG0.TMTASR0 8748 0EB0h, CANFDG0.TMTASR1 8748 0EB4h, CANFDG0.TMTASR2 8748 0EB8h, CANFDG0.TMTASR3 8748 0EBCh,
CANFDG0_NS.TMTASR0 0748 0EB0h, CANFDG0_NS.TMTASR1 0748 0EB4h, CANFDG0_NS.TMTASR2 0748 0EB8h, CANFDG0_NS.TMTASR3 0748 0EBCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TAF7	TAF6	TAF5	TAF4	TAF3	TAF2	TAF1	TAF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAF0	Transmit Message Buffer n Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer n. 1: Transmission is aborted for the transmit message buffer n.	R
b1	TAF1			R
b2	TAF2			R
b3	TAF3			R
b4	TAF4			R
b5	TAF5			R
b6	TAF6			R
b7	TAF7			R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: For the transmit message buffer numbers corresponding to each flag, refer to Table 45.5.

This register shows the transmission abort status for each transmit message buffer.

TAFj Flag (Transmit Message Buffer n Transmission Abort Status Flag) (j = 0 to 7; n = m × 32 + j)

The TAFj flag indicates the transmission abort status for the transmit message buffer n.
Each flag is automatically set to 1 when the TMSRn.TXRF[1:0] flags are set to 01b.
Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1:0] flags are set to 00b
- The CANFDG module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

45.2.52 Transmit Message Buffer Interrupt Enable Register m (TMIERm) (m = 0 to 3)

Address(es): CANFDG0.TMIER0 8748 0F50h, CANFDG0.TMIER1 8748 0F54h, CANFDG0.TMIER2 8748 0F58h, CANFDG0.TMIER3 8748 0F5Ch, CANFDG0_NS.TMIER0 0748 0F50h, CANFDG0_NS.TMIER1 0748 0F54h, CANFDG0_NS.TMIER2 0748 0F58h, CANFDG0_NS.TMIER3 0748 0F5Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TMIE7	TMIE6	TMIE5	TMIE4	TMIE3	TMIE2	TMIE1	TMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMIE0	Transmit Message Buffer n Interrupt Enable	0: Interrupt for transmit message buffer n is disabled. 1: Interrupt for transmit message buffer n is enabled.	R/W
b1	TMIE1			R/W
b2	TMIE2			R/W
b3	TMIE3			R/W
b4	TMIE4			R/W
b5	TMIE5			R/W
b6	TMIE6			R/W
b7	TMIE7			R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: For the transmit message buffer numbers corresponding to each flag, refer to Table 45.5.

This register enables or disables the interrupt for each transmit message buffer.

TMIEj Bit (Transmit Message Buffer n Interrupt Enable) (j = 0 to 7; n = m × 32 + j)

If the TMIEj bit is set to 1, an interrupt is generated when transmission from the transmit message buffer n is completed successfully.

Refer to section 45.11, Interrupts and DTC/DMA Requests for the specification of the interrupt for transmit message buffer.

This bit cannot be written in GL_SLEEP mode.

Do not write to the TMIEj bit when:

- The CAN channel is in CH_SLEEP mode
- The transmit message buffer n is part of a transmit queue
- The transmit message buffer n is linked to a common FIFO by the CFCRn.LTM[4:0] bits.

45.2.53 Transmit Queue n Configuration Register (TQCRn) (n = 0 to 3)

Address(es): CANFD0.TQCR0 8748 1000h, CANFD1.TQCR0 8748 1004h, CANFD0.TQCR1 8748 1060h, CANFD1.TQCR1 8748 1064h, CANFD0.TQCR2 8748 10C0h, CANFD1.TQCR2 8748 10C4h, CANFD0.TQCR3 8748 1120h, CANFD1.TQCR3 8748 1124h, CANFD0_NS.TQCR0 0748 1000h, CANFD1_NS.TQCR0 0748 1004h, CANFD0_NS.TQCR1 0748 1060h, CANFD1_NS.TQCR1 0748 1064h, CANFD0_NS.TQCR2 0748 10C0h, CANFD1_NS.TQCR2 0748 10C4h, CANFD0_NS.TQCR3 0748 1120h, CANFD1_NS.TQCR3 0748 1124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TQOTIE	TQORIE	TQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	QDS[4:0]					TQIM	—	TQIE	—	—	—	TQGWE	TQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQE	Transmit Queue Enable	0: Transmit queue is disabled. 1: Transmit queue is enabled.	R/W
b1	TQGWE	Transmit Queue Gateway Mode Enable*1	0: Transmit queue gateway mode is disabled. 1: Transmit queue gateway mode is enabled.	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	TQIE	Transmit Queue Interrupt Enable	0: Transmit queue transmission interrupt is disabled. 1: Transmit queue transmission interrupt is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TQIM	Transmit Queue Interrupt Mode Setting	0: When the last message is successfully transmitted 1: At every successful transmission	R/W
b12 to b8	QDS[4:0]	Queue Depth Setting	b12 b8 0 0 0 0 0: 0 messages (disabled) 0 0 0 0 1: Setting prohibited 0 0 0 1 0: 3 messages 0 0 0 1 1: 4 messages : 0 0 1 1 1: 8 messages	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TQFIE	Transmit Queue Full Interrupt Enable*1	0: Transmit queue full interrupt is disabled. 1: Transmit queue full interrupt is enabled.	R/W
b17	TQORIE	Transmit Queue One Frame Receive Interrupt Enable*1	0: One frame receive interrupt is disabled. 1: One frame receive interrupt is enabled.	R/W
b18	TQOTIE	Transmit Queue One Frame Transmit Interrupt Enable	0: One frame transmit interrupt is disabled. 1: One frame transmit interrupt is enabled.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is reserved in the TQCR3 register. This bit is read as 0. The write value should be 0.

This register is used to configure the transmit queue n.

A transmit queue is configured using transmit message buffers. A transmit message buffer used for a transmit queue cannot be used as a normal transmit message buffer.

Table 45.6 shows the transmit message buffers that can be used for each transmit queue and the configuration for each queue depth setting.

Table 45.6 Transmit Message Buffers Available for Each Transmit Queue and the Configuration for Each Queue Depth Setting

Transmit Queue	Configuration of Message Buffers	Queue Depth Setting and Transmit Message Buffer to be used
Transmit queue 0	Transmit message buffer 0 to transmit message buffer 7	3 messages (transmit message buffer 0 to 2) 4 messages (transmit message buffer 0 to 3) : 8 messages (transmit message buffer 0 to 7)
Transmit queue 1	Transmit message buffer 7 to transmit message buffer 0	3 messages (transmit message buffer 7 to 5) 4 messages (transmit message buffer 7 to 4) : 8 messages (transmit message buffer 7 to 0)
Transmit queue 2	Transmit message buffer 32 to transmit message buffer 39	3 messages (transmit message buffer 32 to 35) 4 messages (transmit message buffer 32 to 36) : 8 messages (transmit message buffer 32 to 39)
Transmit queue 3	Transmit message buffer 39 to transmit message buffer 32	3 messages (transmit message buffer 39 to 37) 4 messages (transmit message buffer 39 to 36) : 8 messages (transmit message buffer 39 to 32)

Note: When using transmit queue 0 and transmit queue 1 at the same time, and when using transmit queue 2 and transmit queue 3 at the same time, be sure not to overlap the transmit message buffers to be used.

TQE Bit (Transmit Queue Enable)

The TQE bit enables the transmit queue. When the TQE bit is set to 1, transmit message buffers are used to configure the transmit queue.

The TQE bit cannot be set to 1 if the configured transmit queue depth is 0 (TQCRn.QDS[4:0] = 00000b).

This bit cannot be written in GL_SLEEP mode.

Also, this bit cannot be written in CH_RESET or CH_SLEEP mode.

The TQE bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

TQGWE Bit (Transmit Queue Gateway Mode Enable)

When the TQGWE bit is set to 1, the transmit queue is used as transmit queue in the gateway mode.

When the TQGWE bit is set to 1, do not access the transmit queue from the CPU.

This bit cannot be written in GL_SLEEP mode. Also, this bit cannot be written in CH_HALT or CH_OPERATION mode.

Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQIE Bit (Transmit Queue Interrupt Enable)

When the TQIE bit is set to 1, an interrupt is generated based on the setting of the TQIM bit.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQIM Bit (Transmit Queue Interrupt Mode Setting)

The TQIM bit selects the interrupt generation condition for the transmit queue.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP, CH_HALT, or CH_OPERATION mode.

QDS[4:0] Bits (Queue Depth Setting)

The QDS[4:0] bits select the depth of the transmit queue.

For the relationship between the queue depth setting and the transmit message buffers to be used, refer to Table 45.6.

These bits cannot be written in GL_SLEEP mode.

Also, these bits cannot be written in CH_HALT or CH_OPERATION mode.
Do not write to these bits when the CAN channel is in CH_SLEEP mode.

TQFIE Bit (Transmit Queue Full Interrupt Enable)

When the TQFIE bit is set to 1, an interrupt occurs according to the setting of the TQSRn.TQFIF flag.
Set this bit to 1 only when the transmit queue is in gateway mode (TQCRn.TQGWE bit = 1).
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQORIE Bit (Transmit Queue One Frame Receive Interrupt Enable)

When the TQORIE bit is set to 1, an interrupt occurs according to the setting of the TQSRn.TQORIF flag.
Set this bit to 1 only when the transmit queue is in gateway mode (TQCRn.TQGWE bit = 1).
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQOTIE Bit (Transmit Queue One Frame Transmit Interrupt Enable)

When the TQOTIE bit is set to 1, an interrupt occurs according to the setting of the TQSRn.TQOTIF flag.
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit when the CAN channel is in CH_SLEEP mode.

45.2.54 Transmit Queue n Status Register (TQSRn) (n = 0 to 3)

Address(es): CANFD0.TQSR0 8748 1020h, CANFD1.TQSR0 8748 1024h, CANFD0.TQSR1 8748 1080h, CANFD1.TQSR1 8748 1084h, CANFD0.TQSR2 8748 10E0h, CANFD1.TQSR2 8748 10E4h, CANFD0.TQSR3 8748 1140h, CANFD1.TQSR3 8748 1144h, CANFD0_NS.TQSR0 0748 1020h, CANFD1_NS.TQSR0 0748 1024h, CANFD0_NS.TQSR1 0748 1080h, CANFD1_NS.TQSR1 0748 1084h, CANFD0_NS.TQSR2 0748 10E0h, CANFD1_NS.TQSR2 0748 10E4h, CANFD0_NS.TQSR3 0748 1140h, CANFD1_NS.TQSR3 0748 1144h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	LOST	TQOTIF	TQORIF	TQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	—	—	TQIF	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmit Queue Empty Flag	0: There is at least one message in transmit queue 1: No message in transmit queue (empty)	R
b1	FULL	Transmit Queue Full Flag	0: Transmit queue is not full 1: Transmit queue is full	R
b2	TQIF	Transmit Queue Interrupt Flag	0: Transmit queue interrupt condition is not satisfied 1: Transmit queue interrupt condition is satisfied	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Transmit Queue Fill Level	Indicates the number of messages in the transmit queue	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TQFIF	Transmit Queue Full Interrupt Flag *2	0: Transmit queue full interrupt condition is not satisfied 1: Transmit queue full interrupt condition is satisfied	R/(W) *1
b17	TQORIF	Transmit Queue One Frame Receive Interrupt Flag*2	0: Transmit queue one frame receive interrupt condition is not satisfied 1: Transmit queue one frame receive interrupt condition is satisfied	R/(W) *1
b18	TQOTIF	Transmit Queue One Frame Transmit Interrupt Flag	0: Transmit queue one frame transmit interrupt condition is not satisfied 1: Transmit queue one frame transmit interrupt condition is satisfied	R/(W) *1
b19	LOST	Transmit Queue Message Lost Flag *2	0: Message lost has not occurred in the transmit queue 1: Message lost has occurred in the transmit queue	R/(W) *1
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.
Note 2. This bit is reserved in the TQSR3 register. This bit is read as 0. The write value should be 0.

This register shows the status of the transmit queue n.

EMPTY Flag (Transmit Queue Empty Flag)

This flag is automatically set to 1 when:

- The TQCRn.TQE bit is set to 0 (transmit queue is disabled)
- No messages are stored in the transmit queue
- The last message is transmitted from the transmit queue
- The CAN channel enters CH_RESET mode.

This flag is automatically set to 0 when the first message to be transmitted is stored in the transmit queue.

FULL Flag (Transmit Queue Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the transmit queue matches the configured transmit queue depth.
This flag is automatically set to 0 when:

- The number of messages stored in the transmit queue is less than the configured transmit queue depth
- The CAN channel enters CH_RESET mode.

TQIF Flag (Transmit Queue Interrupt Flag)

The TQIF flag is not cleared automatically if the transmit queue is disabled.
When stopping the transmit queue, this flag should be cleared, after setting the TQCRn.TQE bit to 0 and checking an empty state of transmit queue.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 1 when the configured interrupt condition is satisfied for the transmit queue.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag cannot be written in CH_SLEEP or CH_RESET mode.

FLVL[5:0] Bits (Transmit Queue Fill Level)

The FLVL[5:0] bits indicates the number of messages in the transmit queue.
These bits are automatically set to 000000b when the CAN channel enters CH_RESET mode.

TQFIF Flag (Transmit Queue Full Interrupt Flag)

The TQFIF flag is automatically set to 1 when the transmit queue is set to gateway mode (TQCRn.TQGWE bit = 1) and the transmit queue is full.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
This flag is not automatically cleared if the transmit queue is disabled (TQCRn.TQE bit = 0).
When stopping the transmit queue, this flag should be cleared, after setting the TQE bit to 0 and checking an empty state of transmit queue.
This flag cannot be written in CH_SLEEP or CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

TQORIF Flag (Transmit Queue One Frame Receive Interrupt Flag)

The TQORIF flag is automatically set to 1 when the transmit queue is set to gateway mode (TQCRn.TQGWE bit = 1) and received data is transferred to the transmit queue.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
This flag is not automatically cleared if the transmit queue is disabled (TQCRn.TQE bit = 0).
When stopping the transmit queue, this flag should be cleared, after setting the TQE bit to 0 and checking an empty state of transmit queue.
This flag cannot be written in CH_SLEEP or CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

TQOTIF Flag (Transmit Queue One Frame Transmit Interrupt Flag)

The TQOTIF flag is automatically set to 1 when message is successfully transmitted from the transmit queue.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
This flag is not cleared automatically if the transmit queue is disabled (TQCRn.TQE bit = 0).
When stopping the transmit queue, this flag should be cleared, after setting the TQE bit to 0 and checking an empty state of transmit queue.
This flag cannot be written in CH_SLEEP or CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

LOST Flag (Transmit Queue Message Lost Flag)

The LOST flag is automatically set to 1 when the transmit queue is set to gateway mode (TQCRn.TQGWE bit = 1) and message lost has occurred in the transmit queue.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
This flag is not automatically cleared if the transmit queue is disabled (TQCRn.TQE bit = 0).
When stopping the transmit queue, this flag should be cleared, after setting the TQE bit to 0 and checking an empty state of transmit queue.
This flag cannot be written in CH_SLEEP or CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

45.2.55 Transmit Queue n Pointer Control Register (TQPCRn) (n = 0 to 3)

Address(es): CANFD0.TQPCR0 8748 1040h, CANFD1.TQPCR0 8748 1044h, CANFD0.TQPCR1 8748 10A0h, CANFD1.TQPCR1 8748 10A4h, CANFD0.TQPCR2 8748 1100h, CANFD1.TQPCR2 8748 1104h, CANFD0.TQPCR3 8748 1160h, CANFD1.TQPCR3 8748 1164h, CANFD0_NS.TQPCR0 0748 1040h, CANFD1_NS.TQPCR0 0748 1044h, CANFD0_NS.TQPCR1 0748 10A0h, CANFD1_NS.TQPCR1 0748 10A4h, CANFD0_NS.TQPCR2 0748 1100h, CANFD1_NS.TQPCR2 0748 1104h, CANFD0_NS.TQPCR3 0748 1160h, CANFD1_NS.TQPCR3 0748 1164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the write pointer to the transmit queue.
When the value 000000FFh is written to this register, the write pointer to the transmit queue is updated and transmission of the existing message is requested.
The read value from this register is always 00000000h.
This register cannot be written in CH_SLEEP or CH_RESET mode.
Write to this register only when:

- The transmit queue is enabled and not full.
- When DTC/DMA transfer is not used to write to the transmit queue
- When the transmit queue is not in gateway mode

45.2.56 Transmit Queue Empty Status Register (TQESR)

Address(es): CANFDG0.TQESR 8748 1180h, CANFDG0_NS.TQESR 0748 1180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TQEMP13	TQEMP12	TQEMP11	TQEMP10	TQEMP03	TQEMP02	TQEMP01	TQEMP00
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	TQEMP00	Channel 0 Transmit Queue 0 Empty Flag	0: There is at least one message in transmit queue 1: No message in transmit queue (empty)	R
b1	TQEMP01	Channel 0 Transmit Queue 1 Empty Flag		R
b2	TQEMP02	Channel 0 Transmit Queue 2 Empty Flag		R
b3	TQEMP03	Channel 0 Transmit Queue 3 Empty Flag		R
b4	TQEMP10	Channel 1 Transmit Queue 0 Empty Flag		R
b5	TQEMP11	Channel 1 Transmit Queue 1 Empty Flag		R
b6	TQEMP12	Channel 1 Transmit Queue 2 Empty Flag		R
b7	TQEMP13	Channel 1 Transmit Queue 3 Empty Flag		R
b31 to b8	—	Reserved	These bits are read as 0.	R

This register shows the empty status of each transmit queue.

TQEMPin Flag (Channel i Transmit Queue n Empty Flag) (i = 0, 1; n = 0 to 3)

The TQEMPin flag shows the state of the EMPTY flag for the corresponding transmit queue.

This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.

This flag is set to 1 when the CANFDG module enters GL_RESET mode

45.2.57 Transmit Queue Full Interrupt Status Register (TQFISR)

Address(es): CANFDG0.TQFISR 8748 1184h, CANFDG0_NS.TQFISR 0748 1184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TQFIF12	TQFIF11	TQFIF10	—	TQFIF02	TQFIF01	TQFIF00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQFIF00	Channel 0 Transmit Queue 0 Full Interrupt Flag	0: Transmit queue full interrupt condition is not satisfied 1: Transmit queue full interrupt condition is satisfied	R
b1	TQFIF01	Channel 0 Transmit Queue 1 Full Interrupt Flag		R
b2	TQFIF02	Channel 0 Transmit Queue 2 Full Interrupt Flag		R
b3	—	Reserved	These bits are read as 0.	R
b4	TQFIF10	Channel 1 Transmit Queue 0 Full Interrupt Flag	0: Transmit queue full interrupt condition is not satisfied 1: Transmit queue full interrupt condition is satisfied	R
b5	TQFIF11	Channel 1 Transmit Queue 1 Full Interrupt Flag		R
b6	TQFIF12	Channel 1 Transmit Queue 2 Full Interrupt Flag		R
b31 to b7	—	Reserved	These bits are read as 0.	R

This register shows the state of the transmit queue full interrupt flag for each transmit queue.

TQFIFin Flag (Channel i Transmit Queue n Full Interrupt Flag) (i = 0, 1; n = 0 to 2)

The TQFIFin flag shows the state of the TQFIF flag of the corresponding transmit queue.

This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.

This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.58 Transmit Queue Message Lost Status Register (TQMLSR)

Address(es): CANFDG0.TQMLSR 8748 1188h, CANFDG0_NS.TQMLSR 0748 1188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TQML12	TQML11	TQML10	—	TQML02	TQML01	TQML00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQML00	Channel 0 Transmit Queue 0 Message Lost Interrupt Flag	0: Message lost has not occurred in the transmit queue 1: Message lost has occurred in the transmit queue	R
b1	TQML01	Channel 0 Transmit Queue 1 Message Lost Interrupt Flag		R
b2	TQML02	Channel 0 Transmit Queue 2 Message Lost Interrupt Flag		R
b3	—	Reserved	These bits are read as 0.	R
b4	TQML10	Channel 1 Transmit Queue 0 Message Lost Interrupt Flag	0: Message lost has not occurred in the transmit queue 1: Message lost has occurred in the transmit queue	R
b5	TQML11	Channel 1 Transmit Queue 1 Message Lost Interrupt Flag		R
b6	TQML12	Channel 1 Transmit Queue 2 Message Lost Interrupt Flag		R
b31 to b7	—	Reserved	These bits are read as 0.	R

This register shows the state of the transmit queue message lost flag for each transmit queue.

TQMLin Flag (Channel i Transmit Queue n Message Lost Flag) (i = 0, 1; n = 0 to 2)

The TQMLin flag shows the state of the LOST flag of the corresponding transmit queue.

This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.

This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.59 Transmit Queue Interrupt Status Register (TQISR)

Address(es): CANFDG0.TQISR 8748 1190h, CANFDG0_NS.TQISR 0748 1190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TQIF13	TQIF12	TQIF11	TQIF10	TQIF03	TQIF02	TQIF01	TQIF00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQIF00	Channel 0 Transmit Queue 0 Interrupt Flag	0: Transmit queue interrupt condition is not satisfied 1: Transmit queue interrupt condition is satisfied	R
b1	TQIF01	Channel 0 Transmit Queue 1 Interrupt Flag		R
b2	TQIF02	Channel 0 Transmit Queue 2 Interrupt Flag		R
b3	TQIF03	Channel 0 Transmit Queue 3 Interrupt Flag		R
b4	TQIF10	Channel 1 Transmit Queue 0 Interrupt Flag		R
b5	TQIF11	Channel 1 Transmit Queue 1 Interrupt Flag		R
b6	TQIF12	Channel 1 Transmit Queue 2 Interrupt Flag		R
b7	TQIF13	Channel 1 Transmit Queue 3 Interrupt Flag		R
b31 to b8	—	Reserved	These bits are read as 0.	R

This register shows the interrupt status of each transmit queue.

TQIFin Flag (Channel i Transmit Queue n Interrupt Flag) (i = 0, 1; n = 0 to 3)

The TQIFin flag shows the state of the TQIF flag for the corresponding transmit queue.

This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.

This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.60 Transmit Queue One Frame Transmit Interrupt Status Register (TQOTISR)

Address(es): CANFDG0.TQOTISR 8748 1194h, CANFDG0_NS.TQOTISR 0748 1194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TQOTIF13	TQOTIF12	TQOTIF11	TQOTIF10	TQOTIF03	TQOTIF02	TQOTIF01	TQOTIF00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQOTIF00	Channel 0 Transmit Queue 0 One Frame Transmit Interrupt Flag	0: Transmit queue one frame transmit interrupt condition is not satisfied 1: Transmit queue one frame transmit interrupt condition is satisfied	R
b1	TQOTIF01	Channel 0 Transmit Queue 1 One Frame Transmit Interrupt Flag		R
b2	TQOTIF02	Channel 0 Transmit Queue 2 One Frame Transmit Interrupt Flag		R
b3	TQOTIF03	Channel 0 Transmit Queue 3 One Frame Transmit Interrupt Flag		R
b4	TQOTIF10	Channel 1 Transmit Queue 0 One Frame Transmit Interrupt Flag		R
b5	TQOTIF11	Channel 1 Transmit Queue 1 One Frame Transmit Interrupt Flag		R
b6	TQOTIF12	Channel 1 Transmit Queue 2 One Frame Transmit Interrupt Flag		R
b7	TQOTIF13	Channel 1 Transmit Queue 3 One Frame Transmit Interrupt Flag		R
b31 to b8	—	Reserved	These bits are read as 0.	R

This register shows the one frame transmit interrupt status of each transmit queue.

TQOTIFin Flag (Channel i Transmit Queue n One Frame Transmit Interrupt Flag) (i = 0, 1; n = 0 to 3)

The TQOTIFin flag shows the state of the TQOTIF flag for the corresponding transmit queue.

This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.

This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.61 Transmit Queue One Frame Receive Interrupt Status Register (TQORISR)

Address(es): CANFDG0.TQORISR 8748 1198h, CANFDG0_NS.TQORISR 0748 1198h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TQORIF12	TQORIF11	TQORIF10	—	TQORIF02	TQORIF01	TQORIF00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQORIF00	Channel 0 Transmit Queue 0 One Frame Receive Interrupt Flag	0: Transmit queue one frame receive interrupt condition is not satisfied 1: Transmit queue one frame receive interrupt condition is satisfied	R
b1	TQORIF01	Channel 0 Transmit Queue 1 One Frame Receive Interrupt Flag		R
b2	TQORIF02	Channel 0 Transmit Queue 2 One Frame Receive Interrupt Flag		R
b3	—	Reserved	These bits are read as 0.	R
b4	TQORIF10	Channel 1 Transmit Queue 0 One Frame Receive Interrupt Flag	0: Transmit queue one frame receive interrupt condition is not satisfied 1: Transmit queue one frame receive interrupt condition is satisfied	R
b5	TQORIF11	Channel 1 Transmit Queue 1 One Frame Receive Interrupt Flag		R
b6	TQORIF12	Channel 1 Transmit Queue 2 One Frame Receive Interrupt Flag		R
b31 to b7	—	Reserved	These bits are read as 0.	R

This register shows the one frame receive interrupt status of each transmit queue.

TQORIFin Flag (Channel i Transmit Queue n Full Interrupt Flag) (i = 0, 1; n = 0 to 2)

The TQORIFin flag shows the state of the TQORIF flag of the corresponding transmit queue.
This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.
This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.62 Transmit Queue Full Status Register (TQFSR)

Address(es): CANFDG0.TQFSR 8748 119Ch, CANFDG0_NS.TQFSR 0748 119Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TQFUL13	TQFUL12	TQFUL11	TQFUL10	TQFUL03	TQFUL02	TQFUL01	TQFUL00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQFUL00	Channel 0 Transmit Queue 0 Full Flag	0: Transmit queue is not full 1: Transmit queue is full	R
b1	TQFUL01	Channel 0 Transmit Queue 1 Full Flag		R
b2	TQFUL02	Channel 0 Transmit Queue 2 Full Flag		R
b3	TQFUL03	Channel 0 Transmit Queue 3 Full Flag		R
b4	TQFUL10	Channel 1 Transmit Queue 0 Full Flag		R
b5	TQFUL11	Channel 1 Transmit Queue 1 Full Flag		R
b6	TQFUL12	Channel 1 Transmit Queue 2 Full Flag		R
b7	TQFUL13	Channel 1 Transmit Queue 3 Full Flag		R
b31 to b8	—	Reserved	These bits are read as 0.	R

This register shows the state of the full flag for each transmit queue.

TQFULin Flag (Channel i Transmit Queue n Full Flag) (i = 0, 1; n = 0 to 3)

The TQFULin flag shows the state of the FULL flag for the corresponding transmit queue.
This flag is automatically set to 1 when the corresponding flag becomes 1, and automatically set to 0 when the corresponding flag becomes 0.
This flag is set to 0 when the CANFDG module enters GL_RESET mode

45.2.63 Transmission History Configuration Register (THCR)

Address(es): CANFD0.THCR 8748 1200h, CANFD1.THCR 8748 1204h,
CANFD0_NS.THCR 0748 1200h, CANFD1_NS.THCR 0748 1204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	THRGC	THRC	THIM	THIE	—	—	—	—	—	—	—	THE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	THE	Transmission History Enable	0: Transmission history buffer is disabled 1: Transmission history buffer is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	THIE	Transmission History Interrupt Enable	0: Transmission history interrupt is disabled 1: Transmission history interrupt is enabled	R/W
b9	THIM	Transmission History Interrupt Mode Setting	0: Interrupt is generated if transmission history level reaches 3/4 of the transmission history depth 1: Interrupt is generated for every successfully stored entry	R/W
b10	THRC	Transmission History Recording Condition Setting	0: Transmit FIFO + Transmit queue 1: Transmit message buffer + Transmit FIFO + Transmit queue	R/W
b11	THRGC	Transmission History of Gateway FIFO Recording Condition Setting	0: Recording the transmission history of gateway FIFO is disabled 1: Recording the transmission history of gateway FIFO is enabled	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmission history functions.

THE Bit (Transmission History Enable)

The THE bit enables the transmission history buffer when it is set to 1.
This bit cannot be written in CH_RESET or CH_SLEEP mode.
This bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

THIE Bit (Transmission History Interrupt Enable)

The THIE bit enables the generation of the transmission history interrupt when it is set to 1.
This bit cannot be written in GL_SLEEP mode.

THIM Bit (Transmission History Interrupt Mode Setting)

The THIM bit selects the transmission history interrupt generation condition.
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit in GL_HALT or GL_OPERATION mode.

THRC Bit (Transmission History Recording Condition Setting)

The THRC bit selects the condition for storing an entry in the transmission history buffer after successful transmission.
This bit cannot be written in GL_SLEEP mode.

Do not write to this bit in GL_HALT or GL_OPERATION mode.

THRGC Bit (Transmission History of Gateway FIFO Recording Condition Setting)

The THRGC bit selects whether to store in the transmission history buffer after successful transmission in the gateway FIFO.
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit in GL_HALT or GL_OPERATION mode.

45.2.64 Transmission History Status Register (THSR)

Address(es): CANFD0.THSR 8748 1220h, CANFD1.THSR 8748 1224h,
CANFD0_NS.THSR 0748 1220h, CANFD1_NS.THSR 0748 1224h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	—	THIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmission History Empty Flag	0: There is a transmission history in the transmission history buffer 1: There is no transmission history in the transmission history buffer (empty)	R
b1	FULL	Transmission History Full Flag	0: Transmission history buffer is not full 1: Transmission history buffer is full	R
b2	LOST	Transmission History Lost Flag	0: Transmission history has not been lost 1: Transmission history has been lost	R/(W) *1
b3	THIF	Transmission History Interrupt Flag	0: Transmission history interrupt condition is not satisfied 1: Transmission history interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Transmission History Fill Level	Number of transmission histories stored in transmission history buffer	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of data stored in the transmission history buffer.

EMPTY Flag (Transmission History Empty Flag)

This flag is automatically set to 0 when the first transmission history is stored to the transmission history buffer.
This flag is automatically set to 1 when:

- The CPU has read all the transmission histories from the transmission history buffer
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel enters CH_RESET mode.

FULL Flag (Transmission History Full Flag)

Up to 32 transmission histories can be stored in the transmission history buffer for each channel. The FULL flag is automatically set to 1 when the number of transmission histories in the transmission history buffer is 32.
This flag is automatically set to 0 when:

- The number of transmission histories in the transmission history buffer is less than 32
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel enters CH_RESET mode.

LOST Flag (Transmission History Lost Flag)

The LOST flag is set to 1 when a new transmission history cannot be stored because the related transmission history

buffer is already full.
This flag can only be written in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

THIF Flag (Transmission History Interrupt Flag)

The THIF flag is set to 1 when the configured interrupt condition is satisfied.
This flag can only be written in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

FLVL[5:0] Bits (Transmission History Fill Level)

The FLVL[5:0] bits show the number of transmission histories stored in the transmission history buffer.
These bits are automatically set to 000000b when the CAN channel enters CH_RESET mode.

45.2.65 Transmission History Access Register 0 (THACR0)

Address(es): CANFD0.THACR0 8748 8000h, CANFD1.THACR0 8748 8008h,
CANFD0_NS.THACR0 0748 8000h, CANFD1_NS.THACR0 0748 8008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GWMF	—	—	—	—	—	BN[6:0]							BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BT[2:0]	Transmitted Buffer Type	b2 b0 0 0 1: Transmit message buffer 0 1 0: Common FIFO 1 0 0: Transmit Queue	R
b9 to b3	BN[6:0]	Transmitted Buffer Number	Number of the message buffer	R
b14 to b10	—	Reserved	These bits are read as 0.	R
b15	GWMF	Gateway Mode Transfer Flag	0: Transmit from other than gateway 1: Transmit through a gateway	R
b31 to b16	TS[15:0]	Transmitted Timestamp	Transmit timestamp value	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

BT[2:0] Bits (Transmitted Buffer Type)

The BT[2:0] bits indicate which type of buffer the message was transmitted from.

BN[6:0] Bits (Transmitted Buffer Number)

The BN[6:0] bits indicate which buffer number the message was transmitted from.
For a common FIFO, these bits indicate the number of the linked transmit message buffer.

GWMF Flag (Gateway Mode Transfer Flag)

The GWMF flag indicates whether the message was transferred through a gateway.

TS[15:0] Bits (Transmitted Timestamp)

The TS[15:0] bits indicate the timestamp used by the software driver.

45.2.66 Transmission History Access Register 1 (THACR1)

Address(es): CANFD0.THACR1 8748 8004h, CANFD1.THACR1 8748 800Ch,
CANFD0_NS.THACR1 0748 8004h, CANFD1_NS.THACR1 0748 800Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PTR[15:0]	Transmitted Pointer	Stores the value of the PTR[15:0] field attached to the transmit message	R
b17, b16	IFL[1:0]	Transmitted Information Label	Stores the value of the IFL[1:0] field attached to the transmit message	R
b31 to b18	—	Reserved	These bits are read as 0.	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

PTR[15:0] Bits (Transmitted Pointer)

The PTR[15:0] bits store the value of the pointer field (TMBn.HF2.PTR[15:0] bits or CFBn.HF2.PTR[15:0] bits) attached to the transmit message.
When transmitting in gateway mode, the value of the AFL pointer field (AFLn.PTR0.PTR[15:0] bits) is stored instead of the TMBn.HF2.PTR[15:0] bits.

IFL[1:0] Bits (Transmitted Information Label)

The IFL[1:0] bits store the value of the information label field (TMBn.HF2.IFL[1:0] bits or CFBn.HF2.IFL[1:0] bits) attached to the transmit message.
When transmitting in gateway mode, the value of the AFL information label field (AFLn.MASK.IFL1 bit and AFLn.PTR0.IFL0 bit) is stored instead of the TMBn.HF2.IFL[1:0] bits.

45.2.67 Transmission History Pointer Control Register (THPCR)

Address(es): CANFD0.THPCR 8748 1240h, CANFD1.THPCR 8748 1244h,
CANFD0_NS.THPCR 0748 1240h, CANFD1_NS.THPCR 0748 1244h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the read pointer of the transmission history buffer.
When 000000FFh is written to this register, the read pointer of the transmission history buffer is moved to the next history.
The read value from this register is 00000000h.
This register can only be written in CH_HALT or CH_OPERATION mode.
Write 000000FFh to this register only when the transmission history buffer is enabled and not empty.

45.2.68 Global Reset Control Register (GRCR)

Address(es): CANFDG0.GRCR 8748 1380h, CANFDG0_NS.GRCR 0748 1380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SRST	Software Reset	0: Releases the software reset 1: Initiates the software reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the SRST bit. These bits are read as 00h.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SRST Bit (Software Reset)

When the SRST bit is set to 1, the CANFDG module is in the same state as the MCU reset. When a reset is required, write 1 to this bit and then 0.

When the software reset is released, the CANFDG module is in GL_SLEEP mode.

After a software reset, the RAM initialization sequence is not performed. Initialize the RAM with software.

Similarly, if a software reset is performed during RAM initialization, the RAM is not initialized. Initialize the RAM with software.

KEY[7:0] Bits (Key Code)

When rewriting the value of the SRST bit, set these bits to C4h at the same time (write this register in 32 bits).

45.2.69 Global Test Mode Configuration Register (GTMCR)

Address(es): CANFDG0.GTMCR 8748 1308h, CANFDG0_NS.GTMCR 0748 1308h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	RTPS[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICBTMC1	ICBTMC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICBTMC0	Internal CAN Bus Communication Test Mode Channel 0	0: Channel 0 is not tested in the internal CAN bus communication test 1: Channel 0 is tested in the internal CAN bus communication test	R/W
b1	ICBTMC1	Internal CAN Bus Communication Test Mode Channel 1	0: Channel 1 is not tested in the internal CAN bus communication test 1: Channel 1 is tested in the internal CAN bus communication test	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	RTPS[9:0]	RAM Test Page Select	Select the page of RAM to be tested	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to specify the page number of the RAM to be tested in RAM test mode. This register is also used to configure the CAN channel to be tested in the internal CAN bus communication test mode.

ICBTMCi Bit (Channel i Internal CAN Bus Communication Test Mode Enable) (i = 0, 1)

When the ICBTMCi bit is 1 and the CANFDG module is set to the internal CAN bus communication test mode, CAN channel i is tested in the internal CAN bus communication test mode.
This bit cannot be written in GL_RESET or GL_SLEEP mode. Write to this bit only in GL_HALT mode.
This bit is automatically set to 0 when the CAN channel enters GL_RESET mode.

RTPS[9:0] Bits (RAM Test Page Select)

The RTPS[9:0] bits select the RAM page number that the CPU reads and writes when the CANFDG module is in RAM test mode. Specify the page number in the range of 0 to 39.
Refer to section 45.9.2.1, RAM Test Mode for the specifications of the RAM test mode.
These bits cannot be written in GL_RESET or GL_SLEEP mode. Write to these bits only in GL_HALT mode.
These bits are automatically set to 00h when the CAN channel enters GL_RESET mode.

45.2.70 Global Test Mode Enable Register (GTMER)

Address(es): CANFDG0.GTMER 8748 130Ch, CANFDG0_NS.GTMER 0748 130Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBTME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICBTME	Internal CAN Bus Communication Test Mode Enable	0: Internal CAN bus communication test mode is disabled. 1: Internal CAN bus communication test mode is enabled.	R/W
b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RTME	RAM Test Mode Enable	0: RAM Test mode is disabled. 1: RAM Test mode is enabled.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the global test modes of the CANFDG module.

ICBTME Bit (Internal CAN Bus Communication Test Mode Enable)

Setting the ICBTME bit to 1 enables internal CAN bus communication for CAN channels that have the GTMCR.ICBTMCi bit set to 1.

Refer to section 45.9.2.2, Internal CAN Bus Communication Test Mode for the specifications of the internal CAN bus communication test mode.

This bit can only be set to 1 in GL_HALT mode. To exit the internal CAN bus communication test mode, set this bit to 0 in GL_HALT mode.

This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode.

RTME Bit (RAM Test Mode Enable)

When the RTME bit is set to 1, the CANFDG module is in RAM test mode. Refer to section 45.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

This bit can only be set to 1 in GL_HALT mode. To exit the RAM test mode, set this bit to 0 in GL_HALT mode.

This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode.

45.2.71 Global CAN FD Configuration Register (GFDCFG)

Address(es): CANFDG0.GFDCFG 8748 1314h, CANFDG0_NS.GFDCFG 0748 1314h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TSCPS[1:0]		—	—	—	—	—	—	—	PXEDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PXEDIS	Protocol Exception Event Detection Disable	0: Protocol exception event detection is enabled. 1: Protocol exception event detection is disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	TSCPS[1:0]	Timestamp Capturing Point Select	b9 b8 0 0: Sample point of SOF (start of frame) 0 1: EOF (end of frame) when frame is taken valid 1 0: Sample point of SOF (Classical CAN frame), or sample point of res bit following the FDF bit (CAN FD frame) 1 1: Setting prohibited	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PXEDIS Bit (Protocol Exception Event Detection Disable)

The PXEDIS bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is set to 1, protocol exception event detection is disabled, and an error frame is transmitted if a protocol exception (a recessive reserved bit following the FDF bit) is detected.

This bit can only be written in GL_RESET mode.

TSCPS[1:0] Bits (Timestamp Capturing Point Select)

The TSCPS[1:0] bits selects the capture point of the timestamps for transmitted and received messages.

When the TSCPS[1:0] bits are 10b, the timestamp is captured at the reserved bit following the FDF bit in CAN FD frames and at the SOF in Classical CAN frames.

These bits can only be written in GL_RESET mode.

45.2.72 Global Test Mode Lock Key Register (GTMLKR)

Address(es): CANFDG0.GTMLKR 8748 131Ch, CANFDG0_NS.GTMLKR 0748 131Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to unlock the protection for RAM test mode. Refer to section 45.9.2, Global Test Modes for the specification of the lock key.
To put the CANFDG module into RAM test mode, two unlock keys must be written to this register in consecutive bus cycles.
The read value from this register is always 00000000h.
This register cannot be written in GL_SLEEP or GL_RESET mode.
Do not write to this register in GL_OPERATION mode.

45.2.73 RAM Test Page Access Register k (RTPARk) (k = 0 to 63)

Address(es): CANFDG0.RTPAR0 8748 8400h to CANFDG0.RTPAR63 8748 84FCh,
CANFDG0_NS.RTPAR0 0748 8400h to CANFDG0_NS.RTPAR63 0748 84FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be read or written in RAM test mode.
This register can only be written in GL_HALT mode when RAM test mode is enabled.

45.2.74 Bus Load Control Register (BLCCR)

Address(es): CANFD0.BLCCR 8748 1418h, CANFD1.BLCCR 8748 1438h,
CANFD0_NS.BLCCR 0748 1418h, CANFD1_NS.BLCCR 0748 1438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BLCLD	—	—	—	—	—	—	—	BLCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BLCE	Bus Load Counter Enable	0: Bus load counter is disabled 1: Bus load counter is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BLCLD	Bus Load Counter Load	When this bit is set to 1, the bus load counter value is loaded and the counter is reset.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

BLCE Bit (Bus Load Counter Enable)

The BLCE bit enables the operation of the bus load counter. When the BLCE bit is set to 0, the bus load counter stops, but the counter value is not cleared.

This bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

This bit cannot be written in CH_SLEEP mode.

Write to this register after setting the NBCR register.

BLCLD Bit (Bus Load Counter Load)

When the BLCLD bit is set to 1, the bus load counter value is loaded into the BLCR register and then the bus load counter is reset.

The read value is always 0.

45.2.75 Bus Load Counter Register (BLCR)

Address(es): CANFD0.BLCR 8748 141Ch, CANFD1.BLCR 8748 143Ch,
CANFD0_NS.BLCR 0748 141Ch, CANFD1_NS.BLCR 0748 143Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is a read-only register that indicates the value of the bus load counter.
The bus load counter is incremented synchronously with CANFDCLK when the CAN bus is idle.
When the BLCCR.BLCLD bit is set to 1, the bus load counter value is loaded to this register and then the bus load counter is reset.
The lower 3 bits of the BLCR register are 000b.

45.2.76 Acceptance Filter List Ignore Entry Setting Register (AFIGSR)

Address(es): CANFDG0.AFIGSR 8748 1324h, CANFDG0_NS.AFIGSR 0748 1324h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	IGCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IGES[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	IGES[8:0]	Ignore Entry Select	Set the rule number to be ignored during acceptance filtering.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	IGCS[2:0]	Ignore Channel Select	Set the channel number for which entries are to be ignored during acceptance filtering.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGES[8:0] Bits (Ignore Entry Select)

The IGES[8:0] bits are used to specify the rule number to update when updating the acceptance filter.

Write to these bits only when the AFIGER.IGEE bit is 0.

These bits cannot be written in GL_SLEEP mode.

IGCS[2:0] Bits (Ignore Channel Select)

The IGCS[2:0] bits are used to specify the channel number for which the AFL entry is updated when the acceptance filter is updated.

Write to these bits only when the AFIGER.IGEE bit is 0.

These bits cannot be written in GL_SLEEP mode.

45.2.77 Acceptance Filter List Ignore Entry Enable Register (AFIGER)

Address(es): CANFDG0.AFIGER 8748 1328h, CANFDG0_NS.AFIGER 0748 1328h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	IGEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGEE	Ignore Entry Enable	0: The value of the AFIGSR.IGES[8:0] bits is invalid. 1: The value of the AFIGSR.IGES[8:0] bits is valid.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the IGEE bit. These bits are read as 00h.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGEE Bit (Ignore Entry Enable)

When the IGEE bit is set to 1, the entry selected by the AFIGSR.IGES[8:0] bits is ignored during acceptance filtering. This bit is automatically set to 0 when the CANFDG module enters GL_RESET mode.

KEY[7:0] Bits (Key Code)

When rewriting the value of the IGEE bit, set these bits to C4h at the same time (write this register in 32 bits).

45.2.78 Identifier Bits Alignment

Base format (11-bit length identifier): ID-28 to ID-18 are located in b10 to b0, and b28 to b11 are 0.
Extended format (29-bit length identifier): ID-28 to ID-0 are located in b28 to b0

Table 45.7 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18

Table 45.8 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18	ID-17	ID-16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0

45.2.79 Message Buffer Structure

The message buffer configuration consists of four types of message buffers:

- Receive message buffer (RMBn)
- Receive FIFO (RFBn)
- Common FIFO (CFBn)
- Transmit message buffer (TMBn)

"n" is a message buffer number whose range changes depending on the type of message buffer.
Refer to Figure 45.33 for an overview of this configuration. For more information on the number and types of message buffers, refer to section 45.6, FIFO Buffers and Message Buffer Configuration.

45.2.79.1 Start Addresses

The start address of each message buffer is calculated using the number of related message buffer components.
The start addresses for each register in the message buffer are listed in Table 45.9.

Table 45.9 Start Address for Each Register of the Message Buffer Component

Message Buffer	Symbol	n	Register	p	Start Address (Only Secure Addresses Are Listed)
Receive Message Buffer	RMBn	0 to 31	HF0	—	8748 2000h + n × 80h
			HF1	—	8748 2004h + n × 80h
			HF2	—	8748 2008h + n × 80h
			DFp	0 to 15	8748 200Ch + n × 80h + p × 4
Receive FIFO	RFBn	0 to 7	HF0	—	8748 6000h + n × 80h
			HF1	—	8748 6004h + n × 80h
			HF2	—	8748 6008h + n × 80h
			DFp	0 to 15	8748 600Ch + n × 80h + p × 4
Common FIFO	CFBn	0 to 5	HF0	—	8748 6400h + n × 80h
			HF1	—	8748 6404h + n × 80h
			HF2	—	8748 6408h + n × 80h
			DFp	0 to 15	8748 640Ch + n × 80h + p × 4
Transmit Message Buffer	TMBn	0 to 7, 32 to 39, 64 to 71, 96 to 103	HF0	—	8749 0000h + n × 80h
			HF1	—	8749 0004h + n × 80h
			HF2	—	8749 0008h + n × 80h
			DFp	0 to 15	8749 000Ch + n × 80h + p × 4

45.2.79.2 Receive Message Buffer n (RMBn) (n = 0 to 31)

The total number of the receive message buffers (RMB) is 32, as shown in Figure 45.33.
The receive message buffer consists of the following registers:

- RMBn.HF0
- RMBn.HF1
- RMBn.HF2
- RMBn.DF0 to RMBn.DF15

Table 45.10 shows the structure of this buffer.

Table 45.10 Structure of Receive Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	RMBn.HF0	Receive Message Buffer n Header Field 0	IDE, RTR, ID
+04h	RMBn.HF1	Receive Message Buffer n Header Field 1	DLC, timestamp
+08h	RMBn.HF2	Receive Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RMBn.DF0	Receive Message Buffer n Data Field 0	DATA0 to DATA3
+10h	RMBn.DF1	Receive Message Buffer n Data Field 1	DATA4 to DATA7
+14h	RMBn.DF2	Receive Message Buffer n Data Field 2	DATA8 to DATA11
+18h	RMBn.DF3	Receive Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	RMBn.DF4	Receive Message Buffer n Data Field 4	DATA16 to DATA19
+20h	RMBn.DF5	Receive Message Buffer n Data Field 5	DATA20 to DATA23
+24h	RMBn.DF6	Receive Message Buffer n Data Field 6	DATA24 to DATA27
+28h	RMBn.DF7	Receive Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	RMBn.DF8	Receive Message Buffer n Data Field 8	DATA32 to DATA35
+30h	RMBn.DF9	Receive Message Buffer n Data Field 9	DATA36 to DATA39
+34h	RMBn.DF10	Receive Message Buffer n Data Field 10	DATA40 to DATA43
+38h	RMBn.DF11	Receive Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	RMBn.DF12	Receive Message Buffer n Data Field 12	DATA48 to DATA51
+40h	RMBn.DF13	Receive Message Buffer n Data Field 13	DATA52 to DATA55
+44h	RMBn.DF14	Receive Message Buffer n Data Field 14	DATA56 to DATA59
+48h	RMBn.DF15	Receive Message Buffer n Data Field 15	DATA60 to DATA63

45.2.79.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 31)

Address(es): CANFDG0.RMB0.HF0 8748 2000h, CANFDG0.RMB1.HF0 8748 2080h, CANFDG0.RMB2.HF0 8748 2100h, CANFDG0.RMB3.HF0 8748 2180h, CANFDG0.RMB4.HF0 8748 2200h, CANFDG0.RMB5.HF0 8748 2280h, CANFDG0.RMB6.HF0 8748 2300h, CANFDG0.RMB7.HF0 8748 2380h, CANFDG0.RMB8.HF0 8748 2400h, CANFDG0.RMB9.HF0 8748 2480h, CANFDG0.RMB10.HF0 8748 2500h, CANFDG0.RMB11.HF0 8748 2580h, CANFDG0.RMB12.HF0 8748 2600h, CANFDG0.RMB13.HF0 8748 2680h, CANFDG0.RMB14.HF0 8748 2700h, CANFDG0.RMB15.HF0 8748 2780h, CANFDG0.RMB16.HF0 8748 2800h, CANFDG0.RMB17.HF0 8748 2880h, CANFDG0.RMB18.HF0 8748 2900h, CANFDG0.RMB19.HF0 8748 2980h, CANFDG0.RMB20.HF0 8748 2A00h, CANFDG0.RMB21.HF0 8748 2A80h, CANFDG0.RMB22.HF0 8748 2B00h, CANFDG0.RMB23.HF0 8748 2B80h, CANFDG0.RMB24.HF0 8748 2C00h, CANFDG0.RMB25.HF0 8748 2C80h, CANFDG0.RMB26.HF0 8748 2D00h, CANFDG0.RMB27.HF0 8748 2D80h, CANFDG0.RMB28.HF0 8748 2E00h, CANFDG0.RMB29.HF0 8748 2E80h, CANFDG0.RMB30.HF0 8748 2F00h, CANFDG0.RMB31.HF0 8748 2F80h,
CANFDG0_NS.RMB0.HF0 0748 2000h, CANFDG0_NS.RMB1.HF0 0748 2080h, CANFDG0_NS.RMB2.HF0 0748 2100h, CANFDG0_NS.RMB3.HF0 0748 2180h, CANFDG0_NS.RMB4.HF0 0748 2200h, CANFDG0_NS.RMB5.HF0 0748 2280h, CANFDG0_NS.RMB6.HF0 0748 2300h, CANFDG0_NS.RMB7.HF0 0748 2380h, CANFDG0_NS.RMB8.HF0 0748 2400h, CANFDG0_NS.RMB9.HF0 0748 2480h, CANFDG0_NS.RMB10.HF0 0748 2500h, CANFDG0_NS.RMB11.HF0 0748 2580h, CANFDG0_NS.RMB12.HF0 0748 2600h, CANFDG0_NS.RMB13.HF0 0748 2680h, CANFDG0_NS.RMB14.HF0 0748 2700h, CANFDG0_NS.RMB15.HF0 0748 2780h, CANFDG0_NS.RMB16.HF0 0748 2800h, CANFDG0_NS.RMB17.HF0 0748 2880h, CANFDG0_NS.RMB18.HF0 0748 2900h, CANFDG0_NS.RMB19.HF0 0748 2980h, CANFDG0_NS.RMB20.HF0 0748 2A00h, CANFDG0_NS.RMB21.HF0 0748 2A80h, CANFDG0_NS.RMB22.HF0 0748 2B00h, CANFDG0_NS.RMB23.HF0 0748 2B80h, CANFDG0_NS.RMB24.HF0 0748 2C00h, CANFDG0_NS.RMB25.HF0 0748 2C80h, CANFDG0_NS.RMB26.HF0 0748 2D00h, CANFDG0_NS.RMB27.HF0 0748 2D80h, CANFDG0_NS.RMB28.HF0 0748 2E00h, CANFDG0_NS.RMB29.HF0 0748 2E80h, CANFDG0_NS.RMB30.HF0 0748 2F00h, CANFDG0_NS.RMB31.HF0 0748 2F80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0.	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive message buffer.
For alignment of these bits in base and extended format, refer to section 45.2.78, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.
The RTR bit indicates whether the receive message buffer stores a data frame or a remote frame.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.
The IDE bit indicates whether the message stored in the receive message buffer has a standard identifier or extended identifier.

45.2.79.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 31)

Address(es): CANFDG0.RMB0.HF1 8748 2004h, CANFDG0.RMB1.HF1 8748 2084h, CANFDG0.RMB2.HF1 8748 2104h, CANFDG0.RMB3.HF1 8748 2184h, CANFDG0.RMB4.HF1 8748 2204h, CANFDG0.RMB5.HF1 8748 2284h, CANFDG0.RMB6.HF1 8748 2304h, CANFDG0.RMB7.HF1 8748 2384h, CANFDG0.RMB8.HF1 8748 2404h, CANFDG0.RMB9.HF1 8748 2484h, CANFDG0.RMB10.HF1 8748 2504h, CANFDG0.RMB11.HF1 8748 2584h, CANFDG0.RMB12.HF1 8748 2604h, CANFDG0.RMB13.HF1 8748 2684h, CANFDG0.RMB14.HF1 8748 2704h, CANFDG0.RMB15.HF1 8748 2784h, CANFDG0.RMB16.HF1 8748 2804h, CANFDG0.RMB17.HF1 8748 2884h, CANFDG0.RMB18.HF1 8748 2904h, CANFDG0.RMB19.HF1 8748 2984h, CANFDG0.RMB20.HF1 8748 2A04h, CANFDG0.RMB21.HF1 8748 2A84h, CANFDG0.RMB22.HF1 8748 2B04h, CANFDG0.RMB23.HF1 8748 2B84h, CANFDG0.RMB24.HF1 8748 2C04h, CANFDG0.RMB25.HF1 8748 2C84h, CANFDG0.RMB26.HF1 8748 2D04h, CANFDG0.RMB27.HF1 8748 2D84h, CANFDG0.RMB28.HF1 8748 2E04h, CANFDG0.RMB29.HF1 8748 2E84h, CANFDG0.RMB30.HF1 8748 2F04h, CANFDG0.RMB31.HF1 8748 2F84h,
CANFDG0_NS.RMB0.HF1 0748 2004h, CANFDG0_NS.RMB1.HF1 0748 2084h, CANFDG0_NS.RMB2.HF1 0748 2104h, CANFDG0_NS.RMB3.HF1 0748 2184h, CANFDG0_NS.RMB4.HF1 0748 2204h, CANFDG0_NS.RMB5.HF1 0748 2284h, CANFDG0_NS.RMB6.HF1 0748 2304h, CANFDG0_NS.RMB7.HF1 0748 2384h, CANFDG0_NS.RMB8.HF1 0748 2404h, CANFDG0_NS.RMB9.HF1 0748 2484h, CANFDG0_NS.RMB10.HF1 0748 2504h, CANFDG0_NS.RMB11.HF1 0748 2584h, CANFDG0_NS.RMB12.HF1 0748 2604h, CANFDG0_NS.RMB13.HF1 0748 2684h, CANFDG0_NS.RMB14.HF1 0748 2704h, CANFDG0_NS.RMB15.HF1 0748 2784h, CANFDG0_NS.RMB16.HF1 0748 2804h, CANFDG0_NS.RMB17.HF1 0748 2884h, CANFDG0_NS.RMB18.HF1 0748 2904h, CANFDG0_NS.RMB19.HF1 0748 2984h, CANFDG0_NS.RMB20.HF1 0748 2A04h, CANFDG0_NS.RMB21.HF1 0748 2A84h, CANFDG0_NS.RMB22.HF1 0748 2B04h, CANFDG0_NS.RMB23.HF1 0748 2B84h, CANFDG0_NS.RMB24.HF1 0748 2C04h, CANFDG0_NS.RMB25.HF1 0748 2C84h, CANFDG0_NS.RMB26.HF1 0748 2D04h, CANFDG0_NS.RMB27.HF1 0748 2D84h, CANFDG0_NS.RMB28.HF1 0748 2E04h, CANFDG0_NS.RMB29.HF1 0748 2E84h, CANFDG0_NS.RMB30.HF1 0748 2F04h, CANFDG0_NS.RMB31.HF1 0748 2F84h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value stored for the message in the receive message buffer	R
b27 to b16	—	Reserved	These bits are read as 0.	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame.	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RMCR.PLS[2:0] bits.

45.2.79.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 31)

Address(es): CANFDG0.RMB0.HF2 8748 2008h, CANFDG0.RMB1.HF2 8748 2088h, CANFDG0.RMB2.HF2 8748 2108h, CANFDG0.RMB3.HF2 8748 2188h, CANFDG0.RMB4.HF2 8748 2208h, CANFDG0.RMB5.HF2 8748 2288h, CANFDG0.RMB6.HF2 8748 2308h, CANFDG0.RMB7.HF2 8748 2388h, CANFDG0.RMB8.HF2 8748 2408h, CANFDG0.RMB9.HF2 8748 2488h, CANFDG0.RMB10.HF2 8748 2508h, CANFDG0.RMB11.HF2 8748 2588h, CANFDG0.RMB12.HF2 8748 2608h, CANFDG0.RMB13.HF2 8748 2688h, CANFDG0.RMB14.HF2 8748 2708h, CANFDG0.RMB15.HF2 8748 2788h, CANFDG0.RMB16.HF2 8748 2808h, CANFDG0.RMB17.HF2 8748 2888h, CANFDG0.RMB18.HF2 8748 2908h, CANFDG0.RMB19.HF2 8748 2988h, CANFDG0.RMB20.HF2 8748 2A08h, CANFDG0.RMB21.HF2 8748 2A88h, CANFDG0.RMB22.HF2 8748 2B08h, CANFDG0.RMB23.HF2 8748 2B88h, CANFDG0.RMB24.HF2 8748 2C08h, CANFDG0.RMB25.HF2 8748 2C88h, CANFDG0.RMB26.HF2 8748 2D08h, CANFDG0.RMB27.HF2 8748 2D88h, CANFDG0.RMB28.HF2 8748 2E08h, CANFDG0.RMB29.HF2 8748 2E88h, CANFDG0.RMB30.HF2 8748 2F08h, CANFDG0.RMB31.HF2 8748 2F88h,
CANFDG0_NS.RMB0.HF2 0748 2008h, CANFDG0_NS.RMB1.HF2 0748 2088h, CANFDG0_NS.RMB2.HF2 0748 2108h, CANFDG0_NS.RMB3.HF2 0748 2188h, CANFDG0_NS.RMB4.HF2 0748 2208h, CANFDG0_NS.RMB5.HF2 0748 2288h, CANFDG0_NS.RMB6.HF2 0748 2308h, CANFDG0_NS.RMB7.HF2 0748 2388h, CANFDG0_NS.RMB8.HF2 0748 2408h, CANFDG0_NS.RMB9.HF2 0748 2488h, CANFDG0_NS.RMB10.HF2 0748 2508h, CANFDG0_NS.RMB11.HF2 0748 2588h, CANFDG0_NS.RMB12.HF2 0748 2608h, CANFDG0_NS.RMB13.HF2 0748 2688h, CANFDG0_NS.RMB14.HF2 0748 2708h, CANFDG0_NS.RMB15.HF2 0748 2788h, CANFDG0_NS.RMB16.HF2 0748 2808h, CANFDG0_NS.RMB17.HF2 0748 2888h, CANFDG0_NS.RMB18.HF2 0748 2908h, CANFDG0_NS.RMB19.HF2 0748 2988h, CANFDG0_NS.RMB20.HF2 0748 2A08h, CANFDG0_NS.RMB21.HF2 0748 2A88h, CANFDG0_NS.RMB22.HF2 0748 2B08h, CANFDG0_NS.RMB23.HF2 0748 2B88h, CANFDG0_NS.RMB24.HF2 0748 2C08h, CANFDG0_NS.RMB25.HF2 0748 2C88h, CANFDG0_NS.RMB26.HF2 0748 2D08h, CANFDG0_NS.RMB27.HF2 0748 2D88h, CANFDG0_NS.RMB28.HF2 0748 2E08h, CANFDG0_NS.RMB29.HF2 0748 2E88h, CANFDG0_NS.RMB30.HF2 0748 2F08h, CANFDG0_NS.RMB31.HF2 0748 2F88h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0.	R
b9, b8	IFL[1:0]	Information Label	This field stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b31 to b16	PTR[15:0]	Pointer	This field stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

45.2.79.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 31; p = 0 to 15)

Address(es): CANFDG0.RMB0.DF0 8748 200Ch to CANFDG0.RMB0.DF15 8748 2048h,
CANFDG0.RMB1.DF0 8748 208Ch to CANFDG0.RMB1.DF15 8748 20C8h,
CANFDG0.RMB2.DF0 8748 210Ch to CANFDG0.RMB2.DF15 8748 2148h,
CANFDG0.RMB3.DF0 8748 218Ch to CANFDG0.RMB3.DF15 8748 21C8h,
CANFDG0.RMB4.DF0 8748 220Ch to CANFDG0.RMB4.DF15 8748 2248h,
CANFDG0.RMB5.DF0 8748 228Ch to CANFDG0.RMB5.DF15 8748 22C8h,
CANFDG0.RMB6.DF0 8748 230Ch to CANFDG0.RMB6.DF15 8748 2348h,
CANFDG0.RMB7.DF0 8748 238Ch to CANFDG0.RMB7.DF15 8748 23C8h,
CANFDG0.RMB8.DF0 8748 240Ch to CANFDG0.RMB8.DF15 8748 2448h,
CANFDG0.RMB9.DF0 8748 248Ch to CANFDG0.RMB9.DF15 8748 24C8h,
CANFDG0.RMB10.DF0 8748 250Ch to CANFDG0.RMB10.DF15 8748 2548h,
CANFDG0.RMB11.DF0 8748 258Ch to CANFDG0.RMB11.DF15 8748 25C8h,
CANFDG0.RMB12.DF0 8748 260Ch to CANFDG0.RMB12.DF15 8748 2648h,
CANFDG0.RMB13.DF0 8748 268Ch to CANFDG0.RMB13.DF15 8748 26C8h,
CANFDG0.RMB14.DF0 8748 270Ch to CANFDG0.RMB14.DF15 8748 2748h,
CANFDG0.RMB15.DF0 8748 278Ch to CANFDG0.RMB15.DF15 8748 27C8h,
CANFDG0.RMB16.DF0 8748 280Ch to CANFDG0.RMB16.DF15 8748 2848h,
CANFDG0.RMB17.DF0 8748 288Ch to CANFDG0.RMB17.DF15 8748 28C8h,
CANFDG0.RMB18.DF0 8748 290Ch to CANFDG0.RMB18.DF15 8748 2948h,
CANFDG0.RMB19.DF0 8748 298Ch to CANFDG0.RMB19.DF15 8748 29C8h,
CANFDG0.RMB20.DF0 8748 2A0Ch to CANFDG0.RMB20.DF15 8748 2A48h,
CANFDG0.RMB21.DF0 8748 2A8Ch to CANFDG0.RMB21.DF15 8748 2AC8h,
CANFDG0.RMB22.DF0 8748 2B0Ch to CANFDG0.RMB22.DF15 8748 2B48h,
CANFDG0.RMB23.DF0 8748 2B8Ch to CANFDG0.RMB23.DF15 8748 2BC8h,
CANFDG0.RMB24.DF0 8748 2C0Ch to CANFDG0.RMB24.DF15 8748 2C48h,
CANFDG0.RMB25.DF0 8748 2C8Ch to CANFDG0.RMB25.DF15 8748 2CC8h,
CANFDG0.RMB26.DF0 8748 2D0Ch to CANFDG0.RMB26.DF15 8748 2D48h,
CANFDG0.RMB27.DF0 8748 2D8Ch to CANFDG0.RMB27.DF15 8748 2DC8h,
CANFDG0.RMB28.DF0 8748 2E0Ch to CANFDG0.RMB28.DF15 8748 2E48h,
CANFDG0.RMB29.DF0 8748 2E8Ch to CANFDG0.RMB29.DF15 8748 2EC8h,
CANFDG0.RMB30.DF0 8748 2F0Ch to CANFDG0.RMB30.DF15 8748 2F48h,
CANFDG0.RMB31.DF0 8748 2F8Ch to CANFDG0.RMB31.DF15 8748 2FC8h,
CANFDG0_NS.RMB0.DF0 0748 200Ch to CANFDG0_NS.RMB0.DF15 0748 2048h,
CANFDG0_NS.RMB1.DF0 0748 208Ch to CANFDG0_NS.RMB1.DF15 0748 20C8h,
CANFDG0_NS.RMB2.DF0 0748 210Ch to CANFDG0_NS.RMB2.DF15 0748 2148h,
CANFDG0_NS.RMB3.DF0 0748 218Ch to CANFDG0_NS.RMB3.DF15 0748 21C8h,
CANFDG0_NS.RMB4.DF0 0748 220Ch to CANFDG0_NS.RMB4.DF15 0748 2248h,
CANFDG0_NS.RMB5.DF0 0748 228Ch to CANFDG0_NS.RMB5.DF15 0748 22C8h,
CANFDG0_NS.RMB6.DF0 0748 230Ch to CANFDG0_NS.RMB6.DF15 0748 2348h,
CANFDG0_NS.RMB7.DF0 0748 238Ch to CANFDG0_NS.RMB7.DF15 0748 23C8h,
CANFDG0_NS.RMB8.DF0 0748 240Ch to CANFDG0_NS.RMB8.DF15 0748 2448h,
CANFDG0_NS.RMB9.DF0 0748 248Ch to CANFDG0_NS.RMB9.DF15 0748 24C8h,
CANFDG0_NS.RMB10.DF0 0748 250Ch to CANFDG0_NS.RMB10.DF15 0748 2548h,
CANFDG0_NS.RMB11.DF0 0748 258Ch to CANFDG0_NS.RMB11.DF15 0748 25C8h,
CANFDG0_NS.RMB12.DF0 0748 260Ch to CANFDG0_NS.RMB12.DF15 0748 2648h,
CANFDG0_NS.RMB13.DF0 0748 268Ch to CANFDG0_NS.RMB13.DF15 0748 26C8h,
CANFDG0_NS.RMB14.DF0 0748 270Ch to CANFDG0_NS.RMB14.DF15 0748 2748h,
CANFDG0_NS.RMB15.DF0 0748 278Ch to CANFDG0_NS.RMB15.DF15 0748 27C8h,
CANFDG0_NS.RMB16.DF0 0748 280Ch to CANFDG0_NS.RMB16.DF15 0748 2848h,
CANFDG0_NS.RMB17.DF0 0748 288Ch to CANFDG0_NS.RMB17.DF15 0748 28C8h,
CANFDG0_NS.RMB18.DF0 0748 290Ch to CANFDG0_NS.RMB18.DF15 0748 2948h,
CANFDG0_NS.RMB19.DF0 0748 298Ch to CANFDG0_NS.RMB19.DF15 0748 29C8h,
CANFDG0_NS.RMB20.DF0 0748 2A0Ch to CANFDG0_NS.RMB20.DF15 0748 2A48h,
CANFDG0_NS.RMB21.DF0 0748 2A8Ch to CANFDG0_NS.RMB21.DF15 0748 2AC8h,
CANFDG0_NS.RMB22.DF0 0748 2B0Ch to CANFDG0_NS.RMB22.DF15 0748 2B48h,
CANFDG0_NS.RMB23.DF0 0748 2B8Ch to CANFDG0_NS.RMB23.DF15 0748 2BC8h,
CANFDG0_NS.RMB24.DF0 0748 2C0Ch to CANFDG0_NS.RMB24.DF15 0748 2C48h,
CANFDG0_NS.RMB25.DF0 0748 2C8Ch to CANFDG0_NS.RMB25.DF15 0748 2CC8h,
CANFDG0_NS.RMB26.DF0 0748 2D0Ch to CANFDG0_NS.RMB26.DF15 0748 2D48h,
CANFDG0_NS.RMB27.DF0 0748 2D8Ch to CANFDG0_NS.RMB27.DF15 0748 2DC8h,
CANFDG0_NS.RMB28.DF0 0748 2E0Ch to CANFDG0_NS.RMB28.DF15 0748 2E48h,
CANFDG0_NS.RMB29.DF0 0748 2E8Ch to CANFDG0_NS.RMB29.DF15 0748 2EC8h,
CANFDG0_NS.RMB30.DF0 0748 2F0Ch to CANFDG0_NS.RMB30.DF15 0748 2F48h,
CANFDG0_NS.RMB31.DF0 0748 2F8Ch to CANFDG0_NS.RMB31.DF15 0748 2FC8h

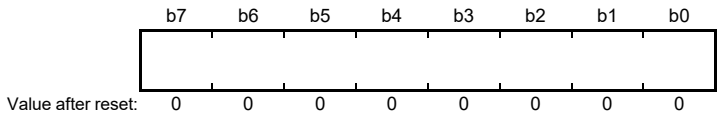
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

45.2.79.7 Receive Message Buffer n Data k (RMBn.DATAk) (n = 0 to 31; k = 0 to 63)

Address(es): CANFDG0.RMB0.DATA0 8748 200Ch to CANFDG0.RMB0.DATA63 8748 204Bh,
CANFDG0.RMB1.DATA0 8748 208Ch to CANFDG0.RMB1.DATA63 8748 20CBh,
CANFDG0.RMB2.DATA0 8748 210Ch to CANFDG0.RMB2.DATA63 8748 214Bh,
CANFDG0.RMB3.DATA0 8748 218Ch to CANFDG0.RMB3.DATA63 8748 21CBh,
CANFDG0.RMB4.DATA0 8748 220Ch to CANFDG0.RMB4.DATA63 8748 224Bh,
CANFDG0.RMB5.DATA0 8748 228Ch to CANFDG0.RMB5.DATA63 8748 22CBh,
CANFDG0.RMB6.DATA0 8748 230Ch to CANFDG0.RMB6.DATA63 8748 234Bh,
CANFDG0.RMB7.DATA0 8748 238Ch to CANFDG0.RMB7.DATA63 8748 23CBh,
CANFDG0.RMB8.DATA0 8748 240Ch to CANFDG0.RMB8.DATA63 8748 244Bh,
CANFDG0.RMB9.DATA0 8748 248Ch to CANFDG0.RMB9.DATA63 8748 24CBh,
CANFDG0.RMB10.DATA0 8748 250Ch to CANFDG0.RMB10.DATA63 8748 254Bh,
CANFDG0.RMB11.DATA0 8748 258Ch to CANFDG0.RMB11.DATA63 8748 25CBh,
CANFDG0.RMB12.DATA0 8748 260Ch to CANFDG0.RMB12.DATA63 8748 264Bh,
CANFDG0.RMB13.DATA0 8748 268Ch to CANFDG0.RMB13.DATA63 8748 26CBh,
CANFDG0.RMB14.DATA0 8748 270Ch to CANFDG0.RMB14.DATA63 8748 274Bh,
CANFDG0.RMB15.DATA0 8748 278Ch to CANFDG0.RMB15.DATA63 8748 27CBh,
CANFDG0.RMB16.DATA0 8748 280Ch to CANFDG0.RMB16.DATA63 8748 284Bh,
CANFDG0.RMB17.DATA0 8748 288Ch to CANFDG0.RMB17.DATA63 8748 28CBh,
CANFDG0.RMB18.DATA0 8748 290Ch to CANFDG0.RMB18.DATA63 8748 294Bh,
CANFDG0.RMB19.DATA0 8748 298Ch to CANFDG0.RMB19.DATA63 8748 29CBh,
CANFDG0.RMB20.DATA0 8748 2A0Ch to CANFDG0.RMB20.DATA63 8748 2A4Bh,
CANFDG0.RMB21.DATA0 8748 2A8Ch to CANFDG0.RMB21.DATA63 8748 2ACBh,
CANFDG0.RMB22.DATA0 8748 2B0Ch to CANFDG0.RMB22.DATA63 8748 2B4Bh,
CANFDG0.RMB23.DATA0 8748 2B8Ch to CANFDG0.RMB23.DATA63 8748 2BCBh,
CANFDG0.RMB24.DATA0 8748 2C0Ch to CANFDG0.RMB24.DATA63 8748 2C4Bh,
CANFDG0.RMB25.DATA0 8748 2C8Ch to CANFDG0.RMB25.DATA63 8748 2CCBh,
CANFDG0.RMB26.DATA0 8748 2D0Ch to CANFDG0.RMB26.DATA63 8748 2D4Bh,
CANFDG0.RMB27.DATA0 8748 2D8Ch to CANFDG0.RMB27.DATA63 8748 2DCBh,
CANFDG0.RMB28.DATA0 8748 2E0Ch to CANFDG0.RMB28.DATA63 8748 2E4Bh,
CANFDG0.RMB29.DATA0 8748 2E8Ch to CANFDG0.RMB29.DATA63 8748 2ECBh,
CANFDG0.RMB30.DATA0 8748 2F0Ch to CANFDG0.RMB30.DATA63 8748 2F4Bh,
CANFDG0.RMB31.DATA0 8748 2F8Ch to CANFDG0.RMB31.DATA63 8748 2FCBh,
CANFDG0_NS.RMB0.DATA0 0748 200Ch to CANFDG0_NS.RMB0.DATA63 0748 204Bh,
CANFDG0_NS.RMB1.DATA0 0748 208Ch to CANFDG0_NS.RMB1.DATA63 0748 20CBh,
CANFDG0_NS.RMB2.DATA0 0748 210Ch to CANFDG0_NS.RMB2.DATA63 0748 214Bh,
CANFDG0_NS.RMB3.DATA0 0748 218Ch to CANFDG0_NS.RMB3.DATA63 0748 21CBh,
CANFDG0_NS.RMB4.DATA0 0748 220Ch to CANFDG0_NS.RMB4.DATA63 0748 224Bh,
CANFDG0_NS.RMB5.DATA0 0748 228Ch to CANFDG0_NS.RMB5.DATA63 0748 22CBh,
CANFDG0_NS.RMB6.DATA0 0748 230Ch to CANFDG0_NS.RMB6.DATA63 0748 234Bh,
CANFDG0_NS.RMB7.DATA0 0748 238Ch to CANFDG0_NS.RMB7.DATA63 0748 23CBh,
CANFDG0_NS.RMB8.DATA0 0748 240Ch to CANFDG0_NS.RMB8.DATA63 0748 244Bh,
CANFDG0_NS.RMB9.DATA0 0748 248Ch to CANFDG0_NS.RMB9.DATA63 0748 24CBh,
CANFDG0_NS.RMB10.DATA0 0748 250Ch to CANFDG0_NS.RMB10.DATA63 0748 254Bh,
CANFDG0_NS.RMB11.DATA0 0748 258Ch to CANFDG0_NS.RMB11.DATA63 0748 25CBh,
CANFDG0_NS.RMB12.DATA0 0748 260Ch to CANFDG0_NS.RMB12.DATA63 0748 264Bh,
CANFDG0_NS.RMB13.DATA0 0748 268Ch to CANFDG0_NS.RMB13.DATA63 0748 26CBh,
CANFDG0_NS.RMB14.DATA0 0748 270Ch to CANFDG0_NS.RMB14.DATA63 0748 274Bh,
CANFDG0_NS.RMB15.DATA0 0748 278Ch to CANFDG0_NS.RMB15.DATA63 0748 27CBh,
CANFDG0_NS.RMB16.DATA0 0748 280Ch to CANFDG0_NS.RMB16.DATA63 0748 284Bh,
CANFDG0_NS.RMB17.DATA0 0748 288Ch to CANFDG0_NS.RMB17.DATA63 0748 28CBh,
CANFDG0_NS.RMB18.DATA0 0748 290Ch to CANFDG0_NS.RMB18.DATA63 0748 294Bh,
CANFDG0_NS.RMB19.DATA0 0748 298Ch to CANFDG0_NS.RMB19.DATA63 0748 29CBh,
CANFDG0_NS.RMB20.DATA0 0748 2A0Ch to CANFDG0_NS.RMB20.DATA63 0748 2A4Bh,
CANFDG0_NS.RMB21.DATA0 0748 2A8Ch to CANFDG0_NS.RMB21.DATA63 0748 2ACBh,
CANFDG0_NS.RMB22.DATA0 0748 2B0Ch to CANFDG0_NS.RMB22.DATA63 0748 2B4Bh,
CANFDG0_NS.RMB23.DATA0 0748 2B8Ch to CANFDG0_NS.RMB23.DATA63 0748 2BCBh,
CANFDG0_NS.RMB24.DATA0 0748 2C0Ch to CANFDG0_NS.RMB24.DATA63 0748 2C4Bh,
CANFDG0_NS.RMB25.DATA0 0748 2C8Ch to CANFDG0_NS.RMB25.DATA63 0748 2CCBh,
CANFDG0_NS.RMB26.DATA0 0748 2D0Ch to CANFDG0_NS.RMB26.DATA63 0748 2D4Bh,
CANFDG0_NS.RMB27.DATA0 0748 2D8Ch to CANFDG0_NS.RMB27.DATA63 0748 2DCBh,
CANFDG0_NS.RMB28.DATA0 0748 2E0Ch to CANFDG0_NS.RMB28.DATA63 0748 2E4Bh,
CANFDG0_NS.RMB29.DATA0 0748 2E8Ch to CANFDG0_NS.RMB29.DATA63 0748 2ECBh,
CANFDG0_NS.RMB30.DATA0 0748 2F0Ch to CANFDG0_NS.RMB30.DATA63 0748 2F4Bh,
CANFDG0_NS.RMB31.DATA0 0748 2F8Ch to CANFDG0_NS.RMB31.DATA63 0748 2FCBh



These registers are read-only registers that store the data bytes of the received message. Unused data bytes are filled with 00h.

45.2.79.8 Receive FIFO n (RFBn) (n = 0 to 7)

The total number of the receive FIFOs (RFB) is eight, as shown in Figure 45.33.
The receive FIFO consists of the following registers:

- RFBn.HF0
- RFBn.HF1
- RFBn.HF2
- RFBn.DF0 to RFBn.DF15

Table 45.11 shows the structure of this buffer.

Table 45.11 Structure of Receive FIFO

Address Offset	Symbol	Register Name	Contents
+00h	RFBn.HF0	Receive FIFO n Header Field 0	IDE, RTR, ID
+04h	RFBn.HF1	Receive FIFO n Header Field 1	DLC, timestamp
+08h	RFBn.HF2	Receive FIFO n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RFBn.DF0	Receive FIFO n Data Field 0	DATA0 to DATA3
+10h	RFBn.DF1	Receive FIFO n Data Field 1	DATA4 to DATA7
+14h	RFBn.DF2	Receive FIFO n Data Field 2	DATA8 to DATA11
+18h	RFBn.DF3	Receive FIFO n Data Field 3	DATA12 to DATA15
+1Ch	RFBn.DF4	Receive FIFO n Data Field 4	DATA16 to DATA19
+20h	RFBn.DF5	Receive FIFO n Data Field 5	DATA20 to DATA23
+24h	RFBn.DF6	Receive FIFO n Data Field 6	DATA24 to DATA27
+28h	RFBn.DF7	Receive FIFO n Data Field 7	DATA28 to DATA31
+2Ch	RFBn.DF8	Receive FIFO n Data Field 8	DATA32 to DATA35
+30h	RFBn.DF9	Receive FIFO n Data Field 9	DATA36 to DATA39
+34h	RFBn.DF10	Receive FIFO n Data Field 10	DATA40 to DATA43
+38h	RFBn.DF11	Receive FIFO n Data Field 11	DATA44 to DATA47
+3Ch	RFBn.DF12	Receive FIFO n Data Field 12	DATA48 to DATA51
+40h	RFBn.DF13	Receive FIFO n Data Field 13	DATA52 to DATA55
+44h	RFBn.DF14	Receive FIFO n Data Field 14	DATA56 to DATA59
+48h	RFBn.DF15	Receive FIFO n Data Field 15	DATA60 to DATA63

45.2.79.9 Receive FIFO n Header Field 0 (RFBn.HF0) (n = 0 to 7)

Address(es): CANFDG0.RFB0.HF0 8748 6000h, CANFDG0.RFB1.HF0 8748 6080h, CANFDG0.RFB2.HF0 8748 6100h, CANFDG0.RFB3.HF0 8748 6180h, CANFDG0.RFB4.HF0 8748 6200h, CANFDG0.RFB5.HF0 8748 6280h, CANFDG0.RFB6.HF0 8748 6300h, CANFDG0.RFB7.HF0 8748 6380h, CANFDG0_NS.RFB0.HF0 0748 6000h, CANFDG0_NS.RFB1.HF0 0748 6080h, CANFDG0_NS.RFB2.HF0 0748 6100h, CANFDG0_NS.RFB3.HF0 0748 6180h, CANFDG0_NS.RFB4.HF0 0748 6200h, CANFDG0_NS.RFB5.HF0 0748 6280h, CANFDG0_NS.RFB6.HF0 0748 6300h, CANFDG0_NS.RFB7.HF0 0748 6380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28-b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0.	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive FIFO.
For alignment of these bits in base and extended format, refer to section 45.2.78, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.
The RTR bit indicates whether the receive message buffer stores a data frame or a remote frame.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.
The IDE bit indicates whether the message stored in the receive FIFO buffer has a standard identifier or extended identifier.

45.2.79.10 Receive FIFO n Header Field 1 (RFBn.HF1) (n = 0 to 7)

Address(es): CANFDG0.RFB0.HF1 8748 6004h, CANFDG0.RFB1.HF1 8748 6084h, CANFDG0.RFB2.HF1 8748 6104h, CANFDG0.RFB3.HF1 8748 6184h, CANFDG0.RFB4.HF1 8748 6204h, CANFDG0.RFB5.HF1 8748 6284h, CANFDG0.RFB6.HF1 8748 6304h, CANFDG0.RFB7.HF1 8748 6384h, CANFDG0_NS.RFB0.HF1 0748 6004h, CANFDG0_NS.RFB1.HF1 0748 6084h, CANFDG0_NS.RFB2.HF1 0748 6104h, CANFDG0_NS.RFB3.HF1 0748 6184h, CANFDG0_NS.RFB4.HF1 0748 6204h, CANFDG0_NS.RFB5.HF1 0748 6284h, CANFDG0_NS.RFB6.HF1 0748 6304h, CANFDG0_NS.RFB7.HF1 0748 6384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame	R
b27 to b16	—	Reserved	These bits are read as 0.	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RFCRn.PLS[2:0] bits.

45.2.79.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0 to 7)

Address(es): CANFDG0.RFB0.HF2 8748 6008h, CANFDG0.RFB1.HF2 8748 6088h, CANFDG0.RFB2.HF2 8748 6108h, CANFDG0.RFB3.HF2 8748 6188h, CANFDG0.RFB4.HF2 8748 6208h, CANFDG0.RFB5.HF2 8748 6288h, CANFDG0.RFB6.HF2 8748 6308h, CANFDG0.RFB7.HF2 8748 6388h, CANFDG0_NS.RFB0.HF2 0748 6008h, CANFDG0_NS.RFB1.HF2 0748 6088h, CANFDG0_NS.RFB2.HF2 0748 6108h, CANFDG0_NS.RFB3.HF2 0748 6188h, CANFDG0_NS.RFB4.HF2 0748 6208h, CANFDG0_NS.RFB5.HF2 0748 6288h, CANFDG0_NS.RFB6.HF2 0748 6308h, CANFDG0_NS.RFB7.HF2 0748 6388h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0.	R
b9, b8	IFL[1:0]	Information Label	This field stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b31 to b16	PTR[15:0]	Pointer	This field stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

45.2.79.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0 to 7; p = 0 to 15)

Address(es): CANFDG0.RFB0.DF0 8748 600Ch to CANFDG0.RFB0.DF15 8748 6048h,
CANFDG0.RFB1.DF0 8748 608Ch to CANFDG0.RFB1.DF15 8748 60C8h,
CANFDG0.RFB2.DF0 8748 610Ch to CANFDG0.RFB2.DF15 8748 6148h,
CANFDG0.RFB3.DF0 8748 618Ch to CANFDG0.RFB3.DF15 8748 61C8h,
CANFDG0.RFB4.DF0 8748 620Ch to CANFDG0.RFB4.DF15 8748 6248h,
CANFDG0.RFB5.DF0 8748 628Ch to CANFDG0.RFB5.DF15 8748 62C8h,
CANFDG0.RFB6.DF0 8748 630Ch to CANFDG0.RFB6.DF15 8748 6348h,
CANFDG0.RFB7.DF0 8748 638Ch to CANFDG0.RFB7.DF15 8748 63C8h,
CANFDG0_NS.RFB0.DF0 0748 600Ch to CANFDG0_NS.RFB0.DF15 0748 6048h,
CANFDG0_NS.RFB1.DF0 0748 608Ch to CANFDG0_NS.RFB1.DF15 0748 60C8h,
CANFDG0_NS.RFB2.DF0 0748 610Ch to CANFDG0_NS.RFB2.DF15 0748 6148h,
CANFDG0_NS.RFB3.DF0 0748 618Ch to CANFDG0_NS.RFB3.DF15 0748 61C8h,
CANFDG0_NS.RFB4.DF0 0748 620Ch to CANFDG0_NS.RFB4.DF15 0748 6248h,
CANFDG0_NS.RFB5.DF0 0748 628Ch to CANFDG0_NS.RFB5.DF15 0748 62C8h,
CANFDG0_NS.RFB6.DF0 0748 630Ch to CANFDG0_NS.RFB6.DF15 0748 6348h,
CANFDG0_NS.RFB7.DF0 0748 638Ch to CANFDG0_NS.RFB7.DF15 0748 63C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

45.2.79.13 Receive FIFO n Data k (RFBn.DATAk) (n = 0 to 7; k = 0 to 63)

Address(es): CANFDG0.RFB0.DATA0 8748 600Ch to CANFDG0.RFB0.DATA63 8748 604Bh,
CANFDG0.RFB1.DATA0 8748 608Ch to CANFDG0.RFB1.DATA63 8748 60CBh,
CANFDG0.RFB2.DATA0 8748 610Ch to CANFDG0.RFB2.DATA63 8748 614Bh,
CANFDG0.RFB3.DATA0 8748 618Ch to CANFDG0.RFB3.DATA63 8748 61CBh,
CANFDG0.RFB4.DATA0 8748 620Ch to CANFDG0.RFB4.DATA63 8748 624Bh,
CANFDG0.RFB5.DATA0 8748 628Ch to CANFDG0.RFB5.DATA63 8748 62CBh,
CANFDG0.RFB6.DATA0 8748 630Ch to CANFDG0.RFB6.DATA63 8748 634Bh,
CANFDG0.RFB7.DATA0 8748 638Ch to CANFDG0.RFB7.DATA63 8748 63CBh,
CANFDG0_NS.RFB0.DATA0 0748 600Ch to CANFDG0_NS.RFB0.DATA63 0748 604Bh,
CANFDG0_NS.RFB1.DATA0 0748 608Ch to CANFDG0_NS.RFB1.DATA63 0748 60CBh,
CANFDG0_NS.RFB2.DATA0 0748 610Ch to CANFDG0_NS.RFB2.DATA63 0748 614Bh,
CANFDG0_NS.RFB3.DATA0 0748 618Ch to CANFDG0_NS.RFB3.DATA63 0748 61CBh,
CANFDG0_NS.RFB4.DATA0 0748 620Ch to CANFDG0_NS.RFB4.DATA63 0748 624Bh,
CANFDG0_NS.RFB5.DATA0 0748 628Ch to CANFDG0_NS.RFB5.DATA63 0748 62CBh,
CANFDG0_NS.RFB6.DATA0 0748 630Ch to CANFDG0_NS.RFB6.DATA63 0748 634Bh,
CANFDG0_NS.RFB7.DATA0 0748 638Ch to CANFDG0_NS.RFB7.DATA63 0748 63CBh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data bytes of the received message. Unused data bytes are filled with 00h.

45.2.79.14 Common FIFO n (CFBn) (n = 0 to 5)

The total number of the common FIFOs (CFB) is six, as shown in Figure 45.33.
The common FIFO consists of the following registers:

- CFBn.HF0
- CFBn.HF1
- CFBn.HF2
- CFBn.DF0 to CFBn.DF15

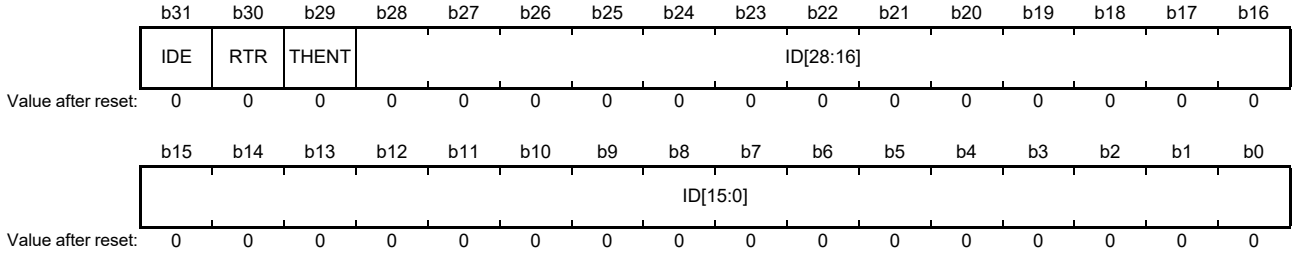
Table 45.12 shows the structure of this buffer.

Table 45.12 Structure of Common FIFO

Address Offset	Symbol	Register Name	Contents
+00h	CFBn.HF0	Common FIFO n Header Field 0	IDE, RTR, ID
+04h	CFBn.HF1	Common FIFO n Header Field 1	DLC, timestamp
+08h	CFBn.HF2	Common FIFO n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	CFBn.DF0	Common FIFO n Data Field 0	DATA0 to DATA3
+10h	CFBn.DF1	Common FIFO n Data Field 1	DATA4 to DATA7
+14h	CFBn.DF2	Common FIFO n Data Field 2	DATA8 to DATA11
+18h	CFBn.DF3	Common FIFO n Data Field 3	DATA12 to DATA15
+1Ch	CFBn.DF4	Common FIFO n Data Field 4	DATA16 to DATA19
+20h	CFBn.DF5	Common FIFO n Data Field 5	DATA20 to DATA23
+24h	CFBn.DF6	Common FIFO n Data Field 6	DATA24 to DATA27
+28h	CFBn.DF7	Common FIFO n Data Field 7	DATA28 to DATA31
+2Ch	CFBn.DF8	Common FIFO n Data Field 8	DATA32 to DATA35
+30h	CFBn.DF9	Common FIFO n Data Field 9	DATA36 to DATA39
+34h	CFBn.DF10	Common FIFO n Data Field 10	DATA40 to DATA43
+38h	CFBn.DF11	Common FIFO n Data Field 11	DATA44 to DATA47
+3Ch	CFBn.DF12	Common FIFO n Data Field 12	DATA48 to DATA51
+40h	CFBn.DF13	Common FIFO n Data Field 13	DATA52 to DATA55
+44h	CFBn.DF14	Common FIFO n Data Field 14	DATA56 to DATA59
+48h	CFBn.DF15	Common FIFO n Data Field 15	DATA60 to DATA63

45.2.79.15 Common FIFO n Header Field 0 (CFBn.HF0) (n = 0 to 5)

Address(es): CANFDG0.CFB0.HF0 8748 6400h, CANFDG0.CFB1.HF0 8748 6480h, CANFDG0.CFB2.HF0 8748 6500h, CANFDG0.CFB3.HF0 8748 6580h, CANFDG0.CFB4.HF0 8748 6600h, CANFDG0.CFB5.HF0 8748 6680h, CANFDG0_NS.CFB0.HF0 0748 6400h, CANFDG0_NS.CFB1.HF0 0748 6480h, CANFDG0_NS.CFB2.HF0 0748 6500h, CANFDG0_NS.CFB3.HF0 0748 6580h, CANFDG0_NS.CFB4.HF0 0748 6600h, CANFDG0_NS.CFB5.HF0 0748 6680h



Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	Receive FIFO mode or gateway FIFO mode: Reserved. This bit is read as 0. Transmit FIFO mode: 0: Entry is not stored to the Transmission History after successful transmission. 1: Entry is stored to the Transmission History after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

In receive FIFO mode or gateway FIFO mode, this register is a read-only register for reading the ID field, IDE bit, and RTR bit of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the ID field, IDE bit, and RTR bit of the message to be transmitted at the end of the FIFO buffer.

ID[28:0] Bits (Identifier)

In receive FIFO mode, the ID[28:0] bits store the standard or extended identifier field of the received message.
In transmit FIFO mode, the ID[28:0] bits are used to set the value of the standard or extended identifier field of the message to be transmitted.
For alignment of these bits in base and extended format, refer to section 45.2.78, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

This bit is valid only in transmit FIFO mode.
The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message.

RTR Bit (Remote Transmission Request)

In receive FIFO mode, the RTR bit stores the value of the RTR bit of the received message.
In transmit FIFO mode, the RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When a CAN FD frame was received (receive mode), this bit reflects the value of the RRS bit. When transmitting a CAN FD frame (CFBn.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

In receive FIFO mode, the IDE bit stores the value of the IDE bit of the received message.
In transmit FIFO mode, the IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

45.2.79.16 Common FIFO n Header Field 1 (CFBn.HF1) (n = 0 to 5)

Address(es): CANFDG0.CFB0.HF1 8748 6404h, CANFDG0.CFB1.HF1 8748 6484h, CANFDG0.CFB2.HF1 8748 6504h, CANFDG0.CFB3.HF1 8748 6584h, CANFDG0.CFB4.HF1 8748 6604h, CANFDG0.CFB5.HF1 8748 6684h, CANFDG0_NS.CFB0.HF1 0748 6404h, CANFDG0_NS.CFB1.HF1 0748 6484h, CANFDG0_NS.CFB2.HF1 0748 6504h, CANFDG0_NS.CFB3.HF1 0748 6584h, CANFDG0_NS.CFB4.HF1 0748 6604h, CANFDG0_NS.CFB5.HF1 0748 6684h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame (in receive FIFO mode)	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame	R/W

In receive FIFO mode or gateway FIFO mode, this register is a read-only register for reading the data length code (DLC) and timestamp of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the data length code (DLC) of the message to be transmitted at the end of the FIFO buffer.

TS[15:0] Bits (Timestamp)

These bits are valid only in receive FIFO mode.
The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

In receive FIFO mode, the DLC[3:0] bits store the number of data bytes of the received message.
In transmit FIFO mode, The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the CFCRn.PLS[2:0] bits.

45.2.79.17 Common FIFO n Header Field 2 (CFBn.HF2) (n = 0 to 5)

Address(es): CANFDG0.CFB0.HF2 8748 6408h, CANFDG0.CFB1.HF2 8748 6488h, CANFDG0.CFB2.HF2 8748 6508h, CANFDG0.CFB3.HF2 8748 6588h, CANFDG0.CFB4.HF2 8748 6608h, CANFDG0.CFB5.HF2 8748 6688h, CANFDG0_NS.CFB0.HF2 0748 6408h, CANFDG0_NS.CFB1.HF2 0748 6488h, CANFDG0_NS.CFB2.HF2 0748 6508h, CANFDG0_NS.CFB3.HF2 0748 6588h, CANFDG0_NS.CFB4.HF2 0748 6608h, CANFDG0_NS.CFB5.HF2 0748 6688h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame received or to transmit by error active node 1: CAN FD frame received or to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame received or to transmit with no bit rate switch 1: CAN FD frame received or to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame received or to transmit 1: CAN FD frame received or to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	This field stores the information label attached by the Acceptance Filter or is used to set the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	This field stores the pointer attached by the Acceptance Filter or is used to set the pointer to be stored in the Transmission History	R/W

In receive FIFO mode or gateway FIFO mode, this register is a read-only register for reading the FDF bit, BRS bit, and ESI flag of the received message, and information label and pointer for the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history at the end of the FIFO buffer.

ESI Bit (Error State Indicator)

In receive FIFO mode or gateway FIFO mode, the ESI bit stores the value of the ESI flag of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted. If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

In receive FIFO mode or gateway FIFO mode, the BRS bit stores the value of the BRS bit of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

In receive FIFO mode or gateway FIFO mode, the FDF bit stores the value of the FDF bit of the received CAN FD frame.
In transmit FIFO mode, the FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

In receive FIFO mode or gateway FIFO mode, the IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

In receive FIFO mode or gateway FIFO mode, the PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

45.2.79.18 Common FIFO n Data Field p (CFBn.DFp) (n = 0 to 5; p = 0 to 15)

Address(es): CANFDG0.CFB0.DF0 8748 640Ch to CANFDG0.CFB0.DF15 8748 6448h,
CANFDG0.CFB1.DF0 8748 648Ch to CANFDG0.CFB1.DF15 8748 64C8h,
CANFDG0.CFB2.DF0 8748 650Ch to CANFDG0.CFB2.DF15 8748 6548h,
CANFDG0.CFB3.DF0 8748 658Ch to CANFDG0.CFB3.DF15 8748 65C8h,
CANFDG0.CFB4.DF0 8748 660Ch to CANFDG0.CFB4.DF15 8748 6648h,
CANFDG0.CFB5.DF0 8748 668Ch to CANFDG0.CFB5.DF15 8748 66C8h,
CANFDG0_NS.CFB0.DF0 0748 640Ch to CANFDG0_NS.CFB0.DF15 0748 6448h,
CANFDG0_NS.CFB1.DF0 0748 648Ch to CANFDG0_NS.CFB1.DF15 0748 64C8h,
CANFDG0_NS.CFB2.DF0 0748 650Ch to CANFDG0_NS.CFB2.DF15 0748 6548h,
CANFDG0_NS.CFB3.DF0 0748 658Ch to CANFDG0_NS.CFB3.DF15 0748 65C8h,
CANFDG0_NS.CFB4.DF0 0748 660Ch to CANFDG0_NS.CFB4.DF15 0748 6648h,
CANFDG0_NS.CFB5.DF0 0748 668Ch to CANFDG0_NS.CFB5.DF15 0748 66C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In receive FIFO mode or gateway FIFO mode, these registers are read-only registers for reading the data byte (p × 4 + 3) to data byte (p × 4) of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.

45.2.79.19 Common FIFO n Data k (CFBn.DATAk) (n = 0 to 5; k = 0 to 63)

Address(es): CANFDG0.CFB0.DATA0 8748 640Ch to CANFDG0.CFB0.DATA63 8748 644Bh,
CANFDG0.CFB1.DATA0 8748 648Ch to CANFDG0.CFB1.DATA63 8748 64CBh,
CANFDG0.CFB2.DATA0 8748 650Ch to CANFDG0.CFB2.DATA63 8748 654Bh,
CANFDG0.CFB3.DATA0 8748 658Ch to CANFDG0.CFB3.DATA63 8748 65CBh,
CANFDG0.CFB4.DATA0 8748 660Ch to CANFDG0.CFB4.DATA63 8748 664Bh,
CANFDG0.CFB5.DATA0 8748 668Ch to CANFDG0.CFB5.DATA63 8748 66CBh,
CANFDG0_NS.CFB0.DATA0 0748 640Ch to CANFDG0_NS.CFB0.DATA63 0748 644Bh,
CANFDG0_NS.CFB1.DATA0 0748 648Ch to CANFDG0_NS.CFB1.DATA63 0748 64CBh,
CANFDG0_NS.CFB2.DATA0 0748 650Ch to CANFDG0_NS.CFB2.DATA63 0748 654Bh,
CANFDG0_NS.CFB3.DATA0 0748 658Ch to CANFDG0_NS.CFB3.DATA63 0748 65CBh,
CANFDG0_NS.CFB4.DATA0 0748 660Ch to CANFDG0_NS.CFB4.DATA63 0748 664Bh,
CANFDG0_NS.CFB5.DATA0 0748 668Ch to CANFDG0_NS.CFB5.DATA63 0748 66CBh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

In receive FIFO mode, these registers are read-only registers for reading the data bytes of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing the data bytes of the message to be transmitted.

45.2.79.20 Transmit Message Buffer n (TMBn) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

The total number of transmit message buffers (TMB) is 32, as shown in Figure 45.33.
The transmit message buffer consists of the following registers:

- TMBn.HF0
- TMBn.HF1
- TMBn.HF2
- TMBn.DF0 to TMBn.DF15

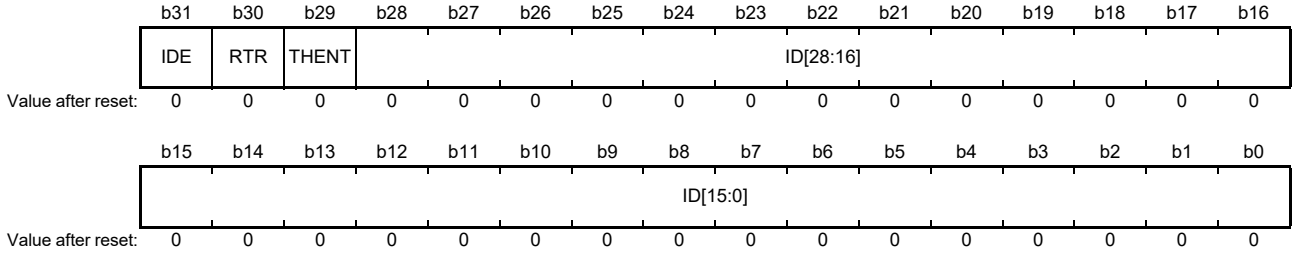
Table 45.13 shows the structure of this buffer.

Table 45.13 Structure of Transmit Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	TMBn.HF0	Transmit Message Buffer n Header Field 0	IDE, RTR, ID
+04h	TMBn.HF1	Transmit Message Buffer n Header Field 1	DLC
+08h	TMBn.HF2	Transmit Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	TMBn.DF0	Transmit Message Buffer n Data Field 0	DATA0 to DATA3
+10h	TMBn.DF1	Transmit Message Buffer n Data Field 1	DATA4 to DATA7
+14h	TMBn.DF2	Transmit Message Buffer n Data Field 2	DATA8 to DATA11
+18h	TMBn.DF3	Transmit Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	TMBn.DF4	Transmit Message Buffer n Data Field 4	DATA16 to DATA19
+20h	TMBn.DF5	Transmit Message Buffer n Data Field 5	DATA20 to DATA23
+24h	TMBn.DF6	Transmit Message Buffer n Data Field 6	DATA24 to DATA27
+28h	TMBn.DF7	Transmit Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	TMBn.DF8	Transmit Message Buffer n Data Field 8	DATA32 to DATA35
+30h	TMBn.DF9	Transmit Message Buffer n Data Field 9	DATA36 to DATA39
+34h	TMBn.DF10	Transmit Message Buffer n Data Field 10	DATA40 to DATA43
+38h	TMBn.DF11	Transmit Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	TMBn.DF12	Transmit Message Buffer n Data Field 12	DATA48 to DATA51
+40h	TMBn.DF13	Transmit Message Buffer n Data Field 13	DATA52 to DATA55
+44h	TMBn.DF14	Transmit Message Buffer n Data Field 14	DATA56 to DATA59
+48h	TMBn.DF15	Transmit Message Buffer n Data Field 15	DATA60 to DATA63

45.2.79.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Address(es): CANFDG0.TMB0.HF0 8749 0000h, CANFDG0.TMB1.HF0 8749 0080h, CANFDG0.TMB2.HF0 8749 0100h, CANFDG0.TMB3.HF0 8749 0180h, CANFDG0.TMB4.HF0 8749 0200h, CANFDG0.TMB5.HF0 8749 0280h, CANFDG0.TMB6.HF0 8749 0300h, CANFDG0.TMB7.HF0 8749 0380h, CANFDG0.TMB32.HF0 8749 1000h, CANFDG0.TMB33.HF0 8749 1080h, CANFDG0.TMB34.HF0 8749 1100h, CANFDG0.TMB35.HF0 8749 1180h, CANFDG0.TMB36.HF0 8749 1200h, CANFDG0.TMB37.HF0 8749 1280h, CANFDG0.TMB38.HF0 8749 1300h, CANFDG0.TMB39.HF0 8749 1380h, CANFDG0.TMB64.HF0 8749 2000h, CANFDG0.TMB65.HF0 8749 2080h, CANFDG0.TMB66.HF0 8749 2100h, CANFDG0.TMB67.HF0 8749 2180h, CANFDG0.TMB68.HF0 8749 2200h, CANFDG0.TMB69.HF0 8749 2280h, CANFDG0.TMB70.HF0 8749 2300h, CANFDG0.TMB71.HF0 8749 2380h, CANFDG0.TMB96.HF0 8749 3000h, CANFDG0.TMB97.HF0 8749 3080h, CANFDG0.TMB98.HF0 8749 3100h, CANFDG0.TMB99.HF0 8749 3180h, CANFDG0.TMB100.HF0 8749 3200h, CANFDG0.TMB101.HF0 8749 3280h, CANFDG0.TMB102.HF0 8749 3300h, CANFDG0.TMB103.HF0 8749 3380h, CANFDG0_NS.TMB0.HF0 0749 0000h, CANFDG0_NS.TMB1.HF0 0749 0080h, CANFDG0_NS.TMB2.HF0 0749 0100h, CANFDG0_NS.TMB3.HF0 0749 0180h, CANFDG0_NS.TMB4.HF0 0749 0200h, CANFDG0_NS.TMB5.HF0 0749 0280h, CANFDG0_NS.TMB6.HF0 0749 0300h, CANFDG0_NS.TMB7.HF0 0749 0380h, CANFDG0_NS.TMB32.HF0 0749 1000h, CANFDG0_NS.TMB33.HF0 0749 1080h, CANFDG0_NS.TMB34.HF0 0749 1100h, CANFDG0_NS.TMB35.HF0 0749 1180h, CANFDG0_NS.TMB36.HF0 0749 1200h, CANFDG0_NS.TMB37.HF0 0749 1280h, CANFDG0_NS.TMB38.HF0 0749 1300h, CANFDG0_NS.TMB39.HF0 0749 1380h, CANFDG0_NS.TMB64.HF0 0749 2000h, CANFDG0_NS.TMB65.HF0 0749 2080h, CANFDG0_NS.TMB66.HF0 0749 2100h, CANFDG0_NS.TMB67.HF0 0749 2180h, CANFDG0_NS.TMB68.HF0 0749 2200h, CANFDG0_NS.TMB69.HF0 0749 2280h, CANFDG0_NS.TMB70.HF0 0749 2300h, CANFDG0_NS.TMB71.HF0 0749 2380h, CANFDG0_NS.TMB96.HF0 0749 3000h, CANFDG0_NS.TMB97.HF0 0749 3080h, CANFDG0_NS.TMB98.HF0 0749 3100h, CANFDG0_NS.TMB99.HF0 0749 3180h, CANFDG0_NS.TMB100.HF0 0749 3200h, CANFDG0_NS.TMB101.HF0 0749 3280h, CANFDG0_NS.TMB102.HF0 0749 3300h, CANFDG0_NS.TMB103.HF0 0749 3380h



Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	0: Entry is not stored in transmission history after successful transmission. 1: Entry is stored in transmission history after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

This register is used to store the ID field, IDE bit, and RTR bit of the message to be transmitted, and to specify whether to store it in the transmission history.
Do not write to this register in CH_SLEEP mode.

ID[28:0] Bits (Identifier)

The ID[28:0] bits are used to set the value of the standard or extended identifier field of the message stored in the transmit message buffer.
For alignment of these bits in base and extended format, refer to section 45.2.78, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message stored in the transmit message buffer.

RTR Bit (Remote Transmission Request)

The RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When transmitting a CAN FD frame (TMBn.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

The IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

45.2.79.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Address(es): CANFDG0.TMB0.HF1 8749 0004h, CANFDG0.TMB1.HF1 8749 0084h, CANFDG0.TMB2.HF1 8749 0104h, CANFDG0.TMB3.HF1 8749 0184h, CANFDG0.TMB4.HF1 8749 0204h, CANFDG0.TMB5.HF1 8749 0284h, CANFDG0.TMB6.HF1 8749 0304h, CANFDG0.TMB7.HF1 8749 0384h, CANFDG0.TMB32.HF1 8749 1004h, CANFDG0.TMB33.HF1 8749 1084h, CANFDG0.TMB34.HF1 8749 1104h, CANFDG0.TMB35.HF1 8749 1184h, CANFDG0.TMB36.HF1 8749 1204h, CANFDG0.TMB37.HF1 8749 1284h, CANFDG0.TMB38.HF1 8749 1304h, CANFDG0.TMB39.HF1 8749 1384h, CANFDG0.TMB64.HF1 8749 2004h, CANFDG0.TMB65.HF1 8749 2084h, CANFDG0.TMB66.HF1 8749 2104h, CANFDG0.TMB67.HF1 8749 2184h, CANFDG0.TMB68.HF1 8749 2204h, CANFDG0.TMB69.HF1 8749 2284h, CANFDG0.TMB70.HF1 8749 2304h, CANFDG0.TMB71.HF1 8749 2384h, CANFDG0.TMB96.HF1 8749 3004h, CANFDG0.TMB97.HF1 8749 3084h, CANFDG0.TMB98.HF1 8749 3104h, CANFDG0.TMB99.HF1 8749 3184h, CANFDG0.TMB100.HF1 8749 3204h, CANFDG0.TMB101.HF1 8749 3284h, CANFDG0.TMB102.HF1 8749 3304h, CANFDG0.TMB103.HF1 8749 3384h, CANFDG0_NS.TMB0.HF1 0749 0004h, CANFDG0_NS.TMB1.HF1 0749 0084h, CANFDG0_NS.TMB2.HF1 0749 0104h, CANFDG0_NS.TMB3.HF1 0749 0184h, CANFDG0_NS.TMB4.HF1 0749 0204h, CANFDG0_NS.TMB5.HF1 0749 0284h, CANFDG0_NS.TMB6.HF1 0749 0304h, CANFDG0_NS.TMB7.HF1 0749 0384h, CANFDG0_NS.TMB32.HF1 0749 1004h, CANFDG0_NS.TMB33.HF1 0749 1084h, CANFDG0_NS.TMB34.HF1 0749 1104h, CANFDG0_NS.TMB35.HF1 0749 1184h, CANFDG0_NS.TMB36.HF1 0749 1204h, CANFDG0_NS.TMB37.HF1 0749 1284h, CANFDG0_NS.TMB38.HF1 0749 1304h, CANFDG0_NS.TMB39.HF1 0749 1384h, CANFDG0_NS.TMB64.HF1 0749 2004h, CANFDG0_NS.TMB65.HF1 0749 2084h, CANFDG0_NS.TMB66.HF1 0749 2104h, CANFDG0_NS.TMB67.HF1 0749 2184h, CANFDG0_NS.TMB68.HF1 0749 2204h, CANFDG0_NS.TMB69.HF1 0749 2284h, CANFDG0_NS.TMB70.HF1 0749 2304h, CANFDG0_NS.TMB71.HF1 0749 2384h, CANFDG0_NS.TMB96.HF1 0749 3004h, CANFDG0_NS.TMB97.HF1 0749 3084h, CANFDG0_NS.TMB98.HF1 0749 3104h, CANFDG0_NS.TMB99.HF1 0749 3184h, CANFDG0_NS.TMB100.HF1 0749 3204h, CANFDG0_NS.TMB101.HF1 0749 3284h, CANFDG0_NS.TMB102.HF1 0749 3304h, CANFDG0_NS.TMB103.HF1 0749 3384h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes to be transmitted in a CAN frame.	R/W

This register is used to store the data length code (DLC) fields of the message to be transmitted.
Do not write to this register in CH_SLEEP mode.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted when the corresponding TMBn.HF0.RTR bit is set to 0.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

45.2.79.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Address(es): CANFDG0.TMB0.HF2 8749 0008h, CANFDG0.TMB1.HF2 8749 0088h, CANFDG0.TMB2.HF2 8749 0108h, CANFDG0.TMB3.HF2 8749 0188h, CANFDG0.TMB4.HF2 8749 0208h, CANFDG0.TMB5.HF2 8749 0288h, CANFDG0.TMB6.HF2 8749 0308h, CANFDG0.TMB7.HF2 8749 0388h, CANFDG0.TMB32.HF2 8749 1008h, CANFDG0.TMB33.HF2 8749 1088h, CANFDG0.TMB34.HF2 8749 1108h, CANFDG0.TMB35.HF2 8749 1188h, CANFDG0.TMB36.HF2 8749 1208h, CANFDG0.TMB37.HF2 8749 1288h, CANFDG0.TMB38.HF2 8749 1308h, CANFDG0.TMB39.HF2 8749 1388h, CANFDG0.TMB64.HF2 8749 2008h, CANFDG0.TMB65.HF2 8749 2088h, CANFDG0.TMB66.HF2 8749 2108h, CANFDG0.TMB67.HF2 8749 2188h, CANFDG0.TMB68.HF2 8749 2208h, CANFDG0.TMB69.HF2 8749 2288h, CANFDG0.TMB70.HF2 8749 2308h, CANFDG0.TMB71.HF2 8749 2388h, CANFDG0.TMB96.HF2 8749 3008h, CANFDG0.TMB97.HF2 8749 3088h, CANFDG0.TMB98.HF2 8749 3108h, CANFDG0.TMB99.HF2 8749 3188h, CANFDG0.TMB100.HF2 8749 3208h, CANFDG0.TMB101.HF2 8749 3288h, CANFDG0.TMB102.HF2 8749 3308h, CANFDG0.TMB103.HF2 8749 3388h, CANFDG0_NS.TMB0.HF2 0749 0008h, CANFDG0_NS.TMB1.HF2 0749 0088h, CANFDG0_NS.TMB2.HF2 0749 0108h, CANFDG0_NS.TMB3.HF2 0749 0188h, CANFDG0_NS.TMB4.HF2 0749 0208h, CANFDG0_NS.TMB5.HF2 0749 0288h, CANFDG0_NS.TMB6.HF2 0749 0308h, CANFDG0_NS.TMB7.HF2 0749 0388h, CANFDG0_NS.TMB32.HF2 0749 1008h, CANFDG0_NS.TMB33.HF2 0749 1088h, CANFDG0_NS.TMB34.HF2 0749 1108h, CANFDG0_NS.TMB35.HF2 0749 1188h, CANFDG0_NS.TMB36.HF2 0749 1208h, CANFDG0_NS.TMB37.HF2 0749 1288h, CANFDG0_NS.TMB38.HF2 0749 1308h, CANFDG0_NS.TMB39.HF2 0749 1388h, CANFDG0_NS.TMB64.HF2 0749 2008h, CANFDG0_NS.TMB65.HF2 0749 2088h, CANFDG0_NS.TMB66.HF2 0749 2108h, CANFDG0_NS.TMB67.HF2 0749 2188h, CANFDG0_NS.TMB68.HF2 0749 2208h, CANFDG0_NS.TMB69.HF2 0749 2288h, CANFDG0_NS.TMB70.HF2 0749 2308h, CANFDG0_NS.TMB71.HF2 0749 2388h, CANFDG0_NS.TMB96.HF2 0749 3008h, CANFDG0_NS.TMB97.HF2 0749 3088h, CANFDG0_NS.TMB98.HF2 0749 3108h, CANFDG0_NS.TMB99.HF2 0749 3188h, CANFDG0_NS.TMB100.HF2 0749 3208h, CANFDG0_NS.TMB101.HF2 0749 3288h, CANFDG0_NS.TMB102.HF2 0749 3308h, CANFDG0_NS.TMB103.HF2 0749 3388h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame to transmit by error active node 1: CAN FD frame to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame to transmit with no bit rate switch 1: CAN FD frame to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame to transmit 1: CAN FD frame to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	This field is used to set the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	This field is used to set the pointer to be stored in the Transmission History	R/W

This register is used to store the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history.
Do not write to this register in CH_SLEEP mode.

ESI Bit (Error State Indicator)

The ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted.
If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

The BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

The FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

45.2.79.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103; p = 0 to 15)

Address(es): CANFDG0.TMB0.DF0 8749 000Ch to CANFDG0.TMB0.DF15 8749 0048h,
CANFDG0.TMB1.DF0 8749 008Ch to CANFDG0.TMB1.DF15 8749 00C8h,
CANFDG0.TMB2.DF0 8749 010Ch to CANFDG0.TMB2.DF15 8749 0148h,
CANFDG0.TMB3.DF0 8749 018Ch to CANFDG0.TMB3.DF15 8749 01C8h,
CANFDG0.TMB4.DF0 8749 020Ch to CANFDG0.TMB4.DF15 8749 0248h,
CANFDG0.TMB5.DF0 8749 028Ch to CANFDG0.TMB5.DF15 8749 02C8h,
CANFDG0.TMB6.DF0 8749 030Ch to CANFDG0.TMB6.DF15 8749 0348h,
CANFDG0.TMB7.DF0 8749 038Ch to CANFDG0.TMB7.DF15 8749 03C8h,
CANFDG0.TMB32.DF0 8749 100Ch to CANFDG0.TMB32.DF15 8749 1048h,
CANFDG0.TMB33.DF0 8749 108Ch to CANFDG0.TMB33.DF15 8749 10C8h,
CANFDG0.TMB34.DF0 8749 110Ch to CANFDG0.TMB34.DF15 8749 1148h,
CANFDG0.TMB35.DF0 8749 118Ch to CANFDG0.TMB35.DF15 8749 11C8h,
CANFDG0.TMB36.DF0 8749 120Ch to CANFDG0.TMB36.DF15 8749 1248h,
CANFDG0.TMB37.DF0 8749 128Ch to CANFDG0.TMB37.DF15 8749 12C8h,
CANFDG0.TMB38.DF0 8749 130Ch to CANFDG0.TMB38.DF15 8749 1348h,
CANFDG0.TMB39.DF0 8749 138Ch to CANFDG0.TMB39.DF15 8749 13C8h,
CANFDG0.TMB64.DF0 8749 200Ch to CANFDG0.TMB64.DF15 8749 2048h,
CANFDG0.TMB65.DF0 8749 208Ch to CANFDG0.TMB65.DF15 8749 20C8h,
CANFDG0.TMB66.DF0 8749 210Ch to CANFDG0.TMB66.DF15 8749 2148h,
CANFDG0.TMB67.DF0 8749 218Ch to CANFDG0.TMB67.DF15 8749 21C8h,
CANFDG0.TMB68.DF0 8749 220Ch to CANFDG0.TMB68.DF15 8749 2248h,
CANFDG0.TMB69.DF0 8749 228Ch to CANFDG0.TMB69.DF15 8749 22C8h,
CANFDG0.TMB70.DF0 8749 230Ch to CANFDG0.TMB70.DF15 8749 2348h,
CANFDG0.TMB71.DF0 8749 238Ch to CANFDG0.TMB71.DF15 8749 23C8h,
CANFDG0.TMB96.DF0 8749 300Ch to CANFDG0.TMB96.DF15 8749 3048h,
CANFDG0.TMB97.DF0 8749 308Ch to CANFDG0.TMB97.DF15 8749 30C8h,
CANFDG0.TMB98.DF0 8749 310Ch to CANFDG0.TMB98.DF15 8749 3148h,
CANFDG0.TMB99.DF0 8749 318Ch to CANFDG0.TMB99.DF15 8749 31C8h,
CANFDG0.TMB100.DF0 8749 320Ch to CANFDG0.TMB100.DF15 8749 3248h,
CANFDG0.TMB101.DF0 8749 328Ch to CANFDG0.TMB101.DF15 8749 32C8h,
CANFDG0.TMB102.DF0 8749 330Ch to CANFDG0.TMB102.DF15 8749 3348h,
CANFDG0.TMB103.DF0 8749 338Ch to CANFDG0.TMB103.DF15 8749 33C8h,
CANFDG0_NS.TMB0.DF0 0749 000Ch to CANFDG0_NS.TMB0.DF15 0749 0048h,
CANFDG0_NS.TMB1.DF0 0749 008Ch to CANFDG0_NS.TMB1.DF15 0749 00C8h,
CANFDG0_NS.TMB2.DF0 0749 010Ch to CANFDG0_NS.TMB2.DF15 0749 0148h,
CANFDG0_NS.TMB3.DF0 0749 018Ch to CANFDG0_NS.TMB3.DF15 0749 01C8h,
CANFDG0_NS.TMB4.DF0 0749 020Ch to CANFDG0_NS.TMB4.DF15 0749 0248h,
CANFDG0_NS.TMB5.DF0 0749 028Ch to CANFDG0_NS.TMB5.DF15 0749 02C8h,
CANFDG0_NS.TMB6.DF0 0749 030Ch to CANFDG0_NS.TMB6.DF15 0749 0348h,
CANFDG0_NS.TMB7.DF0 0749 038Ch to CANFDG0_NS.TMB7.DF15 0749 03C8h,
CANFDG0_NS.TMB32.DF0 0749 100Ch to CANFDG0_NS.TMB32.DF15 0749 1048h,
CANFDG0_NS.TMB33.DF0 0749 108Ch to CANFDG0_NS.TMB33.DF15 0749 10C8h,
CANFDG0_NS.TMB34.DF0 0749 110Ch to CANFDG0_NS.TMB34.DF15 0749 1148h,
CANFDG0_NS.TMB35.DF0 0749 118Ch to CANFDG0_NS.TMB35.DF15 0749 11C8h,
CANFDG0_NS.TMB36.DF0 0749 120Ch to CANFDG0_NS.TMB36.DF15 0749 1248h,
CANFDG0_NS.TMB37.DF0 0749 128Ch to CANFDG0_NS.TMB37.DF15 0749 12C8h,
CANFDG0_NS.TMB38.DF0 0749 130Ch to CANFDG0_NS.TMB38.DF15 0749 1348h,
CANFDG0_NS.TMB39.DF0 0749 138Ch to CANFDG0_NS.TMB39.DF15 0749 13C8h,
CANFDG0_NS.TMB64.DF0 0749 200Ch to CANFDG0_NS.TMB64.DF15 0749 2048h,
CANFDG0_NS.TMB65.DF0 0749 208Ch to CANFDG0_NS.TMB65.DF15 0749 20C8h,
CANFDG0_NS.TMB66.DF0 0749 210Ch to CANFDG0_NS.TMB66.DF15 0749 2148h,
CANFDG0_NS.TMB67.DF0 0749 218Ch to CANFDG0_NS.TMB67.DF15 0749 21C8h,
CANFDG0_NS.TMB68.DF0 0749 220Ch to CANFDG0_NS.TMB68.DF15 0749 2248h,
CANFDG0_NS.TMB69.DF0 0749 228Ch to CANFDG0_NS.TMB69.DF15 0749 22C8h,
CANFDG0_NS.TMB70.DF0 0749 230Ch to CANFDG0_NS.TMB70.DF15 0749 2348h,
CANFDG0_NS.TMB71.DF0 0749 238Ch to CANFDG0_NS.TMB71.DF15 0749 23C8h,
CANFDG0_NS.TMB96.DF0 0749 300Ch to CANFDG0_NS.TMB96.DF15 0749 3048h,
CANFDG0_NS.TMB97.DF0 0749 308Ch to CANFDG0_NS.TMB97.DF15 0749 30C8h,
CANFDG0_NS.TMB98.DF0 0749 310Ch to CANFDG0_NS.TMB98.DF15 0749 3148h,
CANFDG0_NS.TMB99.DF0 0749 318Ch to CANFDG0_NS.TMB99.DF15 0749 31C8h,
CANFDG0_NS.TMB100.DF0 0749 320Ch to CANFDG0_NS.TMB100.DF15 0749 3248h,
CANFDG0_NS.TMB101.DF0 0749 328Ch to CANFDG0_NS.TMB101.DF15 0749 32C8h,
CANFDG0_NS.TMB102.DF0 0749 330Ch to CANFDG0_NS.TMB102.DF15 0749 3348h,
CANFDG0_NS.TMB103.DF0 0749 338Ch to CANFDG0_NS.TMB103.DF15 0749 33C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.

Do not write to these registers in CH_SLEEP mode.

45.2.79.25 Transmit Message Buffer n Data k (TMBn.DATAk) (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103; k = 0 to 63)

Address(es): CANFDG0.TMB0.DATA0 8749 000Ch to CANFDG0.TMB0.DATA63 8749 004Bh,
CANFDG0.TMB1.DATA0 8749 008Ch to CANFDG0.TMB1.DATA63 8749 00CBh,
CANFDG0.TMB2.DATA0 8749 010Ch to CANFDG0.TMB2.DATA63 8749 014Bh,
CANFDG0.TMB3.DATA0 8749 018Ch to CANFDG0.TMB3.DATA63 8749 01CBh,
CANFDG0.TMB4.DATA0 8749 020Ch to CANFDG0.TMB4.DATA63 8749 024Bh,
CANFDG0.TMB5.DATA0 8749 028Ch to CANFDG0.TMB5.DATA63 8749 02CBh,
CANFDG0.TMB6.DATA0 8749 030Ch to CANFDG0.TMB6.DATA63 8749 034Bh,
CANFDG0.TMB7.DATA0 8749 038Ch to CANFDG0.TMB7.DATA63 8749 03CBh,
CANFDG0.TMB32.DATA0 8749 100Ch to CANFDG0.TMB32.DATA63 8749 104Bh,
CANFDG0.TMB33.DATA0 8749 108Ch to CANFDG0.TMB33.DATA63 8749 10CBh,
CANFDG0.TMB34.DATA0 8749 110Ch to CANFDG0.TMB34.DATA63 8749 114Bh,
CANFDG0.TMB35.DATA0 8749 118Ch to CANFDG0.TMB35.DATA63 8749 11CBh,
CANFDG0.TMB36.DATA0 8749 120Ch to CANFDG0.TMB36.DATA63 8749 124Bh,
CANFDG0.TMB37.DATA0 8749 128Ch to CANFDG0.TMB37.DATA63 8749 12CBh,
CANFDG0.TMB38.DATA0 8749 130Ch to CANFDG0.TMB38.DATA63 8749 134Bh,
CANFDG0.TMB39.DATA0 8749 138Ch to CANFDG0.TMB39.DATA63 8749 13CBh,
CANFDG0.TMB64.DATA0 8749 200Ch to CANFDG0.TMB64.DATA63 8749 204Bh,
CANFDG0.TMB65.DATA0 8749 208Ch to CANFDG0.TMB65.DATA63 8749 20CBh,
CANFDG0.TMB66.DATA0 8749 210Ch to CANFDG0.TMB66.DATA63 8749 214Bh,
CANFDG0.TMB67.DATA0 8749 218Ch to CANFDG0.TMB67.DATA63 8749 21CBh,
CANFDG0.TMB68.DATA0 8749 220Ch to CANFDG0.TMB68.DATA63 8749 224Bh,
CANFDG0.TMB69.DATA0 8749 228Ch to CANFDG0.TMB69.DATA63 8749 22CBh,
CANFDG0.TMB70.DATA0 8749 230Ch to CANFDG0.TMB70.DATA63 8749 234Bh,
CANFDG0.TMB71.DATA0 8749 238Ch to CANFDG0.TMB71.DATA63 8749 23CBh,
CANFDG0.TMB96.DATA0 8749 300Ch to CANFDG0.TMB96.DATA63 8749 304Bh,
CANFDG0.TMB97.DATA0 8749 308Ch to CANFDG0.TMB97.DATA63 8749 30CBh,
CANFDG0.TMB98.DATA0 8749 310Ch to CANFDG0.TMB98.DATA63 8749 314Bh,
CANFDG0.TMB99.DATA0 8749 318Ch to CANFDG0.TMB99.DATA63 8749 31CBh,
CANFDG0.TMB100.DATA0 8749 320Ch to CANFDG0.TMB100.DATA63 8749 324Bh,
CANFDG0.TMB101.DATA0 8749 328Ch to CANFDG0.TMB101.DATA63 8749 32CBh,
CANFDG0.TMB102.DATA0 8749 330Ch to CANFDG0.TMB102.DATA63 8749 334Bh,
CANFDG0.TMB103.DATA0 8749 338Ch to CANFDG0.TMB103.DATA63 8749 33CBh,
CANFDG0_NS.TMB0.DATA0 0749 000Ch to CANFDG0_NS.TMB0.DATA63 0749 004Bh,
CANFDG0_NS.TMB1.DATA0 0749 008Ch to CANFDG0_NS.TMB1.DATA63 0749 00CBh,
CANFDG0_NS.TMB2.DATA0 0749 010Ch to CANFDG0_NS.TMB2.DATA63 0749 014Bh,
CANFDG0_NS.TMB3.DATA0 0749 018Ch to CANFDG0_NS.TMB3.DATA63 0749 01CBh,
CANFDG0_NS.TMB4.DATA0 0749 020Ch to CANFDG0_NS.TMB4.DATA63 0749 024Bh,
CANFDG0_NS.TMB5.DATA0 0749 028Ch to CANFDG0_NS.TMB5.DATA63 0749 02CBh,
CANFDG0_NS.TMB6.DATA0 0749 030Ch to CANFDG0_NS.TMB6.DATA63 0749 034Bh,
CANFDG0_NS.TMB7.DATA0 0749 038Ch to CANFDG0_NS.TMB7.DATA63 0749 03CBh,
CANFDG0_NS.TMB32.DATA0 0749 100Ch to CANFDG0_NS.TMB32.DATA63 0749 104Bh,
CANFDG0_NS.TMB33.DATA0 0749 108Ch to CANFDG0_NS.TMB33.DATA63 0749 10CBh,
CANFDG0_NS.TMB34.DATA0 0749 110Ch to CANFDG0_NS.TMB34.DATA63 0749 114Bh,
CANFDG0_NS.TMB35.DATA0 0749 118Ch to CANFDG0_NS.TMB35.DATA63 0749 11CBh,
CANFDG0_NS.TMB36.DATA0 0749 120Ch to CANFDG0_NS.TMB36.DATA63 0749 124Bh,
CANFDG0_NS.TMB37.DATA0 0749 128Ch to CANFDG0_NS.TMB37.DATA63 0749 12CBh,
CANFDG0_NS.TMB38.DATA0 0749 130Ch to CANFDG0_NS.TMB38.DATA63 0749 134Bh,
CANFDG0_NS.TMB39.DATA0 0749 138Ch to CANFDG0_NS.TMB39.DATA63 0749 13CBh,
CANFDG0_NS.TMB64.DATA0 0749 200Ch to CANFDG0_NS.TMB64.DATA63 0749 204Bh,
CANFDG0_NS.TMB65.DATA0 0749 208Ch to CANFDG0_NS.TMB65.DATA63 0749 20CBh,
CANFDG0_NS.TMB66.DATA0 0749 210Ch to CANFDG0_NS.TMB66.DATA63 0749 214Bh,
CANFDG0_NS.TMB67.DATA0 0749 218Ch to CANFDG0_NS.TMB67.DATA63 0749 21CBh,
CANFDG0_NS.TMB68.DATA0 0749 220Ch to CANFDG0_NS.TMB68.DATA63 0749 224Bh,
CANFDG0_NS.TMB69.DATA0 0749 228Ch to CANFDG0_NS.TMB69.DATA63 0749 22CBh,
CANFDG0_NS.TMB70.DATA0 0749 230Ch to CANFDG0_NS.TMB70.DATA63 0749 234Bh,
CANFDG0_NS.TMB71.DATA0 0749 238Ch to CANFDG0_NS.TMB71.DATA63 0749 23CBh,
CANFDG0_NS.TMB96.DATA0 0749 300Ch to CANFDG0_NS.TMB96.DATA63 0749 304Bh,
CANFDG0_NS.TMB97.DATA0 0749 308Ch to CANFDG0_NS.TMB97.DATA63 0749 30CBh,
CANFDG0_NS.TMB98.DATA0 0749 310Ch to CANFDG0_NS.TMB98.DATA63 0749 314Bh,
CANFDG0_NS.TMB99.DATA0 0749 318Ch to CANFDG0_NS.TMB99.DATA63 0749 31CBh,
CANFDG0_NS.TMB100.DATA0 0749 320Ch to CANFDG0_NS.TMB100.DATA63 0749 324Bh,
CANFDG0_NS.TMB101.DATA0 0749 328Ch to CANFDG0_NS.TMB101.DATA63 0749 32CBh,
CANFDG0_NS.TMB102.DATA0 0749 330Ch to CANFDG0_NS.TMB102.DATA63 0749 334Bh,
CANFDG0_NS.TMB103.DATA0 0749 338Ch to CANFDG0_NS.TMB103.DATA63 0749 33CBh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read/write registers for storing the data bytes of the message to be transmitted.
Do not write to these registers in CH_SLEEP mode.

45.2.80 ECC Control/Status Register (AFLRAM0.ECCSR, AFLRAM1.ECCSR, MBRAM.ECCSR)

Address(es): CANFDG0.AFLRAM0.ECCSR 8746 E000h,
CANFDG0.AFLRAM1.ECCSR 8746 E100h,
CANFDG0.MBRAM.ECCSR 8746 E200h,
CANFDG0_NS.AFLRAM0.ECCSR 0746 E000h,
CANFDG0_NS.AFLRAM1.ECCSR 0746 E100h,
CANFDG0_NS.MBRAM.ECCSR 0746 E200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EC2EAS	EC1EAS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECEDWC[1:0]		—	—	ECOVF	EC2EC	EC1EC	—	—	ECEDE	EC1ECD	EC2EIE	EC1EIE	EC2EF	EC1EF	ECEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECEF	ECC Error Flag	0: There is no ECC error in the last RAM data read. 1: There is an ECC error in the last RAM data read.	R
b1	EC1EF	1-Bit ECC Error Detection Flag	0: 1-bit ECC error is not detected. 1: 1-bit ECC error is detected.	R
b2	EC2EF	2-Bit ECC Error Detection Flag	0: 2-bit ECC error is not detected. 1: 2-bit ECC error is detected.	R
b3	EC1EIE	1-Bit ECC Error Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b4	EC2EIE	2-Bit ECC Error Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b5	EC1ECD	1-Bit ECC Error Correction Disable	0: Perform error correction when 1-bit ECC error is detected. 1: Does not perform error correction when 1-bit ECC error is detected.	R/W
b6	ECEDE	ECC Error Detection Enable	0: ECC error detection is disabled. 1: ECC error detection is enabled.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	EC1EC	EC1EF Flag Clear	Writing 1 to this bit clears the EC1EF flag. Writing 0 to this bit is ignored. This bit is read as 0.	R/W
b10	EC2EC	EC2EF Flag Clear	Writing 1 to this bit clears the EC2EF flag. Writing 0 to this bit is ignored. This bit is read as 0.	R/W
b11	ECOVF	ECC Overflow Detection Flag	0: The ECEAR register overflow has not occurred. 1: The ECEAR register overflow has occurred.	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECEDWC[1:0]	ECEDE Bit Write Control	Enables or disables write access to the ECEDE bit.	R/W
b16	EC1EAS	1-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 1-bit ECC error occurred is stored in the ECEAR register.	R
b17	EC2EAS	2-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 2-bit ECC error occurred is stored in the ECEAR register.	R
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ECEF Flag (ECC Error Flag)

The ECEF flag indicates whether there is an ECC error in the last RAM data read. This flag is updated every time the

RAM is read.
If the ECEDE bit is set to 1 without initializing the RAM, the value of this flag has no meaning.

[Setting condition]

- When there is an ECC error in read RAM data while the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- When there are no ECC error in read RAM data
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

EC1EF Flag (1-Bit ECC Error Detection Flag)

The EC1EF flag indicates that a 1-bit ECC error is detected in the data read from RAM.
When this flag is set to 1 while the interrupt is enabled, an ECC error interrupt (AFLRAM0EI, AFLRAM1EI, MBRAMEI) is generated.
If a 1-bit ECC error is detected again while this flag is 1, no further interrupt is generated.

[Setting condition]

- When a 1-bit ECC error is detected in the data read from RAM while the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If a 1-bit ECC error is detected at the same time that the EC1EC bit is set to 1, the EC1EF flag becomes 0.

EC2EF Flag (2-Bit ECC Error Detection Flag)

The EC2EF flag indicates that a 2-bit ECC error is detected in the data read from RAM.
When this flag is set to 1 while the interrupt is enabled, an ECC error interrupt (AFLRAM0EI, AFLRAM1EI, MBRAMEI) is generated.
If a 2-bit ECC error is detected again while this flag is 1, no further interrupt is generated.

[Setting condition]

- When a 2-bit ECC error is detected in the data read from RAM while the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If a 2-bit ECC error is detected at the same time that the EC2EC bit is set to 1, the EC2EF flag becomes 0.

EC1EIE Bit (1-Bit ECC Error Detection Interrupt Enable)

The EC1EIE bit enables or disables the interrupt generated by 1-bit ECC error detection.
When the EC1EF flag is set to 1 while this bit is 1, an ECC error interrupt (AFLRAM0EI, AFLRAM1EI, MBRAMEI) is generated.

EC2EIE Bit (2-Bit ECC Error Detection Interrupt Enable)

The EC2EIE bit enables or disables the interrupt generated by 2-bit ECC error detection.
When the EC2EF flag is set to 1 while this bit is 1, an ECC error interrupt (AFLRAM0EI, AFLRAM1EI, MBRAMEI) is generated.

EC1ECD Bit (1-Bit ECC Error Correction Disable)

The EC1ECD bit sets whether 1-bit ECC error correction is performed when the ECEDE bit is set to 1 (ECC error detection is enabled). When this bit is set to 1, the RAM output data is not corrected even if a 1-bit ECC error is detected.

ECEDE Bit (ECC Error Detection Enable)

Setting this bit to 1 enables the ECC error detection.
Writing to this bit is valid only when the ECEDWC[1:0] bits are set to 01b.

EC1EC Bit (EC1EF Flag Clear)

The EC1EC bit is used to clear the EC1EF flag.
The EC1EF flag is cleared by writing 1 to this bit while the EC1EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If a 1-bit ECC error is detected at the same time that the EC1EC bit is set to 1, the EC1EF flag becomes 0.

EC2EC Bit (EC2EF Flag Clear)

The EC2EC bit is used to clear the EC2EF flag.
The EC2EF flag is cleared by writing 1 to this bit while the EC2EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If a 2-bit ECC error is detected at the same time that the EC2EC bit is set to 1, the EC2EF flag becomes 0.

ECOVF Flag (ECC Overflow Detection Flag)

If a new ECC error is detected and the address is overwritten when an address is already stored in the ECEAR register, the ECOVF flag is set to 1 and an ECC error interrupt (AFLRAM0EI, AFLRAM1EI, MBRAMEI) is generated.
If a new ECC error is detected while this flag is 1, an ECC error interrupt is generated again.

[Setting condition]

- When new error address is captured under the condition that error address is already captured in the ECEAR register.

[Clearing condition]

- When the EC1EC bit is set to 1
- When the EC2EC bit is set to 1
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

When clearing the ECOVF flag by the EC1EC or EC2EC bit and setting the ECOVF flag occur at the same time, the ECOVF flag becomes 0.

ECEDWC[1:0] Bits (ECEDE Bit Write Control)

The ECEDWC[1:0] bits are used to enable or disable write access to the ECEDE bit. The read value is always 00b.
A value can be written to the ECEDE bit only when the value of these bits is 01b. If these bits are set to a value other than 01b, writing to the ECEDE bit is ignored and the value does not change.

EC1EAS Flag (1-Bit ECC Error Detected Address Stored Flag)

The EC1EAS flag indicates that the ECEAR register stores the address where the 1-bit ECC error occurred while the ECEDE bit is set to 1 (ECC error detection is enabled).
If a 1-bit ECC error is detected while an address where the 2-bit ECC error has occurred is already stored in the ECEAR register, the ECEAR register is not updated and this flag is not updated.

[Setting condition]

- When a 1-bit ECC error is detected while the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- When writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

When clearing the EC1EAS flag by the EC1EC bit and setting the EC1EAS flag occur at the same time, the EC1EAS

flag becomes 0.

EC2EAS Flag (2-Bit ECC Error Detected Address Stored Flag)

The EC2EAS flag indicates that the ECEAR register stores the address where the 2-bit ECC error occurred while the ECEDE bit is set to 1 (ECC error detection is enabled).
If a 2-bit ECC error is detected while an address where the 1-bit ECC error has occurred is already stored in the ECEAR register, the ECEAR register is updated and this flag becomes 1.

[Setting condition]

- When a 2-bit ECC error is detected while the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- When writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

When clearing the EC2EAS flag by the EC2EC bit and setting the EC2EAS flag occur at the same time, the EC2EAS flag becomes 0.

45.2.81 ECC Test Mode Register (AFLRAM0.ECTMR, AFLRAM1.ECTMR, MBRAM.ECTMR)

Address(es): CANFDG0.AFLRAM0.ECTMR 8746 E004h,
CANFDG0.AFLRAM1.ECTMR 8746 E104h,
CANFDG0.MBRAM.ECTMR 8746 E204h,
CANFDG0_NS.AFLRAM0.ECTMR 0746 E004h,
CANFDG0_NS.AFLRAM1.ECTMR 0746 E104h,
CANFDG0_NS.MBRAM.ECTMR 0746 E204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECTMWC[1:0]		—	—	—	—	—	—	ECTME	—	—	—	—	—	ECDIS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ECDIS	ECC Decoder Input Select	0: RAM output data is input to data input of decoder 1: Value of the ECTDR register is input to data input of decoder	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ECTME	ECC Test Mode Enable	0: Access to the ECDIS bit and the ECTDR register is disabled 1: Access to the ECDIS bit and the ECTDR register is enabled	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECTMWC[1:0]	ECTME Bit Write Control	Enable or disable write access to the ECTME bit.	R/W

ECDIS Bit (ECC Decoder Input Select)

The ECDIS bit selects whether the data value read from RAM or the value of the ECTDR register is to be used as input data to the ECC decoder.

Writing to this bit is valid only when the ECTME bit is set to 1. It is acceptable to write to this bit at the same time as setting the ECTME bit to 1.

Setting the ECTME bit to 0 also sets this bit to 0.

ECTME Bit (ECC Test Mode Enable)

The ECTME bit is used to enable or disable access to the ECDIS bit and the ECTDR register.

This bit can be written only when the ECTMWC[1:0] bits is set to 10b.

ECTMWC[1:0] Bits (ECTME Bit Write Control)

The ECTMWC[1:0] bits are used to enable or disable write access to the ECTME bit. The read value is always 00b.

A value can be written to the ECTME bit only when the value of these bits is set to 10b. If these bits are set to a value other than 10b, write access to the ECTME bit is ignored and the value does not change.

45.2.82 ECC Decoder Test Data Register (AFLRAM0.ECTDR, AFLRAM1.ECTDR, MBRAM.ECTDR)

Address(es): CANFDG0.AFLRAM0.ECTDR 8746 E00Ch,
CANFDG0.AFLRAM1.ECTDR 8746 E10Ch,
CANFDG0.MBRAM.ECTDR 8746 E20Ch,
CANFDG0_NS.AFLRAM0.ECTDR 0746 E00Ch,
CANFDG0_NS.AFLRAM1.ECTDR 0746 E10Ch,
CANFDG0_NS.MBRAM.ECTDR 0746 E20Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the data for testing the ECC decoder.
When the ECTMR.ECTME bit is set to 1, this register can be read and written.
When the ECTMR.ECTME bit is set to 0, the value of this register becomes 00000000h.
When the ECTMR.ECDIS bit is set to 1, the value set in this register is used as the input data of the ECC decoder instead of the data read from RAM.

45.2.83 ECC Error Address Register (AFLRAM0.ECEAR, AFLRAM1.ECEAR, MBRAM.ECEAR)

Address(es): CANFDG0.AFLRAM0.ECEAR 8746 E010h,
CANFDG0.AFLRAM1.ECEAR 8746 E110h,
CANFDG0.MBRAM.ECEAR 8746 E210h,
CANFDG0_NS.AFLRAM0.ECEAR 0746 E010h,
CANFDG0_NS.AFLRAM1.ECEAR 0746 E110h,
CANFDG0_NS.MBRAM.ECEAR 0746 E210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register holds the address where the ECC error occurred.
If an ECC error is detected when the ECCSR.ECEDE bit is 1 (ECC error detection is enabled), b12 to b2 of the RAM address are stored in b10 to b0 of this register.
If the same error occurred again, this register is not updated.
If a 2-bit ECC error is detected while the address where the 1-bit ECC error occurred is already stored, the ECEAR register is overwritten with the new address and the ECCSR.EC2EAS flag is set to 1.
If a 1-bit ECC error is detected while the address where the 2-bit ECC error occurred is already stored, the ECEAR register is not updated and the ECCSR.EC1EAS flag is not updated.

45.3 Operating Mode

The operating modes of the CANFDG module can be classified into two groups:

- Global modes
- Channel modes

45.3.1 Global Modes

These modes are applicable for the complete CANFDG module and therefore are called Global modes.
The Global modes of the CANFDG module are:

- GL_SLEEP
- GL_RESET
- GL_HALT
- GL_OPERATION

Figure 45.2 shows the possible transitions between the Global modes.

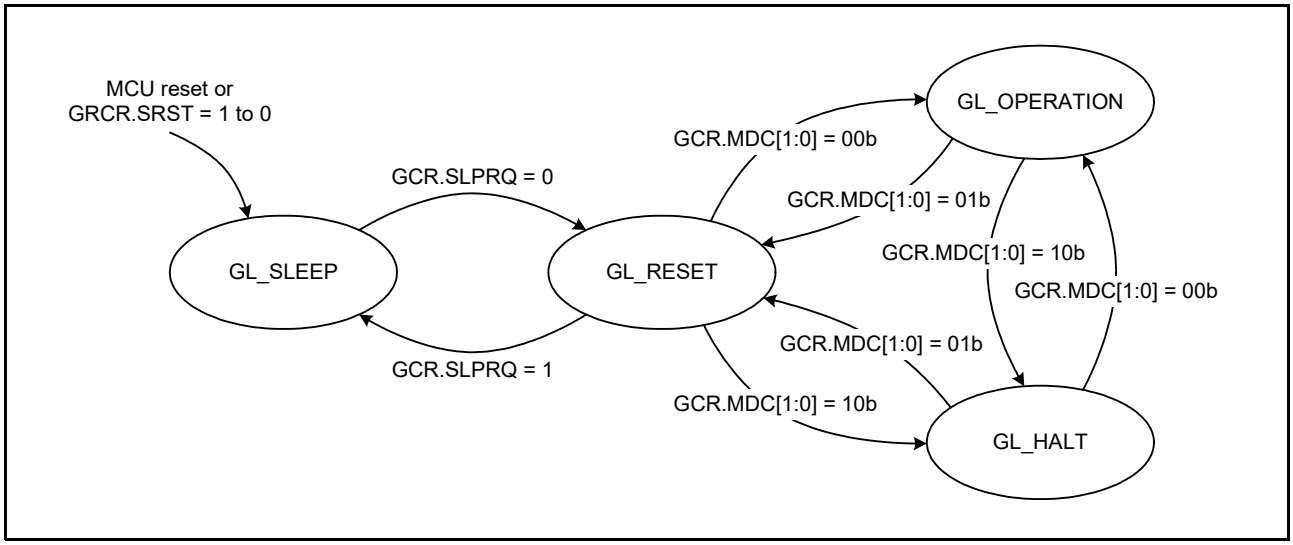


Figure 45.2 Transition between CANFDG Global Modes

Change in the Global mode can affect the Channel mode. For details, refer to section 45.3.3, Global Mode and Channel Mode Transition Interactions.

45.3.1.1 GL_SLEEP Mode

The CANFDG module automatically enters GL_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CANFDG module also enters the GL_SLEEP mode when the GCR.SLPRQ bit is set to 1 while it is in GL_RESET mode. The SLPRQ bit cannot be set to 1 in GL_HALT or GL_OPERATION mode.
Setting the GCR.SLPRQ bit to 1 sets all the CHCR.SLPRQ bits to 1 and forces all CAN channels into the CH_SLEEP mode.
GL_SLEEP mode is used for power saving purpose. When CANFDG module is in GL_SLEEP mode, only the clock for writing to the SLPRQ bit is active. All other clocks are stopped and all other functions of the CANFDG module are suspended.
Read access from all registers is still possible and all register values are preserved.
After setting the GCR.SLPRQ bit to 1, it is necessary to confirm the GSR.SLPST flag that the GL_SLEEP status has been updated, indicating successful transition to GL_SLEEP mode before the GCR.SLPRQ bit can be cleared again.

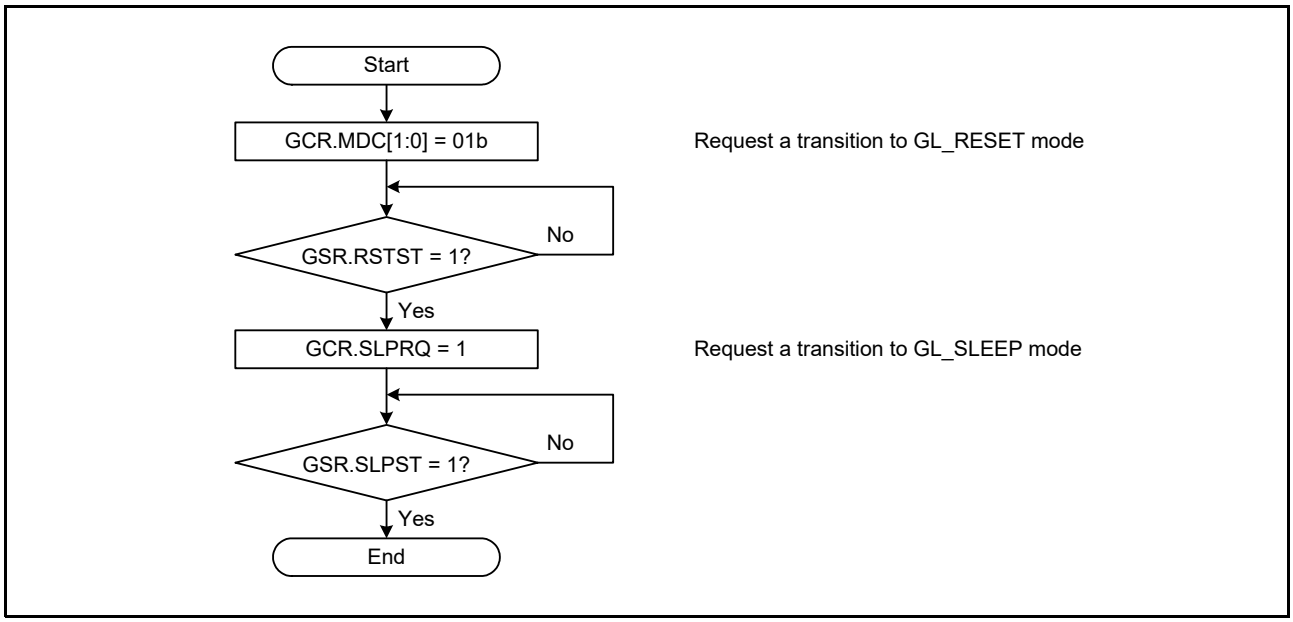


Figure 45.3 Procedure for Entering GL_SLEEP Mode

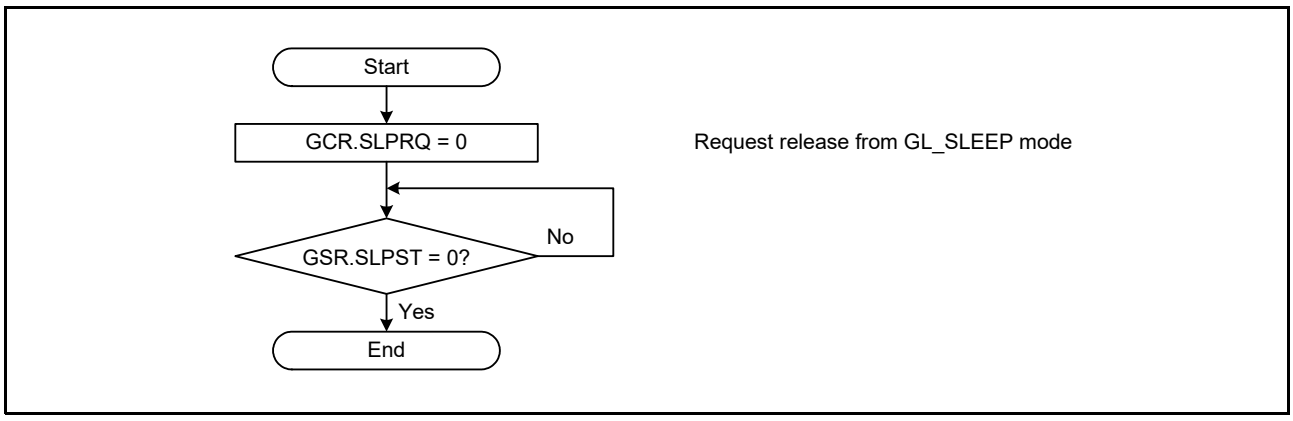


Figure 45.4 Procedure for Exiting GL_SLEEP Mode

45.3.1.2 GL_RESET Mode

The CANFDG module enters GL_RESET mode in the following ways:

- The GCR.MDC[1:0] bits are set to 01b while the CANFDG module is in GL_HALT or GL_OPERATION mode.
- The GCR.SLPRQ bit is set to 0 while the CANFDG module is in GL_SLEEP mode.

In GL_RESET mode, all CANFDG module functions are suspended and all status and flag registers are initialized. Additionally all FIFOs and all transmit queues are disabled and transmission control bits are cleared.
In this mode, configuration registers other than the GTMCR register and interrupt enable registers are not initialized, so the CANFDG module can be configured.
Setting the Global mode to GL_RESET by setting the GCR.MDC[1:0] bits to 01b sets the CHCR.MDC[1:0] bits to 01b and forces the channel into the CH_RESET mode.
For channels that are already in CH_RESET mode or CH_SLEEP mode, the channel mode remains unchanged.
After setting the GCR.MDC[1:0] bits to 01b (GL_RESET mode), it is necessary to confirm that the GSR.RSTST flag has been updated, indicating successful transition to GL_RESET mode before the GCR.MDC[1:0] bits can be changed again.

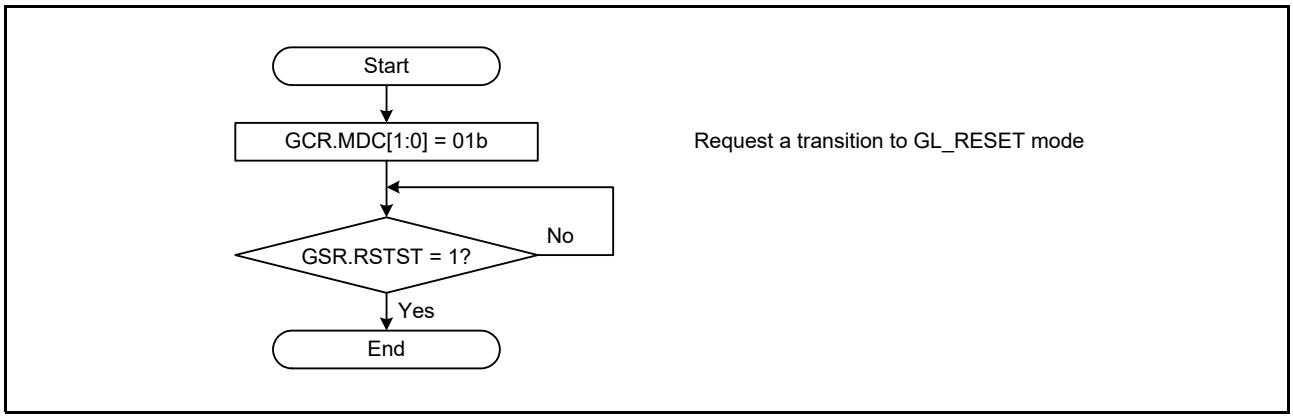


Figure 45.5 Procedure for Entering GL_RESET Mode

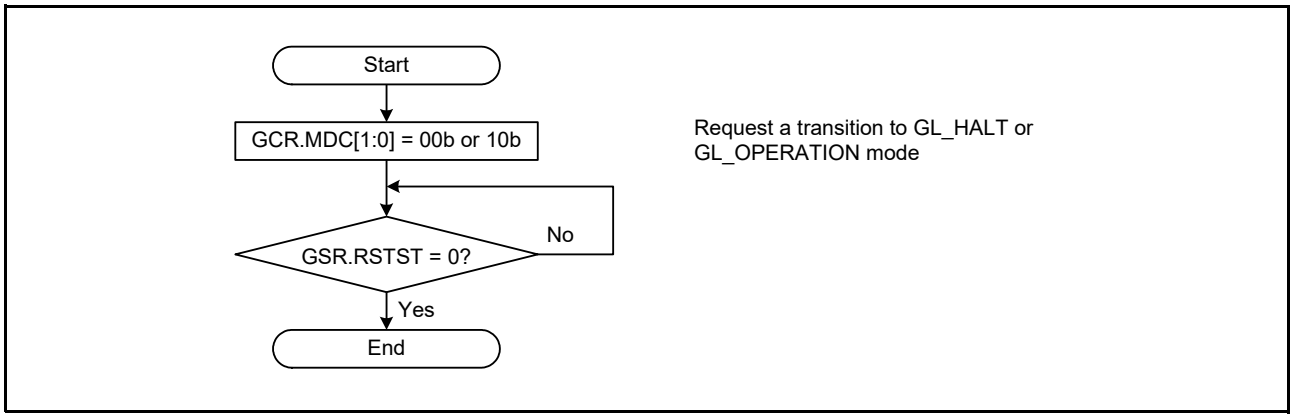


Figure 45.6 Procedure for Exiting GL_RESET Mode

45.3.1.3 GL_HALT Mode

The CANFDG module enters GL_HALT mode in the following ways:

- Set the GCR.MDC[1:0] bits to 10b while the CANFDG module is in GL_RESET mode:
  - The channels are in either CH_RESET or CH_SLEEP mode and remain in this mode.
- Set the GCR.MDC[1:0] bits to 10b while the CANFDG module is in GL_OPERATION mode:
  - All CAN channels in CH_RESET, CH_HALT, or CH_SLEEP mode remain in that mode.
  - All CAN channels in CH_OPERATION mode transit to CH_HALT mode.
  - The GSR.HLTST flag is set to 1 when all CAN channels exit CH_OPERATION mode.

If a transmission or reception is in progress for a CAN channel, the channel enters CH_HALT mode after waiting for completion of the communication.
Similarly, if a CAN channel is in the bus-off state, the channel may not enter CH_HALT mode until the bus-off recovery sequence is completed, depending on the channel configuration.
In GL_HALT mode, all communications are suspended and the status and flag registers do not change (only when a channel is in the bus-off state, its CHSR.REC[7:0] and TEC[7:0] bits are set to 00h).
Additionally, the GTMCR register and GTMER register are not initialized in this mode. The GL_HALT mode is used to configure Global Test Modes.
Setting the Global mode to GL_HALT by setting the GCR.MDC[1:0] bits to 10b sets the CHCR.MDC[1:0] bits to 10b for the channels that are in CH_OPERATION mode and forces these channels into the CH_HALT mode.
For channels that are already in CH_RESET, CH_HALT, or CH_SLEEP mode, the channel mode remains unchanged.
Therefore, the GL_HALT mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).
After setting the GCR.MDC[1:0] bits to 10b (GL_HALT mode), it is necessary to confirm that the GSR.HLTST flag has been updated to indicate a successful transition to GL_HALT mode. Do not set any other registers until confirming the GSR.HLTST flag is set.

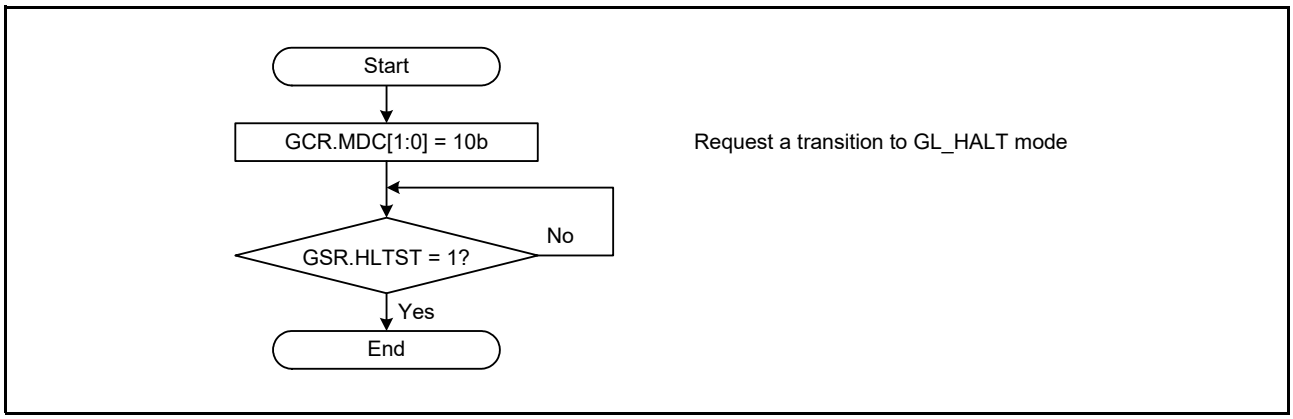


Figure 45.7 Procedure for Entering GL_HALT Mode

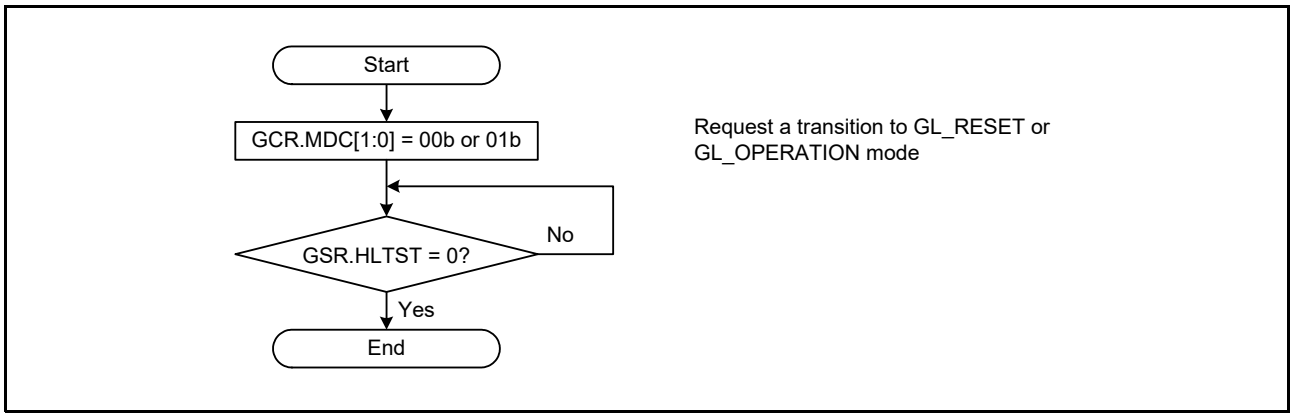


Figure 45.8 Procedure for Exiting GL_HALT Mode

45.3.1.4 GL_OPERATION Mode

The CANFDG module enters this mode when the GCR.MDC[1:0] bits are set to 00b.
The CAN channel can be set to CH_OPERATION mode and start CAN communication only when the CANFDG module is in GL_OPERATION mode.
After setting the GCR.MDC[1:0] bits to 00b (GL_OPERATION mode), it is necessary to confirm that the GSR.RSTST flag and the GSR.HLTST flag have been set to 0 to indicate a successful transition to GL_OPERATION mode before the GCR.MDC[1:0] bits can be modified again.

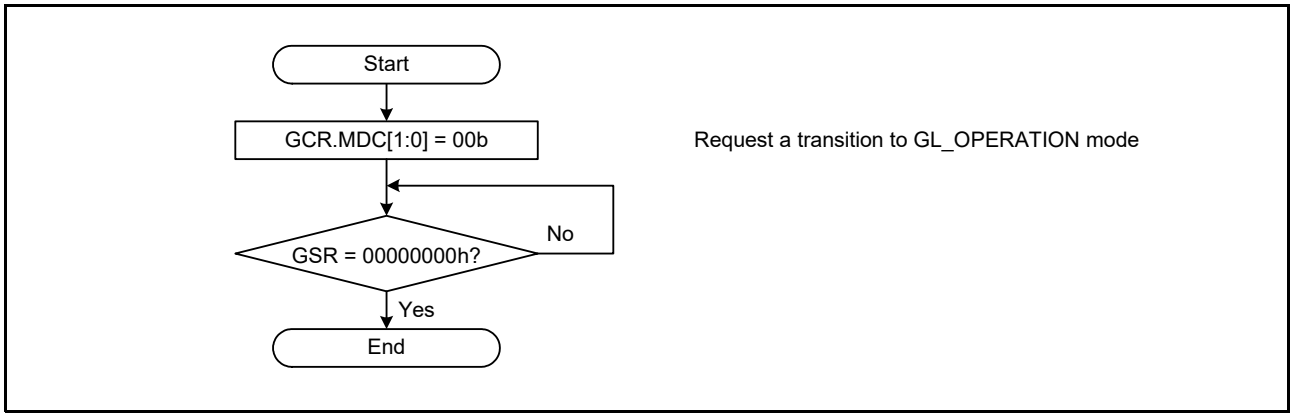


Figure 45.9 Procedure for Entering GL_OPERATION Mode

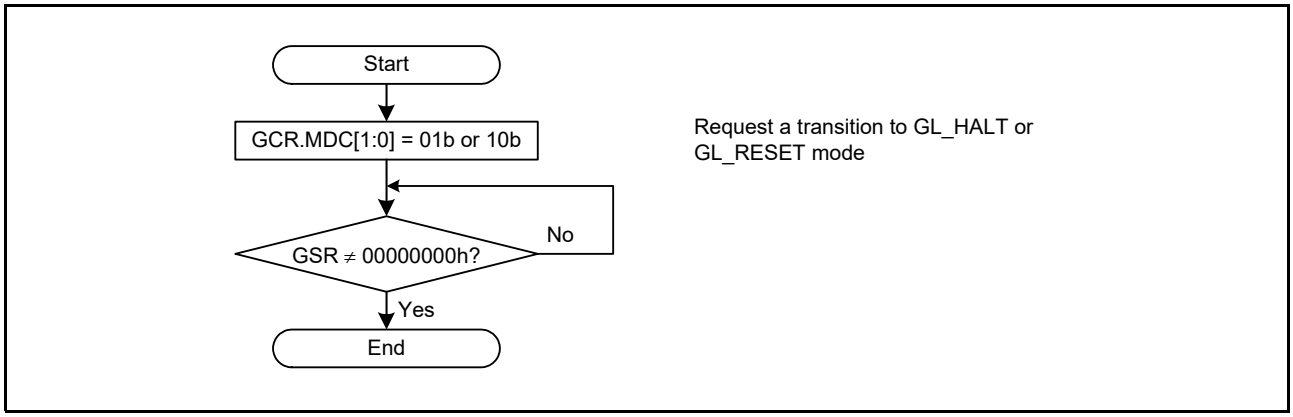


Figure 45.10 Procedure for Exiting GL_OPERATION Mode

45.3.2 Channel Modes

The Channel modes of the CANFDG module are:

- CH_SLEEP
- CH_RESET
- CH_HALT
- CH_OPERATION

Figure 45.11 shows the possible transitions between the channel modes.

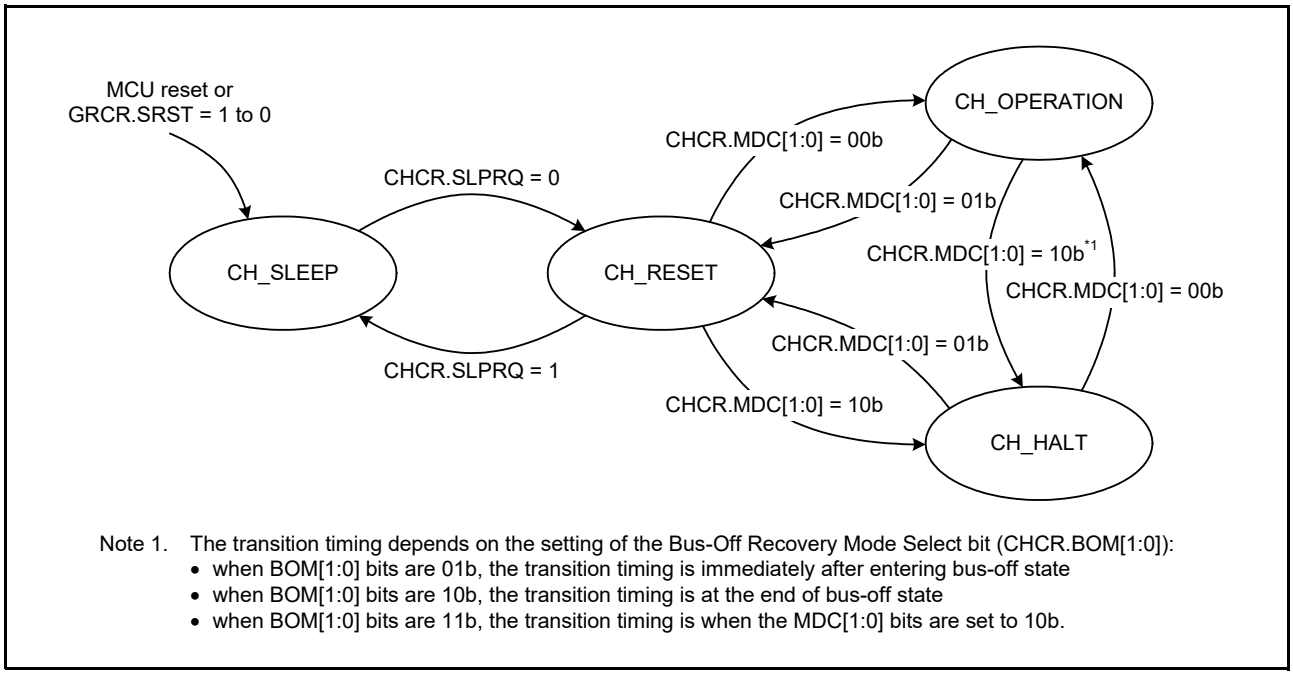


Figure 45.11 Transition between CAN Channel Modes

45.3.2.1 CH_SLEEP Mode

The CAN channel of the CANFDG module automatically enters CH_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.

The CAN channel also enters CH_SLEEP mode when the CHCR.SLPRQ bit is set to 1 while the CAN channel is in CH_RESET mode. The CHCR.SLPRQ bit cannot be set to 1 in CH_HALT or CH_OPERATION mode.

Entering CH_SLEEP mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the CHCR.SLPRQ bit to 1 and before setting it to 0 again, it is necessary to use the CHSR.SLPST flag to confirm that transition to CH_SLEEP mode was successful.

During CH_SLEEP mode, do not write to channel related registers. Read operation is still possible.

45.3.2.2 CH_RESET Mode

The CAN channel of CANFDG module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 01b while the CAN channel is in CH_HALT or CH_OPERATION mode
- The CHCR.SLPRQ bit is set to 0 while the CAN channel is in CH_SLEEP mode
- The GCR.MDC[1:0] bits are set to 01b while the CAN channel is not in CH_SLEEP or CH_RESET mode.

In CH_RESET mode, all CAN channel status and flags are initialized.
In addition, all transmission-related control bits of the channel are cleared, and the transmit queue of the channel is also disabled.
In this mode, the configuration registers except for the bits related to the channel test mode are not initialized, so the CAN channel can be configured for communication.
After setting the CHCR.MDC[1:0] bits to 01b (CH_RESET mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag to confirm that transition to CH_RESET mode was successful.
Refer to Table 45.14 for the behavior of transitioning to CH_RESET mode while CAN communication is in progress.

45.3.2.3 CH_HALT Mode

The CAN channel of CANFDG module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_RESET or CH_OPERATION mode
- The GCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_OPERATION mode.

In CH_HALT mode, all CAN communication of the channel is suspended, but all statuses and flags remain unchanged during CH_HALT mode (except for the bus-off state where the CHSR.REC[7:0] and TEC[7:0] bits for the channel are set to 00h).
In addition, in this mode, the bits related to the channel test mode are not initialized. Use CH_HALT mode to configure the channel test mode.
After setting the CHCR.MDC[1:0] bits to 10b (CH_HALT mode) and before modifying these bits again, it is necessary to use the CHSR.HLTST flag to confirm that transition to CH_HALT mode was successful.
Refer to Table 45.14 for the transition behavior to CH_HALT mode while CAN communication is in progress.

Table 45.14 Transition Behavior in CH_RESET Mode and CH_HALT Mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CH_RESET mode (CHCR.MDC[1:0] = 01b)	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the bus-off recovery.
CH_HALT mode (CHCR.MDC[1:0] = 10b)	CAN channel transits to CH_HALT mode at the end of the ongoing reception or when an error occurs.*2	CAN channel transits to CH_HALT mode after completion of the ongoing transmission.	When the CHCR.BOM[1:0] bits are 00b, a CH_HALT mode request is accepted only after the completion of the full bus-off recovery sequence. When the CHCR.BOM[1:0] bits are 10b, the CAN channel transits automatically to CH_HALT mode after waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 01b, the CAN channel transits automatically to CH_HALT mode without waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 11b, the CAN channel transits to CH_HALT mode as soon as CH_HALT mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to CH_RESET mode is required only at the end of an ongoing communication, then CH_HALT mode can be requested first to prevent interruption of CAN communication by direct transition to CH_RESET mode. After the CAN channel enters CH_HALT mode, the CH_RESET mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to CH_RESET mode.

45.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State)

The CH_OPERATION mode is activated by setting the CHCR.MDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering the CH_OPERATION mode, the CHSR.CRDY flag is set to 1 and the CAN channel:

- Enables the functions of the communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters.

At this point, the CAN channel can start transmission and reception of messages.
Within the CH_OPERATION mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (refer to Figure 45.12):

- Idle mode: The CAN channel is neither receiving nor transmitting
- Receive mode: The channel is receiving a message transmitted by another CAN node
- Transmit mode: The channel is transmitting a message
  (The channel may receive its own message simultaneously when Self Test mode is enabled.)
- Bus-off mode: The CAN channel is cut-off from CAN bus communication.

After setting the CHCR.MDC[1:0] bits to 00b (CH_OPERATION mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag and the CHSR.HLTST flag to confirm that transition to CH_OPERATION mode was successful.

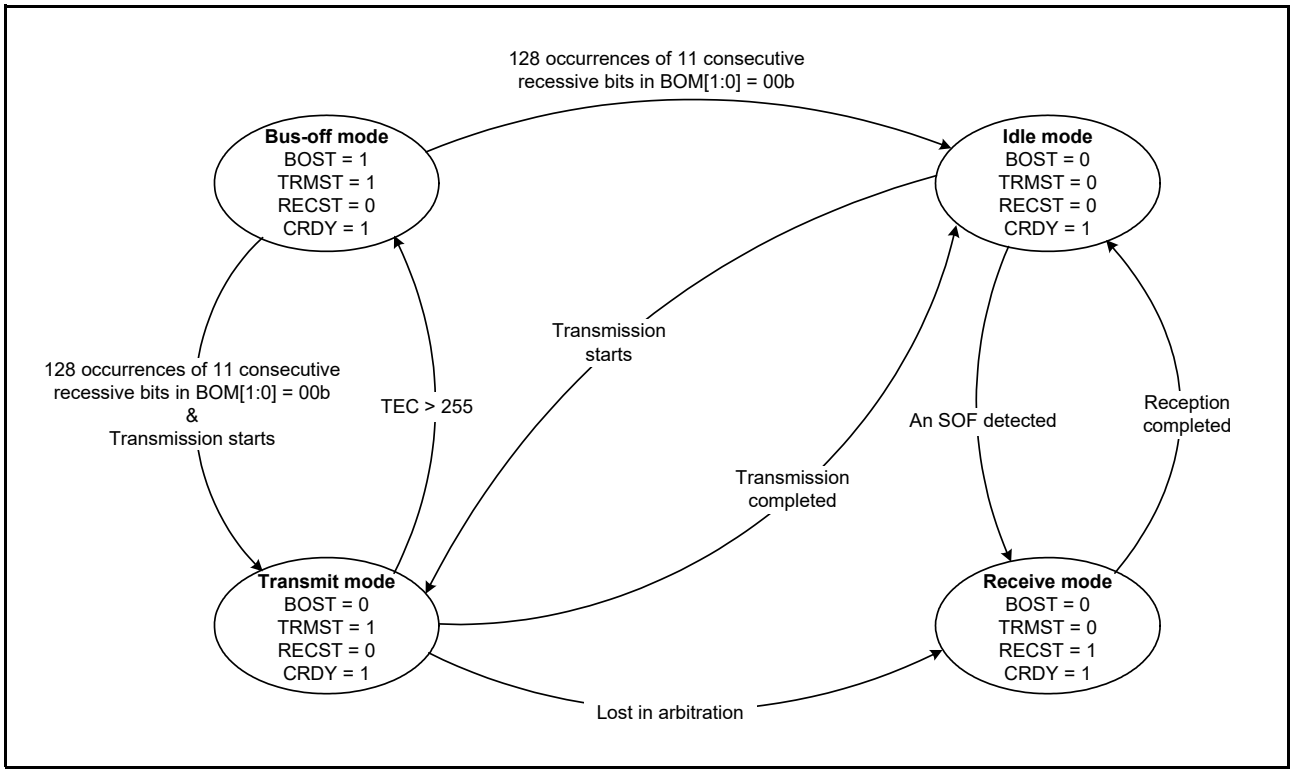


Figure 45.12 Sub-modes of CH_OPERATION Mode (only when CHCR.BOM[1:0] = 00b)

45.3.2.5 CH_OPERATION Mode (in Bus-Off State)

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CH_OPERATION mode from the bus-off state:

- CHCR.BOM[1:0] = 00b:
  Bus-off recovery is compliant to ISO 11898-1. The CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h, and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 01b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1 in this case.
- CHCR.BOM[1:0] = 10b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter CH_HALT mode when still in bus-off state if a request is made to enter CH_HALT mode. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1.
  Without setting CHCR.MDC[1:0] = 10b and when 11 consecutive recessive bits are detected 128 times, transition conditions become the same as CHCR.BOM[1:0] = 00b.

If the recovery from bus-off occurs normally in this mode (after waiting for 128 occurrences of 11 consecutive recessive bits), and no CH_HALT request has been generated during this period, then the CHESR.BORDF flag is set to 1.
When software writes to the CHCR.MDC[1:0] bits at the same time as the CAN channel enters CH_HALT mode (at the start of bus-off when CHCR.BOM[1:0] = 01b, or at the end of bus-off when CHCR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CHCR.MDC[1:0] bits to CH_HALT mode request is performed when the CHCR.MDC[1:0] bits value is previously 00b (CH_OPERATION mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting the CHCR.RTBO bit to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The CHESR.BORDF flag is not set to 1 in this case, and the CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h.
Before setting the CHCR.RTBO bit to 1, all pending transmissions from the transmit message buffers, transmit queues and/or common FIFO in transmit mode or gateway mode should be disabled.
The disable of the pending transmit message buffer, transmit queue or common FIFO must be confirmed by the corresponding acknowledge flags (TMSRn.TXRF[1:0] flags, TQSRn.EMPTY flag, and CFSRn.EMPTY flag).
The CHCR.RTBO bit should be used for bus-off recovery only when the CHCR.BOM[1:0] bits are set to 00b.
Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.
Table 45.15 lists the behavior of the bus-off entry detect flag (CHESR.BOEDF) and the bus-off recovery detect flag (CHESR.BORDF) according to the CHCR.BOM[1:0] bit settings.

Table 45.15 Behavior of Bus-off Entry and Recovery Flags

CHCR.BOM[1:0] Bits	CHESR.BOEDF Flag	CHESR.BORDF Flag
00b	Becomes 1 on entry to bus-off state.	Becomes 1 on exit from bus-off state.
00b Set the CHCR.RTBO bit to 1		Becomes 1 only if normal bus-off recovery occurs before setting the CHCR.RTBO bit to 1.
01b		Does not become 1.
10b		Becomes 1 on exit from bus-off state.
11b		Becomes 1 only if normal bus-off recovery occurs before requesting a transition to CH_HALT mode.

To make efficient software, it is not necessary to wait for the bus-off recovery sequence to end.
It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 45.13.

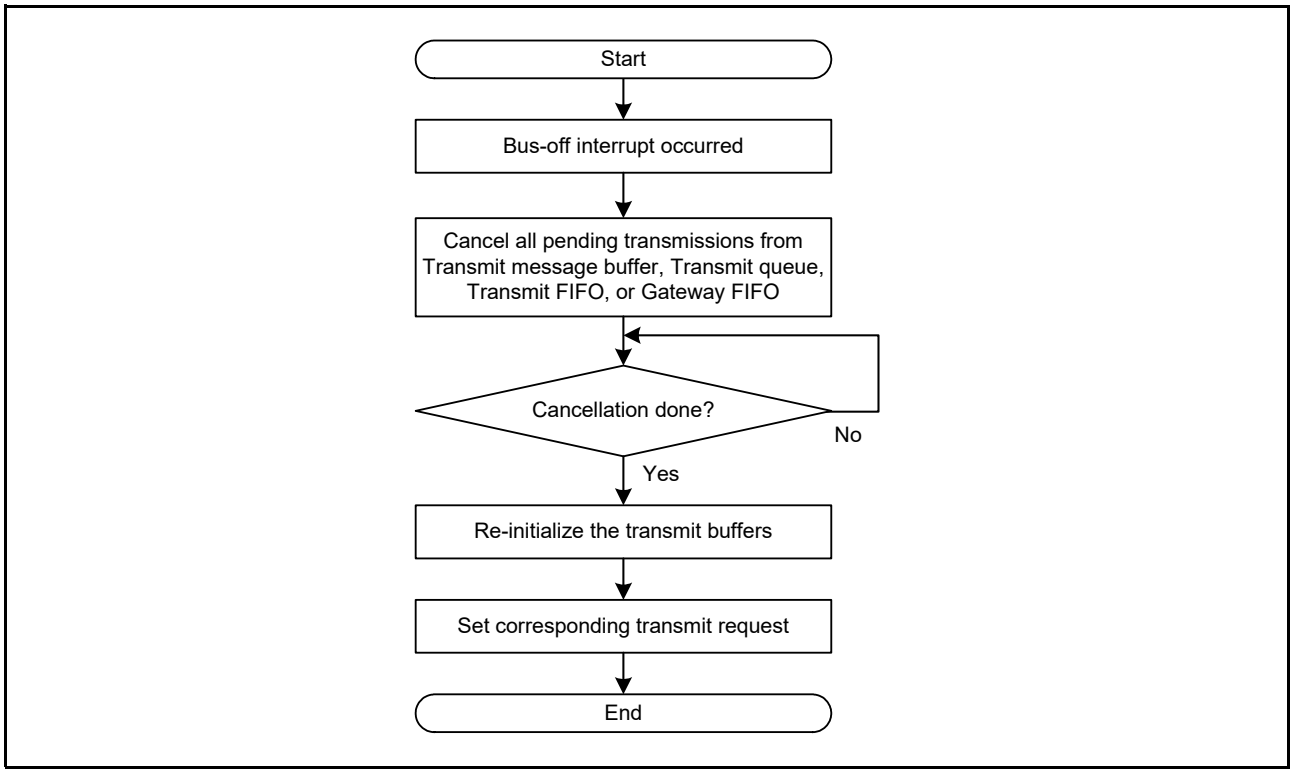


Figure 45.13 Transmission Re-Initialization During Bus-Off

45.3.3 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the CHCR.MDC[1:0] bits does not affect the GCR.MDC[1:0] bits.
- Changing the GCR.MDC[1:0] bits affects the channel mode control as described in Table 45.16.

Table 45.16 Interaction between Global Mode and Channel Mode Transition

Global Mode Change	Channel Mode before Changing Global Mode	Channel Mode after Changing Global Mode
GL_SLEEP → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
GL_RESET → GL_SLEEP	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_SLEEP
GL_RESET → GL_HALT	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
GL_RESET → GL_OPERATION	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
GL_HALT → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_RESET
GL_HALT → GL_OPERATION	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_HALT (no changes)
GL_OPERATION → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_RESET
	CH_OPERATION	CH_RESET
GL_OPERATION → GL_HALT	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_HALT (no changes)
	CH_OPERATION	After communication ends, CH_HALT

45.3.3.1 Transition Time for Global Mode Change

Table 45.17 shows the transition times when changing the global mode.

Table 45.17 Maximum Transition Times for Global Mode Changes

From	To	Maximum Transition Time
GL_SLEEP	GL_RESET	3 × PCLKA*1
GL_RESET	GL_SLEEP	3 × PCLKA
GL_RESET	GL_HALT	10 × PCLKA
GL_RESET	GL_OPERATION	10 × PCLKA
GL_HALT	GL_RESET	2 bit times (1 Tq + 16 × PCLKA + 2 DLL clock cycles)
GL_HALT	GL_OPERATION	3 × PCLKA
GL_OPERATION	GL_RESET	2 bit times (1 Tq + 16 × PCLKA + 2 DLL clock cycles)
GL_OPERATION	GL_HALT	3 CAN frames (1 CAN frame + 3424 × PCLKA)*2, *3

Note 1. Exit GL_SLEEP mode only when the GSR.RAMST flag is cleared.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.
Note 3. Tq, CAN frame and bit times vary for each channel. The maximum transition time must be calculated on the channel with the lowest bit rate.

45.3.3.2 Transition Time for Channel Mode Change

Table 45.18 shows the transition times when changing the channel mode.

Table 45.18 Maximum Transition Times for Channel Mode Changes

From	To	Maximum Transition Time
CH_SLEEP	CH_RESET	3 × PCLKA
CH_RESET	CH_SLEEP	3 × PCLKA
CH_RESET	CH_HALT	3 bit times (1 CAN bit + 2 Tq + 8 × PCLKA + 2 DLL clock cycles)
CH_RESET	CH_OPERATION	4 bit times (2 CAN bits + 1 TSEG1 + 12 × PCLKA + 2 DLL clock cycles)
CH_HALT	CH_RESET	2 bit times (1 Tq + 10 × PCLKA + 2 DLL clock cycles)
CH_HALT	CH_OPERATION	4 bit times (< 4 CAN bits)*1
CH_OPERATION	CH_RESET	2 bit times (1 Tq + 10 × PCLKA + 2 DLL clock cycles)
CH_OPERATION	CH_HALT	2 CAN frames (1 CAN frame + 13 CAN bits)*2, *3

Note 1. In general, if the bit rate prescaler value (NBCR.BRP[9:0]) is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the Tq clock, and new BRP value is captured when the counter reaches the value 0.

Note 2. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CHCR.BOM[1:0] bits.

Note 3. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.

45.4 Initialization of CANFDG Module

Before starting CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data bit rate)
- Bit Rate setting (nominal and data bit rate)
- CANFDG setting
- Acceptance Filter setting (configuration of Acceptance Filter List)
- Receive FIFO, Transmit FIFO, and Gateway FIFO setting
- CAN operating mode setting.

45.4.1 Initialization of CAN Clock, Bit Timing, and Bit Rate

45.4.1.1 Bit Timing Conditions

The following lines describe the settings for each segment and the limitations that apply to those settings.

1. Each segment setting

- SS = Fixed to 1 Tq
  TSEG1 = 2 Tq to 256 Tq (NBCR), 2 Tq to 32 Tq (DBCR)
  TSEG2 = 2 Tq to 128 Tq (NBCR), 2 Tq to 16 Tq (DBCR)
  SJW = 1 Tq to 128 Tq (NBCR), 1 Tq to 16 Tq (DBCR)
  SS + TSEG1 + TSEG2 = 8 Tq to 385 Tq (NBCR), 5 Tq to 49 Tq (DBCR)

2. Limitations on TSEG1, TSEG2 and SJW
   NBCR register: TSEG1 > TSEG2 ≥ SJW
   DBCR register: TSEG1 ≥ TSEG2 ≥ SJW

Table 45.19 shows an example of how to set the bit timing to achieve the required sample point settings.

Table 45.19 Bit Timing Examples

1 Bit	Set Value (Tq)				Sample Point (%)
	SS	TSEG1	TSEG2	SJW	
5 Tq	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66
	1	16	7	1	70.83
50 Tq	1	39	10	4	80.00

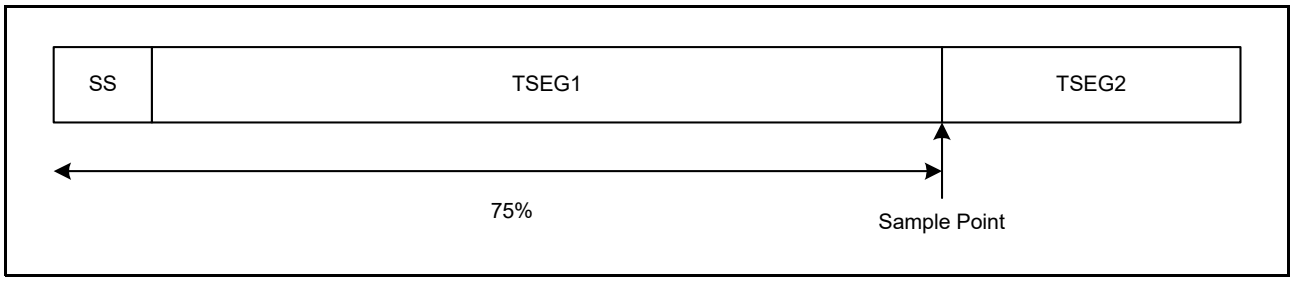


Figure 45.14 The Sample Point (in Case of 75%)

45.4.1.2 Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for each channel using the NBCR and DBCR registers.

Figure 45.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (Tq), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the bit rate prescaler (nominal and data bit rate).

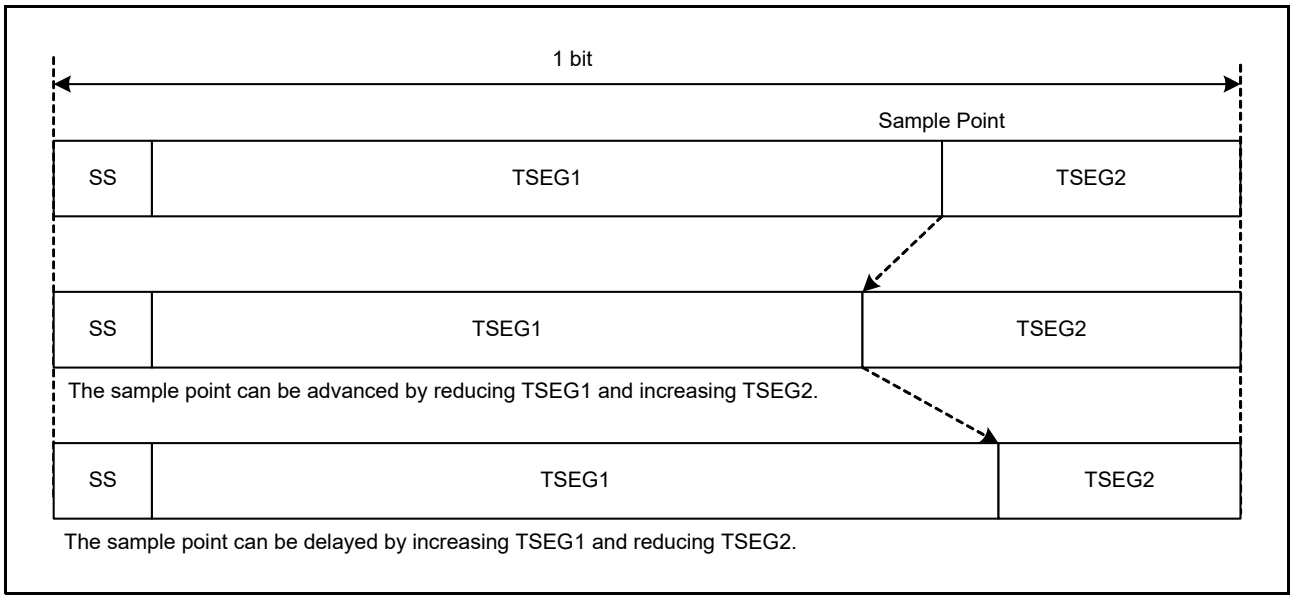


Figure 45.15 Segment Composition of A Bit and The Sample Point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. The interframe space consists of the intermission, suspend transmission, and bus idle fields; during the bus idle field, all nodes may begin transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.

3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While transmitting or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 45.15 shows an example of a typical sample point.

45.4.1.3 Bit Rate

The CAN communication clocks are generated by dividing the operating clock for data link layer (DLL clock). The DLL clock can be selected from either the internal clock (CANFDCLK) or the external clock (CANFDMCLK).
Figure 45.16 shows a block diagram of the circuit that generates the CAN communication clock.

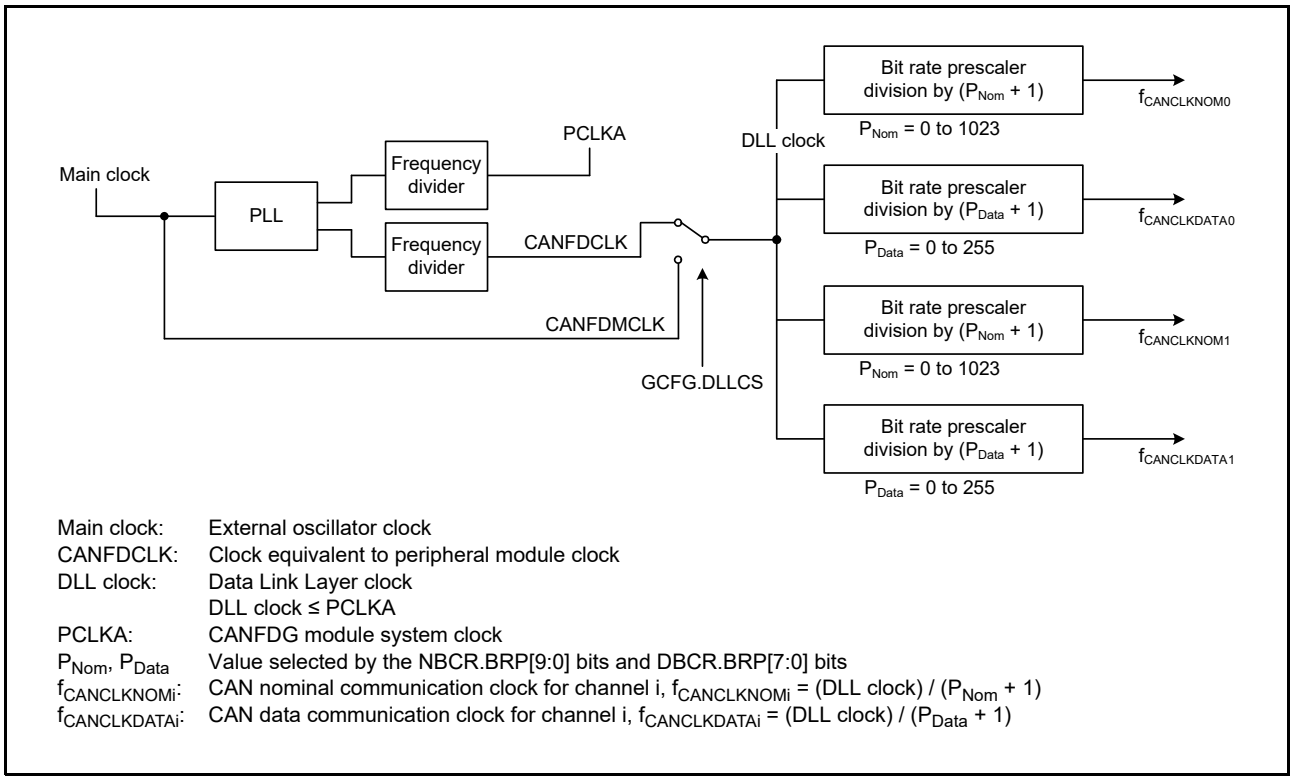


Figure 45.16 Block Diagram of the Circuit that Generates the CAN Communication Clock

The bit rate is determined by the DLL clock frequency, the division value of the bit rate prescaler (P + 1), and the number of Tq per bit.

$$\text{Bit rate} = \frac{\text{DLL clock frequency}}{\text{Number of Tq per bit} \times (\text{P} + 1)} = \frac{\text{CAN communication clock frequency}}{\text{Number of Tq per bit}}$$

Table 45.20 lists examples of setting the nominal bit rate for Classical CAN frame.

Table 45.20 Nominal Bit Rate Setting Examples for Classical CAN Frame

Bit Rate	DLL Clock Frequency																	
	80 MHz		40 MHz		32 MHz		30 MHz		24 MHz		20 MHz		16 MHz		10 MHz		8 MHz*1	
	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1
1 Mbps	8 Tq 20 Tq	10 4	8 Tq 20 Tq	5 2	8 Tq 16 Tq	4 2	10 Tq 15 Tq	3 2	8 Tq 12 Tq 24 Tq	3 2 1	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1	10 Tq	1	8 Tq	1
500 kbps	8 Tq 20 Tq	20 8	8 Tq 20 Tq	10 4	8 Tq 16 Tq	8 4	10 Tq 15 Tq 20 Tq	6 4 3	8 Tq 12 Tq 24 Tq	6 4 2	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1
250 kbps	8 Tq 20 Tq	40 16	8 Tq 20 Tq	20 8	8 Tq 16 Tq	16 8	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 12 Tq 24 Tq	12 8 4	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2
125 kbps	8 Tq 20 Tq	80 32	8 Tq 20 Tq	40 16	8 Tq 16 Tq	32 16	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 12 Tq 24 Tq	24 16 8	10 Tq 20 Tq	16 8	8 Tq 16 Tq	16 8	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4
83.3 kbps	8 Tq 12 Tq 16 Tq 24 Tq	120 80 60 40	8 Tq 12 Tq 16 Tq 24 Tq	60 40 30 20	8 Tq 12 Tq 16 Tq 24 Tq	48 32 24 16	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	45 36 30 24 18 15	8 Tq 12 Tq 16 Tq 24 Tq	36 24 18 12	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	30 24 20 16 15 12 10	8 Tq 12 Tq 16 Tq 24 Tq	24 16 12 8	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	15 12 10 8 6 5	8 Tq	12
33.3 kbps	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	300 200 150 120 100	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	150 100 75 60 50	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	120 96 80 64 60 48 40	10 Tq 12 Tq 15 Tq 20 Tq	90 75 60 45	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	90 72 60 48 45 36 30	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	75 60 50 40 30 25	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	60 48 40 32 30 24 20	10 Tq 12 Tq 15 Tq 20 Tq	30 25 20 15	8 Tq	30

Note 1. Minimum frequency to achieve a nominal bit rate of 1 Mbps.

For optimum clock tolerance in networks using the CAN FD frame, the length of the time quantum should be the same in nominal bit time and in data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
In addition, do not set the DBCR.BRP[7:0] bits greater than 1 when using transceiver delay compensation.
Table 45.21 lists examples of setting the nominal and data bit rate for CAN FD frame.

Table 45.21 Nominal and Data Bit Rate Setting Examples for CAN FD Frame

Bit Rate		DLL Clock Frequency								
		80 MHz			40 MHz			20 MHz		
		No. of Tq			No. of Tq			No. of Tq		
Nominal	Data	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1
1 Mbps	8 Mbps	80 Tq	10 Tq	1	40 Tq	5 Tq	1	20 Tq	—	1
1 Mbps	5 Mbps	80 Tq	16 Tq	1	40 Tq	8 Tq	1	20 Tq	—	1
500 kbps	2 Mbps	160 Tq	40 Tq	1	80 Tq	20 Tq	1	40 Tq	10 Tq	1

45.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate

Figure 45.17 shows the procedure for setting the bit timing and the bit rate.
These settings should be configured when the CAN channel is in CH_RESET mode.
The bit rate must be configured before moving to the channel communication state. Otherwise the mode does not switch correctly.

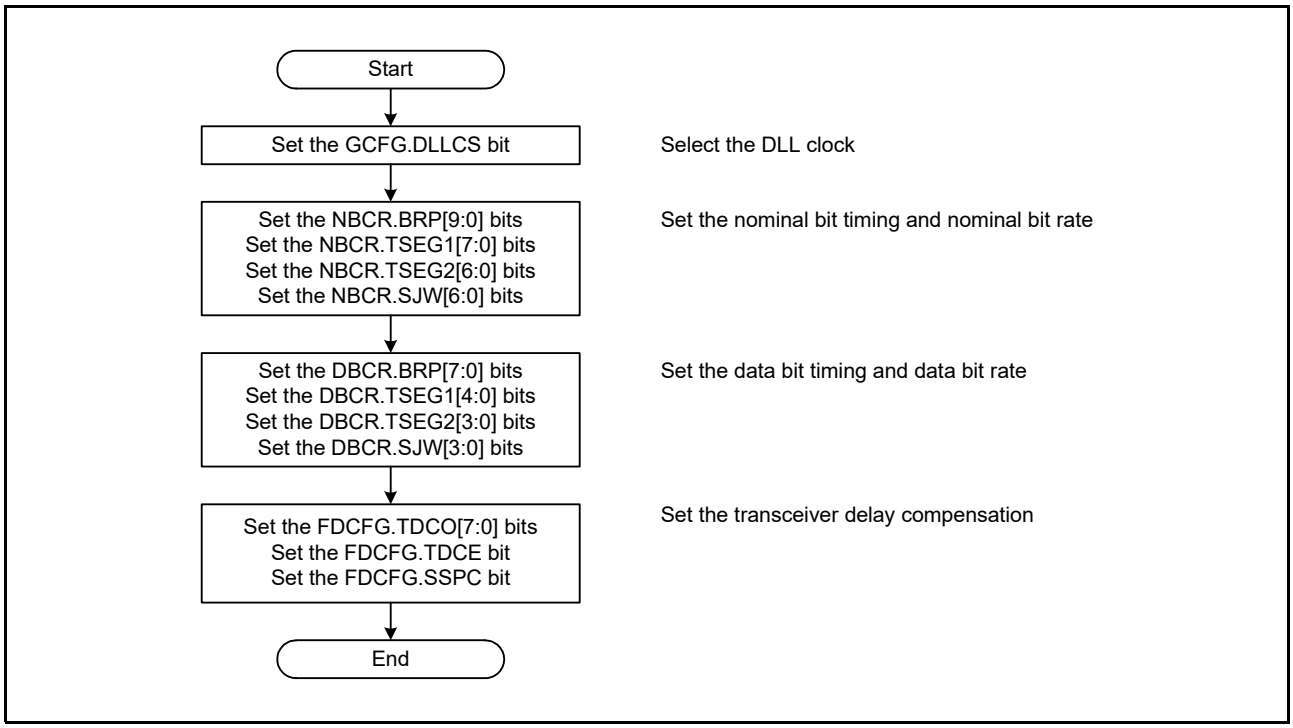


Figure 45.17 Procedure for Setting the Bit Timing and Bit Rate

45.4.1.5 Transceiver Delay Compensation

When a high bit rate is used such as 5 to 8 Mbps for the data phase, the transceiver delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN FD frame. The transceiver delay compensation function compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN FD frames. This is derived from the Transceiver Delay Compensation Result bit (FDSTS.TDCR[7:0]) as shown in Figure 45.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

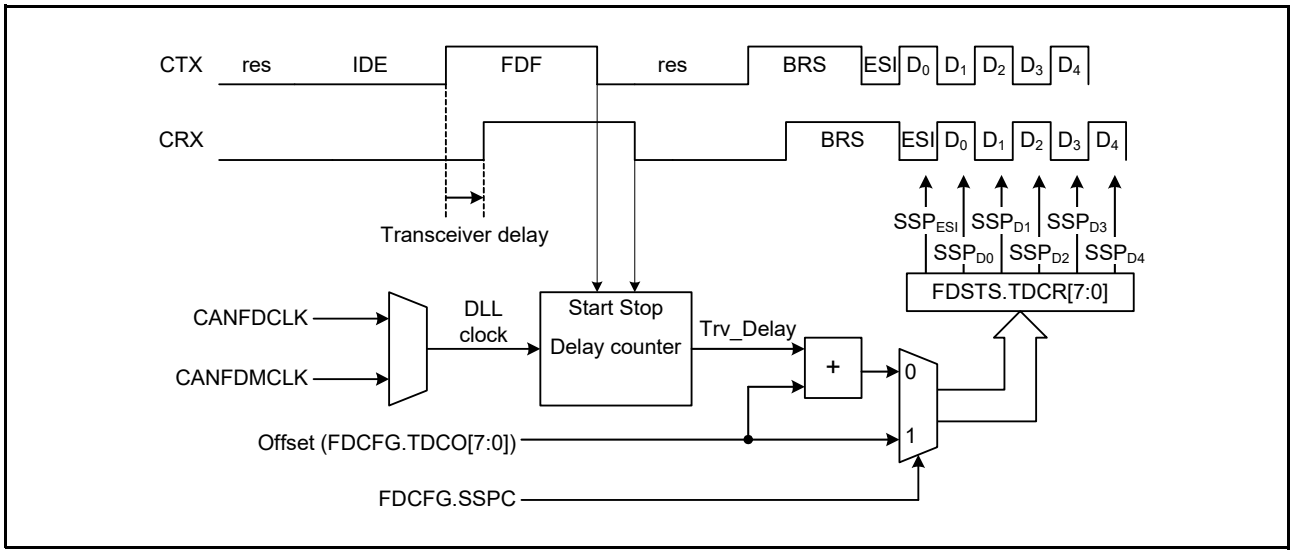


Figure 45.18 Transceiver Delay Compensation

The measured Trv_Delay is based on the number of DLL clock cycles. It counts up by one for each start clock until the dominant value can be observed at the CRX pin. Figure 45.19 shows the measurement example. Trv_Delay is counted up to 127 on each DLL clock.

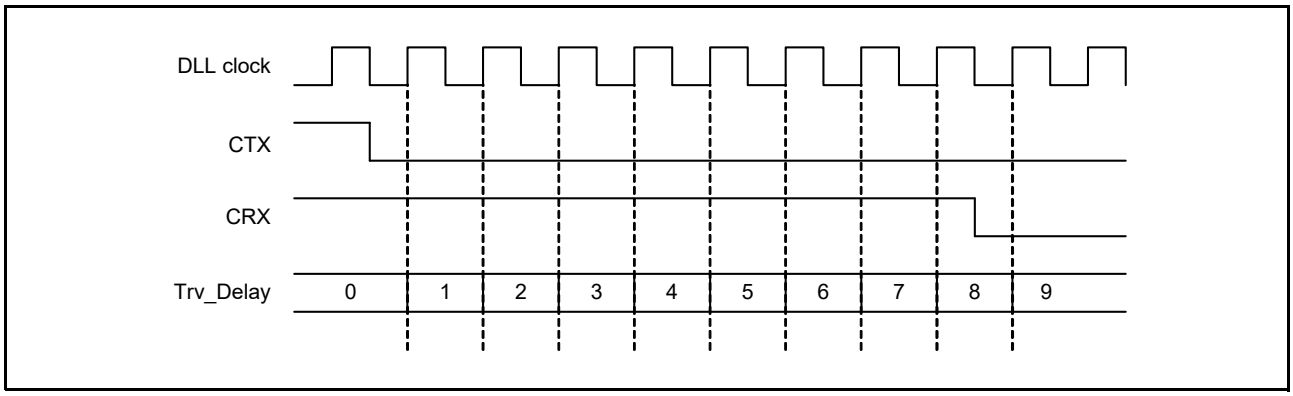


Figure 45.19 Trv_Delay Measurement Example

The SSP is calculated by taking the result from the FDSTS.TDCR[7:0] bits and rounding the value down to the nearest integer number of data Tq.

Figure 45.20 shows the positioning of the secondary sample point (SSP). When the FDCFG.SSPC bit is set to 0, the SSP is equal to the Trv_Delay (measured delay) + FDCFG.TDCO[7:0], rounded down to the nearest integer number of Tq. Normally, the TDCO[7:0] value has the magnitude of SS + TSEG1 in the data phase to position the SSP to a theoretical location of the sample point.

If the FDCFG.SSPC bit is set to 1, the SSP is defined by the FDCFG.TDCO[7:0] bits. If the DBCR.BRP[7:0] bits are greater than 00h, the value is also rounded down to the nearest integer number of Tq.

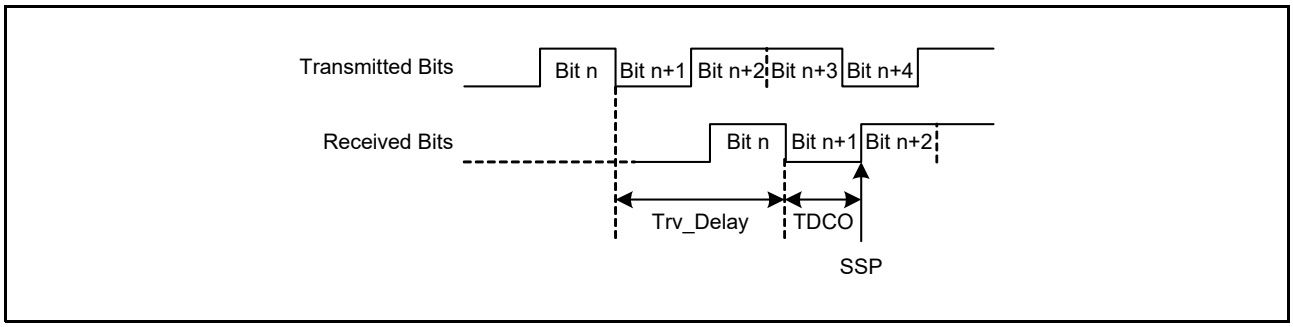


Figure 45.20 Positioning of the Secondary Sample Point (SSP)

The maximum delay time (Trv_Delay + TDCO[7:0]) which can be compensated by the CANFDG module is (6 data bit times – 2 DLL clocks). The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.
If different values are used for the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 45.21.
The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
The bit rate can be changed by selecting different settings for the time segments. The nominal bit rate can be set from 8 to 385 Tqs and the data bit rate from 5 to 49 Tqs.

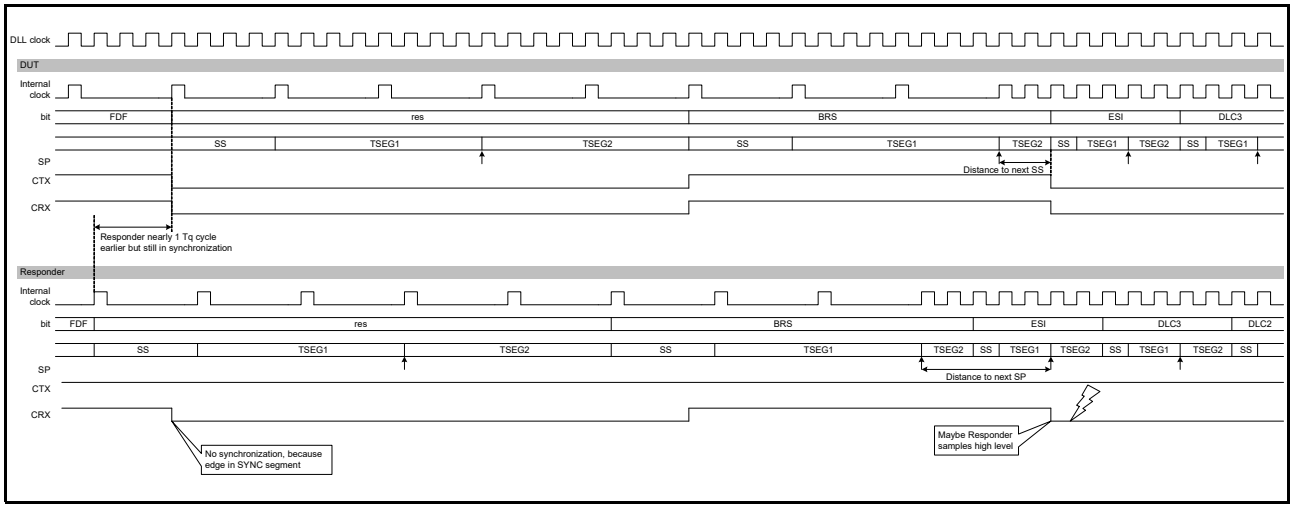


Figure 45.21 Loss of Synchronization Between Two CAN Nodes

The transceiver delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (FDCFG.TDCE = 1, FDCFG.SSPC = 0).
Figure 45.22 shows the read flow to get the measured transceiver delay compensation result.

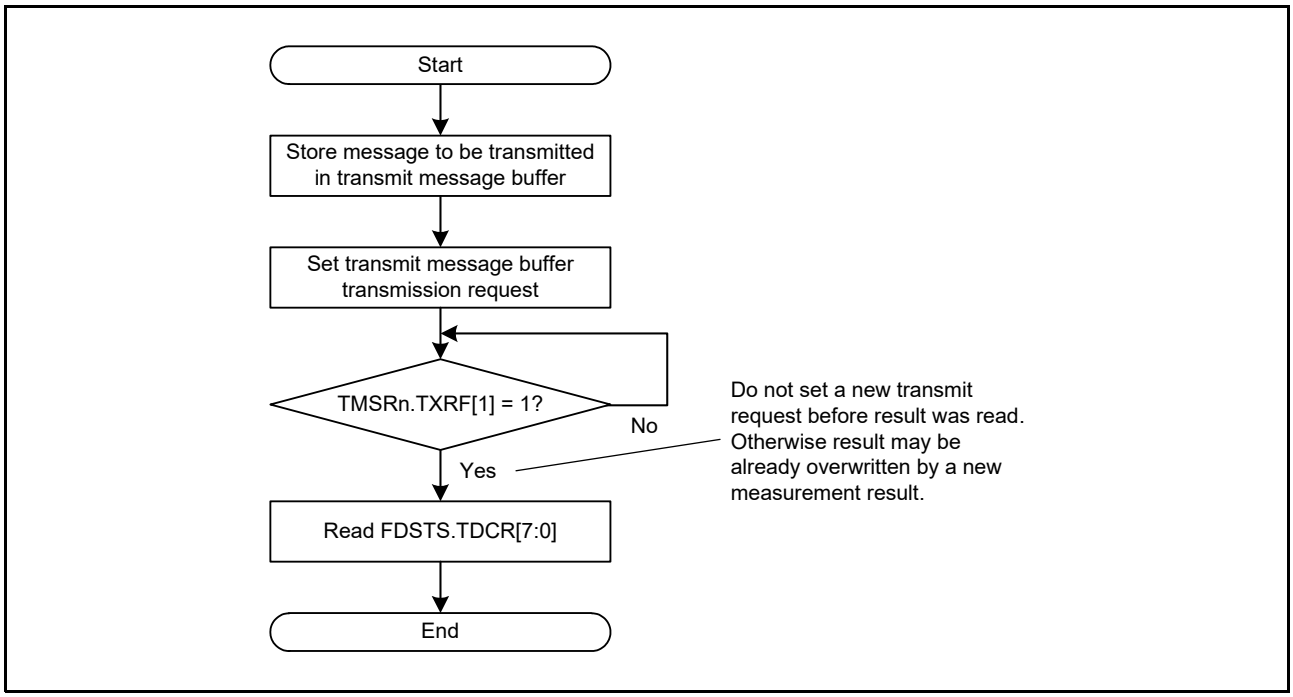


Figure 45.22 Transceiver Delay Compensation Result Read Flow

45.4.2 CANFDG Module Configuration after a Reset

After resetting the MCU or applying a software reset using the GRCR.SRST bit, the CANFDG module automatically enters GL_SLEEP mode.
To configure the CANFDG module, set the GCR.SLPRQ bit to 0 to exit GL_SLEEP mode.
After resetting the MCU, the CANFDG module starts RAM initialization when the module-stop state is released and clocks are supplied. At this time, the GSR.RAMST flag is automatically set to 1, indicating that the CANFDG module is initializing the RAM.
After RAM initialization is complete, this flag is automatically set to 0.
RAM initialization is necessary to prevent false detection of ECC errors due to undefined data in RAM after MCU reset. Do not access (read or write) any other CANFDG registers until the RAM initialization is complete and the RAMST flag is set to 0.
Before entering communication mode, the Acceptance Filter List and message FIFO buffers must be configured. In addition, the CAN channel settings such as CAN bit timing must be configured. To make this configuration, the CAN channel must be released from CH_SLEEP mode and configured for communication in CH_RESET mode (configuration mode).
Figure 45.23 shows the configuration procedure. For details about each step, refer to section 45.5, Filtering Using Acceptance Filter List (AFL), section 45.6, FIFO Buffers and Message Buffer Configuration, section 45.11, Interrupts and DTC/DMA Requests, and section 45.4.1.3, Bit Rate.
The CANFDG module does not perform the RAM initialization sequence after executing a software reset by setting the GRCR.SRST bit to 1.

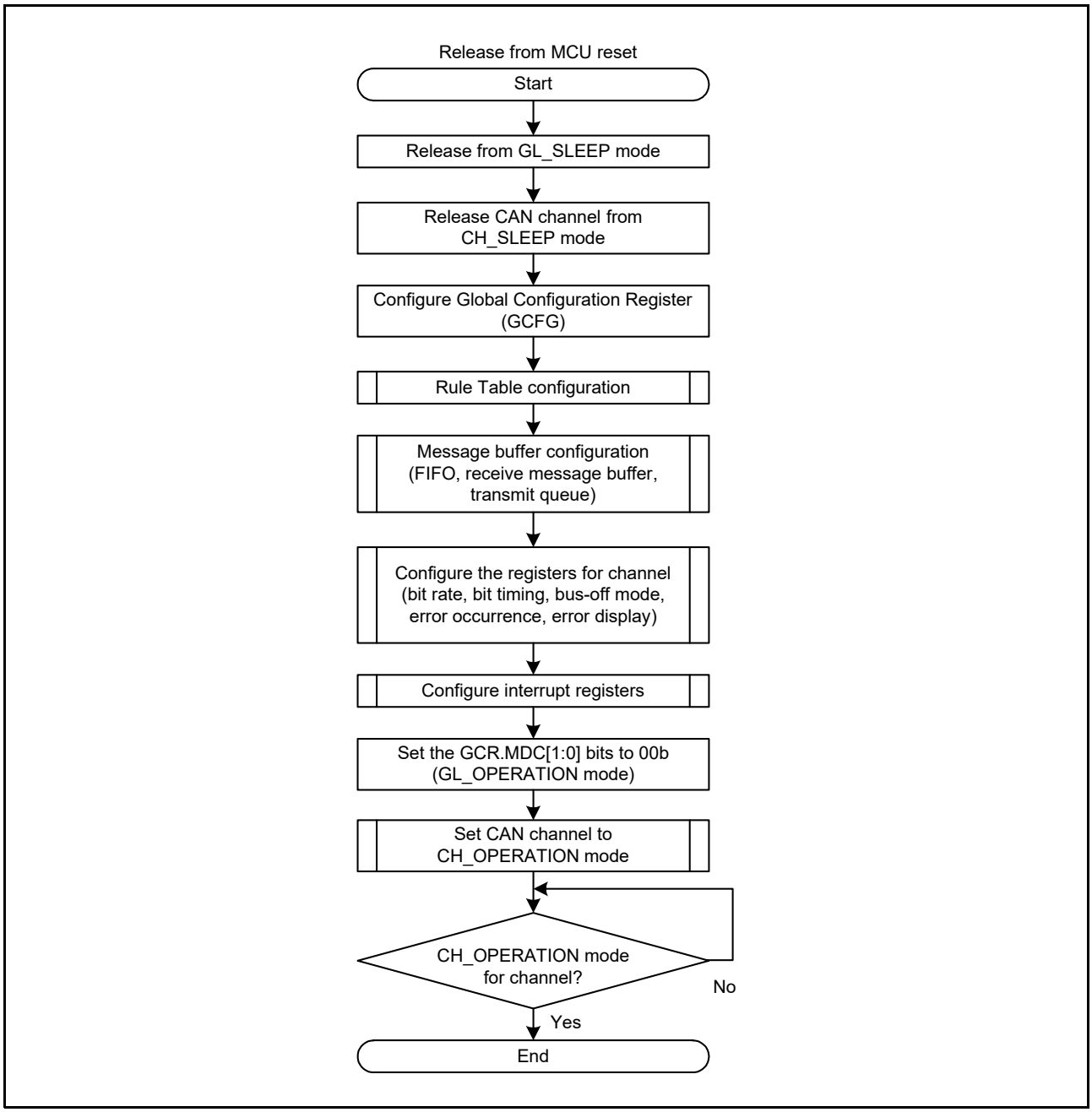


Figure 45.23 Configuration Procedure After An MCU Reset

45.5 Filtering Using Acceptance Filter List (AFL)

The CANFDG module allows you to use the Acceptance Filter List (AFL) to filter message acceptance for channels. Each entry of the AFL defines a filter rule for received messages.
The followings are performed based on the AFL entries:

- Acceptance filtering based on the RTR value, IDE value, and ID value of received message
- DLC filtering based on DLC value of received message
- Payload overflow based on the GCFG.OMRC bit
- Storage of accepted messages in the specified message buffer/FIFO buffer
- Attaching a 16-bit pointer to the stored messages, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages.

The CANFDG module allows up to 128 AFL entries. Up to 64 AFL entries can be used per channel.

45.5.1 Acceptance Filtering Process

Acceptance filtering matches each AFL entry in a channel with the received message. Matching starts with the lowest AFL entry number in the channel.
The AFL search stops when the received message ID matches the specified ID/mask combination, or when the received message ID has been matched against all defined AFL entries. If no match occurs, then the received message is discarded. No notification is given to the application in this case.

45.5.2 DLC Filtering Process

DLC filtering is performed for each accepted message if DLC check is enabled (GCFG.DCE bit = 1). If the DLC value of the received message is equal to or higher than the DLC value specified for the AFL entry whose ID matches in the Acceptance Filtering process (the matching AFL entry), the DLC check is passed.
If DLC replacement is enabled (GCFG.DRE bit = 1), DLC value configured in the matching AFL entry is greater than 0000b and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination receive message buffer (RMBn) or FIFO buffer. If DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are not stored in the RMBn/FIFO buffer. These excess data bytes are stored as 00h in the RMBn/FIFO buffer.
If DLC replacement is enabled and the DLC value specified in the matching AFL entry is 0000b (DLC filter function is disabled), the DLC value of the received message is stored in the RMBn/FIFO buffer.
If DLC replacement is disabled (GCFG.DRE bit = 0) and DLC check passes, the DLC value of the received message is stored in the RMBn/FIFO buffer. If the DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are also stored in the RMBn/FIFO buffer.
If DLC value of the received message is less than the DLC value specified in the matching AFL entry, DLC check fails. In this case, the received message is discarded and is not stored anywhere.
Additionally, when the DLC check fails, the GESR.DEDF flag is set to 1. If interrupts are enabled, an error interrupt is also generated. If the DLC check fails, the DLC replacement setting has no effect.

45.5.3 Message Storage

If a received message has passed both acceptance filtering and DLC filtering, the message is stored in receive message buffers 0 to 31, receive FIFO 0 to 7, or common FIFO 0 to 5 configured in receive FIFO mode or gateway FIFO mode. This message storage target information is also defined in the AFL entry. Do not set a target at the AFL entry which is not configured.
Up to eight message storage destinations can be specified. Do not specify more than eight destinations.

45.5.4 Payload Overflow Process

There is a protection mechanism in case the received message contains data with a payload size that is longer than the size that can be stored in the storage (RMCR.PLS[2:0], RFCRn.PLS[2:0] (n = 0 to 7), or CFCRn.PLS[2:0] (n = 0 to 5)).
If GCFG.OMRC = 0 (message is discarded), the message with data bytes that exceed the specified payload size are discarded and not stored. In this case, even if the FIFO is full, the corresponding FMLSR.RFMLn (n = 0 to 7) or CFMLn (n = 0 to 5) flag is not set to 1.
If GCFG.OMRC = 1 (cut to specified size), only data bytes that exceed the specified payload size are discarded. In this case, if the FIFO is full, the corresponding FMLSR.RFMLn (n = 0 to 7) or CFMLn (n = 0 to 5) flag is set to 1 (message lost has occurred).
Depending on the GCFG.DRE bit setting, either the DLC value of the received message or the DLC value specified in the AFL entry is stored.
Regardless of the GCFG.OMRC bit setting, the GESR.PODF flag is set to 1 if a payload overflow condition is detected. The DLC filtering is performed before the payload overflow processing. Therefore, for each received frame, only one of the GESR.DEDF and GESR.PODF flags is set to 1.

45.5.5 Allocation of AFL Entries to Each CAN Channel

The number of AFL entries (number of rules) per channel can be configured using the AFCFG register (refer to Figure 45.24).

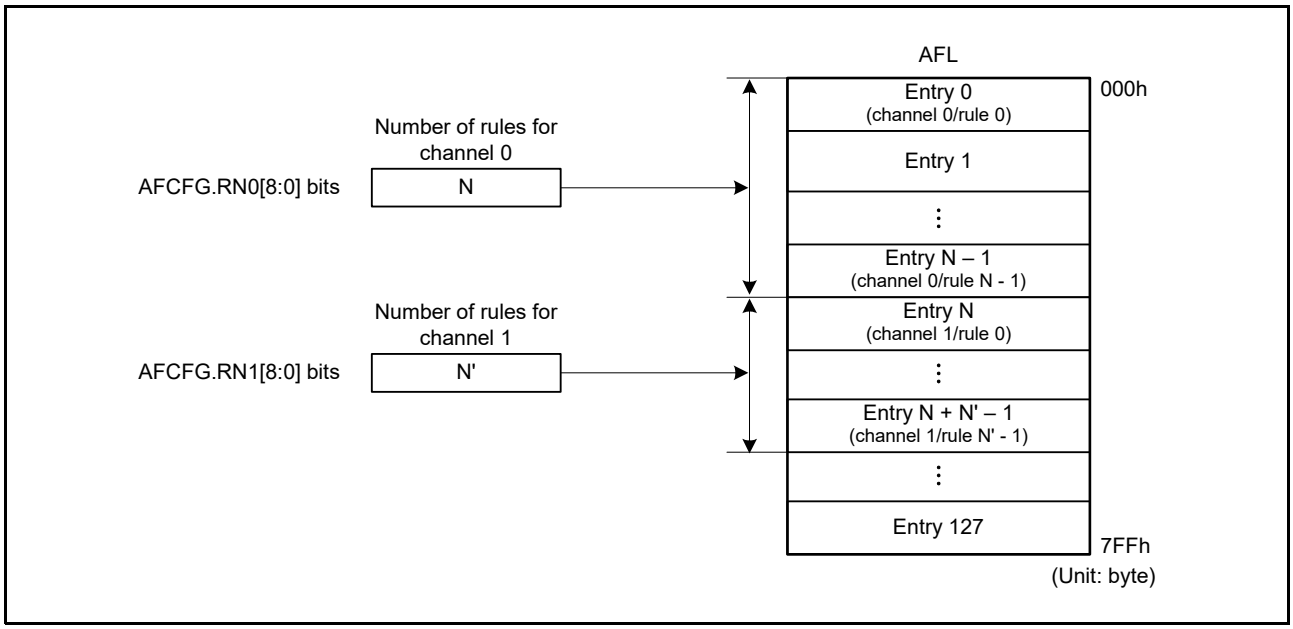


Figure 45.24 Configuration of AFL

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 64. The total number of entries is up to 128.
All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.
The CANFDG module does not flag errors related to the configuration of the AFL.

45.5.6 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.
Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier field (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit matches this field with the identifier field of the received message (29 bits of identifier field can be masked individually, refer to the description of Mask for Identifier field below).
- IDE bit:
  Acceptance filter unit matches this bit with the IDE bit of the received message and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, refer to the description of Mask for IDE bit below).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, refer to the description of Mask for RTR bit below).
- Loop-Back Configuration bit:
  This bit can enable or disable the AFL entry depending on the loop-back configuration or mirror mode condition.
- Mask for Identifier field (29 bits):
  Each bit in the identifier mask field can mask the corresponding identifier bit in the AFL entry (refer to Figure 45.25).
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of an AFL entry, the AFL entry can accept messages in both Standard Identifier format and Extended Identifier format. For messages in Standard Identifier format, the Standard Identifier part of the AFL entry is compared, and for messages in Extended Identifier format, the Extended Identifier part of the AFL entry is compared.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of an AFL entry, the AFL entry can accept frame formats in both data frame and remote frame.
- Pointer (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added when storing a message in the message buffer area and can be used as a support function in the application.
  For example, the pointer information can be used to support the assignment of PDU IDs to the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a message accepted by the related AFL entry. This label is added when storing a message in the message buffer area and can be used as support function in the application.
- DLC field:
  If the DLC value of the received message is equal to or higher than the value set in this field, the DLC check is passed.
  If the DLC value of an AFL entry is set to 0000b, the DLC filtering for this entry is effectively disabled (all accepted messages pass DLC filtering).

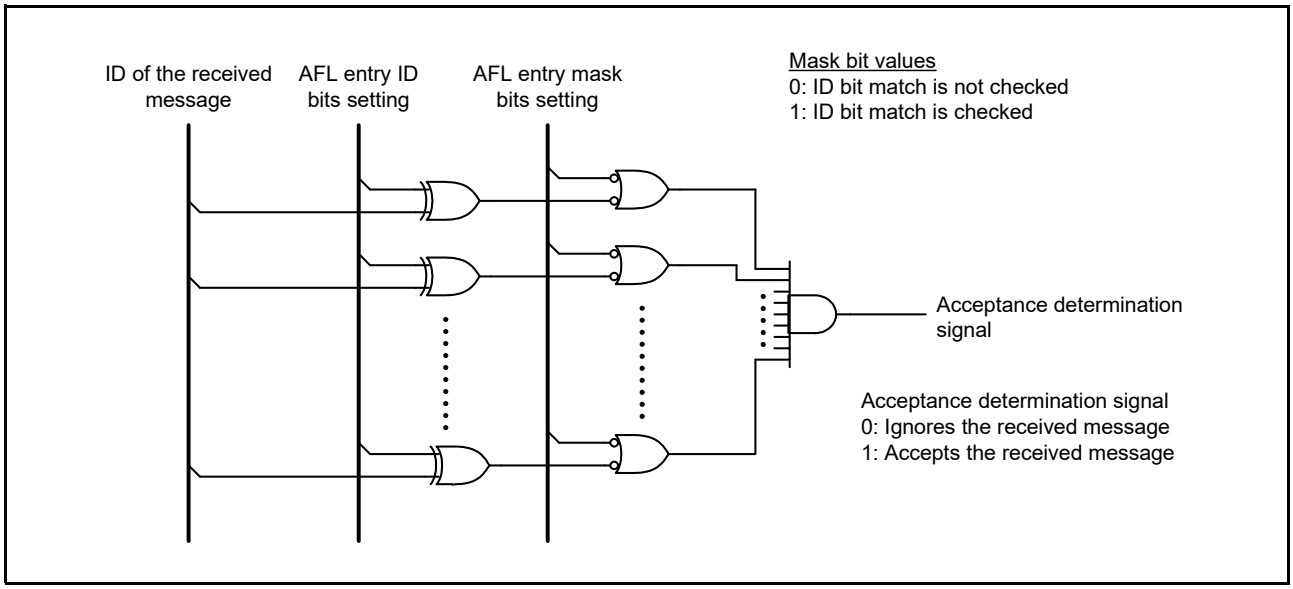


Figure 45.25 Acceptance Filter Function

Each AFL entry contains the following information for processing the received messages:

- Message buffer number of the receive message buffer used as the storage destination for received messages
- The Destination Message Buffer Setting Enable bit to specify the receive message buffer as the storage destination for received messages
- The FIFO Destination Enable bit to specify the FIFO as the storage destination for received messages.

Note: Received messages can be stored in a common FIFO for another channel. If the common FIFO is configured in gateway FIFO mode, the messages stored in the common FIFO is transmitted from that channel.

There is no protection function for storing messages. Therefore, the FIFO Destination Enable bit must be set carefully.

45.5.7 Entering Entries in the AFL

One complete entry can be entered into AFL via the following registers:

- AFLn.IDR register: First part of the AFL entry
- AFLn.MASK register: Second part of the AFL entry
- AFLn.PTR0 register: Third part of the AFL entry
- AFLn.PTR1 register: Fourth part of the AFL entry

A set of 16 of these registers configures one page of AFL entries. The CANFDG module has 128 entries, and all of these entries can be accessed by specifying the page with the AFCR.PAGE[3:0] bits. Set the AFL in CH_RESET or CH_HALT mode. Table 45.22 shows pages and their corresponding AFL entries.

Table 45.22 Pages Linked to the AFL Entries

Page	Linked AFL Entries
Page 0	Entry 0 to 15
Page 1	Entry 16 to 31
Page 2	Entry 32 to 47
Page 3	Entry 48 to 63
Page 4	Entry 64 to 79
Page 5	Entry 80 to 95
Page 6	Entry 96 to 111
Page 7	Entry 112 to 127

AFL access control is performed using the AFCR register (Figure 45.26). This register has the following bits:

- The PAGE[3:0] bits for selecting the AFL page number
- The AFLWE bit that enables or disables the writing of data to prevent unnecessary write access to the AFL.

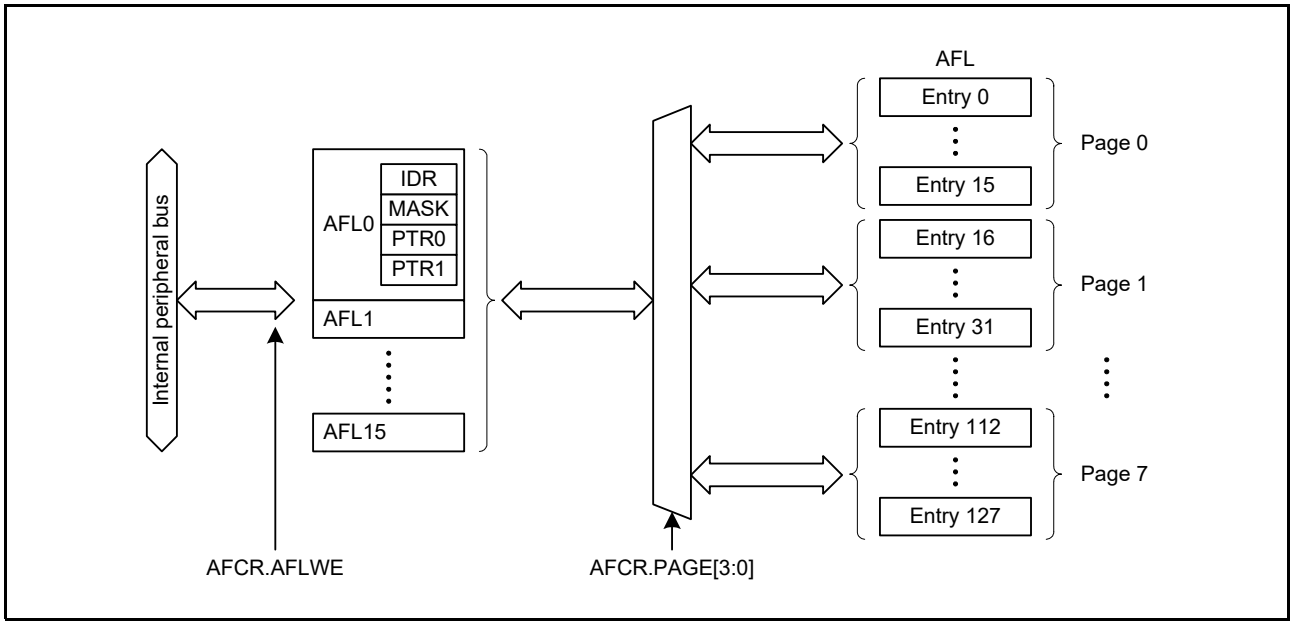


Figure 45.26 AFL Page Access

Follow the flow shown in Figure 45.27 to configure the AFL.
After entering all the entries, writing to the AFL must be disabled to prevent unnecessary write access to the AFL.
If the AFCR.AFLWE bit is set to 0, write protection is enabled during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION).
Even if the AFCR.AFLWE bit is set to 0, reading from AFL is possible during all Global modes (the consistency of AFL contents can be checked during execution).

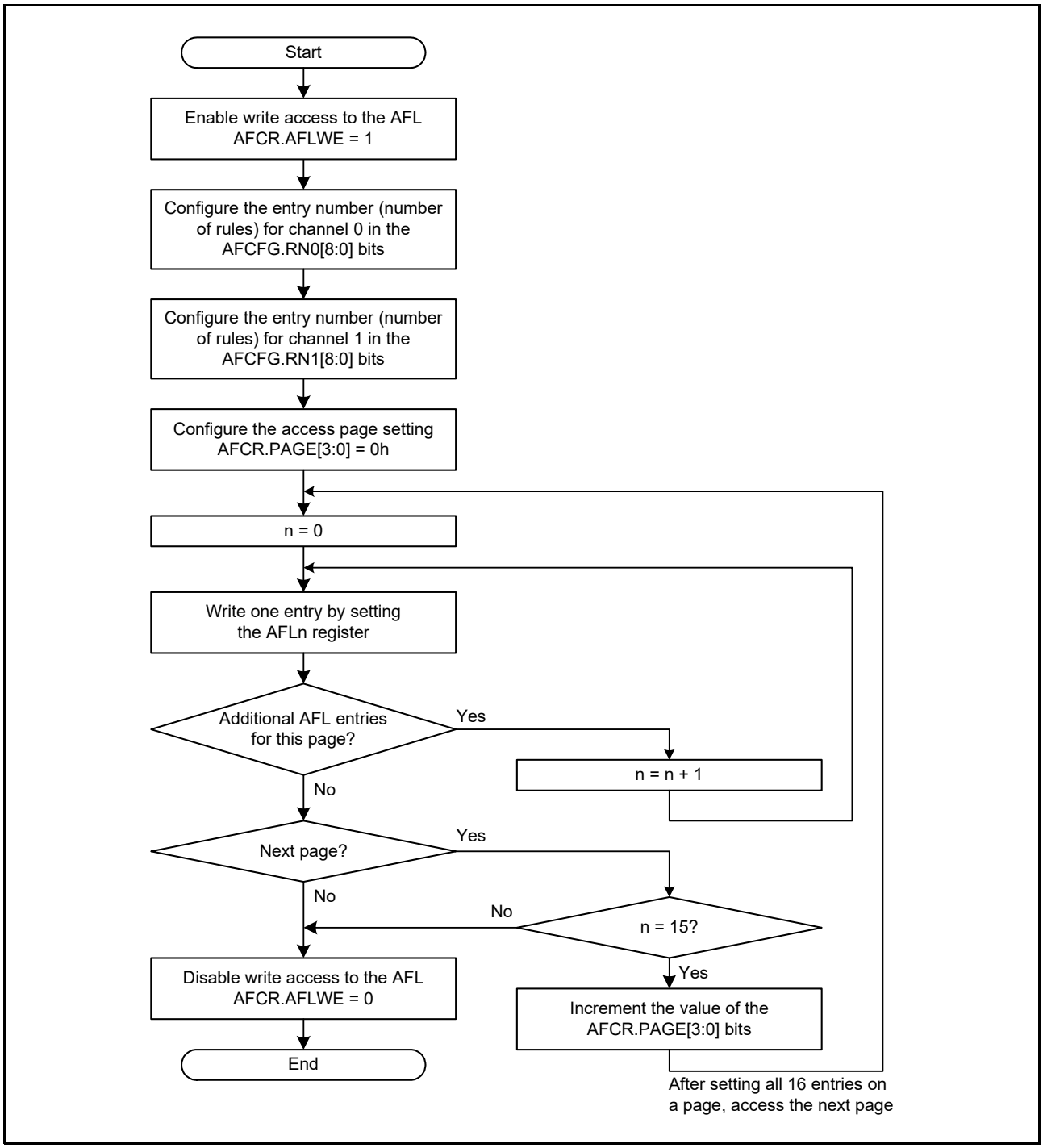


Figure 45.27 AFL Configuration Flow

45.5.8 Loop-Back Configuration

AFL entries with the AFLn.IDR.LPC bit set to 1 are used only when it receives a message transmitted by the CAN channel itself in loop-back mode (self-test mode 0 or self-test mode 1) or mirror mode. If a message transmitted by another node on the CAN bus is received while in loop-back mode, the AFL entry will not be used.

AFL entries with the LPC bit set to 0 are used only when the following messages are received.

- Messages transmitted by other nodes
- Messages transmitted by the CAN channel itself in loop-back mode

The mirror mode can be enabled with the GCFG.MME bit. If the message is successfully transmitted when the GCFG.MME bit is 1, the message is stored in the receive message buffer or FIFO buffer if there is a matching entry in the AFL for that channel. The AFLn.IDR.LPC bit in the matching AFL entry must be set to 1 to store this message. If mirror mode and loop-back mode are set at the same time, the loop-back mode behavior is applied. Table 45.23 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 45.23 Behavior of Acceptance Filter Based on the Loop-Back Setting in AFL Entry

Mirror Mode (MME Bit)	Loop-Back Mode (Self-Test Mode 0 or Self-Test Mode 1)	Channel Mode	LPC Bit	AFL Entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

45.5.9 IDE Masking

The IDE bit set in the AFL entry with the AFLn.MASK.IDEM bit set to 0 is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is selected based on the received IDE bit.
The following lines show the examples.

- The ID and mask fields of an AFL entry x is configured as follows:
  - AFLx.IDR = C0553A20h → IDE = 1, RTR = 1, LPC = 0, ID[10:0] = 220h/ID[28:0] = 00553A20h
  - AFLx.MASK = 0000FFFFh → IDEM = 0, RTRM = 0, IDM[10:0] = 7FFh/IDM[28:0] = 0000FFFFh
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 220h is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 320h is received, this is not a match
  - If a frame with IDE = 1 and ID = 1FFF3A20h is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 08803220h is received, this is not a match.

45.5.10 Updating AFL Entry during Communication

The AFL entry can be updated without disabling CAN communications.
Set the AFL entry number to update to the ignore entry select bits and set the ignore entry enable bit to 1.
The entry number specified here is ignored by AFL matching while the entry is being updated.
Figure 45.28 shows the update flow for an AFL entry.

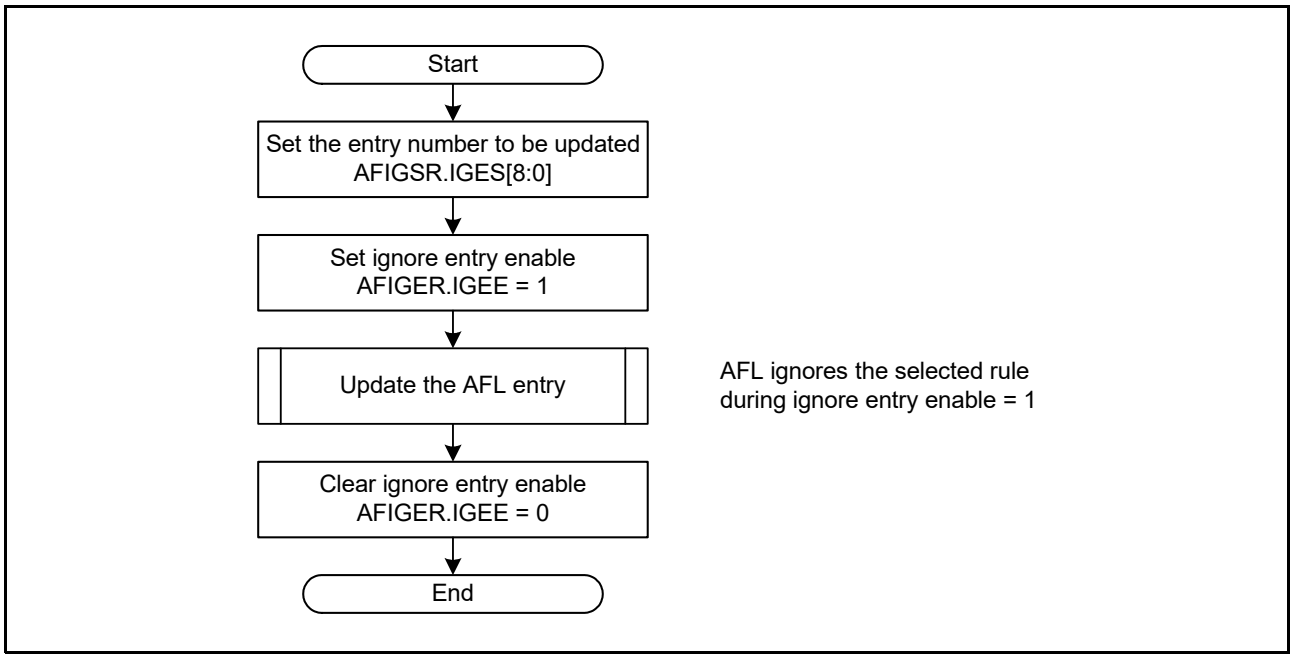


Figure 45.28 Update Flow for An AFL Entry

The method to update an AFL entry is as follows:

1. Set the channel number and entry number*1 to be updated to the AFIGSR.IGCS[2:0] and IGES[8:0] bits.
2. Set the value 0000C401h (key code and ignore entry enabled) to the AFIGER register.
3. Set the page number which includes the entry to be updated to the AFCR.PAGE[3:0] bits. Set the AFCR.AFLWE bit to 1.
4. Set the new rule to the AFLn.IDR, AFLn.MASK, AFLn.PTR0, AFLn.PTR1 registers.
5. The AFCR.AFLWE bit is set to 0.
6. Set the value 0000C400h (key code and ignore entry disabled) to the AFIGER register.

Note 1. This entry number is not used for acceptance filtering between (2) and (5).

(1) Example 1: Deleting an Entry

The following describes how to delete entry 3 in channel 1 when the total number of entries per channel is 6.

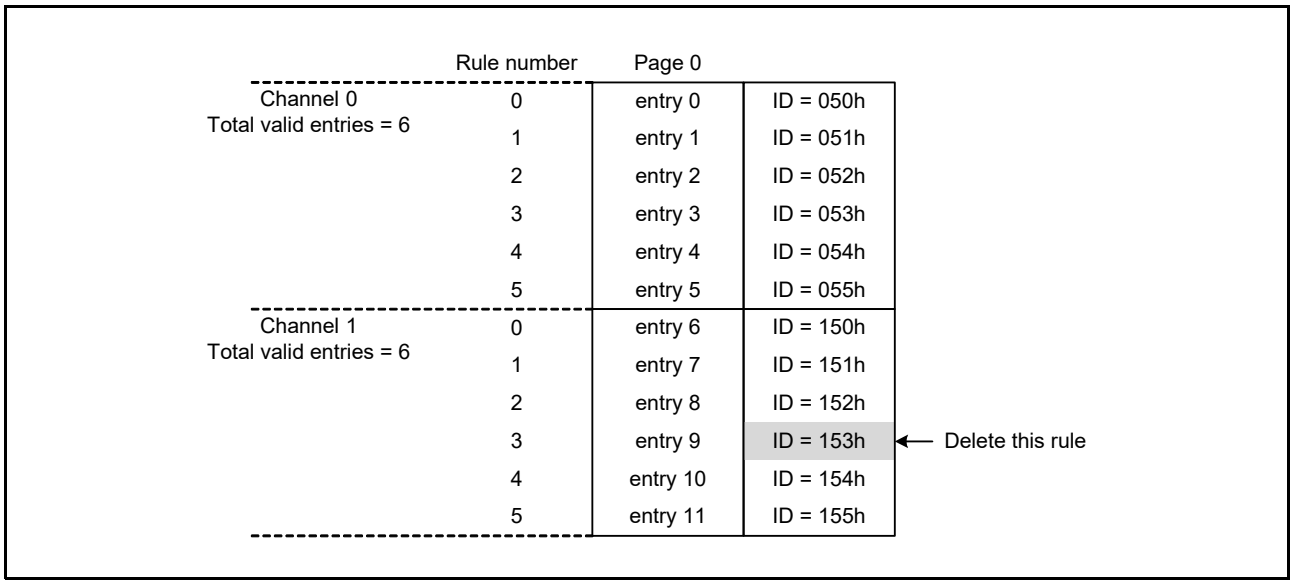


Figure 45.29 Example of Deleting an Entry (Before Deleting Entry 9)

[How to delete an entry]

(1) Set 00010003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the same rule as the previous entry by accessing the AFL9.IDR, AFL9.MASK, AFL9.PTR0, AFL9.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

Entry 9 is now deleted.

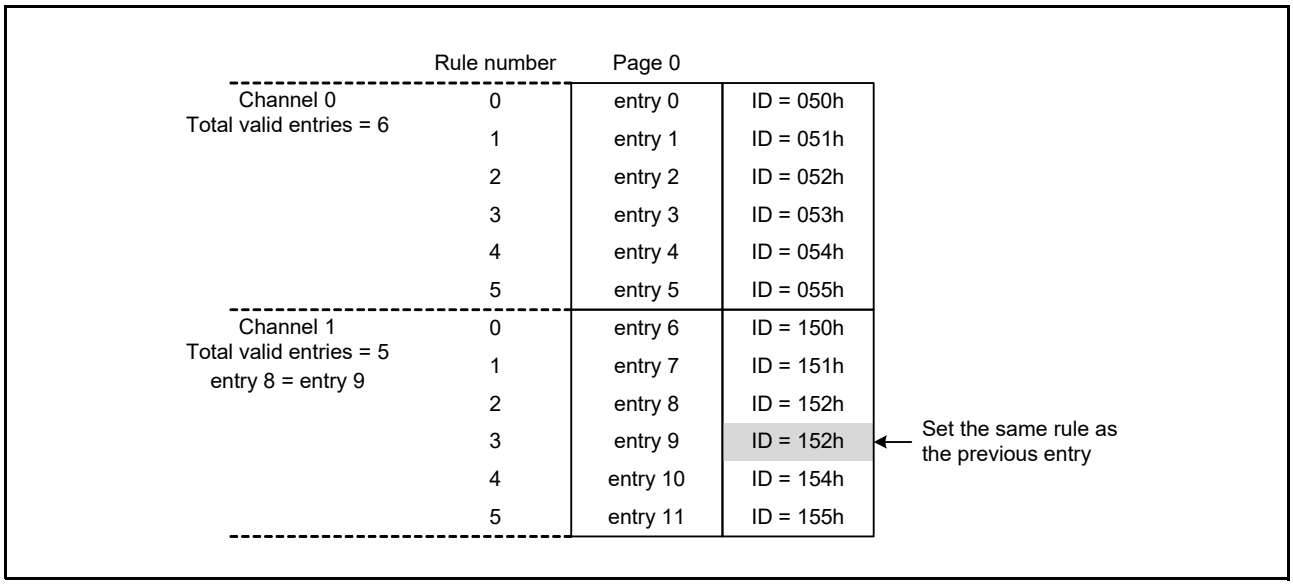


Figure 45.30 Example of Deleting an Entry (After Deleting Entry 9)

(2) Example 2: Adding an Entry (Update Unused Entry)

The following describes how to add a new rule to entry 3 in channel 1 when the total number of entries per channel is 6.

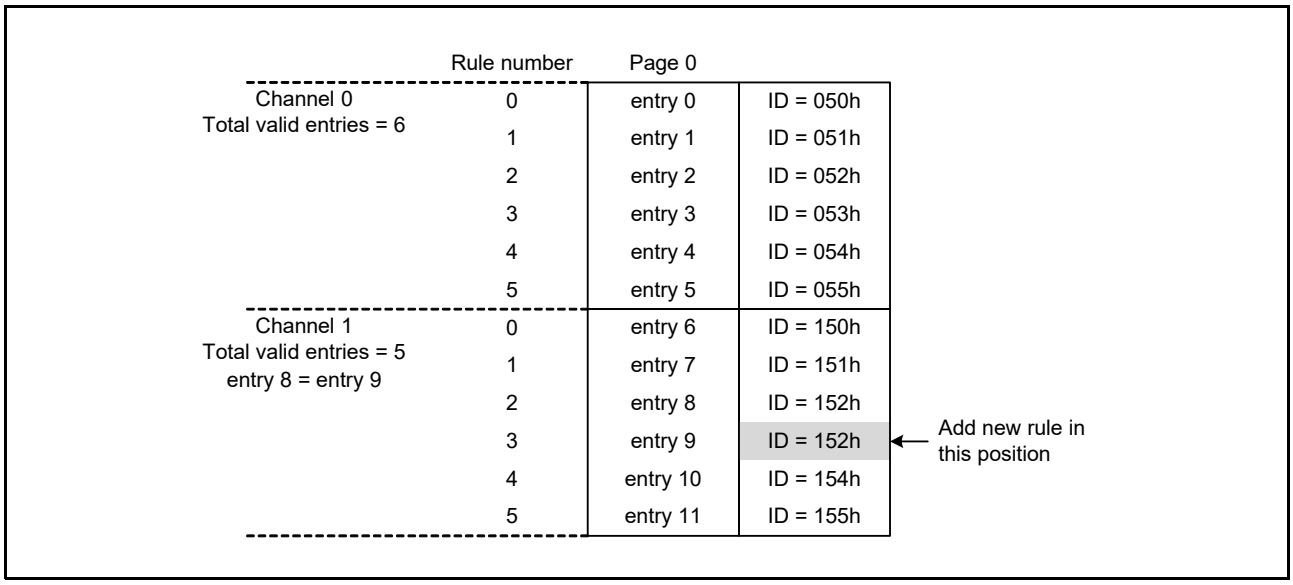


Figure 45.31 Example of Adding an Entry (Before Updating Entry 9)

[How to add an entry]

(1) Set 00010003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the new rule by accessing the AFL9.IDR, AFL9.MASK, AFL9.PTR0, AFL9.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

The new entry is now added.

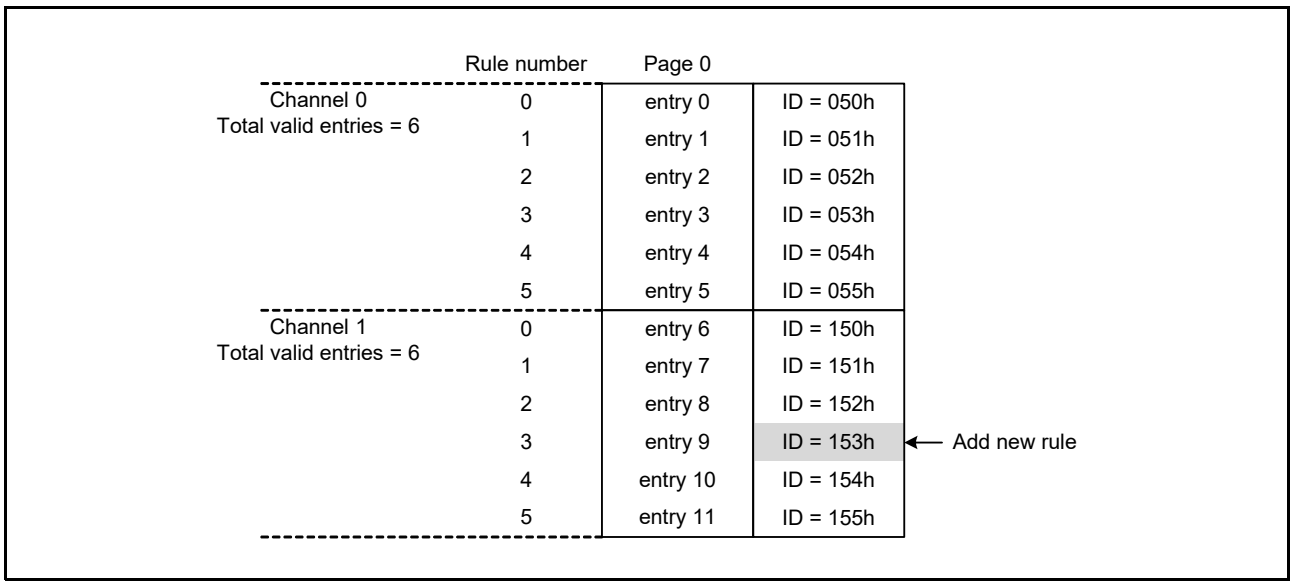


Figure 45.32 Example of Adding an Entry (After Updating Entry 9)

The acceptance filter can use entries in the range of value set in the AFCFG register and entries can be added/deleted within that range. Therefore, the AFCFG register should be set to the maximum number of entries to use.

45.6 FIFO Buffers and Message Buffer Configuration

This section describes the process for configuring the number of receive message buffers, the FIFO buffers, and the transmit message buffers in the CANFDG module. The message buffers are mapped as shown in Figure 45.33.
The receive message buffers can be accessed with the RMBn register (n = 0 to 31).
The receive FIFOs can be accessed with the RFBn register (n = 0 to 7).
The common FIFO can be accessed with the CFBn (n = 0 to 5) register.
If the common FIFO is configured in transmit FIFO mode, data can only be written to the FIFO buffer by the CFBn register.
If the common FIFO is configured in receive FIFO mode or gateway FIFO mode, data can only be read by the CFBn register.
The transmit message buffers can be accessed with the TMBn register (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103).
If unused message buffer is read, it is read as undefined.

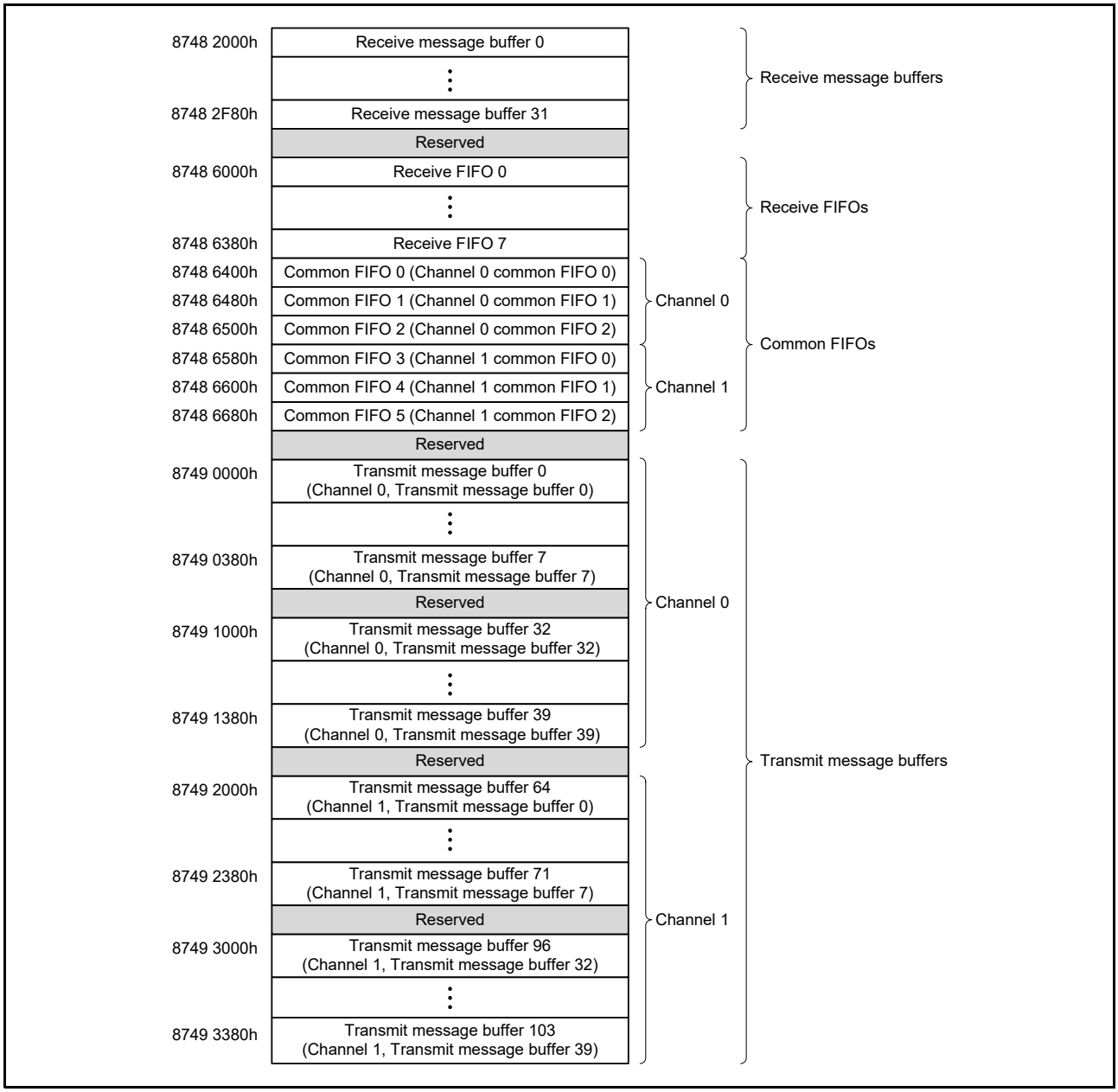


Figure 45.33 Message Buffer Configuration

45.6.1 Receive Message Buffers

In CANFDG module, the received frames can be stored in receive message buffers based on the configuration of the AFL entries.
Additionally, the number of receive message buffers required by the system can be selected from 0 to 32.

45.6.1.1 Receive Message Buffer Configuration

The number of receive message buffers in CANFDG module can be configured by writing to the RMCR.NMB[7:0] bits. Set the number of message buffers in the range 0 (no receive message buffers) to 32. Do not set a value larger than this.
The AFL entries for routing the received messages to receive message buffers must be configured to match the requirements of the system.
The AFL entries must also be configured properly, and an AFL entry for receive message buffers should not exceed the number of message buffers configured in the RMCR.NMB[7:0] bits.

Note: There is no internal check procedure provided in CANFDG module against wrong configuration of the AFL.

The payload size of the receive message buffer can be configured with the RMCR.PLS[2:0] bits. The default size is 8 bytes and the maximum size is 64 bytes.
If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

45.6.2 FIFO Buffers

The CANFDG module provides FIFO buffers for storing received frames and frames to be transmitted, respectively. There are eight receive-only FIFO buffers, but common FIFOs can be configured to store messages for transmission, reception, or gateway use. There are three common FIFOs per channel.
These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- FIFO depth
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the transmit FIFO or gateway FIFO

If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

45.6.2.1 FIFO Buffers Configuration

In CANFDG module, the FIFO buffers can be configured to match the system requirements.
The total number of FIFO buffers is 14 (8 receive FIFOs + 6 common FIFOs).

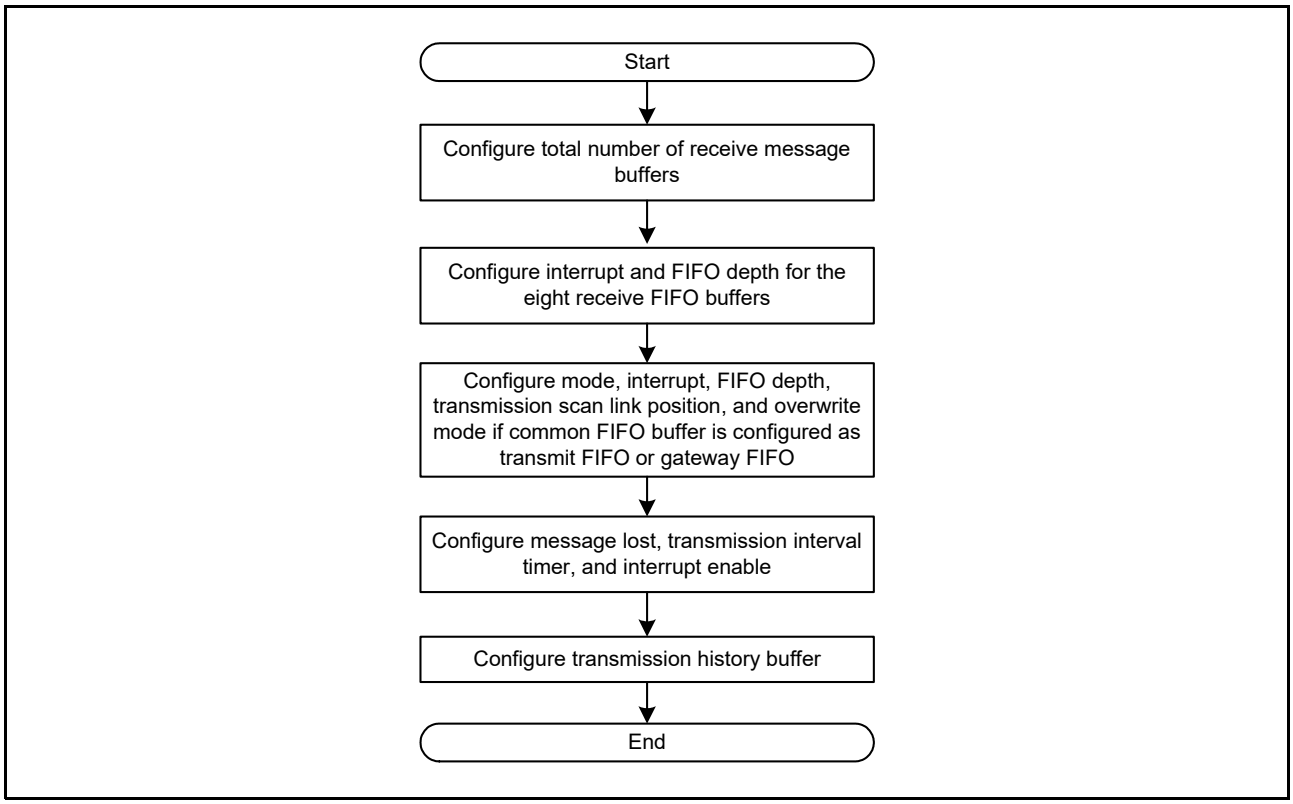


Figure 45.34 FIFO Buffer Configuration Flow in CANFDG module

As shown in Figure 45.34, the various FIFO buffers can be configured by writing to the Receive FIFO n Configuration Register and the Common FIFO n Configuration Register.

For the receive FIFOs, the following parameters can be configured:

- Interrupts
- FIFO depth
- Payload size

For the common FIFO, the following parameters can be configured:

- Mode
- Interrupts
- FIFO depth
- Payload size
- Transmission scan link position

(1) FIFO mode configuration of common FIFO

The mode of the common FIFO can be configured by writing to the CFCRn.MODE[1:0] bits. The modes that can be configured in the common FIFO are as follows:

- 00b: Receive FIFO mode (default mode after an MCU reset)
- 01b: Transmit FIFO mode
- 10b: Gateway FIFO mode
- 11b: Setting prohibited

Messages can only be read from the receive FIFOs and the common FIFO configured in receive FIFO mode. Messages are stored by the CANFDG module in these FIFO buffers based on the AFL entries.
Messages can be read and written into the common FIFO configured in transmit FIFO mode. These messages are transmitted on the corresponding CAN channel.
In the common FIFO configured as gateway FIFO mode, messages can only be read.
The pointers can be incremented only when a new message is stored in the FIFO buffer and decremented only when a message is transmitted on the corresponding CAN channel by the CANFDG module.
After an MCU reset, all the common FIFO is configured in receive FIFO mode by default. Only enable the FIFO buffers after configuring the common FIFO in the required modes.
In gateway FIFO mode, if a new message is received and stored in a FIFO buffer that is already full, the existing message is overwritten or the received message is discarded, depending on the setting of the CFECRn.CFOWM bit, as follows:

- When the CFECRn.CFOWM bit is 0:
  The received message is discarded and the CFSRn.LOST flag is set to 1.
- When the CFECRn.CFOWM bit is 1:
  The oldest message in the buffer is overwritten with the received message, and the read pointer moves to the second oldest message. The CFSRn.CFOWF flag becomes 1, indicating that the message has been overwritten.

In addition, when a CAN bus error or arbitration-lost for the transmit message occurs in the FIFO buffer with full state, the message is discarded and retransmission for the message is not performed. The read pointer automatically moves to the next message.
When the CFCRn.CFE bit is 1, do not change the CFECRn.CFOWM bit.

(2) FIFO transmit message buffer link configuration

When the common FIFO is configured as transmit FIFO or gateway FIFO, the FIFO buffer must be linked to a normal transmit message buffer to participate in the transmission scan of a CAN channel.
Two or more common FIFOs cannot be linked to the same transmit message buffer.
Do not write data into a transmit message buffer that is linked to a common FIFO. Also, the transmit message buffer linked to a common FIFO should not be a part of the transmit queue.
The link to the transmit message buffer of each common FIFO can be configured by writing to the CFCRn.LTM[4:0] bits. The options available for linking the transmit message buffer are:

- 00000b: Transmit message buffer 32
- 00001b: Transmit message buffer 33
- 00010b: Transmit message buffer 34
- 00011b: Transmit message buffer 35
- 00100b: Transmit message buffer 36
- 00101b: Transmit message buffer 37
- 00110b: Transmit message buffer 38
- 00111b: Transmit message buffer 39

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the RFCRn.FDS[2:0] bits and CFCRn.FDS[2:0] bits. The eight available options for depth configuration are:

- 000b: 0 messages (FIFO buffer cannot be used)
- 001b: 4 messages
- 010b: 8 messages
- 011b: 16 messages
- 100b: 32 messages
- 101b: 48 messages
- 110b: 64 messages
- 111b: 128 messages

The RAM allocated to the receive message buffers and FIFO buffers is limited to 64 messages (4864 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFDG module does not have the function to check the validity of the configuration.

Note: If the FIFO depth of the common FIFO is 4 messages or more (CFCRn.FDS[2:0] > 000b), the link between the common FIFO and the transmit message buffer is valid regardless of whether the FIFO is disabled or enabled. If the FIFO depth is 0 messages, the link between the common FIFO and transmit message buffer is invalid regardless of whether the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The payload size of each FIFO buffer can be configured by writing to the RFCRn.PLS[2:0] bits and CFCRn.PLS[2:0] bits. The eight available options for payload size configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 64 messages (4864 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFDG module does not have the function to check the validity of the configuration.

(5) FIFO interrupt configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the RFCRn.RFIM bit and CFCRn.CFIM bit. The two available options are:

- RFIM/CFIM = 0:
  - Receive FIFO mode: An interrupt is generated when the FIFO fill level reaches RFCRn.RFITH[2:0] or CFCRn.CFITH[2:0] value
  - Transmit FIFO mode: An interrupt is generated when the FIFO transmits the last message successfully
  - Gateway FIFO mode:
    When receiving, an interrupt is generated when the FIFO fill level reaches CFCRn.CFITH[2:0] value
    When transmitting, an interrupt is generated when the FIFO transmits the last message successfully.
- RFIM/CFIM = 1:
  - Receive FIFO mode: An interrupt is generated each time the received message is stored
  - Transmit FIFO mode: An interrupt is generated each time a message is successfully transmitted.
  - Gateway FIFO mode:
    When receiving, an interrupt is generated each time the received message is stored
    When transmitting, an interrupt is generated each time a message is successfully transmitted.

If the RFCRn.RFIM bit for the receive FIFO is 0, an interrupt is generated based on the setting of the RFCRn.RFITH[2:0] bits.
Similarly, if the CFCRn.CFIM bit for a common FIFO set to receive FIFO mode is 0, an interrupt is generated based on the setting of the CFCRn.CFITH[2:0] bits.
The eight options available to set the FIFO fill level for generating interrupts are:

- 000b: Interrupt generated when FIFO is 1/8 Full
- 001b: Interrupt generated when FIFO is 1/4 Full
- 010b: Interrupt generated when FIFO is 3/8 Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8 Full
- 101b: Interrupt generated when FIFO is 3/4 Full
- 110b: Interrupt generated when FIFO is 7/8 Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message fill level matches the set value.
However, as shown in Table 45.24, there are some restrictions on the RFITH[2:0] and CFITH[2:0] bit settings, depending on the FDS[2:0] bits (FIFO depth setting) of each register.

Table 45.24 FIFO Interrupt Threshold and FIFO Depth Settings

	RFITH[2:0]/CFITH[2:0]							
FDS[2:0]	**111b (full)**	**110b (7/8)**	**101b (3/4)**	**100b (5/8)**	**011b (1/2)**	**010b (3/8)**	**001b (1/4)**	**000b (1/8)**
000b (0 messages)	Invalid (FIFO cannot be enabled)							
001b (4 messages)	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited
010b (8 messages)	Allowed							
011b (16 messages)	Allowed							
100b (32 messages)	Allowed							
101b (48 messages)	Allowed							
110b (64 messages)	Allowed							
111b (128 messages)	Allowed							

45.6.2.2 FIFO Buffers Control

To enable the receive FIFO interrupt, set the RFIE or RFFIE bit in the RFCRn register (n = 0 to 7) to 1.
To enable the common FIFO interrupt, set either the CFRIE or CFTIE bit in the CFCRn register or the CFORIE or CFOTIE bit in the CFECRn register:
After configuration is complete, each FIFO can be enabled by setting the RFCRn.RFE bit and CFCRn.CFE bit to 1 to allow transmission and reception of messages.
When the CFECRn.CFBME bit is 1, the common FIFO is in buffering mode, and transmit data is stored in the common FIFO but is not transmitted. Transmission starts when the CFECRn.CFBME bit is set to 0.
When the CFCRn.CFE bit is 1, do not change the CFECRn.CFBME bit from 0 to 1.

45.7 Reception and Transmission

45.7.1 Reception

In the CANFDG module, messages received on any of the channels, will be stored in receive message buffers or in receive FIFOs or common FIFO configured in receive FIFO mode/gateway FIFO mode depending upon the Acceptance Filter List entries:

- up to 32 receive message buffers can be configured
- eight receive FIFOs available
- up to six common FIFOs can be configured in receive mode or gateway FIFO mode
- up to six transmit queues can be configured in gateway mode

45.7.1.1 Message Storage in Receive Message Buffers

When a message is successfully received and stored in a receive message buffer, the corresponding NDF[n] flag in the RMNDR0 register is set to 1.
The stored message can be read from the corresponding receive message buffer.
If a new message is stored into a receive message buffer before the previous message in this message buffer is read, the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the receive message buffer. If such a loss of messages is not acceptable, store related messages by using receive FIFO.

Note: Unused data bytes are filled with 00h depending on the DLC value.

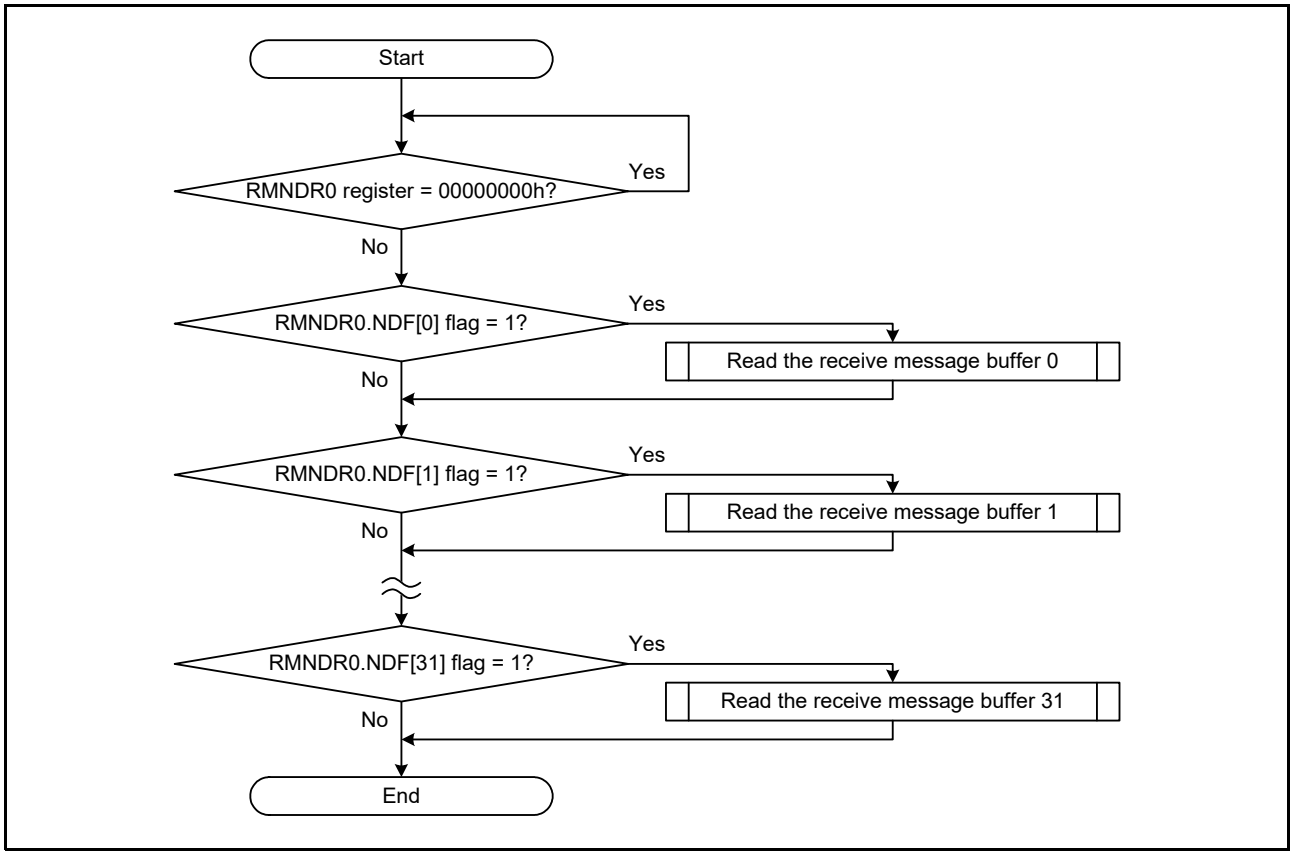


Figure 45.35 Receive Message Buffer Message Access Flow (Example for Polling Case)

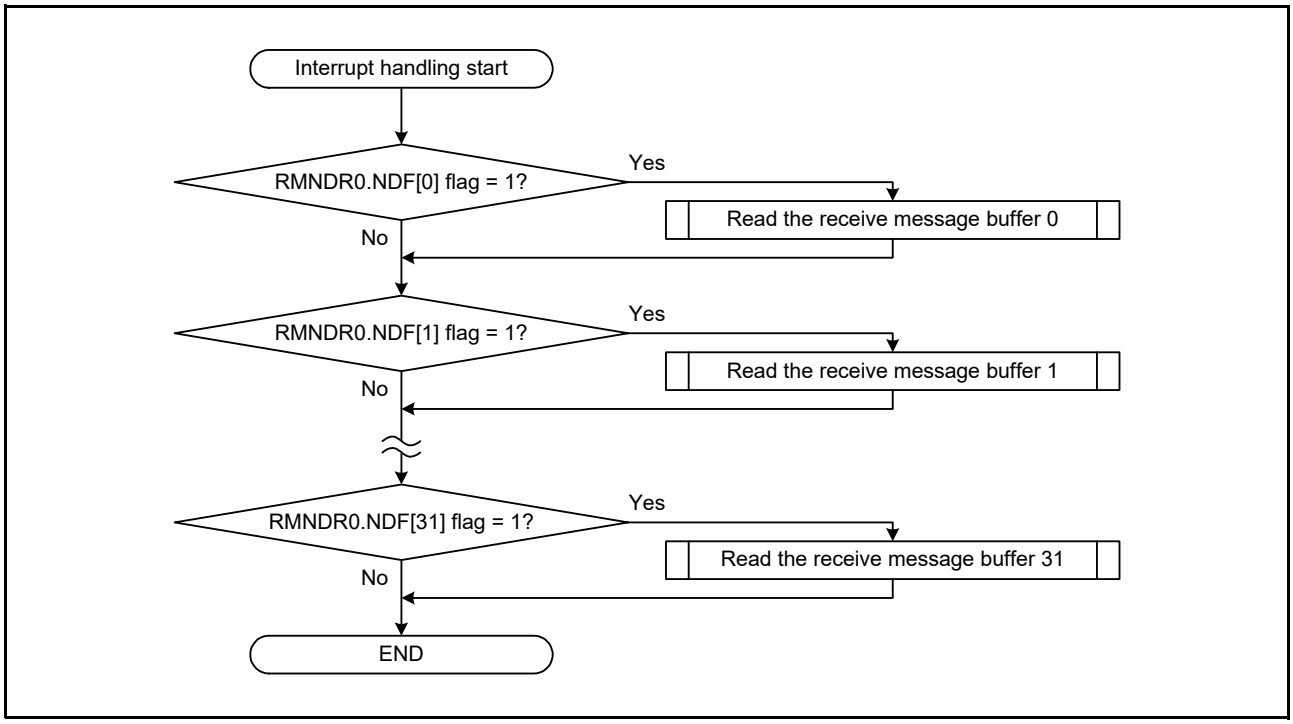


Figure 45.36 Receive Message Buffer Message Access Flow (Example for Interrupt Case)

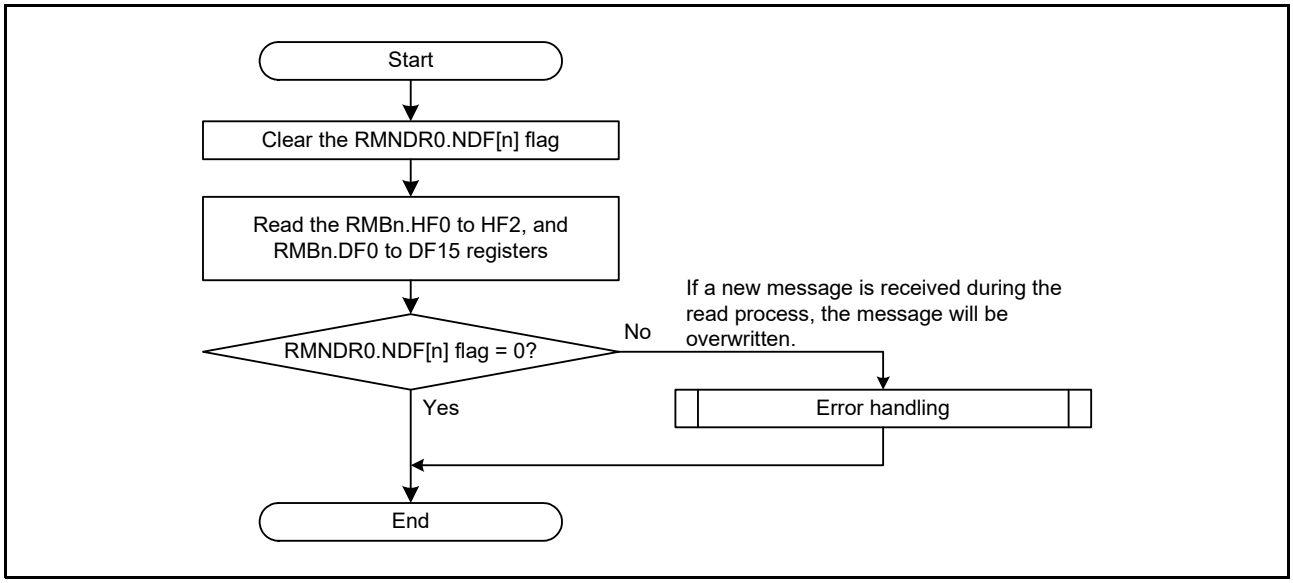


Figure 45.37 Receive Message Buffer n Read Flow

45.7.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to receive FIFOs or common FIFO configured in receive FIFO mode/ gateway FIFO mode should be configured based on the requirements of the system.
The AFLn.PTR1.RF0E to RF7E, or CF0E to CF5E bit in the matching AFL entry selects the FIFO buffers to which the related reception message will be stored.
When the received message is stored in one or more receive FIFOs or common FIFO configured in receive FIFO mode/ gateway FIFO mode, the message counter value is incremented in the corresponding Receive FIFO Status Register or Common FIFO Status Register.
Depending upon the configuration of the FIFO buffers, an interrupt may also be generated. The message can be read from the corresponding FIFO access registers.

Note: Since many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, then the FIFO full flag is set.
When the value 000000FFh is written to the corresponding FIFO Pointer Control Register, then the message count is decremented by 1.

Only write 000000FFh to the FIFO Pointer Control Register after completely reading the message from the FIFO Access Register of the corresponding FIFO.
When all the messages stored in the FIFO are read, then the FIFO empty flag is set.
If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO message lost flag is set and the new message will be lost (messages already stored will not be overwritten).
To prevent message loss due to overrun, set an appropriate interrupt generation threshold and generate an interrupt before the FIFO becomes full.

In gateway FIFO mode, if a new message is received and stored in a FIFO buffer that is already full, the existing message is overwritten or the received message is discarded, depending on the setting of the CFECRn.CFOWM bit, as follows:

- When the CFECRn.CFOWM bit is 0:
  The received message is discarded and the CFSRn.LOST flag is set to 1.
- When the CFECRn.CFOWM bit is 1:
  The oldest message in the buffer is overwritten with the received message, and the read pointer moves to the second oldest message. The CFSRn.CFOWF flag becomes 1, indicating that the message has been overwritten.

In addition, when a CAN bus error or arbitration-lost for the transmit message occurs in the FIFO buffer with full state, the message is discarded and retransmission for the message is not performed. The read pointer automatically moves to the next message.
When the CFCRn.CFE bit is 1, do not change the CFECRn.CFOWM bit.

The common FIFO can be configured to interrupt in the following cases:

- When CAN frame reception is completed
- When FIFO is in full status in receive FIFO mode or gateway FIFO mode

Note: The message lost flag is set by the CANFDG module only in receive FIFO mode or gateway mode, and the flag is not set when the FIFO buffer is overloaded by the CPU.

Note: When the AFLn.PTR0.RDS0 to RDS2 bits are set to 1 (a transmit queue is set as the routing destination) and the TQCRn.TQGWE bit is set to 1 (gateway mode of the transmit queue is enabled), received frames are routed to the specified transmit queue and stored as transmit data.

The receive FIFOs and the common FIFO configured in receive FIFO mode or gateway FIFO mode can be disabled at any time by clearing the RFCRn.RFE bit or CFCRn.CFE bit.

When the RFCRn.RFE bit or CFCRn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further messages can be stored into the FIFO.

When the receive FIFOs or common FIFO configured in receive FIFO mode is set to be read by DTC/DMA transfer, do not read the FIFO buffer on the CPU or write 000000FFh to the FIFO Pointer Control Register (the RFPCRn or CFPCRn register). The FIFO read pointer is automatically updated when read by DTC/DMA transfer.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

Note: When the next frame is received before clearing the receive interrupt flag, the receive interrupt flag is not set again.

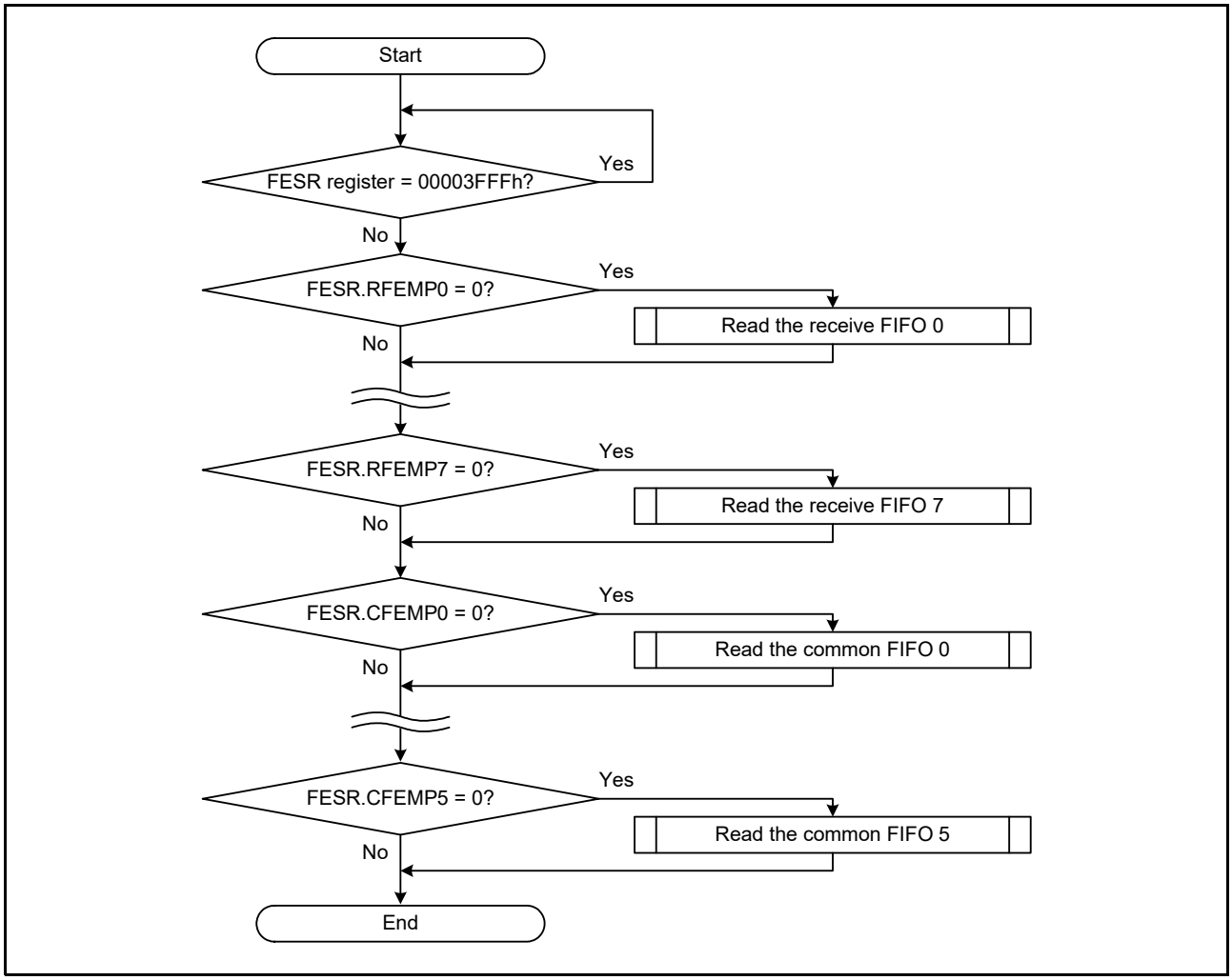


Figure 45.38 FIFO Buffer Message Access Flow (Example for Polling Case)

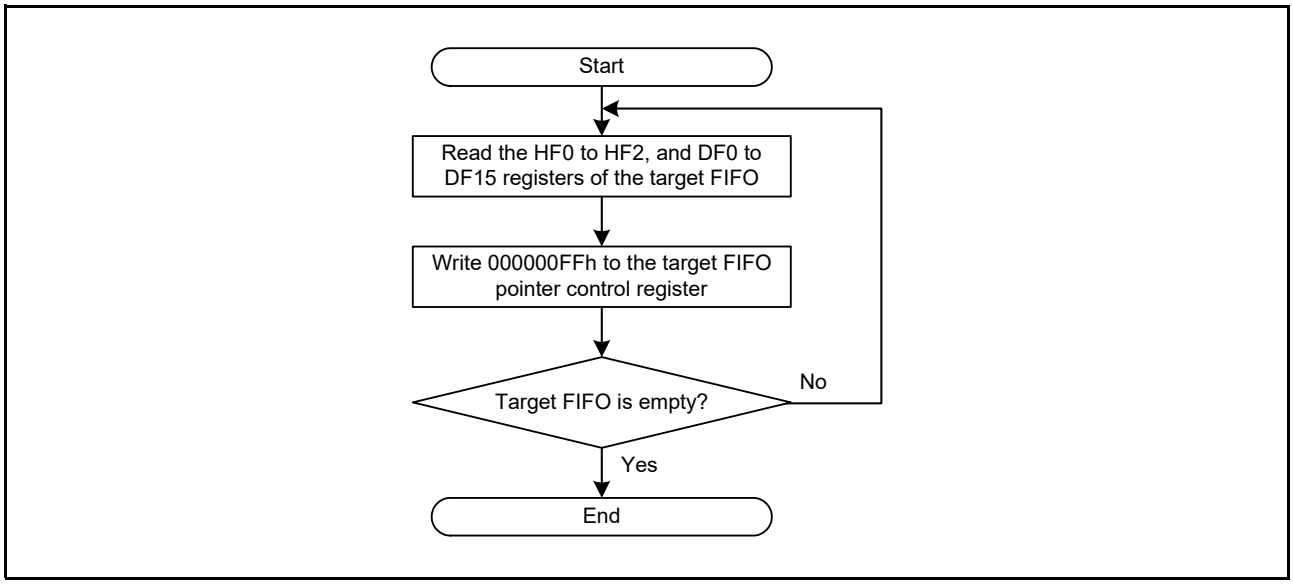


Figure 45.39 FIFO Buffer Read Flow (Example for Polling Case)

If the interrupt flag is cleared after the FIFO read process is completed, the interrupt flag is not set even if the next frame has already been received. The FIFO read process must be performed and the interrupt flag must be cleared before the reception of the next frame is completed. If the process is not in time, make sure the FIFO is empty, clear the interrupt flag, and make sure the FIFO is empty again.

45.7.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of a received message or transmission time of a successful transmitted message. The timestamp counter value will be captured based on the GFDCFG.TSCPS[1:0] bit setting (sample point of SOF, EOF when the frame is taken valid, or sample point of res bit following the FDF bit in case of a CAN FD frame). For reception the timestamp counter value is stored together with the message ID and data into the target receive message buffer, receive FIFO or gateway FIFO.
For transmit message the timestamp counter value is stored as part of the transmission history entry.
The counter can be clocked from PCLKA or the bit timing clock of the CAN channel. The counter clock source can be set with the GCFG.TSCS bit. If the GCFG.TSCS bit is 0, PCLKA is used. If it is 1, the bit time clock of CAN channel selected by the GCFG.TSBTCS[2:0] bits is used.
Care has to be taken when using the bit time clock as clock source. The timestamp counter stops when the selected channel enters CH_HALT or CH_RESET mode. So, also for other CAN channels the timestamp counter value will not be updated.
If PCLKA is selected as the timestamp counter clock source, time stamp counter function is not affected by the Channel mode.
The clock source of the timestamp counter can be divided by the coefficient defined by the GCFG.TSP[3:0] bits (timestamp prescaler).
The timestamp counter can be reset to 00000000h with the GCR.TSCR bit (timestamp counter reset).

45.7.2 Transmission

There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- Transmit queue transmission

The CANFDG module has 16 transmit message buffers for each channel. These message buffers are only used for transmission and cannot be configured for reception.
Additionally transmission from transmit queue and/or common FIFO configured in transmit FIFO mode or gateway FIFO mode can be set in the following way (refer to Figure 45.40):

- Transmit queue
  Up to eight transmit message buffers in a channel can be grouped to form a transmit queue with a common access window. Each channel has four transmit queues.
  Transmit queue 0 and transmit queue 2 use the transmit message buffer from the lower level, and transmit queue 1 and transmit queue 3 use the transmit message buffer from the higher level.
  Do not access the TMCRn or TMSRn registers of the transmit message buffer used for the transmit queue.
  The access window for each transmit queue is as follows:
  – Transmit queue 0: Transmit message buffer 0
  – Transmit queue 1: Transmit message buffer 7
  – Transmit queue 2: Transmit message buffer 32
  – Transmit queue 3: Transmit message buffer 39
  When using transmit queue 0 and transmit queue 1 at the same time, set the total number of queue depth to 8 or less.
  When using transmit queue 2 and transmit queue 3 at the same time, set the total number of queue depth to 8 or less.
- Common FIFO (transmit FIFO mode or gateway FIFO mode)
  The CANFDG module has three common FIFO for each channel. Common FIFO configured in transmit FIFO mode or gateway FIFO mode is linked to any of the transmit message buffers 32 to 39.
  The linked transmit message buffer replaces the common FIFO. Do not access the TMCRn or TMSRn registers of the linked transmit message buffer.
  Refer to Figure 45.33 for information on the assignment of the common FIFO to each channel.

Note: Common FIFO should not be linked to transmit message buffers that are already part of a transmit queue.

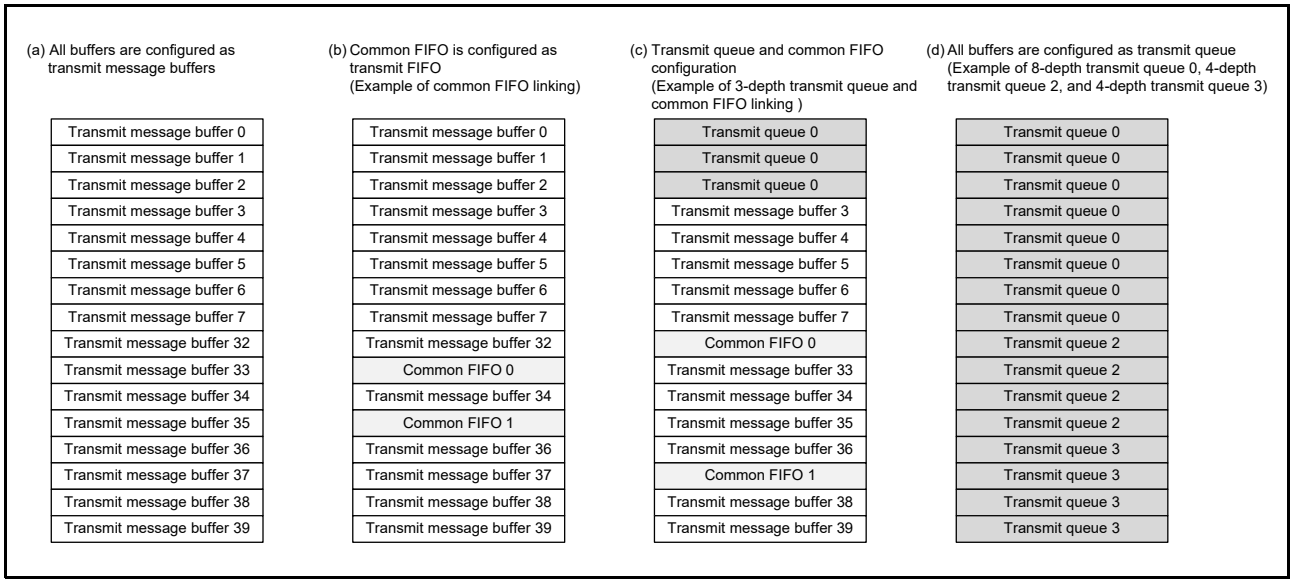


Figure 45.40 Channel Transmit Message Buffer Configuration

45.7.2.1 Transmission Priority

If two or more transmit message buffers are queued for transmission within a channel, messages are transmitted according to one of the following priorities.

- CAN ID priority
- Message buffer number priority

The transmission priority can be set using the GCFG.TPRI bit, and it applies to all message buffers and CAN channels.
For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode or gateway FIFO mode.
Do not select message buffer number priority when using the transmit queue.
For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All transmit message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode or gateway FIFO mode and includes the transmit queue message buffers.
If the ID of two or more message buffers is the same, then the smaller message buffer number will have higher priority for transmission.

Note: For common FIFO configured in transmit FIFO mode or gateway FIFO mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.
If the message is being transmitted from the FIFO, then the next pending message within the same FIFO will be considered in the transmission arbitration.
In contrast to this, all transmit message buffers of a transmit queue will participate in internal transmission arbitration.

Figure 45.41 below shows the transmission configuration flow.

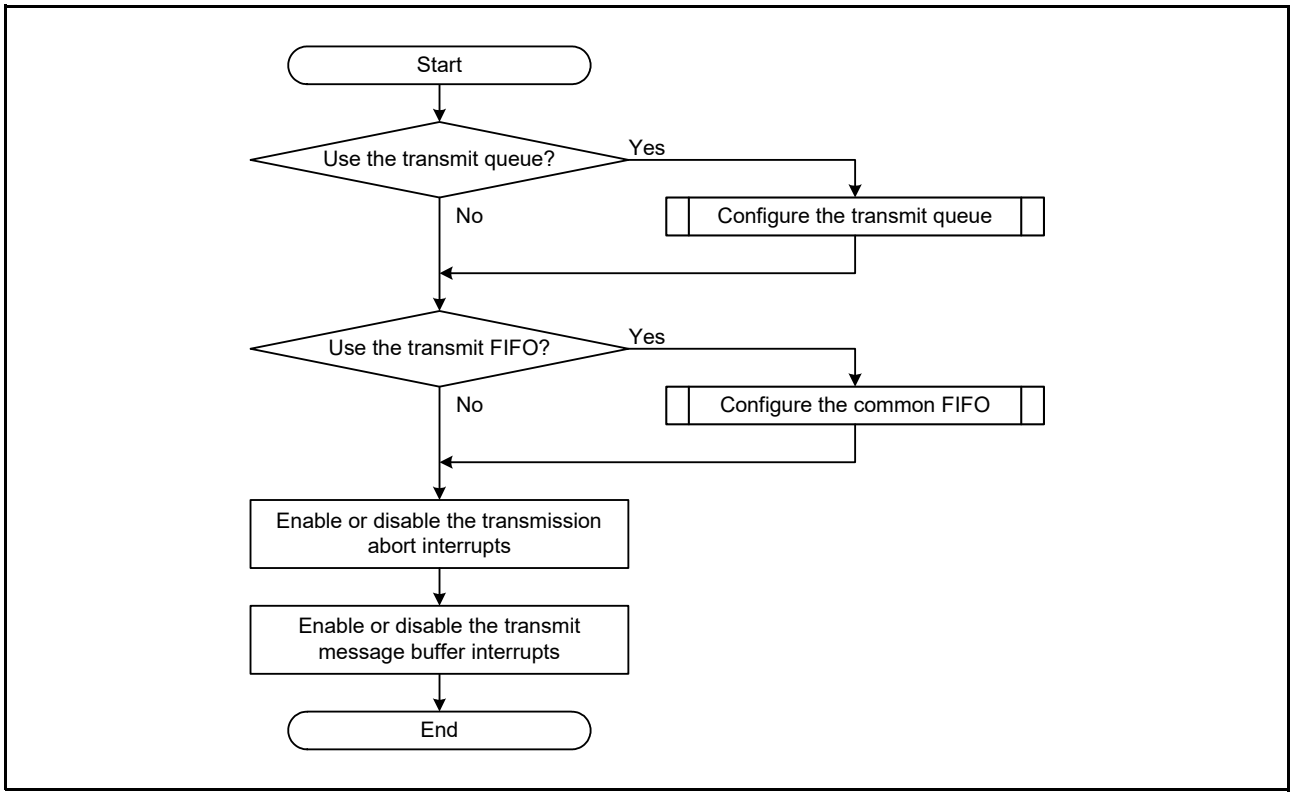


Figure 45.41 Transmission Configuration Flow

45.7.2.2 Message Transmission from Transmit Message Buffer

Each transmit message buffer has two modes of message transmission:

- Normal Transmission Mode
  When the message buffer is set to normal transmission mode, the data frames or remote frames set in that message buffer can be transmitted.
  Completion of normal transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when a normal transmission is successful.
  When arbitration is lost or an error occurs during transmission, the message transmission will be retried if no transmission abort request is set in the transmit message buffer.
  A new internal transmission arbitration is performed for all message buffers that have a transmission request.
- One-Shot Transmission Mode
  When the TMCRn.ONESHOT bit is set to 1, the message buffer is placed in one-shot transmission mode and attempts to transmit the message only once.
  Completion of one-shot transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when the one-shot transmission is successful.
  The TXRF[1:0] flags are set to 01b when arbitration is lost or an error occurs during transmission. In this case, message transmission will not be retried.

Figure 45.42 shows the transmission request procedure from transmit message buffer.

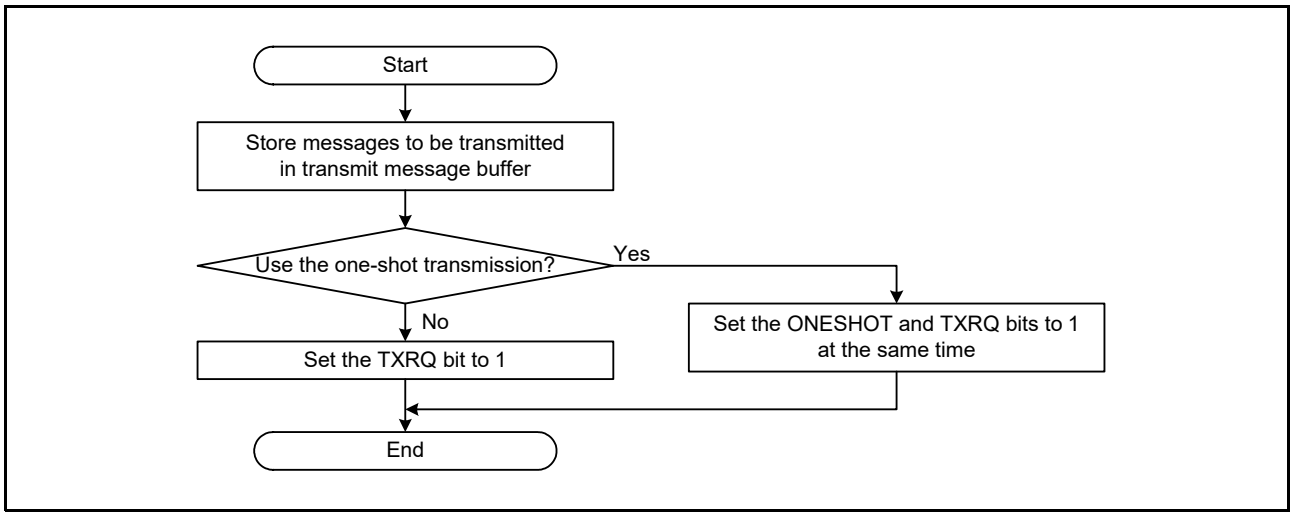


Figure 45.42 Transmission Request Procedure from Transmit Message Buffer

Table 45.25 shows configuration of the TMCRn register.

Table 45.25 Configuration of TMCRn Register

Transmission Request TXRQ Bit	Transmission Abort Request TARQ Bit	One-Shot Transmission Enable ONESHOT Bit	Message Buffer Status
0	0	0	Normal transmission is stopped.
0	0	1	One-shot transmission is stopped.
1	0	0	Data frames or remote frames are transmitted in normal transmission mode.
1	0	1	A data frame or a remote frame is transmitted in one-shot transmission mode.
1	1	0	Transmission abort is requested.
1	1	1	One-shot transmission abort is requested.

Figure 45.43 shows timings for successful transmission from two message buffers on one channel.

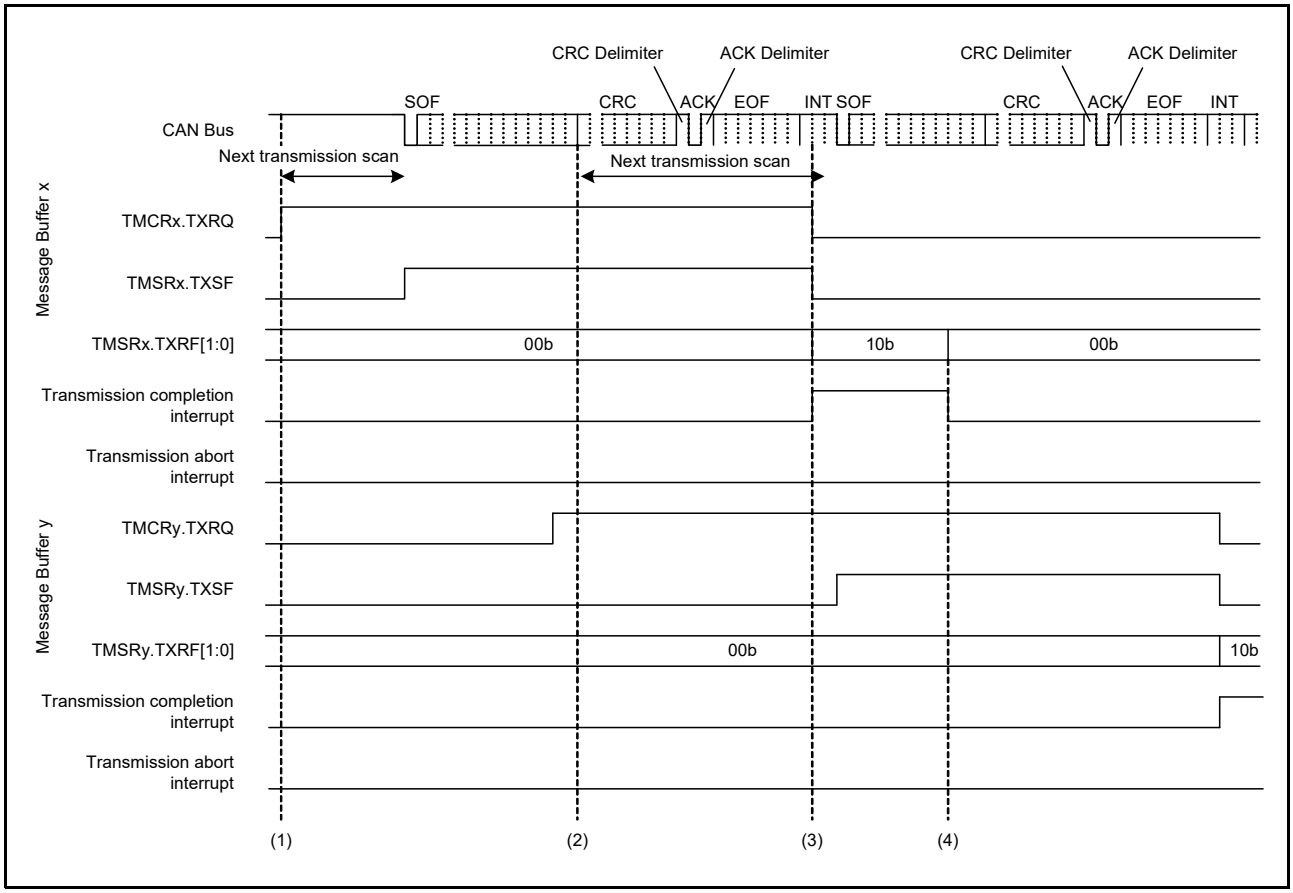


Figure 45.43 Timing of Request and Flag Bits for Successful Transmission

(1) If the TMCRx.TXRQ bit is set to 1 in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.

(3) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 10b and the TMSRx.TXSF flag and the TMCRx.TXRQ bit are cleared.
When the corresponding TMIERm.TMIEj bit is set to 1 (transmit message buffer interrupt enabled), the successful transmission interrupt request is generated. To clear the related interrupt line the TMSRx.TXRF[1:0] flags have to be cleared.

(4) Before starting the next transmission, clear the TMSRx.TXRF[1:0] flags. Load the next message in the transmit message buffer and set the TMCRx.TXRQ bit to 1 again.
The TMCRx.TXRQ bit cannot be set to 1 again before TMSRx.TXRF[1:0] flags are cleared.

Note: The setting point of the TMSRx.TXSF flag is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs either during the transmission or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

The Figure 45.44 shows timings for transmission abort for two message buffers on one channel.

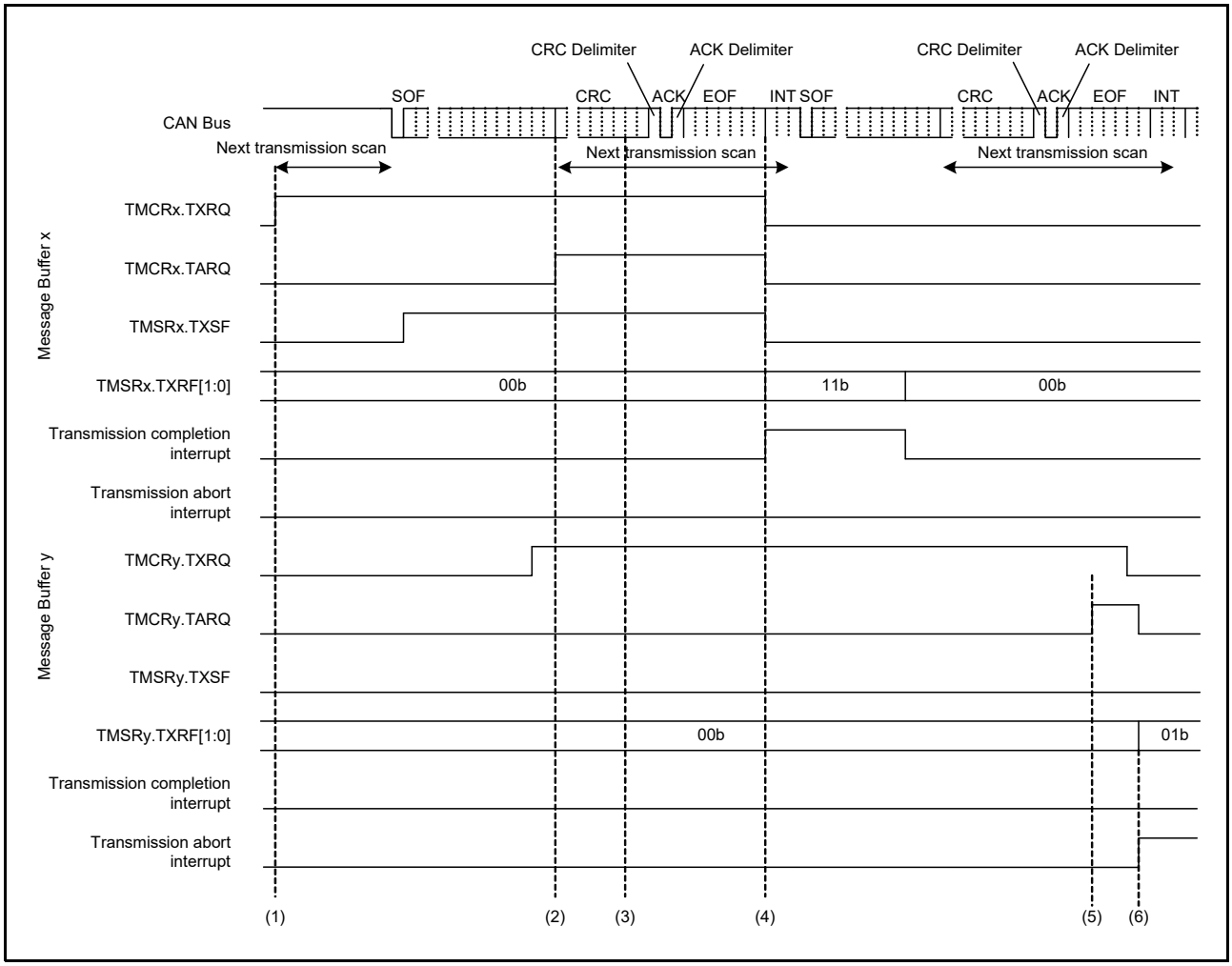


Figure 45.44 Timing of Request and Flag Bits for Transmission Abort

(1) If the TMCRx.TXRQ bit is set in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) If the TMCRx.TARQ bit is set to 1 when the related message buffer is already selected for transmission or currently transmitting, the message will not be aborted, if no error occurs or arbitration is lost.

(3) At the first bit of CRC, the transmission scanning procedure starts for the next transmission. In this example timing chart message buffer y is not selected as next transmit message buffer.

(4) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 11b and the TMSRx.TXSF flag and TMCRx.TXRQ bit are cleared.
When the corresponding TMIERm.TMIEj bit is set to 1 (transmit message buffer interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the TMSRx.TXRF[1:0] flags has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (the TMSRy.TXSF flag is not set): if the TMCRy.TARQ bit is set to 1 when the related channel is under transmission scan, the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the TMSRy.TXRF[1:0] flags are set to 01b.
If the message buffer is not transmitting or selected as next transmit message buffer or under transmit scan, the abort is immediately accepted and the corresponding TMSRy.TXRF[1:0] flags are set to 01b.
In addition, the TMCRy.TXRQ and TMCRy.TARQ bits are cleared automatically.

When the CHCR.TAIE bit is set to 1 (transmission abort interrupt enabled), an interrupt is generated for successful transmission abort.
To clear the related interrupt the TMSRy.TXRF[1:0] flags have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs, either during the transmission, or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

45.7.2.3 Message Transmission from FIFO Buffer

The CANFDG module has three common FIFOs for each channel. The common FIFO can be linked to the transmit message buffer by the CFCRn.LTM[4:0] bits if configured in transmit FIFO mode or gateway FIFO mode.
When a transmission scan is started and the common FIFO corresponding to that transmit message buffer is enabled, the relevant message in the common FIFO will participate in the transmission scan.
Do not configure the transmit message buffer linked to the common FIFO configured in transmit FIFO mode.

(1) Transmit FIFO Operation

Messages can be written into the transmit FIFO by writing to the common FIFO buffer n (CFBn) (n = 0 to 5).
When the value 000000FFh is written into the CFPCRn register, the message count of the FIFO is incremented by 1.
Before writing to the CFPCRn register, wait until the message has been completely written to the CFBn. If the message count matches the FIFO depth, the CFSRn.FULL flag is set to 1.
The oldest message in the transmit FIFO is included in the scan for transmission.
When a message is successfully transmitted from the transmit FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the CFSRn.EMPTY flag is set to 1.
The interrupt generation conditions for the transmit FIFO buffers can be set by the CFCRn.CFIM bit.
If the CFCRn.CFIM bit is set to 0, interrupt is generated when last message is successfully transmitted from the transmit FIFO buffer. If the CFCRn.CFIM bit is set to 1, interrupt is generated for every successfully transmitted message from the transmit FIFO buffer.
Common FIFO can set interrupt, when CAN frame transmission is completed.
The common FIFO configured in transmit FIFO mode can be disabled by setting the CFCRn.CFE bit to 0. If this bit is set to 0, the CFSRn.EMPTY flag is set to 1 as described below:

- immediately if the message from the transmit FIFO is neither scheduled for the next transmission nor in transmission
- following the transmission completion, the detection of an error on the CAN bus, arbitration lost or transition to CH_HALT or GL_HALT mode if the transmission from the transmit FIFO is already scheduled for transmission or already in transmission.

Note: The common FIFO is considered as disabled after setting the CFCRn.CFE bit to 0 only when the CFSRn.EMPTY flag is set to 1 for the corresponding common FIFO.

If there are other pending messages in the transmit FIFO buffer, they will be lost and the transmission needs to be requested again. Before the CFCRn.CFE bit is set to 1 again ensure that the CFSRn.EMPTY flag is set to 1 and that there is no pending abort request from the transmit FIFO.
When the CFCRn.CFE bit is set to 0, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in Figure 45.45.

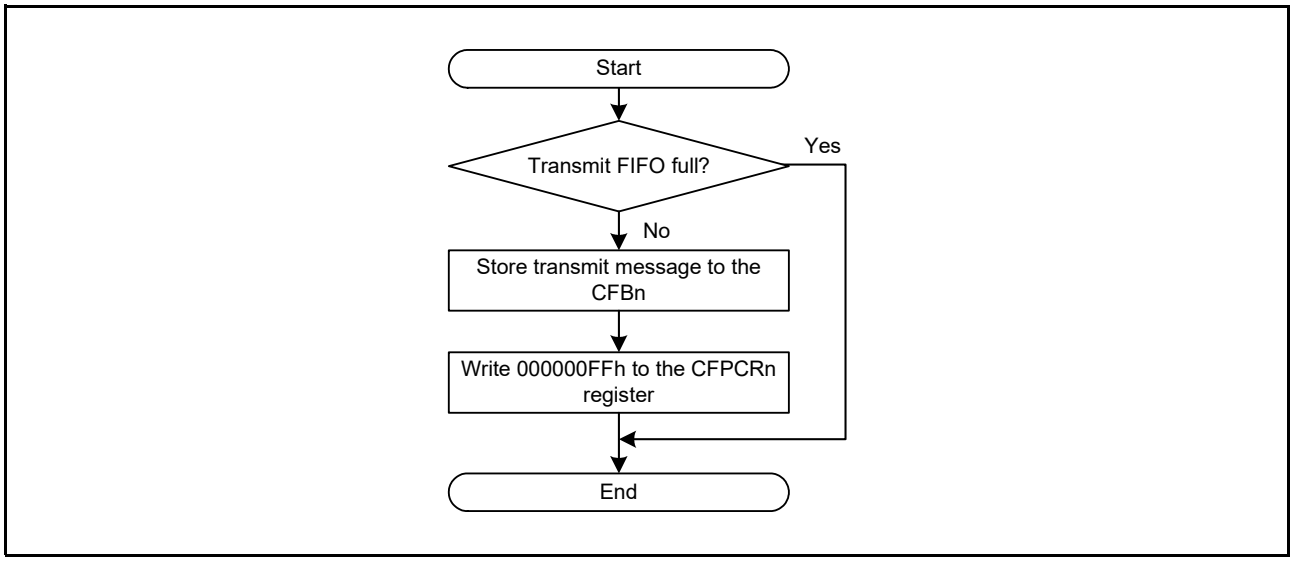


Figure 45.45 Transmit FIFO Transmission Request Procedure

(2) Gateway FIFO Operation

The AFL entry that routes received messages to the gateway FIFO must be configured according to the system requirements. The matching AFL entry selects the gateway FIFO of any CAN channel and stores the received message.
When a message is successfully received and stored in the gateway FIFO, the message count of the FIFO is incremented by 1.
If the message count matches the FIFO depth, the CFSRn.FULL flag is set to 1.
The oldest message in the gateway FIFO is included in the scan for transmission.
When a message is successfully transmitted from the gateway FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the CFSRn.EMPTY flag is set to 1
In gateway FIFO mode, if a new message is received and stored in a FIFO buffer that is already full, the existing message is overwritten or the received message is discarded, depending on the setting of the CFECRn.CFOWM bit, as follows:

- When the CFECRn.CFOWM bit is 0:
  The received message is discarded and the CFSRn.LOST flag is set to 1.
- When the CFECRn.CFOWM bit is 1:
  The oldest message in the buffer is overwritten with the received message, and the read pointer moves to the second oldest message. The CFSRn.CFOWF flag becomes 1, indicating that the message has been overwritten.

In addition, when a CAN bus error or arbitration-lost for the transmit message occurs in the FIFO buffer with full state, the message is discarded and retransmission for the message is not performed. The read pointer automatically moves to the next message.

The common FIFO can be configured to interrupt when CAN frame transmission or reception is completed.
The common FIFO can be configured to interrupt when FIFO is in full status in receive FIFO mode or gateway FIFO mode.
The interrupt generation conditions for the gateway FIFO can be set by the CFCRn.CFIM bit.
If the CFCRn.CFIM bit is set to 0, the number of stored messages is incremented and the receive interrupt flag is set when it reaches the value set in the CFCRn.CFITH[2:0] bits, and the transmit interrupt flag is set when the last message from the FIFO is successfully transmitted.

Setting the CFECRn.CFBME bit to 1 switches to FIFO buffering mode, stores transmit data in the common FIFO, and stops transmission. Transmission starts when the CFECRn.CFBME bit is set to 0.
A common FIFO configured in Gateway FIFO mode can be disabled by setting the CFCRn.CFE bit to 0. When this bit is

set to 0, the CFSRn.EMPTY flag becomes 1 at the following conditions:

- Condition for immediate setting to 1: When there is no next transmission scheduled from the gateway FIFO and no transmission is in progress
- After transmission is completed, after a CAN bus error is detected, after arbitration is lost, or after transition to CH_HALT mode or GL_HALT mode: If the next transmission is scheduled from the gateway FIFO or if transmission is currently in progress

If there are any other messages pending transmission in the gateway FIFO, those messages are lost.
Before setting the CFCRn.CFE bit to 1 again, make sure that the CFSRn.EMPTY flag is 1 and that there are no pending abort requests from the gateway FIFO.
When the CFCRn.CFE bit is set to 0 and the CFSRn.EMPTY flag becomes 1, the message read and write pointers in the gateway FIFO are cleared and become inactive. As a result, all messages in the gateway FIFO are lost and no further messages can be stored in that gateway FIFO.
For applications intended to use CAN-to-CAN gateways, it may be preferable not to replace the value of the ESI (Error State Indication) bit of the routed message with the value of the ESI bit of the transmitter node. For this reason, each channel can use the FDCFG.TESI bit to replace its own node's ESI bit with the value of the ESI bit in the routing message.

Note: If the transmitter node is error passive, the ESI bit is transmitted as error passive (ESI = 1).

(3) Interval Timer for FIFO Transmission

For the common FIFO in transmit mode or gateway FIFO mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFCRn.CFE bit is set to 1.
When the common FIFO in transmit mode or gateway FIFO mode is enabled, then the first message will be transmitted without considering this interval time.
The interval timer will stop counting when:

- FIFO is disabled by set the CFCRn.CFE bit to 0
- CAN channel is in CH_RESET mode.

The interval time is specified by the value of the CFCRn.TINT[7:0] bits and can be specified from 0 to 255 timer units. The timer unit can be defined based on two different clock sources for the interval timer. Select the value 0 to disable the interval timer for FIFO transmission.
The clock source can be selected by the CFCRn.ITCS bit. For the clock source the CAN bit timing clock of the related channel or a reference clock could be selected.
If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, then the interval timer is stopped for that channel.
If the reference clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.
The reference clock can be used to configure the interval time in fixed time units. It is based on the PCLKA. The GCFG.ITP[15:0] bits define the relation between the PCLKA frequency/period and the reference clock period.
Refer to Table 45.26 for configuration values of the GCFG.ITP[15:0] bits to achieve different reference clock periods based on the PCLKA frequency/period.

Table 45.26 Configuration Example for Interval Timer Prescaler

	Reference clock period		
PCLKA frequency (period)	**1 µs**	**100 µs**	**500 µs**
16 MHz (62.5 ns)	16	1600	8000
20 MHz (50 ns)	20	2000	10000
32 MHz (31.25 ns)	32	3200	16000
50 MHz (20 ns)	50	5000	25000
60 MHz (16.67 ns)	60	6000	30000
80 MHz (12.5 ns)	80	8000	40000
100 MHz (10 ns)	100	10000	50000
120 MHz (8.33 ns)	120	12000	60000
150 MHz (6.67 ns)	150	15000	—

Additionally the reference clock resolution can be specified by the CFCRn.ITR bit.
The interval time is based on the reference clock period multiplied by the configured value (×1 or ×10).
The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time.
The whole range for the separation time from 100 µs to 127 ms can be covered.
The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).
When the interval time has elapsed, the next transmission request is raised by the related transmit FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.
The next message will earliest be sent after this interval time. The Figure 45.46 shows an example timing of the internal processing.

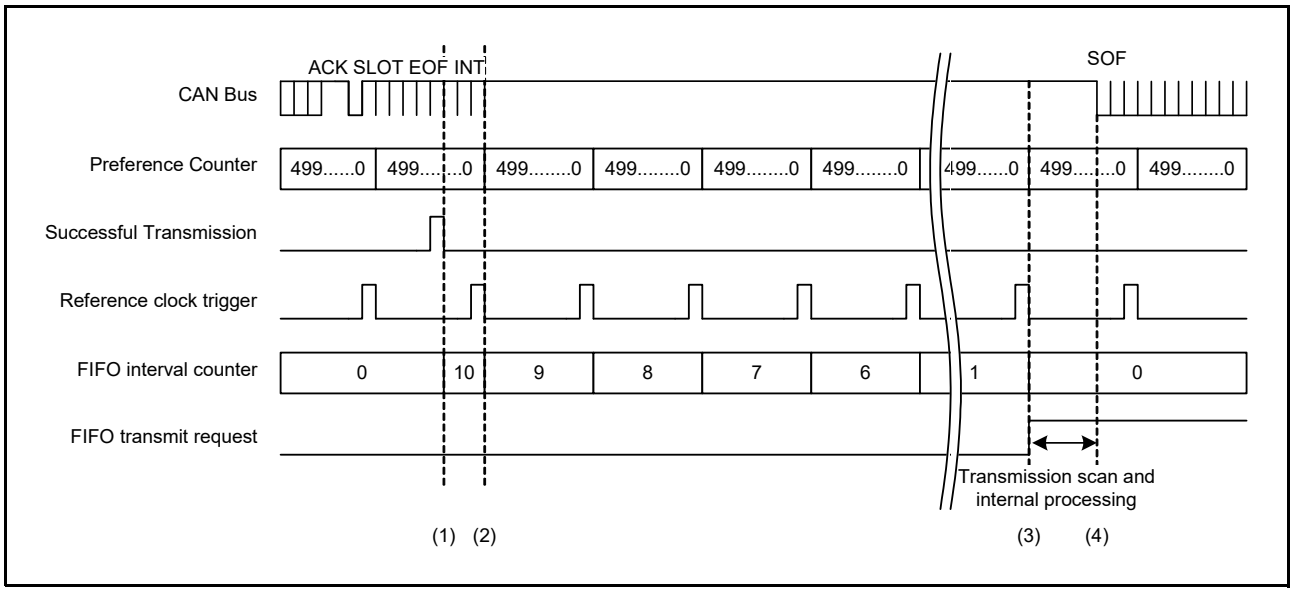


Figure 45.46 Example for Interval Processing Time

The configuration for this timing above is following:

- PCLKA frequency = 50 MHz
- Interval timer prescaler (GCFG.ITP[15:0]) = Divided by 500
- Reference clock due to the settings above = 10 µs
- Common FIFO interval timer clock source selection (CFCRn.ITCS) = 0
- Common FIFO interval timer resolution (CFCRn.ITR) = 0

- Common FIFO transmission interval (CFCRn.TINT[7:0]) = 10 counts
- Theoretical message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore the first interval is counting less or equal to one reference clock interval.
(2) With the next reference clock trigger the FIFO interval timer is decremented.
(3) When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 bit time, between internal FIFO transmit request set 3. (shown above) and actual transmission.
When multiple events occur simultaneously, such as receive scanning, internal message routing, transmit scanning on all channels, it could take up to 432 PCLKA cycles.

As shown in Figure 45.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure the CFCRn.TINT[7:0] bits to required minimum value + 1.
If transmit message buffers, transmit FIFOs or gateway FIFO are configured for transmission for the channel the real delay between two messages transmitted from a transmit FIFO can be much longer than specified by the interval time due to higher priority message transmission from these transmit message buffers or transmit FIFOs.
Figure 45.47 shows a block diagram of the FIFO interval timer.

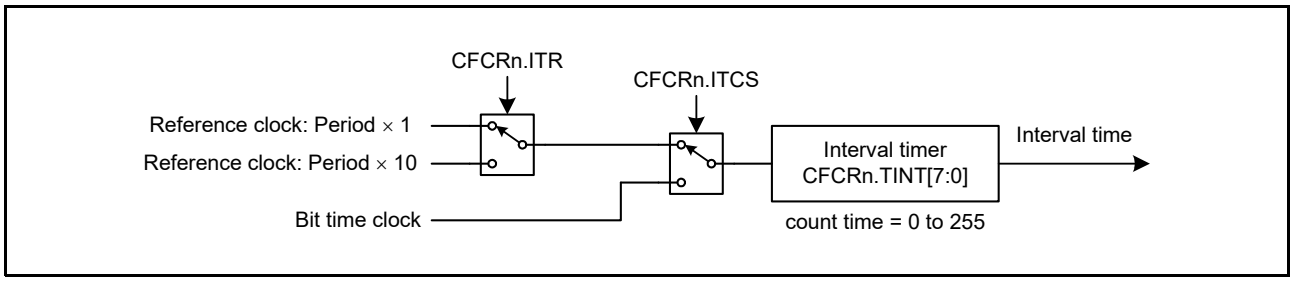


Figure 45.47 Block Diagram of FIFO Interval Timer

45.7.2.4 Transmit Queue

Each channel's enabled transmit queue consists of three to eight transmit message buffers, which are accessed via one access window. There are four transmit queues per channel. The depth in each transmit queue can be set to a value between 3 and 8.
The access window for each transmit queue (hereafter TXQ) is as follows:

- TXQ0: transmit message buffer 0
- TXQ1: transmit message buffer 7
- TXQ2: transmit message buffer 32
- TXQ3: transmit message buffer 39

For the transmit message buffer to be used as the access window, refer to the corresponding message buffer register (TMBn.HF0, TMBn.HF1, TMBn.HF2, TMBn.DF0 to TMBn.DF15).

All the messages in TXQ0 to TXQ3 enter the priority comparison for the transmission, which should be ID priority (GCFG.TPRI = 0).
The registers for TXQ are as follows:

- TXQ0: TQCR0, TQSR0, TQPCR0
- TXQ1: TQCR1, TQSR1, TQPCR1
- TXQ2: TQCR2, TQSR2, TQPCR2
- TXQ3: TQCR3, TQSR3, TQPCR3

The depth of each TXQ buffer can be configured by writing to the QDS[4:0] bits of the corresponding transmit queue configuration register. Each TXQ can set up to one of the following transmit message buffers as a queue buffer.

- TXQ0: transmit message buffer 0 to transmit message buffer 7
- TXQ1: transmit message buffer 7 to transmit message buffer 0
- TXQ2: transmit message buffer 32 to transmit message buffer 39
- TXQ3: transmit message buffer 39 to transmit message buffer 32

The available options for depth configuration of each TXQ buffer are:

- 00000b: 0 messages (transmission queue disabled)
- 00010b: 3 messages
- 00011b: 4 messages
  :
- 00111b: 8 messages

When using TXQ0 and TXQ1 at the same time, set the total number of queue depth to 8 or less.
When using TXQ2 and TXQ3 at the same time, set the total number of queue depth to 8 or less.

Do not access all transmit message buffers that configure the transmit queue, except transmit message buffer 0, transmit message buffer 7, transmit message buffer 32, and transmit message buffer 39, which function as the access window. Also, do not access the TMCRn registers that correspond to the transmit message buffers that configure the transmit queue.
When the AFLn.PTR0.RDS0 to AFLn.PTR0.RDS2 bits are set to 1 and the TQGWE bit for the corresponding transmit queue is set to 1, the received frame is stored as transmit data in the transmit queue specified as the routing destination. In gateway mode, if the transmit queue is accessed by routing when it is full, the transmit data is discarded and the TQSRn.LOST flag becomes 1.

When writing data to TXQ, check the status of TXQ before writing the data to be transmitted.
The messages stored to the transmit queue access window are internally stored to a free buffer of the transmit queue.

When the TXQ buffer is full, do not access the queue anymore. If the transmit data is written when the TXQ is full, the transmit data will be overwritten.
The transmit queue can be disabled by setting the TQE bit in the corresponding transmit queue setting register to 0. If this bit is set to 0, the TQSRn.EMPTY flag is set as described below:

- immediately: if a message from the transmit queue is not scheduled for the next transmission and is not being transmitted
- after the transmission completion, error detection on the CAN bus, loss of arbitration, or transition to CH_HALT or GL_HALT mode: if a message from the transmit queue is scheduled for transmission or is being transmitted.

Note: The transmit queue is disabled only when the TQSRn.EMPTY flag is set to 1 after setting the TQCRn.TQE bit to 0.

If there are other pending messages in the transmit queue, they will be lost, so their transmission needs to be requested again.
Before the TQCRn.TQE bit is set to 1 again, ensure that the TQSRn.EMPTY flag is set to 1 and that there is no pending abort request from the transmit queue.
When the TQE bit is set to 0, all messages in the transmit queue buffers will be lost and no further message can be stored in the transmit queue.
When a message has been stored to the transmit queue, 000000FFh must be written in to the TQPCRn register. This will set the transmit request automatically and change the internal message buffer pointer to the next free message buffer location of the transmit queue.

Note: If two messages with the same ID are stored in the transmit queue, then the order of transmission of these messages could be different from the order in which they were stored in the transmit queue.
To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the transmit queue.

For the transmit queue a dedicated interrupt can be enabled by setting the TQCRn.TQIE bit.
The interrupt mode can be configured with the TQCRn.TQIM bit either to generate an interrupt for every transmitted message or for the last transmitted message.
The transmit queue transmission request procedure after configuration is shown in Figure 45.48.

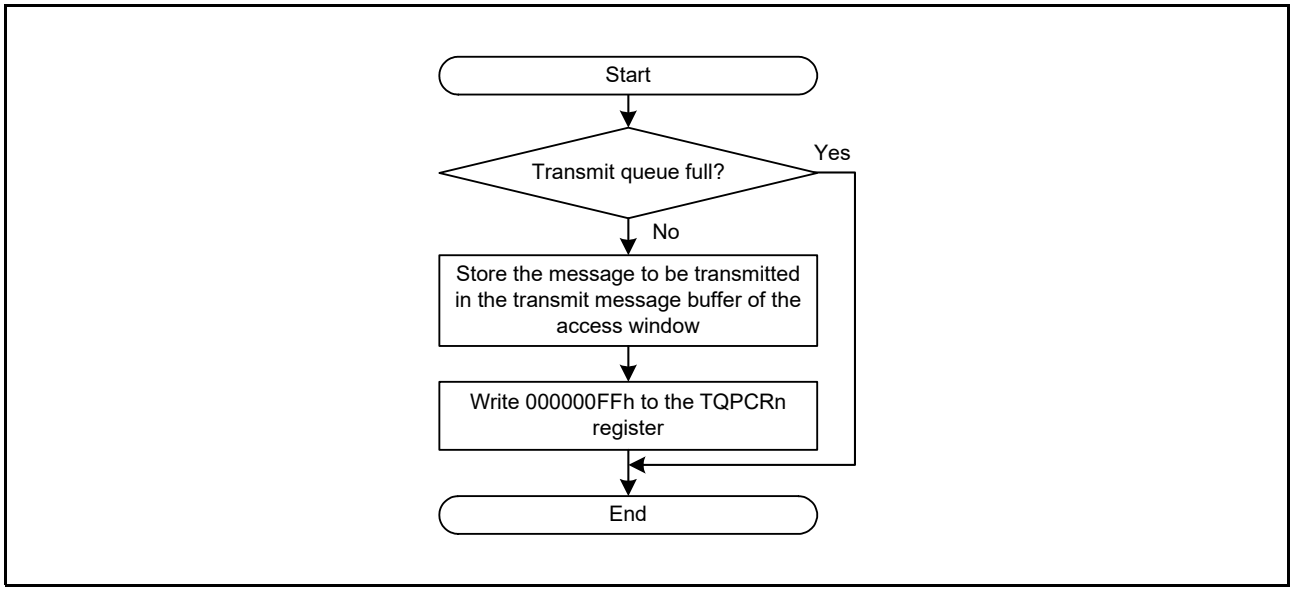


Figure 45.48 Transmit Queue Transmission Request

Table 45.27 shows the access points that can be used with transmit queues.
Do not use HW routing access, CPU access, and DTC/DMA access at the same time. Only one of the access methods is selected.

Table 45.27 Transmit Queue and Available Access Points

Transmit Queue	Access Window	Transmit Queue Depth*1	Message Buffer Direction	Hardware Routing Access Point	CPU Access Point	DMA Access Point
TXQ0	Transmit message buffer 0	0, 3 to 8	0 → 7	Yes	Yes	Yes
TXQ1	Transmit message buffer 7	0, 3 to 8	7 → 0	Yes	Yes	No
TXQ2	Transmit message buffer 32	0, 3 to 8	32 → 39	Yes	Yes	No
TXQ3	Transmit message buffer 39	0, 3 to 8	39 → 32	No	Yes	Yes

Note 1. When using TXQ0 and TXQ1 at the same time, set the total number of queue depth to 8 or less. When using TXQ2 and TXQ3 at the same time, set the total number of queue depth to 8 or less.

45.7.2.5 Transmission History

The transmission history function records the information of the successfully transmitted message in the transmission history buffer. Each channel has one transmission history buffer, and each transmission history buffer can store up to 32 transmission history entries.
The THCR.THRC bit can select whether to store information only for messages transmitted from transmit FIFO and transmit queue, or also for messages transmitted from the transmit message buffers.
When the THCR.THRGC bit is 1, information about all messages transmitted in gateway mode is stored in the transmission history.
Each transmit message can be individually configured for acceptance to the transmission history by the CFBn.HF0.THENT bit.
The message information is stored to the transmission history buffer of a CAN channel after the message is successfully transmitted on that CAN channel.
Storing to the list is not synchronized with the status of TMSRn.TXRF[1:0] flags.
Due to internal processing, the storage to the list could happen with a delay after the successful transmission indication.
Storing the transmission history data can be recognized by the condition that the THSR.THIF flag is set to 1 when the THCR.THIE bit is set to 1 or when the transmission history counter (THSR.FLVL[5:0]) is increased.
Multiple events could occur, such as reception scans and internal message routing.
Maximum delay time from setting the TMSRn.TXRF[1:0] flags to storing the transmission history data is 200 PCLKA cycles.

The transmission history records following information of the transmitted message:

- Transmitted Buffer type:
  - 001: Transmit message buffer
  - 010: Common FIFO in transmit FIFO mode (hereinafter referred to as transmit FIFO)
  - 100: Transmit queue
- Transmitted Buffer number:
  Transmit message buffer, transmit queue message buffer or transmit message buffer link for the common FIFO from which the transmission occurred. The number depends upon the buffer type, refer to Table 45.28.
- Transmitted pointer:
  Pointer set in header field 2 of transmit message (HF2.PTR[15:0])
- Transmitted timestamp:
  Message timestamp captured at capture point as set by the GFDCFG.TSCPS[1:0] bits
- Transmitted information label:
  Information label set in header field 2 of transmit message (HF2.IFL[1:0])
- Transmit gateway identification:
  When the message is transmitted through a gateway, the THACR0.GWMF bit is set to 1.

Table 45.28 Transmission History Buffer Number Entry

Transmitted Buffer Number (THACR0.BN[6:0])	Transmitted Buffer Type (THACR0.BT[2:0])		
	001b	010b	100b
	Transmit Message Buffer	Transmit FIFO	Transmit Queue
0000000b	TMB0	The value of the BN[6:0] bits corresponds to the setting of the CFCRn.LTM[4:0] bits.	The value of the BN[6:0] bits indicates the number of the transmit message buffer from which the message was transmitted.
:	:		
0000111b	TMB7		
0100000b	TMB32		
:	:		
0100111b	TMB39		
1000000b	TMB64		
:	:		
1000111b	TMB71		
1100000b	TMB96		
:	:		
1100111b	TMB103		

The transmitted pointer is used to identify which message of a transmit FIFO or transmit queue has been successfully transmitted because the transmit FIFO or transmit queue number alone is not sufficient.
Therefore, a unique number (pointer) can be attached to each transmit message stored in a transmit FIFO or transmit queue. This unique pointer should be written to the CFBn.HF2.PTR[15:0] bits for a transmit FIFO or to the TMBn.HF2.PTR[15:0] bits of the transmit queue.
When the message is successfully transmitted then this pointer is stored together with the other message related information to the transmission history and can be read via the transmitted pointer field (PTR[15:0]) in the Transmission History Access Register.
Also for normal transmit message buffers, the TMBn.HF2.PTR[15:0] bits will be stored in the transmission history. An information label is the same.
Read access to the Transmission History Access Register will be done for every single entry.
After reading one entry, 000000FFh has to be written to the THPCR register to be able to access the next entry until transmission history is empty.
Figure 45.49 shows an example flow for processing the transmission history.
The transmission history has dedicated interrupts, which can be configured with the THCR.THIM bit and enabled with the THCR.THIE bit, either to generate an interrupt when the transmission history reached a filling level of 75% or for every new transmission history entry.
Loss of transmission history is indicated by the THSR.LOST flag. Status of this flag is also shown by the GESR.THLDF flag.

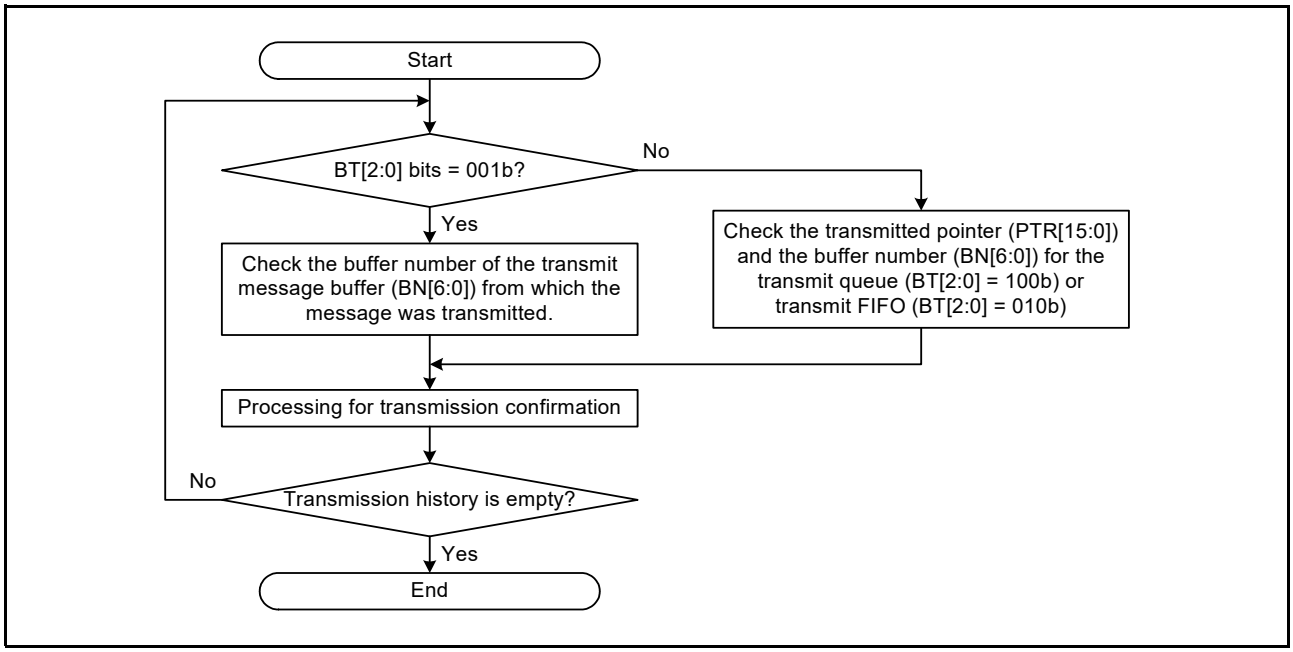


Figure 45.49 Example of Transmission History Processing Flow

45.7.2.6 Transmit Data Padding

If the data length code (DLC) of the transmit message is higher number of data bytes than the buffer size, the data bytes beyond the restricted range will be replaced by bytes with the value of CCh.

This could occur for common FIFO configured in transmit FIFO mode or gateway FIFO mode when the DLC of the transmit message is larger than the payload size set in the CFCRn.PLS[2:0] bits.

This can also occur in FD only mode, if the DLC value for a Classical CAN frame is greater than 8.

45.8 ECC Check

Message buffer RAM has ECC function of the 2-bit error detection and 1-bit error detection/correction.*1 The ECC module appends 7-bit ECC data to 32-bit RAM data.

Note 1. The ECC module cannot detect errors of 3 bits or more. In this case, the ECC module detects a 1-bit or 2-bit error, does not detect the error, or corrects the error bit to error data by configuration. If all RAM data is fixed at 0 or 1, it will be detected as a 2-bit ECC error.

45.8.1 ECC Function Setting

Figure 45.50 shows a procedure for ECC function setting.

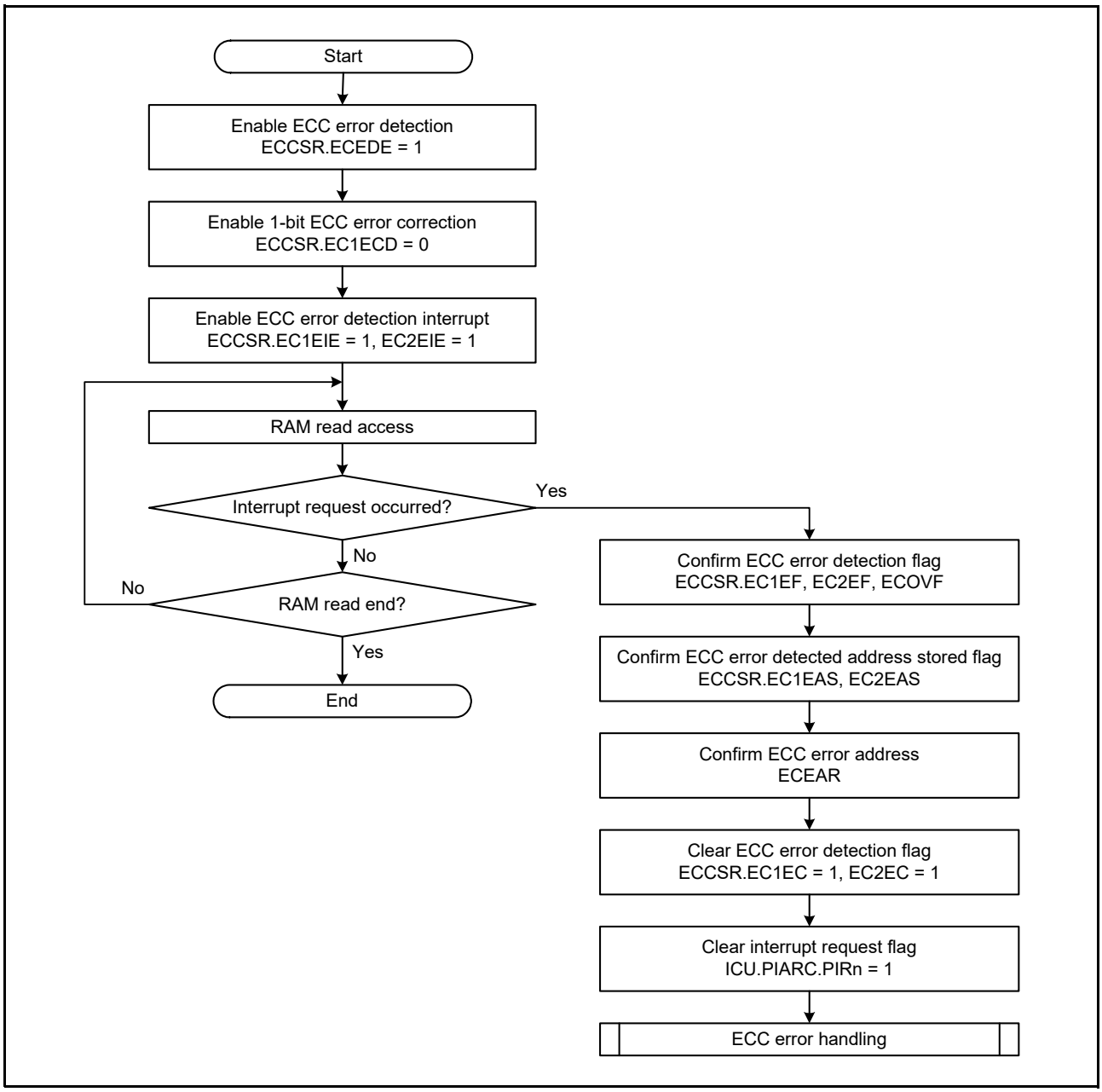


Figure 45.50 Setting Procedure for ECC Function

45.8.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 45.51 shows a procedure for ECC decoder testing.

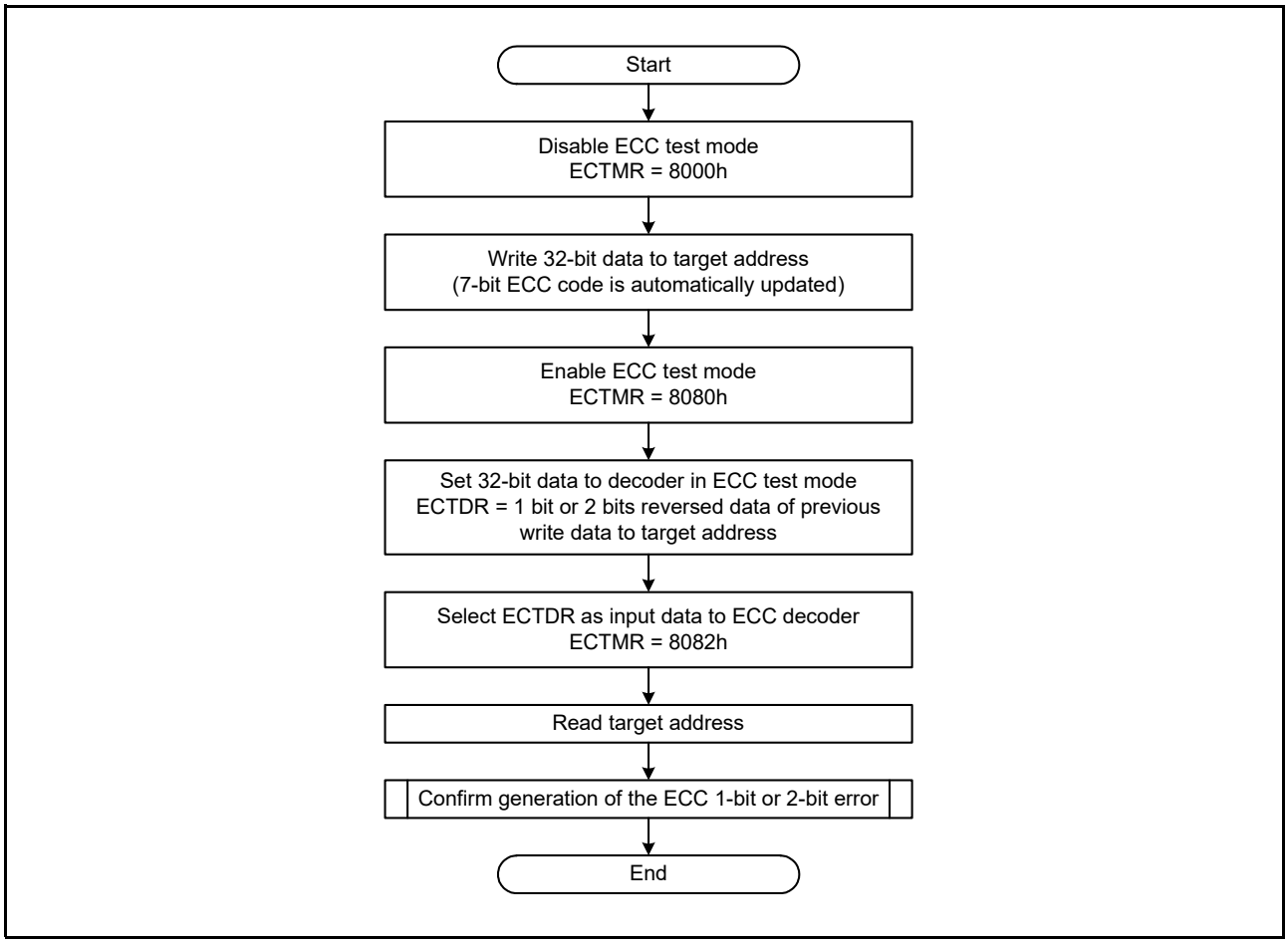


Figure 45.51 Testing Procedure for ECC Decoder

45.9 Test Mode

The CANFDG module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFDG module in the test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.
The test modes can be broadly split into two groups:

- Channel specific test modes
- Global test modes

45.9.1 Channel Specific Test Modes

CAN channel can be configured into following test modes:

- Basic test mode
- Listen-only mode
- Self test mode 0 (external loop-back mode)
- Self test mode 1 (internal loop-back mode)
- Restricted operation mode

45.9.1.1 Basic Test Mode

The basic test mode should be used when a particular test setting needs to be enabled other than when in listen-only and self-test modes.

45.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In this mode, the CAN channel can receive valid data frames and valid remote frames. However, it transmits only recessive bits on the CAN bus and is not allowed to transmit. If the CAN engine is required to transmit a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN engine monitors this dominant bit, although the CTX pin remain in recessive state.
This mode can be used for bit rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.
In this mode, it is not permitted to request transmission from any transmit message buffer, transmit queue, or common FIFO.

Note: If messages are stored in a common FIFO set to gateway FIFO mode or to a transmit queue set to gateway mode, make sure the transmission channel is not in listen-only mode to avoid requests to transmit to the channel.

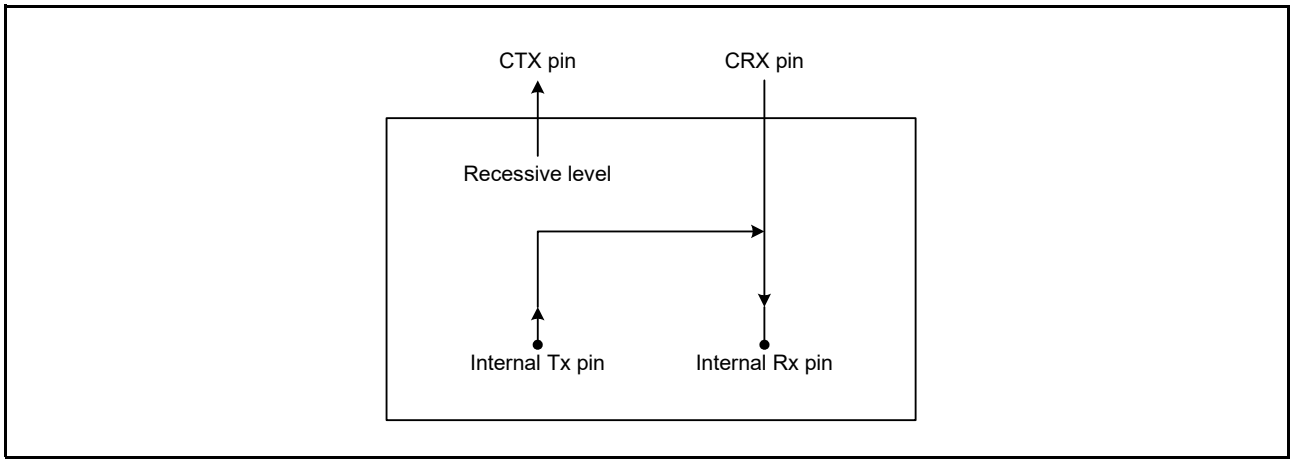


Figure 45.52 Listen-only Mode Configuration

45.9.1.3 Self Test Mode 0 (External Loop-Back Mode)

In self test mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into its receive message buffers.
To be independent of external stimulation the engine generates its own acknowledge bit. This test can be used for CAN transceiver tests.
The CRX and CTX pins should be connected to the transceiver.

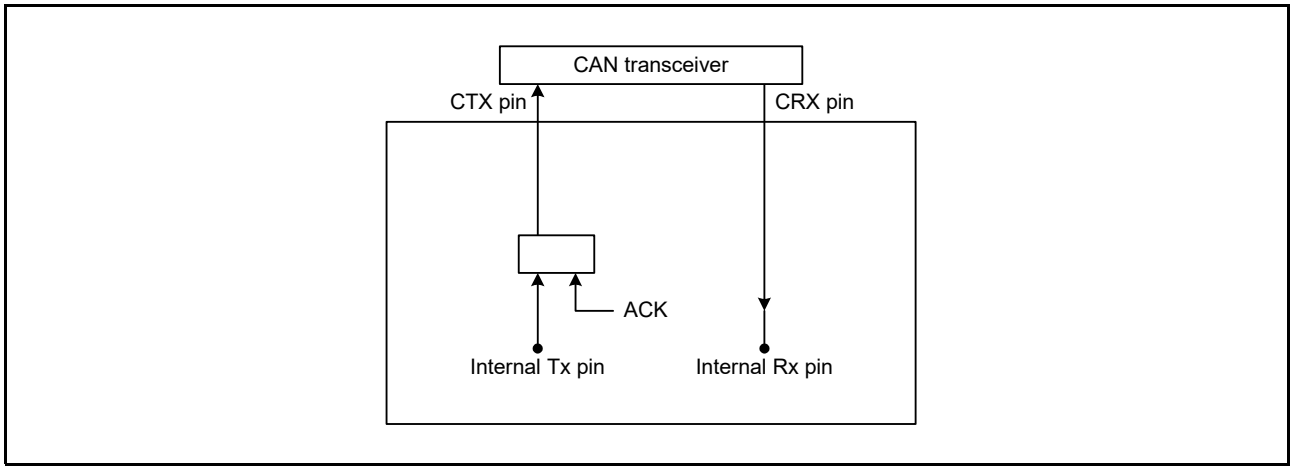


Figure 45.53 Self Test Mode 0 Configuration

45.9.1.4 Self Test Mode 1 (Internal Loop-Back Mode)

In self test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent of external stimulation the CAN engine generates its own acknowledge bit. In this mode the CAN engine performs an internal feedback from internal Tx pin to internal Rx pin. The actual input level of the CRX pin is disregarded by the CAN engine.
The CTX pin is driven high. The CRX and CTX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the Internal CAN bus communication line.

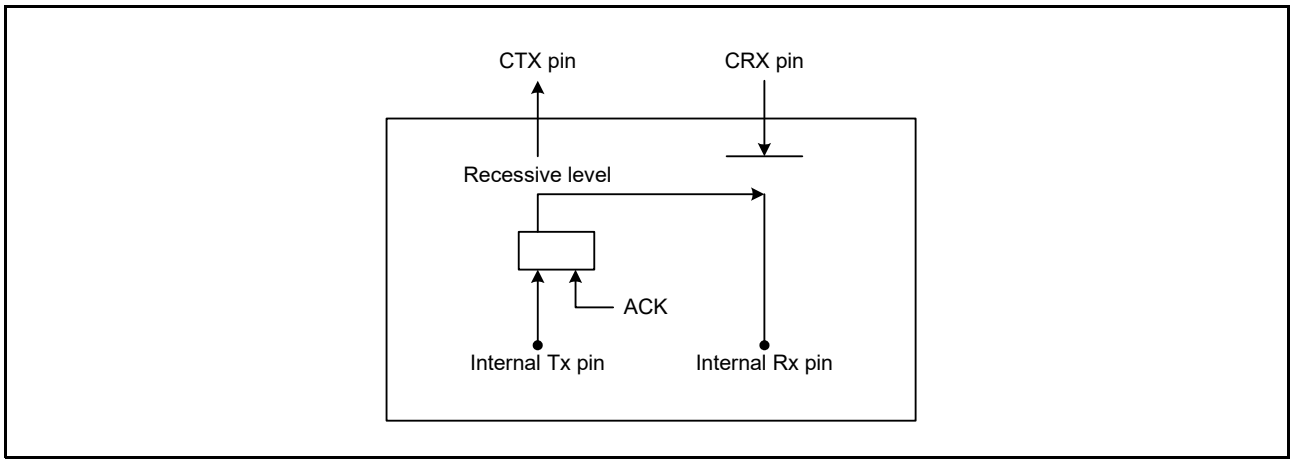


Figure 45.54 Self Test Mode 1 Configuration

45.9.1.5 Restricted Operation Mode

In restricted operation mode the CAN node is able to receive valid data and remote frames generating the acknowledge bit.
Active error and overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.
Moreover the receive error counter (CHSR.REC[7:0]) and transmit error counter (CHSR.TEC[7:0]) are frozen independently of the occurrence of errors.
The mode is specified as in ISO 11898-1; however it is permitted to set any transmit requested.

45.9.2 Global Test Modes

The CANFDG module can be configured into following test modes:

- RAM test mode
- Internal CAN bus communication mode
- CRC error test

The following test mode is protected by a special software procedure to enable the mode. This software procedure enables the write access to the test mode by specific unlock key, the related unlock key can be seen in the table below:

Table 45.29 Test Mode Unlock Key

Test Mode	Unlock Key 1	Unlock Key 2
RAM Test Mode	00007575h	00008A8Ah

If the software sequence of the two consecutive unlock key write accesses is interrupted by any other write access to the registers or if incorrect data is written to the Global Test Mode Lock Key Register then the corresponding test mode cannot be set and the sequence should be re-started.
After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism resets and the test mode enable bit cannot be set and then the unlock sequence should be restarted.

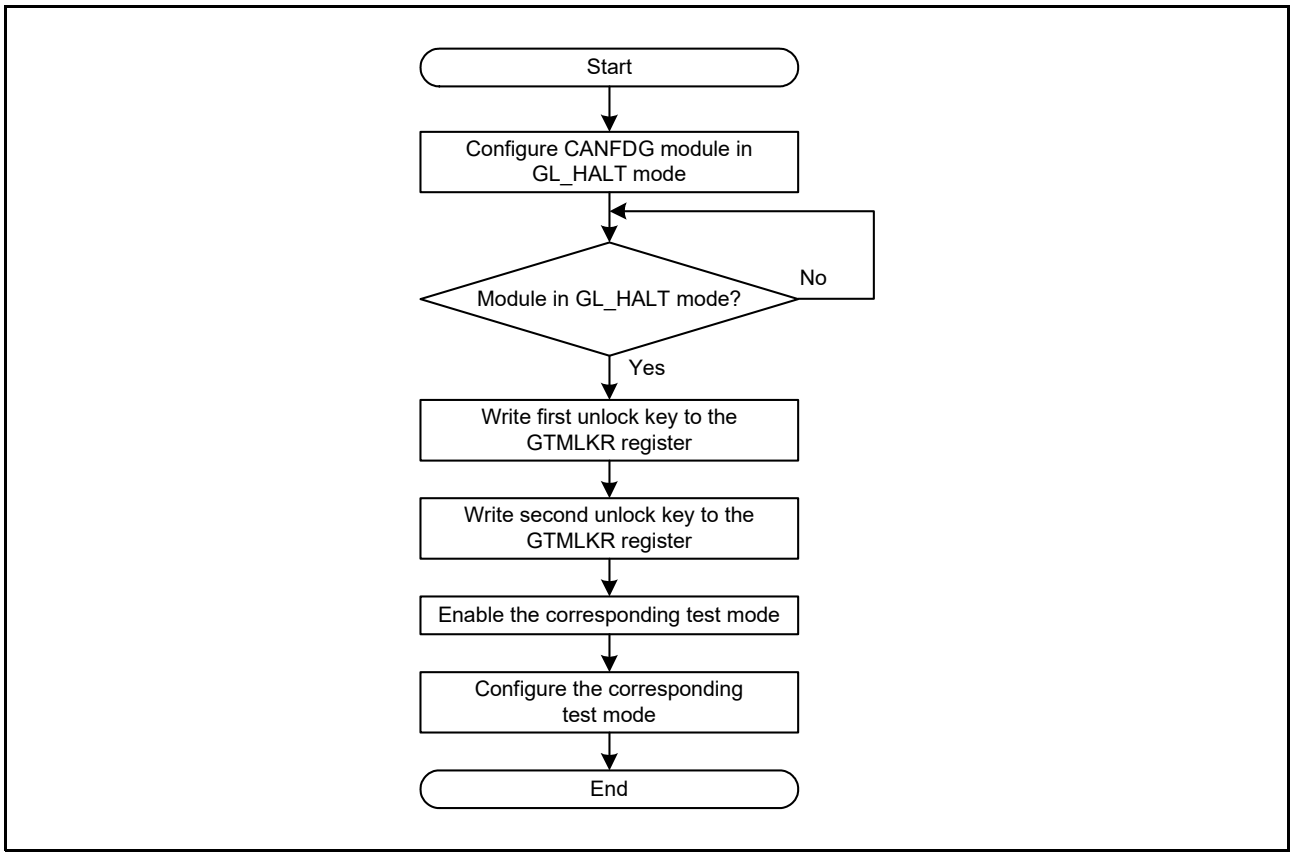


Figure 45.55 Unlock Software Protection Routine

45.9.2.1 RAM Test Mode

The CANFDG module can be configured in RAM test mode by setting the GTMER.RTME bit when the corresponding lock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after MCU reset. Hence, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFDG module is in RAM test mode.

In this mode, the RAM area is split into number of pages of 256 bytes each. Which can be accessed via RTPARk register (k = 0 to 63).
The page should be selected for read/write access by writing to the GTMCR.RTPS[9:0] bits. Then, data can be read from or written in to the RAM Test Page Access Register.
Figure 45.56 shows the structure of the pages in the RAM when performing a RAM test mode.

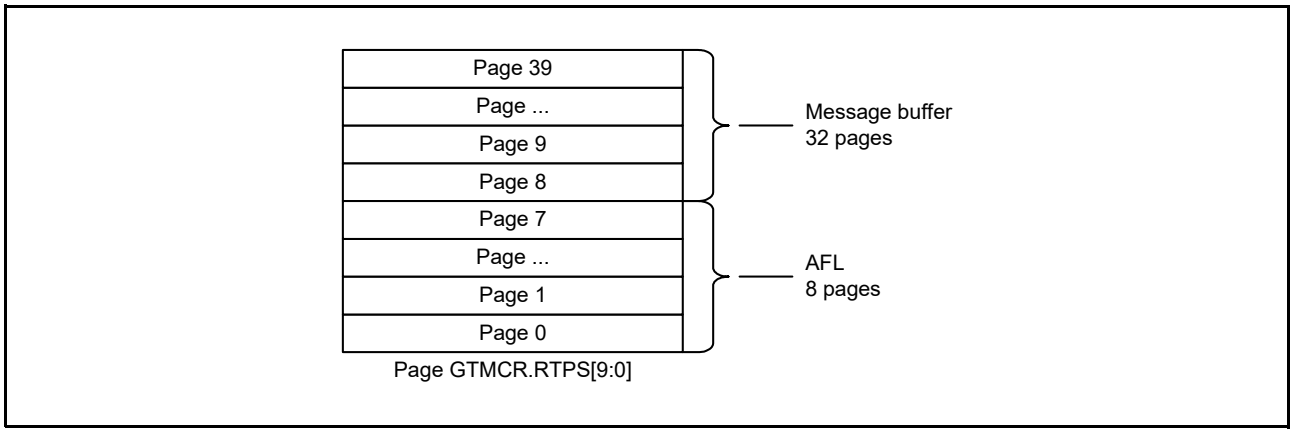


Figure 45.56 RAM Page Structure

The total available RAM size is, 2048 bytes for the AFLRAM and 8128 bytes for the message buffer RAM.
AFLRAM 0/1 can be treated as one RAM in RAM test mode.
Total number of pages for the RAMs and the GTMCR.RTPS[9:0] values are calculated in the following way:

Total number of pages = ceil (total RAM size in bytes / number of bytes per page)

- AFLRAM:
  Total number of pages = ceil (2048 / 256) = 8 pages
  GTMCR.RTPS[9:0] = 0 to 7
- Message buffer RAM:
  Total number of pages = ceil (8128 / 256) = 32 pages
  GTMCR.RTPS[9:0] = 8 to 39

Figure 45.57 shows the software flow for RAM test mode.

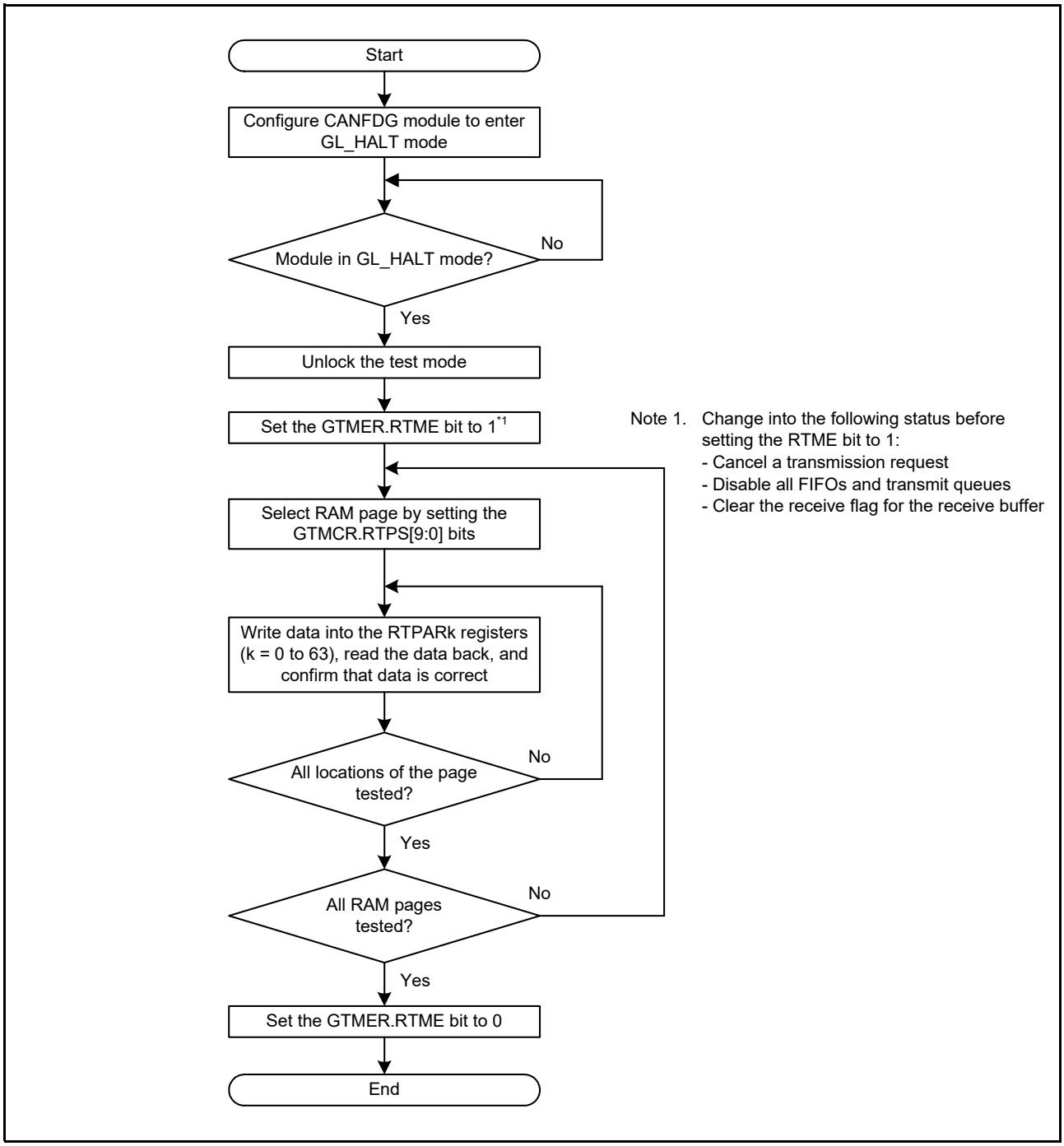


Figure 45.57 RAM Test Mode Software Flow

To exit RAM test mode, set the GTMER.RTME bit to 0.
The GTMER.RTME bit is automatically set to 0 when the CANFDG module enters GL_RESET mode from the test mode.

45.9.2.2 Internal CAN Bus Communication Test Mode

The CANFDG module can be configured in internal CAN bus communication test mode by setting the GTMER.ICBTME bit to 1. This is a special test mode in which the CAN channels are internally connected to create a CAN cluster within the CANFDG module.
Use the following procedure to enter internal CAN bus communication test mode:

(1) Set all channels to CH_HALT mode and confirm that all channels have transitioned to CH_HALT mode (GL_HALT mode).
(2) Set the GTMCR.ICBTMCi bit to select the channel on which to test the internal CAN bus communications.
(3) Set the GTMER.ICBTME bit to 1.
(4) Confirm that the GTMER.ICBTME bit is set to 1.

In this mode, the CTX pin outputs of channels set to internal CAN bus communication mode are logically ORed and connected to the internal CAN bus. This output is connected to the CRX inputs of all desired channels to create a CAN cluster within the CANFDG module. During this test mode the channel is isolated from the external CAN bus.
The AFL, receive message buffer, FIFO buffer, transmit message buffer and various registers can be configured as usual and communication between the channels can be started.
Set channels that do not participate in internal CAN bus communication to CH_HALT mode.

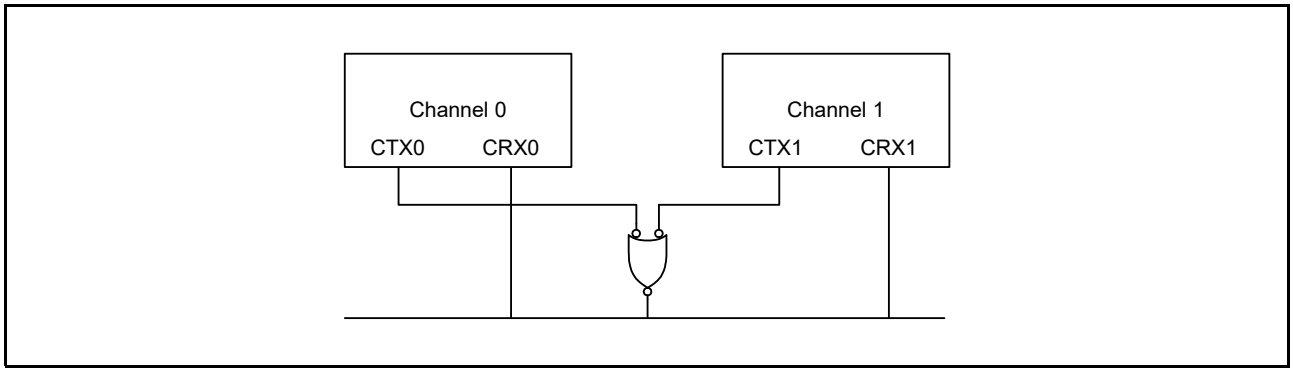


Figure 45.58 Internal CAN Bus Connections

(1) CRC Error Test

CRC error testing can be performed by enabling the bit-flip test function at the receiver node. Enabling the bit flip test allows you to flip the first bit of the received stream (the first bit of the ID).
To perform a CRC error test, follow the steps below after setting the CANFDG module to the internal CAN bus communication test mode.
In the following sequence, channel x is the reference transmitter node and channel y is the receiver node (x = 0, y = 1 or x = 1, y = 0).

1. Configure the channel x node to transmit a reference message.
2. Set the CANFDy.CHCR.BFT bit to 1 to invert the first bit of the bit stream input from channel x.
3. Set the CANFDx.TMCRn.TXRQ bit to 1.
4. Wait for the channel error interrupt on channel y to become 1.
5. Read either the CANFDy.CHESR.CRC15[14:0] bits (Classic CAN frame) or the CANFDy.FDCRC.CRC21[20:0] bits (CAN FD frame). This value must be different from the received CRC value of the reference message from channel x.
6. Confirm that the CANFDy.CHESR.CEDF bit is 1.

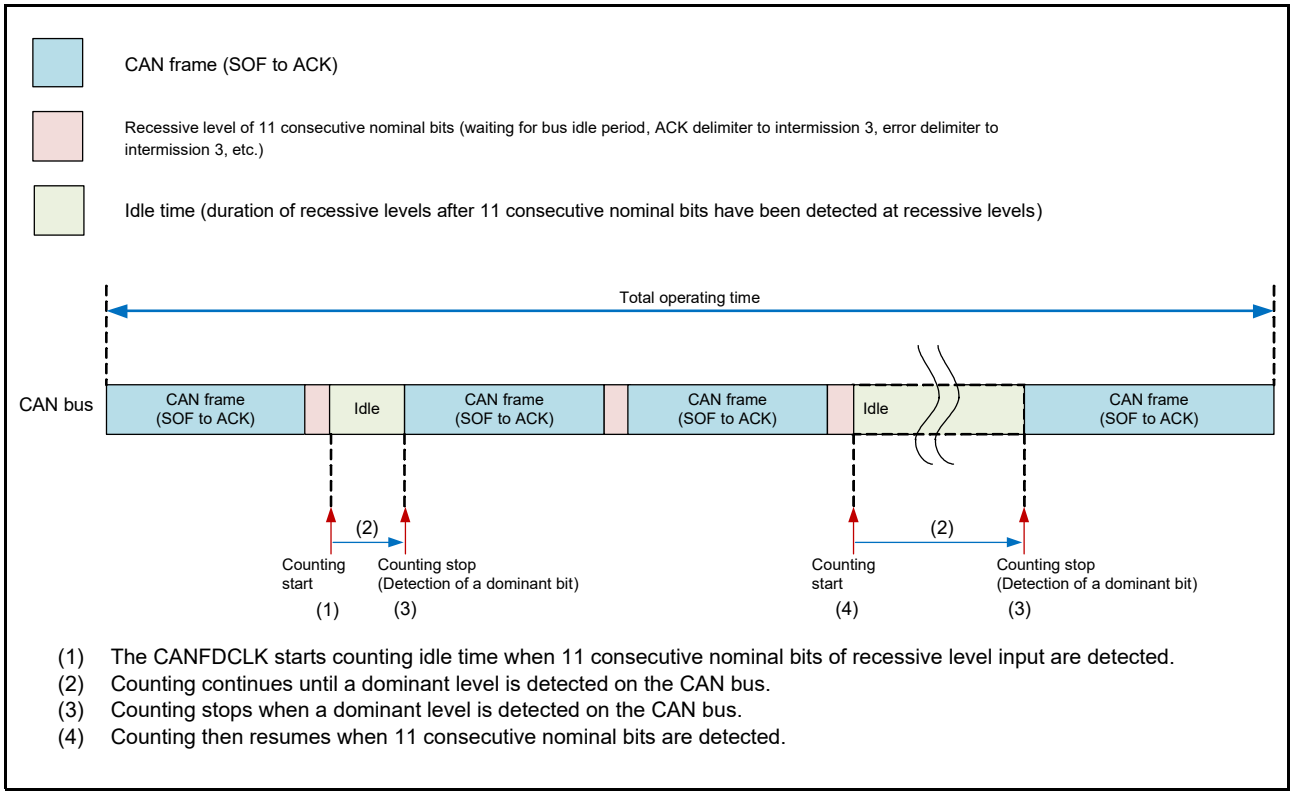
The CRC generator logic is shared between receive and transmit, so there is no need to create a separate transmit CRC error test.

45.10 Bus Load Measurement

The CAN bus idle time can be measured using CANFDCLK or CANFDMCLK. The bus load is calculated based on the measurement results.

45.10.1 How to Count the CAN Bus Idle Time

Figure 45.59 shows a conceptual diagram of measuring CAN bus idle time.



(1) The CANFDCLK starts counting idle time when 11 consecutive nominal bits of recessive level input are detected.
(2) Counting continues until a dominant level is detected on the CAN bus.
(3) Counting stops when a dominant level is detected on the CAN bus.
(4) Counting then resumes when 11 consecutive nominal bits are detected.

Figure 45.59 Conceptual Diagram of Measuring CAN Bus Idle Time

45.10.2 Measurement Procedure

The following procedure shows the steps for measuring idle time of the CAN bus.

(1) Set the channel to be measured to CH_OPERATION mode and confirm that the transition was successful.
(2) Write 1 to BLCCR.BLCE bit to set the measuring counter to operating mode.
(3) Write 1 to BLCCR.BLCLD bit to reset the bus load count.
(4) Starts counting bus idle time when it detects 11 consecutive recessive bits at the nominal bit rate.
(5) When a dominant is detected, the bus load counter stops.
(6) Starts counting bus idle time when it detects 11 consecutive recessive bits at the nominal bit rate.
(7) Write 1 to the BLCCR.BLCLD bit to reset the counter register and load the counter value into the BLCR register.
(8) Reads the value of the BLCR register.

The counter is enabled when the channel to be measured is in CH_OPERATION mode. If the channel is in CH_RESET mode, the counter does not operate. Additionally, accurate measurements cannot be made in test mode.
Write 0 to the BLCCR.BLCE bit to stop the counter.
To initialize the counter, write 1 to the BLCCR.BLCLD bit. The counter is reset and the counter value is loaded into the BLCR register at the same time.
The lower three bits of the BLCR register are fixed to 0.
Based on the counter value, software calculates the CAN bus load according to the following formula:

$$\text{Bus occupancy rate} = \frac{\text{Total operating time} - \text{total idle time}}{\text{Total operating time}} = \frac{\text{Bus operating time}}{\text{Total operating time}}$$

Total idle time: Value read from the BLCR register × CANFDCLK clock cycles
Total operating time: BLCCR.BLCLD bit setting interval

A calculation example under the following conditions is shown below.

- Nominal bit rate: 1 Mbps
- CANFDCLK: 40 MHz (= 25 ns)
- BLCCR.BLCLD bit setting interval: cycle of 1 ms
- Read value of the BLCR register: 4E20h (20000)

$$\frac{\text{Total operating time} - \text{total idle time}}{\text{Total operating time}} = \frac{1000000\text{ns} - 20000 \times 25\text{ns}}{1000000} = 50\%$$

45.11 Interrupts and DTC/DMA Requests

45.11.1 CANFDG Interrupts

The CANFDG module generates several interrupts. The interrupt output, which is connected to the interrupt controller, can be controlled by the corresponding interrupt enable bit.
The status flags will be set independently of these enable bits.
The channel transmit interrupt has an additional status flag register; these status bits will only be set when the corresponding interrupt enables are set.
This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.
The interrupts in the CANFDG module can be classified into two groups: global interrupts and channel interrupts.

(1) Global Interrupts

The CANFDG module can generate the following two global interrupts.

- Receive FIFO interrupt
- Global error interrupt

(2) Channel Interrupts

The CAN channel can generate the following three channel interrupts.

- Channel transmit interrupt
  (a) Successful transmission interrupt
  (b) Transmission abort interrupt
  (c) Transmit queue interrupt
  (d) Common FIFO transmission interrupt
  (e) Transmission history interrupt
- Channel receive interrupt
  (a) Common FIFO receive interrupt
  (b) Transmit queue receive interrupt
- Channel error interrupt

To clear each interrupt request, clear all flags set to 1 from among the sources for which interrupts are enabled. The interrupt request can also be cleared by setting all corresponding interrupt enable bits to 0.

Table 45.30 lists the interrupt sources for CANFDG module.

Table 45.30 Interrupt Sources

Interrupt Name			Interrupt Source Flag
Global interrupts	Receive FIFO interrupt (RFRI)	Receive FIFO interrupt	RFSRn.RFIF
		Receive FIFO full interrupt	RFSRn.RFFIF
	Global error interrupt (GLEI)		GESR.DEDF GESR.MLDF GESR.THLDF GESR.PODF GESR.TQMLDF GESR.CFOWDF
Channel interrupts	Channel transmit interrupt (CHTI)	Transmission successful interrupt*1	TMSRn.TXRF[1]
		Transmission abort interrupt*1	TMSRn.TXRF[1:0] (TXRF[1:0] = 01b)
		Transmit queue interrupt	TQSRn.TQIF
		Transmission history interrupt	THSR.THIF
		Common FIFO transmission interrupt	CFSRn.CFTIF
		Common FIFO one frame transmit interrupt	CFSRn.CFOTIF
		Transmit queue one frame transmit interrupt	TQSRn.TQOTIF
	Channel receive interrupt (CHRI)	Common FIFO receive interrupt	CFSRn.CFRIF
		Common FIFO one frame reception interrupt	CFSRn.CFORIF
		Common FIFO full interrupt	CFSRn.CFFIF
		Transmit queue one frame receive interrupt	TQSRn.TQORIF
		Transmit queue full interrupt	TQSRn.TQFIF
	Channel error interrupt (CHEI)		CHESR.BEDF CHESR.EWDF CHESR.EPDF CHESR.BOEDF CHESR.BORDF CHESR.OLDF CHESR.BLDF CHESR.ALDF FDSTS.ECOV FDSTS.SCOV FDSTS.TDCV

Note 1. These interrupts are only generated for transmit message buffers that are not part of a valid transmit queue and are not linked to a common FIFO. The common FIFO and the transmit queue have other interrupts respectively.

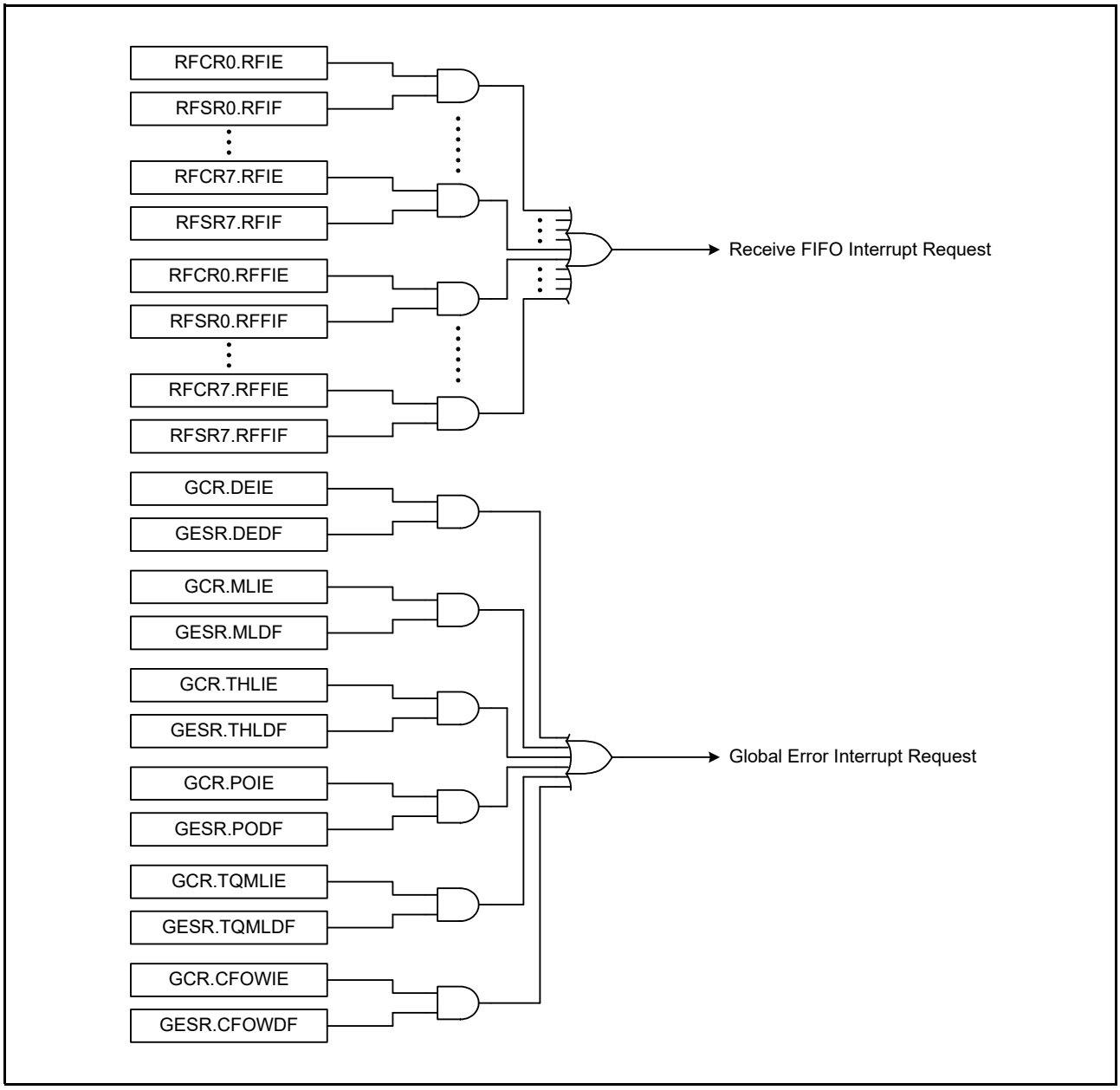


Figure 45.60 Global Interrupt Block Diagram

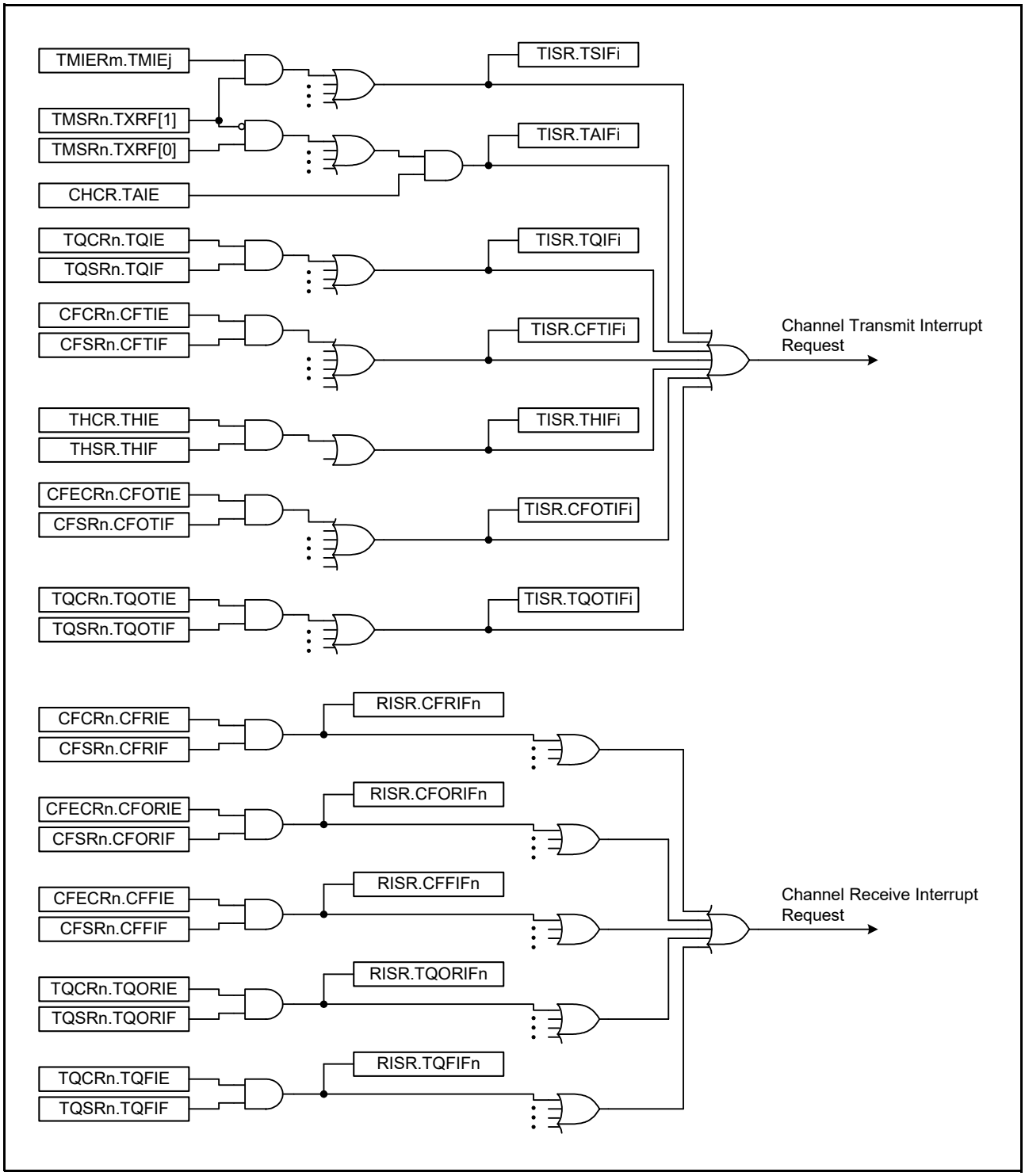


Figure 45.61 Channel Interrupt Block Diagram (1)

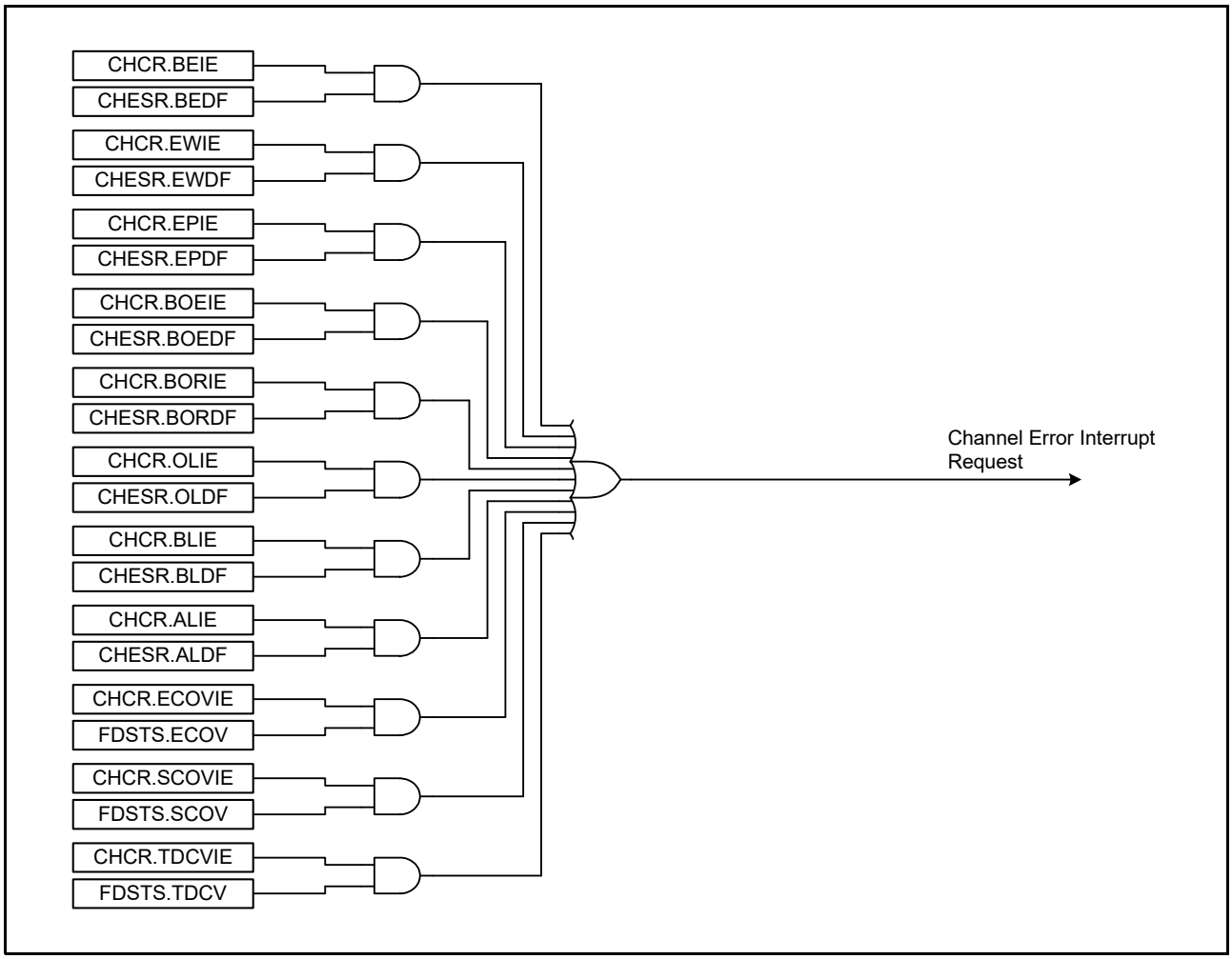


Figure 45.62 Channel Interrupt Block Diagram (2)

45.11.2 ECC Interrupts

The ECC decoder generates ECC error interrupts (AFLRAM0EI, AFLRAM1EI, MBRAMEI). There are three types of interrupt sources for the ECC error interrupt:

- 1-bit ECC error
- 2-bit ECC error
- ECC overflow.

45.11.3 DTC/DMA Transfer Requests

(1) Receive DTC/DMA Transfer

The CANFDG module has message buffers that can be read by DTC/DMA transfer:

- Eight receive FIFO message buffers
- Two common FIFO message buffers (common FIFO 0, common FIFO 3)

A DTC/DMA transfer request is generated when the DTCR.RFDTEn (n = 0 to 7), or CFDTEi (i = 0, 1) bits are set to 1 and the corresponding FIFO is not empty.
For FIFOs with DTC/DMA transfer enabled, disable receive FIFO interrupts (the RFCRn.RFIE (n = 0 to 7) or CFCRn.CFRIE bits (n = 0 to 5) are set to 0).

When the data of the specified payload size (the RFCRn.PLS[2:0] bits (n = 0 to 7) or CFCRn.PLS[2:0] bits (n = 0 to 5)) is read*1, the FIFO read pointer is automatically updated.
When DTC/DMA transfer is permitted, do not write to the FIFO pointer control register (RFPCRn (n = 0 to 7), or CFPCRn (n = 0 to 5)).

Note 1. The DTC/DMA should read the exact length of the specified data payload size (the RFCRn.PLS[2:0] or CFCRn.PLS[2:0] bits).

The permission for DTC/DMA transfer (the DTCR.RFDTEn (n = 0 to 7), or CFDTEi (i = 0, 1) bits) can be set to 1 at any time. Figure 45.63 shows the DTC/DMA transfer configuration flow.

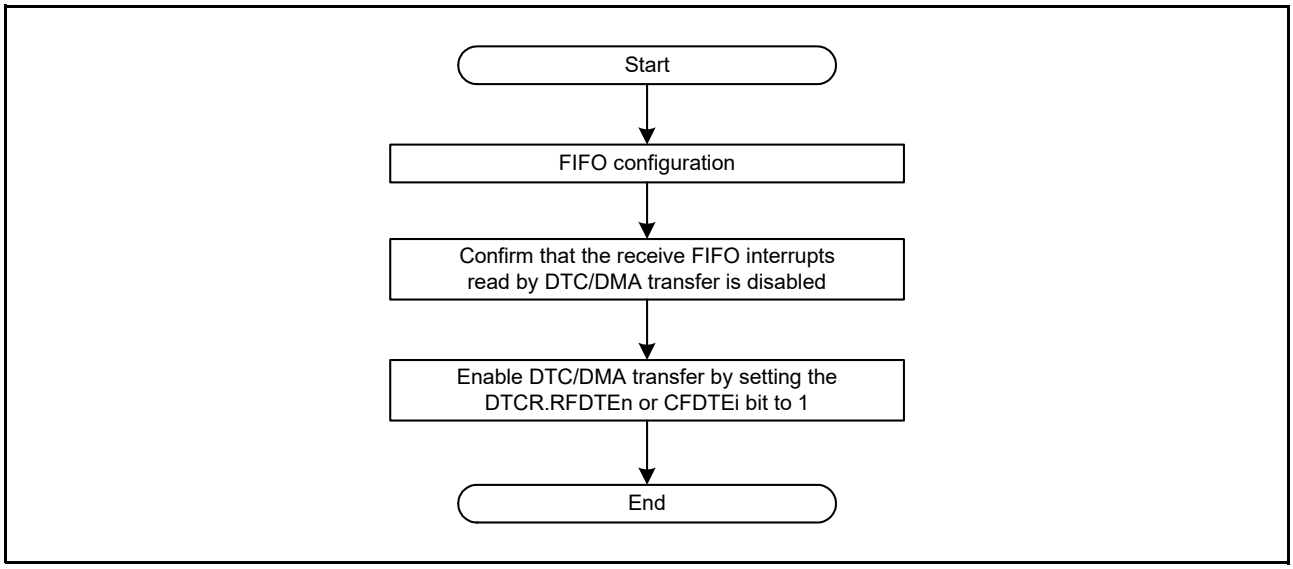


Figure 45.63 DTC/DMA Transfer Enable Flow

To disable the DTC/DMA transfer, set the corresponding DMA transfer enable bit (the DTCR.RFDTEn or CFDTEi bit)

to 0. If the disable is made during a DTC/DMA transfer, wait until the ongoing transfer is complete before performing the following operations. The transfer status can be confirmed by the DTSR.RFDTSn (n = 0 to 7) or CFDTSi (i = 0, 1) bit. Figure 45.64 shows the DTC/DMA transfer disable flow.
When the DTC/DMA transfer is disabled then consider what to do with the remaining or new incoming messages to this particular reception FIFOs. When the FIFO is not disabled then reception to the FIFO will continue.

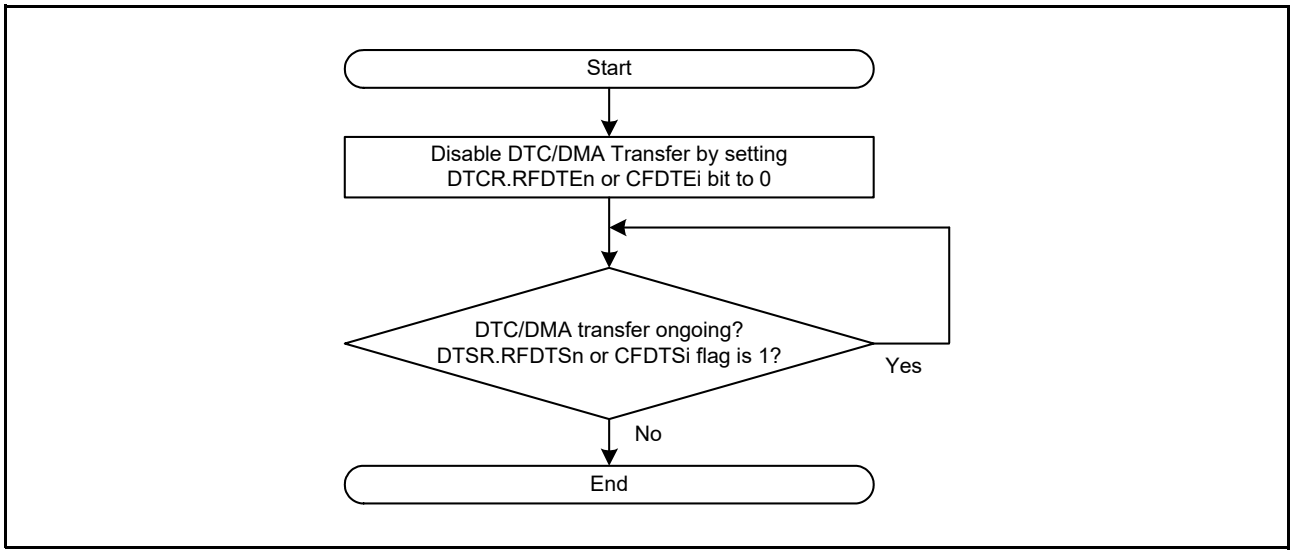


Figure 45.64 DTC/DMA Transfer Disable Flow

(2) Transmit DTC/DMA Transfer

The CANFDG module has message buffers to which data can be written by DTC/DMA transfer:

- Four transmit queue buffers
- Two common FIFO message buffers (common FIFO 2, common FIFO 5)

When the TDTCR.TQDTEi0, TQDTEi3 bit (i = 0, 1) or CFDTEi bit (i = 0, 1) is set to 1, a transmit message can be transferred to the corresponding transmit queue or common FIFO using the DTC/DMAC.
To transfer data to a transmit queue or common FIFO using DTC/DMAC transfer, follow the procedure below.

1. Confirm that the transmit queue or common FIFO is not full.
2. After preparing a message to be transmitted, enable DTC/DMA transfer.
   When using the common FIFO, transmit data is written to the CFBn.HF0, CFBn.HF1, CFBn.HF2, and CFBn.DFp registers. When using a transmit queue, transmit data is written to the TMBn.HF0, TMBn.HF1, TMBn.HF2, and TMBn.DFp registers.
3. For common FIFOs, the common FIFO pointer is automatically incremented when the DTC/DMAC writes the last data of the payload size set by the CFCRn.PLS[2:0] bits.
   For transmit queues, writing 64 bytes of payload data automatically increments the transmit queue pointer. If the payload size is less than 64 bytes, dummy data must be added to make it 64 bytes.

Note: DTC/DMA transfers can only be written in 32-bit units.

To disable DTC/DMA transfers, set the corresponding DMA transfer enable bit (TDTCR.TQDTEi0, TQDTEi3 or CFDTEi bit) to 0.

45.12 Usage Notes

45.12.1 Setting the Module Stop Function

The CANFDG operation can be disabled or enabled by module stop control register C (MSTPCRC). After a reset, the CANFDG module is stopped. The registers of the module become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

45.12.2 Notes on the DLL Clock Settings

Set the DLL clock to be used before releasing the module-stop state. For details on the DLL clock settings, refer to section 9, Clock Generation Circuit.
Release the module-stop state after the oscillation of the DLL clock has been stabilized.

45.12.3 Note on Configuration of Receive Message Buffers and FIFO Buffers

The maximum memory size available for receive message buffers and FIFO buffers is 4864 bytes.
For example, if all payload sizes are set to 64 bytes, each message size becomes 76 bytes, so the total number of messages must be 64 or less. If 32 receive message buffers are reserved, the total FIFO depth must be 32 or less. Operation is not guaranteed if the setting exceeds 4864 bytes.

45.12.4 Transition to Software Standby Mode

To reduce power consumption, power to the acceptance filter list RAMs and the message buffer RAM is turned off in software standby mode. Therefore, the respective sets of operations listed below must be performed before a transition to and after a return from software standby mode.

(1) Before a Transition to Software Standby Mode

Before a transition to software standby mode, perform the following operations.

1. Put the CANFDG module into GL_SLEEP mode.
2. Read the GTMLKR register and verify that its value is 00000000h.
3. Perform the following operations for AFLRAM0, AFLRAM1, and the MBRAM, respectively.
   (a) Set the ECCSR.EC1EC and ECCSR.EC2EC bits to 1 to clear the ECC error detection flags.
   (b) Set the ECCSR.ECEDE bit to 0 to disable ECC error detection.

(2) After a Return from Software Standby Mode 1

After a return from software standby mode 1, the CANFDG module is not reset, meaning that the RAMs are not initialized. Use RAM test mode to initialize the RAMs. After having initialized the RAMs, initialize the CANFDG module by following the standard procedure. For details on RAM test mode, refer to section 45.9.2.1, RAM Test Mode.

(3) After a Return from Software Standby Mode 2

After a return from software standby mode 2, the CANFDG module is reset. Initialize the CANFDG module through the same procedure as when the MCU is released from the reset state.

46. CAN FD Module (CANFD)

46.1 Overview

This MCU implements one channel of the CAN FD (Controller Area Network with Flexible Data Rate) module that complies with the ISO 11898-1:2015 standard.

Table 46.1 lists the specifications of the CANFD module, and Figure 46.1 shows a block diagram of the CANFD module.

Table 46.1 CANFD Module Specifications

Item	Description
Protocol	ISO 11898-1:2015 compliant
Data transfer rate	Arbitration phase: up to 1 Mbps Data phase: up to 8 Mbps*1, *2
Operating frequency*3	Register block: up to 150 MHz (PCLKA) Message buffer RAM: up to 300 MHz (PCLKE)
Operating clock for data link layer (DLL clock)	Up to 80 MHz (either CANFDMCLK or CANFDCLK can be selected)
Frame types	Classic CAN (CAN 2.0) • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID) • Remote frame in base format (11-bit ID) • Remote frame in extended format (29-bit ID) CAN FD*1 • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID)
Data length	Classic CAN: 0 to 8 bytes CAN FD: 0 to 8, 12, 16, 20, 24, 32, 48, or 64 bytes*1
Message buffers	• 16 receive message buffers • Four transmit message buffers • One transmit queue Messages can be automatically transferred to the transmit queue.
FIFOs	Variable size FIFO buffers • Two receive FIFOs • One common FIFO that can be configured as a receive FIFO or transmit FIFO
Automatic transmission interval adjustment	Available when the common FIFO is configured as a transmit FIFO The interval between messages sent from the FIFO can be adjusted.
Acceptance filter	Filterable in the following fields: • IDE bit (base format, extended format, or both) • ID field • RTR bit (data frame or remote frame) (only for Classic CAN) • DLC field (data length) Protection function when the payload size is exceeded Acceptance filter list (AFL) entries can be updated during communication.
Software support	Label information is automatically added to received messages
Timer	Transmission and reception timestamp function
Power down function	Module start/stop function for each CAN node (CH_SLEEP and GL_SLEEP mode) Transition to module stop state is possible.
RAM	RAM with ECC protection
Protect Zone filter	Security attribute can be set.*4

Note 1. This is only available for products that support the CAN FD protocol.
Note 2. The bit rate for communications depends on the board design and external environment. Determine it following sufficient evaluation.
Note 3. The frequency ratio of PCLKE and PCLKA should be 2 : 1. Also, the frequency of PCLKA should be equal to or higher than that of the DLL clock.
Note 4. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

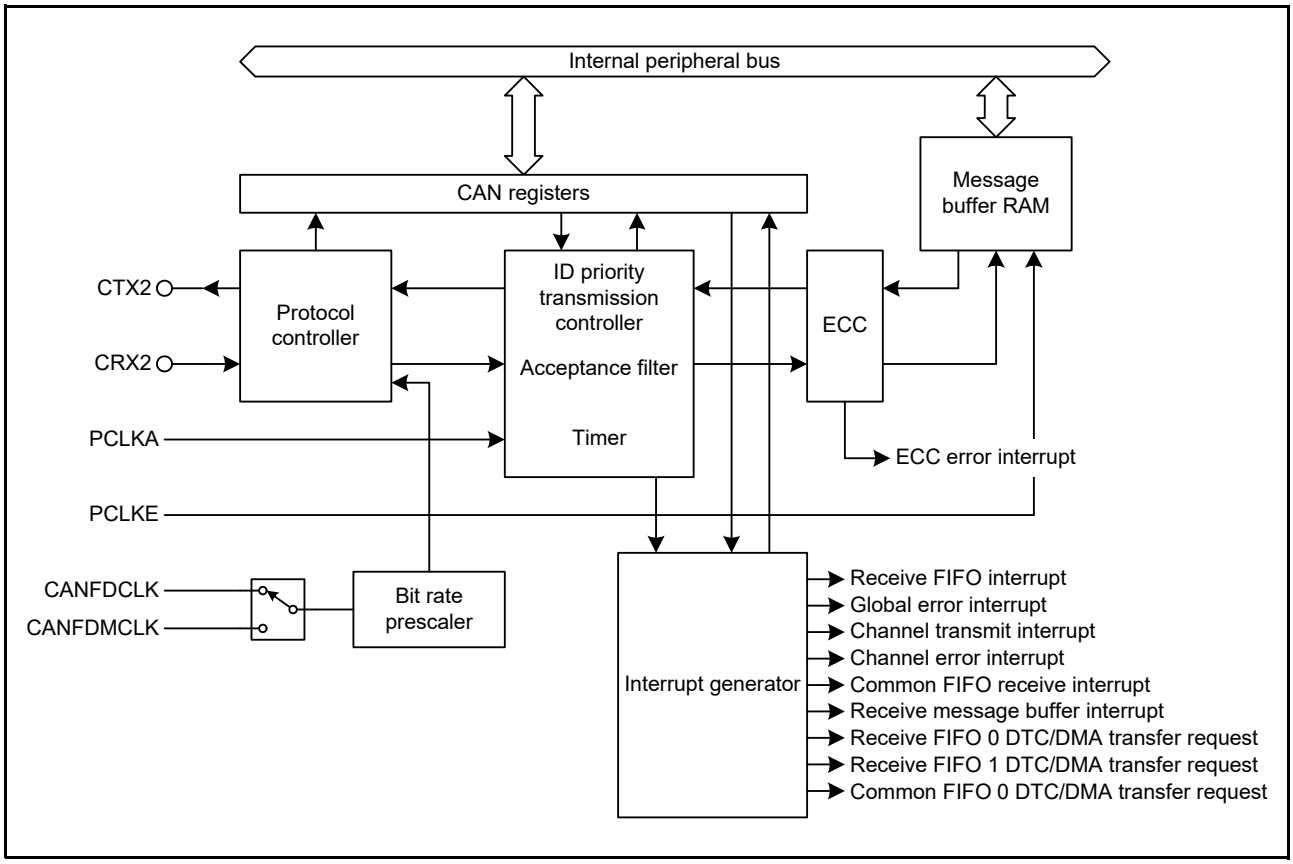


Figure 46.1 CANFD Module Block Diagram

- CRX2, CTX2
  I/O pins of the CANFD module
- Protocol controller
  Handles CAN FD protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.
- Message buffer RAM
  Used for message buffers or FIFOs for transmitting and receiving messages. Each message has an individual identifier, data length code, a data field, a message pointer for upper layer application, and a timestamp.
- Acceptance filter
  Performs filtering of received messages. The entries set in the acceptance filter list are used for the filtering process.
- Timers
  Two timers: one used for the receive timestamp function and the other for adjusting the message transmission interval from the transmit FIFO.

Table 46.2 lists the pin configuration of the CANFD module.

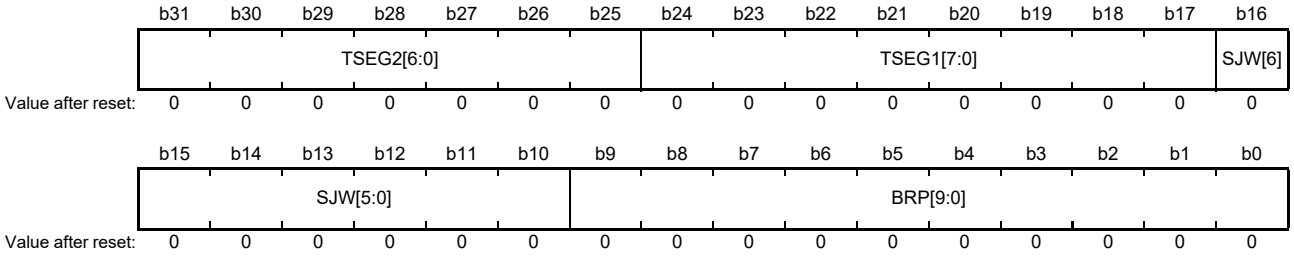
Table 46.2 CANFD Module Pin Configuration

Pin Name	I/O	Function
CRX2	Input	Receive data input
CTX2	Output	Transmit data output

46.2 Register Descriptions

46.2.1 Nominal Bit Rate Configuration Register (NBCR)

Address(es): CANFD2.NBCR 8747 0000h, CANFD2_NS.NBCR 0747 0000h



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	BRP[9:0]	Bit Rate Prescaler Setting	000h: No division 001h: Divided by 2 : : 3FEh: Divided by 1023 3FFh: Divided by 1024	R/W
b16 to b10	SJW[6:0]	Resynchronization Jump Width Control	00h: 1 Tq 01h: 2 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W
b24 to b17	TSEG1[7:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : FEh: 255 Tq FFh: 256 Tq	R/W
b31 to b25	TSEG2[6:0]	Time Segment 2 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W

This register is used to set the nominal bit rate during transmission and reception.
The value cannot be changed in CH_OPERATION or CH_SLEEP mode. Rewrite this register in CH_RESET or CH_HALT mode.
For details on the setting values, refer to section 46.4.1.2, Bit Timing.

BRP[9:0] Bits (Bit Rate Prescaler Setting)

These bits are used to define a period of 1 Tq (Time Quantum) that is the basis for CAN communication. Set the division ratio for the operating clock of the data link layer (DLL clock) selected by the GCFG.DLLCS bit. If the set value is n, the bit rate prescaler divides the DLL clock by n + 1.

SJW[6:0] Bits (Resynchronization Jump Width Control)

These bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 128 Tq can be set. Set a value less than or equal to that of the TSEG2[6:0] bits.

TSEG1[7:0] Bits (Time Segment 1 Control)

These bits are used to specify the total value (TSEG1) of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a Tq value. A value from 2 to 256 Tq can be set.

TSEG2[6:0] Bits (Time Segment 2 Control)

These bits are used to specify the value (TSEG2) of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 128 Tq can be set. Set a value less than that of the TSEG1[7:0] bits.

46.2.2 Channel Control Register (CHCR)

Address(es): CANFD2.CHCR 8747 0004h, CANFD2_NS.CHCR 0747 0004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ROME	BFT	—	—	—	CTMS[1:0]		CTME	EDM	BOM[1:0]		—	TDCVIE	SCOVIE	ECOVIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Channel Mode Control	b1 b0 0 0: Requests transition to CH_OPERATION mode 0 1: Requests transition to CH_RESET mode 1 0: Requests transition to CH_HALT mode 1 1: Keep current mode	R/W
b2	SLPRQ	CH_SLEEP Mode Request	0: Requests release from CH_SLEEP mode 1: Requests transition to CH_SLEEP mode	R/W
b3	RTBO	Forced Recovery from Bus-Off*1	0: Forced recovery from bus-off state is disabled. 1: Forced recovery from bus-off state is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BEIE	Bus Error Interrupt Enable*2	0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.	R/W
b9	EWIE	Error Warning Interrupt Enable*2	0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.	R/W
b10	EPIE	Error Passive Interrupt Enable*2	0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.	R/W
b11	BOEIE	Bus-Off Entry Interrupt Enable*2	0: Bus-off entry interrupt is disabled. 1: Bus-off entry interrupt is enabled.	R/W
b12	BORIE	Bus-Off Recovery Interrupt Enable*2	0: Bus-off recovery interrupt is disabled. 1: Bus-off recovery interrupt is enabled.	R/W
b13	OLIE	Overload Interrupt Enable*2	0: Overload interrupt is disabled. 1: Overload interrupt is enabled.	R/W
b14	BLIE	Bus Lock Interrupt Enable*2	0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.	R/W
b15	ALIE	Arbitration Lost Interrupt Enable*2	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b16	TAIE	Transmission Abort Interrupt Enable *2	0: Transmission abort interrupt is disabled. 1: Transmission abort interrupt is enabled.	R/W
b17	ECOVIE	Error Occurrence Counter Overflow Interrupt Enable*2	0: Error occurrence counter overflow interrupt is disabled. 1: Error occurrence counter overflow interrupt is enabled.	R/W
b18	SCOVIE	Success Occurrence Counter Overflow Interrupt Enable*2	0: Success occurrence counter overflow interrupt is disabled. 1: Success occurrence counter overflow interrupt is enabled.	R/W
b19	TDCVIE	Transceiver Delay Compensation Violation Interrupt Enable*2, *3	0: Transceiver delay compensation violation interrupt is disabled. 1: Transceiver delay compensation violation interrupt is enabled.	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22, b21	BOM[1:0]	Bus-Off Recovery Mode Select*2	b22 b21 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Automatically enters CH_HALT mode at bus-off entry 1 0: Automatically enters CH_HALT mode at bus-off end 1 1: Enters CH_HALT mode by software (during bus-off recovery)	R/W
b23	EDM	Error Display Mode Select*4	0: Only the first error detected is indicated. 1: All detected errors are indicated.	R/W
b24	CTME	Channel Test Mode Enable*5	0: Channel test mode is disabled. 1: Channel test mode is enabled.	R/W
b26, b25	CTMS[1:0]	Channel Test Mode Select*5	b26 b25 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loop-back mode) 1 1: Self-test mode 1 (internal loop-back mode)	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	BFT	Bit Flip Test*5	0: First bit of received data stream is not inverted. 1: First bit of received data stream is inverted.	R/W
b31	ROME	Restricted Operation Mode Enable *3, *5	0: Limited operation mode is disabled. 1: Limited operation mode is enabled.	R/W

Note 1. Set this bit in CH_OPERATION mode.
Note 2. Rewrite these bits in CH_RESET mode.
Note 3. Do not set this bit to 1 in Classic Only mode.
Note 4. Set this bit in CH_RESET or CH_HALT mode.
Note 5. Set these bits in CH_HALT mode.

This register controls the modes of the channel. It is also used to enable interrupt generation when an error is detected on the CAN bus and to set the test mode.
In CH_SLEEP mode, values other than the SLPRQ bit cannot be changed.

MDC[1:0] Bits (Channel Mode Control)

The MDC[1:0] bits are used to specify the mode of the CAN channel. For the CAN mode transition, refer to section 46.3.2, Channel Modes.
When the CANFD module is in GL_HALT mode, these bits can only be set to 10b (CH_HALT mode) or 01b (CH_RESET mode).
These bits automatically become 10b when the mode is changed to CH_HALT mode by the setting of the CHCR.BOM[1:0] bits.
When writing to these bits and transition to CH_HALT mode (at the entry to bus-off state when the CHCR.BOM[1:0] bits are 01b, and at the end of bus-off state when the CHCR.BOM[1:0] bits are 10b) occur at the same time, priority is given to writing from the CPU. The value of these bits are automatically updated only if the above event occurs when these bits are 00b (CH_OPERATION mode).

SLPRQ Bit (CH_SLEEP Mode Request)

This bit is used to control the transition to CH_SLEEP mode and the return from CH_SLEEP mode.
Setting this bit to 1 in CH_RESET mode requests the channel to transition to CH_SLEEP mode. Setting this bit to 0 in CH_SLEEP mode requests the channel to transition to CH_RESET mode. These bits cannot be changed in other modes.

RTBO Bit (Forced Recovery from Bus-Off)

This bit is used to forcibly recover from the bus-off state. Use this bit only when the CHCR.BOM[1:0] bits are 00b.
This bit is automatically set to 0 after being set to 1. The read value is always 0.
When this bit is set to 1 during bus-off state, the channel transitions from bus-off state to integrating state within 1 bit time. Also the CHSR.REC[7:0] and TEC[7:0] bits are set to 00h and the CHSR.BOST flag is set to 0. Other registers and

bits do not change. In this case, even if the bus-off recovery interrupt is enabled, the bus-off recovery interrupt does not generated.
If this bit is set to 1 except in the bus-off state, nothing occurs.

BEIE Bit (Bus Error Interrupt Enable)

When the CHESR.BEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EWIE Bit (Error Warning Interrupt Enable)

When the CHESR.EWDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EPIE Bit (Error Passive Interrupt Enable)

When the CHESR.EPDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the CHESR.BOEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the CHESR.BORDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

OLIE Bit (Overload Interrupt Enable)

When the CHESR.OLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BLIE Bit (Bus Lock Interrupt Enable)

When the CHESR.BLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

ALIE Bit (Arbitration Lost Interrupt Enable)

When the CHESR.ALDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TAIE Bit (Transmission Abort Interrupt Enable)

When the transmission from the transmit message buffer of the channel is successfully aborted while this bit is 1, a channel transmit interrupt request is generated.

ECOVIE Bit (Error Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.ECOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

SCOVIE Bit (Success Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.SCOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TDCVIE Bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the FDSTS.TDCV flag is set to 1 while this bit is 1, a channel error interrupt request is generated. Do not set this bit to 1 in Classic only mode.

BOM[1:0] Bits (Bus-Off Recovery Mode Select)

These bits control the recovery timing from the bus-off state.

EDM Bit (Error Display Mode Select)

This bit controls the indication mode of the error flags (b14 to b8) in the CHESR register.
When this bit is 0, only the flag corresponding to the first error detected becomes 1. If multiple errors are detected at the

same time, all corresponding flags become 1. The other flags do not become 1 until all the flags in b14 to b8 have been cleared.
When this bit is 1, the error flags are updated each time an error is detected.

CTME Bit (Channel Test Mode Enable)

This bit is used to enable the channel test mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

CTMS[1:0] Bits (Channel Test Mode Select)

These bits are used to select the test mode.
When the channel transitions to CH_RESET mode, these bits become 00b.

BFT Bit (Bit Flip Test)

This bit is used to check the CRC generator in the protocol controller.
Setting this bit to 1 inverts the first bit of the message data stream (ID bit) so that the result of the internally generated CRC does not match the received CRC value.
Note that as a result of bit inversion, a stuff error may be detected instead of a CRC error. Refer to the bit stuffing rule when using this function.
The CRC value generated internally can be checked in the following bits.

- The CHESR.CRC15[14:0] bits (for Classical CAN frames)
- The FDCRC.CRC21[20:0] bits (for CAN FD frames)

When using the BFT bit, it is necessary to have another CAN node transmit a reference message.

Note: Since the transmit and receive modes share the same CRC generator, there is no need to test for CRC errors in transmit mode separately.

Bit flip test mode is enabled when both the BFT and CTME bits are 1 and the CTMS[1:0] bits are 00b (basic test mode).
Using this function on the transmitter node causes a bit error or arbitration lost.
When the channel transitions to CH_RESET mode, this bit becomes 0.

ROME Bit (Restricted Operation Mode Enable)

When both this bit and the CTME bit are 1, the limited operation mode is enabled. Use this mode only in basic test mode (CTMS[1:0] bits = 00b). Also, do not set this bit to 1 in Classic only mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

46.2.3 Channel Status Register (CHSR)

Address(es): CANFD2.CHSR 8747 0008h, CANFD2_NS.CHSR 0747 0008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RESI	CRDY	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	CH_RESET Status Flag	0: Not in CH_RESET mode 1: In CH_RESET mode	R
b1	HLTST	CH_HALT Status Flag	0: Not in CH_HALT mode 1: In CH_HALT mode	R
b2	SLPST	CH_SLEEP Status Flag	0: Not in CH_SLEEP mode 1: In CH_SLEEP mode	R
b3	EPST	Error Passive Status Flag	0: Not in error passive state 1: In error passive state	R
b4	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b5	TRMST	Transmit Status Flag	0: Channel is not transmitting 1: Transmission in progress	R
b6	RECST	Receive Status Flag	0: Channel is not receiving 1: Reception in progress	R
b7	CRDY	Communication Ready Flag	0: Channel is not ready for communication 1: Channel is ready for communication	R
b8	RESI	Receive ESI Flag*1	0: No message with ESI flag set to 1 was received. 1: At least 1 message with ESI flag set to 1 was received.	R/(W) *2
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	REC[7:0]	Reception Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
b31 to b24	TEC[7:0]	Transmission Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the mode, error and transmission or reception status of the CAN channel together with its reception and transmission error count values.

RSTST Flag (CH_RESET Status Flag)

The RSTST flag indicates whether the CAN channel is in CH_RESET mode.
This flag is automatically set to 1 when the CAN channel enters CH_RESET mode, and is automatically set to 0 when the CAN channel exits CH_RESET mode. When the mode is changed from CH_RESET mode to CH_SLEEP mode, the RSTST flag remains 1.

HLTST Flag (CH_HALT Status Flag)

The HLTST flag indicates whether the CAN channel is in CH_HALT mode.
This flag is automatically set to 1 when the CAN channel enters CH_HALT mode, and is automatically set to 0 when the CAN channel exits CH_HALT mode.

SLPST Flag (CH_SLEEP Status Flag)

The SLPST flag indicates whether the CAN channel is in CH_SLEEP mode.
This flag is automatically set to 1 when the CAN channel enters CH_SLEEP mode, and is automatically set to 0 when the CAN channel exits CH_SLEEP mode.

EPST Flag (Error Passive Status Flag)

The EPST flag indicates whether the CAN channel has entered the error passive state.
This flag is automatically set to 1 when the value of the REC[7:0] or TEC[7:0] bits exceeds 127.
This flag is automatically set to 0 when the CAN channel exits the error passive state or enters CH_RESET mode.

BOST Flag (Bus-Off Status Flag)

The BOST flag indicates whether the CAN channel has entered the bus-off state.
This flag is automatically set to 1 when the value of the TEC[7:0] bits exceeds 255 and the CAN channel is in the bus-off state.
This flag is automatically set to 0 when the CAN channel exits bus-off state.

TRMST Flag (Transmit Status Flag)

The TRMST flag indicates whether the CAN channel is transmitting a message.
This flag is automatically set to 1 when the CAN channel is operating as a transmitter node or is in the bus-off state, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a receiver node.

RECST Flag (Receive Status Flag)

The RECST flag indicates whether the CAN channel is receiving a message.
This flag is automatically set to 1 when the CAN channel is operating as a receiver node, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a transmitter node.

CRDY Flag (Communication Ready Flag)

The CRDY flag indicates whether the CAN channel is ready for communication.
This flag is automatically set to 1 when the CAN channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting CH_RESET or CH_HALT mode.
This flag is automatically set to 0 when the CAN channel enters CH_RESET or CH_HALT mode.

Note: This flag is 1 in the bus-off state.

RESI Flag (Receive ESI Flag)

The RESI flag is set to 1 when the ESI bit is sampled recessively for a received message without any error. When in loop-back or mirror mode, the self-transmitted messages are considered reception messages.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is cleared by writing 0 to it. This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only this flag to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

REC[7:0] Bits (Reception Error Count)

The REC[7:0] bits indicate the value of the reception error counter.
The value in bus-off state is indeterminate.
These bits are automatically set to 00h when the CANFD module enters GL_RESET or the CAN channel enters CH_RESET mode.

TEC[7:0] Bits (Transmission Error Count)

The TEC[7:0] bits indicate the value of the transmission error counter.
These bits are automatically set to 00h when CANFD module enters GL_RESET or CAN channel enters CH_RESET mode.

46.2.4 Channel Error Status Register (CHESR)

Address(es): CANFD2.CHESR 8747 000Ch, CANFD2_NS.CHESR 0747 000Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CRC15[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADEDF	B0EDF	B1EDF	CEDF	AEDF	FEDF	SEDF	ALDF	BLDF	OLDF	BORDF	BOEDF	EPDF	EWDF	BEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEDF	Bus Error Detect Flag*1	0: Bus error is not detected 1: Bus error is detected	R/(W) *2
b1	EWDF	Error Warning Detect Flag*1	0: Error warning is not detected 1: Error warning is detected	R/(W) *2
b2	EPDF	Error Passive Detect Flag*1	0: Error passive is not detected 1: Error passive is detected	R/(W) *2
b3	BOEDF	Bus-Off Entry Detect Flag*1	0: Bus-off entry is not detected 1: Bus-off entry is detected	R/(W) *2
b4	BORDF	Bus-Off Recovery Detect Flag*1	0: Bus-off recovery is not detected 1: Bus-off recovery is detected	R/(W) *2
b5	OLDF	Overload Detect Flag*1	0: Overload is not detected 1: Overload is detected	R/(W) *2
b6	BLDF	Bus Lock Detect Flag*1	0: Bus lock is not detected 1: Bus lock is detected	R/(W) *2
b7	ALDF	Arbitration Lost Detect Flag*1	0: Arbitration lost is not detected 1: Arbitration lost is detected	R/(W) *2
b8	SEDF	Stuff Error Detect Flag*1	0: Stuff error is not detected 1: Stuff error is detected	R/(W) *2
b9	FEDF	Form Error Detect Flag*1	0: Form error is not detected 1: Form error is detected	R/(W) *2
b10	AEDF	Acknowledge Error Detect Flag*1	0: Acknowledge error is not detected 1: Acknowledge error is detected	R/(W) *2
b11	CEDF	CRC Error Detect Flag*1	0: CRC error is not detected 1: CRC error is detected	R/(W) *2
b12	B1EDF	Bit 1 Error Detect Flag*1	0: Bit 1 error is not detected 1: Bit 1 error is detected	R/(W) *2
b13	B0EDF	Bit 0 Error Detect Flag*1	0: Bit 0 error is not detected 1: Bit 0 error is detected	R/(W) *2
b14	ADEDF	ACK Delimiter Error Detect Flag*1	0: Acknowledge delimiter error is not detected 1: Acknowledge delimiter error is detected	R/(W) *2
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30-b16	CRC15[14:0]	CRC_15 Test	These bits show the CRC_15 value calculated for the CAN2.0 CAN frame.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of various error conditions detectable regardless of the interrupt enable/disable setting of the Channel Control Register (CHCR). It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for the conditions under which each error occurs.

Only a single bit can be cleared at a time. Do not use the bit clear instruction to clear the flag. Set only the flag to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

BEDF Flag (Bus Error Detect Flag)

The BEDF flag indicates a detection of an error state, flagged by b14 to b8 in this register.
This flag is automatically set to 1 when a bus error is detected, and is automatically set to 0 when the CAN channel enters CH_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

EWDF Flag (Error Warning Detect Flag)

The EWDF flag indicates whether an error warning condition has been detected for the CAN channel.
This flag is automatically set to 1 when the value of either the CHSR.TEC[7:0] or REC[7:0] bits exceeds 95.
The setting of this flag only occurs when the value of the TEC[7:0] or REC[7:0] bits initially exceeds 95. Therefore, if the TEC[7:0] or REC[7:0] bits remains > 95 and the EWDF flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 96 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 95 to a value > 95.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel enters CH_RESET mode.

EPDF Flag (Error Passive Detect Flag)

The EPDF flag indicates a detection of a CAN channel error passive state.
This flag is automatically set to 1 when the CAN error state becomes error passive state.
The setting of this flag only occurs when the value of either the CHSR.TEC[7:0] or REC[7:0] bits initially exceeds 127. Therefore, if the value of either the TEC[7:0] or REC[7:0] bits remains > 127 and the flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 128 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 127 to a value > 127.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel enters CH_RESET mode.

BOEDF Flag (Bus-Off Entry Detect Flag)

The BOEDF flag indicates a detection of a CAN channel bus-off entry state.
This flag is automatically set to 1 when the CAN error state enters the bus-off state.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.

BORDF Flag (Bus-Off Recovery Detect Flag)

The BORDF flag indicates a detection of a CAN channel bus-off recovery state.
This flag is automatically set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CHCR.BOM[1:0] bits are 00b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 10b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 11b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs.

The flag is not set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CH_RESET mode is requested
- When setting the CHCR.RTBO bit to 1 (the CAN channel returns to error active)
- When the CHCR.BOM[1:0] bits are 01b

- When the CHCR.BOM[1:0] bits are 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set to 1.

OLDF Flag (Overload Detect Flag)

The OLDF flag indicates a detection of a CAN channel overload state.
This flag is automatically set to 1 when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.

BLDF Flag (Bus Lock Detect Flag)

The BLDF flag indicates a detection of a CAN channel bus lock condition.
This flag is automatically set to 1 when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in CH_OPERATION mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.

ALDF Flag (Arbitration Lost Detect Flag)

The ALDF flag indicates a detection of a CAN channel bus arbitration lost condition.
The flag is automatically set to 1 when an arbitration lost condition is detected on the CAN bus while the CAN channel is in CH_OPERATION mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.

SEDF Flag (Stuff Error Detect Flag)

The SEDF flag indicates a detection of a CAN stuff error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a stuff error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

FEDF Flag (Form Error Detect Flag)

The FEDF flag indicates a detection of a CAN form error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

AEDF Flag (Acknowledge Error Detect Flag)

The AEDF flag indicates a detection of a CAN acknowledge error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.

This flag is automatically set to 1 when an acknowledge error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CEDF Flag (CRC Error Detect Flag)

The CEDF flag indicates a detection of a CAN CRC error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a CRC error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B1EDF Flag (Bit 1 Error Detect Flag)

The B1EDF flag indicates a detection of a recessive bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B0EDF Flag (Bit 0 Error Detect Flag)

The B0EDF flag indicates a detection of a dominant bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

ADEDF Flag (ACK Delimiter Error Detect Flag)

The ADEDF flag indicates a detection of an acknowledge delimiter bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected during the acknowledge delimiter state of frame transmission. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CRC15[14:0] Bits (CRC_15 Test)

The CRC15[14:0] bits indicate the calculated CRC_15 value when the CHCR.CTME bit is 1 (channel test mode enabled). If the CHCR.CTME bit is 0, then these bits are always read as 0000h.
The CRC_15 value which is read from these bits show the CAN2.0 CRC value calculated by the CAN channel logic.
The value of the CRC15[14:0] bits is updated in the first bit of the CRC field of the Classical CAN frame.
These bits are automatically set to 0000h when the CAN channel enters CH_RESET mode.

46.2.5 Data Bit Rate Configuration Register (DBCR)

Address(es): CANFD2.DBCR 8747 0100h, CANFD2_NS.DBCR 0747 0100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SJW[3:0]				—	—	—	—	TSEG2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSEG1[4:0]					BRP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BRP[7:0]	Bit Rate Prescaler Setting	00h: No division 01h: Divided by 2 : : FEh: Divided by 255 FFh: Divided by 256	R/W
b12 to b8	TSEG1[4:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : 1Eh: 31 Tq 1Fh: 32 Tq	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	TSEG2[3:0]	Time Segment 2 Control	0h: Setting prohibited 1h: 2 Tq : : Eh: 15 Tq Fh: 16 Tq	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	SJW[3:0]	Resynchronization Jump Width Control	0h: 1 Tq 1h: 2 Tq : : Fh: 16 Tq	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register configures the transmission/reception data bit rate parameters of the channel.
In Classic only mode, there is no need to configure this register.

BRP[7:0] Bits (Bit Rate Prescaler Setting)

The BRP[7:0] bits define the number of the DLL clock contained in 1 Tq (Time Quantum).

TSEG1[4:0] Bits (Time Segment 1 Control)

The TSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 Tq can be set.
The TSEG1[4:0] bits are also used to set the propagation time segment.
Do not write any other value to these bits. Refer to section 46.4.1.2, Bit Timing for more details.

TSEG2[3:0] Bits (Time Segment 2 Control)

The TSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A

value from 2 to 16 Tq can be set.
Do not write any other value to these bits.

SJW[3:0] Bits (Resynchronization Jump Width Control)

The SJW[3:0] bits set the resynchronization jump width. A value from 1 to 16 Tq can be set.

46.2.6 CAN FD Configuration Register (FDCFG)

Address(es): CANFD2.FDCFG 8747 0104h, CANFD2_NS.FDCFG 0747 0104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TESI	TDCE	SSPC	—	—	—	—	—	ECC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ECC[2:0]	Error Occurrence Counter Configuration*2	b2 b0 0 0 0: All CAN transmitter or receiver frames 0 0 1: All CAN transmitter frames 0 1 0: All CAN receiver frames 0 1 1: Setting prohibited 1 0 0: Only transmitter or receiver CAN FD data-phase 1 0 1: Only transmitter CAN FD data-phase 1 1 0: Only receiver CAN FD data-phase 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SSPC	Secondary Sample Point Configuration*2	0: Measured + offset 1: Offset-only	R/W
b9	TDCE	Transceiver Delay Compensation Enable*2	0: Transceiver delay compensation is disabled 1: Transceiver delay compensation is enabled	R/W
b10	TESI	Transmit ESI Configuration*2	0: The ESI flag in the transmission frame reflects the error status of the node itself. 1: The ESI flag in the transmission frame reflects the ESI bit in the message buffer if the node is not error passive, and the error status of the node itself If the node is error passive.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TDCO[7:0]	Transceiver Delay Compensation Offset*2	Sets the offset value for the transceiver delay compensation	R/W
b27 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	FDOE	FD Only Mode Enable*3	0: FD only mode is disabled 1: FD only mode is enabled	R/W
b29	REFE	Receive Edge Filter Enable*3	0: Reception edge filter is disabled 1: Reception edge filter is enabled	R/W
b30	CLOE	Classic Only Mode Enable*3, *4	0: Classic only mode is disabled 1: Classic only mode is enabled	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The value after reset is 0 for products that support the CAN FD protocol, and 1 for products that support only CAN 2.0 protocol.
Note 2. Set these bits in CH_RESET or CH_HALT mode.
Note 3. Set these bits in CH_RESET mode.
Note 4. This bit can only be written for products that support the CAN FD protocol. For products that support only CAN 2.0 protocol, this bit is reserved and fixed to 1.

This register configures which communication direction (transmitter/receiver) errors are counted.

ECC[2:0] Bits (Error Occurrence Counter Configuration)

The ECC[2:0] bits select which type of CAN frame the protocol errors should be counted for.

SSPC Bit (Secondary Sample Point Configuration)

The SSPC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN channel. If the bit is 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.
Do not set this bit to 1 in Classic only mode.

TDCE Bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN channel.
Do not set this bit to 1 in Classic only mode.

TESI Bit (Transmit ESI Configuration)

The TESI bit selects whether to reflect the error status of the node itself or the value of the ESI bit in the message buffer (CFB0.HF2.ESI bit or TMBn.HF2.ESI bit) in the ESI flag of the transmit message.
Do not set this bit to 1 in Classic only mode.

TDCO[7:0] Bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the offset of the secondary sample point. How this value is used, depends on the SSPC setting. If the SSPC bit is 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in the TDCO[7:0] bits, rounded down to the nearest integer number of Tq. Otherwise, the result is equal to the value in the TDCO[7:0] bits. Refer to section 46.4.1.5, Transceiver Delay Compensation for details.
The actual offset value is interpreted as TDCO[7:0] + 1. For example, if 4 is set in TDCO[7:0], the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.
Do not set to these bits in Classic only mode.

FDOE Bit (FD Only Mode Enable)

The FDOE bit enables the transmission and reception of CAN FD frames only. If enabled, communication in Classical CAN frame format is disabled. The value of the FDF bit in the message buffer (CFB0.HF2.FDF bit or TMBn.HF2.FDF bit) is arbitrary because transmission of Classical CAN frames is not possible.
If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and responds with error frames. If a Classical CAN frame is configured for transmission, the FDF bit is transmitted recessive, therefore a CAN FD frame is transmitted. If the data length code (DLC) is configured to be 9 bytes or more, the remaining data bytes are padded with CCh.
Do not set the FDOE and CLOE bits to 1 simultaneously.

REFE Bit (Receive Edge Filter Enable)

The REFE bit enables the reception edge filter during the integrating state. When this bit is 1, two or more consecutive dominant Tq are required to detect an edge for hard synchronization.
Do not set this bit to 1 in Classic only mode.

CLOE Bit (Classic Only Mode Enable)

The CLOE bit enables the Classic only mode. If this bit is set to 1, the protocol controller can only transmit Classical CAN frames and responds to CAN FD frames with a form error or CRC error.
Do not set the CLOE and FDOE bits to 1 simultaneously.

Table 46.3 Operation Mode Configuration

CLOE bit	FDOE bit	Operation mode
0	0	CAN FD mode
0	1	FD only mode
1	0	Classic only mode
1	1	Setting prohibited

46.2.7 CAN FD Control Register (FDCTR)

Address(es): CANFD2.FDCTR 8747 0108h, CANFD2_NS.FDCTR 0747 0108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCCL	ECCL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECCL	Error Occurrence Counter Clear	When 1 is written to this bit, the error occurrence counter is cleared. This bit is read as 0.	R/W
b1	SCCL	Success Occurrence Counter Clear	When 1 is written to this bit, the success occurrence counter is cleared. This bit is read as 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the error and success occurrence counters.

ECCL Bit (Error Occurrence Counter Clear)

The ECCL bit is used to clear the error occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel enters CH_RESET mode.

SCCL Bit (Success Occurrence Counter Clear)

The SCCL bit is used to clear the success occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel enters CH_RESET mode.

46.2.8 CAN FD Status Register (FDSTS)

Address(es): CANFD2.FDSTS 8747 010Ch, CANFD2_NS.FDSTS 0747 010Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SC[7:0]								EC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDCV	—	—	—	—	—	SCOV	ECOV	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TDCR[7:0]	Transceiver Delay Compensation Result	Indicates the transceiver delay compensation result when the transceiver delay has been measured	R
b8	ECOV	Error Occurrence Counter Overflow Flag*1	0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/(W) *2
b9	SCOV	Success Occurrence Counter Overflow Flag*1	0: Success occurrence counter has not overflowed 1: Success occurrence counter has overflowed	R/(W) *2
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TDCV	Transceiver Delay Compensation Violation Flag*1	0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/(W) *2
b23 to b16	EC[7:0]	Error Occurrence Counter	These bits show the error occurrence counter value.	R
b31 to b24	SC[7:0]	Success Occurrence Counter	These bits show the success occurrence counter value.	R

Note 1. Set this flag to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] Bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.
The measured delay is represented by the number of the DLL clock cycles. The result depends on the configuration of the FDCFG.SSPC bit and the offset value in the FDCFG.TDCO[7:0] bits. Refer to section 46.4.1.5, Transceiver Delay Compensation for details.
The TDCR[7:0] bits are updated at the falling edge between the FDF bit and res bit when the FDCFG.SSPC bit is 0 and the FDCFG.TDCE bit is 1 (transceiver delay compensation is enabled).
These bits are automatically set to 0 when the CAN channel enters CH_RESET mode.

ECOV Flag (Error Occurrence Counter Overflow Flag)

The ECOV flag indicates whether the CAN channel error occurrence counter has overflowed.
This flag is set to 1 if a CAN bus error specified in the FDCFG.ECC[2:0] bits is detected when the EC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

SCOV Flag (Success Occurrence Counter Overflow Flag)

The SCOV flag indicates whether the CAN channel success occurrence counter has overflowed.

This flag is set to 1 if a successful message reception or successful message transmission occurs when the SC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

TDCV Flag (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.
The transceiver delay has some variations depending on the physical parameters such as temperature. The TDCR[7:0] bits are updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCV flag captures this violation.
This flag is set to 1 when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times – 2 DLL clocks) and the internal bit overruns.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel enters CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

EC[7:0] Bits (Error Occurrence Counter)

The EC[7:0] bits are used together with the SC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
This higher error rate can be detected depending on the configuration of the FDCFG.ECC[2:0] bits.
The EC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.ECCL bit.
These bits are updated when an error occurs, according to the configuration of the FDCFG.ECC[2:0] bits. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel enters CH_RESET mode.

SC[7:0] Bits (Success Occurrence Counter)

The SC[7:0] bits are used together with the EC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
The SC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.SCCL bit.
These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel enters CH_RESET mode.

Note: In loop-back mode, the counter is incremented twice.

46.2.9 CAN FD CRC Register (FDCRC)

Address(es): CANFD2.FDCRC 8747 0110h, CANFD2_NS.FDCRC 0747 0110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SBC[3:0]				—	—	—	CRC21[20:16]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CRC21[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b20 to b0	CRC21[20:0]	CRC_21 Test	These bits show the CRC_17 value or CRC_21 value calculated for the CAN FD frame.	R
b23 to b21	—	Reserved	These bits are read as 0.	R
b27 to b24	SBC[3:0]	Stuff Bit Counter	These bits shows the stuff bit count (Mod 8) for the CAN FD frame.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

This register holds the CRC value calculated for the CAN FD frame.

CRC21[20:0] Bits (CRC_21 Test)

The calculated CRC_17 value or CRC_21 value can be read from this bits when the CHCR.CTME bit is 1 (channel test mode).
When the CHCR.CTME bit is 0, the CRC21[20:0] bits are read as 000000h.
The CRC21[20:0] bits is updated in the first bit of the CRC field of the CAN FD frame.
When the CRC_17 field is used, the CRC21[20:17] bits are read as 0.
These bits are automatically set to 000000h when the CAN channel enters CH_RESET mode.

SBC[3:0] Bits (Stuff Bit Counter)

The SBC[3:0] bits contain the stuff count value of the CAN FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN FD frame when the CHCR.CTME bit is enabled in SBC[3:1]. SBC[0] is the parity bit.
When the CHCR.CTME bit is 0, the SBC[3:0] bits are always read as 0000b.
The value of the SBC[3:0] bits is updated in the first bit of CRC field of the CAN FD frame. These bits are automatically set to 0000b when the CAN channel enters CH_RESET mode.

46.2.10 Global Configuration Register (GCFG)

Address(es): CANFD.GCFG 8747 0014h, CANFD_NS.GCFG 0747 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ITP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSCS	TSP[3:0]				—	—	OMRC	DLLCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TPRI	Transmission Priority Setting	0: ID priority 1: Message buffer number priority	R/W
b1	DCE	DLC Check Enable	0: DLC check is disabled 1: DLC check is enabled	R/W
b2	DRE	DLC Replacement Enable	0: DLC replacement is disabled 1: DLC replacement is enabled	R/W
b3	MME	Mirror Mode Enable	0: Mirror mode is disabled 1: Mirror mode is enabled	R/W
b4	DLLCS	DLL Clock Select	0: CANFDCLK 1: CANFDMCLK	R/W
b5	OMRC	Payload-Overflowed Message Reception Configuration	0: The message is discarded. 1: The message payload is cut to fit the specified message size	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TSP[3:0]	Timestamp Counter Prescaler Setting	b11 b8 0 0 0 0: No division 0 0 0 1: Divide-by-2 (= 2^1) 0 0 1 0: Divide-by-4 (= 2^2) 0 0 1 1: Divide-by-8 (= 2^3) : : 1 1 0 1: Divide-by-8192 (= 2^{13}) 1 1 1 0: Divide-by-16384 (= 2^{14}) 1 1 1 1: Divide-by-32768 (= 2^{15})	R/W
b12	TSCS	Timestamp Counter Source Select	0: Clock source for timestamp counter is peripheral module clock (PCLKA) 1: Clock source for timestamp counter is bit time clock	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	ITP[15:0]	Interval Timer Prescaler Setting	FIFO interval timer prescaler value. Set the divided value for the peripheral module clock (PCLKA)	R/W

This register is used to select the transmission priority to be used for all the transmit message buffers and the clock source for the CAN protocol engine. The GCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI Bit (Transmission Priority Setting)

The TPRI bit selects the transmission priority for the CAN channel.

This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode. Do not select message buffer number priority when using the transmit queue.

DCE Bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

DRE Bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE bit is 1, the CANFD module stores the configured value (AFLn.PTR0.DLC[3:0]) of the DLC in the destination receive message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination receive message buffer or FIFO buffer is unchanged.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

MME Bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

DLLCS Bit (DLL Clock Select)

The DLLCS bit selects the clock source for CAN communication.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only in GL_RESET mode.

OMRC Bit (Payload-Overflowed Message Reception Configuration)

The OMRC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size (RMCR.PLS[2:0], RFCRn.PLS[2:0], and CFCR0.PLS[2:0]). The received message payload is always compared with the available message payload size in the message buffer.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only in GL_RESET mode.
When this bit is set to 1 and payload overflow occurs, the DLC value is stored in the receive message buffer or FIFO buffer unchanged.

TSP[3:0] Bits (Timestamp Counter Prescaler Setting)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.
These bits cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode.

TSCS Bit (Timestamp Counter Source Select)

The TSCS bit allows the selection of the clock source for the timestamp counter.
This bit cannot be written in GL_SLEEP mode. Write to this bit only in GL_RESET mode. Additionally, do not set this bit to 1 when CAN FD communication is used.

Note: The bit time clock varies depending on the nominal bit rate and data bit rate configuration.

ITP[15:0] Bits (Interval Timer Prescaler Setting)

The ITP[15:0] bits allow the definition of a reference clock for the FIFO interval timer clock source. When these bits are 0000h, the timer is disabled.
These bits cannot be written in GL_SLEEP mode. Write to these bits only in GL_RESET mode.

46.2.11 Global Control Register (GCR)

Address(es): CANFD.GCR 8747 0018h, CANFD_NS.GCR 0747 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	POIE	THLIE	MLIE	DEIE	—	—	—	—	—	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Global Mode Control	b1 b0 0 0: Request transition to GL_OPERATION mode 0 1: Request transition to GL_RESET mode 1 0: Request transition to GL_HALT mode 1 1: Keep current value	R/W
b2	SLPRQ	GL_SLEEP Mode Request	0: Request to release GL_SLEEP 1: Request transition to GL_SLEEP	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DEIE	DLC Error Interrupt Enable	0: DLC error interrupt is disabled 1: DLC error interrupt is enabled	R/W
b9	MLIE	Message Lost Interrupt Enable	0: Message lost interrupt is disabled 1: Message lost interrupt is enabled	R/W
b10	THLIE	Transmission History Entry Lost Interrupt Enable	0: Transmission history entry lost interrupt is disabled 1: Transmission history entry lost interrupt is enabled	R/W
b11	POIE	Payload Overflow Interrupt Enable	0: Payload overflow interrupt is disabled 1: Payload overflow interrupt is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TSCR	Timestamp Counter Reset	When 1 is written to this bit, the timestamp counter is reset. This bit is read as 0.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the Global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

MDC[1:0] Bits (Global Mode Control)

The MDC[1:0] bits are used to specify the modes of the CANFD module. For the mode transition of the CANFD module, refer to section 46.3.1, Global Modes.

These bits cannot be written in GL_SLEEP mode.

To request the CANFD module to transition to GL_SLEEP mode, first set these bits to 01b to enter GL_RESET mode, then set the GCR.SLPRQ bit to 1.

SLPRQ Bit (GL_SLEEP Mode Request)

The SLPRQ bit is used to control the transition to GL_SLEEP mode and the return from GL_SLEEP mode.

When this bit is set to 1, the transition to CH_SLEEP mode is also requested for CAN channel.

This bit can only be written in GL_RESET or GL_SLEEP mode.

DEIE Bit (DLC Error Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

This bit cannot be written in GL_SLEEP mode.

MLIE Bit (Message Lost Interrupt Enable)

When the MLIE bit is 1, an interrupt is generated if a message lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

THLIE Bit (Transmission History Entry Lost Interrupt Enable)

When the THLIE bit is 1, an interrupt is generated if a transmission history entry lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

POIE Bit (Payload Overflow Interrupt Enable)

When the POIE bit is 1, an interrupt is generated when a message payload overflow condition occurs.
This bit cannot be written in GL_SLEEP mode.

TSCR Bit (Timestamp Counter Reset)

When the TSCR bit is 1, the Timestamp Counter Register (TSCR) is reset to 00000000h.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit in GL_RESET mode.
This bit is automatically set to 0.

46.2.12 Global Status Register (GSR)

Address(es): CANFD.GSR 8747 001Ch, CANFD_NS.GSR 0747 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RAMST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	GL_RESET Status Flag	0: Not in GL_RESET mode 1: In GL_RESET mode	R
b1	HLTST	GL_HALT Status Flag	0: Not in GL_HALT mode 1: In GL_HALT mode	R
b2	SLPST	GL_SLEEP Status Flag	0: Not in GL_SLEEP mode 1: In GL_SLEEP mode	R
b3	RAMST	RAM Initialization Status Flag	0: RAM initialization is completed 1: RAM initialization is in progress	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register indicates the global status of the CANFD module.

RSTST Flag (GL_RESET Status Flag)

The RSTST flag indicates the status of the CANFD module in GL_RESET mode.
This flag is automatically set to 1 when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains 1. This flag is automatically set to 0 when the CANFD module enters GL_HALT or GL_OPERATION mode.

HLTST Flag (GL_HALT Status Flag)

The HLTST flag indicates the status of the CANFD module in GL_HALT mode.
This flag is automatically set to 1 when the CANFD module enters GL_HALT mode. This flag is automatically set to 0 when the CANFD module exits the GL_HALT mode.

SLPST Flag (GL_SLEEP Status Flag)

The SLPST flag indicates the status of the CANFD module in GL_SLEEP mode.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode. This flag is automatically set to 0 when the CANFD module exits the GL_SLEEP mode.

RAMST Flag (RAM Initialization Status Flag)

The RAMST flag indicates the status of the CANFD module's RAM initialization.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode by resetting the MCU. This flag is automatically set to 0 when the CANFD module completed RAM initialization.

46.2.13 Global Error Status Register (GESR)

Address(es): CANFD.GESR 8747 0020h, CANFD_NS.GESR 0747 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PODF	THLDF	MLDF	DEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DEDF	DLC Error Detect Flag	0: DLC error is not detected 1: DLC error is detected	R/(W) *1
b1	MLDF	Message Lost Detect Flag	0: Message lost error is not detected 1: Message lost error is detected	R
b2	THLDF	Transmission History Entry Lost Detect Flag	0: Transmission history entry lost is not detected 1: Transmission history entry lost is detected	R
b3	PODF	Payload Overflow Detect Flag	0: Payload overflow is not detected 1: Payload overflow is detected	R/(W) *1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EEDF0	ECC Error Detect Flag	0: ECC error is not detected during transmission scan 1: ECC error is detected during transmission scan	R/(W) *1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the detection of global errors.

DEDF Flag (DLC Error Detect Flag)

The DEDF flag indicates the error status of the DLC.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 1 when a DLC error is detected in a received frame.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The flag is cleared by writing 0 to it. This flag is automatically set to 0 when the CANFD module enters GL_RESET mode.

MLDF Flag (Message Lost Detect Flag)

The MLDF flag indicates the status of the message lost error.
This flag is automatically set to 1 when a FIFO message lost error is detected.
This flag is automatically set to 0 when:

- All FIFO message lost flags (RFSRn.LOST, CFSR0.LOST) are cleared
- The CANFD module enters GL_RESET mode.

THLDF Flag (Transmission History Entry Lost Detect Flag)

The THLDF flag indicates the status of the transmission history entry lost error.
This flag is automatically set to 1 when a transmission history entry lost error is detected.
This flag is automatically set to 0 when:

- The transmission history lost flag (THSR.LOST) is cleared
- The CANFD module enters GL_RESET mode.

PODF Flag (Payload Overflow Detect Flag)

The PODF flag is automatically set to 1 when a message payload overflow is detected on at least one channel.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 0 when the CANFD module enters GL_RESET mode.

EEDF0 Flag (ECC Error Detect Flag)

The EEDF0 flag indicates whether an ECC error has occurred.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This bit is automatically set to 0 when the CANFD module enters GL_RESET mode.

46.2.14 Transmit Interrupt Status Register (TISR)

Address(es): CANFD.TISR 8747 00A4h, CANFD_NS.TISR 0747 00A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSIF0	Transmission Successful Interrupt Flag	0: Transmission successful interrupt is not generated. 1: Transmission successful interrupt is generated.	R
b1	TAIF0	Transmission Abort Interrupt Flag	0: Transmission abort interrupt is not generated. 1: Transmission abort interrupt is generated.	R
b2	TQIF0	Transmit Queue Interrupt Flag	0: Transmit queue interrupt is not generated. 1: Transmit queue interrupt is generated.	R
b3	CFTIF0	Common FIFO Transmission Interrupt Flag	0: Common FIFO transmission interrupt is not generated. 1: Common FIFO transmission interrupt is generated.	R
b4	THIF0	Transmission History Interrupt Flag	0: Transmission history interrupt is not generated. 1: Transmission history interrupt is generated.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

This register indicates the detection of transmit specific interrupts.

TSIF0 Flag (Transmission Successful Interrupt Flag)

The TSIF0 flag becomes 1 if transmission from the transmit message buffer n is successful when the transmit message buffer n interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (TMIER0.TMIEn bits = 0)
- When the transmission result flag (TMSRn.TXRF[1:0]) is cleared
- When in GL_RESET or CH_RESET mode.

TAIF0 Flag (Transmission Abort Interrupt Flag)

The TAIF0 flag becomes 1 if transmission from the transmit message buffer n is aborted when the transmission abort interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (CHCR.TAIE bit = 0)
- When the transmission result flag (TMSRn.TXRF[1:0]) is cleared
- When in GL_RESET or CH_RESET mode.

TQIF0 Flag (Transmit Queue Interrupt Flag)

The TQIF0 flag becomes 1 if the Transmit Queue Interrupt flag is set to 1 when the transmit queue interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQIE bit = 0)
- When the transmission queue interrupt flag (TQSR0.TQIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFTIF0 Flag (Common FIFO Transmission Interrupt Flag)

The CFTIF0 flag becomes 1 if the Common FIFO Transmit Interrupt flag (CFSR0.CFTIF) is set to 1 when the common FIFO transmit interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CFCR0.CFTIE bit = 0)
- When the common transmit FIFO interrupt flag (CFSR0.CFTIF) is cleared
- When in GL_RESET or CH_RESET mode.

THIF0 Flag (Transmission History Interrupt Flag)

The THIF0 flag becomes 1 if the Transmission History Interrupt flag (THSR.THIF) is set to 1 when the transmission history interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (THCR.THIE bit = 0)
- When the transmission history interrupt flag (THSR.THIF) is cleared
- When in GL_RESET or CH_RESET mode.

46.2.15 Timestamp Counter Register (TSCR)

Address(es): CANFD.TSCR 8747 0024h, CANFD_NS.TSCR 0747 0024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is a read-only register that stores a timestamp based on the selected configuration.
The timestamp value is stored in the TSCR register based on the configuration of the GCFG.TSCS bit and GCFG.TSP[3:0] bits. The accuracy of the timestamp counter cannot be guaranteed when transitioning to GL_HALT state.
This register is automatically set to 00000000h in GL_RESET mode.

46.2.16 Acceptance Filter List Control Register (AFCR)

Address(es): CANFD.AFCR 8747 0028h, CANFD_NS.AFCR 0747 0028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	AFLWE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	AFLWE	AFL Write Enable	0: Acceptance Filter List data access is disabled 1: Acceptance Filter List data access is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to select the Acceptance Filter List page for reading or writing entries into the Acceptance Filter List.

AFLWE Bit (AFL Write Enable)

The AFLWE bit allows writing to the Acceptance Filter List.

Setting this bit to 1 enables writing to the Acceptance Filter List. Setting this bit to 0 after configuring the Acceptance Filter List prevents writing to the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

This bit cannot be written in GL_SLEEP mode.

46.2.17 Acceptance Filter List Configuration Register (AFCFG)

Address(es): CANFD.AFCFG 8747 002Ch, CANFD_NS.AFCFG 0747 002Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	RN0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	RN0[4:0]	Number of Rules Setting	Number of rules in the Acceptance Filter List	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

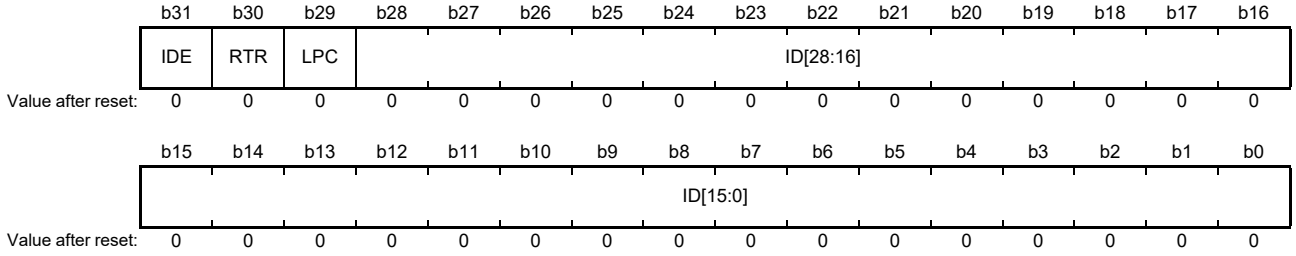
This register is used to define the number of rules for entry in the Acceptance Filter List.
The maximum number of available entries in the Acceptance Filter List is 16.

RN0[4:0] Bits (Number of Rules Setting)

The RN0[4:0] bits define the number of rules in the Acceptance Filter List. Set a value of 16 or less.
These bits can only be written in GL_RESET mode.

46.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15)

Address(es): CANFD.AFL0.IDR 8747 0120h, CANFD.AFL1.IDR 8747 0130h, CANFD.AFL2.IDR 8747 0140h, CANFD.AFL3.IDR 8747 0150h, CANFD.AFL4.IDR 8747 0160h, CANFD.AFL5.IDR 8747 0170h, CANFD.AFL6.IDR 8747 0180h, CANFD.AFL7.IDR 8747 0190h, CANFD.AFL8.IDR 8747 01A0h, CANFD.AFL9.IDR 8747 01B0h, CANFD.AFL10.IDR 8747 01C0h, CANFD.AFL11.IDR 8747 01D0h, CANFD.AFL12.IDR 8747 01E0h, CANFD.AFL13.IDR 8747 01F0h, CANFD.AFL14.IDR 8747 0200h, CANFD.AFL15.IDR 8747 0210h, CANFD_NS.AFL0.IDR 0747 0120h, CANFD_NS.AFL1.IDR 0747 0130h, CANFD_NS.AFL2.IDR 0747 0140h, CANFD_NS.AFL3.IDR 0747 0150h, CANFD_NS.AFL4.IDR 0747 0160h, CANFD_NS.AFL5.IDR 0747 0170h, CANFD_NS.AFL6.IDR 0747 0180h, CANFD_NS.AFL7.IDR 0747 0190h, CANFD_NS.AFL8.IDR 0747 01A0h, CANFD_NS.AFL9.IDR 0747 01B0h, CANFD_NS.AFL10.IDR 0747 01C0h, CANFD_NS.AFL11.IDR 0747 01D0h, CANFD_NS.AFL12.IDR 0747 01E0h, CANFD_NS.AFL13.IDR 0747 01F0h, CANFD_NS.AFL14.IDR 0747 0200h, CANFD_NS.AFL15.IDR 0747 0210h



Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	ID Field	ID part of the Acceptance Filter List entry	R/W
b29	LPC	Loop-Back Configuration	0: Perform normal processing of received messages 1: Only accept its own transmitted messages	R/W
b30	RTR	RTR	0: Data frame 1: Remote frame	R/W
b31	IDE	IDE	0: Standard ID 1: Extended ID	R/W

Note: Set these registers in CH_RESET or CH_HALT mode. However, using the AFIGSR and AFIGER registers allows these registers to be rewritten even in CH_OPERATION mode. For details, refer to section 46.5.10, Updating AFL Entry during Communication.

These registers are used to configure the ID field in the rule entry of the Acceptance Filter List.
When rewriting these registers, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

ID[28:0] Bits (ID Field)

The ID[28:0] bits indicate the CAN identifier (ID) field of each entry in the Acceptance Filter List.
The acceptance filter process compares this field against the ID of a received message. For alignment of these bits in base and extended formats, refer to section 46.2.60, Identifier Bits Alignment.

LPC Bit (Loop-Back Configuration)

The LPC bit specifies the processing of received messages in loop-back mode and mirror mode for each entry in the Acceptance Filter List. For normal received messages, set this bit to 0.
Entries with this bit set to 1 only match its own transmitted messages in loop-back mode or mirror mode. Entries with this bit set to 0 match all messages transmitted by other nodes, and its own transmitted messages in loop-back mode.
For details, refer to section 46.5.8, Loop-Back Configuration.

RTR Bit (RTR)

The RTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received message.

IDE Bit (IDE)

The IDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received message.

46.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15)

Address(es): CANFD.AFL0.MASK 8747 0124h, CANFD.AFL1.MASK 8747 0134h, CANFD.AFL2.MASK 8747 0144h, CANFD.AFL3.MASK 8747 0154h, CANFD.AFL4.MASK 8747 0164h, CANFD.AFL5.MASK 8747 0174h, CANFD.AFL6.MASK 8747 0184h, CANFD.AFL7.MASK 8747 0194h, CANFD.AFL8.MASK 8747 01A4h, CANFD.AFL9.MASK 8747 01B4h, CANFD.AFL10.MASK 8747 01C4h, CANFD.AFL11.MASK 8747 01D4h, CANFD.AFL12.MASK 8747 01E4h, CANFD.AFL13.MASK 8747 01F4h, CANFD.AFL14.MASK 8747 0204h, CANFD.AFL15.MASK 8747 0214h,
CANFD_NS.AFL0.MASK 0747 0124h, CANFD_NS.AFL1.MASK 0747 0134h, CANFD_NS.AFL2.MASK 0747 0144h, CANFD_NS.AFL3.MASK 0747 0154h, CANFD_NS.AFL4.MASK 0747 0164h, CANFD_NS.AFL5.MASK 0747 0174h, CANFD_NS.AFL6.MASK 0747 0184h, CANFD_NS.AFL7.MASK 0747 0194h, CANFD_NS.AFL8.MASK 0747 01A4h, CANFD_NS.AFL9.MASK 0747 01B4h, CANFD_NS.AFL10.MASK 0747 01C4h, CANFD_NS.AFL11.MASK 0747 01D4h, CANFD_NS.AFL12.MASK 0747 01E4h, CANFD_NS.AFL13.MASK 0747 01F4h, CANFD_NS.AFL14.MASK 0747 0204h, CANFD_NS.AFL15.MASK 0747 0214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDEM	RTRM	IFL1	IDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	IDM[28:0]	ID Mask Field	0: Corresponding ID bit is not used for ID matching 1: Corresponding ID bit is used for ID matching	R/W
b29	IFL1	Information Label 1	Configure an information label 1 to be attached to a received message	R/W
b30	RTRM	RTR Mask	0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
b31	IDEM	IDE Mask	0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

Note: Set these registers in CH_RESET or CH_HALT mode. However, using the AFIGSR and AFIGER registers allows these registers to be rewritten even in CH_OPERATION mode. For details, refer to section 46.5.10, Updating AFL Entry during Communication.

These registers are used to configure the mask field of each rule for entries in the Acceptance Filter List.
When rewriting these registers, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

IDM[28:0] Bits (ID Mask Field)

The IDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Acceptance Filter List entry.

IFL1 Bit (Information Label 1)

The IFL1 bit is the upper bit of a 2-bit information label to be attached to a received message accepted by the associated entry in the Acceptance Filter List.
This bit is stored in the upper bit of the Information Label Field (RMBn.HF2.IFL[1], RFBn.HF2.IFL[1], CFB0.HF2.IFL[1]) of the storage location of a received message.

RTRM Bit (RTR Mask)

The RTRM bit is the RTR mask bit for each entry in the Acceptance Filter List.

IDEM Bit (IDE Mask)

The IDEM bit is the IDE mask bit for each entry in the Acceptance Filter List.
When the IDE mask bit is 0, the ID comparison depends on the IDE bit of the received message.

- If the IDE bit of the received message is 0, the standard ID comparison takes place.
- If the IDE bit of the received message is 1, the extended ID comparison takes place.

46.2.20 Acceptance Filter List n Pointer 0 Register (AFLn.PTR0) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR0 8747 0128h, CANFD.AFL1.PTR0 8747 0138h, CANFD.AFL2.PTR0 8747 0148h, CANFD.AFL3.PTR0 8747 0158h, CANFD.AFL4.PTR0 8747 0168h, CANFD.AFL5.PTR0 8747 0178h, CANFD.AFL6.PTR0 8747 0188h, CANFD.AFL7.PTR0 8747 0198h, CANFD.AFL8.PTR0 8747 01A8h, CANFD.AFL9.PTR0 8747 01B8h, CANFD.AFL10.PTR0 8747 01C8h, CANFD.AFL11.PTR0 8747 01D8h, CANFD.AFL12.PTR0 8747 01E8h, CANFD.AFL13.PTR0 8747 01F8h, CANFD.AFL14.PTR0 8747 0208h, CANFD.AFL15.PTR0 8747 0218h, CANFD_NS.AFL0.PTR0 0747 0128h, CANFD_NS.AFL1.PTR0 0747 0138h, CANFD_NS.AFL2.PTR0 0747 0148h, CANFD_NS.AFL3.PTR0 0747 0158h, CANFD_NS.AFL4.PTR0 0747 0168h, CANFD_NS.AFL5.PTR0 0747 0178h, CANFD_NS.AFL6.PTR0 0747 0188h, CANFD_NS.AFL7.PTR0 0747 0198h, CANFD_NS.AFL8.PTR0 0747 01A8h, CANFD_NS.AFL9.PTR0 0747 01B8h, CANFD_NS.AFL10.PTR0 0747 01C8h, CANFD_NS.AFL11.PTR0 0747 01D8h, CANFD_NS.AFL12.PTR0 0747 01E8h, CANFD_NS.AFL13.PTR0 0747 01F8h, CANFD_NS.AFL14.PTR0 0747 0208h, CANFD_NS.AFL15.PTR0 0747 0218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMBE	—	—	DMB[4:0]					IFL0	—	—	—	DLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	DLC Field	Configure a minimum DLC value of received message	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IFL0	Information Label 0	Configure an information label 0 to be attached to a received message	R/W
b12 to b8	DMB[4:0]	Destination Message Buffer Setting	Configure the receive message buffer number for storage of received messages	R/W
b14 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DMBE	Destination Message Buffer Setting Enable	0: The Destination Message Buffer Setting bit is disabled 1: The Destination Message Buffer Setting bit is enabled	R/W
b31 to b16	PTR[15:0]	Pointer	Configure a 16-bit pointer to be attached to a received message	R/W

Note: Set these registers in CH_RESET or CH_HALT mode. However, using the AFIGSR and AFIGER registers allows these registers to be rewritten even in CH_OPERATION mode. For details, refer to section 46.5.10, Updating AFL Entry during Communication.

These registers are used to configure the data length code (DLC), software pointer, and destination message buffer for each rule entry in the Acceptance Filter List.
When rewriting these registers, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

DLC[3:0] Bits (DLC Field)

The DLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Acceptance Filter List (automatic DLC filter function).
DLC filter process is only passed if the DLC value of the message accepted by an entry in the Acceptance Filter List is equal to or higher than the DLC value configured for this associated Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0000b.

IFL0 Bit (Information Label 0)

The IFL0 bit is the lower bit of a 2-bit information label that can be attached to a received message accepted by the Acceptance Filter List entry.
This bit is stored in the lower bit of the Information Label Field (RMBn.HF2.IFL[0], RFBn.HF2.IFL[0], CFB0.HF2.IFL[0]) of the storage location of a received message.

DMB[4:0] Bits (Destination Message Buffer Setting)

The DMB[4:0] bits allow the configuration of a message buffer as the destination for a received message that passes the acceptance check of the Acceptance Filter List entry. Configure the destination message buffer number.
Write the RMCR.NMB[4:0] bits to configure the number of receive message buffers. The value to be entered in the DMB[4:0] bits should only be between 00000b and 'NMB[4:0] – 1'.
If the RMCR.NMB[4:0] bits are 00000b, set the DMBE bit to 0.

DMBE Bit (Destination Message Buffer Setting Enable)

The DMBE bit allows the enabling or disabling of the receive message buffer as the destination message buffer for a received message that passes the acceptance check of the Acceptance Filter List entry.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the Acceptance Filter List entry. The pointer is added during message storage in the message buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

46.2.21 Acceptance Filter List n Pointer 1 Register (AFLn.PTR1) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR1 8747 012Ch, CANFD.AFL1.PTR1 8747 013Ch, CANFD.AFL2.PTR1 8747 014Ch, CANFD.AFL3.PTR1 8747 015Ch, CANFD.AFL4.PTR1 8747 016Ch, CANFD.AFL5.PTR1 8747 017Ch, CANFD.AFL6.PTR1 8747 018Ch, CANFD.AFL7.PTR1 8747 019Ch, CANFD.AFL8.PTR1 8747 01ACh, CANFD.AFL9.PTR1 8747 01BCh, CANFD.AFL10.PTR1 8747 01CCh, CANFD.AFL11.PTR1 8747 01DCh, CANFD.AFL12.PTR1 8747 01ECh, CANFD.AFL13.PTR1 8747 01FCh, CANFD.AFL14.PTR1 8747 020Ch, CANFD.AFL15.PTR1 8747 021Ch, CANFD_NS.AFL0.PTR1 0747 012Ch, CANFD_NS.AFL1.PTR1 0747 013Ch, CANFD_NS.AFL2.PTR1 0747 014Ch, CANFD_NS.AFL3.PTR1 0747 015Ch, CANFD_NS.AFL4.PTR1 0747 016Ch, CANFD_NS.AFL5.PTR1 0747 017Ch, CANFD_NS.AFL6.PTR1 0747 018Ch, CANFD_NS.AFL7.PTR1 0747 019Ch, CANFD_NS.AFL8.PTR1 0747 01ACh, CANFD_NS.AFL9.PTR1 0747 01BCh, CANFD_NS.AFL10.PTR1 0747 01CCh, CANFD_NS.AFL11.PTR1 0747 01DCh, CANFD_NS.AFL12.PTR1 0747 01ECh, CANFD_NS.AFL13.PTR1 0747 01FCh, CANFD_NS.AFL14.PTR1 0747 020Ch, CANFD_NS.AFL15.PTR1 0747 021Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CF0E	—	—	—	—	—	—	RF1E	RF0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RF0E	Receive FIFO 0 Destination Enable	0: Do not select receive FIFO 0 as the message storage destination 1: Selects receive FIFO 0 as the message storage destination	R/W
b1	RF1E	Receive FIFO 1 Destination Enable	0: Do not select receive FIFO 1 as the message storage destination 1: Selects receive FIFO 1 as the message storage destination	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CF0E	Common FIFO 0 Destination Enable	0: Do not select common FIFO 0 as the message storage destination 1: Selects common FIFO 0 as the message storage destination	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set these registers in CH_RESET or CH_HALT mode. However, using the AFIGSR and AFIGER registers allows these registers to be rewritten even in CH_OPERATION mode. For details, refer to section 46.5.10, Updating AFL Entry during Communication.

These registers are used to configure the destination FIFO buffer for each rule entry in the Acceptance Filter List.
Up to two storage destinations can be specified for received messages. Two FIFO buffers, or one FIFO buffer and one receive message buffer are valid.
When rewriting these registers, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

RF0E Bit (Receive FIFO 0 Destination Enable)

The RF0E bit allows the configuration of the receive FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

RF1E Bit (Receive FIFO 1 Destination Enable)

The RF1E bit allows the configuration of the receive FIFO 1 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

CF0E Bit (Common FIFO 0 Destination Enable)

The CF0E bit allows the configuration of the common FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

The common FIFO 0 must be configured as the receive FIFO.

46.2.22 Receive Message Buffer Configuration Register (RMCR)

Address(es): CANFD.RMCR 8747 0030h, CANFD_NS.RMCR 0747 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PLS[2:0]			—	—	—	NMB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	NMB[4:0]	Number of Message Buffer Setting	Configure the number of receive message buffers	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	PLS[2:0]	Payload Size Setting	b10 b8 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the total number of receive message buffers allocated to a channel.

NMB[4:0] Bits (Number of Message Buffer Setting)

The NMB[4:0] bits are used to configure the number of receive message buffers.
These bits can only be written in GL_RESET mode.
Set a value between 0 and 16. A value of 0 indicates that no receive message buffer is allocated.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits are used to configure the message buffer payload size.
These bits can only be written in GL_RESET mode.

46.2.23 Receive Message Buffer New Data Register (RMNDR)

Address(es): CANFD.RMNDR 8747 0034h, CANFD_NS.RMNDR 0747 0034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NDF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	NDF[15:0]	New Data Flag	0: New data is not stored in corresponding receive message buffer 1: New data is stored in corresponding receive message buffer	R/(W) *1

Note 1. Writing 1 has no effect. These flags are cleared by writing 0 to them.

This register specifies the new data storage status of the receive message buffers.
The bit number of this register corresponds to the buffer number of the receive message buffer.

NDF[15:0] Flags (New Data Flag)

The NDF[15:0] flags indicate that the new data is stored in the corresponding receive message buffer. The NDF[0] flag corresponds to receive message buffer 0.
These flags are automatically set to 1 when storage of new messages are in the corresponding receive message buffer. When RMCR.PLS[2:0] = 000b (payload size is 8 bytes), the time required to store a message is 6 PCLKA cycles. When RMCR.PLS[2:0] > 000b, the time increases by 1 PCLKA cycle for every 4 bytes of payload size (20 PCLKA cycles for 64 bytes).
Do not write to these flags in GL_RESET or GL_SLEEP mode.
These flags are automatically set to 0 when the CANFD module enters GL_RESET mode.
These flags cannot be cleared when message storage in the corresponding receive message buffer is in progress.
Do not use the bit clear instruction to clear the flags. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 16-bit units using MOV instruction.

46.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0, 1)

Address(es): CANFD.RFCR0 8747 003Ch, CANFD.RFCR1 8747 0040h, CANFD_NS.RFCR0 0747 003Ch, CANFD_NS.RFCR1 0747 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFITH[2:0]			RFIM	—	FDS[2:0]			—	PLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	RFIE	Receive FIFO Interrupt Enable	0: FIFO interrupt generation is disabled 1: FIFO interrupt generation is enabled	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	FDS[2:0]	FIFO Depth Setting	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	RFIM	Receive FIFO Interrupt Mode Setting	0: Interrupt is generated when the number of messages stored in the receive FIFO reaches the value of the RFITH[2:0] bits from a value smaller than the RFITH[2:0] bits 1: Interrupt is generated at the end of every received message storage	R/W
b15 to b13	RFITH[2:0]	Receive FIFO Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt generated when FIFO is 1/8 full 0 0 1: Interrupt generated when FIFO is 1/4 full 0 1 0: Interrupt generated when FIFO is 3/8 full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8 full 1 0 1: Interrupt generated when FIFO is 3/4 full 1 1 0: Interrupt generated when FIFO is 7/8 full 1 1 1: Interrupt generated when FIFO is full	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

These registers are used to configure and control the two receive FIFOs.

RFE Bit (Receive FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the receive FIFO is cleared to empty.
This bit can only be written in GL_HALT or GL_OPERATION mode.
This bit can be set to 1 only when the FIFO depth is 4 to 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the RFE bit to 1 with a separate write access to the RFCRn register, after all the other flags in the RFCRn register are set.
This bit is automatically set to 0 when the CANFD module enters GL_RESET mode.

RFIE Bit (Receive FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.
This bit cannot be written in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM.
This is the maximum number of bytes which can be received by this FIFO.
These bits can only be written in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits select the depth of the FIFO in units of the number of messages. If the FIFO depth is set to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

RFIM Bit (Receive FIFO Interrupt Mode Setting)

The RFIM bit selects the interrupt generation condition for the FIFO.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

RFITH[2:0] Bits (Receive FIFO Interrupt Threshold Setting)

The RFITH[2:0] bits select the counter value of the FIFO for generation of receive FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the RFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 46.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

46.2.25 Receive FIFO n Status Register (RFSRn) (n = 0, 1)

Address(es): CANFD.RFSR0 8747 0044h, CANFD.RFSR1 8747 0048h, CANFD_NS.RFSR0 0747 0044h, CANFD_NS.RFSR1 0747 0048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	—	RFIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Receive FIFO Empty Flag	0: Message in receive FIFO 1: No message in receive FIFO (empty)	R
b1	FULL	Receive FIFO Full Flag	0: Receive FIFO not full 1: Receive FIFO full	R
b2	LOST	Message Lost Flag	0: Receive FIFO message lost has not occurred 1: Receive FIFO message lost has occurred	R/(W) *1
b3	RFIF	Receive FIFO Interrupt Flag	0: Receive FIFO interrupt condition is not satisfied 1: Receive FIFO interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Receive FIFO Fill Level	Indicate the number of messages stored in receive FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

These registers show the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Receive FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The FLVL[5:0] flags are 00000b
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module enters GL_RESET mode.

The EMPTY flag is automatically set to 0 when the first message is stored in the receive FIFO.

FULL Flag (Receive FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO buffer matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO buffer is less than the configured FIFO depth
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module enters GL_RESET mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage when the FIFO buffer is already full. This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

The flag is set to 0:

- By writing 0 to it

- When the CANFD module enters GL_RESET mode.

This flag can only be written in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

RFIF Flag (Receive FIFO Interrupt Flag)

The RFIF flag is automatically set to 1 when the configured interrupt condition is satisfied. This flag is not automatically cleared when the receive FIFO is disabled.
The flag is set to 0:

- By writing 0 to it
- When the CANFD module enters GL_RESET mode.

This flag can only be written in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

FLVL[5:0] Bits (Receive FIFO Fill Level)

The FLVL[5:0] bits indicate the number of messages stored in the receive FIFO that can be read by the CPU. These bits are automatically set to 000000b when the FIFO is disabled or when the CANFD module enters GL_RESET mode.

46.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0, 1)

Address(es): CANFD.RFPCR0 8747 004Ch, CANFD.RFPCR1 8747 0050h, CANFD_NS.RFPCR0 0747 004Ch, CANFD_NS.RFPCR1 0747 0050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read pointer of the corresponding receive FIFOs.
When the value 000000FFh is written to this register, the pointer of the corresponding receive FIFO is moved to the next FIFO entry. Write to this register only when the corresponding receive FIFO is enabled and not empty.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Do not write to this register when DTC/DMA transfer is enabled (DTCR.RFDTEn bit = 1).

46.2.27 Common FIFO 0 Configuration Register (CFCR0)

Address(es): CANFD.CFCR0 8747 0054h, CANFD_NS.CFCR0 0747 0054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TINT[7:0]								FDS[2:0]			—	—	—	LTM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFITH[2:0]			CFIM	ITR	ITCS	—	MODE	—	PLS[2:0]			—	CFTIE	CFRIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFE	Common FIFO Enable	0: Common FIFO is disabled 1: Common FIFO is enabled	R/W
b1	CFRIE	Common FIFO Receive Interrupt Enable	0: FIFO receive interrupt generation is disabled 1: FIFO receive interrupt generation is enabled	R/W
b2	CFTIE	Common FIFO Transmit Interrupt Enable	0: FIFO transmit interrupt generation is disabled 1: FIFO transmit interrupt generation is enabled	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	MODE	Operation Mode Setting	0: Receive FIFO mode 1: Transmit FIFO mode	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	ITCS	Interval Timer Clock Source Select	0: Reference clock (× 1 / × 10 period) 1: Bit time clock	R/W
b11	ITR	Interval Timer Resolution Select	0: Reference clock period × 1 1: Reference clock period × 10	R/W
b12	CFIM	Common FIFO Interrupt Mode Setting	Receive FIFO mode: 0: Reception interrupt generated when the number of messages stored in the common FIFO reaches CFITH[2:0] value from a lower value 1: Reception interrupt generated at the end of every received message storage Transmit FIFO mode: 0: Transmission interrupt generated when common FIFO transmits the last message successfully 1: Transmission interrupt generated for every successfully transmitted message	R/W
b15 to b13	CFITH[2:0]	Common FIFO Receive Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt is generated when FIFO is 1/8 full 0 0 1: Interrupt is generated when FIFO is 1/4 full 0 1 0: Interrupt is generated when FIFO is 3/8 full 0 1 1: Interrupt is generated when FIFO is 1/2 full 1 0 0: Interrupt is generated when FIFO is 5/8 full 1 0 1: Interrupt is generated when FIFO is 3/4 full 1 1 0: Interrupt is generated when FIFO is 7/8 full 1 1 1: Interrupt is generated when FIFO is full	R/W

Bit	Symbol	Bit Name	Description	R/W
b17 to b16	LTM[1:0]	Linked Transmit Message Buffer Select	Transmission scan link position of the corresponding channel	R/W
b20 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b21	FDS[2:0]	FIFO Depth Setting	b23 b21 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than the above are prohibited.	R/W
b31 to b24	TINT[7:0]	Transmission Interval Setting	Delay the start of transmission from the FIFO if configured in transmit mode, delay is a multiple of basic interval timer clock source unit	R/W

CFE Bit (Common FIFO Enable)

Setting the CFE bit to 1 enables the FIFO. The FIFO is disabled when this bit is set to 0.
This bit can also be used, by clearing it, to abort transmission from common FIFO when configured in transmit FIFO mode, or to stop reception into the common FIFO in receive FIFO mode.
This bit can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
This bit can only be set to 1 when the FIFO depth is between 4 and 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the CFE bit to 1 with a separate write access to the CFCR0 register, after setting all the other bits in this register.
This bit is automatically set to 0 when the CANFD module enters GL_RESET mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

CFRIE Bit (Common FIFO Receive Interrupt Enable)

The CFRIE bit enables or disables generation of common FIFO receive interrupt. If this flag is 1, common FIFO receive interrupt is generated when the common FIFO receive interrupt flag is set to 1 after reception of a frame in the corresponding FIFO buffer.
Do not write to this bit in GL_SLEEP mode.

CFTIE Bit (Common FIFO Transmit Interrupt Enable)

The CFTIE bit enables or disables generation of common FIFO transmit interrupt. If this flag is 1, common FIFO transmit interrupt is generated when the common FIFO transmit interrupt flag is set to 1 after transmission of a frame from the corresponding FIFO buffer.
Do not write to this bit in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.
For details, refer to section 46.6, FIFO Buffers and Message Buffer Configuration.
These bits can only be written in GL_RESET mode.

MODE Bit (Operation Mode Setting)

The MODE bit is used to select the operation mode of the common FIFO. When the MCU reset is applied, all the common FIFO are configured in receive FIFO mode.
This bit cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

ITCS Bit (Interval Timer Clock Source Select)

The ITCS bit is used to select the clock source for the transmission interval timer.
Do not write to this bit in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.
Do not write 1 to this bit when CAN FD communication is used.

Note: The bit time clock can vary depending on the nominal bit rate and data bit rate configuration.

ITR Bit (Interval Timer Resolution Select)

The ITR bit is used to select the resolution of the reference clock for the transmission interval timer.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit when the CFE bit is set to 1.

CFIM Bit (Common FIFO Interrupt Mode Setting)

The CFIM bit is used to select the condition for the generation of common FIFO interrupts.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

CFITH[2:0] Bits (Common FIFO Receive Interrupt Threshold Setting)

The CFITH[2:0] bits are used to select the message counter value for the generation of common FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the CFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 46.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

LTM[1:0] Bits (Linked Transmit Message Buffer Select)

The LTM[1:0] bits are used to select the number of the transmit message buffer that links common FIFO configured in transmit FIFO mode, for transmission scanning.
These bits cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to these bits only in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits are used to select the FIFO depth in units of the number of messages. If the FIFO depth is configured to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

TINT[7:0] Bits (Transmission Interval Setting)

The TINT[7:0] bits are used to select the delay in the start of transmission for all messages transmitted from the common FIFO configured in transmit FIFO mode. The delay is an integral multiple of the clock source cycle of the interval timer (reference clock period × 1, reference clock period × 10, or bit-time clock period).
These bits cannot be written in GL_SLEEP mode.
Do not write to these bits when the CFE bit is set to 1.
When the GCFG.ITP[15:0] bits are set to 0000h, set the TINT[7:0] bits to 00h.

46.2.28 Common FIFO 0 Status Register (CFSR0)

Address(es): CANFD.CFSR0 8747 0058h, CANFD_NS.CFSR0 0747 0058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	CFTIF	CFRIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Common FIFO Empty Flag	0: Message in common FIFO 1: No message in common FIFO (empty)	R
b1	FULL	Common FIFO Full Flag	0: Common FIFO is not full 1: Common FIFO is full	R
b2	LOST	Message Lost Flag	0: No message lost in common FIFO 1: Common FIFO message lost	R/(W) *1
b3	CFRIF	Common FIFO Receive Interrupt Flag	0: Common FIFO receive interrupt condition is not satisfied 1: Common FIFO receive interrupt condition is satisfied	R/(W) *1
b4	CFTIF	Common FIFO Transmit Interrupt Flag	0: Common FIFO transmit interrupt condition is not satisfied 1: Common FIFO transmit interrupt condition is satisfied	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Common FIFO Fill Level	Indicate the number of messages stored in common FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Common FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The CPU has read all messages from the FIFO buffer in receive FIFO mode
- All messages have been transmitted from the FIFO buffer in transmit FIFO mode
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode.

The EMPTY flag is automatically set to 0 when:

- The first reception message is stored in the FIFO buffer in receive FIFO mode
- The first message to be transmitted is stored in the FIFO buffer in transmit FIFO mode.

FULL Flag (Common FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO is less than the configured FIFO depth
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage of a new message when FIFO buffer is already full in receive FIFO mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
The LOST flag is set to 0:

- By writing 0 to it
- When the CANFD module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode.

CFRIF Flag (Common FIFO Receive Interrupt Flag)

The CFRIF flag is set to 1 when the configured interrupt condition is satisfied in receive FIFO mode.
The CFRIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFRIF flag is set to 0:

- By writing 0 to it
- When the CANFD module enters GL_RESET mode.

CFTIF Flag (Common FIFO Transmit Interrupt Flag)

The CFTIF flag is set to 1 when the configured interrupt condition is satisfied in transmit FIFO mode.
The CFTIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFTIF flag is set to 0:

- By writing 0 to it
- When the CANFD module enters GL_RESET mode
- When the CAN channel enters CH_RESET mode in transmit FIFO mode.

FLVL[5:0] Bits (Common FIFO Fill Level)

The FLVL[5:0] bits indicate the following:

- The number of messages stored by the CPU and waiting to be transmitted in the transmit FIFO mode

- The number of CPU-readable messages stored by CANFD in receive FIFO mode.

The FLVL[5:0] bits are automatically set to 000000b when:

- The CFCR0.CFE bit is set to 0 (common FIFO is disabled)
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode in transmit FIFO mode.

46.2.29 Common FIFO 0 Pointer Control Register (CFPCR0)

Address(es): CANFD.CFPCR0 8747 005Ch, CANFD_NS.CFPCR0 0747 005Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read or write pointer of the corresponding common FIFO.
When the value 000000FFh is written into this register, the read pointer of the corresponding common FIFO in receive FIFO mode, or the write pointer of the corresponding common FIFO in transmit FIFO mode moves to the next FIFO entry.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Write to this register only when:

- The common FIFO is enabled and is not empty in receive FIFO mode
- The common FIFO is enabled and is not full in transmit FIFO mode.

Do not write to the CFPCR0 register when DTC/DMA transfer is enabled (DTCR.CFDTE0 bit = 1).

46.2.30 FIFO Empty Status Register (FESR)

Address(es): CANFD.FESR 8747 0060h, CANFD_NS.FESR 0747 0060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFEMP0	—	—	—	—	—	—	RFEMP1	RFEMP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RFEMP0	Receive FIFO 0 Empty Flag	0: Message in corresponding receive FIFO 0 1: No message in corresponding receive FIFO 0 (empty)	R
b1	RFEMP1	Receive FIFO 1 Empty Flag	0: Message in corresponding receive FIFO 1 1: No message in corresponding receive FIFO 1 (empty)	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFEMP0	Common FIFO 0 Empty Flag	0: Message in corresponding common FIFO 0 1: No message in corresponding common FIFO 0 (empty)	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the empty flags of the FIFO buffers.

RFEMP0 Flag (Receive FIFO 0 Empty Flag)

The RFEMP0 flag is set to 1 when the RFSR0.EMPTY flag is set to 1, and is set to 0 when the RFSR0.EMPTY flag is set to 0.

The RFEMP0 flag is set to 1 when the CANFD module enters GL_RESET mode.

RFEMP1 Flag (Receive FIFO 1 Empty Flag)

The RFEMP1 flag is set to 1 when the RFSR1.EMPTY flag is set to 1, and is set to 0 when the RFSR1.EMPTY flag is set to 0.

The RFEMP1 flag is set to 1 when the CANFD module enters GL_RESET mode.

CFEMP0 Flag (Common FIFO 0 Empty Flag)

The CFEMP0 flag is set to 1 when the CFSR0.EMPTY flag is set to 1, and is set to 0 when the CFSR0.EMPTY flag is set to 0.

The CFEMP0 flag is set to 1 when the CANFD module enters GL_RESET mode.

46.2.31 FIFO Full Status Register (FFSR)

Address(es): CANFD.FFSR 8747 0064h, CANFD_NS.FFSR 0747 0064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFFUL0	—	—	—	—	—	—	RFFUL1	RFFUL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFFUL0	Receive FIFO 0 Full Flag	0: Corresponding receive FIFO 0 is not full 1: Corresponding receive FIFO 0 is full	R
b1	RFFUL1	Receive FIFO 1 Full Flag	0: Corresponding receive FIFO 1 is not full 1: Corresponding receive FIFO 1 is full	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFFUL0	Common FIFO 0 Full Flag	0: Common FIFO 0 is not full 1: Common FIFO 0 is full	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the full flags of the FIFO buffers.

RFFUL0 Flag (Receive FIFO 0 Full Flag)

The RFFUL0 flag is set to 1 when the RFSR0.FULL flag is set to 1, and is set to 0 when the RFSR0.FULL flag is set to 0.

The RFFUL0 flag is set to 0 when the CANFD module enters GL_RESET mode.

RFFUL1 Flag (Receive FIFO 1 Full Flag)

The RFFUL1 flag is set to 1 when the RFSR1.FULL flag is set to 1, and is set to 0 when the RFSR1.FULL flag is set to 0.

The RFFUL1 flag is set to 0 when the CANFD module enters GL_RESET mode.

CFFUL0 Flag (Common FIFO 0 Full Flag)

The CFFUL0 flag is set to 1 when the CFSR0.FULL flag is set to 1 is set to 0 when the CFSR0.FULL flag is set to 0.

The CFFUL0 flag is set to 0 when the CANFD module enters GL_RESET mode.

46.2.32 FIFO Message Lost Status Register (FMLSR)

Address(es): CANFD.FMLSR 8747 0068h, CANFD_NS.FMLSR 0747 0068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFML0	—	—	—	—	—	—	RFML1	RFML0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFML0	Receive FIFO 0 Message Lost Flag	0: No message lost has occurred in receive FIFO 0 1: Message lost has occurred in receive FIFO 0	R
b1	RFML1	Receive FIFO 1 Message Lost Flag	0: No message lost has occurred in receive FIFO 1 1: Message lost has occurred in receive FIFO 1	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFML0	Common FIFO 0 Message Lost Flag	0: No message lost has occurred in common FIFO 0 1: Message lost has occurred in common FIFO 0	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the message lost flags of the FIFO buffers.

RFML0 Flag (Receive FIFO 0 Message Lost Flag)

The RFML0 flag is set to 1 when the RFSR0.LOST flag is set to 1, and is set to 0 when the RFSR0.LOST flag is set to 0. The RFML0 flag is cleared when the CANFD module enters GL_RESET mode.

RFML1 Flag (Receive FIFO 1 Message Lost Flag)

The RFML1 flag is set to 1 when the RFSR1.LOST flag is set to 1, and is set to 0 when the RFSR1.LOST flag is set to 0. The RFML1 flag is cleared when the CANFD module enters GL_RESET mode.

CFML0 Flag (Common FIFO 0 Message Lost Flag)

The CFML0 flag is set to 1 when the CFSR0.LOST flag is set to 1, and is set to 0 when the CFSR0.LOST flag is set to 0. The CFML0 flag is cleared when the CANFD module enters GL_RESET mode.

46.2.33 Receive FIFO Interrupt Status Register (RFISR)

Address(es): CANFD.RFISR 8747 006Ch, CANFD_NS.RFISR 0747 006Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFIF1	RFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFIF0	Receive FIFO 0 Interrupt Flag	0: Receive FIFO 0 interrupt condition is not satisfied 1: Receive FIFO 0 interrupt condition is satisfied	R
b1	RFIF1	Receive FIFO 1 Interrupt Flag	0: Receive FIFO 1 interrupt condition is not satisfied 1: Receive FIFO 1 interrupt condition is satisfied	R
b31 to b2	—	Reserved	These bits are read as 0.	R

This register shows the status of the interrupt flags of the receive FIFOs.

RFIF0 Flag (Receive FIFO 0 Interrupt Flag)

The RFIF0 flag is set to 1 when the RFSR0.RFIF flag is set to 1, and is set to 0 when the RFSR0.RFIF flag is set to 0. The RFIF0 flag is cleared when the CANFD module enters GL_RESET mode.

RFIF1 Flag (Receive FIFO 1 Interrupt Flag)

The RFIF1 flag is set to 1 when the RFSR1.RFIF flag is set to 1, and is set to 0 when the RFSR1.RFIF flag is set to 0. The RFIF1 flag is cleared when the CANFD module enters GL_RESET mode.

46.2.34 DMA Transfer Control Register (DTCR)

Address(es): CANFD.DTCR 8747 00C8h, CANFD_NS.DTCR 0747 00C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTE0	—	—	—	—	—	—	RFDTE1	RFDTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTE0	Receive FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b1	RFDTE1	Receive FIFO 1 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CFDTE0	Common FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of DTC/DMA transfer operation.

RFDTE0 Bit (Receive FIFO 0 DMA Transfer Enable)

The RFDTE0 bit enables or disables DTC/DMA transfer request for receive FIFO 0.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module enters GL_RESET mode.

RFDTE1 Bit (Receive FIFO 1 DMA Transfer Enable)

The RFDTE1 bit enables or disables DTC/DMA transfer request for receive FIFO 1.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module enters GL_RESET mode.

CFDTE0 Bit (Common FIFO 0 DMA Transfer Enable)

The CFDTE0 bit enables or disables DTC/DMA transfer request for common FIFO.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
Do not enable DTC/DMA transfer for the common FIFO that is configured as transmit FIFO.
This bit is set to 0 when the CANFD module enters GL_RESET mode.

46.2.35 DMA Transfer Status Register (DTSR)

Address(es): CANFD.DTSR 8747 00CCh, CANFD_NS.DTSR 0747 00CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTS0	—	—	—	—	—	—	RFDTS1	RFDTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTS0	Receive FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b1	RFDTS1	Receive FIFO 1 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFDTS0	Common FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the DTC/DMA transfer.

RFDTS0 Flag (Receive FIFO 0 DMA Transfer Status Flag)

The RFDTS0 flag is automatically set to 1 when the DTCR.RFDTE0 bit is set to 1 and the receive FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE0 bit is set to 0 or the receive FIFO 0 is empty.

When the DTCR.RFDTE0 bit is set to 0 while DTC/DMA transfer for the receive FIFO 0 is in progress, the RFDTS0 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module enters GL_RESET mode.

RFDTS1 Flag (Receive FIFO 1 DMA Transfer Status Flag)

The RFDTS1 flag is automatically set to 1 when the DTCR.RFDTE1 bit is set to 1 and the receive FIFO 1 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE1 bit is set to 0 or the receive FIFO 1 is empty.

When the DTCR.RFDTE1 bit is set to 0 while DTC/DMA transfer for the receive FIFO 1 is in progress, the RFDTS1 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module enters GL_RESET mode.

CFDTS0 Flag (Common FIFO 0 DMA Transfer Status Flag)

The CFDTS0 flag is automatically set to 1 when the DTCR.CFDTE0 bit is set to 1 and the common FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.CFDTE0 bit is set to 0 or the common FIFO 0 is empty.

When the DTCR.CFDTE0 bit is set to 0 while DTC/DMA transfer for the common FIFO 0 is in progress, the CFDTS0 flag becomes 0 when the DTC/DMA transfer is complete.

This flag is set to 0 when the CANFD module enters GL_RESET mode.

46.2.36 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 3)

Address(es): CANFD.TMCR0 8747 0070h, CANFD.TMCR1 8747 0071h, CANFD.TMCR2 8747 0072h, CANFD.TMCR3 8747 0073h, CANFD_NS.TMCR0 0747 0070h, CANFD_NS.TMCR1 0747 0071h, CANFD_NS.TMCR2 0747 0072h, CANFD_NS.TMCR3 0747 0073h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ONESHOT	TARQ	TXRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQ	Transmission Request	0: Do not request a message transmission 1: Request a message transmission	R/W
b1	TARQ	Transmission Abort Request	0: Do not request to abort a message transmission 1: Request to abort a message transmission	R/W
b2	ONESHOT	One-shot Transmission Enable	0: One-shot transmission is disabled. 1: One-shot transmission is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmit message buffer functions.

TXRQ Bit (Transmission Request)

When the TXRQ bit is set to 1, the CANFD module logic requests transmitting the message stored in the corresponding message buffer.

This bit can only be written in CH_HALT or CH_OPERATION mode.

This bit cannot be set to 1 if the corresponding transmit message buffer is linked to the common FIFO configured in transmit FIFO or is a part of the transmit queue.

This bit can be set to 1 only when the TMSRn.TXRF[1:0] flags are set to 00b.

This bit cannot be directly cleared by a CPU write access. The TXRQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort, requested by the TARQ bit
- When a CAN bus error or arbitration lost is detected if the ONESHOT bit is set to 1
- When the CANFD module enters GL_RESET or the CAN channel enters CH_RESET mode.

TARQ Bit (Transmission Abort Request)

When the TARQ bit is set to 1, the CANFD module logic requests aborting the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame will be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (pin for reception) before it starts transmission from the selected message buffer.

This bit can only be written in CH_HALT or CH_OPERATION mode. This bit can be set to 1 only when the TXRQ bit is set to 1.

This bit cannot be set to 0 by a CPU write access. This bit is set to 0 if writing 1 by CPU and clearing by the CAN channel occur at the same time.

The TARQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort

- When a CAN bus error or arbitration lost is detected
- When the CANFD module enters GL_RESET or the CAN channel enters CH_RESET mode.

ONESHOT Bit (One-shot Transmission Enable)

When the ONESHOT bit is set to 1, the CANFD module logic request transmitting the message only once.
If the transmission is successful, the TMSRn.TXRF[1:0] flags are set to 10b or 11b. Otherwise, the transmission is automatically aborted and TMSRn.TXRF[1:0] flags are set to 01b due to a bus error or an arbitration lost detection.
The ONESHOT bit remains 1 if the transmission has completed successfully or aborted due to a CAN bus error or an arbitration lost detection.
This bit can only be written in CH_HALT or CH_OPERATION mode.
Set this bit at the same time as the TXRQ bit. Clear this bit with a write access.
If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.
The ONESHOT bit is automatically set to 0 when the CANFD module enters GL_RESET or the CAN channel enters CH_RESET mode.

46.2.37 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 3)

Address(es): CANFD.TMSR0 8747 0074h, CANFD.TMSR1 8747 0075h, CANFD.TMSR2 8747 0076h, CANFD.TMSR3 8747 0077h, CANFD_NS.TMSR0 0747 0074h, CANFD_NS.TMSR1 0747 0075h, CANFD_NS.TMSR2 0747 0076h, CANFD_NS.TMSR3 0747 0077h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TARQS	TXRQS	TXRF[1:0]		TXSF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXSF	Transmission Status Flag	0: No transmission is in progress. 1: Transmission is in progress.	R
b2, b1	TXRF[1:0]	Transmission Result Flag	b2 b1 0 0: No result (transmission is not requested or in progress) 0 1: Transmission was aborted. 1 0: Transmission was successful and transmission abort is not requested. 1 1: Transmission was successful and transmission abort is requested.	R/W
b3	TXRQS	Transmission Request Status Flag	0: Transmission is not requested. 1: Transmission is requested.	R
b4	TARQS	Transmission Abort Request Status Flag	0: Transmission abort is not requested. 1: Transmission abort is requested.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register shows the status of the transmission and transmission abort for the transmit message buffers.

TXSF Flag (Transmission Status Flag)

The TXSF flag is automatically set to 1 at the start of the transmission from the corresponding transmit message buffer. This flag is automatically set to 0 when:

- Transmission stops
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

TXRF[1:0] Flags (Transmission Result Flag)

The TXRF[1:0] flags show the transmission result for the transmit message buffer.
These flags can only be written in CH_HALT or CH_OPERATION mode.
Each bit of these flags is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The TXRF[1:0] flags are automatically set to 00b when the CANFD module enters GL_RESET or the CAN channel enters CH_RESET mode.

TXRQS Flag (Transmission Request Status Flag)

The TXRQS flag reflects the value of the TMCRn.TXRQ bit.
The TXRQS flag is set to 1 when the TMCRn.TXRQ bit is set to 1. This flag is set to 0 when the TMCRn.TXRQ bit is set to 0.

TARQS Flag (Transmission Abort Request Status Flag)

The TARQS flag reflects the value of the TMCRn.TARQ bit.
The TARQS flag is set to 1 when the TMCRn.TARQ bit is set to 1. This flag is set to 0 when the TMCRn.TARQ bit is set to 0.

46.2.38 Transmit Message Buffer Transmission Request Status Register 0 (TMTRSR0)

Address(es): CANFD.TMTRSR0 8747 0078h, CANFD_NS.TMTRSR0 0747 0078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXRQS3	TXRQS2	TXRQS1	TXRQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQS0	Transmit Message Buffer 0 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 0. 1: Transmission is requested for the transmit message buffer 0	R
b1	TXRQS1	Transmit Message Buffer 1 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 1. 1: Transmission is requested for the transmit message buffer 1.	R
b2	TXRQS2	Transmit Message Buffer 2 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 2. 1: Transmission is requested for the transmit message buffer 2.	R
b3	TXRQS3	Transmit Message Buffer 3 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 3. 1: Transmission is requested for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register shows the status of the transmission request in each transmit message buffer.

TXRQSn Flag (Transmit Message Buffer n Transmission Request Status Flag) (n = 0 to 3)

The TXRQSn flag indicates status of the TMCRn.TXRQ bit.

Each flag is set to 1 only when the TMCRn.TXRQ bit is set to 1 and the message buffer does not belong to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TXRQ bit is set to 0
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

46.2.39 Transmit Message Buffer Transmission Abort Request Status Register 0 (TMARSR0)

Address(es): CANFD.TMARSR0 8747 007Ch, CANFD_NS.TMARSR0 0747 007Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TARQS3	TARQS2	TARQS1	TARQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TARQS0	Transmit Message Buffer 0 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 0 1: Transmission abort is requested for the transmit message buffer 0	R
b1	TARQS1	Transmit Message Buffer 1 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 1 1: Transmission abort is requested for the transmit message buffer 1	R
b2	TARQS2	Transmit Message Buffer 2 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 2 1: Transmission abort is requested for the transmit message buffer 2	R
b3	TARQS3	Transmit Message Buffer 3 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 3 1: Transmission abort is requested for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register shows the status of the transmission abort request in each transmit message buffer.

TARQSn Flag (Transmit Message Buffer n Transmission Abort Request Status Flag) (n = 0 to 3)

The TARQSn flag indicates status of the TMCRn.TARQ bit.

Each flag is set to 1 when the TMCRn.TARQ bit is set to 1 or the message buffer belongs to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TARQ bit is set to 0
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

46.2.40 Transmit Message Buffer Transmission Completion Status Register 0 (TMTCSR0)

Address(es): CANFD.TMTCSR0 8747 0080h, CANFD_NS.TMTCSR0 0747 0080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXCF3	TXCF2	TXCF1	TXCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXCF0	Transmit Message Buffer 0 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 0 1: Transmission is completed for the transmit message buffer 0	R
b1	TXCF1	Transmit Message Buffer 1 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 1 1: Transmission is completed for the transmit message buffer 1	R
b2	TXCF2	Transmit Message Buffer 2 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 2 1: Transmission is completed for the transmit message buffer 2	R
b3	TXCF3	Transmit Message Buffer 3 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 3 1: Transmission is completed for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register shows the transmission completion status for each transmit message buffer.

TXCFn Flag (Transmit Message Buffer n Transmission Completion Status) (n = 0 to 3)

The TXCFn flag indicates the transmission completion status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1] bit is set to 1.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1] bit is set to 0
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

46.2.41 Transmit Message Buffer Transmission Abort Status Register 0 (TMTASR0)

Address(es): CANFD.TMTASR0 8747 0084h, CANFD_NS.TMTASR0 0747 0084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TAF3	TAF2	TAF1	TAF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAF0	Transmit Message Buffer 0 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 0. 1: Transmission is aborted for the transmit message buffer 0.	R
b1	TAF1	Transmit Message Buffer 1 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 1. 1: Transmission is aborted for the transmit message buffer 1.	R
b2	TAF2	Transmit Message Buffer 2 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 2. 1: Transmission is aborted for the transmit message buffer 2.	R
b3	TAF3	Transmit Message Buffer 3 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 3. 1: Transmission is aborted for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register shows the transmission abort status for each transmit message buffer.

TAFn Flag (Transmit Message Buffer n Transmission Abort Status Flag) (n = 0 to 3)

The TAFn flag indicates the transmission abort status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1:0] flags are set to 01b.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1:0] flags are set to 00b
- The CANFD module enters GL_RESET mode
- The CAN channel enters CH_RESET mode.

46.2.42 Transmit Message Buffer Interrupt Enable Register (TMIER0)

Address(es): CANFD.TMIER0 8747 0088h, CANFD_NS.TMIER0 0747 0088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMIE3	TMIE2	TMIE1	TMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMIE0	Transmit Message Buffer 0 Interrupt Enable	0: Interrupt for transmit message buffer 0 is disabled. 1: Interrupt for transmit message buffer 0 is enabled.	R/W
b1	TMIE1	Transmit Message Buffer 1 Interrupt Enable	0: Interrupt for transmit message buffer 1 is disabled. 1: Interrupt for transmit message buffer 1 is enabled.	R/W
b2	TMIE2	Transmit Message Buffer 2 Interrupt Enable	0: Interrupt for transmit message buffer 2 is disabled. 1: Interrupt for transmit message buffer 2 is enabled.	R/W
b3	TMIE3	Transmit Message Buffer 3 Interrupt Enable	0: Interrupt for transmit message buffer 3 is disabled. 1: Interrupt for transmit message buffer 3 is enabled.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the interrupt for each transmit message buffer.

TMIEn Bit (Transmit Message Buffer n Interrupt Enable) (n = 0 to 3)

If the TMIEn bit is set to 1, an interrupt is generated when transmission from the transmit message buffer n is completed successfully.

Refer to section 46.10, Interrupts and DTC/DMA Requests for the specification of the interrupt for transmit message buffer.

This bit cannot be written in GL_SLEEP mode.

Do not write to the TMIEn bit when:

- The CAN channel is in CH_SLEEP mode
- The transmit message buffer n is part of a transmit queue
- The transmit message buffer n is linked to a common FIFO by the CFCR0.LTM[1:0] bits.

46.2.43 Transmit Queue 0 Configuration Register (TQCR0)

Address(es): CANFD2.TQCR0 8747 008Ch, CANFD2_NS.TQCR0 0747 008Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	QDS[1:0]		TQIM	—	TQIE	—	—	—	—	TQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQE	Transmit Queue Enable	0: Transmit queue is disabled. 1: Transmit queue is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	TQIE	Transmit Queue Interrupt Enable	0: Transmit queue transmission interrupt is disabled. 1: Transmit queue transmission interrupt is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TQIM	Transmit Queue Interrupt Mode Setting	0: When the last message is successfully transmitted 1: At every successful transmission	R/W
b9, b8	QDS[1:0]	Queue Depth Setting	b9 b8 0 0: 0 messages (disabled) 0 1: Setting prohibited 1 0: 3 messages (transmit message buffer 0 to 2) 1 1: 4 messages (transmit message buffer 0 to 3)	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the transmit queue 0.

TQE Bit (Transmit Queue Enable)

The TQE bit enables the transmit queue. When the TQE bit is set to 1, the transmit message buffer is used to configure the transmit queue.

The TQE bit cannot be set to 1 if the configured transmit queue depth is 0 (QDS[1:0] = 00b).

This bit cannot be written in GL_SLEEP mode.

Also, this bit cannot be written in CH_RESET or CH_SLEEP mode.

The TQE bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

TQIE Bit (Transmit Queue Interrupt Enable)

When the TQIE bit is set to 1, an interrupt is generated based on the setting of the TQIM bit.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQIM Bit (Transmit Queue Interrupt Mode Setting)

The TQIM bit selects the interrupt generation condition for the transmit queue.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP, CH_HALT, or CH_OPERATION mode.

QDS[1:0] Bits (Queue Depth Setting)

The QDS[1:0] bits select the depth of the transmission queue.

When these bits are set to 10b, the message buffers 0 to 2 are used, and when these bits are set to 11b, the message buffers 0 to 3 are used.
These bits cannot be written in GL_SLEEP mode.
Also, these bits cannot be written in CH_HALT or CH_OPERATION mode.
Do not write to these bits when the CAN channel is in CH_SLEEP mode.

46.2.44 Transmit Queue 0 Status Register (TQSR0)

Address(es): CANFD2.TQSR0 8747 0090h, CANFD2_NS.TQSR0 0747 0090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLVL[2:0]			—	—	—	—	—	TQIF	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmit Queue Empty Flag	0: There is at least one message in transmit queue 1: No message in transmit queue (empty)	R
b1	FULL	Transmit Queue Full Flag	0: Transmit queue is not full 1: Transmit queue is full	R
b2	TQIF	Transmit Queue Interrupt Flag	0: Transmit queue interrupt condition is not satisfied 1: Transmit queue interrupt condition is satisfied	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	FLVL[2:0]	Transmit Queue Fill Level	Indicates the number of messages in the transmit queue	R
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of the transmit queue 0.

EMPTY Flag (Transmit Queue Empty Flag)

This flag is automatically set to 1 when:

- The TQCR0.TQE bit is set to 0 (transmit queue is disabled)
- No messages are stored in the transmit queue
- The last message is transmitted from the transmit queue
- The CAN channel enters CH_RESET mode.

This flag is automatically set to 0 when the first message to be transmitted is stored in the transmit queue.

FULL Flag (Transmit Queue Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the transmit queue matches the configured transmit queue depth.

This flag is automatically set to 0 when:

- The number of messages stored in the transmit queue is less than the configured transmit queue depth
- The CAN channel enters CH_RESET mode.

TQIF Flag (Transmit Queue Interrupt Flag)

The TQIF flag is not cleared automatically if the transmit queue is disabled.

When stopping the transmit queue, this flag should be cleared, after setting the TQCR0.TQE bit to 0 and checking an empty state of transmit queue.

Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

This flag is automatically set to 1 when the configured interrupt condition is satisfied for the transmit queue.

This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag cannot be written in CH_SLEEP or CH_RESET mode.

FLVL[2:0] Bits (Transmit Queue Fill Level)

The FLVL[2:0] bits indicates the number of messages in the transmit queue.
These bits are automatically set to 000b when the CAN channel enters CH_RESET mode.

46.2.45 Transmit Queue 0 Pointer Control Register (TQPCR0)

Address(es): CANFD2.TQPCR0 8747 0094h, CANFD2_NS.TQPCR0 0747 0094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the write pointer to the transmit queue.
When the value 000000FFh is written to this register, the write pointer to the transmit queue is updated and transmission of the existing message is requested.
The read value from this register is always 00000000h.
This register cannot be written in CH_SLEEP or CH_RESET mode.
Write to this register only when:

- The transmit queue is enabled and not full.

46.2.46 Transmission History Configuration Register (THCR)

Address(es): CANFD2.THCR 8747 0098h, CANFD2_NS.THCR 0747 0098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THRC	THIM	THIE	—	—	—	—	—	—	—	THE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	THE	Transmission History Enable	0: Transmission history buffer is disabled 1: Transmission history buffer is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	THIE	Transmission History Interrupt Enable	0: Transmission history interrupt is disabled 1: Transmission history interrupt is enabled	R/W
b9	THIM	Transmission History Interrupt Mode Setting	0: Interrupt is generated if transmission history level reaches 3/4 of the transmission history depth 1: Interrupt is generated for every successfully stored entry	R/W
b10	THRC	Transmission History Recording Condition Setting	0: Transmit FIFO + Transmit queue 1: Transmit message buffer + Transmit FIFO + Transmit queue	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmission history functions.

THE Bit (Transmission History Enable)

The THE bit enables the transmission history buffer when it is set to 1.
This bit cannot be written in CH_RESET or CH_SLEEP mode.
This bit is automatically set to 0 when the CAN channel enters CH_RESET mode.

THIE Bit (Transmission History Interrupt Enable)

The THIE bit enables the generation of the transmission history interrupt when it is set to 1.
This bit cannot be written in GL_SLEEP mode.

THIM Bit (Transmission History Interrupt Mode Setting)

The THIM bit selects the transmission history interrupt generation condition.
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit in GL_HALT or GL_OPERATION mode.

THRC Bit (Transmission History Recording Condition Setting)

The THRC bit selects the condition for storing an entry in the transmission history buffer after successful transmission.
This bit cannot be written in GL_SLEEP mode.
Do not write to this bit in GL_HALT or GL_OPERATION mode.

46.2.47 Transmission History Status Register (THSR)

Address(es): CANFD2.THSR 8747 009Ch, CANFD2_NS.THSR 0747 009Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FLVL[3:0]				—	—	—	—	THIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmission History Empty Flag	0: There is a transmission history in the transmission history buffer 1: There is no transmission history in the transmission history buffer (empty)	R
b1	FULL	Transmission History Full Flag	0: Transmission history buffer is not full 1: Transmission history buffer is full	R
b2	LOST	Transmission History Lost Flag	0: Transmission history has not been lost 1: Transmission history has been lost	R/(W) *1
b3	THIF	Transmission History Interrupt Flag	0: Transmission history interrupt condition is not satisfied 1: Transmission history interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	FLVL[3:0]	Transmission History Fill Level	Number of transmission histories stored in transmission history buffer	R
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of data stored in the transmission history buffer.

EMPTY Flag (Transmission History Empty Flag)

This flag is automatically set to 0 when the first transmission history is stored to the transmission history buffer.
This flag is automatically set to 1 when:

- The CPU has read all the transmission histories from the transmission history buffer
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel enters CH_RESET mode.

FULL Flag (Transmission History Full Flag)

The FULL flag is automatically set to 1 when the number of transmission histories in the transmission history buffer is eight.
This flag is automatically set to 0 when:

- The number of transmission histories in the transmission history buffer is less than eight
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel enters CH_RESET mode.

LOST Flag (Transmission History Lost Flag)

The LOST flag is set to 1 when a new transmission history cannot be stored because the related transmission history buffer is already full.

This flag can only be written in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

THIF Flag (Transmission History Interrupt Flag)

The THIF flag is set to 1 when the configured interrupt condition is satisfied.
This flag can only be written in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel enters CH_RESET mode.

FLVL[3:0] Bits (Transmission History Fill Level)

The FLVL[3:0] bits show the number of transmission histories stored in the transmission history buffer.
These bits are automatically set to 0000b when the CAN channel enters CH_RESET mode.

46.2.48 Transmission History Access Register 0 (THACR0)

Address(es): CANFD2.THACR0 8747 0740h, CANFD2_NS.THACR0 0747 0740h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BN[1:0]		BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BT[2:0]	Transmitted Buffer Type	b2 b0 0 0 1: Transmit message buffer 0 1 0: Common FIFO 1 0 0: Transmit Queue	R
b4, b3	BN[1:0]	Transmitted Buffer Number	Number of the message buffer	R
b15 to b5	—	Reserved	These bits are read as 0.	R
b31 to b16	TS[15:0]	Transmitted Timestamp	Transmit timestamp value	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

BT[2:0] Bits (Transmitted Buffer Type)

The BT[2:0] bits indicate which type of buffer the read history came from.

BN[1:0] Bits (Transmitted Buffer Number)

The BN[1:0] bits indicate which buffer number the read history came from.
For a common FIFO, these bits indicate the number of the linked transmit message buffer.

TS[15:0] Bits (Transmitted Timestamp)

The TS[15:0] bits indicate the timestamp used by the software driver.

46.2.49 Transmission History Access Register 1 (THACR1)

Address(es): CANFD2.THACR1 8747 0744h, CANFD2_NS.THACR1 0747 0744h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PTR[15:0]	Transmitted Pointer	Stores the value of the PTR[15:0] field attached to the transmit message	R
b17, b16	IFL[1:0]	Transmitted Information Label	Stores the value of the IFL[1:0] field attached to the transmit message	R
b31 to b18	—	Reserved	These bits are read as 0.	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

PTR[15:0] Bits (Transmitted Pointer)

The PTR[15:0] bits store the value of the pointer field (TMBn.HF2.PTR[15:0] bits or CFB0.HF2.PTR[15:0] bits) attached to the transmit message.

IFL[1:0] Bits (Transmitted Information Label)

The IFL[1:0] bits store the value of the information label field (TMBn.HF2.IFL[1:0] bits or CFB0.HF2.IFL[1:0] bits) attached to the transmit message.

46.2.50 Transmission History Pointer Control Register (THPCR)

Address(es): CANFD2.THPCR 8747 00A0h, CANFD2_NS.THPCR 0747 00A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the read pointer of the transmission history buffer.
When 000000FFh is written to this register, the read pointer of the transmission history buffer is moved to the next history.
The read value from this register is 00000000h.
This register can only be written in CH_HALT or CH_OPERATION mode.
Write 000000FFh to this register only when the transmission history buffer is enabled and not empty.

46.2.51 Global Reset Control Register (GRCR)

Address(es): CANFD.GRCR 8747 00D8h, CANFD_NS.GRCR 0747 00D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SRST	Software Reset	0: Releases the software reset 1: Initiates the software reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the SRST bit. These bits are read as 00h.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SRST Bit (Software Reset)

When the SRST bit is set to 1, the CANFD module is in the same state as the MCU reset. When a reset is required, write 1 to this bit and then 0.

When the software reset is released, the CANFD module is in GL_SLEEP mode.

After a software reset, the RAM initialization sequence is not performed. Initialize the RAM with software.

Similarly, if a software reset is performed during RAM initialization, the RAM is not initialized. Initialize the RAM with software.

KEY[7:0] Bits (Key Code)

When rewriting the value of the SRST bit, set these bits to C4h at the same time (write this register in 32 bits).

46.2.52 Global Test Mode Configuration Register (GTMCR)

Address(es): CANFD.GTMCR 8747 00A8h, CANFD_NS.GTMCR 0747 00A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RTPS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	RTPS[3:0]	RAM Test Page Select	Select the page of RAM to be tested	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to specify the page number of the RAM to be tested in RAM test mode.

RTPS[3:0] Bits (RAM Test Page Select)

The RTPS[3:0] bits select the RAM page number that the CPU reads and writes when the CANFD module is in RAM test mode. Specify the page number in the range of 0 to 8.

Refer to section 46.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

These bits cannot be written in GL_RESET or GL_SLEEP mode. Write to these bits only in GL_HALT mode.

These bits are automatically set to 0000b when the CAN channel enters GL_RESET mode.

46.2.53 Global Test Mode Enable Register (GTMER)

Address(es): CANFD.GTMER 8747 00ACh, CANFD_NS.GTMER 0747 00ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RTME	RAM Test Mode Enable	0: RAM Test mode is disabled. 1: RAM Test mode is enabled.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the global test modes of the CANFD module.

RTME Bit (RAM Test Mode Enable)

When the RTME bit is set to 1, the CANFD module is in RAM test mode. Refer to section 46.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

This bit can only be set to 1 in GL_HALT mode. To exit the RAM test mode, set this bit to 0 in GL_HALT mode.

This bit is automatically set to 0 when the CANFD module enters GL_RESET mode.

46.2.54 Global CAN FD Configuration Register (GFDCFG)

Address(es): CANFD.GFDCFG 8747 00B0h, CANFD_NS.GFDCFG 0747 00B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TSCPS[1:0]		—	—	—	—	—	—	—	PXEDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PXEDIS	Protocol Exception Event Detection Disable	0: Protocol exception event detection is enabled. 1: Protocol exception event detection is disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	TSCPS[1:0]	Timestamp Capturing Point Select	b9 b8 0 0: Sample point of SOF (start of frame) 0 1: EOF (end of frame) when frame is taken valid 1 0: Sample point of SOF (Classical CAN frame), or sample point of res bit following the FDF bit (CAN FD frame) 1 1: Setting prohibited	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PXEDIS Bit (Protocol Exception Event Detection Disable)

The PXEDIS bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is set to 1, protocol exception event detection is disabled, and an error frame is transmitted if a protocol exception (a recessive reserved bit following the FDF bit) is detected.

This bit can only be written in GL_RESET mode.

TSCPS[1:0] Bits (Timestamp Capturing Point Select)

The TSCPS[1:0] bits selects the capture point of the timestamps for transmitted and received messages.

When the TSCPS[1:0] bits are 10b, the timestamp is captured at the reserved bit following the FDF bit in CAN FD frames and at the SOF in Classical CAN frames.

These bits can only be written in GL_RESET mode.

46.2.55 Global Test Mode Lock Key Register (GTMLKR)

Address(es): CANFD.GTMLKR 8747 00B8h, CANFD_NS.GTMLKR 0747 00B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to unlock the protection for RAM test mode. Refer to section 46.9.2, Global Test Modes for the specification of the lock key.
To put the CANFD module into RAM test mode, two unlock keys must be written to this register in consecutive bus cycles.
The read value from this register is always 00000000h.
This register cannot be written in GL_SLEEP or GL_RESET mode.
Do not write to this register in GL_OPERATION mode.

46.2.56 RAM Test Page Access Register k (RTPARk) (k = 0 to 63)

Address(es): CANFD.RTPAR0 8747 0280h to CANFD.RTPAR63 8747 037Ch,
CANFD_NS.RTPAR0 0747 0280h to CANFD_NS.RTPAR63 0747 037Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be read or written in RAM test mode.
This register can only be written in GL_HALT mode when RAM test mode is enabled.

46.2.57 Acceptance Filter List Ignore Entry Setting Register (AFIGSR)

Address(es): CANFD.AFIGSR 8747 00C0h, CANFD_NS.AFIGSR 0747 00C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—		—	IGES[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IGES[3:0]	Ignore Entry Select	Set the rule number to be ignored during acceptance filtering.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGES[3:0] Bits (Ignore Entry Select)

The IGES[3:0] bits are used to specify the rule number to update when updating the acceptance filter.

Write to these bits only when the AFIGER.IGEE bit is 0.

These bits cannot be written in GL_SLEEP mode.

46.2.58 Acceptance Filter List Ignore Entry Enable Register (AFIGER)

Address(es): CANFD.AFIGER 8747 00C4h, CANFD_NS.AFIGER 0747 00C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	IGEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGEE	Ignore Entry Enable	0: The value of the AFIGSR.IGES[3:0] bits is invalid. 1: The value of the AFIGSR.IGES[3:0] bits is valid.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the IGEE bit. These bits are read as 00h.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGEE Bit (Ignore Entry Enable)

When the IGEE bit is set to 1, the entry selected by the AFIGSR.IGES[3:0] bits is ignored.

This bit is automatically set to 0 when the CANFD module enters GL_RESET mode.

KEY[7:0] Bits (Key Code)

When rewriting the value of the IGEE bit, set these bits to C4h at the same time (write this register in 32 bits).

46.2.59 Receive Message Buffer Interrupt Enable Register (RMIER)

Address(es): CANFD.RMIER 8747 0038h, CANFD_NS.RMIER 0747 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMIE15	RMIE14	RMIE13	RMIE12	RMIE11	RMIE10	RMIE9	RMIE8	RMIE7	RMIE6	RMIE5	RMIE4	RMIE3	RMIE2	RMIE1	RMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RMIE0	Receive Message Buffer 0 Interrupt Enable	0: Interrupt for receive message buffer 0 is disabled. 1: Interrupt for receive message buffer 0 is enabled.	R/W
b1	RMIE1	Receive Message Buffer 1 Interrupt Enable	0: Interrupt for receive message buffer 1 is disabled. 1: Interrupt for receive message buffer 1 is enabled.	R/W
b2	RMIE2	Receive Message Buffer 2 Interrupt Enable	0: Interrupt for receive message buffer 2 is disabled. 1: Interrupt for receive message buffer 2 is enabled.	R/W
b3	RMIE3	Receive Message Buffer 3 Interrupt Enable	0: Interrupt for receive message buffer 3 is disabled. 1: Interrupt for receive message buffer 3 is enabled.	R/W
b4	RMIE4	Receive Message Buffer 4 Interrupt Enable	0: Interrupt for receive message buffer 4 is disabled. 1: Interrupt for receive message buffer 4 is enabled.	R/W
b5	RMIE5	Receive Message Buffer 5 Interrupt Enable	0: Interrupt for receive message buffer 5 is disabled. 1: Interrupt for receive message buffer 5 is enabled.	R/W
b6	RMIE6	Receive Message Buffer 6 Interrupt Enable	0: Interrupt for receive message buffer 6 is disabled. 1: Interrupt for receive message buffer 6 is enabled.	R/W
b7	RMIE7	Receive Message Buffer 7 Interrupt Enable	0: Interrupt for receive message buffer 7 is disabled. 1: Interrupt for receive message buffer 7 is enabled.	R/W
b8	RMIE8	Receive Message Buffer 8 Interrupt Enable	0: Interrupt for receive message buffer 8 is disabled. 1: Interrupt for receive message buffer 8 is enabled.	R/W
b9	RMIE9	Receive Message Buffer 9 Interrupt Enable	0: Interrupt for receive message buffer 9 is disabled. 1: Interrupt for receive message buffer 9 is enabled.	R/W
b10	RMIE10	Receive Message Buffer 10 Interrupt Enable	0: Interrupt for receive message buffer 10 is disabled. 1: Interrupt for receive message buffer 10 is enabled.	R/W
b11	RMIE11	Receive Message Buffer 11 Interrupt Enable	0: Interrupt for receive message buffer 11 is disabled. 1: Interrupt for receive message buffer 11 is enabled.	R/W
b12	RMIE12	Receive Message Buffer 12 Interrupt Enable	0: Interrupt for receive message buffer 12 is disabled. 1: Interrupt for receive message buffer 12 is enabled.	R/W
b13	RMIE13	Receive Message Buffer 13 Interrupt Enable	0: Interrupt for receive message buffer 13 is disabled. 1: Interrupt for receive message buffer 13 is enabled.	R/W
b14	RMIE14	Receive Message Buffer 14 Interrupt Enable	0: Interrupt for receive message buffer 14 is disabled. 1: Interrupt for receive message buffer 14 is enabled.	R/W
b15	RMIE15	Receive Message Buffer 15 Interrupt Enable	0: Interrupt for receive message buffer 15 is disabled. 1: Interrupt for receive message buffer 15 is enabled.	R/W

This register enables or disables the interrupt for each receive message buffer.

RMIEn Bit (Receive Message Buffer n Interrupt Enable) (n = 0 to 15)

If the RMIEn bit is set to 1, an interrupt is generated when reception to the receive message buffer n is completed successfully.

Refer to section 46.10, Interrupts and DTC/DMA Requests for the specification of the receive message buffer interrupt.

This bit cannot be written in GL_SLEEP mode.

46.2.60 Identifier Bits Alignment

Base format (11-bit length identifier): ID-28 to ID-18 are located in b10 to b0, and b28 to b11 are 0.
Extended format (29-bit length identifier): ID-28 to ID-0 are located in b28 to b0

Table 46.4 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18

Table 46.5 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18	ID-17	ID-16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0

46.2.61 Message Buffer Structure

The message buffer configuration consists of four types of message buffers:

- Receive message buffer (RMBn)
- Receive FIFO (RFBn)
- Common FIFO (CFB0)
- Transmit message buffer (TMBn)

“n” is a message buffer number whose range changes depending on the type of message buffer.
Refer to Figure 46.32 for an overview of this configuration. For more information on the number and types of message buffers, refer to section 46.6, FIFO Buffers and Message Buffer Configuration.

46.2.61.1 Start Addresses

The start address of each message buffer is calculated using the number of related message buffer components.
The start addresses for each register in the message buffer are listed in Table 46.6.

Table 46.6 Start Address for Each Register of the Message Buffer Component

Message Buffer	Symbol	n	Register	p	Start Address (Only Secure Addresses Are Listed)
Receive Message Buffer	RMBn	0 to 7	HF0	—	8747 0920h + n × 4Ch
			HF1	—	8747 0924h + n × 4Ch
			HF2	—	8747 0928h + n × 4Ch
			DFp	0 to 15	8747 092Ch + n × 4Ch + p × 4
	RMBn	8 to 15	HF0	—	8747 0D20h + (n – 8) × 4Ch
			HF1	—	8747 0D24h + (n – 8) × 4Ch
			HF2	—	8747 0D28h + (n – 8) × 4Ch
			DFp	0 to 15	8747 0D2Ch + (n – 8) × 4Ch + p × 4
Receive FIFO	RFBn	0, 1	HF0	—	8747 0520h + n × 4Ch
			HF1	—	8747 0524h + n × 4Ch
			HF2	—	8747 0528h + n × 4Ch
			DFp	0 to 15	8747 052Ch + n × 4Ch + p × 4
Common FIFO	CFB0	0	HF0	—	8747 05B8h
			HF1	—	8747 05BCh
			HF2	—	8747 05C0h
			DFp	0 to 15	8747 05C4h + p × 4
Transmit Message Buffer	TMBn	0 to 3	HF0	—	8747 0604h + n × 4Ch
			HF1	—	8747 0608h + n × 4Ch
			HF2	—	8747 060Ch + n × 4Ch
			DFp	0 to 15	8747 0610h + n × 4Ch + p × 4

46.2.61.2 Receive Message Buffer n (RMBn) (n = 0 to 15)

The total number of the receive message buffers (RMB) is 16, as shown in Figure 46.32.
The receive message buffer consists of the following registers:

- RMBn.HF0
- RMBn.HF1
- RMBn.HF2
- RMBn.DF0 to RMBn.DF15

Table 46.7 shows the structure of this buffer.

Table 46.7 Structure of Receive Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	RMBn.HF0	Receive Message Buffer n Header Field 0	IDE, RTR, ID
+04h	RMBn.HF1	Receive Message Buffer n Header Field 1	DLC, timestamp
+08h	RMBn.HF2	Receive Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RMBn.DF0	Receive Message Buffer n Data Field 0	DATA0 to DATA3
+10h	RMBn.DF1	Receive Message Buffer n Data Field 1	DATA4 to DATA7
+14h	RMBn.DF2	Receive Message Buffer n Data Field 2	DATA8 to DATA11
+18h	RMBn.DF3	Receive Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	RMBn.DF4	Receive Message Buffer n Data Field 4	DATA16 to DATA19
+20h	RMBn.DF5	Receive Message Buffer n Data Field 5	DATA20 to DATA23
+24h	RMBn.DF6	Receive Message Buffer n Data Field 6	DATA24 to DATA27
+28h	RMBn.DF7	Receive Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	RMBn.DF8	Receive Message Buffer n Data Field 8	DATA32 to DATA35
+30h	RMBn.DF9	Receive Message Buffer n Data Field 9	DATA36 to DATA39
+34h	RMBn.DF10	Receive Message Buffer n Data Field 10	DATA40 to DATA43
+38h	RMBn.DF11	Receive Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	RMBn.DF12	Receive Message Buffer n Data Field 12	DATA48 to DATA51
+40h	RMBn.DF13	Receive Message Buffer n Data Field 13	DATA52 to DATA55
+44h	RMBn.DF14	Receive Message Buffer n Data Field 14	DATA56 to DATA59
+48h	RMBn.DF15	Receive Message Buffer n Data Field 15	DATA60 to DATA63

46.2.61.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 15)

Address(es): CANFD.RMB0.HF0 8747 0920h, CANFD.RMB1.HF0 8747 096Ch, CANFD.RMB2.HF0 8747 09B8h, CANFD.RMB3.HF0 8747 0A04h, CANFD.RMB4.HF0 8747 0A50h, CANFD.RMB5.HF0 8747 0A9Ch, CANFD.RMB6.HF0 8747 0AE8h, CANFD.RMB7.HF0 8747 0B34h, CANFD.RMB8.HF0 8747 0D20h, CANFD.RMB9.HF0 8747 0D6Ch, CANFD.RMB10.HF0 8747 0DB8h, CANFD.RMB11.HF0 8747 0E04h, CANFD.RMB12.HF0 8747 0E50h, CANFD.RMB13.HF0 8747 0E9Ch, CANFD.RMB14.HF0 8747 0EE8h, CANFD.RMB15.HF0 8747 0F34h,
CANFD_NS.RMB0.HF0 0747 0920h, CANFD_NS.RMB1.HF0 0747 096Ch, CANFD_NS.RMB2.HF0 0747 09B8h, CANFD_NS.RMB3.HF0 0747 0A04h, CANFD_NS.RMB4.HF0 0747 0A50h, CANFD_NS.RMB5.HF0 0747 0A9Ch, CANFD_NS.RMB6.HF0 0747 0AE8h, CANFD_NS.RMB7.HF0 0747 0B34h, CANFD_NS.RMB8.HF0 0747 0D20h, CANFD_NS.RMB9.HF0 0747 0D6Ch, CANFD_NS.RMB10.HF0 0747 0DB8h, CANFD_NS.RMB11.HF0 0747 0E04h, CANFD_NS.RMB12.HF0 0747 0E50h, CANFD_NS.RMB13.HF0 0747 0E9Ch, CANFD_NS.RMB14.HF0 0747 0EE8h, CANFD_NS.RMB15.HF0 0747 0F34h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0.	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive message buffer.
For alignment of these bits in base and extended format, refer to section 46.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.
The RTR bit indicates whether the receive message buffer stores a data frame or a remote frame.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.
The IDE bit indicates whether the message stored in the receive message buffer has a standard identifier or extended identifier.

46.2.61.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 15)

Address(es): CANFD.RMB0.HF1 8747 0924h, CANFD.RMB1.HF1 8747 0970h, CANFD.RMB2.HF1 8747 09BCh, CANFD.RMB3.HF1 8747 0A08h, CANFD.RMB4.HF1 8747 0A54h, CANFD.RMB5.HF1 8747 0AA0h, CANFD.RMB6.HF1 8747 0AECh, CANFD.RMB7.HF1 8747 0B38h, CANFD.RMB8.HF1 8747 0D24h, CANFD.RMB9.HF1 8747 0D70h, CANFD.RMB10.HF1 8747 0DBCh, CANFD.RMB11.HF1 8747 0E08h, CANFD.RMB12.HF1 8747 0E54h, CANFD.RMB13.HF1 8747 0EA0h, CANFD.RMB14.HF1 8747 0EECh, CANFD.RMB15.HF1 8747 0F38h,
CANFD_NS.RMB0.HF1 0747 0924h, CANFD_NS.RMB1.HF1 0747 0970h, CANFD_NS.RMB2.HF1 0747 09BCh, CANFD_NS.RMB3.HF1 0747 0A08h, CANFD_NS.RMB4.HF1 0747 0A54h, CANFD_NS.RMB5.HF1 0747 0AA0h, CANFD_NS.RMB6.HF1 0747 0AECh, CANFD_NS.RMB7.HF1 0747 0B38h, CANFD_NS.RMB8.HF1 0747 0D24h, CANFD_NS.RMB9.HF1 0747 0D70h, CANFD_NS.RMB10.HF1 0747 0DBCh, CANFD_NS.RMB11.HF1 0747 0E08h, CANFD_NS.RMB12.HF1 0747 0E54h, CANFD_NS.RMB13.HF1 0747 0EA0h, CANFD_NS.RMB14.HF1 0747 0EECh, CANFD_NS.RMB15.HF1 0747 0F38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value stored for the message in the receive message buffer	R
b27 to b16	—	Reserved	These bits are read as 0.	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame.	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

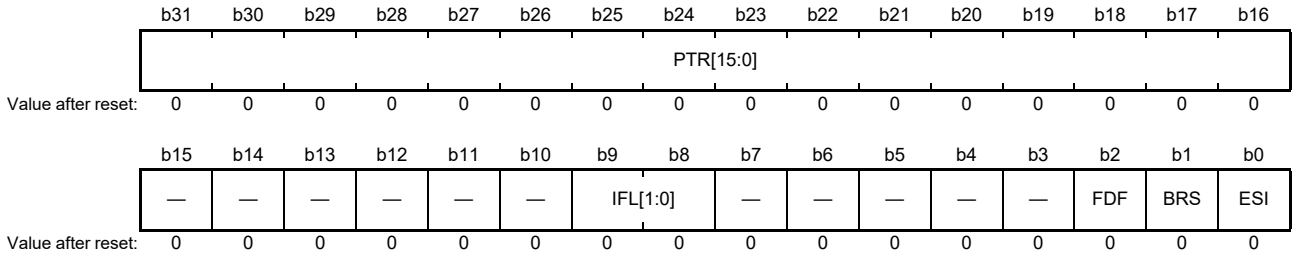
DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RMCR.PLS[2:0] bits.

46.2.61.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 15)

Address(es): CANFD.RMB0.HF2 8747 0928h, CANFD.RMB1.HF2 8747 0974h, CANFD.RMB2.HF2 8747 09C0h, CANFD.RMB3.HF2 8747 0A0Ch, CANFD.RMB4.HF2 8747 0A58h, CANFD.RMB5.HF2 8747 0AA4h, CANFD.RMB6.HF2 8747 0AF0h, CANFD.RMB7.HF2 8747 0B3Ch, CANFD.RMB8.HF2 8747 0D28h, CANFD.RMB9.HF2 8747 0D74h, CANFD.RMB10.HF2 8747 0DC0h, CANFD.RMB11.HF2 8747 0E0Ch, CANFD.RMB12.HF2 8747 0E58h, CANFD.RMB13.HF2 8747 0EA4h, CANFD.RMB14.HF2 8747 0EF0h, CANFD.RMB15.HF2 8747 0F3Ch,
CANFD_NS.RMB0.HF2 0747 0928h, CANFD_NS.RMB1.HF2 0747 0974h, CANFD_NS.RMB2.HF2 0747 09C0h, CANFD_NS.RMB3.HF2 0747 0A0Ch, CANFD_NS.RMB4.HF2 0747 0A58h, CANFD_NS.RMB5.HF2 0747 0AA4h, CANFD_NS.RMB6.HF2 0747 0AF0h, CANFD_NS.RMB7.HF2 0747 0B3Ch, CANFD_NS.RMB8.HF2 0747 0D28h, CANFD_NS.RMB9.HF2 0747 0D74h, CANFD_NS.RMB10.HF2 0747 0DC0h, CANFD_NS.RMB11.HF2 0747 0E0Ch, CANFD_NS.RMB12.HF2 0747 0E58h, CANFD_NS.RMB13.HF2 0747 0EA4h, CANFD_NS.RMB14.HF2 0747 0EF0h, CANFD_NS.RMB15.HF2 0747 0F3Ch



Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0.	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

46.2.61.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 15; p = 0 to 15)

Address(es): CANFD.RMB0.DF0 8747 092Ch to CANFD.RMB0.DF15 8747 0968h,
CANFD.RMB1.DF0 8747 0978h to CANFD.RMB1.DF15 8747 09B4h,
CANFD.RMB2.DF0 8747 09C4h to CANFD.RMB2.DF15 8747 0A00h,
CANFD.RMB3.DF0 8747 0A10h to CANFD.RMB3.DF15 8747 0A4Ch,
CANFD.RMB4.DF0 8747 0A5Ch to CANFD.RMB4.DF15 8747 0A98h,
CANFD.RMB5.DF0 8747 0AA8h to CANFD.RMB5.DF15 8747 0AE4h,
CANFD.RMB6.DF0 8747 0AF4h to CANFD.RMB6.DF15 8747 0B30h,
CANFD.RMB7.DF0 8747 0B40h to CANFD.RMB7.DF15 8747 0B7Ch,
CANFD.RMB8.DF0 8747 0D2Ch to CANFD.RMB8.DF15 8747 0D68h,
CANFD.RMB9.DF0 8747 0D78h to CANFD.RMB9.DF15 8747 0DB4h,
CANFD.RMB10.DF0 8747 0DC4h to CANFD.RMB10.DF15 8747 0E00h,
CANFD.RMB11.DF0 8747 0E10h to CANFD.RMB11.DF15 8747 0E4Ch,
CANFD.RMB12.DF0 8747 0E5Ch to CANFD.RMB12.DF15 8747 0E98h,
CANFD.RMB13.DF0 8747 0EA8h to CANFD.RMB13.DF15 8747 0EE4h,
CANFD.RMB14.DF0 8747 0EF4h to CANFD.RMB14.DF15 8747 0F30h,
CANFD.RMB15.DF0 8747 0F40h to CANFD.RMB15.DF15 8747 0F7Ch,
CANFD_NS.RMB0.DF0 0747 092Ch to CANFD_NS.RMB0.DF15 0747 0968h,
CANFD_NS.RMB1.DF0 0747 0978h to CANFD_NS.RMB1.DF15 0747 09B4h,
CANFD_NS.RMB2.DF0 0747 09C4h to CANFD_NS.RMB2.DF15 0747 0A00h,
CANFD_NS.RMB3.DF0 0747 0A10h to CANFD_NS.RMB3.DF15 0747 0A4Ch,
CANFD_NS.RMB4.DF0 0747 0A5Ch to CANFD_NS.RMB4.DF15 0747 0A98h,
CANFD_NS.RMB5.DF0 0747 0AA8h to CANFD_NS.RMB5.DF15 0747 0AE4h,
CANFD_NS.RMB6.DF0 0747 0AF4h to CANFD_NS.RMB6.DF15 0747 0B30h,
CANFD_NS.RMB7.DF0 0747 0B40h to CANFD_NS.RMB7.DF15 0747 0B7Ch,
CANFD_NS.RMB8.DF0 0747 0D2Ch to CANFD_NS.RMB8.DF15 0747 0D68h,
CANFD_NS.RMB9.DF0 0747 0D78h to CANFD_NS.RMB9.DF15 0747 0DB4h,
CANFD_NS.RMB10.DF0 0747 0DC4h to CANFD_NS.RMB10.DF15 0747 0E00h,
CANFD_NS.RMB11.DF0 0747 0E10h to CANFD_NS.RMB11.DF15 0747 0E4Ch,
CANFD_NS.RMB12.DF0 0747 0E5Ch to CANFD_NS.RMB12.DF15 0747 0E98h,
CANFD_NS.RMB13.DF0 0747 0EA8h to CANFD_NS.RMB13.DF15 0747 0EE4h,
CANFD_NS.RMB14.DF0 0747 0EF4h to CANFD_NS.RMB14.DF15 0747 0F30h,
CANFD_NS.RMB15.DF0 0747 0F40h to CANFD_NS.RMB15.DF15 0747 0F7Ch

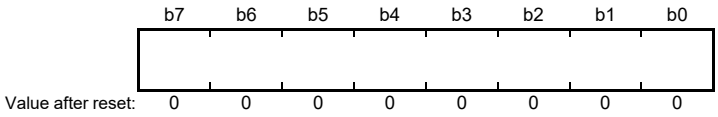
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

46.2.61.7 Receive Message Buffer n Data k (RMBn.DATAk) (n = 0 to 15; k = 0 to 63)

Address(es): CANFD.RMB0.DATA0 8747 092Ch to CANFD.RMB0.DATA63 8747 096Bh,
CANFD.RMB1.DATA0 8747 0978h to CANFD.RMB1.DATA63 8747 09B7h,
CANFD.RMB2.DATA0 8747 09C4h to CANFD.RMB2.DATA63 8747 0A03h,
CANFD.RMB3.DATA0 8747 0A10h to CANFD.RMB3.DATA63 8747 0A4Fh,
CANFD.RMB4.DATA0 8747 0A5Ch to CANFD.RMB4.DATA63 8747 0A9Bh,
CANFD.RMB5.DATA0 8747 0AA8h to CANFD.RMB5.DATA63 8747 0AE7h,
CANFD.RMB6.DATA0 8747 0AF4h to CANFD.RMB6.DATA63 8747 0B33h,
CANFD.RMB7.DATA0 8747 0B40h to CANFD.RMB7.DATA63 8747 0B7Fh,
CANFD.RMB8.DATA0 8747 0D2Ch to CANFD.RMB8.DATA63 8747 0D6Bh,
CANFD.RMB9.DATA0 8747 0D78h to CANFD.RMB9.DATA63 8747 0DB7h,
CANFD.RMB10.DATA0 8747 0DC4h to CANFD.RMB10.DATA63 8747 0E03h,
CANFD.RMB11.DATA0 8747 0E10h to CANFD.RMB11.DATA63 8747 0E4Fh,
CANFD.RMB12.DATA0 8747 0E5Ch to CANFD.RMB12.DATA63 8747 0E9Bh,
CANFD.RMB13.DATA0 8747 0EA8h to CANFD.RMB13.DATA63 8747 0EE7h,
CANFD.RMB14.DATA0 8747 0EF4h to CANFD.RMB14.DATA63 8747 0F33h,
CANFD.RMB15.DATA0 8747 0F40h to CANFD.RMB15.DATA63 8747 0F7Fh,
CANFD_NS.RMB0.DATA0 0747 092Ch to CANFD_NS.RMB0.DATA63 0747 096Bh,
CANFD_NS.RMB1.DATA0 0747 0978h to CANFD_NS.RMB1.DATA63 0747 09B7h,
CANFD_NS.RMB2.DATA0 0747 09C4h to CANFD_NS.RMB2.DATA63 0747 0A03h,
CANFD_NS.RMB3.DATA0 0747 0A10h to CANFD_NS.RMB3.DATA63 0747 0A4Fh,
CANFD_NS.RMB4.DATA0 0747 0A5Ch to CANFD_NS.RMB4.DATA63 0747 0A9Bh,
CANFD_NS.RMB5.DATA0 0747 0AA8h to CANFD_NS.RMB5.DATA63 0747 0AE7h,
CANFD_NS.RMB6.DATA0 0747 0AF4h to CANFD_NS.RMB6.DATA63 0747 0B33h,
CANFD_NS.RMB7.DATA0 0747 0B40h to CANFD_NS.RMB7.DATA63 0747 0B7Fh,
CANFD_NS.RMB8.DATA0 0747 0D2Ch to CANFD_NS.RMB8.DATA63 0747 0D6Bh,
CANFD_NS.RMB9.DATA0 0747 0D78h to CANFD_NS.RMB9.DATA63 0747 0DB7h,
CANFD_NS.RMB10.DATA0 0747 0DC4h to CANFD_NS.RMB10.DATA63 0747 0E03h,
CANFD_NS.RMB11.DATA0 0747 0E10h to CANFD_NS.RMB11.DATA63 0747 0E4Fh,
CANFD_NS.RMB12.DATA0 0747 0E5Ch to CANFD_NS.RMB12.DATA63 0747 0E9Bh,
CANFD_NS.RMB13.DATA0 0747 0EA8h to CANFD_NS.RMB13.DATA63 0747 0EE7h,
CANFD_NS.RMB14.DATA0 0747 0EF4h to CANFD_NS.RMB14.DATA63 0747 0F33h,
CANFD_NS.RMB15.DATA0 0747 0F40h to CANFD_NS.RMB15.DATA63 0747 0F7Fh



These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

46.2.61.8 Receive FIFO n (RFBn) (n = 0, 1)

The total number of the receive FIFOs (RFB) is two, as shown in Figure 46.32.
The receive FIFO consists of the following registers:

- RFBn.HF0
- RFBn.HF1
- RFBn.HF2
- RFBn.DF0 to RFBn.DF15

Table 46.8 shows the structure of this buffer.

Table 46.8 Structure of Receive FIFO

Address Offset	Symbol	Register Name	Contents
+00h	RFBn.HF0	Receive FIFO n Header Field 0	IDE, RTR, ID
+04h	RFBn.HF1	Receive FIFO n Header Field 1	DLC, timestamp
+08h	RFBn.HF2	Receive FIFO n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RFBn.DF0	Receive FIFO n Data Field 0	DATA0 to DATA3
+10h	RFBn.DF1	Receive FIFO n Data Field 1	DATA4 to DATA7
+14h	RFBn.DF2	Receive FIFO n Data Field 2	DATA8 to DATA11
+18h	RFBn.DF3	Receive FIFO n Data Field 3	DATA12 to DATA15
+1Ch	RFBn.DF4	Receive FIFO n Data Field 4	DATA16 to DATA19
+20h	RFBn.DF5	Receive FIFO n Data Field 5	DATA20 to DATA23
+24h	RFBn.DF6	Receive FIFO n Data Field 6	DATA24 to DATA27
+28h	RFBn.DF7	Receive FIFO n Data Field 7	DATA28 to DATA31
+2Ch	RFBn.DF8	Receive FIFO n Data Field 8	DATA32 to DATA35
+30h	RFBn.DF9	Receive FIFO n Data Field 9	DATA36 to DATA39
+34h	RFBn.DF10	Receive FIFO n Data Field 10	DATA40 to DATA43
+38h	RFBn.DF11	Receive FIFO n Data Field 11	DATA44 to DATA47
+3Ch	RFBn.DF12	Receive FIFO n Data Field 12	DATA48 to DATA51
+40h	RFBn.DF13	Receive FIFO n Data Field 13	DATA52 to DATA55
+44h	RFBn.DF14	Receive FIFO n Data Field 14	DATA56 to DATA59
+48h	RFBn.DF15	Receive FIFO n Data Field 15	DATA60 to DATA63

46.2.61.9 Receive FIFO n Header Field 0 (RFBn.HF0) (n = 0, 1)

Address(es): CANFD.RFB0.HF0 8747 0520h, CANFD.RFB1.HF0 8747 056Ch,
CANFD_NS.RFB0.HF0 0747 0520h, CANFD_NS.RFB1.HF0 0747 056Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28-b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0.	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive FIFO.
For alignment of these bits in base and extended format, refer to section 46.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.

46.2.61.10 Receive FIFO n Header Field 1 (RFBn.HF1) (n = 0, 1)

Address(es): CANFD.RFB0.HF1 8747 0524h, CANFD.RFB1.HF1 8747 0570h,
CANFD_NS.RFB0.HF1 0747 0524h, CANFD_NS.RFB1.HF1 0747 0570h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame	R
b27 to b16	—	Reserved	These bits are read as 0.	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RFCRn.PLS[2:0] bits.

46.2.61.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0, 1)

Address(es): CANFD.RFB0.HF2 8747 0528h, CANFD.RFB1.HF2 8747 0574h,
CANFD_NS.RFB0.HF2 0747 0528h, CANFD_NS.RFB1.HF2 0747 0574h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0.	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

46.2.61.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0, 1; p = 0 to 15)

Address(es): CANFD.RFB0.DF0 8747 052Ch to CANFD.RFB0.DF15 8747 0568h,
CANFD.RFB1.DF0 8747 0578h to CANFD.RFB1.DF15 8747 05B4h,
CANFD_NS.RFB0.DF0 0747 052Ch to CANFD_NS.RFB0.DF15 0747 0568h,
CANFD_NS.RFB1.DF0 0747 0578h to CANFD_NS.RFB1.DF15 0747 05B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

46.2.61.13 Receive FIFO n Data k (RFBn.DATAk) (n = 0, 1; k = 0 to 63)

Address(es): CANFD.RFB0.DATA0 8747 052Ch to CANFD.RFB0.DATA63 8747 056Bh,
CANFD.RFB1.DATA0 8747 0578h to CANFD.RFB1.DATA63 8747 05B7h,
CANFD_NS.RFB0.DATA0 0747 052Ch to CANFD_NS.RFB0.DATA63 0747 056Bh,
CANFD_NS.RFB1.DATA0 0747 0578h to CANFD_NS.RFB1.DATA63 0747 05B7h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

46.2.61.14 Common FIFO 0 (CFB0)

The total number of the common FIFO (CFB) is one, as shown in Figure 46.32.
The common FIFO consists of the following registers:

- CFB0.HF0
- CFB0.HF1
- CFB0.HF2
- CFB0.DF0 to CFB0.DF15

Table 46.9 shows the structure of this buffer.

Table 46.9 Structure of Common FIFO

Address Offset	Symbol	Register Name	Contents
+00h	CFB0.HF0	Common FIFO 0 Header Field 0	IDE, RTR, ID
+04h	CFB0.HF1	Common FIFO 0 Header Field 1	DLC, timestamp
+08h	CFB0.HF2	Common FIFO 0 Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	CFB0.DF0	Common FIFO 0 Data Field 0	DATA0 to DATA3
+10h	CFB0.DF1	Common FIFO 0 Data Field 1	DATA4 to DATA7
+14h	CFB0.DF2	Common FIFO 0 Data Field 2	DATA8 to DATA11
+18h	CFB0.DF3	Common FIFO 0 Data Field 3	DATA12 to DATA15
+1Ch	CFB0.DF4	Common FIFO 0 Data Field 4	DATA16 to DATA19
+20h	CFB0.DF5	Common FIFO 0 Data Field 5	DATA20 to DATA23
+24h	CFB0.DF6	Common FIFO 0 Data Field 6	DATA24 to DATA27
+28h	CFB0.DF7	Common FIFO 0 Data Field 7	DATA28 to DATA31
+2Ch	CFB0.DF8	Common FIFO 0 Data Field 8	DATA32 to DATA35
+30h	CFB0.DF9	Common FIFO 0 Data Field 9	DATA36 to DATA39
+34h	CFB0.DF10	Common FIFO 0 Data Field 10	DATA40 to DATA43
+38h	CFB0.DF11	Common FIFO 0 Data Field 11	DATA44 to DATA47
+3Ch	CFB0.DF12	Common FIFO 0 Data Field 12	DATA48 to DATA51
+40h	CFB0.DF13	Common FIFO 0 Data Field 13	DATA52 to DATA55
+44h	CFB0.DF14	Common FIFO 0 Data Field 14	DATA56 to DATA59
+48h	CFB0.DF15	Common FIFO 0 Data Field 15	DATA60 to DATA63

46.2.61.15 Common FIFO 0 Header Field 0 (CFB0.HF0)

Address(es): CANFD.CFB0.HF0 8747 05B8h, CANFD_NS.CFB0.HF0 0747 05B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	Receive FIFO mode: Reserved. This bit is read as 0. Transmit FIFO mode: 0: Entry is not stored to the Transmission History after successful transmission. 1: Entry is stored to the Transmission History after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

In receive FIFO mode, this register is a read-only register for reading the ID field, IDE bit, and RTR bit of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the ID field, IDE bit, and RTR bit of the message to be transmitted at the end of the FIFO buffer.

ID[28:0] Bits (Identifier)

In receive FIFO mode, the ID[28:0] bits store the standard or extended identifier field of the received message.
In transmit FIFO mode, the ID[28:0] bits are used to set the value of the standard or extended identifier field of the message to be transmitted.
For alignment of these bits in base and extended format, refer to section 46.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

This bit is valid only in transmit FIFO mode.
The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message.

RTR Bit (Remote Transmission Request)

In receive FIFO mode, the RTR bit stores the value of the RTR bit of the received message.
In transmit FIFO mode, the RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When a CAN FD frame was received (receive mode), this bit reflects the value of the RRS bit. When transmitting a CAN FD frame (CFB0.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

In receive FIFO mode, the IDE bit stores the value of the IDE bit of the received message.
In transmit FIFO mode, the IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

46.2.61.16 Common FIFO 0 Header Field 1 (CFB0.HF1)

Address(es): CANFD.CFB0.HF1 8747 05BCh, CANFD_NS.CFB0.HF1 0747 05BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame (in receive FIFO mode)	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame	R/W

In receive FIFO mode, this register is a read-only register for reading the data length code (DLC) and timestamp of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the data length code (DLC) of the message to be transmitted at the end of the FIFO buffer.

TS[15:0] Bits (Timestamp)

These bits are valid only in receive FIFO mode.
The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

In receive FIFO mode, the DLC[3:0] bits store the number of data bytes of the received message.
In transmit FIFO mode, The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the CFCR0.PLS[2:0] bits.

46.2.61.17 Common FIFO 0 Header Field 2 (CFB0.HF2)

Address(es): CANFD.CFB0.HF2 8747 05C0h, CANFD_NS.CFB0.HF2 0747 05C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame received or to transmit by error active node 1: CAN FD frame received or to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame received or to transmit with no bit rate switch 1: CAN FD frame received or to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame received or to transmit 1: CAN FD frame received or to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter or sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter or sets the pointer to be stored in the Transmission History	R/W

In receive FIFO mode, this register is a read-only register for reading the FDF bit, BRS bit, and ESI flag of the received message, and information label and pointer for the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history at the end of the FIFO buffer.

ESI Bit (Error State Indicator)

In receive FIFO mode, the ESI bit stores the value of the ESI flag of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted. If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

In receive FIFO mode, the BRS bit stores the value of the BRS bit of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

In receive FIFO mode, the FDF bit stores the value of the FDF bit of the received CAN FD frame.
In transmit FIFO mode, the FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

In receive FIFO mode, the IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

In receive FIFO mode, the PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

46.2.61.18 Common FIFO 0 Data Field p (CFB0.DFp) (p = 0 to 15)

Address(es): CANFD.CFB0.DF0 8747 05C4h to CANFD.CFB0.DF15 8747 0600h,
CANFD_NS.CFB0.DF0 0747 05C4h to CANFD_NS.CFB0.DF15 0747 0600h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In receive FIFO mode, these registers are read-only registers for reading the data byte (p × 4 + 3) to data byte (p × 4) of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.

46.2.61.19 Common FIFO 0 Data k (CFB0.DATAk) (k = 0 to 63)

Address(es): CANFD.CFB0.DATA0 8747 05C4h to CANFD.CFB0.DATA63 8747 0603h,
CANFD_NS.CFB0.DATA0 0747 05C4h to CANFD_NS.CFB0.DATA63 0747 0603h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

In receive FIFO mode, these registers are read-only registers for reading the data bytes of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing the data bytes of the message to be transmitted.

46.2.61.20 Transmit Message Buffer n (TMBn) (n = 0 to 3)

The total number of transmit message buffers (TMB) is four, as shown in Figure 46.32.
The transmit message buffer consists of the following registers:

- TMBn.HF0
- TMBn.HF1
- TMBn.HF2
- TMBn.DF0 to TMBn.DF15

Table 46.10 shows the structure of this buffer.

Table 46.10 Structure of Transmit Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	TMBn.HF0	Transmit Message Buffer n Header Field 0	IDE, RTR, ID
+04h	TMBn.HF1	Transmit Message Buffer n Header Field 1	DLC
+08h	TMBn.HF2	Transmit Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	TMBn.DF0	Transmit Message Buffer n Data Field 0	DATA0 to DATA3
+10h	TMBn.DF1	Transmit Message Buffer n Data Field 1	DATA4 to DATA7
+14h	TMBn.DF2	Transmit Message Buffer n Data Field 2	DATA8 to DATA11
+18h	TMBn.DF3	Transmit Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	TMBn.DF4	Transmit Message Buffer n Data Field 4	DATA16 to DATA19
+20h	TMBn.DF5	Transmit Message Buffer n Data Field 5	DATA20 to DATA23
+24h	TMBn.DF6	Transmit Message Buffer n Data Field 6	DATA24 to DATA27
+28h	TMBn.DF7	Transmit Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	TMBn.DF8	Transmit Message Buffer n Data Field 8	DATA32 to DATA35
+30h	TMBn.DF9	Transmit Message Buffer n Data Field 9	DATA36 to DATA39
+34h	TMBn.DF10	Transmit Message Buffer n Data Field 10	DATA40 to DATA43
+38h	TMBn.DF11	Transmit Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	TMBn.DF12	Transmit Message Buffer n Data Field 12	DATA48 to DATA51
+40h	TMBn.DF13	Transmit Message Buffer n Data Field 13	DATA52 to DATA55
+44h	TMBn.DF14	Transmit Message Buffer n Data Field 14	DATA56 to DATA59
+48h	TMBn.DF15	Transmit Message Buffer n Data Field 15	DATA60 to DATA63

46.2.61.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 3)

Address(es): CANFD.TMB0.HF0 8747 0604h, CANFD.TMB1.HF0 8747 0650h, CANFD.TMB2.HF0 8747 069Ch, CANFD.TMB3.HF0 8747 06E8h, CANFD_NS.TMB0.HF0 0747 0604h, CANFD_NS.TMB1.HF0 0747 0650h, CANFD_NS.TMB2.HF0 0747 069Ch, CANFD_NS.TMB3.HF0 0747 06E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	0: Entry is not stored in transmission history after successful transmission. 1: Entry is stored in transmission history after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

This register is used to store the ID field, IDE bit, and RTR bit of the message to be transmitted, and to specify whether to store it in the transmission history.
Do not write to this register in CH_SLEEP mode.

ID[28:0] Bits (Identifier)

The ID[28:0] bits are used to set the value of the standard or extended identifier field of the message stored in the transmit message buffer.
For alignment of these bits in base and extended format, refer to section 46.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message stored in the transmit message buffer.

RTR Bit (Remote Transmission Request)

The RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When transmitting a CAN FD frame (TMBn.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

The IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

46.2.61.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 3)

Address(es): CANFD.TMB0.HF1 8747 0608h, CANFD.TMB1.HF1 8747 0654h, CANFD.TMB2.HF1 8747 06A0h, CANFD.TMB3.HF1 8747 06ECh, CANFD_NS.TMB0.HF1 0747 0608h, CANFD_NS.TMB1.HF1 0747 0654h, CANFD_NS.TMB2.HF1 0747 06A0h, CANFD_NS.TMB3.HF1 0747 06ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes to be transmitted in a CAN frame.	R/W

This register is used to store the data length code (DLC) fields of the message to be transmitted.
Do not write to this register in CH_SLEEP mode.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted when the corresponding TMBn.HF0.RTR bit is set to 0.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

46.2.61.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 3)

Address(es): CANFD.TMB0.HF2 8747 060Ch, CANFD.TMB1.HF2 8747 0658h, CANFD.TMB2.HF2 8747 06A4h, CANFD.TMB3.HF2 8747 06F0h, CANFD_NS.TMB0.HF2 0747 060Ch, CANFD_NS.TMB1.HF2 0747 0658h, CANFD_NS.TMB2.HF2 0747 06A4h, CANFD_NS.TMB3.HF2 0747 06F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame to transmit by error active node 1: CAN FD frame to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame to transmit with no bit rate switch 1: CAN FD frame to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame to transmit 1: CAN FD frame to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that sets the pointer to be stored in the Transmission History	R/W

This register is used to store the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history.
Do not write to this register in CH_SLEEP mode.

ESI Bit (Error State Indicator)

The ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted.
If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

The BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

The FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

46.2.61.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 3; p = 0 to 15)

Address(es): CANFD.TMB0.DF0 8747 0610h to CANFD.TMB0.DF15 8747 064Ch,
CANFD.TMB1.DF0 8747 065Ch to CANFD.TMB1.DF15 8747 0698h,
CANFD.TMB2.DF0 8747 06A8h to CANFD.TMB2.DF15 8747 06E4h,
CANFD.TMB3.DF0 8747 06F4h to CANFD.TMB3.DF15 8747 0730h,
CANFD_NS.TMB0.DF0 0747 0610h to CANFD_NS.TMB0.DF15 0747 064Ch,
CANFD_NS.TMB1.DF0 0747 065Ch to CANFD_NS.TMB1.DF15 0747 0698h,
CANFD_NS.TMB2.DF0 0747 06A8h to CANFD_NS.TMB2.DF15 0747 06E4h,
CANFD_NS.TMB3.DF0 0747 06F4h to CANFD_NS.TMB3.DF15 0747 0730h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.
Do not write to these registers in CH_SLEEP mode.

46.2.61.25 Transmit Message Buffer n Data k (TMBn.DATAk) (n = 0 to 3; k = 0 to 63)

Address(es): CANFD.TMB0.DATA0 8747 0610h to CANFD.TMB0.DATA63 8747 064Fh,
CANFD.TMB1.DATA0 8747 065Ch to CANFD.TMB1.DATA63 8747 069Bh,
CANFD.TMB2.DATA0 8747 06A8h to CANFD.TMB2.DATA63 8747 06E7h,
CANFD.TMB3.DATA0 8747 06F4h to CANFD.TMB3.DATA63 8747 0733h,
CANFD_NS.TMB0.DATA0 0747 0610h to CANFD_NS.TMB0.DATA63 0747 064Fh,
CANFD_NS.TMB1.DATA0 0747 065Ch to CANFD_NS.TMB1.DATA63 0747 069Bh,
CANFD_NS.TMB2.DATA0 0747 06A8h to CANFD_NS.TMB2.DATA63 0747 06E7h,
CANFD_NS.TMB3.DATA0 0747 06F4h to CANFD_NS.TMB3.DATA63 0747 0733h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read/write registers for storing the data bytes of the message to be transmitted.
Do not write to these registers in CH_SLEEP mode.

46.2.62 ECC Control/Status Register (ECCSR)

Address(es): CANFD.ECCSR 8746 F200h, CANFD_NS.ECCSR 0746 F200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EC2EAS	EC1EAS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECEDWC[1:0]		—	—	ECOVF	EC2EC	EC1EC	—	—	ECEDE	EC1ECD	EC2EIE	EC1EIE	EC2EF	EC1EF	ECEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECEF	ECC Error Flag	0: There is no ECC error in the last RAM data read. 1: There is an ECC error in the last RAM data read.	R
b1	EC1EF	1-Bit ECC Error Detection Flag	0: 1-bit ECC error is not detected. 1: 1-bit ECC error is detected.	R
b2	EC2EF	2-Bit ECC Error Detection Flag	0: 2-bit ECC error is not detected. 1: 2-bit ECC error is detected.	R
b3	EC1EIE	1-Bit ECC Error Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b4	EC2EIE	2-Bit ECC Error Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b5	EC1ECD	1-Bit ECC Error Correction Disable	0: Perform error correction when 1-bit ECC error is detected. 1: Does not perform error correction when 1-bit ECC error is detected.	R/W
b6	ECEDE	ECC Error Detection Enable	0: ECC error detection is disabled. 1: ECC error detection is enabled.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	EC1EC	EC1EF Flag Clear	Writing 1 to this bit clears the EC1EF flag. Writing 0 to this bit is ignored. This bit is read as 0.	R/W
b10	EC2EC	EC2EF Flag Clear	Writing 1 to this bit clears the EC2EF flag. Writing 0 to this bit is ignored. This bit is read as 0.	R/W
b11	ECOVF	ECC Overflow Detection Flag	0: The ECEAR register overflow has not occurred. 1: The ECEAR register overflow has occurred.	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECEDWC[1:0]	ECEDE Bit Write Control	Enables or disables write access to the ECEDE bit.	R/W
b16	EC1EAS	1-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 1-bit ECC error occurred is stored in the ECEAR register.	R
b17	EC2EAS	2-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 2-bit ECC error occurred is stored in the ECEAR register.	R
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ECEF Flag (ECC Error Flag)

The ECEF flag indicates whether there is an ECC error in the last RAM data read. This flag is updated every time the RAM is read.

If the ECEDE bit is set to 1 without initializing the RAM, the value of this flag has no meaning.

[Setting condition]

- If there is an ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Under the condition that there is no ECC error in read RAM data
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

EC1EF Flag (1-Bit ECC Error Detection Flag)

The EC1EF flag indicates that a 1-bit ECC error is detected in the data read from RAM.
When this flag is set to 1 while the interrupt is enabled, an ECC error interrupt (ECCEI) is generated.
If a 1-bit ECC error is detected again while this flag is 1, no further interrupt is generated.

[Setting condition]

- When a 1-bit ECC error is detected in the data read from RAM while the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If a 1-bit ECC error is detected at the same time that the EC1EC bit is set to 1, the EC1EF flag becomes 0.

EC2EF Flag (2-Bit ECC Error Detection Flag)

The EC2EF flag indicates that a 2-bit ECC error is detected in the data read from RAM.
When this flag is set to 1 while the interrupt is enabled, an ECC error interrupt (ECCEI) is generated.
If a 2-bit ECC error is detected again while this flag is 1, no further interrupt is generated.

[Setting condition]

- When a 2-bit ECC error is detected in the data read from RAM while the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If a 2-bit ECC error is detected at the same time that the EC2EC bit is set to 1, the EC2EF flag becomes 0.

EC1EIE Bit (1-Bit ECC Error Detection Interrupt Enable)

The EC1EIE bit enables or disables the interrupt generated by 1-bit ECC error detection.
When the EC1EF flag is set to 1 while this bit is 1, an ECC error interrupt (ECCEI) is generated.

EC2EIE Bit (2-Bit ECC Error Detection Interrupt Enable)

The EC2EIE bit enables or disables the interrupt generated by 2-bit ECC error detection.
When the EC2EF flag is set to 1 while this bit is 1, an ECC error interrupt (ECCEI) is generated.

EC1ECD Bit (1-Bit ECC Error Correction Disable)

The EC1ECD bit sets whether 1-bit ECC error correction is performed when the ECEDE bit is set to 1 (ECC error detection is enabled). When this bit is set to 1, the RAM output data is not corrected even if a 1-bit ECC error is detected.

ECEDE Bit (ECC Error Detection Enable)

Setting this bit to 1 enables the ECC error detection.
Writing to this bit is valid only when the ECEDWC[1:0] bits are set to 01b.

EC1EC Bit (EC1EF Flag Clear)

The EC1EC bit is used to clear the EC1EF flag.
The EC1EF flag is cleared by writing 1 to this bit while the EC1EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If a 1-bit ECC error is detected at the same time that the EC1EC bit is set to 1, the EC1EF flag becomes 0.

EC2EC Bit (EC2EF Flag Clear)

The EC2EC bit is used to clear the EC2EF flag.
The EC2EF flag is cleared by writing 1 to this bit while the EC2EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If a 2-bit ECC error is detected at the same time that the EC2EC bit is set to 1, the EC2EF flag becomes 0.

ECOVF Flag (ECC Overflow Detection Flag)

If a new ECC error is detected and the address is overwritten when an address is already stored in the ECEAR register, the ECOVF flag is set to 1 and an ECC error interrupt (ECCEI) is generated.
If a new ECC error is detected while this flag is 1, an ECC error interrupt is generated again.

[Setting condition]

- When new error address is captured under the condition that error address is already captured in the ECEAR register.

[Clearing condition]

- When the EC1EC bit is set to 1
- When the EC2EC bit is set to 1
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the ECOVF flag is cleared by the EC1EC or EC2EC bit and the setting condition of the ECOVF flag occurs at the same time, the ECOVF flag becomes 0.

ECEDWC[1:0] Bits (ECEDE Bit Write Control)

The ECEDWC[1:0] bits are used to enable or disable write access to the ECEDE bit. The read value is always 00b.
When the value of these bits is 01b, it is possible to have write access to the ECEDE bit. If these bits are other than 01b, write access to the ECEDE bit is ignored and the value does not change.

EC1EAS Flag (1-Bit ECC Error Detected Address Stored Flag)

The EC1EAS flag indicates that the address where the 1-bit ECC error occurred is stored in the ECEAR register when the ECEDE bit is set to 1 (ECC error detection is enabled).
When 1-bit ECC error is detected while the 2-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is not updated and this flag is not updated.

[Setting condition]

- When a 1-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC1EAS flag is cleared by the EC1EC bit and the setting factor of the EC1EAS flag occurs at the same time, the EC1EAS flag becomes 0.

EC2EAS Flag (2-Bit ECC Error Detected Address Stored Flag)

The EC2EAS flag indicates that the address where the 2-bit ECC error occurred is stored in the ECEAR register when

the ECEDE bit is set to 1 (ECC error detection is enabled).
When 2-bit ECC error is detected while the 1-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is updated and this flag becomes 1.

[Setting condition]

- When a 2-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC2EAS flag is cleared by the EC2EC bit and the setting factor of the EC2EAS flag occurs at the same time, the EC2EAS flag becomes 0.

46.2.63 ECC Test Mode Register (ECTMR)

Address(es): CANFD.ECTMR 8746 F204h, CANFD_NS.ECTMR 0746 F204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECTMWC[1:0]		—	—	—	—	—	—	ECTME	—	—	—	—	—	ECDIS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ECDIS	ECC Decoder Input Select	0: Input RAM output data to data input of decode circuit 1: Select the ECTDR register to data input of decode circuit	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ECTME	ECC Test Mode Enable	0: Access to the ECDIS bit and the ECTDR register is disabled 1: Access to the ECDIS bit and the ECTDR register is enabled	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECTMWC[1:0]	ECTME Bit Write Control	Enable or disable write access to the ECTME bit.	R/W

ECDIS Bit (ECC Decoder Input Select)

The ECDIS bit selects which of the data value read from RAM and the value of the ECTDR register is used as input data to the ECC decoder.

The write access to this bit is enabled when the ECTME bit is set to 1. It is also possible to set them at the same time.

This bit is cleared by setting the ECTME bit to 0.

ECTME Bit (ECC Test Mode Enable)

The ECTME bit is used to enable or disable the access to the ECDIS bit and the ECTDR register.

This bit can be written only when the ECTMWC[1:0] bits is set to 10b.

ECTMWC[1:0] Bits (ECTME Bit Write Control)

The ECTMWC[1:0] bits are used to enable or disable the write access to the ECTME bit. The read value is always 00b.

When the value of these bits is 10b, it is possible to have write access to the ECTME bit. If these bits are other than 10b, write access to the ECTME bit is ignored and the value does not change.

46.2.64 ECC Decoder Test Data Register (ECTDR)

Address(es): CANFD.ECTDR 8746 F20Ch, CANFD_NS.ECTDR 0746 F20Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the data for testing the ECC decode.
When the ECTMR.ECTME bit is set to 1, this register can be read and written.
When the ECTMR.ECTME bit is set to 0, the value of this register becomes 00000000h.
When the ECTMR.ECDIS bit is set to 1, the value set in this register is used as the input data of the ECC decoder instead of the data read from RAM.

46.2.65 ECC Error Address Register (ECEAR)

Address(es): CANFD.ECEAR 8746 F210h, CANFD_NS.ECEAR 0746 F210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register holds the address where the ECC error occurred.
If an ECC error is detected when the ECCSR.ECEDE bit is 1 (ECC error detection is enabled), b11 to b2 of the RAM address are stored in b9 to b0 of this register.
If the same error occurred again, this register is not updated.
If a 2-bit ECC error is detected while the address where the 1-bit ECC error occurred is already stored, the ECEAR register is overwritten with the new address and the ECCSR.EC2EAS flag is set to 1.
If a 1-bit ECC error is detected while the address where the 2-bit ECC error occurred is already stored, the ECEAR register is not updated and the ECCSR.EC1EAS flag is not updated.

46.3 Operating Mode

The operating modes of the CANFD module can be classified into two groups:

- Global modes
- Channel modes

46.3.1 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes.
The Global modes of the CANFD module are:

- GL_SLEEP
- GL_RESET
- GL_HALT
- GL_OPERATION

Figure 46.2 shows the possible transitions between the Global modes.

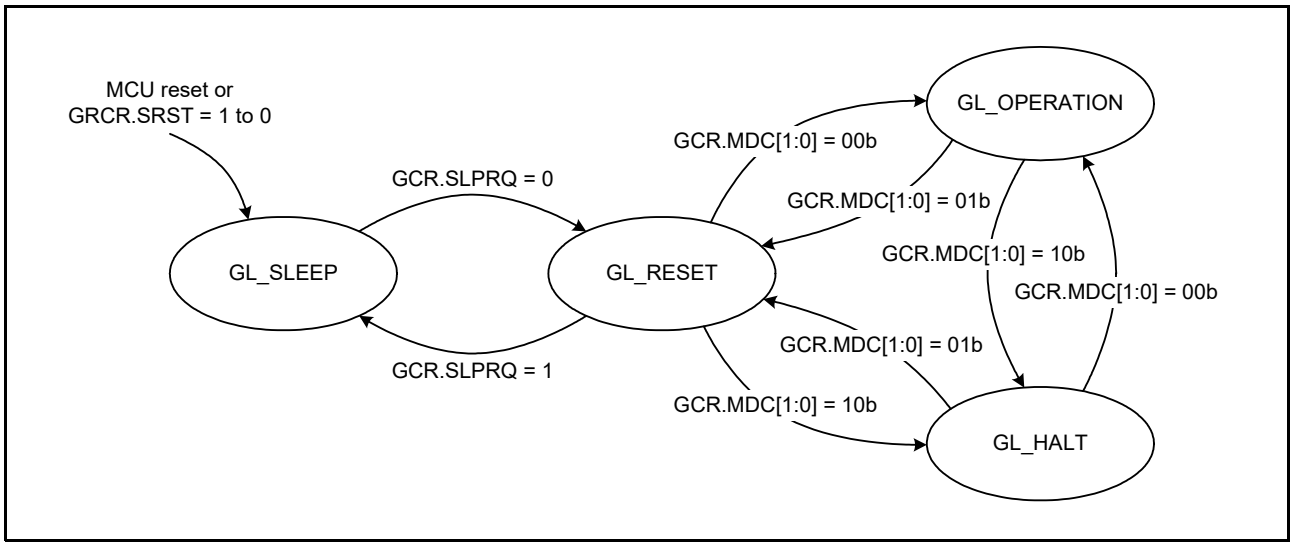


Figure 46.2 Transition between CANFD Global Modes

Change in the Global mode can affect the Channel mode. For details, refer to section 46.3.3, Global Mode and Channel Mode Transition Interactions.

46.3.1.1 GL_SLEEP Mode

The CANFD module automatically enters GL_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CANFD module also enters the GL_SLEEP mode when the GCR.SLPRQ bit is set to 1 while it is in GL_RESET mode. The SLPRQ bit cannot be set to 1 in GL_HALT or GL_OPERATION mode.
Setting the GCR.SLPRQ bit to 1 sets the CHCR.SLPRQ bit to 1 and forces CAN channel into the CH_SLEEP mode.
GL_SLEEP mode is used for power saving purpose. When CANFD module is in GL_SLEEP mode, only the clock for writing to the SLPRQ bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.
Read access from all registers is still possible and all register values are preserved.
After setting the GCR.SLPRQ bit to 1, it is necessary to confirm the GSR.SLPST flag that the GL_SLEEP status has been updated, indicating successful transition to GL_SLEEP mode before the GCR.SLPRQ bit can be cleared again.

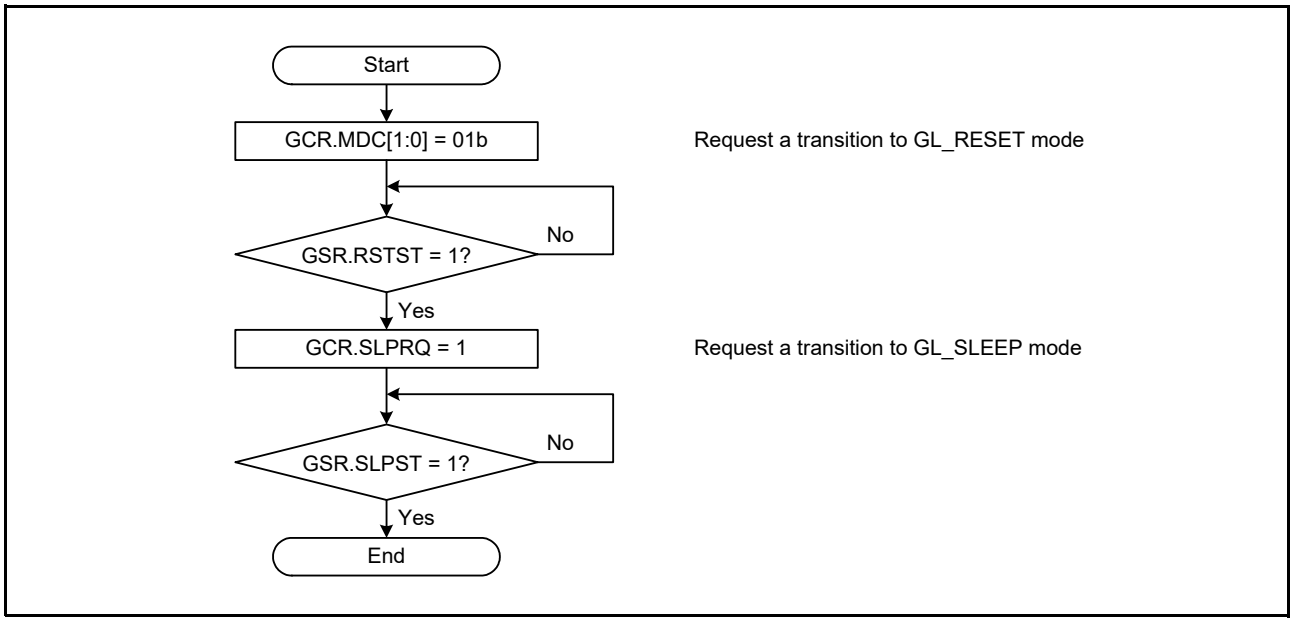


Figure 46.3 Procedure for Entering GL_SLEEP Mode

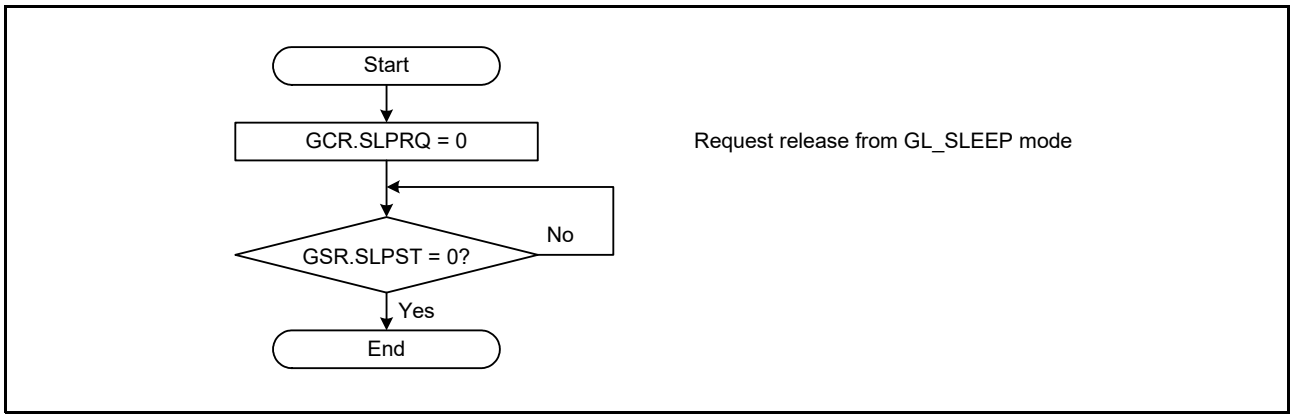


Figure 46.4 Procedure for Exiting GL_SLEEP Mode

46.3.1.2 GL_RESET Mode

The CANFD module enters GL_RESET mode in the following ways:

- The GCR.MDC[1:0] bits are set to 01b while the CANFD module is in GL_HALT or GL_OPERATION mode.
- The GCR.SLPRQ bit is set to 0 while the CANFD module is in GL_SLEEP mode.

In GL_RESET mode, all CANFD module functions are suspended and all status and flag registers are initialized. Additionally all FIFOs and all transmit queues are disabled and transmission control bits are cleared.
In this mode, configuration registers other than the GTMCR register and interrupt enable registers are not initialized, so the CANFD module can be configured.
Setting the Global mode to GL_RESET by setting the GCR.MDC[1:0] bits to 01b sets the CHCR.MDC[1:0] bits to 01b and forces the channel into the CH_RESET mode.
If the channel is already in CH_RESET mode or CH_SLEEP mode, the channel mode remains unchanged.
After setting the GCR.MDC[1:0] bits to 01b (GL_RESET mode), it is necessary to confirm that the GSR.RSTST flag has been updated, indicating successful transition to GL_RESET mode before the GCR.MDC[1:0] bits can be changed again.

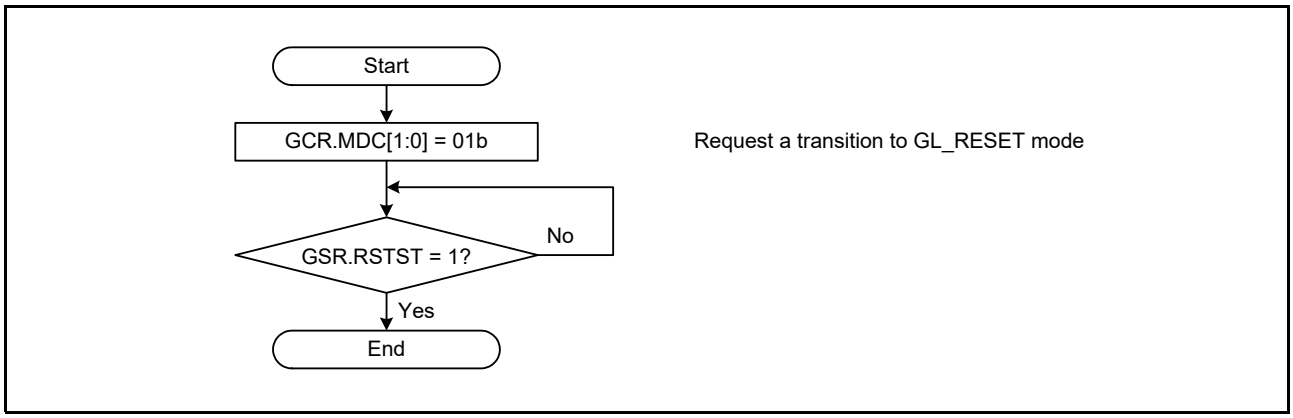


Figure 46.5 Procedure for Entering GL_RESET Mode

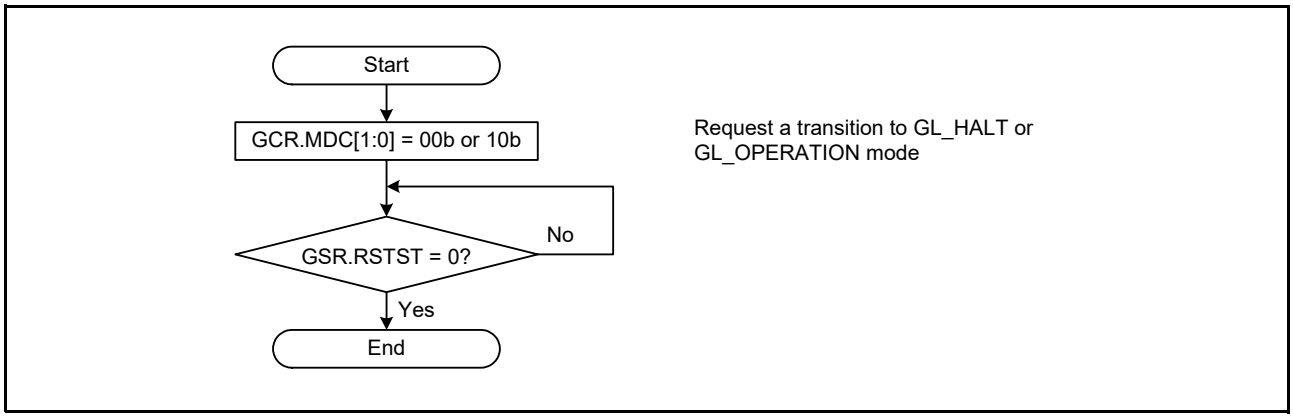


Figure 46.6 Procedure for Exiting GL_RESET Mode

46.3.1.3 GL_HALT Mode

The CANFD module enters GL_HALT mode in the following ways:

- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_RESET mode:
  - The CAN channel is in either CH_RESET or CH_SLEEP mode and remains in that mode.
- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_OPERATION mode:
  - If the CAN channel is in CH_RESET, CH_HALT, or CH_SLEEP mode, it remains in that mode.
  - If the CAN channel is in CH_OPERATION mode, it transits to CH_HALT mode.
  - the GSR.HLTST flag is set to 1 when the CAN channel exits CH_OPERATION mode.

If a transmission or reception is in progress for a CAN channel, the channel enters CH_HALT mode after waiting for completion of the communication.
Similarly, if a CAN channel is in the bus-off state, the channel may not enter CH_HALT mode until the bus-off recovery sequence is completed, depending on the channel configuration.
In GL_HALT mode, all communications are suspended and the status and flag registers do not change (only when a channel is in the bus-off state, its CHSR.REC[7:0] and TEC[7:0] bits are set to 00h).
Additionally, the GTMCR register and GTMER register are not initialized in this mode. The GL_HALT mode is used to configure Global Test Modes.
Setting the Global mode to GL_HALT by setting the GCR.MDC[1:0] bits to 10b sets the CHCR.MDC[1:0] bits to 10b and forces the channel into the CH_HALT mode.
If the channel is already in CH_RESET, CH_HALT, or CH_SLEEP mode, the channel mode remains unchanged.
Therefore, the GL_HALT mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channel).
After setting the GCR.MDC[1:0] bits to 10b (GL_HALT mode), it is necessary to confirm that the GSR.HLTST flag has been updated to indicate a successful transition to GL_HALT mode. Do not set any other registers until confirming the GSR.HLTST flag is set.

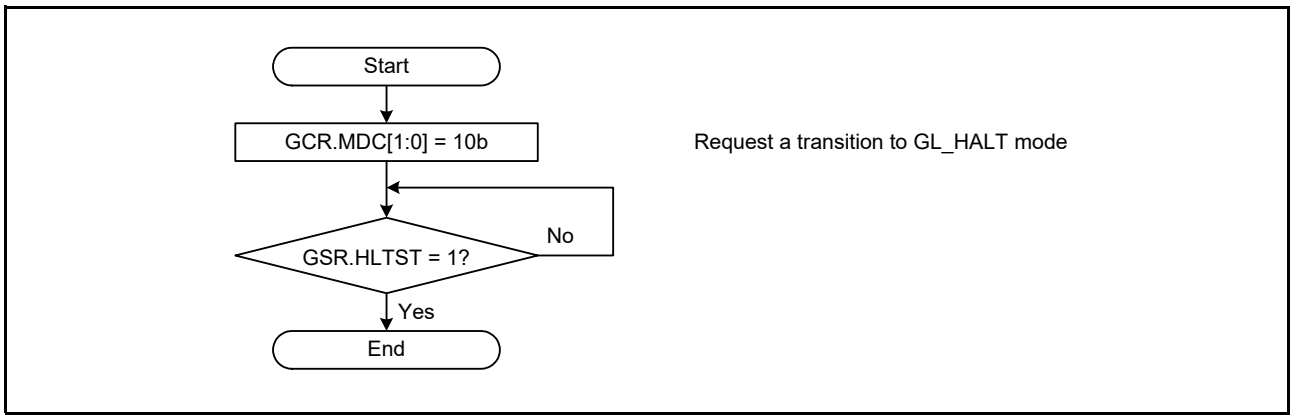


Figure 46.7 Procedure for Entering GL_HALT Mode

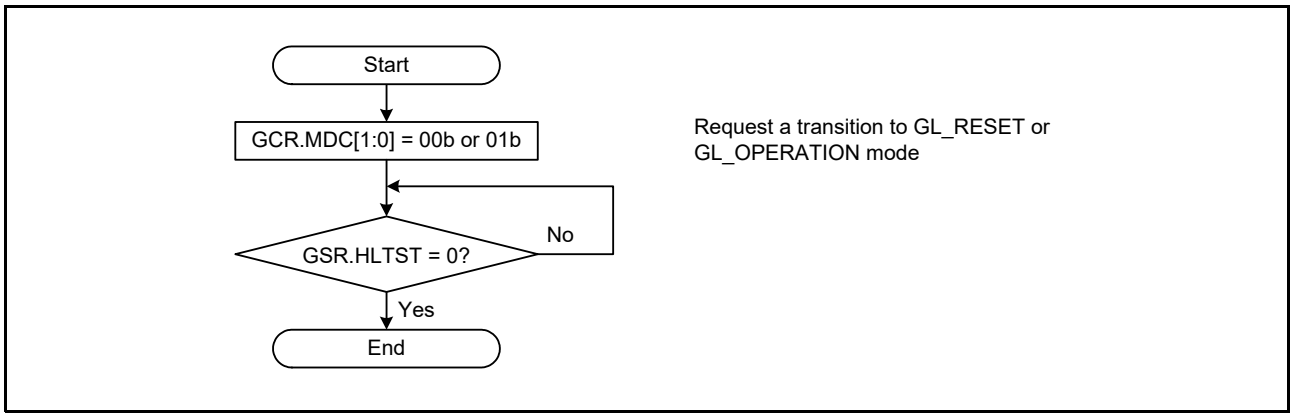


Figure 46.8 Procedure for Exiting GL_HALT Mode

46.3.1.4 GL_OPERATION Mode

The CANFD module enters this mode when the GCR.MDC[1:0] bits are set to 00b.
The CAN channel can be set to CH_OPERATION mode and start CAN communication only when the CANFD module is in GL_OPERATION mode.
After setting the GCR.MDC[1:0] bits to 00b (GL_OPERATION mode), it is necessary to confirm that the GSR.RSTST flag and the GSR.HLTST flag have been set to 0 to indicate a successful transition to GL_OPERATION mode before the GCR.MDC[1:0] bits can be modified again.

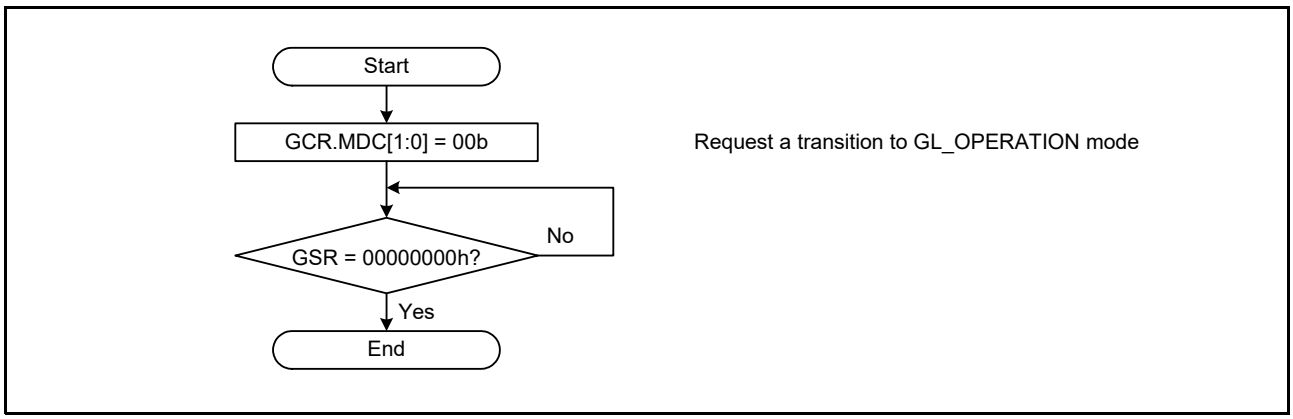


Figure 46.9 Procedure for Entering GL_OPERATION Mode

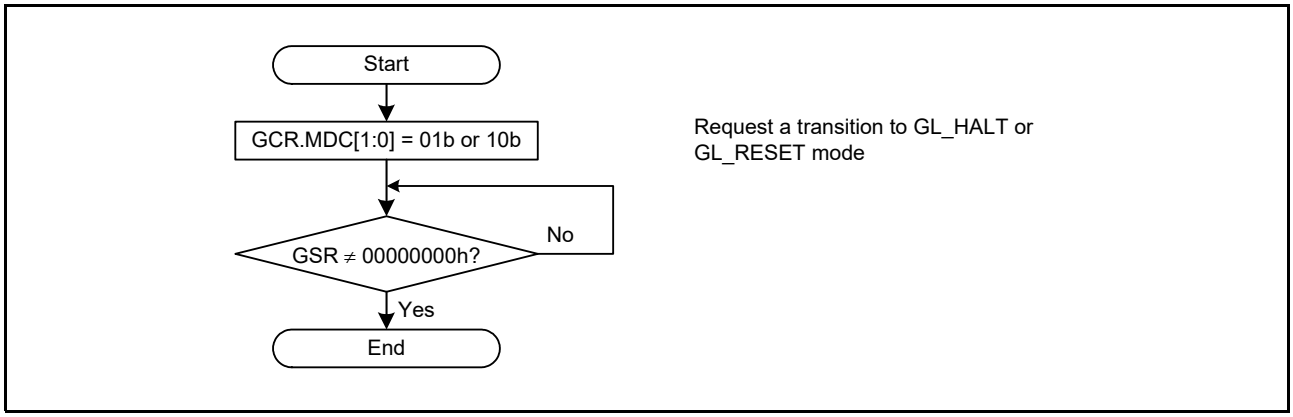


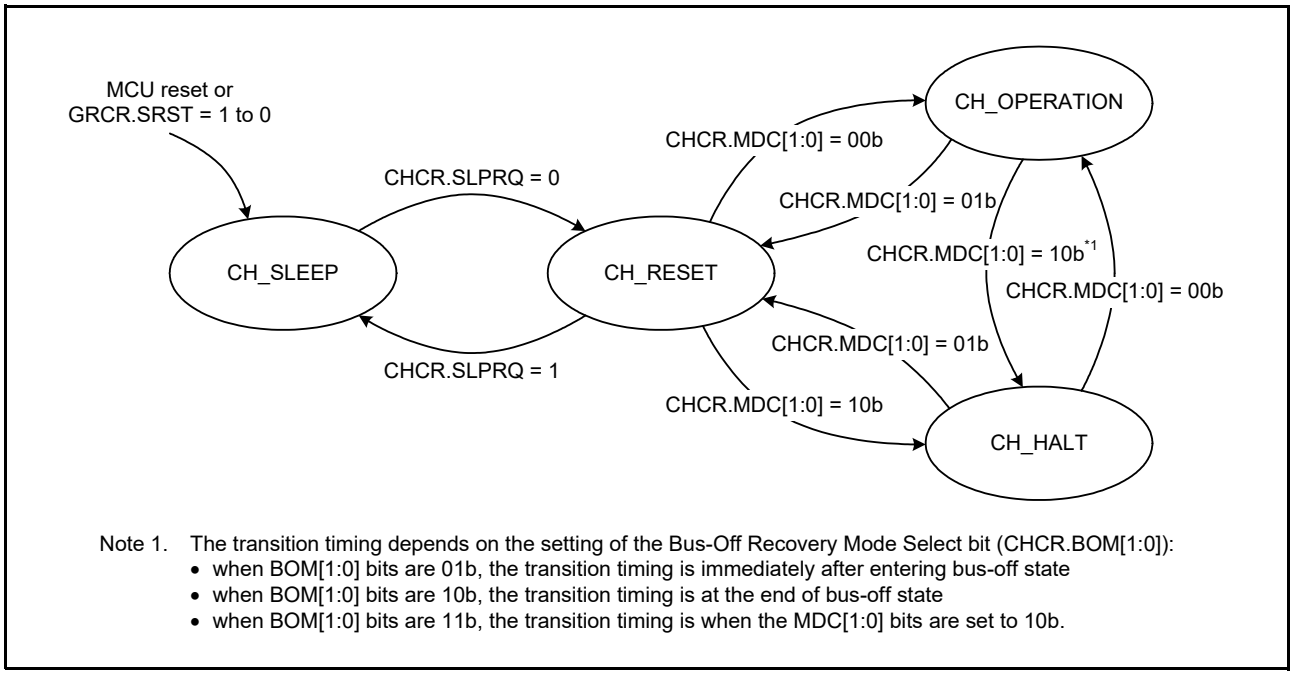
Figure 46.10 Procedure for Exiting GL_OPERATION Mode

46.3.2 Channel Modes

The Channel modes of the CANFD module are:

- CH_RESET
- CH_HALT
- CH_OPERATION
- CH_SLEEP

Figure 46.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on the setting of the Bus-Off Recovery Mode Select bit (CHCR.BOM[1:0]):
- when BOM[1:0] bits are 01b, the transition timing is immediately after entering bus-off state
- when BOM[1:0] bits are 10b, the transition timing is at the end of bus-off state
- when BOM[1:0] bits are 11b, the transition timing is when the MDC[1:0] bits are set to 10b.

Figure 46.11 Transition between CAN Channel Modes

46.3.2.1 CH_SLEEP Mode

The CAN channel of the CANFD module automatically enters CH_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CAN channel also enters CH_SLEEP mode when the CHCR.SLPRQ bit is set to 1 while the CAN channel is in CH_RESET mode. The CHCR.SLPRQ bit cannot be set to 1 in CH_HALT or CH_OPERATION mode.
Entering CH_SLEEP mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.
After setting the CHCR.SLPRQ bit to 1 and before setting it to 0 again, it is necessary to use the CHSR.SLPST flag to confirm that transition to CH_SLEEP mode was successful.
During CH_SLEEP mode, do not write to channel related registers. Read operation is still possible.

46.3.2.2 CH_RESET Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 01b while the CAN channel is in CH_HALT or CH_OPERATION mode
- The CHCR.SLPRQ bit is set to 0 while the CAN channel is in CH_SLEEP mode
- The GCR.MDC[1:0] bits are set to 01b while the CAN channel is not in CH_SLEEP or CH_RESET mode.

In CH_RESET mode, all CAN channel status and flags are initialized.
In addition, all transmission-related control bits of the channel are cleared, and the transmit queue of the channel is also disabled.
In this mode, the configuration registers except for the bits related to the channel test mode are not initialized, so the CAN channel can be configured for communication.
After setting the CHCR.MDC[1:0] bits to 01b (CH_RESET mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag to confirm that transition to CH_RESET mode was successful.
Refer to Table 46.11 for the behavior of transitioning to CH_RESET mode while CAN communication is in progress.

46.3.2.3 CH_HALT Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_RESET or CH_OPERATION mode
- The GCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_OPERATION mode.

In CH_HALT mode, all CAN communication of the channel is suspended, but all statuses and flags remain unchanged during CH_HALT mode (except for the bus-off state where the CHSR.REC[7:0] and TEC[7:0] bits for the channel are set to 00h).
In addition, in this mode, the bits related to the channel test mode are not initialized. Use CH_HALT mode to configure the channel test mode.
After setting the CHCR.MDC[1:0] bits to 10b (CH_HALT mode) and before modifying these bits again, it is necessary to use the CHSR.HLTST flag to confirm that transition to CH_HALT mode was successful.
Refer to Table 46.11 for the transition behavior to CH_HALT mode while CAN communication is in progress.

Table 46.11 Transition Behavior in CH_RESET Mode and CH_HALT Mode

Mode	State		
	Receiver	Transmitter	Bus-Off
CH_RESET mode (CHCR.MDC[1:0] = 01b)	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the bus-off recovery.
CH_HALT mode (CHCR.MDC[1:0] = 10b)	CAN channel transits to CH_HALT mode at the end of the ongoing reception or when an error occurs.*2	CAN channel transits to CH_HALT mode after completion of the ongoing transmission.	When the CHCR.BOM[1:0] bits are 00b, a CH_HALT mode request is accepted only after the completion of the full bus-off recovery sequence. When the CHCR.BOM[1:0] bits are 10b, the CAN channel transits automatically to CH_HALT mode after waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 01b, the CAN channel transits automatically to CH_HALT mode without waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 11b, the CAN channel transits to CH_HALT mode as soon as CH_HALT mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to CH_RESET mode is required only at the end of an ongoing communication, then CH_HALT mode can be requested first to prevent interruption of CAN communication by direct transition to CH_RESET mode. After the CAN channel enters CH_HALT mode, the CH_RESET mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to CH_RESET mode.

46.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State)

The CH_OPERATION mode is activated by setting the CHCR.MDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering the CH_OPERATION mode, the CHSR.CRDY flag is set to 1 and the CAN channel:

- Enables the functions of the communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters.

At this point, the CAN channel can start transmission and reception of messages.
Within the CH_OPERATION mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (refer to Figure 46.12):

- Idle mode: The CAN channel is neither receiving nor transmitting
- Receive mode: The channel is receiving a message transmitted by another CAN node
- Transmit mode: The channel is transmitting a message
  (The channel may receive its own message simultaneously when Self Test mode is enabled.)
- Bus-off mode: The CAN channel is cut-off from CAN bus communication

After setting the CHCR.MDC[1:0] bits to 00b (CH_OPERATION mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag and the CHSR.HLTST flag to confirm that transition to CH_OPERATION mode was successful.

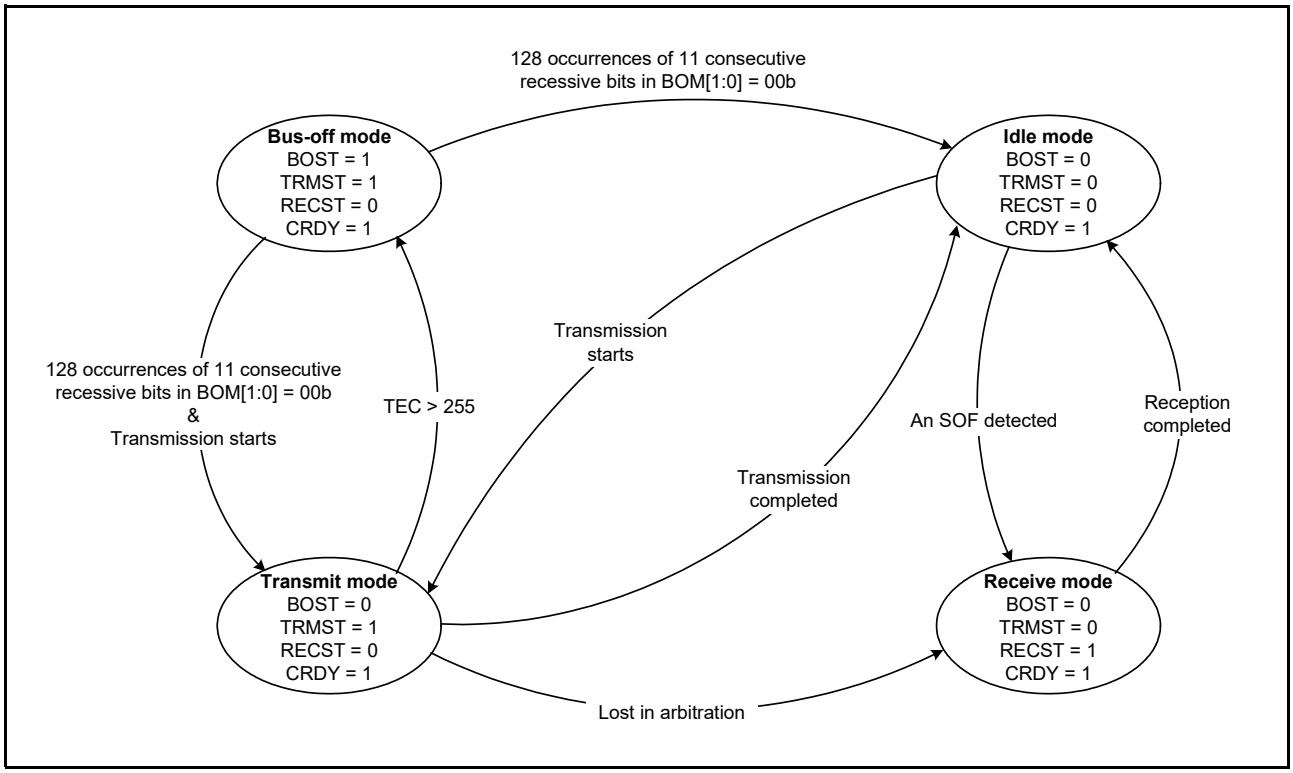


Figure 46.12 Sub-modes of CH_OPERATION Mode (only when CHCR.BOM[1:0] = 00b)

46.3.2.5 CH_OPERATION Mode (in Bus-Off State)

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CH_OPERATION mode from the bus-off state:

- CHCR.BOM[1:0] = 00b:
  Bus-off recovery is compliant to ISO 11898-1. The CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h, and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 01b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1 in this case.
- CHCR.BOM[1:0] = 10b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter CH_HALT mode when still in bus-off state if a request is made to enter CH_HALT mode. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1.
  Without setting CHCR.MDC[1:0] = 10b and when 11 consecutive recessive bits are detected 128 times, transition conditions become the same as CHCR.BOM[1:0] = 00b.

If the recovery from bus-off occurs normally in this mode (after waiting for 128 occurrences of 11 consecutive recessive bits), and no CH_HALT request has been generated during this period, then the CHESR.BORDF flag is set to 1.
When software writes to the CHCR.MDC[1:0] bits at the same time as the CAN channel enters CH_HALT mode (at the start of bus-off when CHCR.BOM[1:0] = 01b, or at the end of bus-off when CHCR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CHCR.MDC[1:0] bits to CH_HALT mode request is performed when the CHCR.MDC[1:0] bits value is previously 00b (CH_OPERATION mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting the CHCR.RTBO bit to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The CHESR.BORDF flag is not set to 1 in this case, and the CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h.
Before setting the CHCR.RTBO bit to 1, all pending transmissions from the transmit message buffers, transmit queues and/or common FIFO in transmit mode should be disabled.
The disable of the pending transmit message buffer, transmit queue or common FIFO must be confirmed by the corresponding acknowledge flags (TMSRn.TXRF[1:0] flags, TQSR0.EMPTY flag, and CFSR0.EMPTY flag).
The CHCR.RTBO bit should be used for bus-off recovery only when the CHCR.BOM[1:0] bits are set to 00b.
Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.
Table 46.12 lists the behavior of the bus-off entry detect flag (CHESR.BOEDF) and the bus-off recovery detect flag (CHESR.BORDF) according to the CHCR.BOM[1:0] bit settings.

Table 46.12 Behavior of Bus-off Entry and Recovery Flags

CHCR.BOM[1:0] Bits	CHESR.BOEDF Flag	CHESR.BORDF Flag
00b	Becomes 1 on entry to bus-off state.	Becomes 1 on exit from bus-off state.
00b Set the CHCR.RTBO bit to 1		Becomes 1 only if normal bus-off recovery occurs before setting the CHCR.RTBO bit to 1.
01b		Does not become 1.
10b		Becomes 1 on exit from bus-off state.
11b		Becomes 1 only if normal bus-off recovery occurs before requesting a transition to CH_HALT mode.

To make efficient software, it is not necessary to wait for the bus-off recovery sequence to end.
It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 46.13.

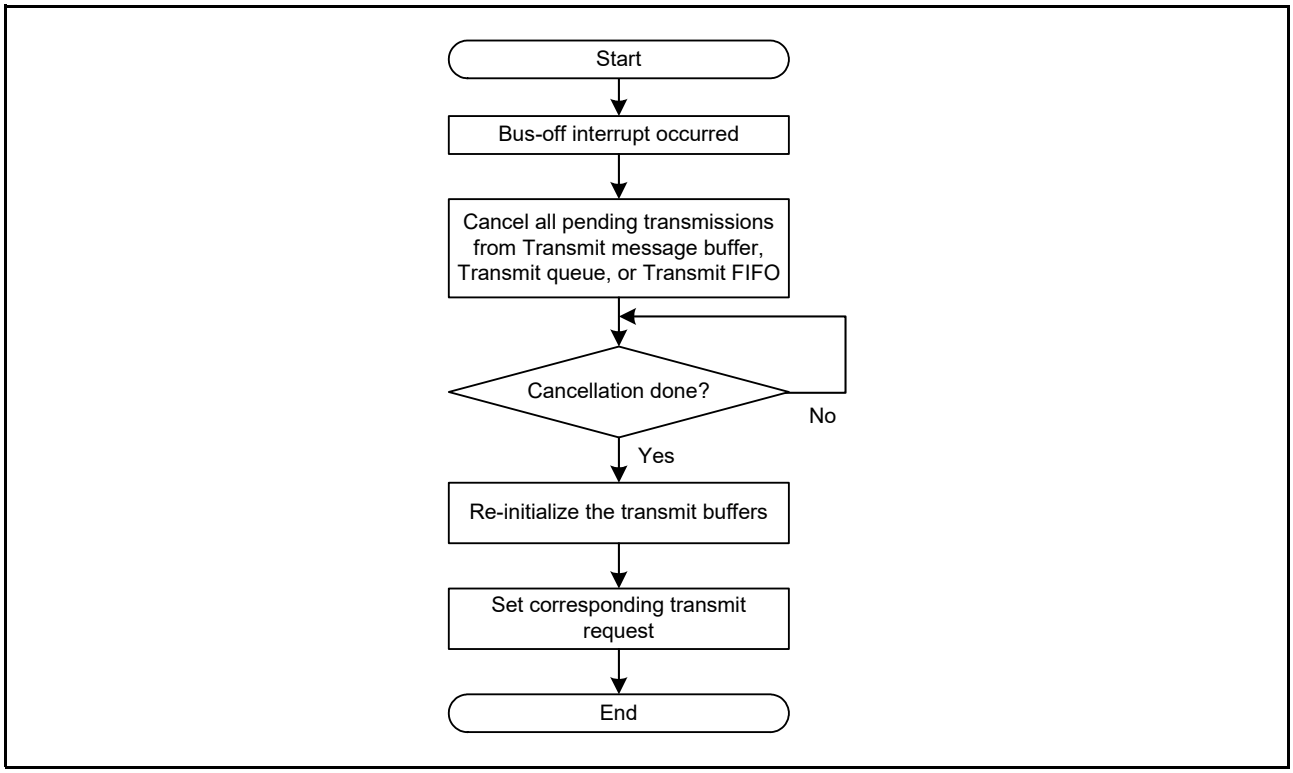


Figure 46.13 Transmission Re-Initialization During Bus-Off

46.3.3 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the CHCR.MDC[1:0] bits does not affect the GCR.MDC[1:0] bits.
- Changing the GCR.MDC[1:0] bits affects the channel mode control as described in Table 46.13.

Table 46.13 Interaction between Global Mode and Channel Mode Transition

Global Mode Change	Channel Mode before Changing Global Mode	Channel Mode after Changing Global Mode
GL_SLEEP → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
GL_RESET → GL_SLEEP	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_SLEEP
GL_RESET → GL_HALT	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
GL_RESET → GL_OPERATION	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
GL_HALT → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_RESET
GL_HALT → GL_OPERATION	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_HALT (no changes)
GL_OPERATION → GL_RESET	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_RESET
	CH_OPERATION	CH_RESET
GL_OPERATION → GL_HALT	CH_SLEEP	CH_SLEEP (no changes)
	CH_RESET	CH_RESET (no changes)
	CH_HALT	CH_HALT (no changes)
	CH_OPERATION	After communication ends, CH_HALT

46.3.3.1 Transition Time for Global Mode Change

Table 46.14 shows the transition times when changing the global mode.

Table 46.14 Maximum Transition Times for Global Mode Changes

From	To	Maximum Transition Time
GL_SLEEP	GL_RESET	3 × PCLKA*2
GL_RESET	GL_SLEEP	3 × PCLKA
GL_RESET	GL_HALT	10 × PCLKA
GL_RESET	GL_OPERATION	10 × PCLKA
GL_HALT	GL_RESET	2 bit times (1 Tq + 16 × PCLKA + 2 DLL clock cycles)
GL_HALT	GL_OPERATION	3 × PCLKA
GL_OPERATION	GL_RESET	2 bit times (1 Tq + 16 × PCLKA + 2 DLL clock cycles)
GL_OPERATION	GL_HALT	3 CAN frames (1 CAN frame + 3424 × PCLKA)*1

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.

Note 2. Exit GL_SLEEP mode only when the GSR.RAMST flag is cleared.

46.3.3.2 Transition Time for Channel Mode Change

Table 46.15 shows the transition times when changing the channel mode.

Table 46.15 Maximum Transition Times for Channel Mode Changes

From	To	Maximum Transition Time
CH_SLEEP	CH_RESET	3 × PCLKA
CH_RESET	CH_SLEEP	3 × PCLKA
CH_RESET	CH_HALT	3 bit times (1 CAN bit + 2 Tq + 8 × PCLKA + 2 DLL clock cycles)
CH_RESET	CH_OPERATION	4 bit times (2 CAN bits + 1 TSEG1 + 12 × PCLKA + 2 DLL clock cycles)
CH_HALT	CH_RESET	2 bit times (1 Tq + 10 × PCLKA + 2 DLL clock cycles)
CH_HALT	CH_OPERATION	4 bit times (< 4 CAN bits)*3
CH_OPERATION	CH_RESET	2 bit times (1 Tq + 10 × PCLKA + 2 DLL clock cycles)
CH_OPERATION	CH_HALT	2 CAN frames (1 CAN frame + 13 CAN bits)*1, *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CHCR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.

Note 3. In general, if the bit rate prescaler value (NBCR.BRP[9:0]) is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the Tq clock, and new BRP value is captured when the counter reaches the value 0.

46.4 Initialization of CANFD Module

Before starting CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data bit rate)
- Bit Rate setting (nominal and data bit rate)
- CANFD setting
- Acceptance Filter setting (configuration of Acceptance Filter List)
- Receive FIFO and Transmit FIFO setting
- CAN operating mode setting.

46.4.1 Initialization of CAN Clock, Bit Timing and Bit Rate

46.4.1.1 Bit Timing Conditions

The following lines describe the settings for each segment and the limitations that apply to those settings.

1. Each segment setting
   SS = Fixed to 1 Tq
   TSEG1 = 2 Tq to 256 Tq (NBCR), 2 Tq to 32 Tq (DBCR)
   TSEG2 = 2 Tq to 128 Tq (NBCR), 2 Tq to 16 Tq (DBCR)
   SJW = 1 Tq to 128 Tq (NBCR), 1 Tq to 16 Tq (DBCR)
   SS + TSEG1 + TSEG2 = 8 Tq to 385 Tq (NBCR), 5 Tq to 49 Tq (DBCR)
2. Limitations on TSEG1, TSEG2 and SJW
   NBCR register: TSEG1 > TSEG2 ≥ SJW
   DBCR register: TSEG1 ≥ TSEG2 ≥ SJW

Table 46.16 shows an example of how to set the bit timing to achieve the required sample point settings.

Table 46.16 Bit Timing Examples

1 Bit	Set Value (Tq)				Sample Point (%)
	SS	TSEG1	TSEG2	SJW	
5 Tq	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66
	1	16	7	1	70.83
50 Tq	1	39	10	4	80.00

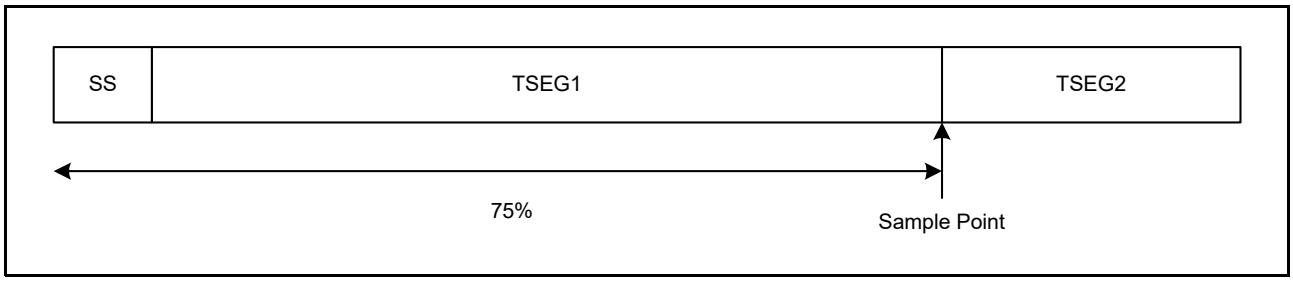


Figure 46.14 The Sample Point (in Case of 75%)

46.4.1.2 Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured using the NBCR and DBCR registers.

Figure 46.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (Tq), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the bit rate prescaler (nominal and data bit rate).

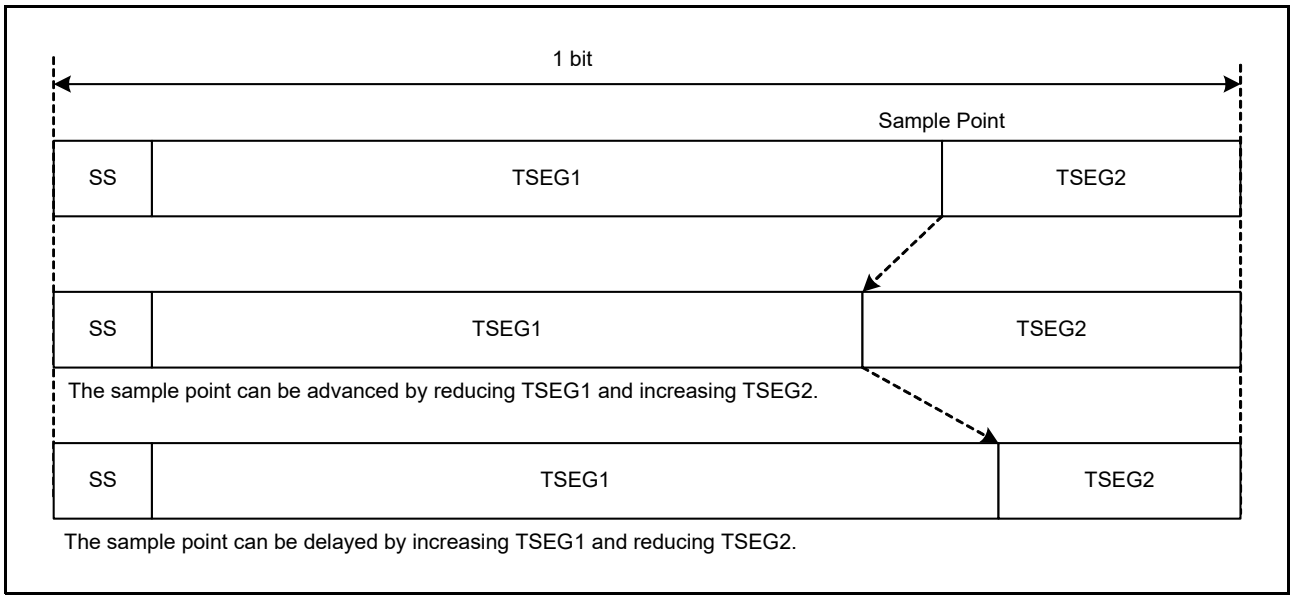


Figure 46.15 Segment Composition of A Bit and The Sample Point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. The interframe space consists of the intermission, suspend transmission, and bus idle fields; during the bus idle field, all nodes may begin transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While transmitting or receiving a message, communication frames between some nodes may get out of sync due to a drift

in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.

4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 46.15 shows an example of a typical sample point.

46.4.1.3 Bit Rate

The CAN communication clocks are generated by dividing the operating clock for data link layer (DLL clock). The DLL clock can be selected from either the internal clock (CANFDCLK) or the external clock (CANFDMCLK).
Figure 46.16 shows a block diagram of the circuit that generates the CAN communication clock.

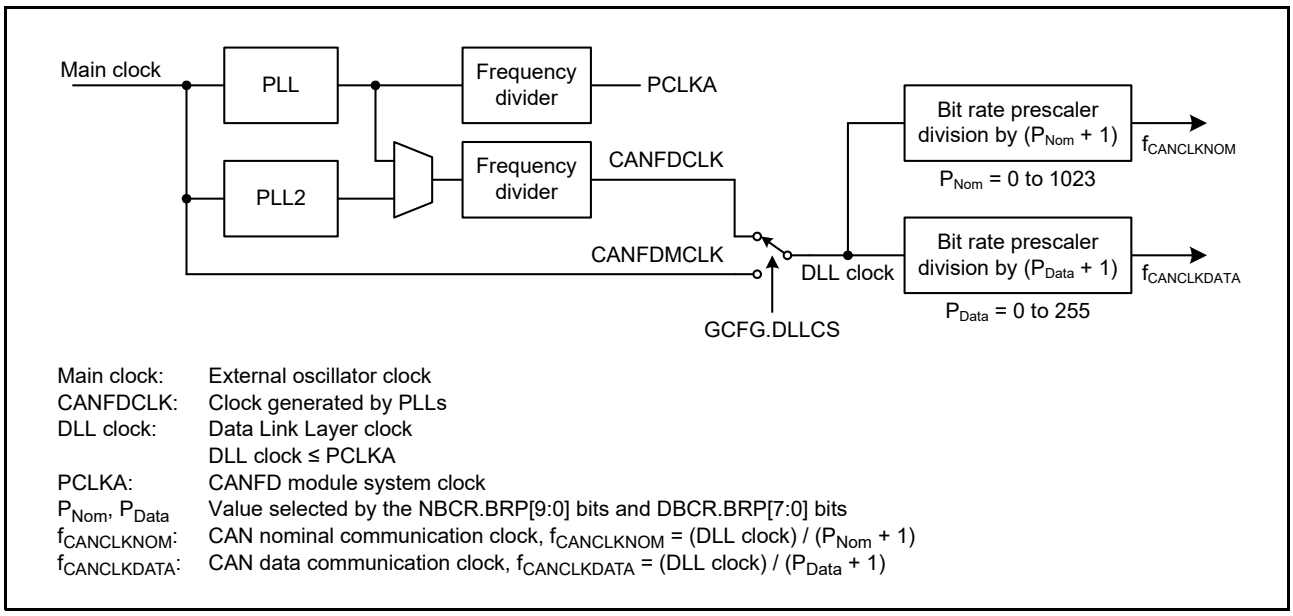


Figure 46.16 Block Diagram of the Circuit that Generates the CAN Communication Clock

The bit rate is determined by the DLL clock frequency, the division value of the bit rate prescaler (P + 1), and the number of Tq per bit.

$$\text{Bit rate} = \frac{\text{DLL clock frequency}}{\text{Number of Tq per bit} \times (\text{P} + 1)} = \frac{\text{CAN communication clock frequency}}{\text{Number of Tq per bit}}$$

Table 46.17 lists examples of setting the nominal bit rate for Classical CAN frame.

Table 46.17 Nominal Bit Rate Setting Examples for Classical CAN Frame

Bit Rate	DLL Clock Frequency																	
	80 MHz		40 MHz		32 MHz		30 MHz		24 MHz		20 MHz		16 MHz		10 MHz		8 MHz*1	
	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1
1 Mbps	8 Tq 20 Tq	10 4	8 Tq 20 Tq	5 2	8 Tq 16 Tq	4 2	10 Tq 15 Tq	3 2	8 Tq 12 Tq 24 Tq	3 2 1	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1	10 Tq	1	8 Tq	1
500 kbps	8 Tq 20 Tq	20 8	8 Tq 20 Tq	10 4	8 Tq 16 Tq	8 4	10 Tq 15 Tq 20 Tq	6 4 3	8 Tq 12 Tq 24 Tq	6 4 2	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1
250 kbps	8 Tq 20 Tq	40 16	8 Tq 20 Tq	20 8	8 Tq 16 Tq	16 8	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 12 Tq 24 Tq	12 8 4	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2
125 kbps	8 Tq 20 Tq	80 32	8 Tq 20 Tq	40 16	8 Tq 16 Tq	32 16	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 12 Tq 24 Tq	24 16 8	10 Tq 20 Tq	16 8	8 Tq 16 Tq	16 8	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4
83.3 kbps	8 Tq 12 Tq 16 Tq 24 Tq	120 80 60 40	8 Tq 12 Tq 16 Tq 24 Tq	60 40 30 20	8 Tq 12 Tq 16 Tq 24 Tq	48 32 24 16	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	45 36 30 24 18 15	8 Tq 12 Tq 16 Tq 24 Tq	36 24 18 12	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	30 24 20 16 15 12 10	8 Tq 12 Tq 16 Tq 24 Tq	24 16 12 8	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	15 12 10 8 6 5	8 Tq	12
33.3 kbps	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	300 200 150 120 100	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	150 100 75 60 50	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	120 96 80 64 60 48 40	10 Tq 12 Tq 15 Tq 20 Tq	90 75 60 45	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	90 72 60 48 45 36 30	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	75 60 50 40 30 25	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	60 48 40 32 30 24 20	10 Tq 12 Tq 15 Tq 20 Tq	30 25 20 15	8 Tq	30

Note 1. Minimum frequency to achieve a nominal bit rate of 1 Mbps.

For optimum clock tolerance in networks using the CAN FD frame, the length of the time quantum should be the same in nominal bit time and in data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
In addition, do not set the DBCR.BRP[7:0] bits greater than 1 when using transceiver delay compensation.
Table 46.18 lists examples of setting the nominal and data bit rate for CAN FD frame.

Table 46.18 Nominal and Data Bit Rate Setting Examples for CAN FD Frame

Bit Rate		DLL Clock Frequency											
		80 MHz			40 MHz			30 MHz			20 MHz		
		No. of Tq			No. of Tq			No. of Tq			No. of Tq		
Nominal	Data	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1
1 Mbps	8 Mbps	80 Tq	10 Tq	1	40 Tq	5 Tq	1	—	—	—	—	—	—
1 Mbps	5 Mbps	80 Tq	16 Tq	1	40 Tq	8 Tq	1	30 Tq	6 Tq	1	—	—	—
1 Mbps	4 Mbps	80 Tq	20 Tq	1	40 Tq	10 Tq	1	—	—	—	20 Tq	5 Tq	1
1 Mbps	2 Mbps	80 Tq	40 Tq	1	40 Tq	20 Tq	1	30 Tq	15 Tq	1	20 Tq	10 Tq	1
500 kbps	2 Mbps	160 Tq	40 Tq	1	80 Tq	20 Tq	1	60 Tq	15 Tq	1	40 Tq	10 Tq	1

46.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate

Figure 46.17 shows the procedure for setting the bit timing and the bit rate.
These settings should be configured when the CAN channel is in CH_RESET mode.
The bit rate must be configured before moving to the channel communication state. Otherwise the mode does not switch correctly.

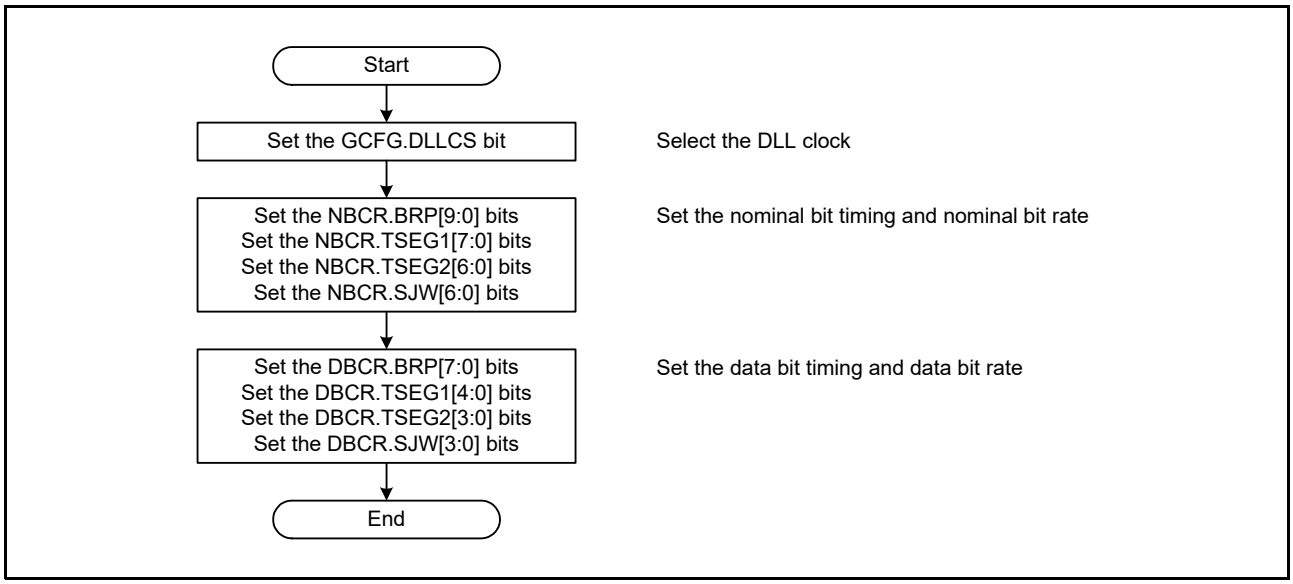


Figure 46.17 Procedure for Setting the Bit Timing and Bit Rate

46.4.1.5 Transceiver Delay Compensation

When a high bit rate is used such as 5 to 8 Mbps for the data phase, the transceiver delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN FD frame. The transceiver delay compensation function compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.
There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN FD frames. This is derived from the Transceiver Delay Compensation Result bit (FDSTS.TDCR[7:0]) as shown in Figure 46.18.
The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

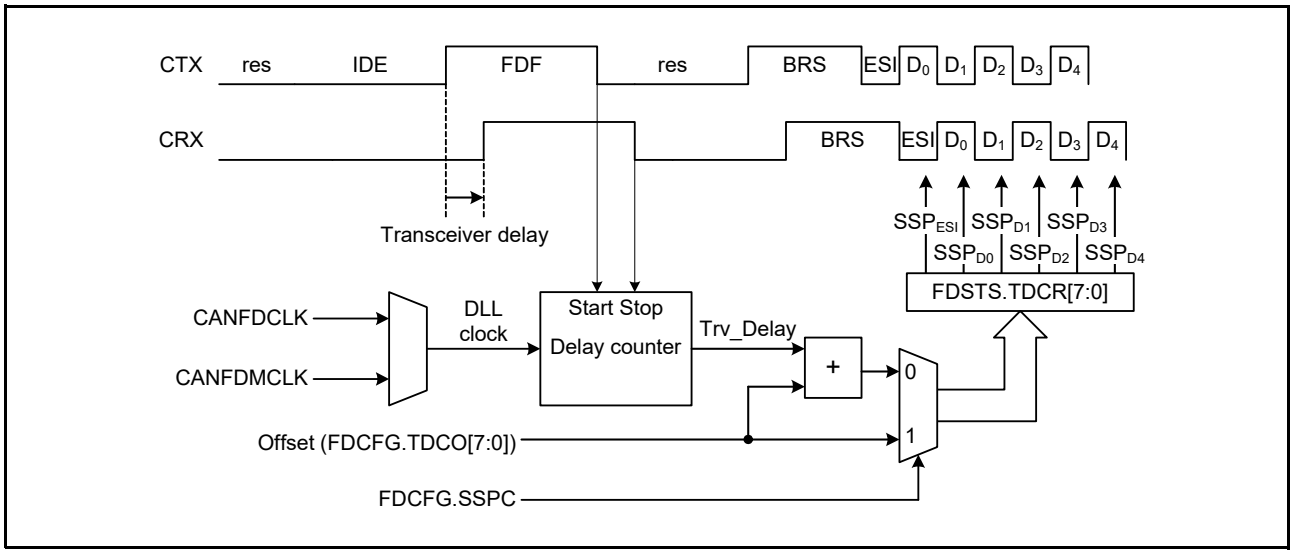


Figure 46.18 Transceiver Delay Compensation

The measured Trv_Delay is based on the number of DLL clock cycles. It counts up by one for each start clock until the dominant value can be observed at the CRX pin. Figure 46.19 shows the measurement example. Trv_Delay is counted up to 127 on each DLL clock.

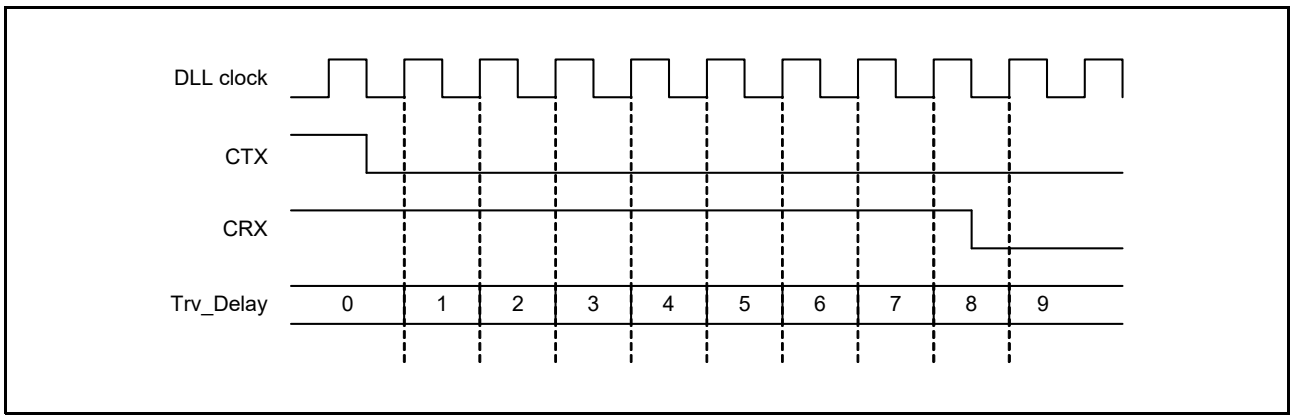


Figure 46.19 Trv_Delay Measurement Example

The SSP is calculated by taking the result from the FDSTS.TDCR[7:0] bits and rounding the value down to the nearest integer number of data Tq.

Figure 46.20 shows the positioning of the secondary sample point (SSP). When the FDCFG.SSPC bit is set to 0, the SSP is equal to the Trv_Delay (measured delay) + FDCFG.TDCO[7:0], rounded down to the nearest integer number of Tq. Normally, the TDCO[7:0] value has the magnitude of SS + TSEG1 in the data phase to position the SSP to a theoretical location of the sample point.

If the FDCFG.SSPC bit is set to 1, the SSP is defined by the FDCFG.TDCO[7:0] bits. If the DBCR.BRP[7:0] bits are greater than 00h, the value is also rounded down to the nearest integer number of Tq.

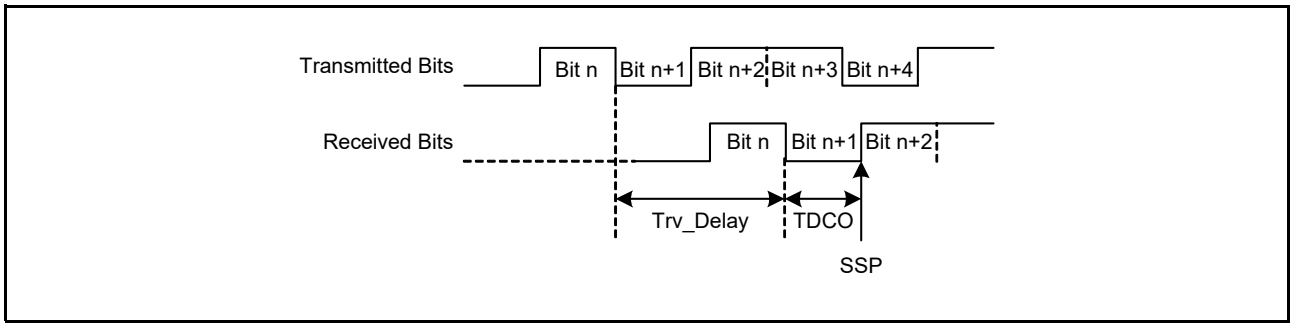


Figure 46.20 Positioning of the Secondary Sample Point (SSP)

The maximum delay time (Trv_Delay + TDCO[7:0]) which can be compensated by the CANFD module is (6 data bit times – 2 DLL clocks). The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.
If different values are used for the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 46.21.
The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
The bit rate can be changed by selecting different settings for the time segments. The nominal bit rate can be set from 8 to 385 Tqs and the data bit rate from 5 to 49 Tqs.

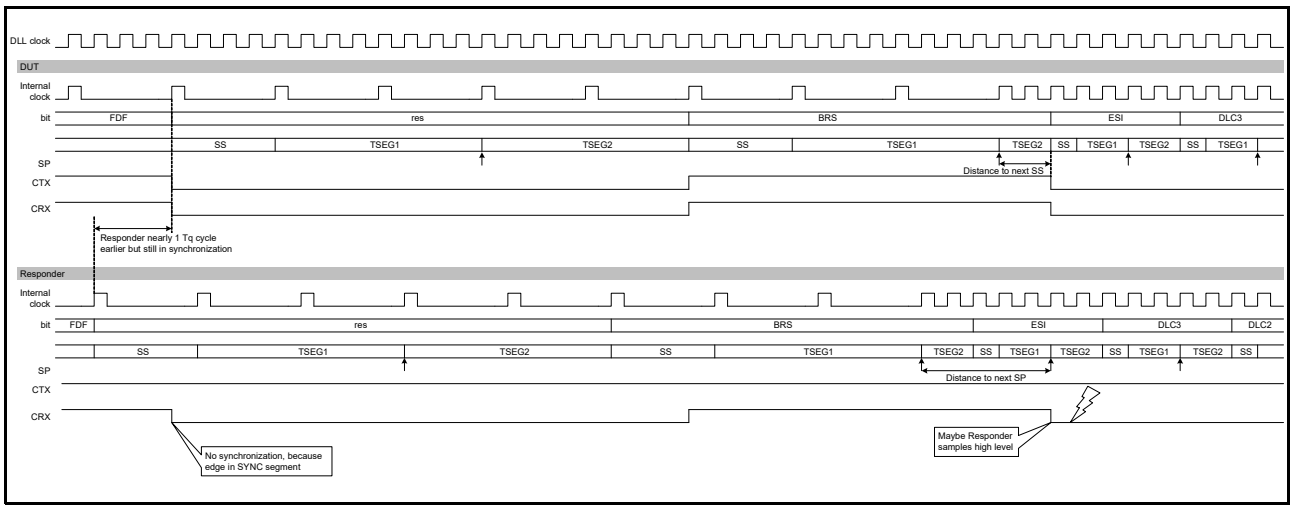


Figure 46.21 Loss of Synchronization Between Two CAN Nodes

The transceiver delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (FDCFG.TDCE = 1, FDCFG.SSPC = 0).
Figure 46.22 shows the read flow to get the measured transceiver delay compensation result.

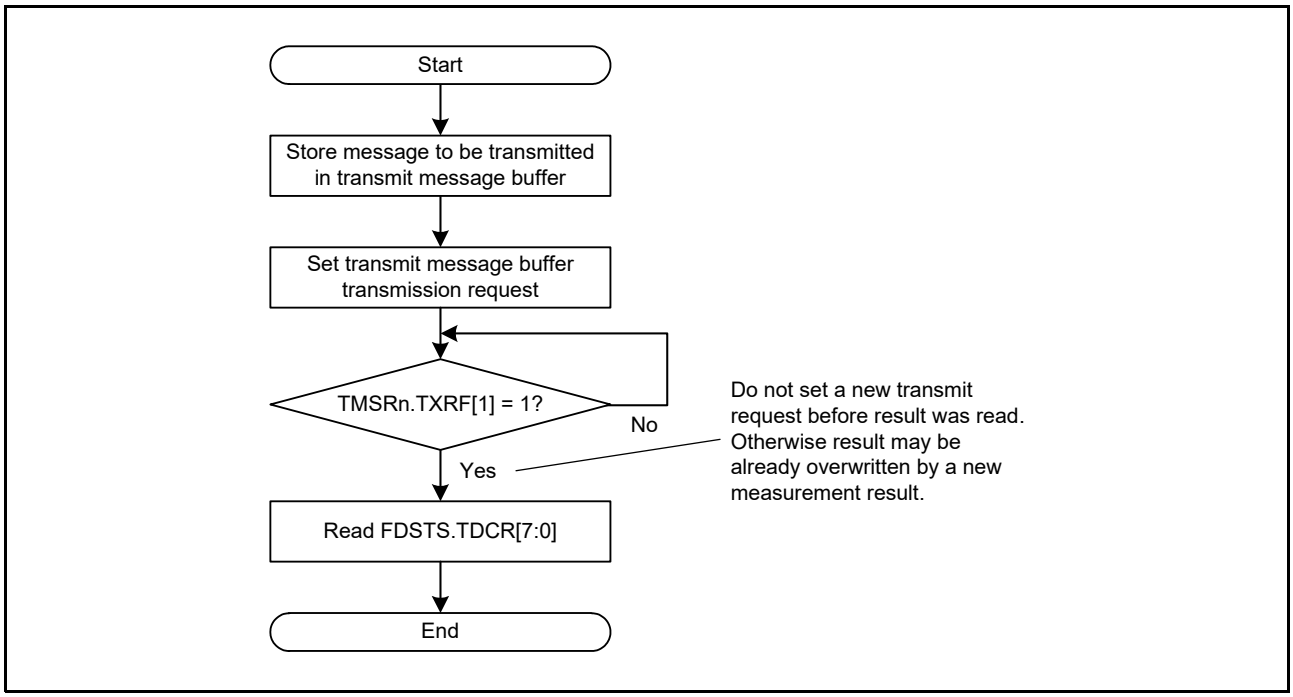


Figure 46.22 Transceiver Delay Compensation Result Read Flow

46.4.2 CANFD Module Configuration after a Reset

After resetting the MCU or applying a software reset using the GRCR.SRST bit, the CANFD module automatically enters GL_SLEEP mode.

To configure the CANFD module, set the GCR.SLPRQ bit to 0 to exit GL_SLEEP mode.

After resetting the MCU, the CANFD module starts RAM initialization when the module-stop state is released and clocks are supplied. At this time, the GSR.RAMST flag is automatically set to 1, indicating that the CANFD module is initializing the RAM.

After RAM initialization is complete, this flag is automatically set to 0.

RAM initialization is necessary to prevent false detection of ECC errors due to undefined data in RAM after MCU reset. Do not access (read or write) any other CANFD registers until the RAM initialization is complete and the RAMST flag is set to 0.

Before entering communication mode, the Acceptance Filter List and message FIFO buffers must be configured. In addition, the CAN channel settings such as CAN bit timing must be configured. To make this configuration, the CAN channel must be released from CH_SLEEP mode and configured for communication in CH_RESET mode (configuration mode).

Figure 46.23 shows the configuration procedure. For details about each step, refer to section 46.5, Filtering Using Acceptance Filter List (AFL), section 46.6, FIFO Buffers and Message Buffer Configuration, section 46.10, Interrupts and DTC/DMA Requests, and section 46.4.1.3, Bit Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting the GRCR.SRST bit to 1.

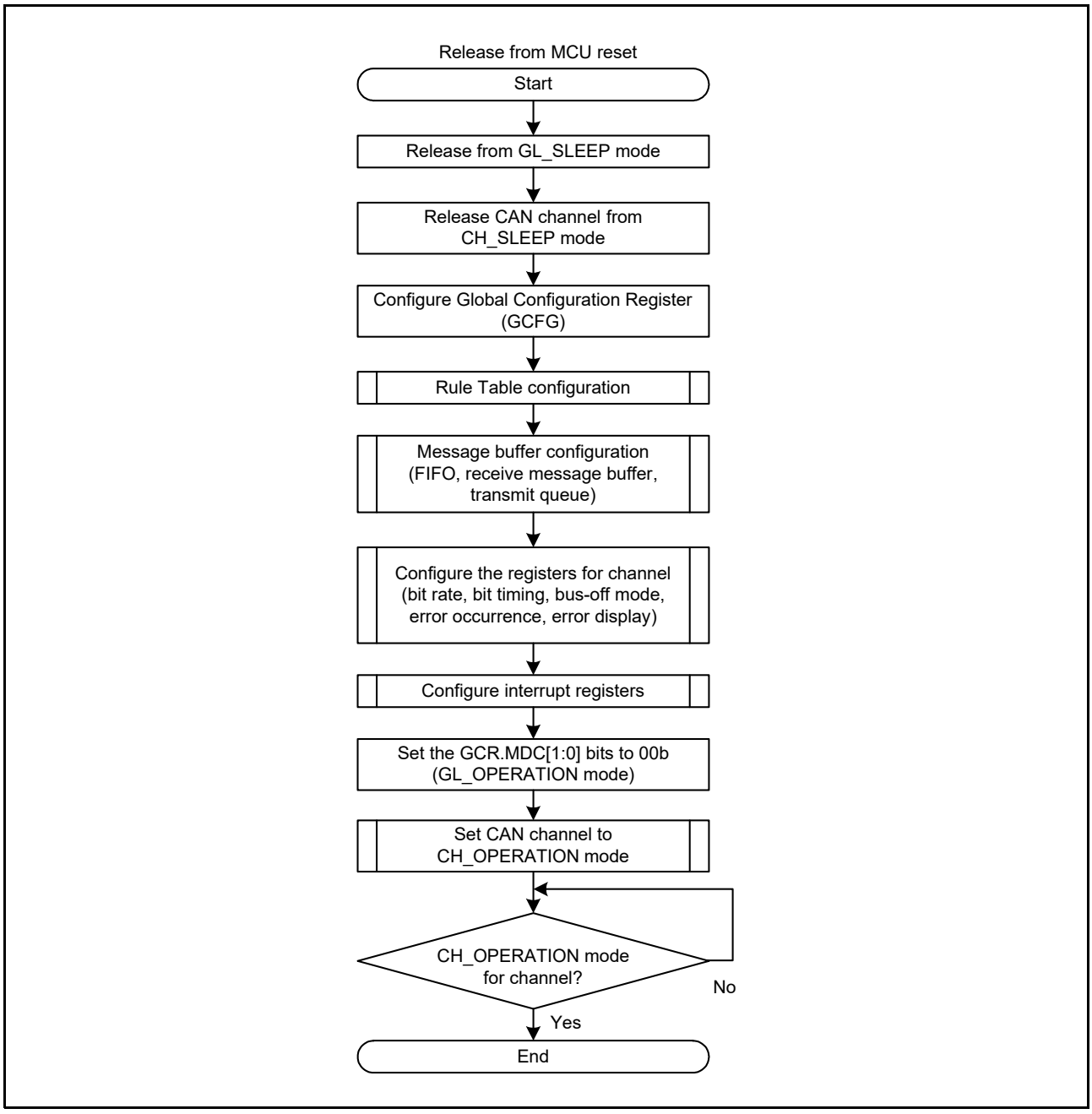


Figure 46.23 Configuration Procedure After An MCU Reset

46.5 Filtering Using Acceptance Filter List (AFL)

The CANFD module can use the Acceptance Filter List (AFL) to filter messages for acceptance. Each entry in the AFL defines a filtering rule for received messages.
The followings are performed based on the AFL entries:

- Acceptance filtering based on the RTR value, IDE value, and ID value of received message
- DLC filtering based on DLC value of received message
- Payload overflow based on the GCFG.OMRC bit
- Storage of accepted messages in the specified message buffer/FIFO buffer
- Attaching a 16-bit pointer to the stored messages, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages.

The CANFD module allows a maximum of 16 AFL entries.

46.5.1 Acceptance Filtering Process

Acceptance filtering matches each AFL entry with the received message. Matching starts with the lowest AFL entry number.
The AFL search stops when the received message ID matches the specified ID/mask combination, or when the received message ID has been matched against all defined AFL entries. If no match occurs, then the received message is discarded. No notification is given to the application in this case.

46.5.2 DLC Filtering Process

DLC filtering is performed for each accepted message if DLC check is enabled (GCFG.DCE bit = 1). If the DLC value of the received message is equal to or higher than the DLC value specified for the AFL entry whose ID matches in the Acceptance Filtering process (the matching AFL entry), the DLC check is passed.
If DLC replacement is enabled (GCFG.DRE bit = 1), DLC value configured in the matching AFL entry is greater than 0000b and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination receive message buffer (RMBn) or FIFO buffer. If DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are not stored in the RMBn/FIFO buffer. These excess data bytes are stored as 00h in the RMBn/FIFO buffer.
If DLC replacement is enabled and the DLC value specified in the matching AFL entry is 0000b (DLC filter function is disabled), the DLC value of the received message is stored in the RMBn/FIFO buffer.
If DLC replacement is disabled (GCFG.DRE bit = 0) and DLC check passes, the DLC value of the received message is stored in the RMBn/FIFO buffer. If the DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are also stored in the RMBn/FIFO buffer.
If DLC value of the received message is less than the DLC value specified in the matching AFL entry, DLC check fails. In this case, the received message is discarded and is not stored anywhere.
Additionally, when the DLC check fails, the GESR.DEDF flag is set to 1. If interrupts are enabled, an error interrupt is also generated. If the DLC check fails, the DLC replacement setting has no effect.

46.5.3 Message Storage

If a received message has passed both acceptance filtering and DLC filtering, the message is stored in receive message buffers 0 to 15, receive FIFO 0, 1, or common FIFO 0 configured in receive FIFO mode.
This message storage target information is also defined in the AFL entry. Do not set a target at the AFL entry which is not configured.
Up to two message storage destinations can be specified. Do not specify more than 3 destinations.

46.5.4 Payload Overflow Process

There is a protection mechanism in case the received message contains data with a payload size that is longer than the size that can be stored in the storage (RMCR.PLS[2:0], RFCR0.PLS[2:0], RFCR1.PLS[2:0], or CFCR0.PLS[2:0]).
If GCFG.OMRC = 0 (message is discarded), the message with data bytes that exceed the specified payload size are discarded and not stored. In this case, even if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is not set to 1.
If GCFG.OMRC = 1 (cut to specified size), only data bytes that exceed the specified payload size are discarded. In this case, if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is set to 1 (message lost has occurred).
Depending on the GCFG.DRE bit setting, either the DLC value of the received message or the DLC value specified in the AFL entry is stored.
Regardless of the GCFG.OMRC bit setting, the GESR.PODF flag is set to 1 if a payload overflow condition is detected.
The DLC filtering is performed before the payload overflow processing. Therefore, for each received frame, only one of the GESR.DEDF and GESR.PODF flags is set to 1.

46.5.5 Allocation of AFL Entries

The number of AFL entries (number of rules) can be configured using the AFCFG.RN0[4:0] bits (refer to Figure 46.24).

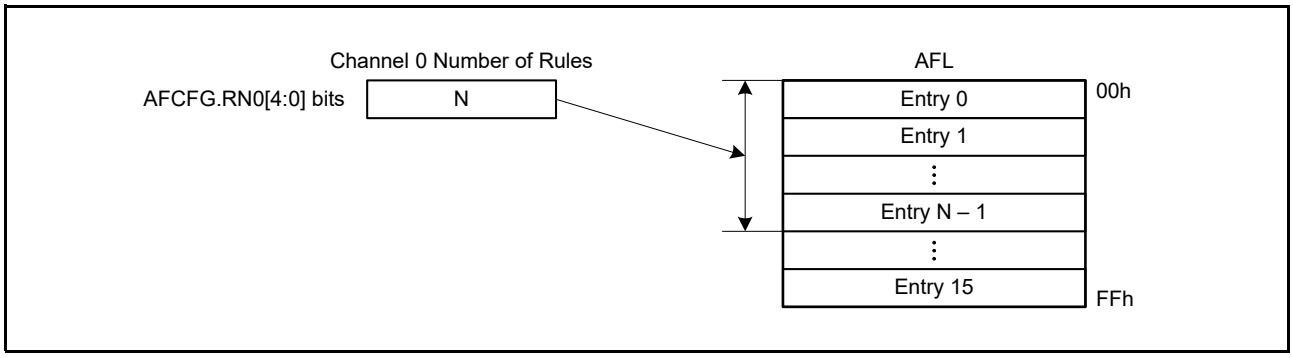


Figure 46.24 Configuration of AFL

The minimum number of entries is 0 (no entries defined for the channel) and the maximum number of entries is 16.
The CANFD module does not flag errors related to the configuration of the AFL.

46.5.6 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.
Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier field (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit matches this field with the identifier field of the received message (29 bits of identifier field can be masked individually, refer to the description of Mask for Identifier field below).
- IDE bit:
  Acceptance filter unit matches this bit with the IDE bit of the received message and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, refer to the description of Mask for IDE bit below).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, refer to the description of Mask for RTR bit below).
- Loop-Back Configuration bit:
  This bit can enable or disable the AFL entry depending on the loop-back configuration or mirror mode condition.
- Mask for Identifier field (29 bits):
  Each bit in the identifier mask field can mask the corresponding identifier bit in the AFL entry (refer to Figure 46.25).
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of an AFL entry, the AFL entry can accept messages in both Standard Identifier format and Extended Identifier format. For messages in Standard Identifier format, the Standard Identifier part of the AFL entry is compared, and for messages in Extended Identifier format, the Extended Identifier part of the AFL entry is compared.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of an AFL entry, the AFL entry can accept frame formats in both data frame and remote frame.
- Pointer (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added when storing a message in the message buffer area and can be used as a support function in the application.
  For example, the pointer information can be used to support the assignment of PDU IDs to the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a message accepted by the related AFL entry. This label is added when storing a message in the message buffer area and can be used as support function in the application.
- DLC field:
  If the DLC value of the received message is equal to or higher than the value set in this field, the DLC check is passed.
  If the DLC value of an AFL entry is set to 0000b, the DLC filtering for this entry is effectively disabled (all accepted messages pass DLC filtering).

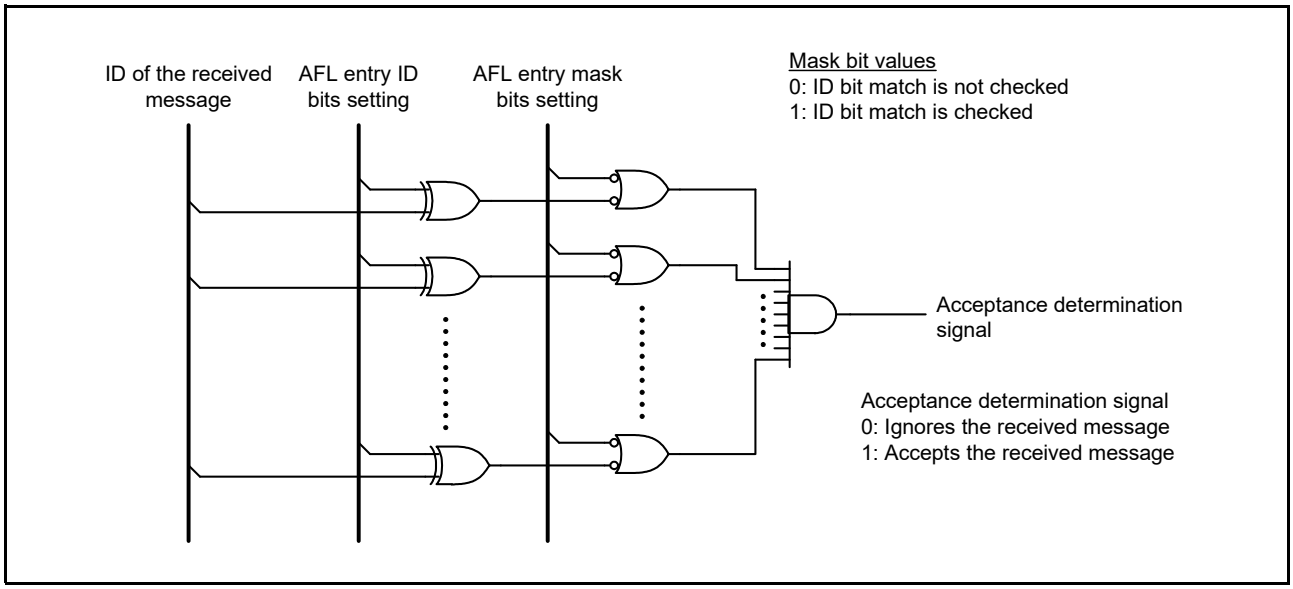


Figure 46.25 Acceptance Filter Function

Each AFL entry contains the following information for processing the received messages:

- Message buffer number of the receive message buffer used as the storage destination for received messages
- The Destination Message Buffer Setting Enable bit to specify the receive message buffer as the storage destination for received messages
- The FIFO Destination Enable bit to specify the FIFO as the storage destination for received messages.

There is no protection function for storing messages. Therefore, the FIFO Destination Enable bit must be set carefully.

46.5.7 Entering Entries in the AFL

One complete entry can be entered into AFL via the following registers:

- AFLn.IDR register: First part of the AFL entry
- AFLn.MASK register: Second part of the AFL entry
- AFLn.PTR0 register: Third part of the AFL entry
- AFLn.PTR1 register: Fourth part of the AFL entry

Follow the flow shown in Figure 46.26 to configure the AFL.
After entering all the entries, writing to the AFL must be disabled to prevent unnecessary write access to the AFL.
If the AFCR.AFLWE bit is set to 0, write protection is enabled during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION).
Even if the AFCR.AFLWE bit is set to 0, reading from AFL is possible during all Global modes (the consistency of AFL contents can be checked during execution).

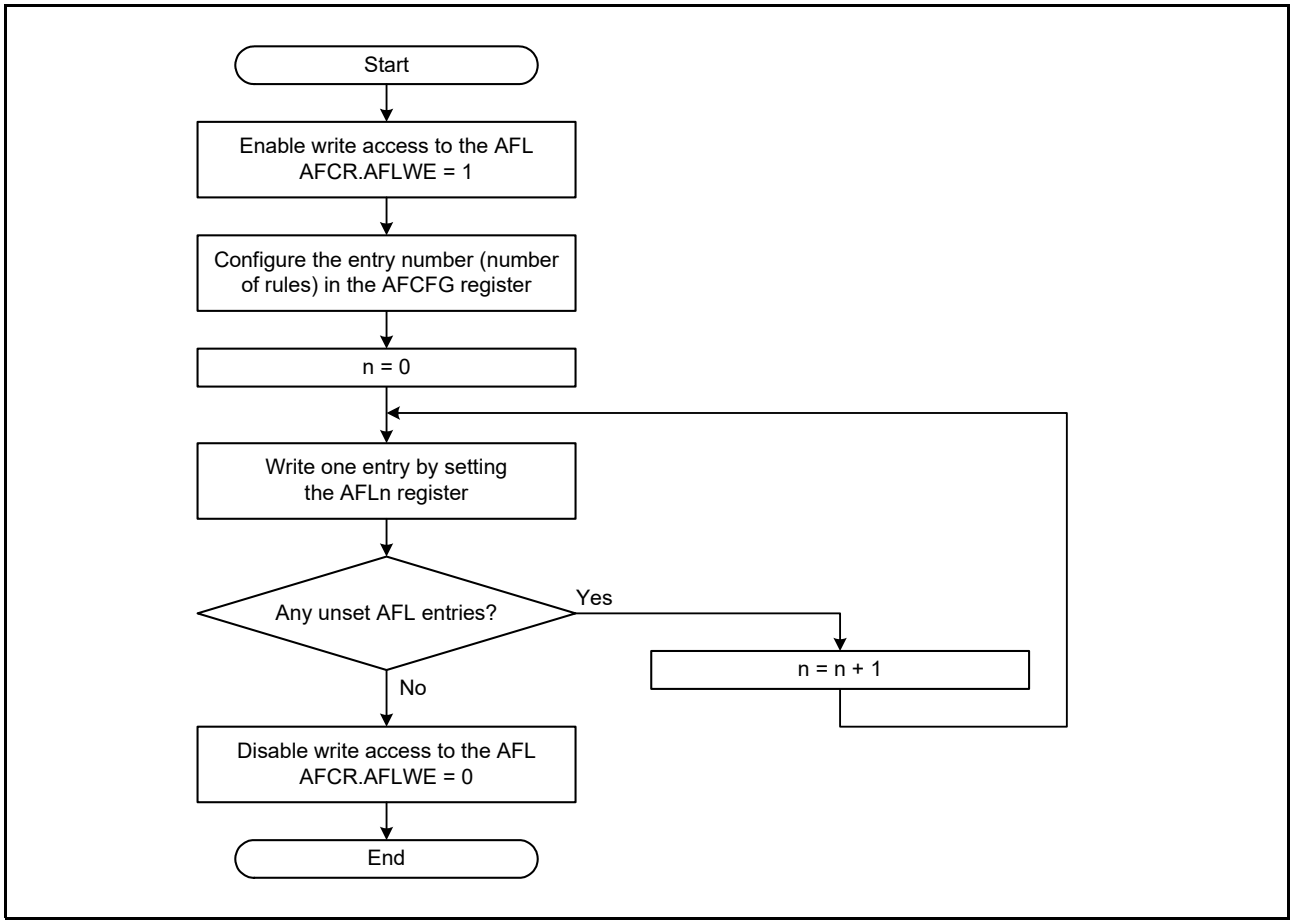


Figure 46.26 AFL Configuration Flow

46.5.8 Loop-Back Configuration

AFL entries with the AFLn.IDR.LPC bit set to 1 are used only in loop-back mode (self-test mode 0 or self-test mode 1) or mirror mode. If a message transmitted by another node on the CAN bus is received while in loop-back mode, the AFL entry will not be used.
AFL entries with the LPC bit set to 0 are used only when the following messages are received.

- Messages transmitted by other nodes
- Messages transmitted by the CAN channel itself in loop-back mode

The mirror mode can be enabled with the GCFG.MME bit. If the message is successfully transmitted when the GCFG.MME bit is 1, the message is stored in the receive message buffer or FIFO buffer if there is a matching entry in the AFL. The AFLn.IDR.LPC bit in the matching AFL entry must be set to 1 to store this message.
If mirror mode and loop-back mode are set at the same time, the loop-back mode behavior is applied. Table 46.19 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 46.19 Behavior of Acceptance Filter Based on the Loop-Back Setting in AFL Entry

Mirror Mode (MME Bit)	Loop-Back Mode (Self-Test Mode 0 or Self-Test Mode 1)	Channel Mode	LPC Bit	AFL Entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

46.5.9 IDE Masking

The IDE bit set in the AFL entry with the AFLn.MASK.IDEM bit set to 0 is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is selected based on the received IDE bit.
The following lines show the examples.

- The ID and mask fields of an AFL entry x is configured as follows:
  - AFLx.IDR = C0553A20h → IDE = 1, RTR = 1, LPC = 0, ID[10:0] = 220h/ID[28:0] = 00553A20h
  - AFLx.MASK = 0000FFFFh → IDEM = 0, RTRM = 0, IDM[10:0] = 7FFh/IDM[28:0] = 0000FFFFh
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 220h is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 320h is received, this is not a match
  - If a frame with IDE = 1 and ID = 1FFF3A20h is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 08803220h is received, this is not a match.

46.5.10 Updating AFL Entry during Communication

The AFL entry can be updated without disabling CAN communications.
Set the AFL entry number to update to the ignore entry select bits and set the ignore entry enable bit to 1.
The entry number specified here is ignored by AFL matching while the entry is being updated.
Figure 46.27 shows the update flow for an AFL entry.

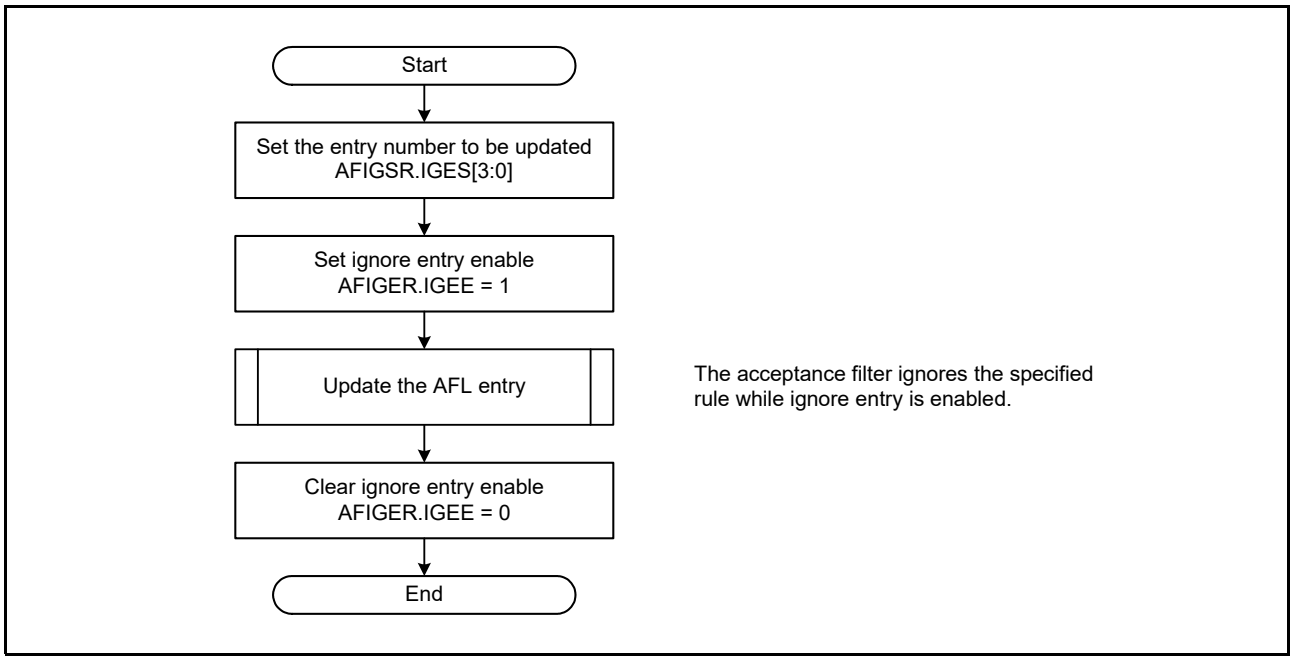


Figure 46.27 Update Flow for An AFL Entry

The method to update an AFL entry is as follows:
(1) Set the entry number*1 to be updated to the AFIGSR.IGES[3:0] bits.
(2) Set the value 0000C401h (key code and ignore entry enabled) to the AFIGER register.
(3) Set the AFCR.AFLWE bit to 1.
(4) Set the new rule to the AFLn.IDR, AFLn.MASK, AFLn.PTR0, AFLn.PTR1 registers.
(5) The AFCR.AFLWE bit is set to 0.
(6) Set the value 0000C400h (key code and ignore entry disabled) to the AFIGER register.

Note 1. The specified entry is not used for acceptance filtering between (2) and (5).

(1) Example 1: Deleting an Entry

The following describes how to delete entry 3 when the total number of entries is 6.

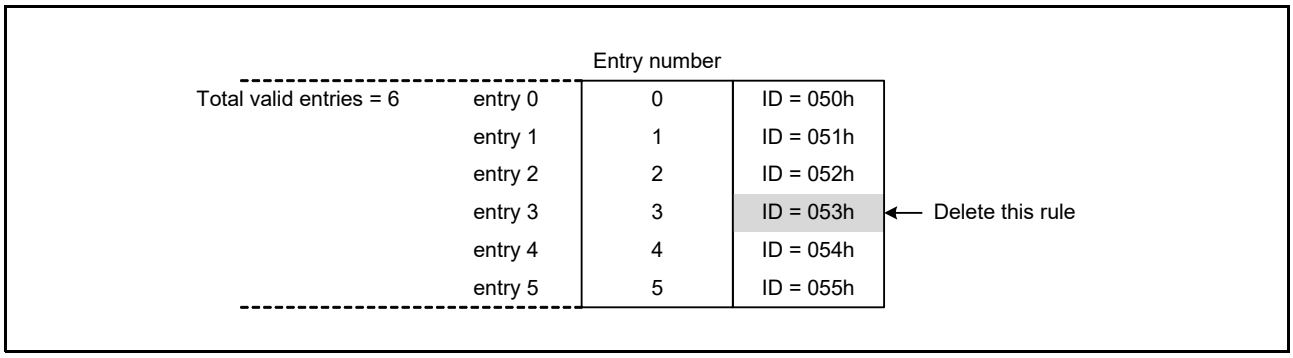


Figure 46.28 Example of Deleting an Entry (Before Deleting Entry 3)

[How to delete an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the same rule as the previous entry by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

Entry 3 is now deleted.

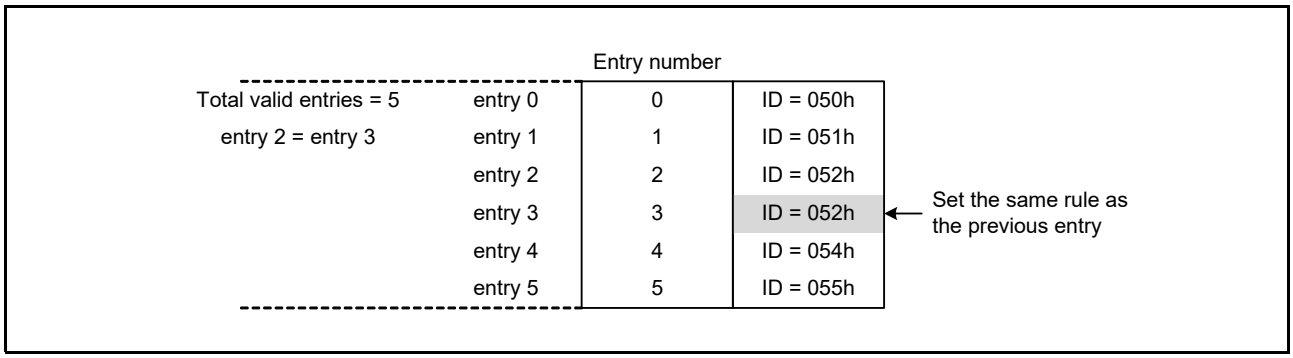


Figure 46.29 Example of Deleting an Entry (After Deleting Entry 3)

(2) Example 2: Adding an Entry (Update Unused Entry)

The following describes how to add a new rule to entry 3 when the total number of entries is 6.

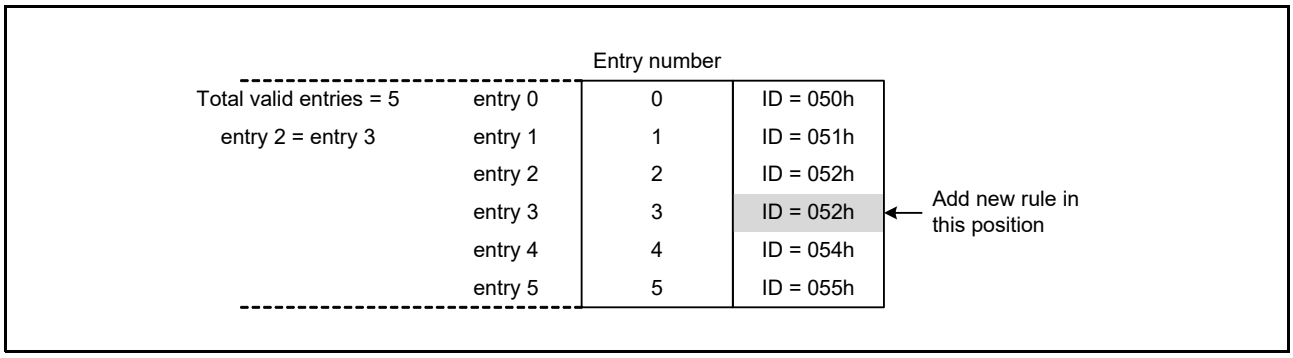


Figure 46.30 Example of Adding an Entry (Before Updating Entry 3)

[How to add an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the new rule by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

The new entry is now added.

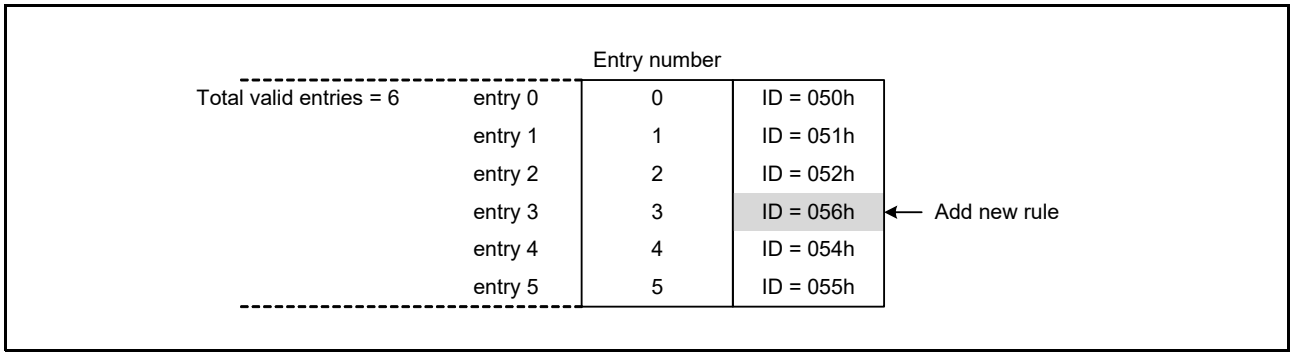


Figure 46.31 Example of Adding an Entry (After Updating Entry 3)

The acceptance filter can use entries in the range of value set in the AFCFG register and entries can be added/deleted within that range. Therefore, the AFCFG register should be set to the maximum number of entries to use.

46.6 FIFO Buffers and Message Buffer Configuration

This section describes the process for configuring the number of receive message buffers, the FIFO buffers, and the transmit message buffers in the CANFD module. The message buffers are mapped as shown in Figure 46.32.
The receive message buffers can be accessed with the RMBn register (n = 0 to 15).
The receive FIFOs can be accessed with the RFBn register (n = 0, 1).
The common FIFO can be accessed with the CFB0 register.
When the common FIFO is configured in transmit FIFO mode, data can only be written to the FIFO buffer via the CFB0 register.
When the common FIFO is configured in receive FIFO mode, data can only be read via the CFB0 register.
The transmit message buffers can be accessed with the TMBn register (n = 0 to 3).
Reading an unused message buffer results in reading an undefined value.

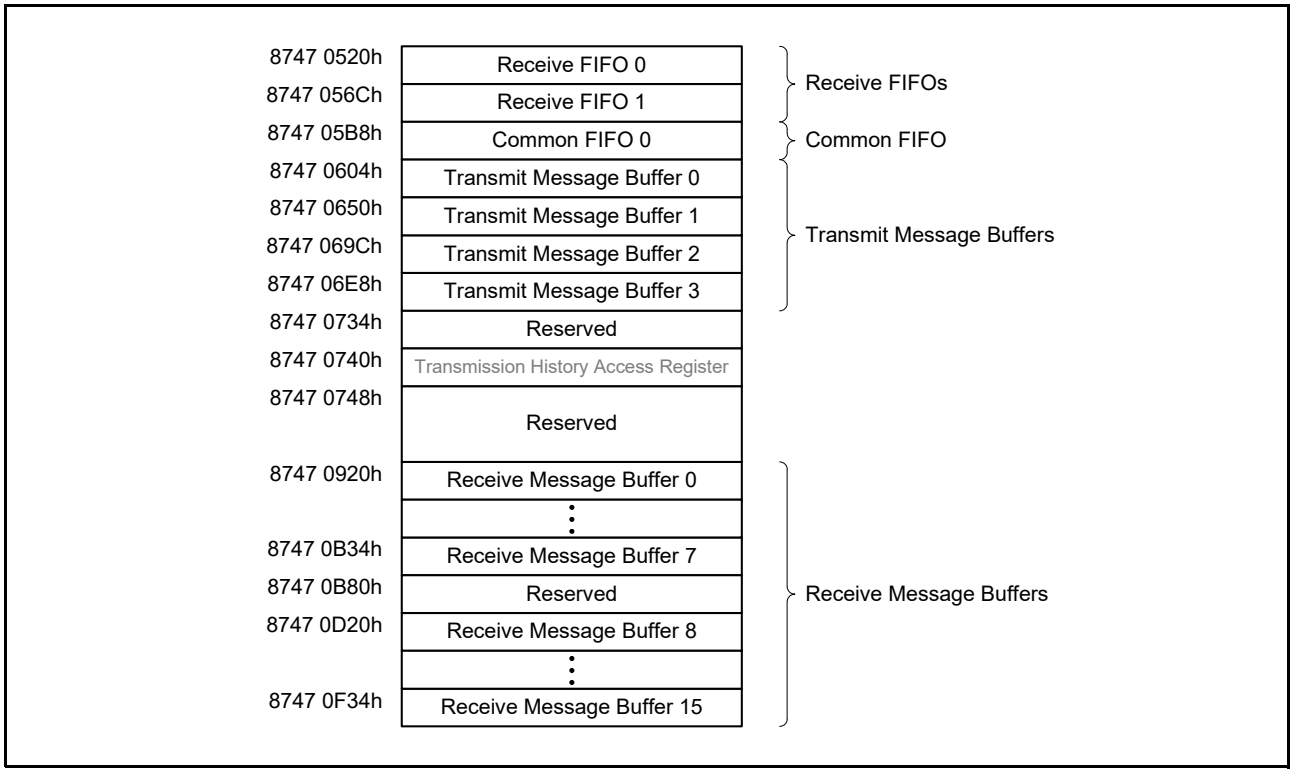


Figure 46.32 Message Buffer Configuration

46.6.1 Receive Message Buffers

In CANFD module, the received frames can be stored in receive message buffers based on the configuration of the AFL entries.
Additionally, the number of receive message buffers required by the system can be selected from 0 to 16.

46.6.1.1 Receive Message Buffer Configuration

The number of receive message buffers in CANFD module can be configured by writing to the RMCR.NMB[4:0] bits. Set the number of message buffers in the range 0 (no receive message buffers) to 16. Do not set a value larger than this.
The AFL entries for routing the received messages to receive message buffers must be configured to match the requirements of the system.
The AFL entries must also be configured properly, and an AFL entry for receive message buffers should not exceed the number of message buffers configured in the RMCR.NMB[4:0] bits.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The payload size of the receive message buffer can be configured with the RMCR.PLS[2:0] bits. The default size is 8 bytes and the maximum size is 64 bytes.
If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

46.6.2 FIFO Buffers

The CANFD module provides FIFO buffers for storing received frames and frames to be transmitted, respectively.
There are two receive-only FIFOs, but common FIFO can be configured to store messages for transmission or reception.
These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- FIFO depth
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the transmit FIFO

If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

46.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.
The total number of FIFO buffers is three (two receive FIFOs + one common FIFO).

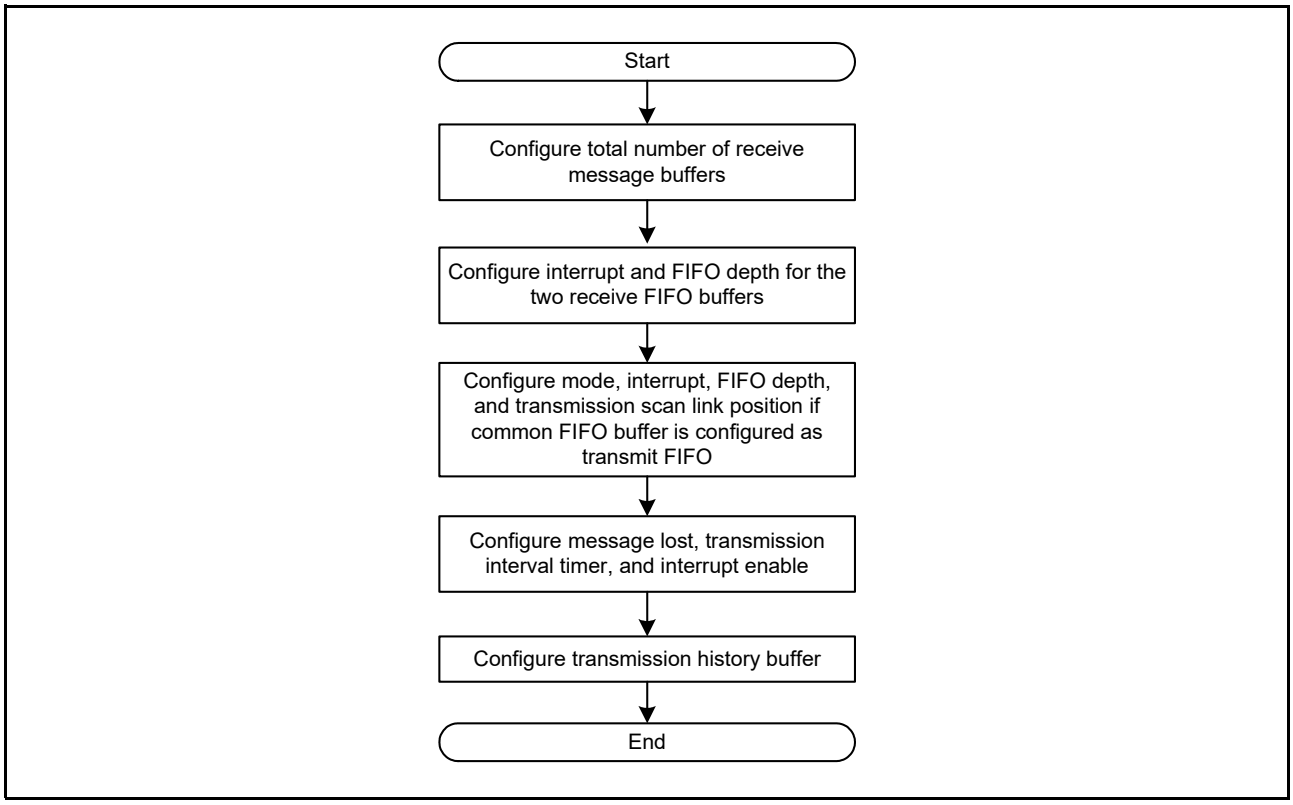


Figure 46.33 FIFO Buffer Configuration Flow in CANFD Module

As shown in Figure 46.33, the various FIFO buffers can be configured by writing to the RFCRn registers and the CFCR0 register (n = 0, 1).

For the two receive FIFOs, the following parameters can be configured:

- Interrupts
- FIFO depth
- Payload size

For the common FIFO, the following parameters can be configured:

- Mode
- Interrupts
- FIFO depth
- Payload size
- Transmission scan link position

(1) FIFO mode configuration of common FIFO

The mode of the common FIFO can be configured by writing to the CFCR0.MODE bit. The modes that can be configured in the common FIFO are as follows:

- 00b: Receive FIFO mode (default mode after an MCU reset)
- 01b: Transmit FIFO mode.

Messages can only be read from the receive FIFOs and the common FIFO configured in receive FIFO mode. Messages are stored by the CANFD module in these FIFO buffers based on the AFL entries.
Messages can be read and written into the common FIFO configured in transmit FIFO mode.
The pointers can be incremented only when a new message is stored in the FIFO buffer and decremented only when a message is transmitted on the corresponding CAN channel by the CANFD module.
After an MCU reset, all the common FIFO is configured in receive FIFO mode by default. Only enable the FIFO buffers after configuring the common FIFO in the required modes.

(2) FIFO transmit message buffer link configuration

When the common FIFO is configured as transmit FIFO, the FIFO buffer must be linked to a normal transmit message buffer to participate in the transmission scan of a CAN channel.
Do not write data into a transmit message buffer that is linked to a common FIFO. Also, the transmit message buffer linked to a common FIFO should not be a part of the transmit queue.
The link to the transmit message buffer of common FIFO can be configured by writing to the CFCR0.LTM[1:0] bits. The options available for linking the transmit message buffer are:

- 00b: Transmit message buffer 0
- 01b: Transmit message buffer 1
- 10b: Transmit message buffer 2
- 11b: Transmit message buffer 3.

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the RFCRn.FDS[2:0] bits and CFCR0.FDS[2:0] bits. The six available options for depth configuration are:

- 000b: 0 messages (FIFO buffer cannot be used)
- 001b: 4 messages
- 010b: 8 messages
- 011b: 16 messages
- 100b: 32 messages
- 101b: 48 messages.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

Note: If the FIFO depth of the common FIFO is 4 messages or more (CFCR0.FDS[2:0] > 000b), the link between the common FIFO and the transmit message buffer is valid regardless of whether the FIFO is disabled or enabled. If the FIFO depth is 0 messages, the link between the common FIFO and transmit message buffer is invalid regardless of whether the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The payload size of each FIFO buffer can be configured by writing to the RFCRn.PLS[2:0] bits and CFCR0.PLS[2:0] bits. The eight available options for payload size configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

(5) FIFO interrupt configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the RFCRn.RFIM bit and CFCR0.CFIM bit. The two available options are:

- RFIM/CFIM = 0:
  - Receive FIFO mode: An interrupt is generated when the FIFO fill level reaches RFCRn.RFITH[2:0] or CFCR0.CFITH[2:0] value
  - Transmit FIFO mode: An interrupt is generated when the FIFO transmits the last message successfully
- RFIM/CFIM = 1:
  - Receive FIFO mode: An interrupt is generated each time the received message is stored
  - Transmit FIFO mode: An interrupt is generated each time a message is successfully transmitted.

If the RFCRn.RFIM bit for the receive FIFO is 0, an interrupt is generated based on the setting of the RFCRn.RFITH[2:0] bits.
Similarly, if the CFCR0.CFIM bit for a common FIFO set to receive FIFO mode is 0, an interrupt is generated based on the setting of the CFCR0.CFITH[2:0] bits.
The eight options available to set the FIFO fill level for generating interrupts are:

- 000b: Interrupt generated when FIFO is 1/8 Full
- 001b: Interrupt generated when FIFO is 1/4 Full
- 010b: Interrupt generated when FIFO is 3/8 Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8 Full
- 101b: Interrupt generated when FIFO is 3/4 Full
- 110b: Interrupt generated when FIFO is 7/8 Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message fill level matches the set value.
However, as shown in Table 46.20, there are some restrictions on the RFITH[2:0] and CFITH[2:0] bit settings, depending on the FDS[2:0] bits (FIFO depth setting) of each register.

Table 46.20 FIFO Interrupt Threshold and FIFO Depth Settings

	RFITH[2:0]/CFITH[2:0]							
FDS[2:0]	**111b (full)**	**110b (7/8)**	**101b (3/4)**	**100b (5/8)**	**011b (1/2)**	**010b (3/8)**	**001b (1/4)**	**000b (1/8)**
000b (0 messages)	Invalid (FIFO cannot be enabled)							
001b (4 messages)	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited
010b (8 messages)	Allowed							
011b (16 messages)	Allowed							
100b (32 messages)	Allowed							
101b (48 messages)	Allowed							

46.6.2.2 FIFO Buffers Control

To enable the receive FIFO interrupt, set the RFIE bit in the RFCRn register to 1 (n = 0, 1).
To enable the common FIFO interrupt, set either the CFRIE or CFTIE bit in the CFCR0 register:
After configuration is complete, each FIFO can be enabled by setting the RFCRn.RFE bit and CFCR0.CFE bit to 1 to allow transmission and reception of messages.

46.7 Reception and Transmission

46.7.1 Reception

In the CANFD module, received messages are stored in the receive message buffers, the receive FIFOs, or the common FIFO configured in receive FIFO mode, depending on the entries in the Acceptance Filter List:

- up to 16 receive message buffers can be configured
- two receive FIFOs available
- up to one common FIFO can be configured in receive mode

46.7.1.1 Message Storage in Receive Message Buffers

When a message is successfully received and stored in a receive message buffer, the corresponding NDF[n] flag in the RMNDR register is set to 1.
The stored message can be read from the corresponding receive message buffer.
If a new message is stored into a receive message buffer before the previous message in this message buffer is read, the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the receive message buffer. If such a loss of messages is not acceptable, store related messages by using receive FIFO.

Note: Unused data bytes are filled with 00h depending on the DLC value.

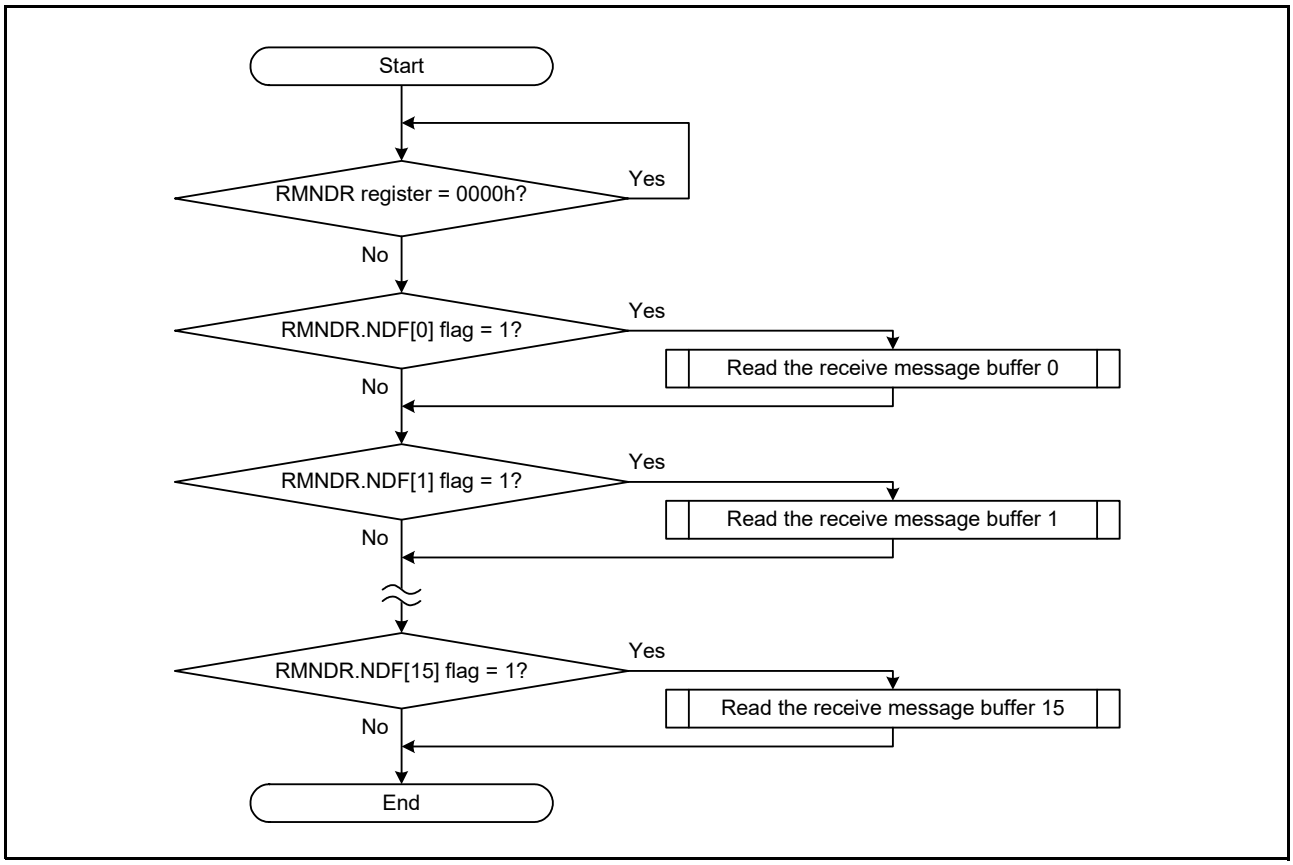


Figure 46.34 Receive Message Buffer Message Access Flow (Example for Polling Case)

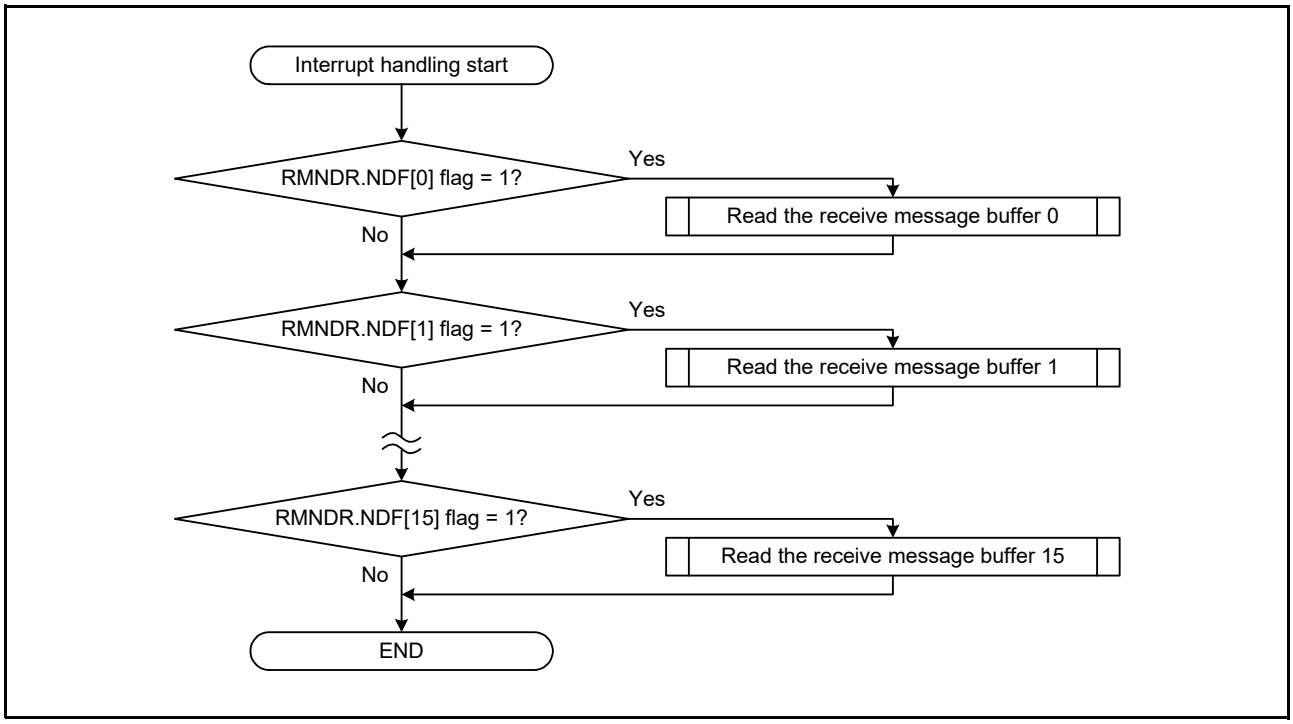


Figure 46.35 Receive Message Buffer Message Access Flow (Example for Interrupt Case)

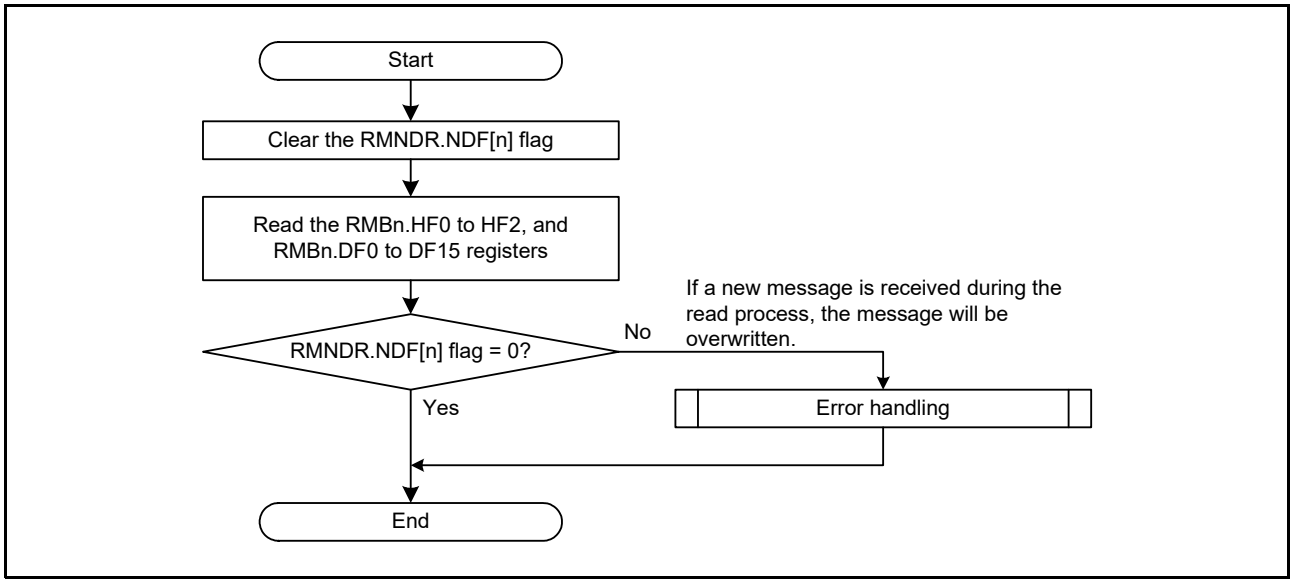


Figure 46.36 Receive Message Buffer n Read Flow

46.7.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to receive FIFOs or common FIFO configured in receive mode should be configured based on the requirements of the system.
The AFLn.PTR1.RF0E, RF1E, or CF0E bit in the matching AFL entry selects the FIFO buffers to which the related reception message will be stored.
When the received message is stored in one or more receive FIFOs or common FIFO configured in receive FIFO mode, the message counter value is incremented in the corresponding Receive FIFO Status Register or Common FIFO Status Register.
Depending upon the configuration of the FIFO buffers, an interrupt may also be generated. The message can be read from the corresponding FIFO access registers.

Note: Since many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, then the FIFO full flag is set.
When the value 000000FFh is written to the corresponding FIFO Pointer Control Register, then the message count is decremented by 1.

Only write 000000FFh to the FIFO Pointer Control Register after completely reading the message from the FIFO Access Register of the corresponding FIFO.
When all the messages stored in the FIFO are read, then the FIFO empty flag is set.
If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO message lost flag is set and the new message will be lost (messages already stored will not be overwritten).
To prevent message loss due to overrun, set an appropriate interrupt generation threshold and generate an interrupt before the FIFO becomes full.

The receive FIFOs and the common FIFO configured in receive FIFO mode can be disabled at any time by clearing the RFCRn.RFE bit or CFCR0.CFE bit.
When the RFCRn.RFE bit or CFCR0.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further messages can be stored into the FIFO.
When the receive FIFOs or common FIFO configured in receive FIFO mode is set to be read by DTC/DMA transfer, do not read the FIFO buffer on the CPU or write 000000FFh to the FIFO Pointer Control Register (RFPCR0, RFPCR1, or CFPCR0). The FIFO read pointer is automatically updated when read by DTC / DMA transfer.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

Note: When the next frame is received before clearing the receive interrupt flag, the receive interrupt flag is not set again.

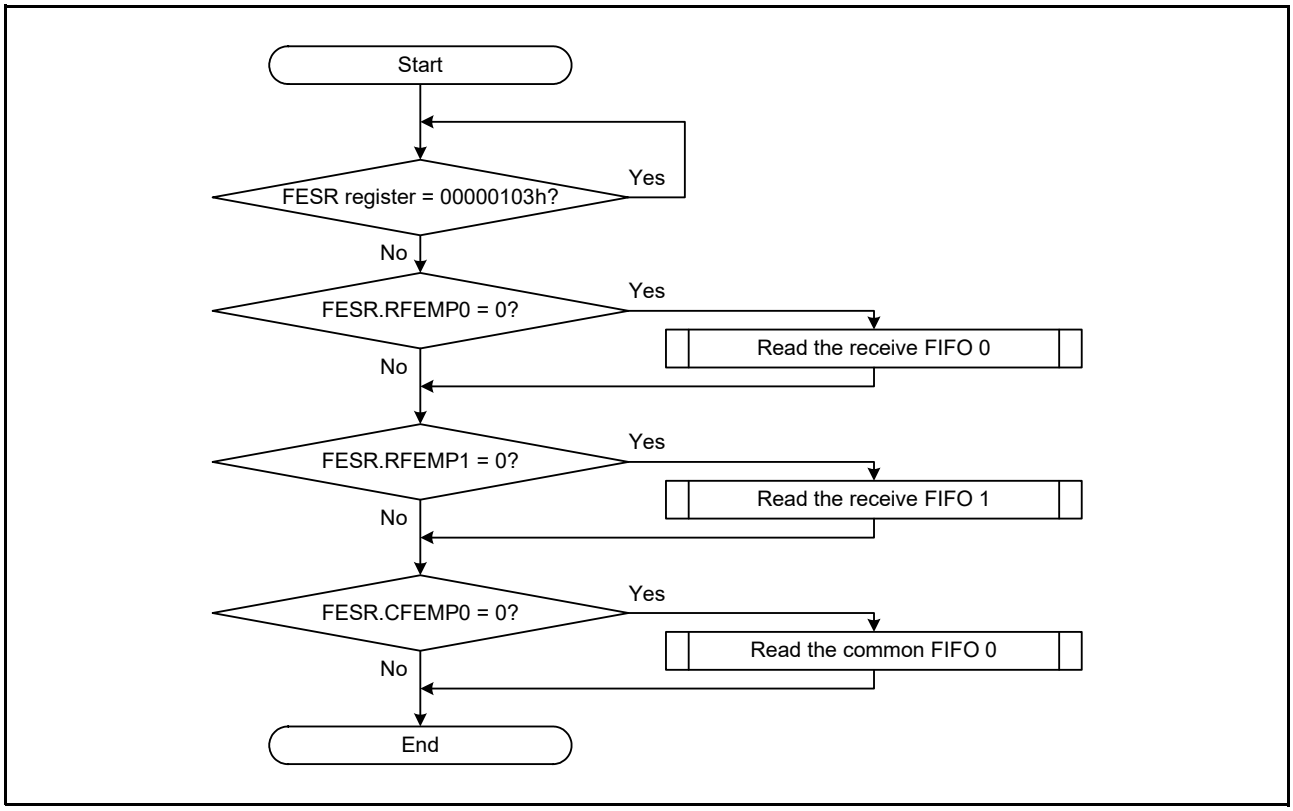


Figure 46.37 FIFO Buffer Message Access Flow (Example for Polling Case)

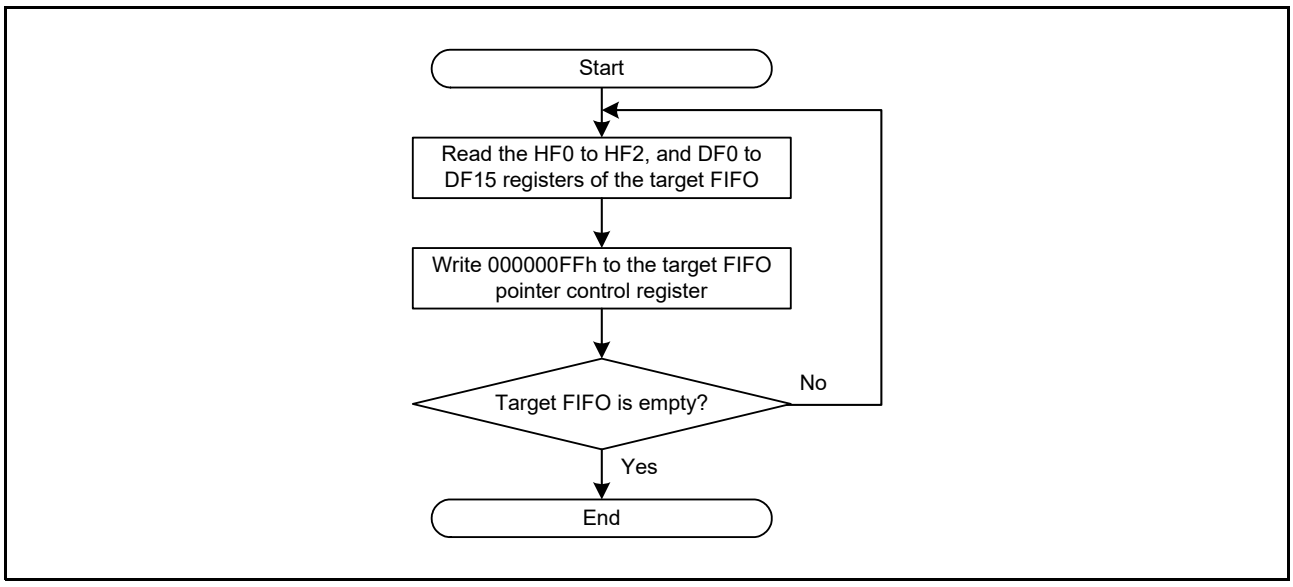


Figure 46.38 FIFO Buffer Read Flow (Example for Polling Case)

If the interrupt flag is cleared after the FIFO read process is completed, the interrupt flag is not set even if the next frame has already been received. The FIFO read process must be performed and the interrupt flag must be cleared before the reception of the next frame is completed. If the process is not in time, make sure the FIFO is empty, clear the interrupt flag, and make sure the FIFO is empty again.

46.7.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of a received message or transmission time of a successful transmitted message. The timestamp counter value will be captured based on the GFDCFG.TSCPS[1:0] bit setting (sample point of SOF, EOF when the frame is taken valid, or sample point of res bit following the FDF bit in case of a CAN FD frame). For reception the timestamp counter value is stored together with the message ID and data into the target receive message buffer or receive FIFO.
For transmit message the timestamp counter value is stored as part of the transmission history entry.
The counter can be clocked from PCLKA or the bit timing clock of the CAN channel. The counter clock source can be set with the GCFG.TSCS bit. If the GCFG.TSCS bit is 0, PCLKA is used. If it is 1, the bit time clock of CAN channel is used.
The clock source of the timestamp counter can be divided by the coefficient defined by the GCFG.TSP[3:0] bits (timestamp prescaler).
The timestamp counter can be reset to 0000h with the GCR.TSCR bit (timestamp counter reset).

46.7.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- Transmit queue transmission

The CANFD module has four transmit message buffers. These message buffers are only used for transmission and cannot be configured for reception.
Additionally transmission from transmit queue and/or common FIFO configured in transmit FIFO mode can be set in the following way (refer to Figure 46.39):

- Transmit queue
  3 or 4 transmit message buffers can be grouped to form a transmit queue with a common access window.
  Transmit message buffer 0 acts as an access window for transmit queue 0 (TXQ0).
- Common FIFO (transmit FIFO mode)
  The CANFD module has one common FIFO. Common FIFO configured in transmit FIFO mode is linked to any of the transmit message buffers 0 to 3.
  The linked transmit message buffer replaces the common FIFO. Do not access the TMCRn or TMSRn registers of the linked transmit message buffer.

Note: Common FIFO should not be linked to transmit message buffers that are already part of a transmit queue.

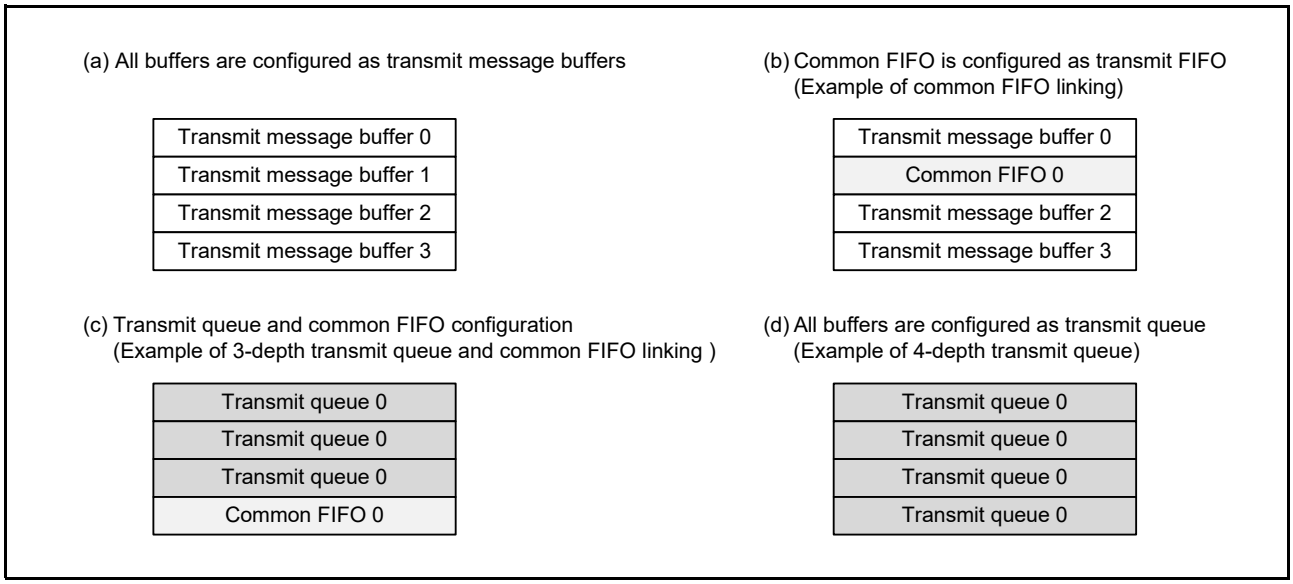


Figure 46.39 Channel Transmit Message Buffer Configuration

46.7.2.1 Transmission Priority

If two or more transmit message buffers are queued for transmission within a channel, messages are transmitted according to one of the following priorities.

- CAN ID priority
- Message buffer number priority

The transmission priority can be set using the GCFG.TPRI bit, and it applies to all message buffers.
For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode.
Do not select message buffer number priority when using the transmit queue.
For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All transmit message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode and includes the transmit queue message buffers.
If the ID of two or more message buffers is the same, then the smaller message buffer number will have higher priority for transmission.

Note: For common FIFO configured in transmit FIFO mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.
If the message is being transmitted from the FIFO, then the next pending message within the same FIFO will be considered in the transmission arbitration.
In contrast to this, all transmit message buffers of a transmit queue will participate in internal transmission arbitration.

Figure 46.40 below shows the transmission configuration flow.

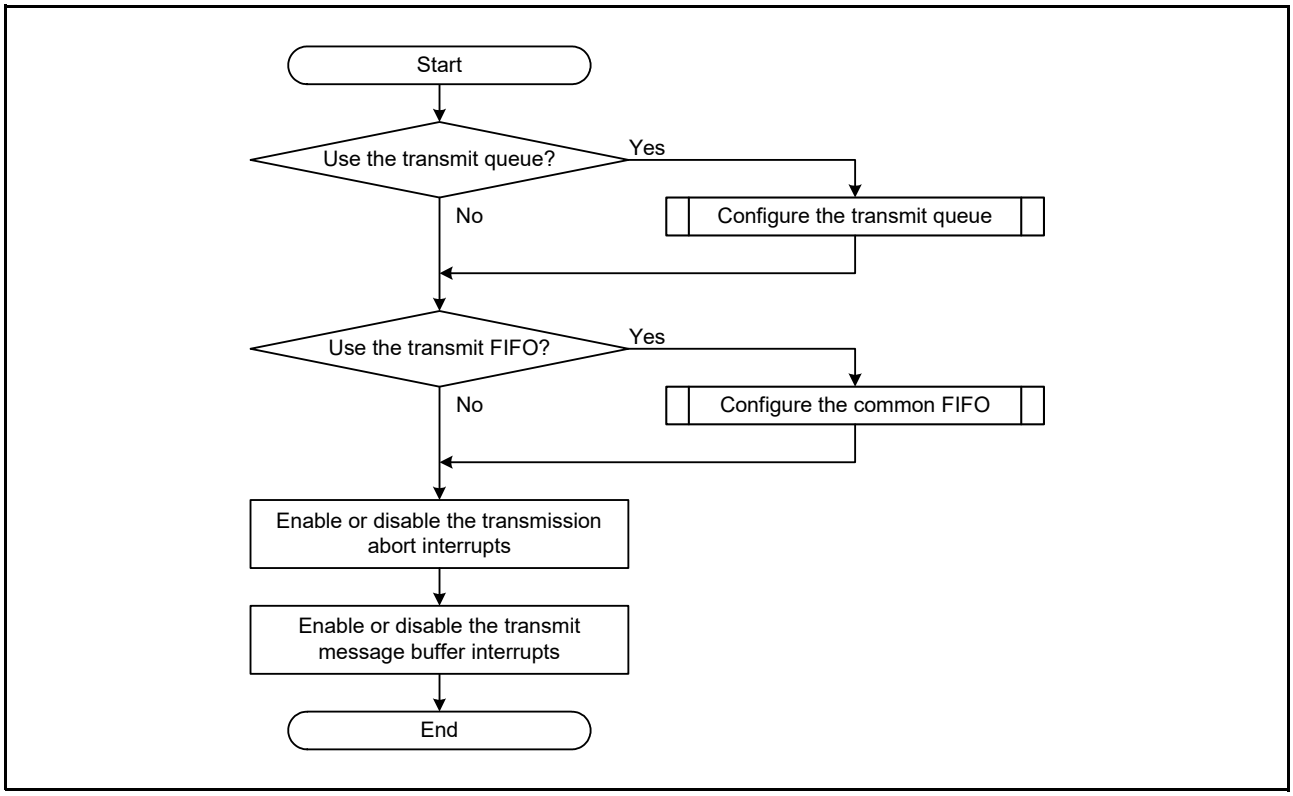


Figure 46.40 Transmission Configuration Flow

46.7.2.2 Message Transmission from Transmit Message Buffer

Each transmit message buffer has two modes of message transmission:

- Normal Transmission Mode
  When the message buffer is set to normal transmission mode, the data frames or remote frames set in that message buffer can be transmitted.
  Completion of normal transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when a normal transmission is successful.
  When arbitration is lost or an error occurs during transmission, the message transmission will be retried if no transmission abort request is set in the transmit message buffer.
  A new internal transmission arbitration is performed for all message buffers that have a transmission request.
- One-Shot Transmission Mode
  When the TMCRn.ONESHOT bit is set to 1, the message buffer is placed in one-shot transmission mode and attempts to transmit the message only once.
  Completion of one-shot transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when the one-shot transmission is successful.
  The TXRF[1:0] flags are set to 01b when arbitration is lost or an error occurs during transmission. In this case, message transmission will not be retried.

Figure 46.41 shows the transmission request procedure from transmit message buffer.

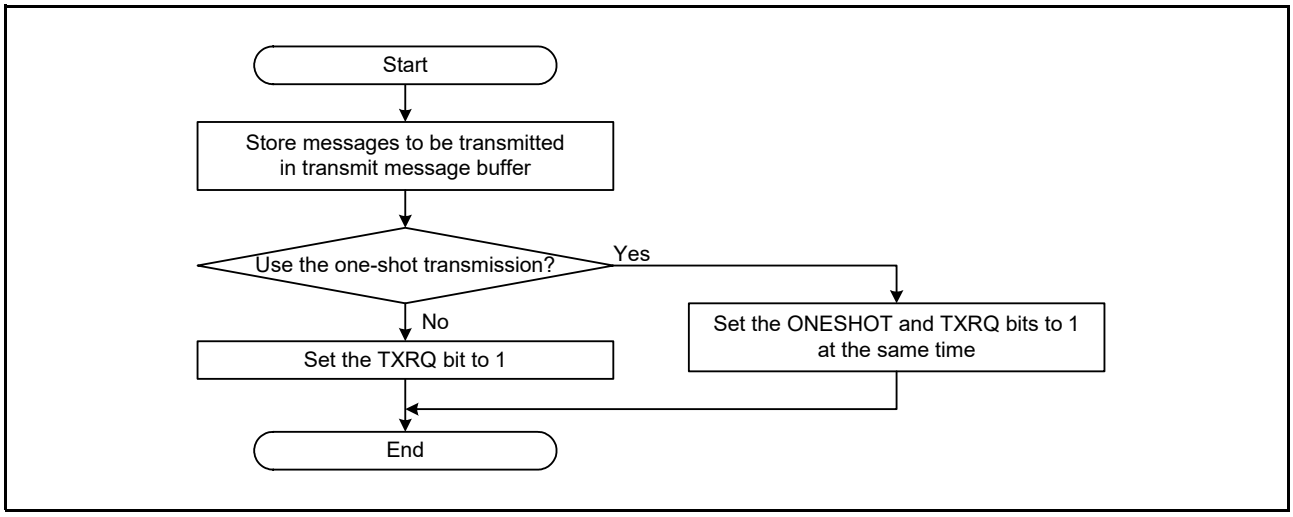


Figure 46.41 Transmission Request Procedure from Transmit Message Buffer

Table 46.21 shows configuration of the TMCRn register.

Table 46.21 Configuration of TMCRn Register

Transmission Request TXRQ Bit	Transmission Abort Request TARQ Bit	One-Shot Transmission Enable ONESHOT Bit	Message Buffer Status
0	0	0	Normal transmission is stopped.
0	0	1	One-shot transmission is stopped.
1	0	0	Data frames or remote frames are transmitted in normal transmission mode.
1	0	1	A data frame or a remote frame is transmitted in one-shot transmission mode.
1	1	0	Transmission abort is requested.
1	1	1	One-shot transmission abort is requested.

Figure 46.42 shows timings for successful transmission from two message buffers.

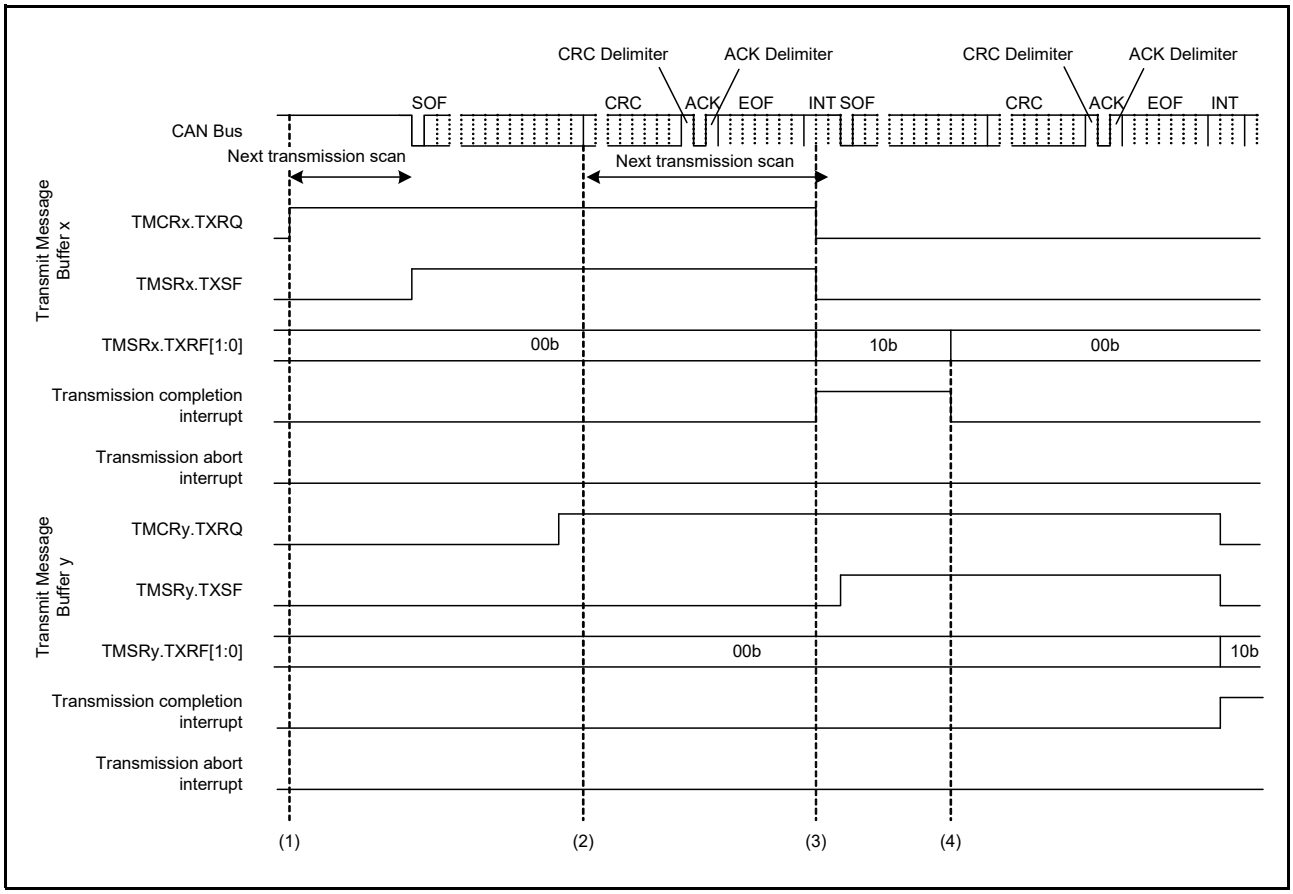


Figure 46.42 Timing of Request and Flag Bits for Successful Transmission

(1) If the TMCRx.TXRQ bit is set to 1 in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.

(3) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 10b and the TMSRx.TXSF flag and the TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the successful transmission interrupt request is generated. To clear the related interrupt line the TMSRx.TXRF[1:0] flags have to be cleared.

(4) Before starting the next transmission, clear the TMSRx.TXRF[1:0] flags. Load the next message in the transmit message buffer and set the TMCRx.TXRQ bit to 1 again.
The TMCRx.TXRQ bit cannot be set to 1 again before TMSRx.TXRF[1:0] flags are cleared.

Note: The setting point of the TMSRx.TXSF flag is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs either during the transmission or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

The Figure 46.43 shows timings for transmission abort for two message buffers.

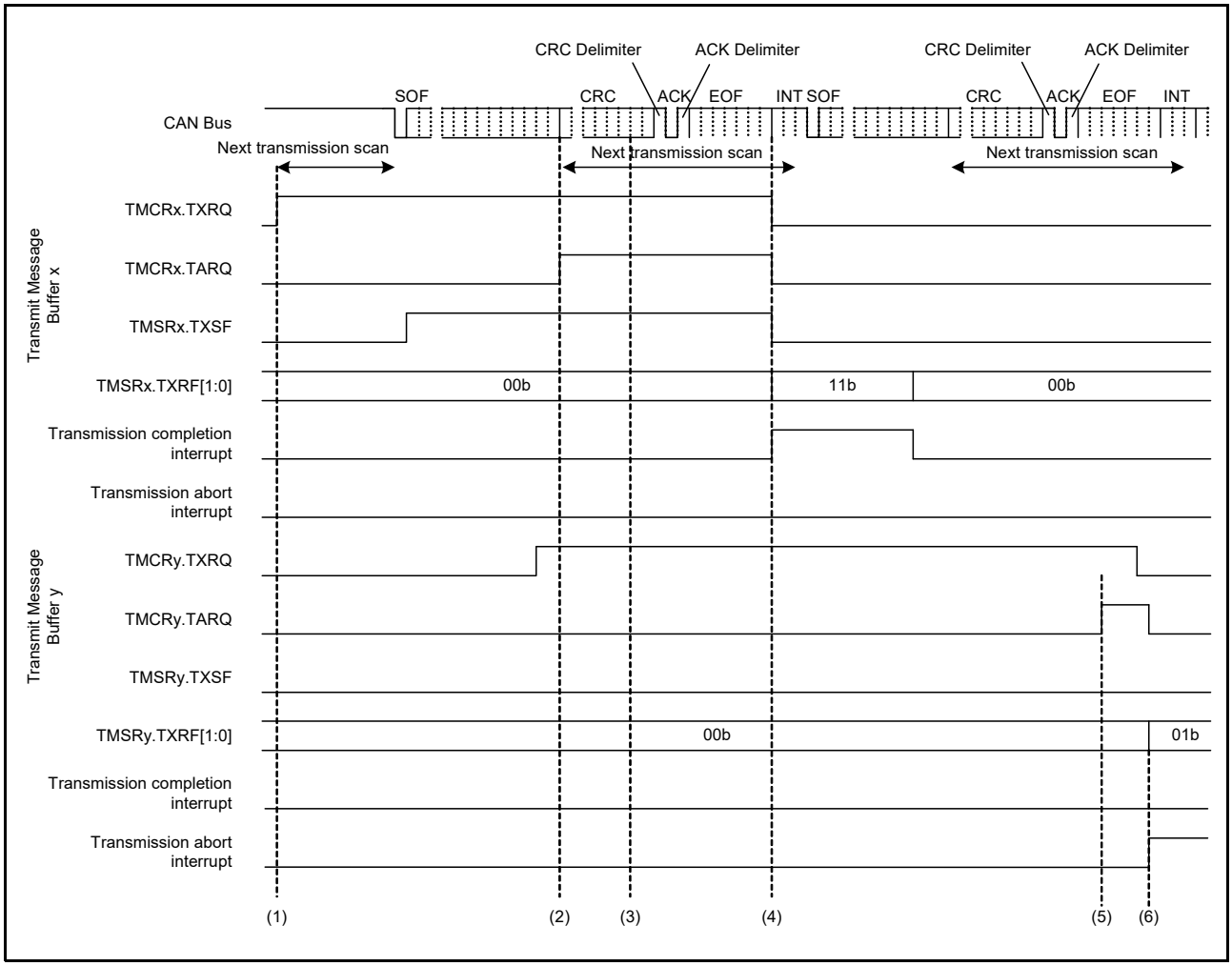


Figure 46.43 Timing of Request and Flag Bits for Transmission Abort

(1) If the TMCRx.TXRQ bit is set in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) If the TMCRx.TARQ bit is set to 1 when the related message buffer is already selected for transmission or currently transmitting, the message will not be aborted, if no error occurs or arbitration is lost.

(3) At the first bit of CRC, the transmission scanning procedure starts for the next transmission. In this example timing chart message buffer y is not selected as next transmit message buffer.

(4) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 11b and the TMSRx.TXSF flag and TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the TMSRx.TXRF[1:0] flags has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (the TMSRy.TXSF flag is not set): if the TMCRy.TARQ bit is set to 1 when the related channel is under transmission scan, the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the TMSRy.TXRF[1:0] flags are set to 01b.
If the message buffer is not transmitting or selected as next transmit message buffer or under transmit scan, the abort is immediately accepted and the corresponding TMSRy.TXRF[1:0] flags are set to 01b.
In addition, the TMCRy.TXRQ and TMCRy.TARQ bits are cleared automatically.

When the CHCR.TAIE bit is set to 1 (transmission abort interrupt enabled), an interrupt is generated for successful transmission abort.
To clear the related interrupt the TMSRy.TXRF[1:0] flags have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs, either during the transmission, or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

46.7.2.3 Message Transmission from FIFO Buffer

The CANFD module has one common FIFO. The common FIFO can be linked to the transmit message buffer by the CFCR0.LTM[1:0] bits if configured in transmit FIFO mode.
When a transmission scan is started and the common FIFO corresponding to that transmit message buffer is enabled, the relevant message in the common FIFO will participate in the transmission scan.
Do not configure the transmit message buffer linked to the common FIFO configured in transmit FIFO mode.

(1) Transmit FIFO Operation

Messages can be written into the transmit FIFO by writing to the common FIFO buffer 0 (CFB0).
When the value 000000FFh is written into the CFPCR0 register, the message count of the FIFO is incremented by 1.
Before writing to the CFPCR0 register, wait until the message has been completely written to the CFB0. If the message count matches the FIFO depth, the CFSR0.FULL flag is set to 1.
The oldest message in the transmit FIFO is included in the scan for transmission.
When a message is successfully transmitted from the transmit FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the CFSR0.EMPTY flag is set to 1.
The interrupt generation conditions for the transmit FIFO buffers can be set by the CFCR0.CFIM bit.
If the CFCR0.CFIM bit is set to 0, interrupt is generated when last message is successfully transmitted from the transmit FIFO buffer. If the CFCR0.CFIM bit is set to 1, interrupt is generated for every successfully transmitted message from the transmit FIFO buffer.
Common FIFO can set interrupt, when CAN frame transmission is completed.
The common FIFO configured in transmit FIFO mode can be disabled by setting the CFCR0.CFE bit to 0. If this bit is set to 0, the CFSR0.EMPTY flag is set to 1 as described below:

- immediately if the message from the transmit FIFO is neither scheduled for the next transmission nor in transmission
- following the transmission completion, the detection of an error on the CAN bus, arbitration lost or transition to CH_HALT or GL_HALT mode if the transmission from the transmit FIFO is already scheduled for transmission or already in transmission.

Note: The common FIFO is considered as disabled after setting the CFCR0.CFE bit to 0 only when the CFSR0.EMPTY flag is set to 1 for the corresponding common FIFO.

If there are other pending messages in the transmit FIFO buffer, they will be lost and the transmission needs to be requested again. Before the CFCR0.CFE bit is set to 1 again ensure that the CFSR0.EMPTY flag is set to 1 and that there is no pending abort request from the transmit FIFO.
When the CFCR0.CFE bit is set to 0, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in Figure 46.44.

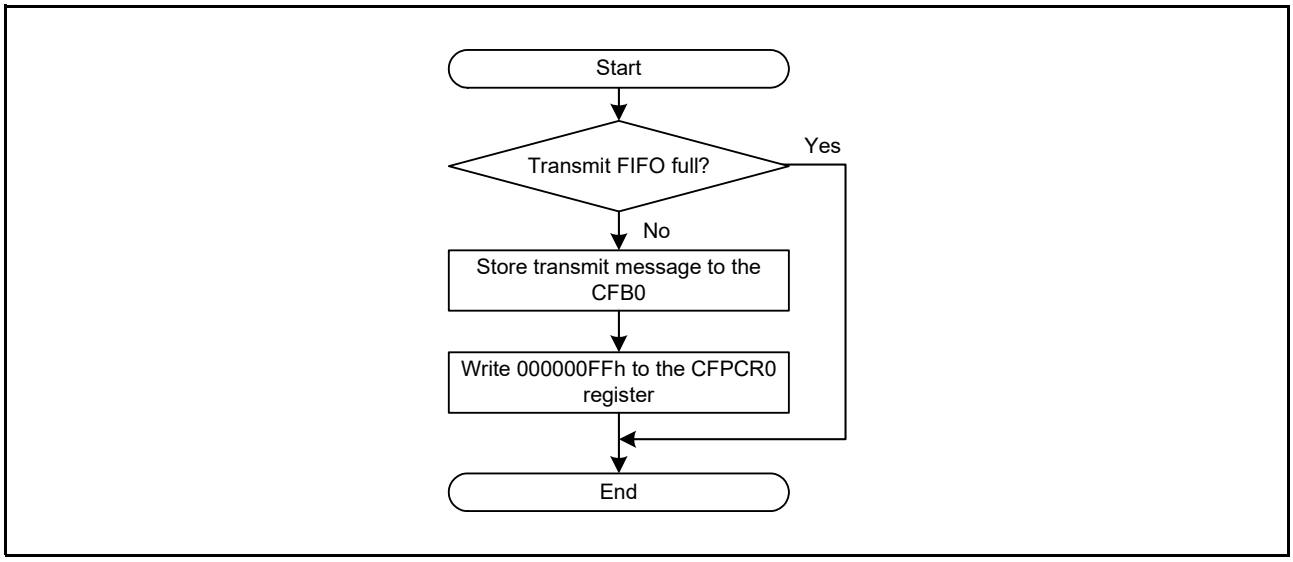


Figure 46.44 Transmit FIFO Transmission Request Procedure

(2) Interval Timer for FIFO Transmission

For the common FIFO in transmit mode it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFCR0.CFE bit is set to 1.
When the common FIFO in transmit mode is enabled, then the first message will be transmitted without considering this interval time.
The interval timer will stop counting when:

- FIFO is disabled by set the CFCR0.CFE bit to 0
- CAN channel is in CH_RESET mode.

The interval time is specified by the value of the CFCR0.TINT[7:0] bits and can be specified from 0 to 255 timer units. The timer unit can be defined based on two different clock sources for the interval timer. Select the value 0 to disable the interval timer for FIFO transmission.
The clock source can be selected by the CFCR0.ITCS bit. For the clock source the CAN bit timing clock of the related channel or a reference clock could be selected.
If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, then the interval timer is stopped for that channel.
If the reference clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.
The reference clock can be used to configure the interval time in fixed time units. It is based on the PCLKA. The GCFG.ITP[15:0] bits define the relation between the PCLKA frequency/period and the reference clock period.
Refer to Table 46.22 for configuration values of the GCFG.ITP[15:0] bits to achieve different reference clock periods based on the PCLKA frequency/period.

Table 46.22 Configuration Example for Interval Timer Prescaler

	Reference clock period		
PCLKA frequency (period)	**1 µs**	**100 µs**	**500 µs**
16 MHz (62.5 ns)	16	1600	8000
20 MHz (50 ns)	20	2000	10000
32 MHz (31.25 ns)	32	3200	16000
50 MHz (20 ns)	50	5000	25000
60 MHz (16.67 ns)	60	6000	30000
80 MHz (12.5 ns)	80	8000	40000
100 MHz (10 ns)	100	10000	50000
120 MHz (8.33 ns)	120	12000	60000
150 MHz (6.67 ns)	150	15000	—

Additionally the reference clock resolution can be specified by the CFCR0.ITR bit.
The interval time is based on the reference clock period multiplied by the configured value (×1 or ×10).
The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time.
The whole range for the separation time from 100 µs to 127 ms can be covered.
The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).
When the interval time has elapsed, the next transmission request is raised by the related transmit FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.
The next message will earliest be sent after this interval time. The Figure 46.45 shows an example timing of the internal processing.

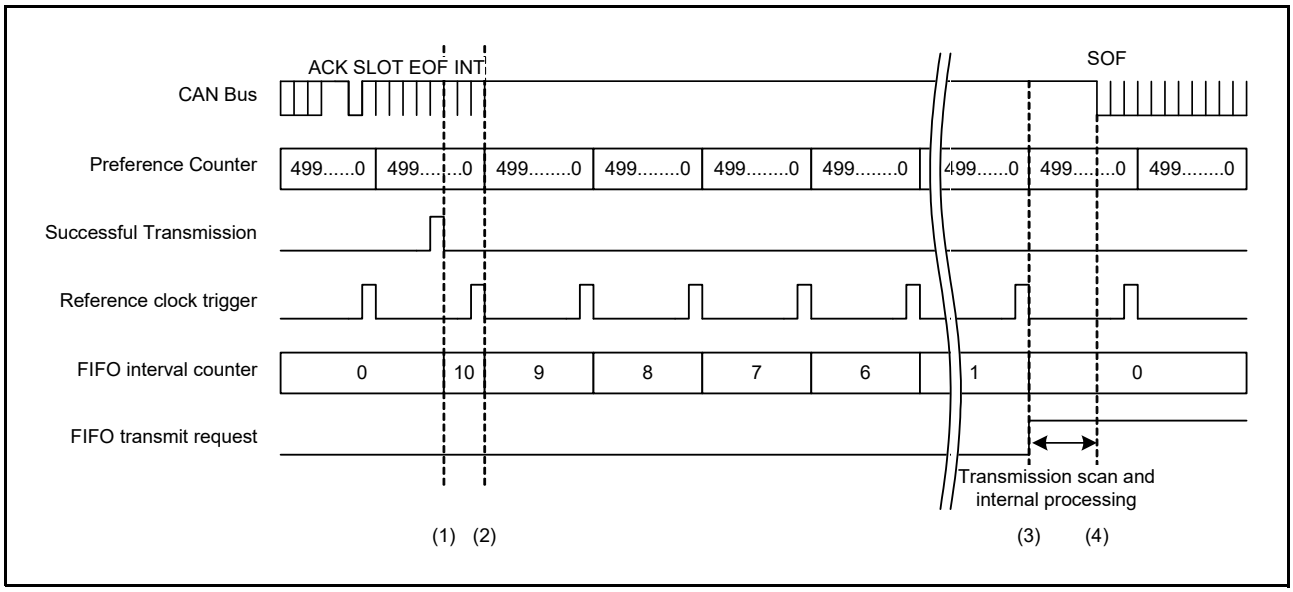


Figure 46.45 Example for Interval Processing Time

The configuration for this timing above is following:

- PCLKA frequency = 50 MHz
- Interval timer prescaler (GCFG.ITP[15:0]) = Divided by 500
- Reference clock due to the settings above = 10 µs
- Common FIFO interval timer clock source selection (CFCR0.ITCS) = 0
- Common FIFO interval timer resolution (CFCR0.ITR) = 0

- Common FIFO transmission interval (CFCR0.TINT[7:0]) = 10 counts
- Theoretical message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore the first interval is counting less or equal to one reference clock interval.
(2) With the next reference clock trigger the FIFO interval timer is decremented.
(3) When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 bit time, between internal FIFO transmit request set (3) (shown above) and actual transmission.
When multi events like reception scan, internal message routing, transmit scan happen, then it could take up to 120 PCLKA cycles.

As shown in Figure 46.45, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure the CFCR0.TINT[7:0] bits to required minimum value + 1.
If transmit message buffers or transmit FIFOs are configured for transmission for the channel the real delay between two messages transmitted from a transmit FIFO can be much longer than specified by the interval time due to higher priority message transmission from these transmit message buffers or transmit FIFOs.
Figure 46.46 shows a block diagram of the FIFO interval timer.

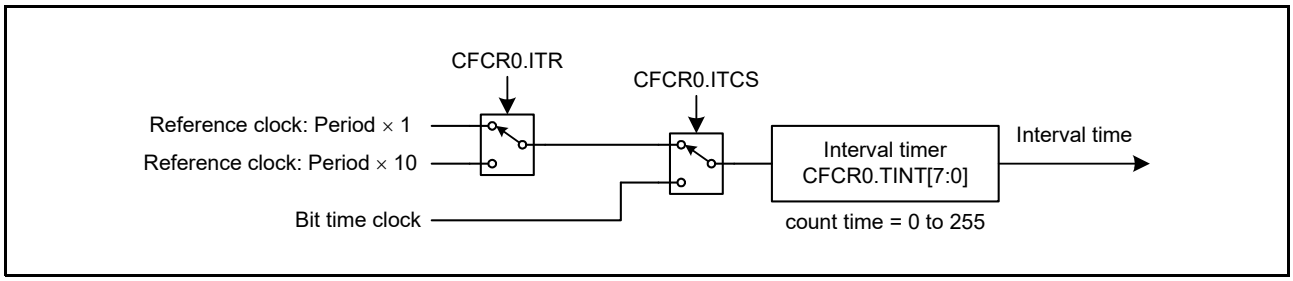


Figure 46.46 Block Diagram of FIFO Interval Timer

46.7.2.4 Transmit Queue

Each enabled transmit queue consists of three or four transmit message buffers, which are accessed via one access window.
The transmit queue 0 (hereafter TXQ0) can be configured with a depth of three to four, and it is using the transmit message buffer 0 as access window. All the message of TXQ0 enter the priority comparison for the transmission, which should be ID priority (GCFG.TPRI = 0).
The registers for TXQ0 are TQCR0, TQSR0, and TQPCR0.
As access window transmit message buffer 0 (TXQ0) is used, refer to related access registers TMB0.HF0, TMB0.HF1, TMB0.HF2, and TMB0.DF0 to TMB0.DF15.
The depth of each TXQ0 buffer can be configured by writing to the TQCR0.QDS[1:0] bits. TXQ0 can be set from TMB0 to TMB3 as a queue buffer at the maximum.
The available options for depth configuration are:

- 10b: 3 messages
- 11b: 4 messages.

Do not access all transmit message buffers except transmit message buffer 0 that configure the transmit queue. Also, do not access the TMCRn registers that correspond to the transmit message buffers that configure the transmit queue.
When writing data to TXQ0, check the status of TXQ0 before writing the data to be transmitted.

The messages stored to the transmit queue access window are internally stored to a free buffer of the transmit queue.
When the TXQ0 buffer is full, do not access the queue anymore. If the transmit data is written when the TXQ0 is full, the transmit data will be overwritten.
The transmit queue can be disabled by setting the TQCR0.TQE bit to 0. If this bit is set to 0, the TQSR0.EMPTY flag is set as described below:

- immediately: if a message from the transmit queue is not scheduled for the next transmission and is not being transmitted
- after the transmission completion, error detection on the CAN bus, loss of arbitration, or transition to CH_HALT or GL_HALT mode: if a message from the transmit queue is scheduled for transmission or is being transmitted.

Note: The transmit queue is disabled only when the TQSR0.EMPTY flag is set to 1 after setting the TQCR0.TQE bit to 0.

If there are other pending messages in the transmit queue, they will be lost, so their transmission needs to be requested again.
Before the TQCR0.TQE bit is set to 1 again, ensure that the TQSR0.EMPTY flag is set to 1 and that there is no pending abort request from the transmit queue.
When the TQE bit is set to 0, all messages in the transmit queue buffers will be lost and no further message can be stored in the transmit queue.
When a message has been stored to the transmit queue, 000000FFh must be written in to the TQPCR0 register. This will set the transmit request automatically and change the internal message buffer pointer to the next free message buffer location of the transmit queue.

Note: If two messages with the same ID are stored in the transmit queue, then the order of transmission of these messages could be different from the order in which they were stored in the transmit queue.
To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the transmit queue.

For the transmit queue a dedicated interrupt can be enabled by setting the TQCR0.TQIE bit.
The interrupt mode can be configured with the TQCR0.TQIM bit either to generate an interrupt for every transmitted message or for the last transmitted message.
The transmit queue transmission request procedure after configuration is shown in Figure 46.47.

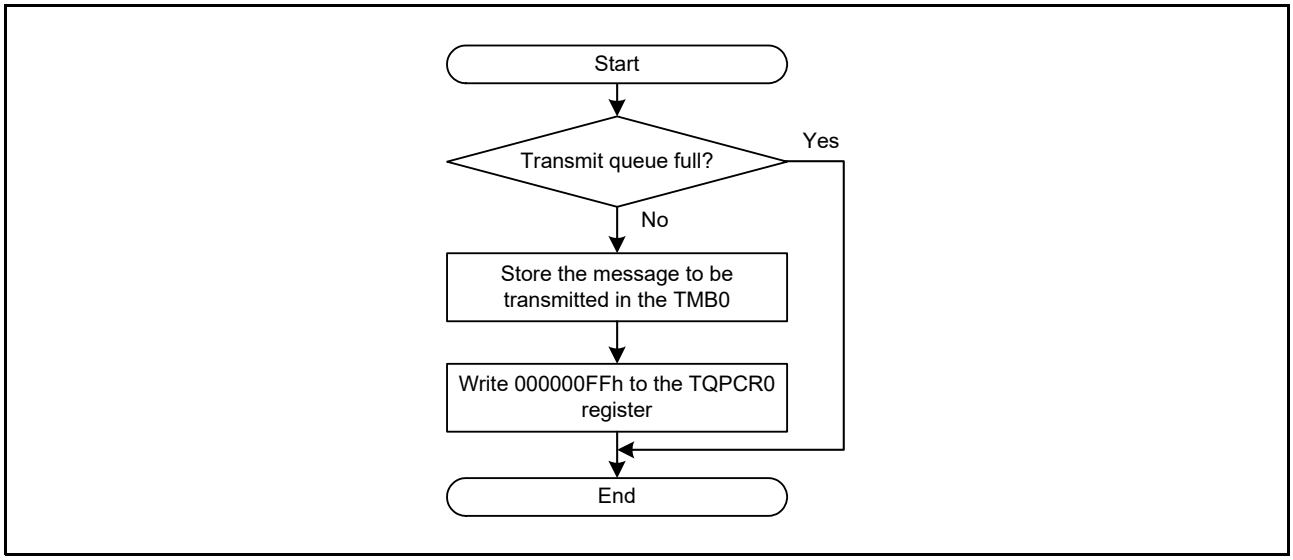


Figure 46.47 Transmit Queue Transmission Request

46.7.2.5 Transmission History

The transmission history function records the information of the successfully transmitted message in the transmission history buffer. The transmission history buffer can store up to eight transmission history entries.
The THCR.THRC bit can select whether to store information only for messages transmitted from transmit FIFO and transmit queue, or also for messages transmitted from the transmit message buffers.
Each transmit message can be individually configured for acceptance to the transmission history by the CFB0.HF0.THENT bit.
The message information is stored to the transmission history buffer of a CAN channel after the message is successfully transmitted on that CAN channel.
Storing to the list is not synchronized with the status of TMSRn.TXRF[1:0] flags.
Due to internal processing, the storage to the list could happen with a delay after the successful transmission indication.
Storing the transmission history data can be recognized by the condition that the THSR.THIF flag is set to 1 when the THCR.THIE bit is set to 1 or when the transmission history counter (THSR.FLVL[3:0]) is increased.
Multiple events could occur, such as reception scans and internal message routing.
Maximum delay time from setting the TMSRn.TXRF[1:0] flags to storing the transmission history data is 70 PCLKA cycles.

The transmission history records following information of the transmitted message:

- Transmitted Buffer type:
  - 001: Transmit message buffer
  - 010: Common FIFO in transmit FIFO mode (hereinafter referred to as transmit FIFO)
  - 100: Transmit queue
- Transmitted Buffer number:
  Transmit message buffer, transmit queue message buffer or transmit message buffer link for the common FIFO from which the transmission occurred. The number depends upon the buffer type, refer to Table 46.23.
- Transmitted pointer:
  Pointer set in header field 2 of transmit message (HF2.PTR[15:0])
- Transmitted timestamp:
  Message timestamp captured at capture point as set by the GFDCFG.TSCPS[1:0] bits
- Transmitted information label:
  Information label set in header field 2 of transmit message (HF2.IFL[1:0])

Table 46.23 Transmission History Buffer Number Entry

Transmitted Buffer Number (THACR0.BN[1:0])	Transmitted Buffer Type (THACR0.BT[2:0])		
	001b	010b	100b
	Transmit Message Buffer	Transmit FIFO	Transmit Queue
00b	TMB0	The value of the BN[1:0] bits corresponds to the setting of the CFCR0.LTM[1:0] bits.	The value of the BN[1:0] bits indicates the number of the transmit message buffer from which the message was transmitted.
01b	TMB1		
10b	TMB2		
11b	TMB3		

The transmitted pointer is used to identify which message of a transmit FIFO or transmit queue has been successfully transmitted because the transmit FIFO or transmit queue number alone is not sufficient.
Therefore, a unique number (pointer) can be attached to each transmit message stored in a transmit FIFO or transmit queue. This unique pointer should be written to the CFB0.HF2.PTR[15:0] bits for a transmit FIFO or to the TMB0.HF2.PTR[15:0] bits of the transmit queue.
When the message is successfully transmitted then this pointer is stored together with the other message related

information to the transmission history and can be read via the transmitted pointer field (PTR[15:0]) in the Transmission History Access Register.
Also for normal transmit message buffers, the TMBn.HF2.PTR[15:0] bits will be stored in the transmission history. An information label is the same.
Read access to the Transmission History Access Register will be done for every single entry.
After reading one entry, 000000FFh has to be written to the THPCR register to be able to access the next entry until transmission history is empty.
Figure 46.48 shows an example flow for processing the transmission history.
The transmission history has dedicated interrupts, which can be configured with the THCR.THIM bit and enabled with the THCR.THIE bit, either to generate an interrupt when the transmission history reached a filling level of 75% or for every new transmission history entry.
Loss of transmission history is indicated by the THSR.LOST flag. Status of this flag is also shown by the GESR.THLDF flag.

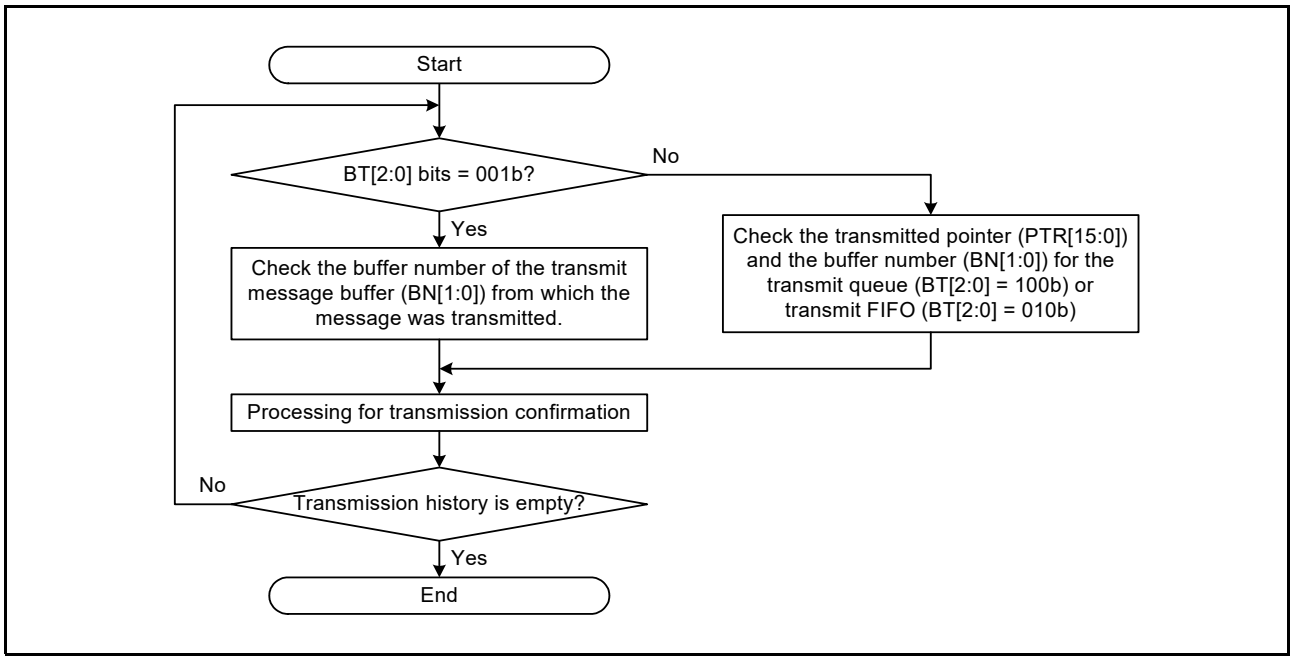


Figure 46.48 Example of Transmission History Processing Flow

46.7.2.6 Transmit Data Padding

If the data length code (DLC) of the transmit message is higher number of data bytes than the buffer size, the data bytes beyond the restricted range will be replaced by bytes with the value of CCh.
This could occur for common FIFO configured in transmit FIFO mode when the DLC of the transmit message is larger than the payload size set in the CFCR0.PLS[2:0] bits.
This can also occur in FD only mode, if the DLC value for a Classical CAN frame is greater than 8.

46.8 ECC Check

Message buffer RAM has ECC function of the 2-bit error detection and 1-bit error detection/correction.*1 The ECC module appends 7-bit ECC data to 32-bit RAM data.

Note 1. The ECC module cannot detect errors of 3 bits or more. In this case, the ECC module detects a 1-bit or 2-bit error, does not detect the error, or corrects the error bit to error data by configuration. If all RAM data is fixed at 0 or 1, it will be detected as a 2-bit ECC error.

46.8.1 ECC Function Setting

Figure 46.49 shows a procedure for ECC function setting.

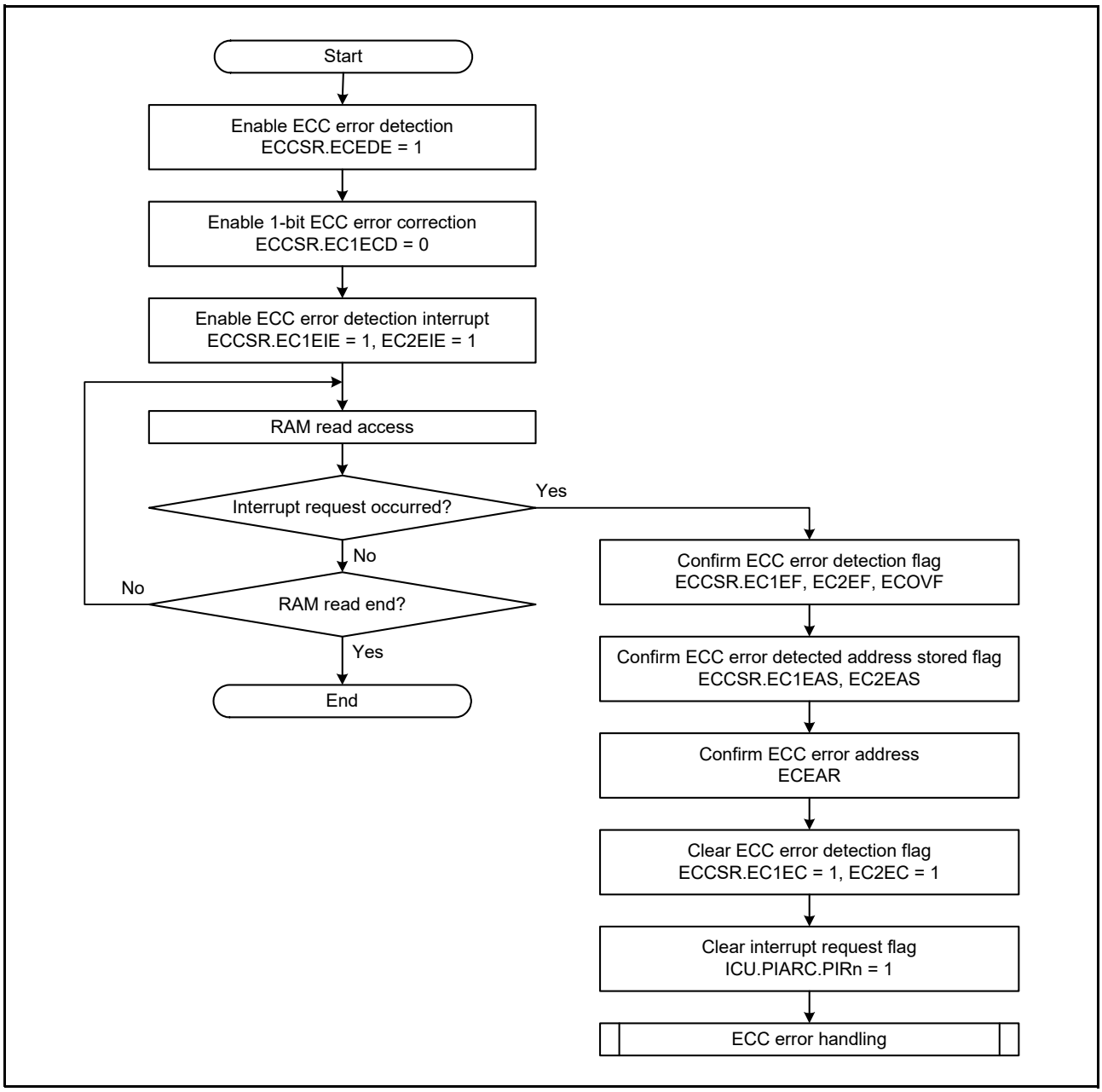


Figure 46.49 Setting Procedure for ECC Function

46.8.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 46.50 shows a procedure for ECC decoder testing.

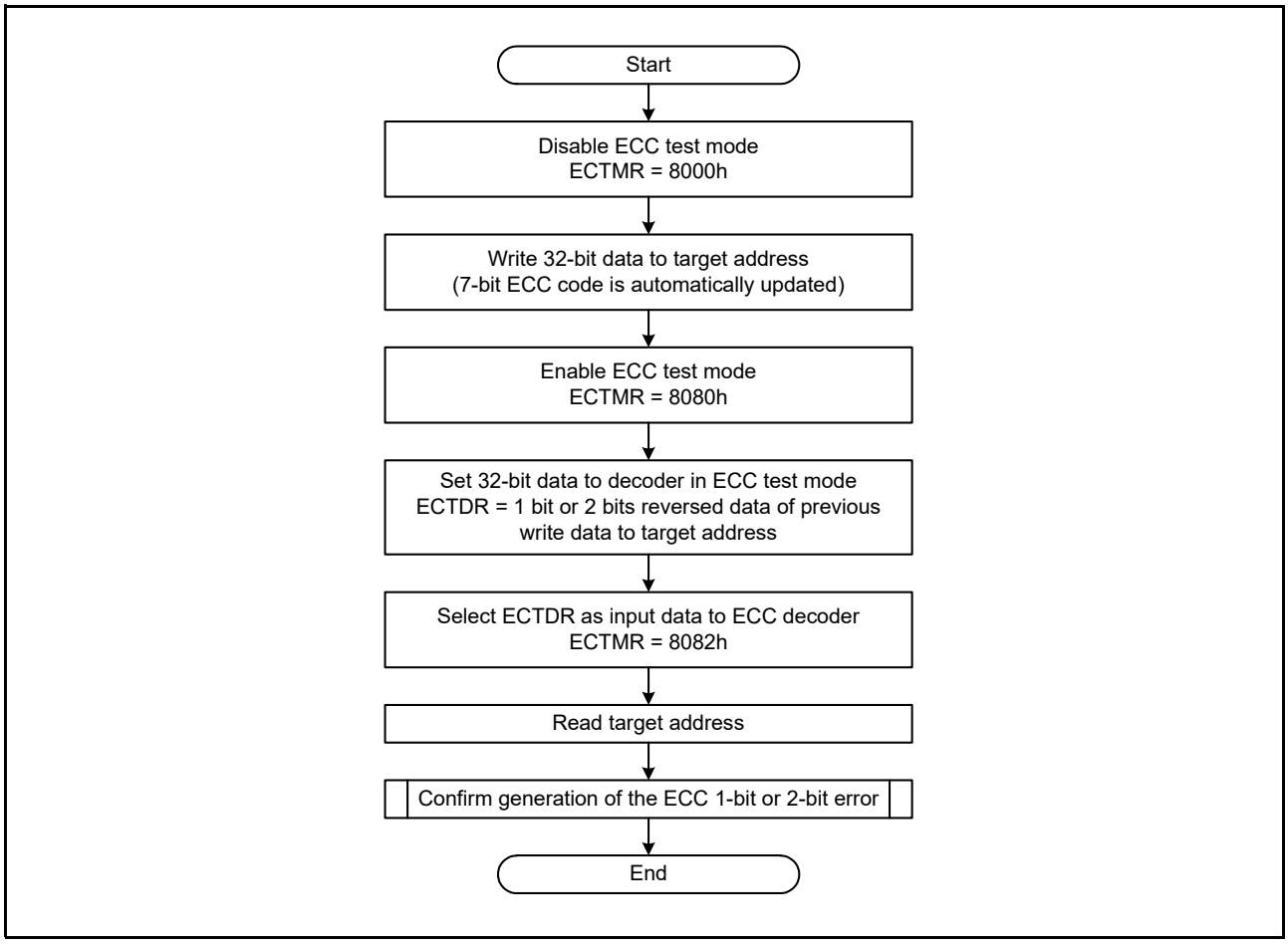


Figure 46.50 Testing Procedure for ECC Decoder

46.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in the test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.
The test modes can be broadly split into two groups:

- Channel specific test modes
- Global test modes

46.9.1 Channel Specific Test Modes

CAN channel can be configured into following test modes:

- Basic test mode
- Listen-only mode
- Self test mode 0 (external loop-back mode)
- Self test mode 1 (internal loop-back mode)
- Restricted operation mode

46.9.1.1 Basic Test Mode

The basic test mode should be used when a particular test setting needs to be enabled other than when in listen-only and self-test modes.

46.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In this mode, the CAN channel can receive valid data frames and valid remote frames. However, it transmits only recessive bits on the CAN bus and is not allowed to transmit. If the CAN engine is required to transmit a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN engine monitors this dominant bit, although the CTX pin remain in recessive state.
This mode can be used for bit rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.
In this mode, it is not permitted to request transmission from any transmit message buffer, transmit queue, or common FIFO.

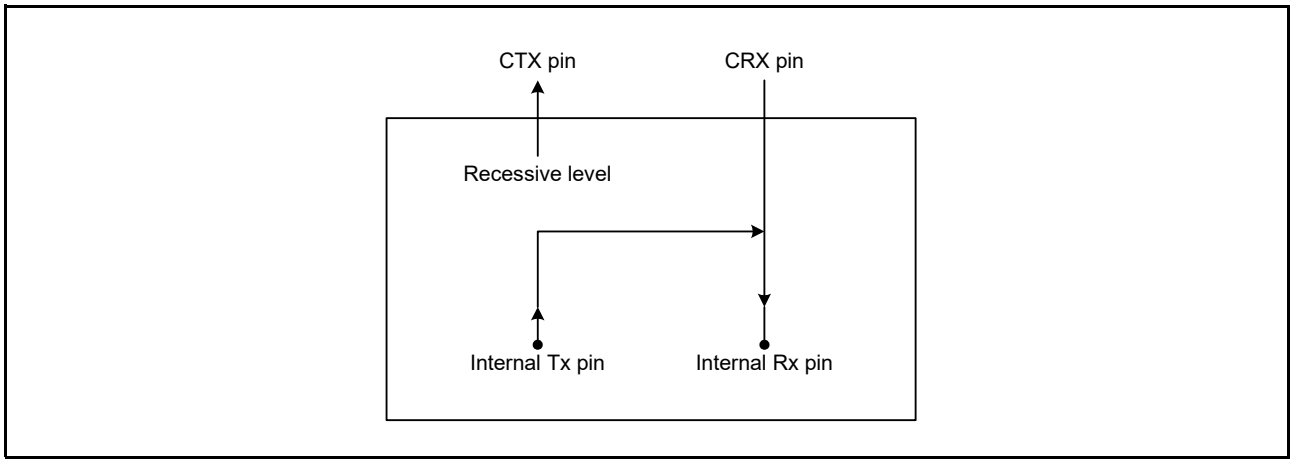


Figure 46.51 Listen-only Mode Configuration

46.9.1.3 Self Test Mode 0 (External Loop-Back Mode)

In self test mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into its receive message buffers.
To be independent of external stimulation the engine generates its own acknowledge bit. This test can be used for CAN transceiver tests.
The CRX and CTX pins should be connected to the transceiver.

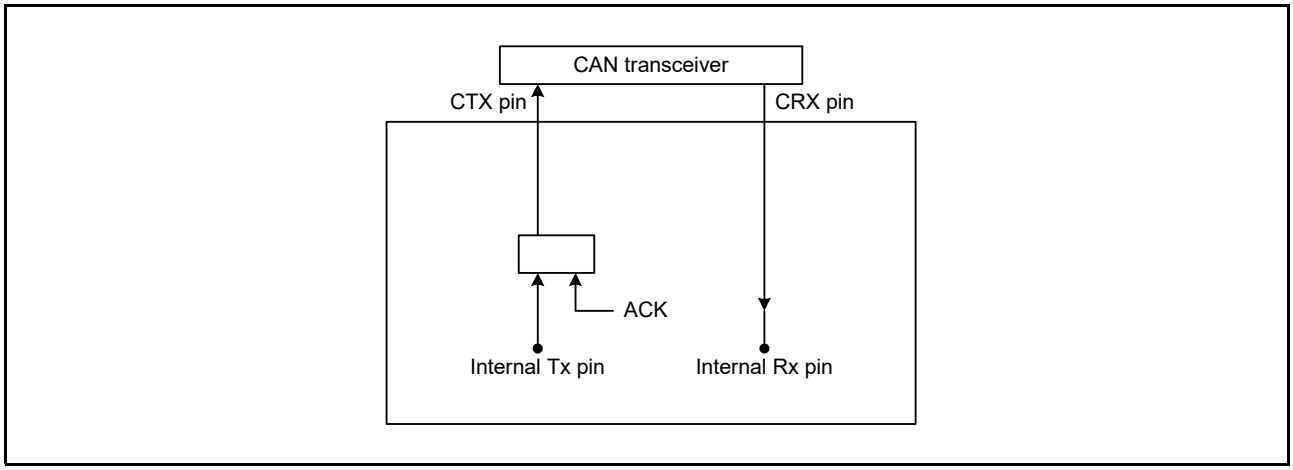


Figure 46.52 Self Test Mode 0 Configuration

46.9.1.4 Self Test Mode 1 (Internal Loop-Back Mode)

In self test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent of external stimulation the CAN engine generates its own acknowledge bit. In this mode the CAN engine performs an internal feedback from internal Tx pin to internal Rx pin. The actual input level of the CRX pin is disregarded by the CAN engine.
The CTX pin is driven high. The CRX and CTX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the Internal CAN bus communication line.

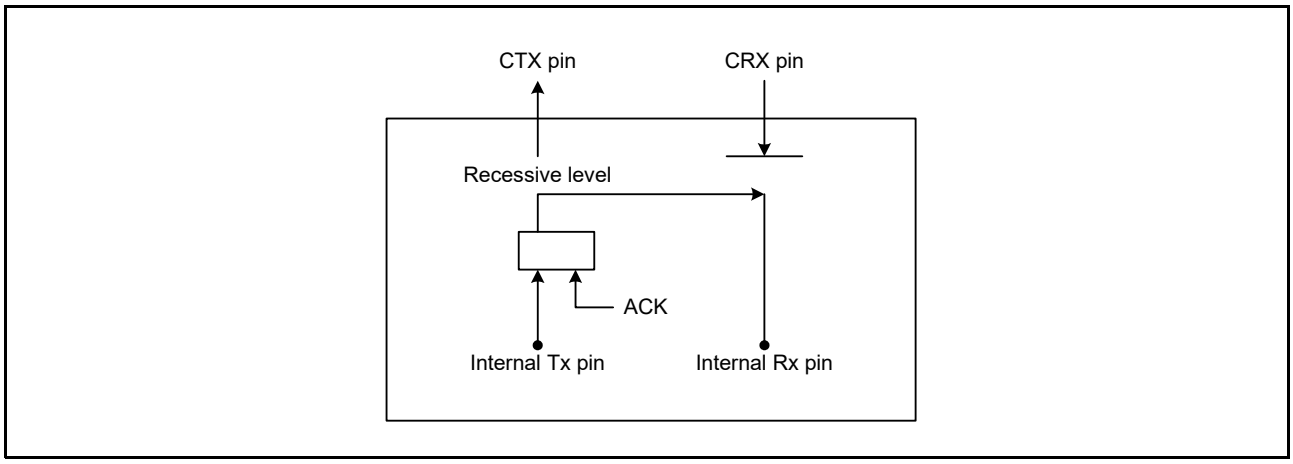


Figure 46.53 Self Test Mode 1 Configuration

46.9.1.5 Restricted Operation Mode

In restricted operation mode the CAN node is able to receive valid data and remote frames generating the acknowledge bit.
Active error and overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.
Moreover the receive error counter (CHSR.REC[7:0]) and transmit error counter (CHSR.TEC[7:0]) are frozen independently of the occurrence of errors.
The mode is specified as in ISO 11898-1; however it is permitted to set any transmit requested.

46.9.2 Global Test Modes

The CANFD module can be configured into following test modes:

- RAM test mode
- Bit flip test

The following test mode is protected by a special software procedure to enable the mode. This software procedure enables the write access to the test mode by specific unlock key, the related unlock key can be seen in the table below:

Table 46.24 Test Mode Unlock Key

Test Mode	Unlock Key 1	Unlock Key 2
RAM Test Mode	00007575h	00008A8Ah

If the software sequence of the two consecutive unlock key write accesses is interrupted by any other write access to the SFR or if incorrect data is written to the Global Test Mode Lock Key Register then the corresponding test mode cannot be set and the sequence should be re-started.
After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism resets and the test mode enable bit cannot be set and then the unlock sequence should be restarted.

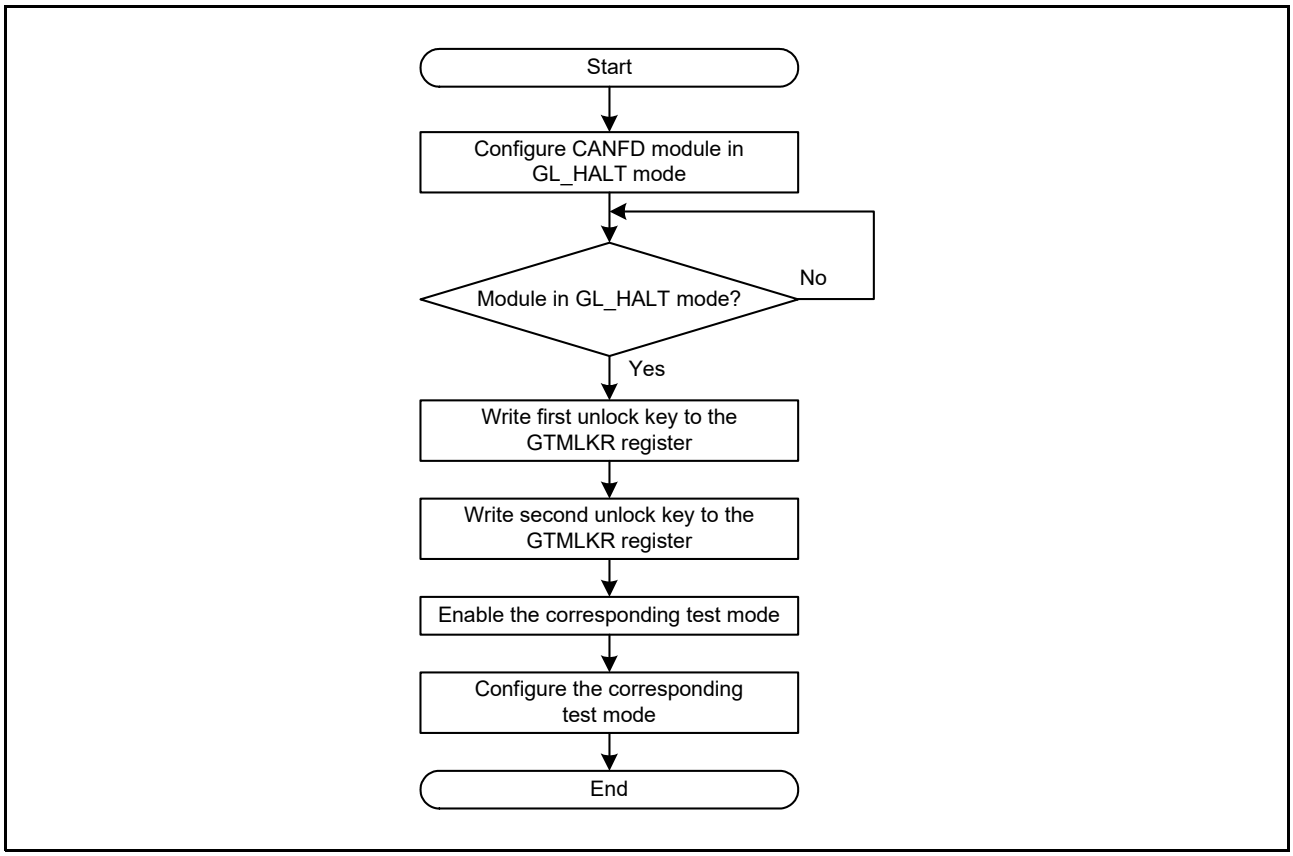


Figure 46.54 Unlock Software Protection Routine

46.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the GTMER.RTME bit when the corresponding lock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after MCU reset. Hence, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages of 256 bytes each. Which can be accessed via RTPARk register (k = 0 to 63).
The page should be selected for read/write access by writing to the GTMCR.RTPS[3:0] bits. Then, data can be read from or written in to the RAM Test Page Access Register.
Figure 46.55 shows the structure of the pages in the RAM when performing a RAM test mode.

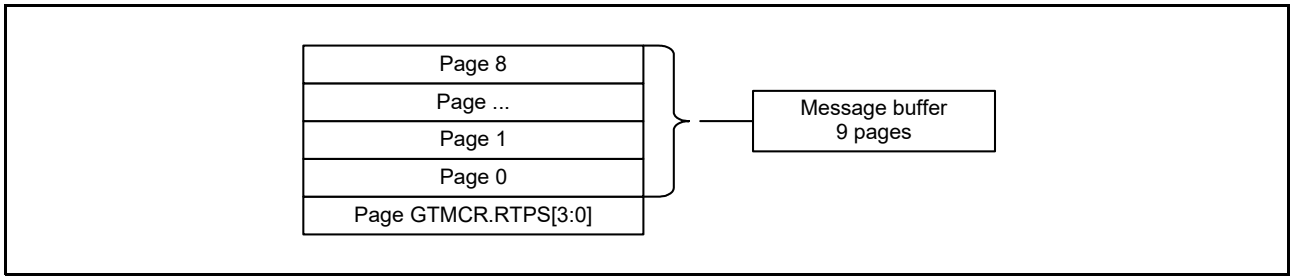


Figure 46.55 RAM Page Structure

The total available RAM size is, 2072 bytes for the message buffer RAM.
Total number of pages for the RAMs and the GTMCR.RTPS[3:0] values are calculated in the following way:

Total number of pages = ceil (total RAM size in bytes / number of bytes per page)

Message buffer RAM:

Total number of pages = ceil (2072 / 256) = 9 pages
GTMCR.RTPS[3:0] = 0 to 8

Figure 46.56 below shows the software flow for RAM test mode.

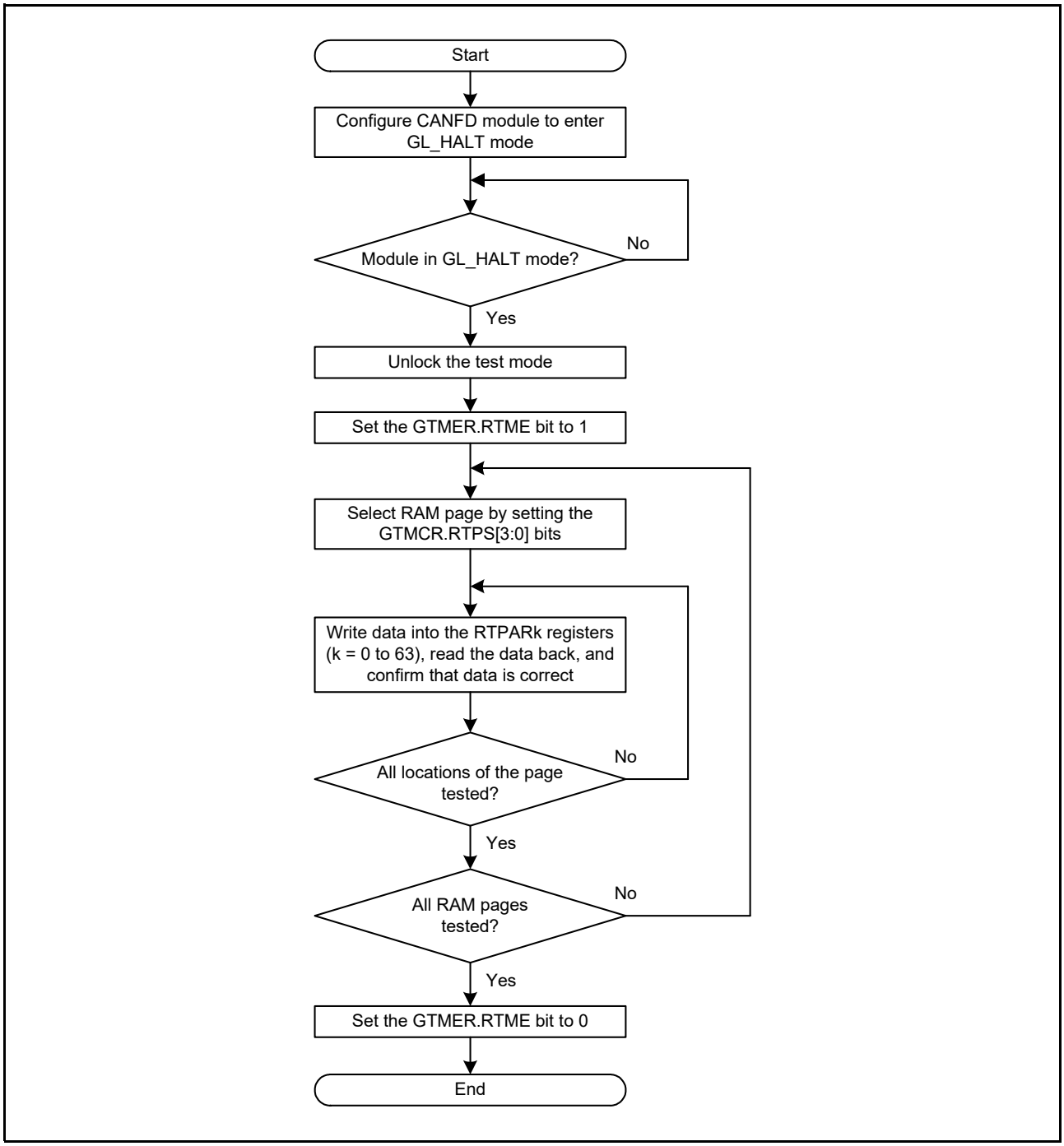


Figure 46.56 RAM Test Mode Software Flow

To exit RAM test mode, set the GTMER.RTME bit to 0.
The GTMER.RTME bit is automatically set to 0 when the CANFD module enters GL_RESET mode from the test mode.

46.9.2.2 Bit Flip Test

Bit flip test can invert the bit (the first bit of ID) of the beginning of the bit stream to receive.
If this function is used by a transmitter node, a bit error or an arbitration lost will occur.
If this function is used by a receiver node, a CRC error or a stuff error will occur.
Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.
The following sequence should be used to perform CRC error testing. In the sequence below CANFD module is the receiver.

(1) Set the CHCR.BFT bit to 1, in order to invert the first bit of the incoming bit stream from transmitter node.
(2) Wait for the can_cherr_int output signal to set to 1.
(3) Read either the CHESR.CRC15[14:0] or FDCRC.CRC21[20:0] bits (depending on the received frame type: Classical CAN or CAN FD). The value should be different from the received CRC value of the reference message from transmitter node.
(4) Check that CHESR.CEDF is 1.
As the CRC generator logic is shared for receive and transmit there is no need to create a separate transmit CRC error test.

46.10 Interrupts and DTC/DMA Requests

46.10.1 CANFD Interrupts

The CANFD module generates several interrupts. The interrupt output, which is connected to the interrupt controller, can be controlled by the corresponding interrupt enable bit.
The status flags will be set independently of these enable bits.
The channel transmit interrupt has an additional status flag register; these status bits will only be set when the corresponding interrupt enables are set.
This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.
The interrupts in the CANFD module can be classified into two groups: global interrupts and channel interrupts.

(1) Global Interrupts

The CANFD module can generate the following three global interrupts.

- Receive FIFO interrupt
- Global error interrupt
- Receive message buffer interrupt

(2) Channel Interrupts

The CAN channel can generate the following three channel interrupts.

- Channel transmit interrupt
  (a) Successful transmission interrupt
  (b) Transmission abort interrupt
  (c) Transmit queue interrupt
  (d) Common FIFO transmission interrupt
  (e) Transmission history interrupt
- Channel error interrupt
- Common FIFO receive interrupt

Table 46.25 lists the interrupt sources for CANFD module.
To clear each interrupt request, clear all flags set to 1 from among the sources for which interrupts are enabled. The interrupt request can also be cleared by setting all corresponding interrupt enable bits to 0.

Table 46.25 Interrupt Sources

Interrupt Name			Interrupt Source Flag	Interrupt Enable Bit	Interrupt Status Flag
Global interrupts	Receive FIFO interrupt (RFRI)		RFSRn.RFIF	RFCRn.RFIE	—
	Global error interrupt (GLEI)		GESR.DEDF GESR.MLDF GESR.THLDF GESR.PODF	GCR.DEIE GCR.MLIE GCR.THLIE GCR.POIE	—
	Receive message buffer interrupt (RMRI)		RMNDR.NDF[n]	RMIER.RMIEn	—
Channel interrupts	Channel transmit interrupt (CHTI)	Transmission successful interrupt*1	TMSRn.TXRF[1]	TMIER0.TMIEn	TISR.TSIF0
		Transmission abort interrupt*1	TMSRn.TXRF[1:0] (TXRF[1:0] = 01b)	CHCR.TAIE	TISR.TAIF0
		Transmit queue interrupt	TQSR0.TQIF	TQCR0.TQIE	TISR.TQIF0
		Common FIFO transmission interrupt	CFSR0.CFTIF	CFCR0.CFTIE	TISR.CFTIF0
		Transmission history interrupt	THSR.THIF	THCR.THIE	TISR.THIF0
	Channel error interrupt (CHEI)		CHESR.BEDF CHESR.EWDF CHESR.EPDF CHESR.BOEDF CHESR.BORDF CHESR.OLDF CHESR.BLDF CHESR.ALDF FDSTS.ECOV FDSTS.SCOV FDSTS.TDCV	CHCR.BEIE CHCR.EWIE CHCR.EPIE CHCR.BOEIE CHCR.BORIE CHCR.OLIE CHCR.BLIE CHCR.ALIE CHCR.ECOVIE CHCR.SCOVIE CHCR.TDCVIE	—
	Common FIFO receive interrupt (CFRI)		CFSR0.CFRIF	CFCR0.CFRIE	—

Note 1. These interrupts are only generated for transmit message buffers that are not part of a valid transmit queue and are not linked to a common FIFO. The common FIFO and the transmit queue have other interrupts respectively.

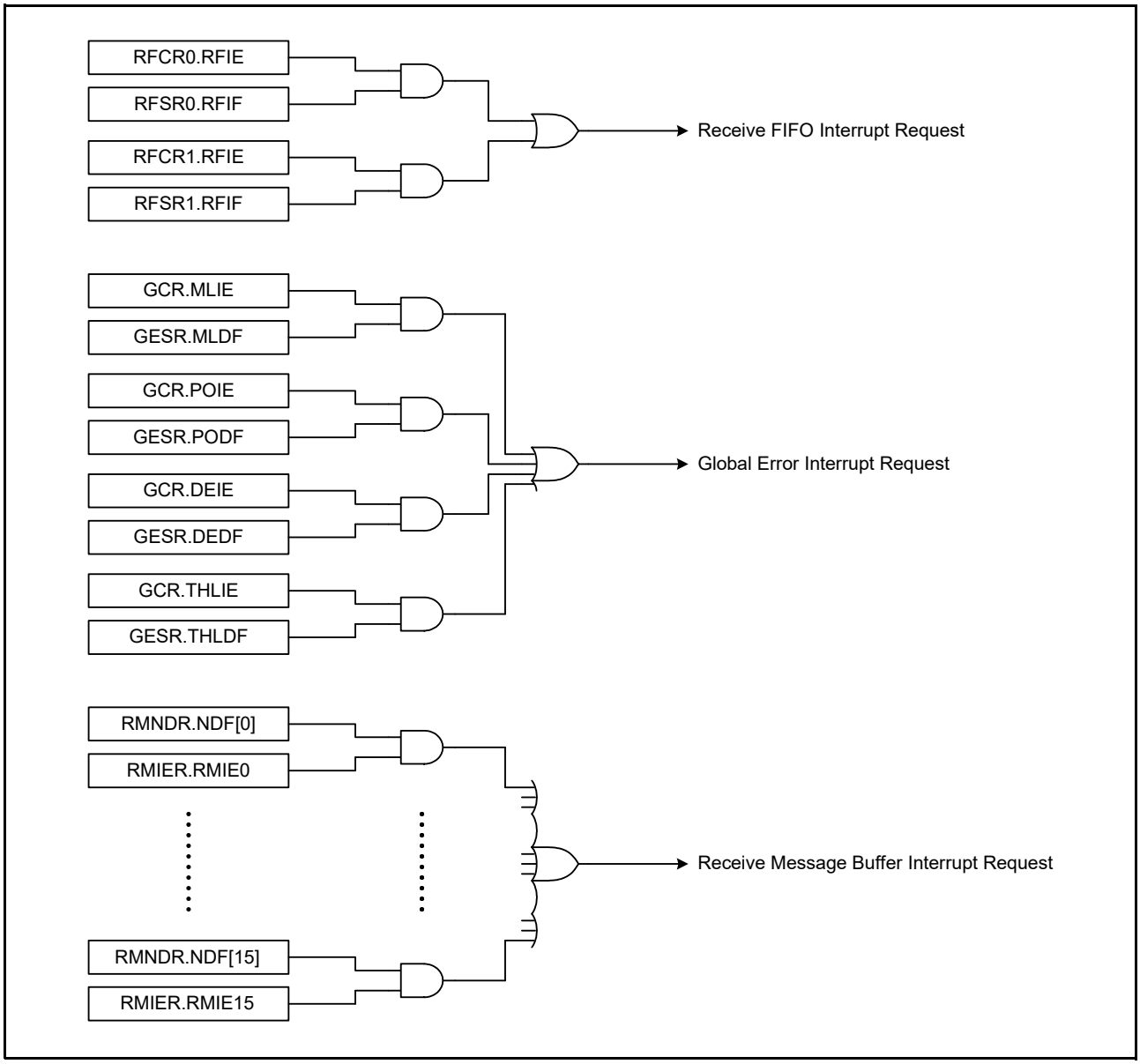


Figure 46.57 Global Interrupt Block Diagram

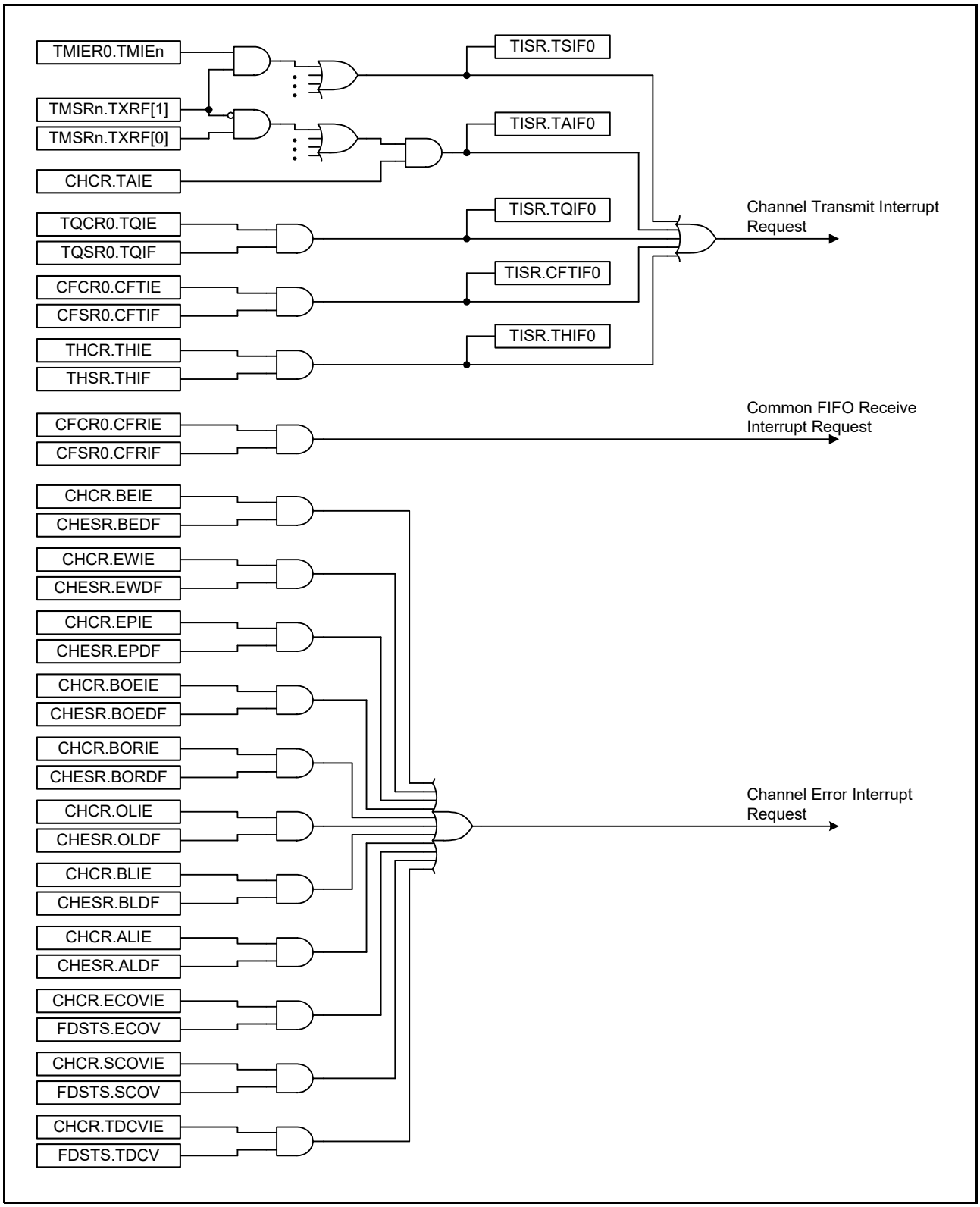


Figure 46.58 Channel Interrupt Block Diagram

46.10.2 ECC Interrupts

The ECC decoder generates an ECC error interrupt (ECCEI). There are three types of interrupt sources for the ECC error interrupt:

- 1-bit ECC error
- 2-bit ECC error
- ECC overflow.

46.10.3 DTC/DMA Transfer Requests

The CANFD module has message buffers that can be read by DTC/DMA transfer:

- Two receive FIFO message buffers
- Common FIFO message buffer

A DTC/DMA transfer request is generated when the DTCR.RFDTE0, RFDTE1, or CFDTE0 bit is set to 1 and the corresponding FIFO is not empty.
For FIFOs with DTC / DMA transfer enabled, disable receive FIFO interrupts (the RFCR0.RFIE, RFCR1.RFIE, or CFCR0.CFRIE bit).

When the data of the specified payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0], or CFCR0.PLS[2:0] bits) is read*1, the FIFO read pointer is automatically updated.
When DTC/DMA transfer is permitted, do not write to the FIFO pointer control register (the RFPCR0, RFPCR1, or CFPCR0 register).

Note 1. The DTC/DMA should read the exact length of the specified data payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0], or CFCR0.PLS[2:0] bits).

The permission for DTC/DMA transfer (the DTCR.RFDTE0, RFDTE1, or CFDTE0 bit) can be set to 1 at any time. Figure 46.59 shows the DTC/DMA transfer configuration flow.

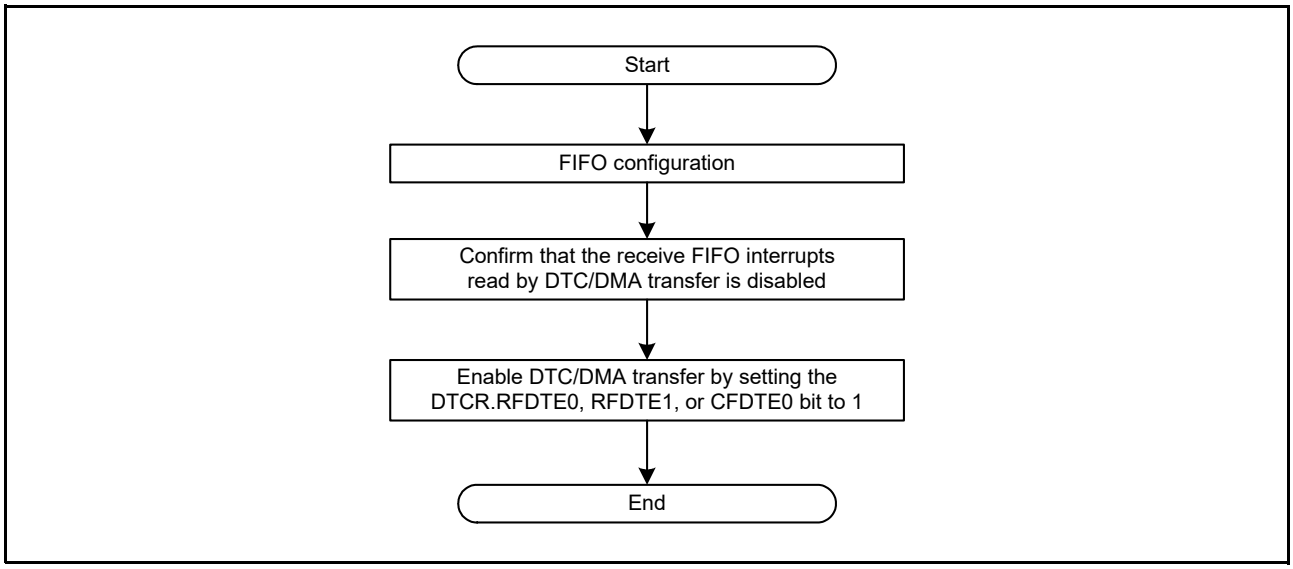


Figure 46.59 DTC/DMA Transfer Enable Flow

To disable the DTC/DMA transfer, set the corresponding DMA transfer enable bit (the DTCR.RFDTE0, RFDTE1, or CFDTE0 bit) to 0. If the disable is made during a DTC/DMA transfer, wait until the ongoing transfer is complete before

performing the following operations. The transfer status can be confirmed by the DTSR.RFDTS0, RFDTS1, or CFDTS0 flag. Figure 46.60 shows the DTC/DMA transfer disable flow.
When the DTC/DMA transfer is disabled then consider what to do with the remaining or new incoming messages to this particular reception FIFOs. When the FIFO is not disabled then reception to the FIFO will continue.

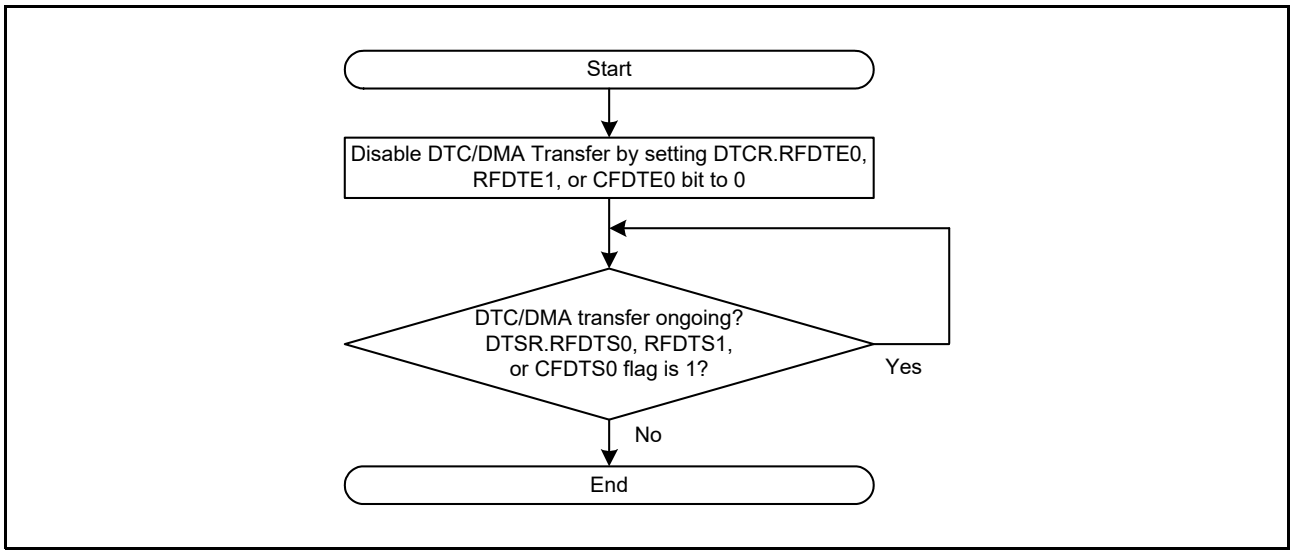


Figure 46.60 DTC/DMA Transfer Disable Flow

46.11 Usage Notes

46.11.1 Setting the Module Stop Function

The CANFD module operation can be enabled or disabled by module stop control register C (MSTPCRC). After a reset, the CANFD module is stopped. The registers of the module become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

46.11.2 Notes on the DLL Clock Settings

Set the DLL clock to be used before releasing the module-stop state. For details on the DLL clock settings, refer to section 9, Clock Generation Circuit.
Release the module-stop state after the oscillation of the DLL clock has been stabilized.

46.11.3 Note on Configuration of Receive Message Buffers and FIFO Buffers

The maximum memory size available for receive message buffers and FIFO buffers is 1216 bytes.
For example, if all payload sizes are set to 8 bytes, each message size becomes 20 bytes, so the total number of messages must be 60 or less. If 16 receive message buffers are reserved, the total FIFO depth must be 44 or less.
Similarly, if all payload sizes are set to 64 bytes, each message size becomes 76 bytes, so the total number of messages must be 16 or less. If all FIFO depths are set to 4 messages, up to four receive message buffers can be used.
Operation is not guaranteed if the setting exceeds 1216 bytes.

46.11.4 Transition to Software Standby Mode

To reduce power consumption, power to the message buffer RAM is turned off in software standby mode. Therefore, the respective sets of operations listed below must be performed before a transition to and after a return from software standby mode.

(1) Before a Transition to Software Standby Mode

Before a transition to software standby mode, perform the following operations.

1. Put the CANFD module into GL_SLEEP mode.
2. Read the GTMLKR register and verify that its value is 00000000h.
3. Set the ECCSR.EC1EC and ECCSR.EC2EC bits to 1 to clear the ECC error detection flags.
4. Set the ECCSR.ECEDE bit to 0 to disable ECC error detection.

(2) After a Return from Software Standby Mode 1

After a return from software standby mode 1, the CANFD module is not reset, meaning that the RAM is not initialized. Use RAM test mode to initialize the RAM. After having initialized the RAM, initialize the CANFD module by following the standard procedure. For details on RAM test mode, refer to section 46.9.2.1, RAM Test Mode.

(3) After a Return from Software Standby Mode 2

After a return from software standby mode 2, the CANFD module is reset. Initialize the CANFD module through the same procedure as when the MCU is released from the reset state.

47. Serial Peripheral Interface (RSPIA)

This MCU incorporates three channels of serial peripheral interface (RSPIA0, RSPIA1, RSPIA2).
The RSPIA supports high-speed, full-duplex or simplex synchronous serial communications with multiple processors and peripheral devices.

47.1 Overview

Table 47.1 lists the specifications of the RSPIA and Figure 47.1 shows a block diagram of the RSPIA.

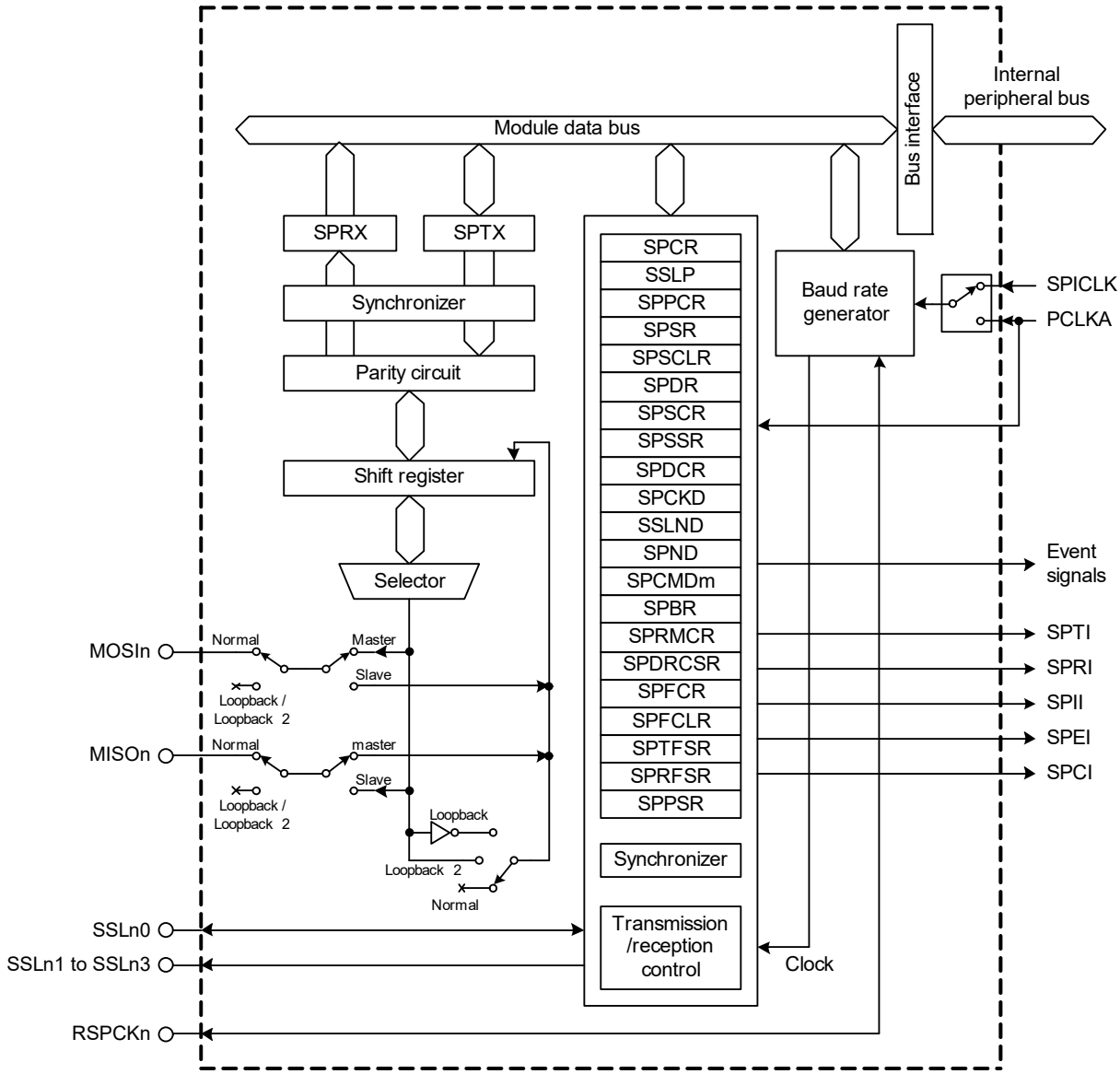
Table 47.1 RSPIA Specifications (1/2)

Item	Description
Transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Full-duplex or simplex (transmit-only or receive-only) communications can be selected. • Serial communication is possible in master mode and slave mode. • Polarity of RSPCK can be selected. • Phase of RSPCK can be selected.
Data format	• MSB first/LSB first selectable • Transfer bit length can be changed to 4 through 32 bits. • Transmit buffer size/receive buffer size: 32 bits × 4 stages FIFO • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). • Byte swapping of transmit and receive data is selectable. • Logic level of transmit and receive data can be inverted.
Operating clock (TCLK)	Selectable from peripheral module clock (PCLKA) or SPI clock (SPICLK)
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing TCLK. The division ratio ranges from divided by 2 to divided by 4096. • In slave mode, an external input clock is used as a serial clock. The maximum frequency of RSPCK is that of TCLK divided by 2 (width at high level: 1 cycle of TCLK; width at low level: 1 cycle of TCLK).
Buffer configuration	Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection • Receive data ready detection
SSL control function	[Motorola SPI] • Four SSL signals per RSPIA channel (SSLn0 to SSLn3) • In single-master mode, the SSLn0 to SSLn3 signals are output. • In multi-master mode: The SSLn0 signal is input and the SSLn1 to SSLn3 signals are output or Hi-Z. • In slave mode: The SSLn0 signal is input and the SSLn1 to SSLn3 signals are Hi-Z (not used). • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity

Table 47.1 RSPIA Specifications (2/2)

Item	Description
SSL control function	[TI SSP] • Four SSL signals per RSPIA channel (SSLn0 to SSLn3) • In single-master mode, the SSLn0 to SSLn3 signals are output. • In multi-master mode: The SSLn0 signal is input and the SSLn1 to SSLn3 signals are output or Hi-Z. • In slave mode: The SSLn0 signal is input and the SSLn1 to SSLn3 signals are Hi-Z (not used). • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to OE output negation (OE negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Communication Protocol	• Motorola SPI • TI SSP (synchronous serial protocol)
Control in master transfer	[Motorola SPI] • A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced. [TI SSP] • A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay (OE negation delay), and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced.
Interrupt sources	Five interrupt sources • Receive buffer full/receive data ready interrupt • Transmit buffer empty interrupt • Communication end interrupt • Error interrupt (mode fault error, underrun error, overrun error, parity error, or receive data ready) • Idle interrupt
Event link function	Five event types can be output to the event link controller. • Receive buffer full/receive data ready event signal • Transmit buffer empty event signal • Mode fault error, underrun error, overrun error, parity error, or receive data ready event signal • Idle event signal • Communication end event signal
Others	• Function for disabling (initializing) the RSPIA • Loopback mode • SPE bit status polling function
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attribute can be set for each channel.*1

Note 1. When each channel is set to the secure attribute, registers in the channel are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each channel is set to the non-secure attribute, registers in the channel are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.



SPCR: RSPI control register
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPSCLR: RSPI status clear register
SPDR: RSPI data register
SPSCR: RSPI sequence control register
SPSSR: RSPI sequence status register
SPDCR: RSPI data control register
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negation delay register
SPND: RSPI next-access delay register
SPCMDm: RSPI command register m (m = 0 to 7)
SPBR: RSPI bit rate register
SPRMCR: RSPI receive-only mode control register
SPDRCSR: RSPI receive data ready detect condition setting register
SPFCR: RSPI FIFO control register
SPFCLR: RSPI FIFO clear register
SPTFSR: RSPI transmit FIFO status register
SPRFSR: RSPI receive FIFO status register
SPPSR: RSPI polling status register

SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer
SPTI: Transmit interrupt
SPRI: Receive interrupt
SPII: Idle interrupt
SPEI: Error interrupt
SPCI: Communication end interrupt

Figure 47.1 RSPIA Block Diagram

Table 47.2 lists the I/O pins used in the RSPIA.
The RSPIA automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the RSPIA is a single master and as an input when the RSPIA is a multi-master or a slave. Pins RSPCKn, MOSIn, and MISOn are automatically set as inputs or outputs according to the setting of master or slave, the SPI operation (4-wire method) or clock synchronous operation (3-wire method), and the level input on the SSLn0 pin. Refer to section 47.3.2, Controlling RSPI Pins for details.

Table 47.2 RSPIA Pin Configuration

Channel	Pin Name	I/O	Function
RSPIA0	RSPCK0	I/O	Clock I/O
	MOSI0	I/O	Master transmit data I/O
	MISO0	I/O	Slave transmit data I/O
	SSL00	I/O	Slave selection I/O
	SSL01	Output	Slave selection output
	SSL02	Output	Slave selection output
	SSL03	Output	Slave selection output
RSPIA1	RSPCK1	I/O	Clock I/O
	MOSI1	I/O	Master transmit data I/O
	MISO1	I/O	Slave transmit data I/O
	SSL10	I/O	Slave selection I/O
	SSL11	Output	Slave selection output
	SSL12	Output	Slave selection output
	SSL13	Output	Slave selection output
RSPIA2	RSPCK2	I/O	Clock I/O
	MOSI2	I/O	Master transmit data I/O
	MISO2	I/O	Slave transmit data I/O
	SSL20	I/O	Slave selection I/O
	SSL21	Output	Slave selection output
	SSL22	Output	Slave selection output
	SSL23	Output	Slave selection output

47.2 Register Descriptions

47.2.1 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPIA0.SPDR 8745 C000h, RSPIA1.SPDR 8745 C100h, RSPIA2.SPDR 8745 C200h, RSPIA0_NS.SPDR 0745 C000h, RSPIA1_NS.SPDR 0745 C100h, RSPIA2_NS.SPDR 0745 C200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): RSPIA0.SPDR 8745 C000h, RSPIA1.SPDR 8745 C100h, RSPIA2.SPDR 8745 C200h, RSPIA0_NS.SPDR 0745 C000h, RSPIA1_NS.SPDR 0745 C100h, RSPIA2_NS.SPDR 0745 C200h

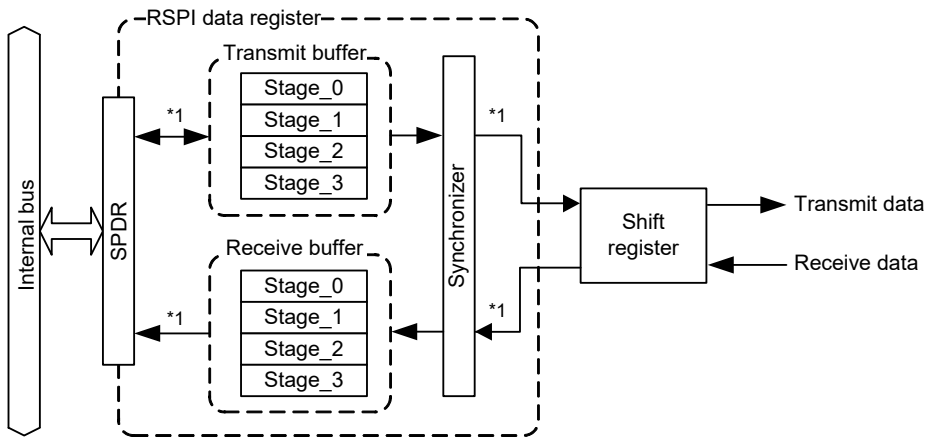
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	L															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): RSPIA0.SPDR 8745 C000h, RSPIA1.SPDR 8745 C100h, RSPIA2.SPDR 8745 C200h, RSPIA0_NS.SPDR 0745 C000h, RSPIA1_NS.SPDR 0745 C100h, RSPIA2_NS.SPDR 0745 C200h

	b7	b6	b5	b4	b3	b2	b1	b0
	LL							
Value after reset:	0	0	0	0	0	0	0	0

The SPDR register is the interface with the buffers that hold data for transmission and reception by the RSPI. The transmit buffer and receive buffer are independent. Figure 47.2 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 47.2 Configuration of SPDR Register

The transmit and receive buffers each have 32 bits × 4 stages of FIFO. The eight stages of FIFO are all mapped to the single address of the SPDR register. Data written to the SPDR register are written to a transmit-buffer stage (SPTXn, n = 0 to 3) and then transmitted from the buffer. The receive buffer (SPRXn, n = 0 to 3) holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

(1) Bus Interface

The SPDR register is the interface with 32-bit wide transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3), each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for the SPDR register.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end.
Operations involved in writing to and reading from the SPDR register are described below.

(a) Writing

The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to the SPDR register. Figure 47.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

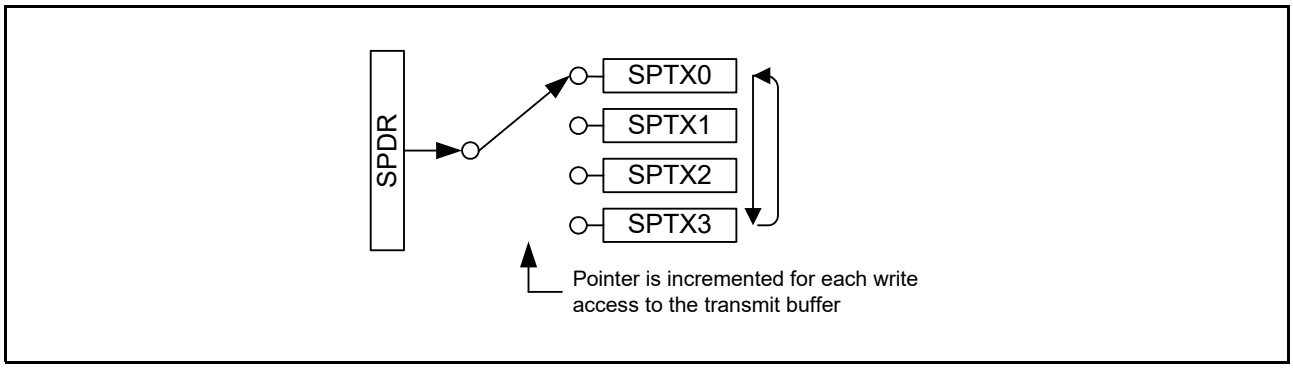


Figure 47.3 Configuration of SPDR Register (Writing)

Sequence of switching the pointer among SPTX0 to SPTX3:
SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames + 1 set by the transmission FIFO threshold setting bits of RSPI FIFO control register (SPFCR.TTRG[1:0]). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stages in the transmit FIFO does not update the buffer value.

(b) Reading

The SPDR register can be read to read the value of a receive buffer (SPRXn, n = 0 to 3) or a transmit buffer (SPTXn, n = 0 to 3). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.
The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.
Figure 47.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

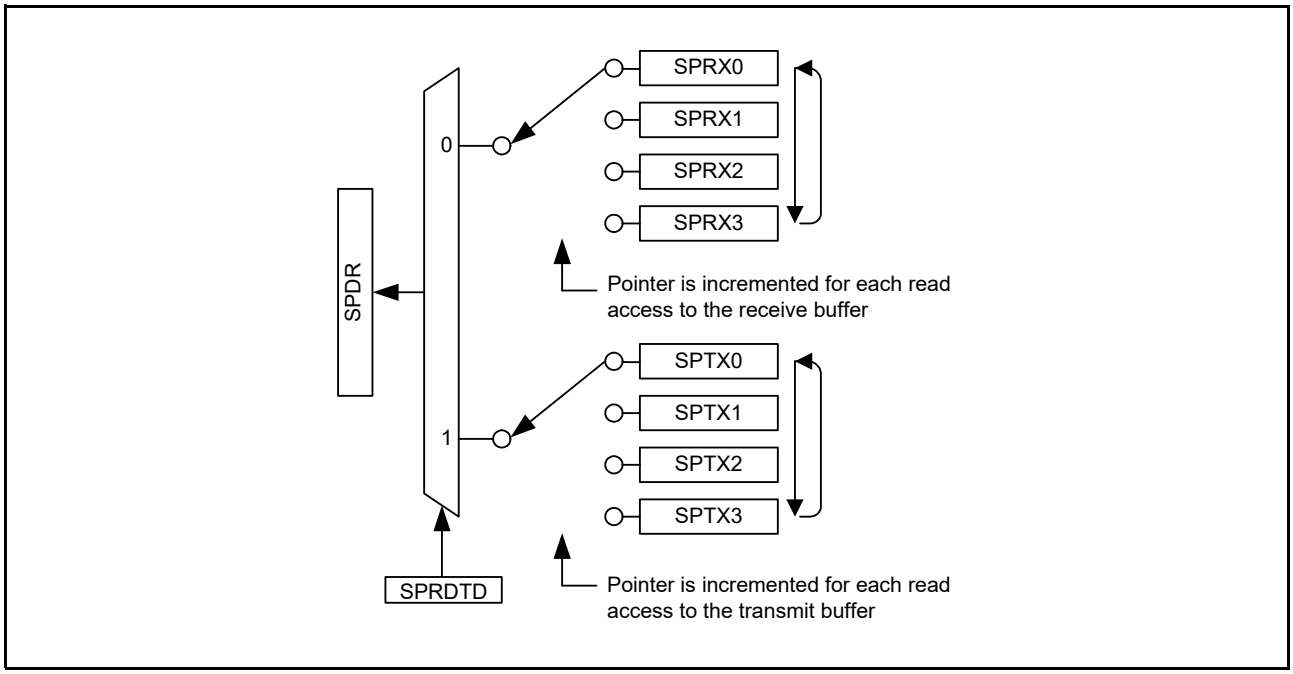


Figure 47.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.
The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer.
The transmit buffer read pointer is updated when writing to the SPDR register, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to the SPDR register is read.

47.2.2 RSPI Clock Delay Register (SPCKD)

Address(es): RSPIA0.SPCKD 8745 C004h, RSPIA1.SPCKD 8745 C104h, RSPIA2.SPCKD 8745 C204h, RSPIA0_NS.SPCKD 0745 C004h, RSPIA1_NS.SPCKD 0745 C104h, RSPIA2_NS.SPCKD 0745 C204h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SPCKD register sets a period from the beginning of SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operation is not guaranteed.

[TI SSP]
The SPCKD register sets a period from the beginning of SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1, and also sets a period until the SSLni signal is negated. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operation is not guaranteed.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPIA in slave mode, set the SCKDL[2:0] bits to 000b.

47.2.3 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPIA0.SSLND 8745 C005h, RSPIA1.SSLND 8745 C105h, RSPIA2.SSLND 8745 C205h, RSPIA0_NS.SSLND 0745 C005h, RSPIA1_NS.SSLND 0745 C105h, RSPIA2_NS.SSLND 0745 C205h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	[In master mode] b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK [In TI SSP slave mode] b2 b0 0 0 0: 1 TCLK 0 0 1: 2 TCLK 0 1 0: 3 TCLK 0 1 1: 4 TCLK 1 0 0: 5 TCLK 1 0 1: 6 TCLK 1 1 0: 7 TCLK 1 1 1: 8 TCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the RSPIA in master mode when the SPCMDm.SLNDEN bit is 1. If the value of the SSLND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operation is not guaranteed.

[TI SSP]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in master mode, or from the detection of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in slave mode. If the value of the SSLND register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

[Motorola SPI]
The SLNDL[2:0] bits set an SSL negation delay value when the SPCMDm.SLNDEN bit is 1.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDL[2:0] bits set an OE negation delay value when the SPCMDm.SLNDEN bit is 1.

[Clock synchronous]
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

47.2.4 RSPI Next-Access Delay Register (SPND)

Address(es): RSPIA0.SPND 8745 C006h, RSPIA1.SPND 8745 C106h, RSPIA2.SPND 8745 C206h, RSPIA0_NS.SPND 0745 C006h, RSPIA1_NS.SPND 0745 C106h, RSPIA2_NS.SPND 0745 C206h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 5 TCLK 0 0 1: 2 RSPCK + 5 TCLK 0 1 0: 3 RSPCK + 5 TCLK 0 1 1: 4 RSPCK + 5 TCLK 1 0 0: 5 RSPCK + 5 TCLK 1 0 1: 6 RSPCK + 5 TCLK 1 1 0: 7 RSPCK + 5 TCLK 1 1 1: 8 RSPCK + 5 TCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPND register sets a non-active period (next-access delay) of the SSLni signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the value of the SPND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operation is not guaranteed.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPIA in slave mode, set the SPNDL[2:0] bits to 000b.

47.2.5 RSPI Control Register (SPCR)

Address(es): RSPIA0.SPCR 8745 C008h, RSPIA1.SPCR 8745 C108h, RSPIA2.SPCR 8745 C208h, RSPIA0_NS.SPCR 0745 C008h, RSPIA1_NS.SPCR 0745 C108h, RSPIA2_NS.SPCR 0745 C208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNDIS	MSTR	CMMD[1:0]		—	—	FRFS	SPMS	—	—	SPCIE	SPTIE	RDRIS	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MODFEN	SCKDDIS	SCKASE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	SPPE	Parity Enable	0: Does not add a parity bit to transmit data and does not check the parity bit of received data 1: Adds a parity bit to transmit data and checks the parity bit of received data	R/W
b9	SPOE	Parity Mode	0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R
b11	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b12	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b13	SCKDDIS	RSPCK Delay Between Frames Disable	0: Inter-frame delay is inserted during burst transfer. 1: Inter-frame delay is not inserted during burst transfer.	R/W
b14	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	SPEIE	Error Interrupt Enable	0: Disables the error interrupt 1: Enables the error interrupt	R/W
b17	SPRIE	Receive Buffer Full Interrupt Enable	0: Disables the receive buffer full interrupt 1: Enables the receive buffer full interrupt	R/W
b18	SPIIE	Idle Interrupt Enable	0: Disables the idle interrupt 1: Enables the idle interrupt	R/W
b19	RDRIS	Receive Data Ready Interrupt Select	0: Receive buffer full interrupt 1: Error interrupt	R/W
b20	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the transmit buffer empty interrupt 1: Enables the transmit buffer empty interrupt	R/W
b21	SPCIE	Communication End Interrupt Enable	0: Disables the communication end interrupt 1: Enables the communication end interrupt	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b25	FRFS	Frame Format Select*1	0: Motorola SPI 1: TI SSP	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b29, b28	CMMD[1:0]	Communication Mode Select	b29 b28 0 0: Transmit-receive mode (full-duplex) 0 1: Transmit-only mode (simplex, receiver disabled) 1 0: Receive-only mode (simplex, transmitter disabled) 1 1: Setting prohibited	R/W
b30	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b31	SYNDIS	Synchronizer Disable	0: TCLK = SPICLK, synchronizer enabled 1: TCLK = PCLKA, synchronizer disabled	R/W

Note 1. This bit is only valid when the SPMS bit is 0.

The SPCR register is used to specify the operating mode of the RSPIA. If the value of the SPPE, SPOE, PTE, SCKASE, SCKDDIS, MODFEN, SPMS, FRFS, CMMD[1:0], MSTR, or SYNDIS bit is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

SPE Bit (RSPI Function Enable)

This bit enables or disables the RSPI function. Setting this bit to 1 enables the RSPI function. When the SPSR.MODF flag is 1, this bit is set to 0. This bit cannot be set to 1 until the MODF flag is cleared. For details, refer to section 47.3.10, Error Detection.

Setting this bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 47.3.12, Initializing RSPIA.

SPPE Bit (Parity Enable)

This bit enables or disables the parity function.

SPOE Bit (Parity Mode)

This bit specifies even or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is effective only when the SPCR.SPPE bit is set to 1.

PTE Bit (Parity Self-Diagnosis)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK stops immediately before an overrun error occurs during data reception in master mode. For details, refer to section 47.3.10.1, Overrun Error.

SCKDDIS Bit (RSPCK Delay Between Frames Disable)

This bit controls whether or not a delay is inserted between frames during burst transfer.

This bit is valid for frames with the SPCMDm.SSLKP bit set to 1 in master mode (SPCR.MSTR = 1).

Set this bit to 0 in slave mode (SPCR.MSTR = 0). Table 47.3 shows the usage of SSL delay control in frame transmission. For details, refer to (4) Burst Transfer in section 47.3.13.1, Master Mode Operation.

Table 47.3 Usage of SSL Delay Control in Frame Transmission (in Master Mode)

Case		SPCMDm.SSLKP bit	SPCR.SCKDDIS bit	RSPCK delay, SSL negation delay, next access delay
Non-burst transfers		0	0	RSPCK delay, SSL negation delay, and next access delay are controlled according to the register settings*1.
Burst transfer with delay between frames	From the first frame to before the last frame	1	0	
	The last frame	0	0	
Burst transfer with no delay between frames	From the first frame to before the last frame	1	1	Each delay control is only performed in the following cases. • RSPCK delay of the first frame • SSL negation delay and next access delay of the last frame
	The last frame	0	1	

Note 1. RSPCK delay is controlled by the SPCKD.SCKDL[2:0] bits and SPCMDm.SCKDEN bit.
SSL negation delay is controlled by the SSLND.SLNDL[2:0] bits and SPCMDm.SLNDEN bit.
Next access delay is controlled by the SPND.SPNDL[2:0] bits and SPCMDm.SPNDEN bit.

Table 47.4 Operation Example in Motorola SPI (SCKDDIS = 1)

Setting	Operation
SPCMD0.SSLKP = 1	Between frames 0 and 1: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD1.SSLKP = 1	Between frames 1 and 2: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD2.SSLKP = 1	Between frames 2 and 3: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD3.SSLKP = 1	Between frames 3 and 4: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD4.SSLKP = 0	Burst transfer is not performed and the SSLni pin is negated once.*1
SPCMD5.SSLKP = 1	Between frames 5 and 6: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD6.SSLKP = 1	Between frames 6 and 7: burst transfer and no inter-frame delay (SSLni pin remains asserted)
SPCMD7.SSLKP = 0	Burst transfer is not performed and the SSLni pin is negated once.*1

Note 1. The SCKDDIS bit setting is invalid because burst transfer is not performed.

MODFEN Bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (Refer to section 47.3.10, Error Detection.) The RSPIA determines SSLn0 pin input or output direction according to the combination of the MODFEN and MSTR bits (refer to section 47.3.2, Controlling RSPI Pins).

SPEIE Bit (Error Interrupt Enable)

This bit is used to enable or disable an error interrupt request.
When the SPSR.MODF, OVRF, or PERF flag is set to 1 while this bit is 1, an error interrupt request is generated (refer to section 47.3.10, Error Detection).

SPRIE Bit (Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request.

SPIIE Bit (Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt requests.
When the SPSR.IDLNF flag changes from 1 to 0 while this bit is 1, an idle interrupt request is generated.

RDRIS Bit (Receive Data Ready Interrupt Select)

This bit is used to select the SPRI interrupt request or SPEI interrupt request to be generated when the receive data ready is detected (SPSR.RRDYF = 1).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request.

SPCIE Bit (Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS Bit (RSPI Mode Select)

This bit is used to select SPI operation (4-wire method) or clock synchronous operation (3-wire method).
For clock synchronous operation, the SSLni pin is not used but three pins RSPCKn, MOSIn, and MISOn are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the FRFS bit is invalid.
To perform clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), the operation is not guaranteed.
Table 47.5 lists the communication status according to the settings of the MSTR bit, CMMD[1:0] bits, FRFS bit, and SPMS bit of the RSPI control register (SPCR) as follows.

Table 47.5 RSPI Communication Status

SPCR. MSTR	SPCR. CMMD[1:0]	SPCR. FRFS	SPCR. SPMS	Communication Status	Communi-cation Status No.
1	00b	0	0	Transmit-Receive Master mode/Motorola SPI/SPI operation (4-wire)	1-1
1	00b	1	0	Transmit-Receive Master mode/TI SSP/SPI operation (4-wire)	1-2
1	00b	—	1	Transmit-Receive Master mode/Clock synchronous operation (3-wire)	1-3
1	01b	0	0	Transmit-only Master mode/Motorola SPI/SPI operation (4-wire)	1-4
1	01b	1	0	Transmit-only Master mode/TI SSP/SPI operation (4-wire)	1-5
1	01b	—	1	Transmit-only Master mode/Clock synchronous operation (3-wire)	1-6
1	10b	0	0	Receive-only Master mode/Motorola SPI/SPI operation (4-wire)	1-7
1	10b	1	0	Receive-only Master mode/TI SSP/SPI operation (4-wire)	1-8
1	10b	—	1	Receive-only Master mode/Clock synchronous operation (3-wire)	1-9
0	00b	0	0	Transmit-Receive Slave mode/Motorola SPI/SPI operation (4-wire) (default)	0-1
0	00b	1	0	Transmit-Receive Slave mode/TI SSP/SPI operation (4-wire)	0-2
0	00b	—	1	Transmit-Receive Slave mode/Clock synchronous operation (3-wire)	0-3
0	01b	0	0	Transmit-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-4
0	01b	1	0	Transmit-only Slave mode/TI SSP/SPI operation (4-wire)	0-5
0	01b	—	1	Transmit-only Slave mode/Clock synchronous operation (3-wire)	0-6
0	10b	0	0	Receive-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-7
0	10b	1	0	Receive-only Slave mode/TI SSP/SPI operation (4-wire)	0-8
0	10b	—	1	Receive-only Slave mode/Clock synchronous operation (3-wire)	0-9

—: don't care

FRFS Bit (Frame Format Select)

This bit selects the communication protocol. The format of the RSPI pin (RSPCKn, SSLni (i = 0 to 3) can be set according to the set communication protocol.
When the SPMS bit is 1 (clock synchronous operation (3-wire method)), this bit is invalid because SSLni pin is not used.

CMMD[1:0] Bits (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.
When CMMD[1:0] is set to 01b for communication, transmit-only is performed without reception.
When CMMD[1:0] is set to 10b for communication, receive-only is performed without transmission.
When CMMD[1:0] is set to 01b for communication, a receive buffer full interrupt request cannot be used.

When CMMD[1:0] is set to 10b for communication, a transmit buffer empty interrupt request cannot be used.
Refer to section 47.3.6, Communication Mode.

MSTR Bit (RSPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the RSPIA. The RSPIA determines input/output directions of pins RSPCKn, MOSIn, MISOn, and SSLn1 to SSLn3 according to the MSTR bit setting.

SYNDIS Bit (Synchronizer Disable)

This bit is used to bypass the synchronizers for clock domain crossing. When this bit is set to 1, the peripheral module clock (PCLKA) is selected as the operating clock (TCLK) and the synchronizers are bypassed. For details, refer to section 47.3.19, Synchronizer Bypass Function.

47.2.6 RSPI Receive-Only Mode Control Register (SPRMCR)

Address(es): RSPIA0.SPRMCR 8745 C00Ch, RSPIA1.SPRMCR 8745 C10Ch, RSPIA2.SPRMCR 8745 C20Ch, RSPIA0_NS.SPRMCR 0745 C00Ch, RSPIA1_NS.SPRMCR 0745 C10Ch, RSPIA2_NS.SPRMCR 0745 C20Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	START	TERM	—	RFC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RFC[4:0]	Receive Frame Count Select	The number of received frames can be adjusted in receive-only master mode. b4 b0 0 0 0 0 0: This function is not used. 0 0 0 0 1: Automatically stop communication after processing 1 received frame : 1 1 1 1 1: Automatically stop communication after processing 31 received frames Settings are not effective except in receive-only master mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	TERM	Reception Terminate	1: Receive End (Writable only in receive-only master mode) The read value is always 0.	W
b7	START	Reception Start	1: Receive Start (Writable only in receive-only master mode) The read value is always 0.	W

The SPRMCR register is used to control the start and completion of communication for receive-only master mode. If the RFC[4:0] bits are changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

RFC[4:0] Bits (Receive Frame Count Select)

The number of received frames can be adjusted when operating in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b. After reception is started by the START bit, the communication is automatically stopped after processing frames according to the value set in these bits.

TERM Bit (Reception Terminate)

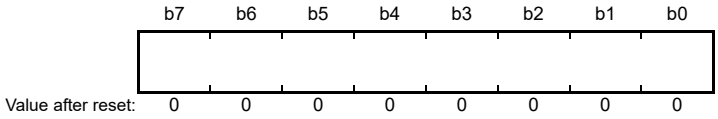
This bit is used to end reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.

START Bit (Reception Start)

This bit is used to start reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.
Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

47.2.7 RSPI Receive Data Ready Detect Condition Setting Register (SPDRCSR)

Address(es): RSPIA0.SPDRCSR 8745 C00Dh, RSPIA1.SPDRCSR 8745 C10Dh, RSPIA2.SPDRCSR 8745 C20Dh, RSPIA0_NS.SPDRCSR 0745 C00Dh, RSPIA1_NS.SPDRCSR 0745 C10Dh, RSPIA2_NS.SPDRCSR 0745 C20Dh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	0h: Disables receive data ready detection function 1h: Performs receive data ready detection after 1 TCLK : FFh: Performs receive data ready detection after 255 TCLK	R/W

The SPDRCSR register is used to set the RSPI receive data ready detection function. If the setting value is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 TCLK. The value set in the SPDRCSR register is used to 1 set the RRDYF flag. For details, see the description of the RRDYF flag in section 47.2.16, RSPI Status Register (SPSR).

47.2.8 RSPI Pin Control Register (SPPCR)

Address(es): RSPIA0.SPPCR 8745 C00Eh, RSPIA1.SPPCR 8745 C10Eh, RSPIA2.SPPCR 8745 C20Eh, RSPIA0_NS.SPPCR 0745 C00Eh, RSPIA1_NS.SPPCR 0745 C10Eh, RSPIA2_NS.SPPCR 0745 C20Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	MOIFV	MOSI Idle Fixed Value	0: The fixed value during MOSI idling corresponds to low 1: The fixed value during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPPCR register is used to set pin mode of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

SPLP Bit (RSPI Loopback)

When the SPLP bit is set to 1, the RSPIA shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

When the SPLP2 bit is set to 1, the RSPIA shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).
If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIn pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIn output value when the RSPIA in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPIA outputs the last data from the previous serial transfer during the SSL negation period to the MOSIn pin. When the MOIFE bit is 1, the RSPIA outputs the fixed value set in the MOIFV bit to the MOSIn pin.

47.2.9 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPIA0.SSLP 8745 C010h, RSPIA1.SSLP 8745 C110h, RSPIA2.SSLP 8745 C210h, RSPIA0_NS.SSLP 0745 C010h, RSPIA1_NS.SSLP 0745 C110h, RSPIA2_NS.SSLP 0745 C210h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	[Motorola SPI] 0: The SSLn0 signal is active low. 1: The SSLn0 signal is active high. [TI SSP] 0: The SSLn0 signal is active high. 1: The SSLn0 signal is active low.	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	[Motorola SPI] 0: The SSLn1 signal is active low. 1: The SSLn1 signal is active high. [TI SSP] 0: The SSLn1 signal is active high. 1: The SSLn1 signal is active low.	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	[Motorola SPI] 0: The SSLn2 signal is active low. 1: The SSLn2 signal is active high. [TI SSP] 0: The SSLn2 signal is active high. 1: The SSLn2 signal is active low.	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	[Motorola SPI] 0: The SSLn3 signal is active low. 1: The SSLn3 signal is active high. [TI SSP] 0: The SSLn3 signal is active high. 1: The SSLn3 signal is active low.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLP register is used to set the polarity of SSLni signals (i = 0 to 3) of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

SSLiP Bits (SSLi Signal Polarity Setting) (i = 0 to 3)

These bits are used to specify the active level of the SSLni signals.

Note: The SSLn0 signal functions as an input signal in slave or multi-master mode, unlike the SSLn1 to SSLn3 signals. For details, refer to section 47.3.3.2, Single Master/Single Slave (with This MCU Acting as Slave), and section 47.3.3.5, Multi-Master/Multi-Slave (with This MCU Acting as Master).

47.2.10 RSPI Bit Rate Register (SPBR)

Address(es): RSPIA0.SPBR 8745 C011h, RSPIA1.SPBR 8745 C111h, RSPIA2.SPBR 8745 C211h, RSPIA0_NS.SPBR 0745 C011h, RSPIA1_NS.SPBR 0745 C111h, RSPIA2_NS.SPBR 0745 C211h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The SPBR register is used to set the bit rate in master mode. If the SPBR register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operation is not guaranteed.

The bit rate is determined by combinations of the SPBR register setting and the SPCMDm.BRDV[1:0] bit setting.

When the RSPIA is used in slave mode, the bit rate depends on the frequency of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of the SPBR register and the SPCMDm.BRDV[1:0] bits.

The equation for calculating the bit rate is given below. In the equation, n denotes the SPBR register setting (0, 1, 2, …, 255), and N denotes the BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(TCLK)}{2 \times (n + 1) \times 2^{N}}$$

Table 47.6 lists examples of the relationship among the SPBR register settings, the BRDV[1:0] bit settings, and bit rates.

Table 47.6 Relationship among SPBR Register Settings, BRDV[1:0] Bit Settings, and Bit Rates

SPBR Register (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate						
			TCLK = 32 MHz	TCLK = 36 MHz	TCLK = 40 MHz	TCLK = 50 MHz	TCLK = 100 MHz	TCLK = 120 MHz	TCLK = 150 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	50.0 Mbps	60.0 Mbps	—
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	25.0 Mbps	30.0 Mbps	37.5 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	16.7 Mbps	20.0 Mbps	25.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	12.5 Mbps	15.0 Mbps	18.8 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	10.0 Mbps	12.0 Mbps	15.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	8.33 Mbps	10.0 Mbps	12.5 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	4.17 Mbps	5.00 Mbps	6.25 Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.08 Mbps	2.50 Mbps	3.13 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.04 Mbps	1.25 Mbps	1.56 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	24.4 kbps	29.3 kbps	36.6 kbps

47.2.11 RSPI Sequence Control Register (SPSCR)

Address(es): RSPIA0.SPSCR 8745 C013h, RSPIA1.SPSCR 8745 C113h, RSPIA2.SPSCR 8745 C213h, RSPIA0_NS.SPSCR 0745 C013h, RSPIA1_NS.SPSCR 0745 C113h, RSPIA2_NS.SPSCR 0745 C213h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Setting	b2 b0 Sequence Length Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 0→0→… 0 0 1: 2 0→1→0→… 0 1 0: 3 0→1→2→0→… 0 1 1: 4 0→1→2→3→0→… 1 0 0: 5 0→1→2→3→4→0→… 1 0 1: 6 0→1→2→3→4→5→0→… 1 1 0: 7 0→1→2→3→4→5→6→0→… 1 1 1: 8 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPIA is shown above. In slave mode, the SPCMD0 register is referenced.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPSCR register is used to set the sequence length when the RSPIA operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Setting)

The SPSLN[2:0] bits specify a sequence length when the RSPIA in master mode performs sequential operations. The RSPIA in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits. For details, refer to (3) Sequence Control in section 47.3.13.1, Master Mode Operation.

In slave mode, the SPCMD0 register is referenced.

47.2.12 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPIA0.SPCMD0 8745 C014h, RSPIA0.SPCMD1 8745 C018h, RSPIA0.SPCMD2 8745 C01Ch, RSPIA0.SPCMD3 8745 C020h, RSPIA0.SPCMD4 8745 C024h, RSPIA0.SPCMD5 8745 C028h, RSPIA0.SPCMD6 8745 C02Ch, RSPIA0.SPCMD7 8745 C030h, RSPIA1.SPCMD0 8745 C114h, RSPIA1.SPCMD1 8745 C118h, RSPIA1.SPCMD2 8745 C11Ch, RSPIA1.SPCMD3 8745 C120h, RSPIA1.SPCMD4 8745 C124h, RSPIA1.SPCMD5 8745 C128h, RSPIA1.SPCMD6 8745 C12Ch, RSPIA1.SPCMD7 8745 C130h, RSPIA2.SPCMD0 8745 C214h, RSPIA2.SPCMD1 8745 C218h, RSPIA2.SPCMD2 8745 C21Ch, RSPIA2.SPCMD3 8745 C220h, RSPIA2.SPCMD4 8745 C224h, RSPIA2.SPCMD5 8745 C228h, RSPIA2.SPCMD6 8745 C22Ch, RSPIA2.SPCMD7 8745 C230h, RSPIA0_NS.SPCMD0 0745 C014h, RSPIA0_NS.SPCMD1 0745 C018h, RSPIA0_NS.SPCMD2 0745 C01Ch, RSPIA0_NS.SPCMD3 0745 C020h, RSPIA0_NS.SPCMD4 0745 C024h, RSPIA0_NS.SPCMD5 0745 C028h, RSPIA0_NS.SPCMD6 0745 C02Ch, RSPIA0_NS.SPCMD7 0745 C030h, RSPIA1_NS.SPCMD0 0745 C114h, RSPIA1_NS.SPCMD1 0745 C118h, RSPIA1_NS.SPCMD2 0745 C11Ch, RSPIA1_NS.SPCMD3 0745 C120h, RSPIA1_NS.SPCMD4 0745 C124h, RSPIA1_NS.SPCMD5 0745 C128h, RSPIA1_NS.SPCMD6 0745 C12Ch, RSPIA1_NS.SPCMD7 0745 C130h, RSPIA2_NS.SPCMD0 0745 C214h, RSPIA2_NS.SPCMD1 0745 C218h, RSPIA2_NS.SPCMD2 0745 C21Ch, RSPIA2_NS.SPCMD3 0745 C220h, RSPIA2_NS.SPCMD4 0745 C224h, RSPIA2_NS.SPCMD5 0745 C228h, RSPIA2_NS.SPCMD6 0745 C22Ch, RSPIA2_NS.SPCMD7 0745 C230h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 5 TCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: In master mode, an SSL negation delay of 1 RSPCK In TI SSP slave mode, an SSL negation delay of 1 TCLK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: In Motorola SPI protocol, an RSPCK delay of 1 RSPCK In TI SSP protocol, an RSPCK delay of 0 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W
b20 to b16	SPB[4:0]	RSPI Data Length Setting	b20 b16 0 0 0 1 1: 4 bits 0 0 1 0 0: 5 bits 0 0 1 0 1: 6 bits : : 1 1 1 1 0: 31 bits 1 1 1 1 1: 32 bits Settings other than above are prohibited.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b26 to b24	SSLA[2:0]	SSL Signal Assertion Setting	b26 b24 0 0 0: SSLn0 0 0 1: SSLn1 0 1 0: SSLn2 0 1 1: SSLn3 Settings other than above are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPCMDm register is used to set a transfer format for the RSPIA in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPIA in slave mode. The RSPIA in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
The SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPIA in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit is used to set an RSPCK phase of the RSPIA in master mode or slave mode. Data communications between RSPIA modules require the same RSPCK phase setting between the modules.
When the SPCR.SPMS is 0 and the SPCR.FRFS bit is 1 in TI SSP protocol, setting the CPHA bit to 0 is not effective.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit is used to set an RSPCK polarity of the RSPIA in master mode or slave mode. Data communications between RSPIA modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and the SPBR register (refer to section 47.2.10, RSPI Bit Rate Register (SPBR)). The settings in the SPBR register determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPIA in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLni signal level for the current command is to be kept or negated between the SSLni negation timing associated with the current command and the SSLni assertion timing associated with the next command (i = 0 to 3).
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 47.3.13.1, Master Mode Operation (4) Burst Transfer.
When using the RSPIA in slave mode, the SSLKP bit should be set to 0.

LSBF Bit (RSPI LSB First)

The LSBF bit is used to set the data format of the RSPIA in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit is used to set the period from the time the RSPIA in master mode terminates a serial transfer and sets the SSLni signal inactive until the RSPIA enables the SSLni signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPIA sets the next-access delay to 1 RSPCK + 5 TCLK. If the SPNDEN bit is 1, the RSPIA inserts a next-access delay in compliance with the SPND setting.

When using the RSPIA in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

[Motorola SPI]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the SSL signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPIA sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the OE signal inactive, or from the detection of a final RSPCK edge to the negation of the OE signal by the RSPIA in slave mode. If the SLNDEN bit is 0, the SSL negation delay is 1 RSPCK in master mode and 1 TCLK in slave mode. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

[Clock synchronous]
When using the RSPIA in slave mode, set this bit to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

[Motorola SPI]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay). If the SCKDEN bit is 0, the RSPIA sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

[TI SSP]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay) and to the negation of the SSLni signal. If the SCKDEN bit is 0, the RSPIA does not set the RSPCK delay. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.
When using the RSPIA in slave mode, the SCKDEN bit should be set to 0.

SPB[4:0] Bits (RSPI Data Length Setting)

The SPB[4:0] bits are used to set a transmit data length for the RSPIA in master mode or slave mode.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLni signal assertion when the RSPIA performs serial transfers in master mode. Setting the SSLA[2:0] bits controls the assertion for the SSLni signal (i = 0 to 3). When an SSLni signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLn0 pin acts as input).
When using the RSPIA in slave mode, set the SSLA[2:0] bits to 000b.

47.2.13 RSPI Data Control Register (SPDCR)

Address(es): RSPIA0.SPDCR 8745 C040h, RSPIA1.SPDCR 8745 C140h, RSPIA2.SPDCR 8745 C240h, RSPIA0_NS.SPDCR 0745 C040h, RSPIA1_NS.SPDCR 0745 C140h, RSPIA2_NS.SPDCR 0745 C240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SPFC[1:0]		—	—	—	DINV	SPRDTD	—	—	BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BYSW	Byte Swap	0: Byte swapping of SPDR data disabled 1: Byte swapping of SPDR data enabled	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	SPRDTD	RSPI Receive/Transmit Data Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer	R/W
b4	DINV	Transfer Data Invert	0: Not invert serial data 1: Invert serial data	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	SPFC[1:0]	Number of Frames Setting	b9 b8 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPDCR register is used to control the data format of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

BYSW Bit (Byte Swap)

The BYSW bit is used to swap a transmit/receive data in byte units. A data alignment after byte swap is different by a data length (setting of SPCMDm.SPB[4:0]).

When using byte swap, set the data length (setting of SPCMDm.SPB[4:0]) to 32 bits or 16 bits. For other data length (i.e. 4 to 15, 17 to 31-bit length), byte swap operation is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, refer to section 47.3.4.4, Byte Swap Transmission and section 47.3.4.5, Byte Swap Reception.

The operation is also not guaranteed when the parity function is enabled (the SPCR.SPPE bit is 1).

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit is used to whether the SPDR register reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the value written to the SPDR register immediately beforehand is read.

DINV Bit (Transfer Data Invert)

The DINV bit is used to invert transmit data and receive data.

When the DINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] Bits (Number of Frames Setting)

The SPFC[1:0] bits are used for the condition to set the SPSR.SPCF flag in slave receive-only mode.

For details on the SPCF flag setting conditions, refer to [Setting condition] of the SPCF flag in section 47.2.16.

Note that these bits are not effective except in the slave receive-only mode.

47.2.14 RSPI FIFO Control Register (SPFCR)

Address(es): RSPIA0.SPFCR 8745 C044h, RSPIA1.SPFCR 8745 C144h, RSPIA2.SPFCR 8745 C244h, RSPIA0_NS.SPFCR 0745 C044h, RSPIA1_NS.SPFCR 0745 C144h, RSPIA2_NS.SPFCR 0745 C244h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RTRG[1:0]	Receive FIFO Threshold Setting	b1 b0 0 0: When the receive FIFO fill level is 0. 0 1: When the receive FIFO fill level is 1. 1 0: When the receive FIFO fill level is 2. 1 1: When the receive FIFO fill level is 3.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	TTRG[1:0]	Transmit FIFO Threshold Setting	b9 b8 0 0: When the transmit FIFO free level is 0. 0 1: When the transmit FIFO free level is 1. 1 0: When the transmit FIFO free level is 2. 1 1: When the transmit FIFO free level is 3.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCR register is used to control the FIFO threshold. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

RTRG[1:0] Bits (Receive FIFO Threshold Setting)

The RTRG[1:0] bits are used to select receive FIFO threshold. When the number of data stored in the receive FIFO is greater than the threshold value set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

The TTRG[1:0] bits are used to select transmit FIFO threshold. When the number of empty stages in the transmit FIFO is greater than the threshold value set in TTRG[1:0], the transmit buffer empty flag is set.

47.2.15 RSPI Sequence Status Register (SPSSR)

Address(es): RSPIA0.SPSSR 8745 C051h, RSPIA1.SPSSR 8745 C151h, RSPIA2.SPSSR 8745 C251h, RSPIA0_NS.SPSSR 0745 C051h, RSPIA1_NS.SPSSR 0745 C151h, RSPIA2_NS.SPSSR 0745 C251h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the RSPIA operates in master mode.
Any writing to the SPSSR register is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is currently pointed to by the pointer during sequence control by the RSPIA.
For the RSPIA's sequence control, refer to section 47.3.13.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPIA. The RSPIA updates the SPECM[2:0] bits only when an error is detected. If the SPSR.OVRF, MODF, and PERF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.
For the RSPIA's error detection function, refer to section 47.3.10, Error Detection. For the RSPIA's sequence control, refer to section 47.3.13.1, Master Mode Operation.

47.2.16 RSPI Status Register (SPSR)

Address(es): RSPIA0.SPSR 8745 C052h, RSPIA1.SPSR 8745 C152h, RSPIA2.SPSR 8745 C252h, RSPIA0_NS.SPSR 0745 C052h, RSPIA1_NS.SPSR 0745 C152h, RSPIA2_NS.SPSR 0745 C252h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	SPCF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	RRDYF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0.	R
b7	RRDYF	Receive Data Ready Flag	0: Receive data ready is not detected 1: Receive data ready is detected	R
b8	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R
b9	IDLNF	Idle Flag	0: RSPIA is in the idle state 1: RSPIA is in the transfer state	R
b10	MODF	Mode Fault Error Flag	0: Neither a mode fault error nor an underrun error occurs 1: A mode fault error or an underrun error occurs	R
b11	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R
b12	UDRF	Underrun Error Flag	This flag is used with the MODF flag to check the state in terms of mode fault errors and underrun errors. MODF UDRF 0 x: Neither a mode fault error nor an underrun error occurs 1 0: A mode fault error occurs 1 1: An underrun error occurs	R
b13	SPTEF	Transmit Buffer Empty Flag	0: The number of empty stages in the transmit FIFO ≤ the value set by the SPFCR.TTRG[1:0] bits 1: The number of empty stages in the transmit FIFO > the value set by the SPFCR.TTRG[1:0] bits	R
b14	SPCF	Communication End Flag	0: RSPIA is not communicating or communicating 1: RSPIA communication is completed	R
b15	SPRF	Receive Buffer Full Flag	0: The number of data stored in the receive FIFO ≤ number of frames set by the SPFCR.RTRG[1:0] bits 1: The number of data stored in the receive FIFO > number of frames set by the SPFCR.RTRG[1:0] bits	R

x: don't care

The SPSR register is used to store flags that indicate RSPIA's operating status.

RRDYF Flag (Receive Data Ready Flag)

The RRDYF flag indicates a certain period of time has elapsed while the number of data stored in the receive FIFO during communication (SPCR.SPE = 1) is equal to or less than the receive FIFO threshold value.
This flag is set to 0 when the receive operation is not performed (SPCR.CMMD[1:0] = 01b).

[Setting condition]

When all the following conditions are satisfied.

- SPDRCSR register ≠ 00h.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO is equal to or less than the receive FIFO threshold value and the value set by the SPDRCSR register has elapsed

[Clearing condition]
When 1 is written to the SPSCLR.RRDYFC bit.

OVRF Flag (Overrun Error Flag)
The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK auto-stop function is enabled (the SPCR.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 47.3.10.1, Overrun Error.

[Setting condition]
When serial transfer is completed with no free space in the receive FIFO under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode).
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode).

[Clearing condition]
When 1 is written to the SPSCLR.OVRFC bit

IDLNF Flag (Idle Flag)
The IDLNF flag indicates the transfer status of the RSPIA.

[Setting condition]
(1) Transmit-receive master mode or transmit-only master mode

- When none of the conditions are satisfied in transmit-receive master mode or transmit-only master mode under the [Clearing condition] below.

(2) Transmit-only master mode

- When 1 is written to the SPRMCR.START bit.

(3) Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]
(1) Transmit-receive master mode or transmit-only master mode
Communication status: No. 1-1 to 1-6 (for details, refer to Table 47.5 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When all of the following conditions are satisfied.
  (a) The the next transmit data is not set in the transmit buffer (SPTX)
  (b) The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
  (c) The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode
Communication status: No. 1-7 to 1-9 (for details, refer to Table 47.5 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When any of the following conditions is satisfied.
  (a) When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(b) When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(c) When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Slave mode

Communication status: No. 0-1 to 0-9 (for details, refer to Table 47.5 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

(1) Multi-master mode

- When the input level of the SSLn0 pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error

(2) Slave mode (SPI operation, Motorola SPI)

When any of the following conditions are satisfied.

- When the SSLn0 pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

(3) Slave mode (SPI operation, TI SSP)

When any of the following conditions are satisfied.

- When the SSLn0 pin is asserted before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSL signal polarity setting bit).

[Clearing condition]

When 1 is written to the SPSCLR.MODFC bit

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

When the serial transfer ends and a parity error is detected while the SPCR.SPPE bit is 1 under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode)
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode)

[Clearing condition]
When 1 is written to the SPSCLR.PERFC bit

UDRF Flag (Underrun Error Flag)
When a mode fault error or an underrun error has occurred, the UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]
When the serial transfer starts while the SPCR.MSTR bit is 0, the SPCR.CMMD[1:0] bits are 00b or 01b (transmit-receive slave mode or transmit-only slave mode), the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

[Clearing condition]
When 1 is written to the SPSCLR.UDRFC bit

SPTEF Flag (Transmit Buffer Empty Flag)
Indicates whether the transmit buffer (SPTXn, n = 0 to 3) in the RSPI data register (SPDR) has valid data.

[Setting condition]
When any of the following conditions are satisfied.

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When the number of empty transmission FIFO stages is greater than the threshold value set in SPFCR.TTRG[1:0]
- When 1 is written to the SPFCLR.FCLR bit

[Clearing condition]
When any of the following conditions are satisfied.

- At the final access when transmit data is written to SPDR (SPTXn) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPTEFC bit

The SPDR register can be set only when the SPSR.SPTEF flag is 1. The data in the transmit buffer (SPTXn) is not updated when the SPDR register is set while the SPSR.SPTEF flag is 0.

SPCF Flag (Communication End Flag)
Indicates communication end status of RSPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]
(1) Transmit-receive master mode or transmit-only master mode
Communication status: No. 1-1 to 1-6 (for details, refer to Table 47.5 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode
Communication status: No. 1-7 to 1-9 (for details, refer to Table 47.5 RSPI Communication Status)
When any of the following conditions are satisfied.

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the

next-access delay have finished (the master main state machine has entered the idle state)

- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Transmit-receive slave mode or transmit-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-1, 0-4 (for details, refer to Table 47.5 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The SSLn0 input is negated

(4) Transmit-receive slave mode or transmit-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-2, 0-5 (for details, refer to Table 47.5 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- When the SSL negation delay is completed

(5) Transmit-receive slave mode or transmit-only slave mode (clock synchronous operation)
Communication status: No. 0-3, 0-6 (for details, refer to Table 47.5 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The last even edge of RSPCK of the frame was detected (SPCMDm.CPHA = 1)

(6) Receive-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-7 (for details, refer to Table 47.5 RSPI Communication Status)

- SSLn0 input was negated after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX)

(7) Receive-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-8 (for details, refer to Table 47.5 RSPI Communication Status)

- SSL negation delay is completed after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX).

(8) Receive-only slave mode (clock synchronous operation)
Communication status: No. 0-9 (for details, refer to Table 47.5 RSPI Communication Status)

- The last even edge of RSPCK of the last frame received for SPFC[1:0] set value was detected (SPCMDm.CPHA = 1)

[Clearing condition]
(1) Transmit-receive master mode or transmit-only master mode
Communication status: No. 1-1 to 1-6 (for details, refer to Table 47.5 RSPI Communication Status)
When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 47.5 RSPI Communication Status)

When any of the following conditions are satisfied.

- When 1 is written to the SPRMCR.START bit while the SPCR.SPE bit is 1
- When 1 is written to the SPSCLR.SPCFC bit

(3) Transmit-receive slave mode or transmit-only slave mode

Communication status: No. 0-1 to 0-6 (for details, refer to Table 47.5 RSPI Communication Status)

When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(4) Receive-only slave mode (SPI operation)

Communication status: No. 0-7, 0-8 (for details, refer to Table 47.5 RSPI Communication Status)

When any of the following conditions are satisfied.

- The SSLn0 input assertion of next data is detected
- When 1 is written to the SPSCLR.SPCFC bit

(5) Receive-only slave mode (clock synchronous operation)

Communication status: No. 0-9 (for details, refer to Table 47.5 RSPI Communication Status)

When any of the following conditions are satisfied.

- The first edge of RSPCK of the next data was detected
- When 1 is written to the SPSCLR.SPCFC bit

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

When the number of data stored in the receive FIFO is greater than the number of frames set in the SPFCR.RTRG[1:0] bits in transmit-receive mode or receive-only mode.

Note that the SPRF flag does not become 1 when the OVRF flag is 1 (Refer to section 47.3.10, Error Detection, for details).

[Clearing condition]

When any of the following conditions are satisfied.

- At the final access when receive data is read from SPDR register (SPRXn, n = 0 to 3) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPRFC bit
- When 1 is written to the SPFCLR.FCLR bit

47.2.17 RSPI Transmit FIFO Status Register (SPTFSR)

Address(es): RSPIA0.SPTFSR 8745 C058h, RSPIA1.SPTFSR 8745 C158h, RSPIA2.SPTFSR 8745 C258h, RSPIA0_NS.SPTFSR 0745 C058h, RSPIA1_NS.SPTFSR 0745 C158h, RSPIA2_NS.SPTFSR 0745 C258h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FREE[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FREE[2:0]	Transmit FIFO Free Level	b2 b0 0 0 0: Number of empty stages 0 0 0 1: Number of empty stages 1 0 1 0: Number of empty stages 2 0 1 1: Number of empty stages 3 1 0 0: Number of empty stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FREE[2:0] Bits (Transmit FIFO Free Level)

Indicate the number of empty transmit FIFO stages. By setting the SPCR.SPE bit to 0, the FREE[2:0] bits will be 100b.

47.2.18 RSPI Receive FIFO Status Register (SPRFSR)

Address(es): RSPIA0.SPRFSR 8745 C05Ch, RSPIA1.SPRFSR 8745 C15Ch, RSPIA2.SPRFSR 8745 C25Ch, RSPIA0_NS.SPRFSR 0745 C05Ch, RSPIA1_NS.SPRFSR 0745 C15Ch, RSPIA2_NS.SPRFSR 0745 C25Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FILL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FILL[2:0]	Receive FIFO Fill Level	b2 b0 0 0 0: Number of stored stages 0 0 0 1: Number of stored stages 1 0 1 0: Number of stored stages 2 0 1 1: Number of stored stages 3 1 0 0: Number of stored stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FILL[2:0] Bits (Receive FIFO Fill Level)

Indicate the number of stored receive FIFO stages. By setting the SPCR.SPE bit to 0, the FILL[2:0] bits will be 000b.

47.2.19 RSPI Polling Status Register (SPPSR)

Address(es): RSPIA0.SPPSR 8745 C060h, RSPIA1.SPPSR 8745 C160h, RSPIA2.SPPSR 8745 C260h, RSPIA0_NS.SPPSR 0745 C060h, RSPIA1_NS.SPPSR 0745 C160h, RSPIA2_NS.SPPSR 0745 C260h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPEPS	SPE Polling Status Flag	0: The value of the SPCR.SPE bit after synchronization with TCLK is 0 1: The value of the SPCR.SPE bit after synchronization with TCLK is 1	R
b31 to b1	—	Reserved	These bits are read as 0.	R

SPEPS Flag (SPE Polling Status Flag)

This flag indicates the status after synchronizing the value of the SPCR.SPE bit with the operating clock (TCLK). In other words, this flag indicates whether the RSPI function is enabled or disabled within the transmitter/receiver. When the SPE bit is changed from 1 to 0 while the SPCR.SYNDIS bit is 0, confirm that this flag is set to 0, and then proceed to the next processing.

47.2.20 RSPI Status Clear Register (SPSCLR)

Address(es): RSPIA0.SPSCLR 8745 C06Ah, RSPIA1.SPSCLR 8745 C16Ah, RSPIA2.SPSCLR 8745 C26Ah, RSPIA0_NS.SPSCLR 0745 C06Ah, RSPIA1_NS.SPSCLR 0745 C16Ah, RSPIA2_NS.SPSCLR 0745 C26Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRFC	SPCFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	RRDYFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	RRDYFC	Receive Data Ready Flag Clear	Writing 1 to this bit clears the receive data ready flag. The read value is always 0.	W
b8	OVRFC	Overrun Error Flag Clear	Writing 1 to this bit clears the overrun error flag. The read value is always 0.	W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R
b10	MODFC	Mode Fault Error Flag Clear	Writing 1 to this bit clears the mode fault error flag. The read value is always 0.	W*1
b11	PERFC	Parity Error Flag Clear	Writing 1 to this bit clears the parity error flag. The read value is always 0.	W
b12	UDRFC	Underrun Error Flag Clear	Writing 1 to this bit clears the underrun error flag. The read value is always 0.	W*1, *2
b13	SPTEFC	Transmit Buffer Empty Flag Clear	Writing 1 to this bit clears the transmit buffer empty flag. The read value is always 0.	W
b14	SPCFC	Communication End Flag Clear	Writing 1 to this bit clears the communication end flag. The read value is always 0.	W
b15	SPRFC	Receive Buffer Full Flag Clear	Writing 1 to this bit clears the receive buffer full flag. The read value is always 0.	W

Note 1. Before writing 1 to the MODFC bit, make sure that the SPSR.MODF flag is set to 1. Before writing 1 to the UDRFC bit, make sure that the SPSR.UDRF flag is set to 1.

Note 2. When clearing the UDRF flag, be sure to clear the MODF flag at the same time (write 1 to both UDRFC and MODFC bits simultaneously).

The SPSCLR register is used to clear each status flag in the SPSR register, which indicates the operating state of the RSPIA.

47.2.21 RSPI FIFO Clear Register (SPFCLR)

Address(es): RSPIA0.SPFCLR 8745 C06Ch, RSPIA1.SPFCLR 8745 C16Ch, RSPIA2.SPFCLR 8745 C26Ch, RSPIA0_NS.SPFCLR 0745 C06Ch, RSPIA1_NS.SPFCLR 0745 C16Ch, RSPIA2_NS.SPFCLR 0745 C26Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FCLR	FIFO Clear	By writing 1, the pointer in the FIFO and the stored data are initialized. The read value is always 0.	W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCLR register is used to clear the FIFO. If this register is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

FCLR Bit (FIFO Clear)

Writing 1 to this bit initializes the pointer and stored data in the transmit/receive FIFO.

47.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

47.3.1 Overview of RSPI Operations

The RSPIA is capable of serial transfers in the following five modes.

(1) Slave mode (SPI operation)
(2) Single-master mode (SPI operation)
(3) Multi-master mode (SPI operation)
(4) Slave mode (clock synchronous operation)
(5) Master mode (clock synchronous operation).

A particular mode of the RSPIA can be selected by using the MSTR, MODFEN, SPMS, and FRFS bits in SPCR. Table 47.7 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 47.7 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			Clock Synchronous Operation	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
FRFS bit setting	Enabled	Enabled	Enabled	Disabled	Disabled
RSPCKn signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z*1	Input	Output
MISOn signal	Output/Hi-Z*2	Input	Input	Output	Input
SSLn0 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SSLn1 to SSLn3 signals	Hi-Z (not used)	Output	Output/Hi-Z*1	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open-drain				
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to TCLK/2				
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*3	Two*3	Two*3	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transmit data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*4	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5	Supported	Supported	Supported*5	Supported
Receive buffer full detection	Supported*6				
Overrun error detection	Supported*6	Supported*6, *7	Supported*6, *7	Supported*6	Supported*6, *7
Parity error detection	Supported*6, *8				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. Motorola SPI: When SSLn0 is asserted by another master device, the pin becomes Hi-Z.
TI SSP: When the SPCR.SPE bit is 1, the pin becomes Hi-Z from the time SSLn0 is asserted by another master device until the end of communication.

Note 2. Motorola SPI: When SSLn0 is negated or the SPCR.SPE bit is 0, the pin becomes Hi-Z.
TI SSP: The pin becomes Hi-Z when the RSPIA is not communicating (from the end of communication to the next assertion of SSLn0) or when the SPCR.SPE bit is 0.

Note 3. In TI SSP protocol, setting the SPCMDm.CPHA bit to 0 is not effective.

Note 4. Available only in TI SSP protocol.

Note 5. When RSPIA is receive-only slave mode (refer to Table 47.5 RSPI Communication Status), transmit buffer empty error detection and underrun error detection are not performed.

Note 6. When RSPIA is transmit-only master mode or transmit-only slave mode (refer to Table 47.5 RSPI Communication Status), receiver buffer full detection, overrun error detection, and parity error detection are not performed.

Note 7. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.

Note 8. When the SPCR.SPPE bit is 0, parity error detection is not performed.

47.3.2 Controlling RSPI Pins

The RSPIA in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 47.8.

Table 47.8 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSI Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

47.3.3 RSPI System Configuration Examples

This configuration example describes that the low level of SSLni signals is the active level.
When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola SPI or TI SSP.

47.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 47.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLn0 to SSLn3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLni output of this MCU should be connected to the SSL input of the slave device.
This MCU (master) drives the RSPCKn and MOSIn. The SPI slave drives the MISO.

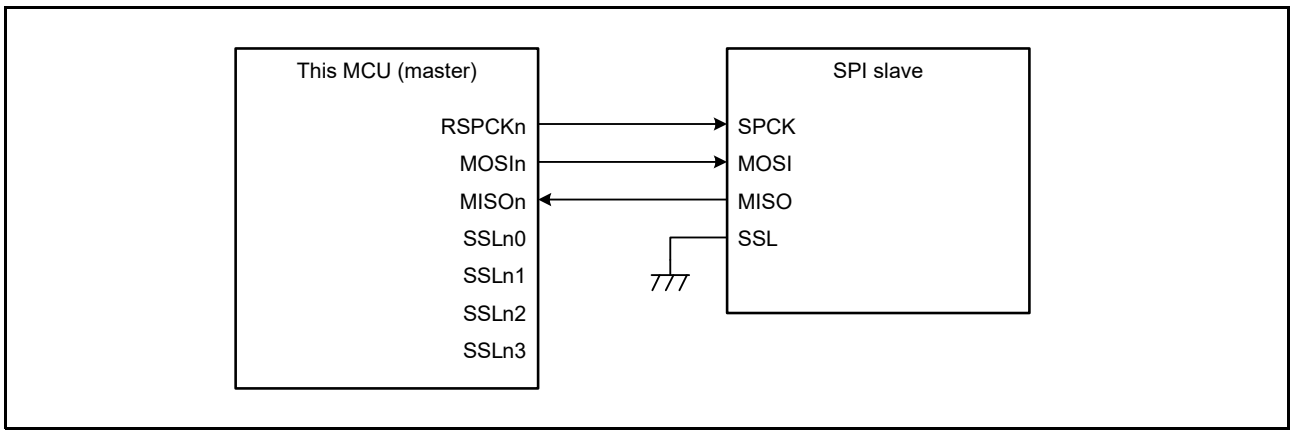


Figure 47.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

47.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 47.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the SPCK and MOSI. This MCU (slave) drives the MISOn. When SSLn0 is at the non-active level, the pin state is Hi-Z.
In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, set the SPCR.FRFS bit to 0, and set the SPCR.SPMS bit to 0, the SSLn0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 47.7). Note that the communication end event and the communication end interrupt do not output when SSLn0 input was fixed as Figure 47.7.

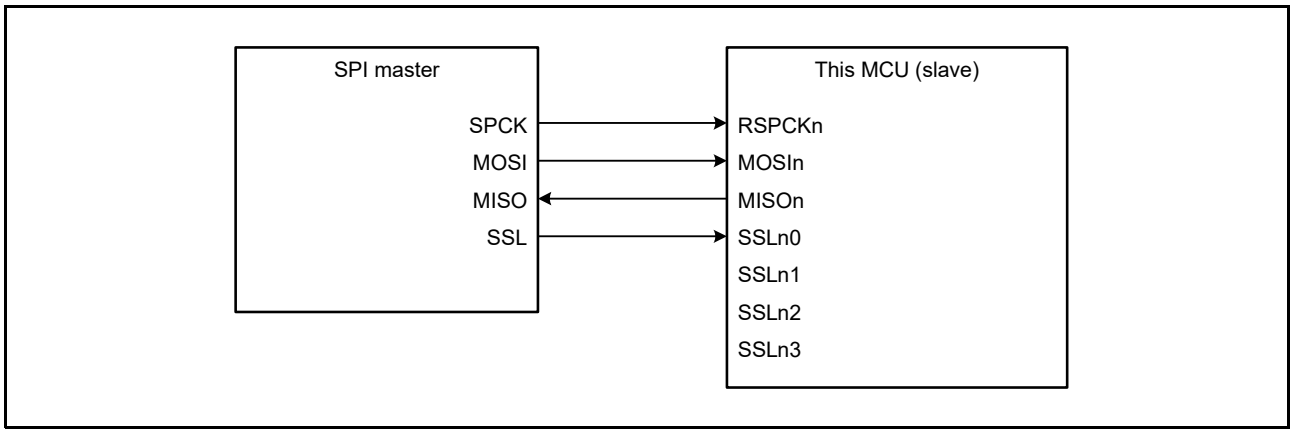


Figure 47.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

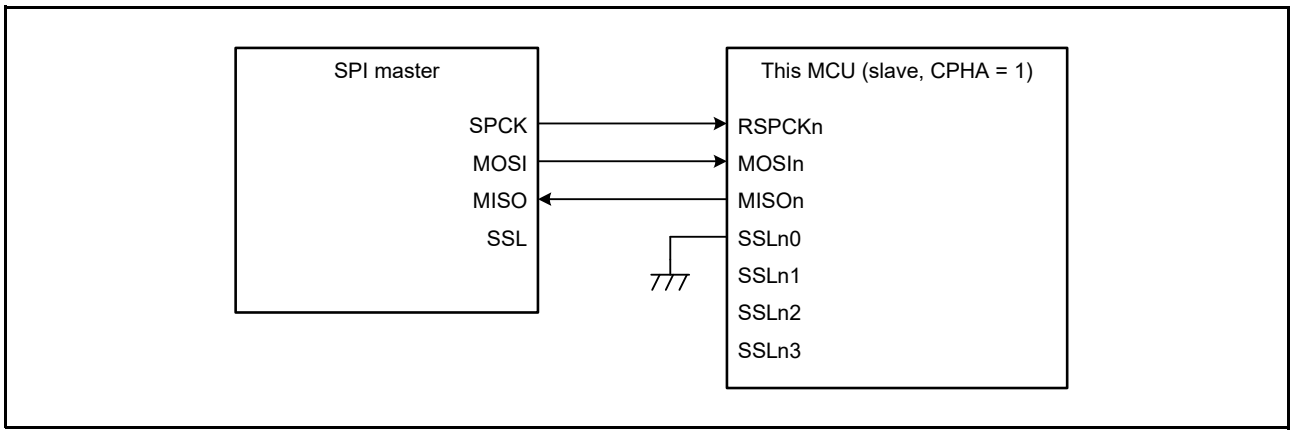


Figure 47.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

47.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 47.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 47.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of this MCU (master) are connected to the SPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOn input of this MCU (master). SSLn0 to SSLn3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKn, MOSIn, and SSLn0 to SSLn3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

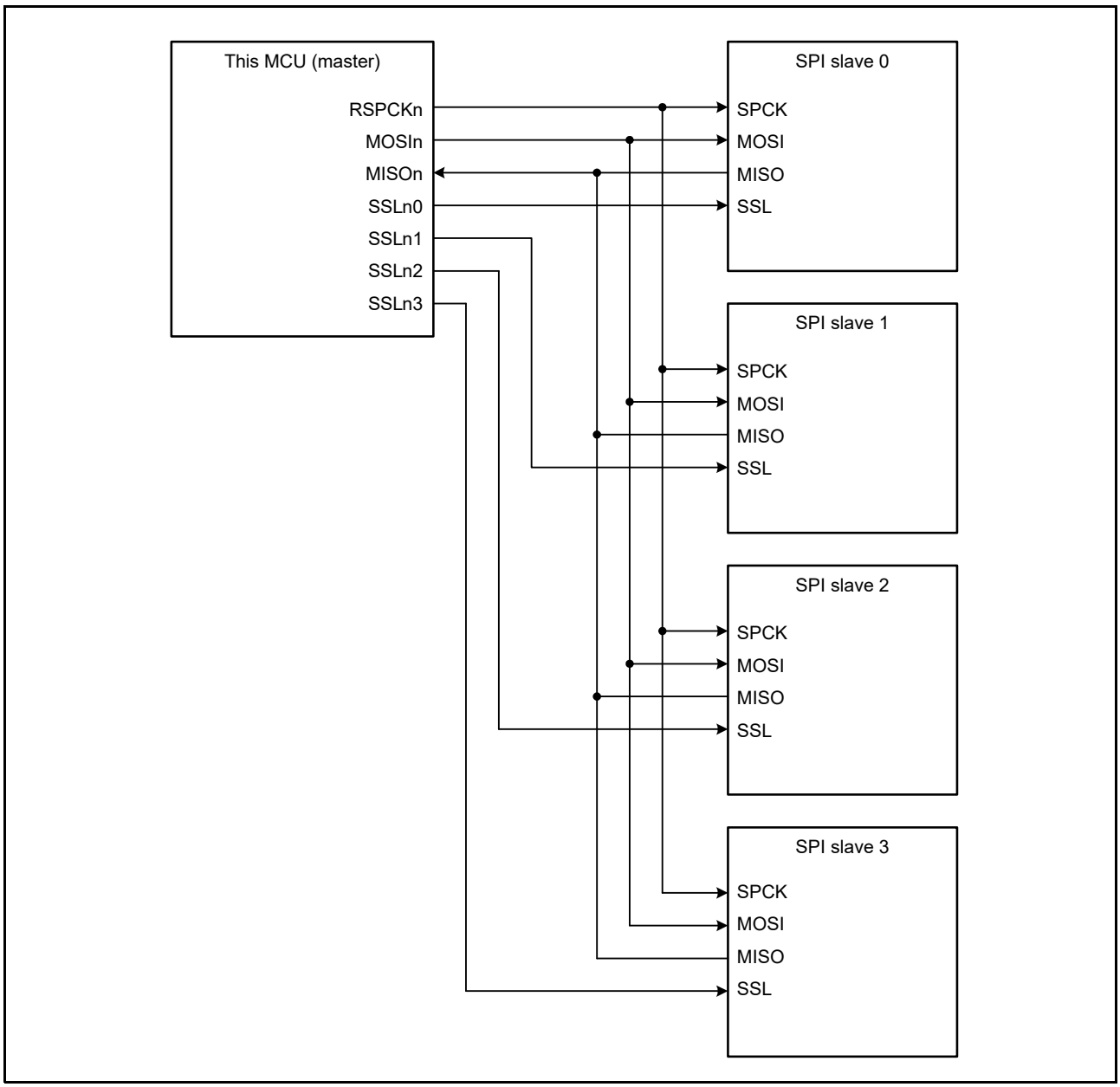


Figure 47.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

47.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 47.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 47.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y). The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slave X and slave Y). The MISOn outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slave X and slave Y), respectively.

The SPI master always drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLn0 input drives MISOn.

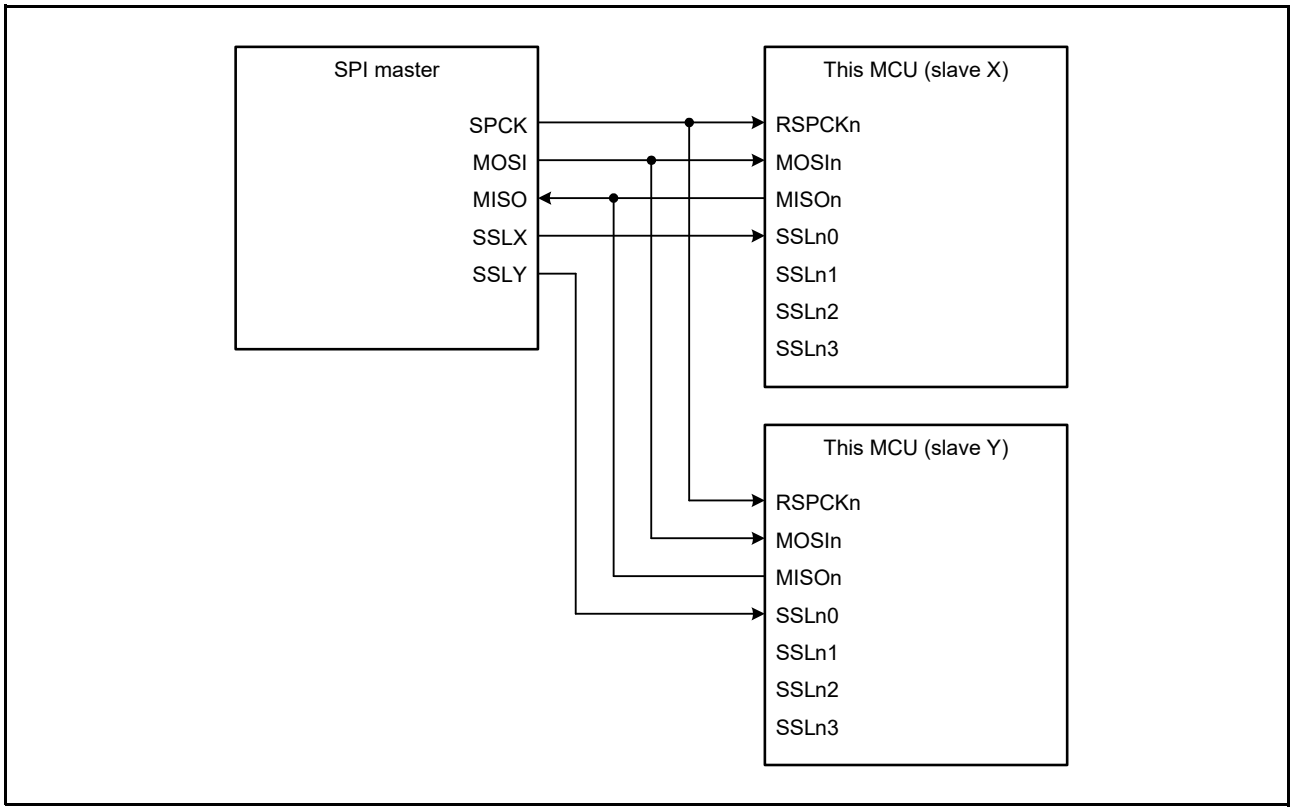


Figure 47.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

47.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 47.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 47.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKn and MOSIn outputs of the MCUs (master X and master Y) are connected to the SPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLn0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLn0 input of this MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of this MCU is not required.

This MCU drives RSPCKn, MOSIn, SSLn1, and SSLn2 when the SSLn0 input level is high. When the SSLn0 input level is low, this MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

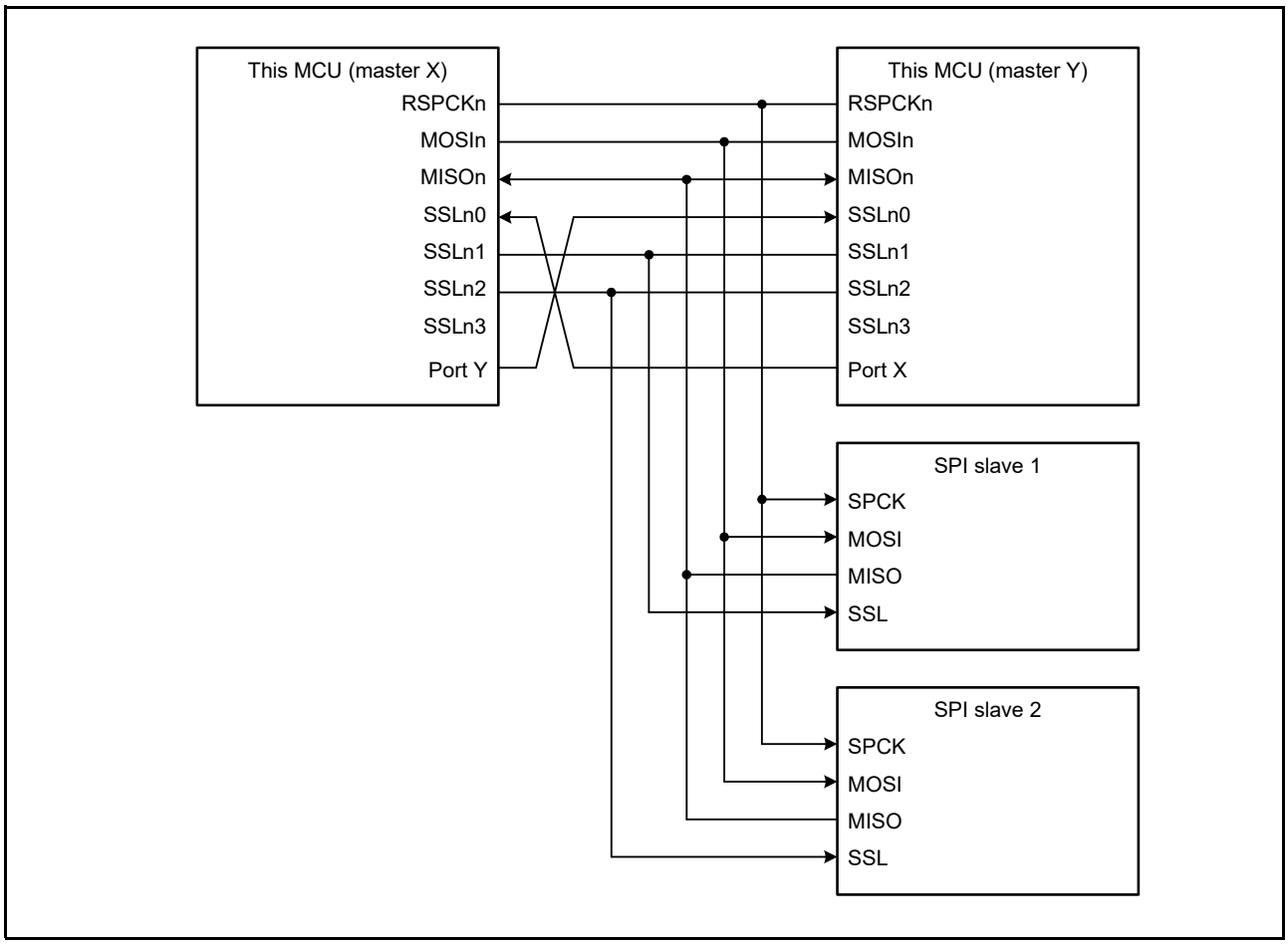


Figure 47.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SSLP.SSL0P of the other master
- End of communication: the inverted value of SSLP.SSL0P of the other master

47.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 47.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLn0 to SSLn3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKn and MOSIn. The SPI slave drives the MISO.

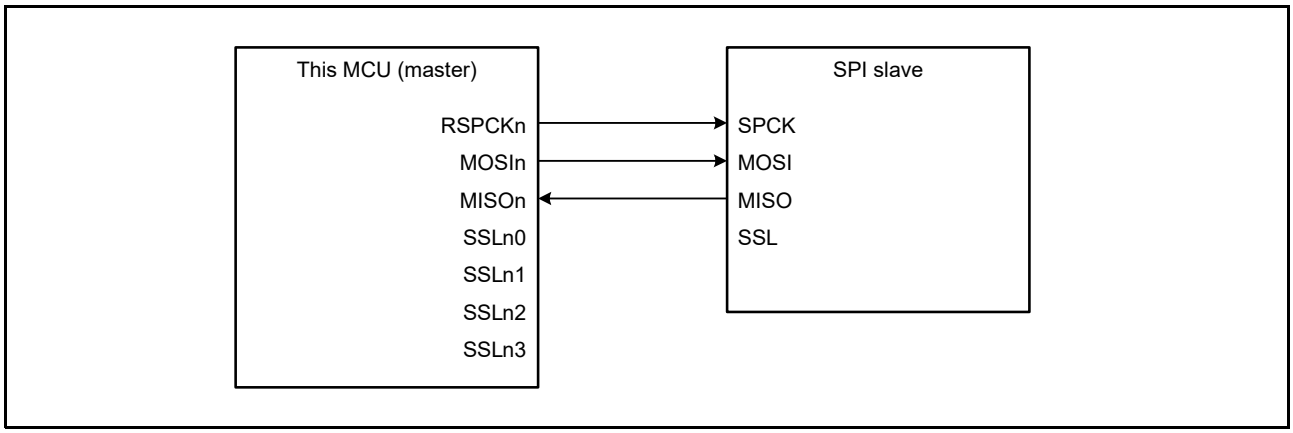


Figure 47.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

47.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 47.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOn and the SPI master drives the SPCK and MOSI. In addition, SSLn0 to SSLn3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

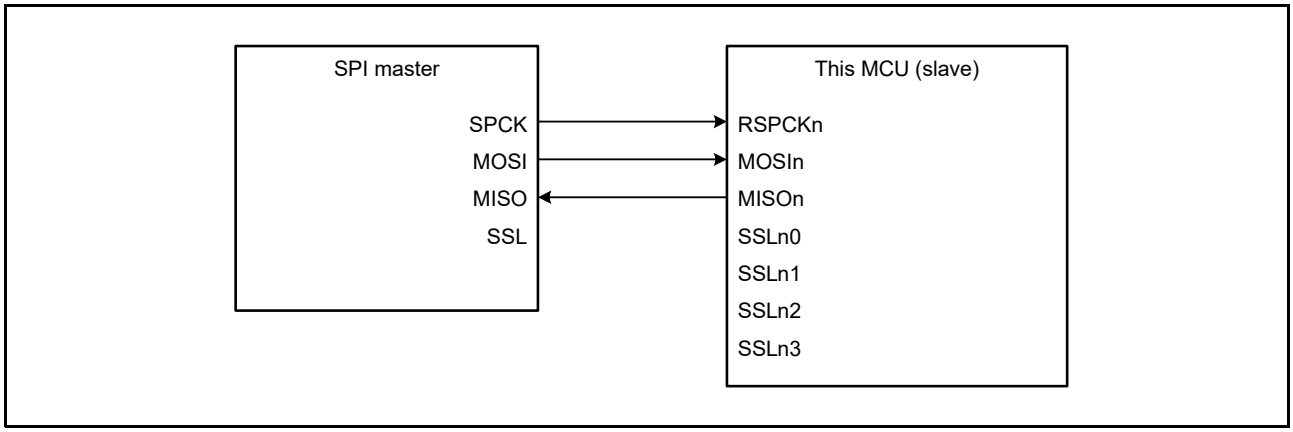


Figure 47.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

47.3.4 Data Format

The RSPIA's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the RSPIA treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transmit data.

47.3.4.1 Data Format of One Frame

The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]).

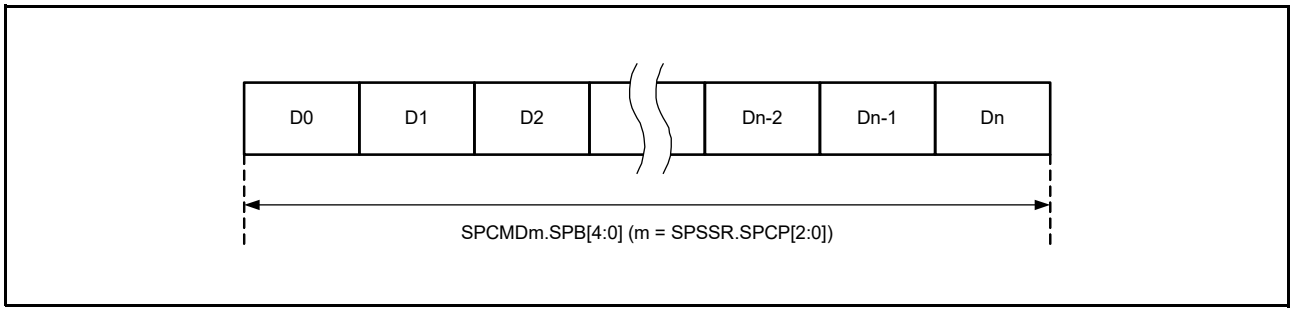


Figure 47.13 Outline of the Data Format (with Parity Disabled)

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

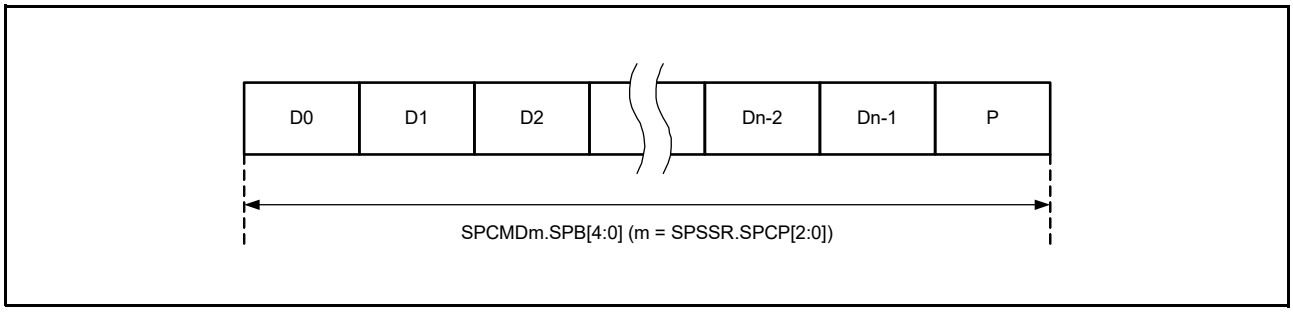


Figure 47.14 Outline of the Data Format (with Parity Enabled)

47.3.4.2 When Parity is Disabled (SPCR.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

(1) MSB First Transfer (32-Bit Data)

Figure 47.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

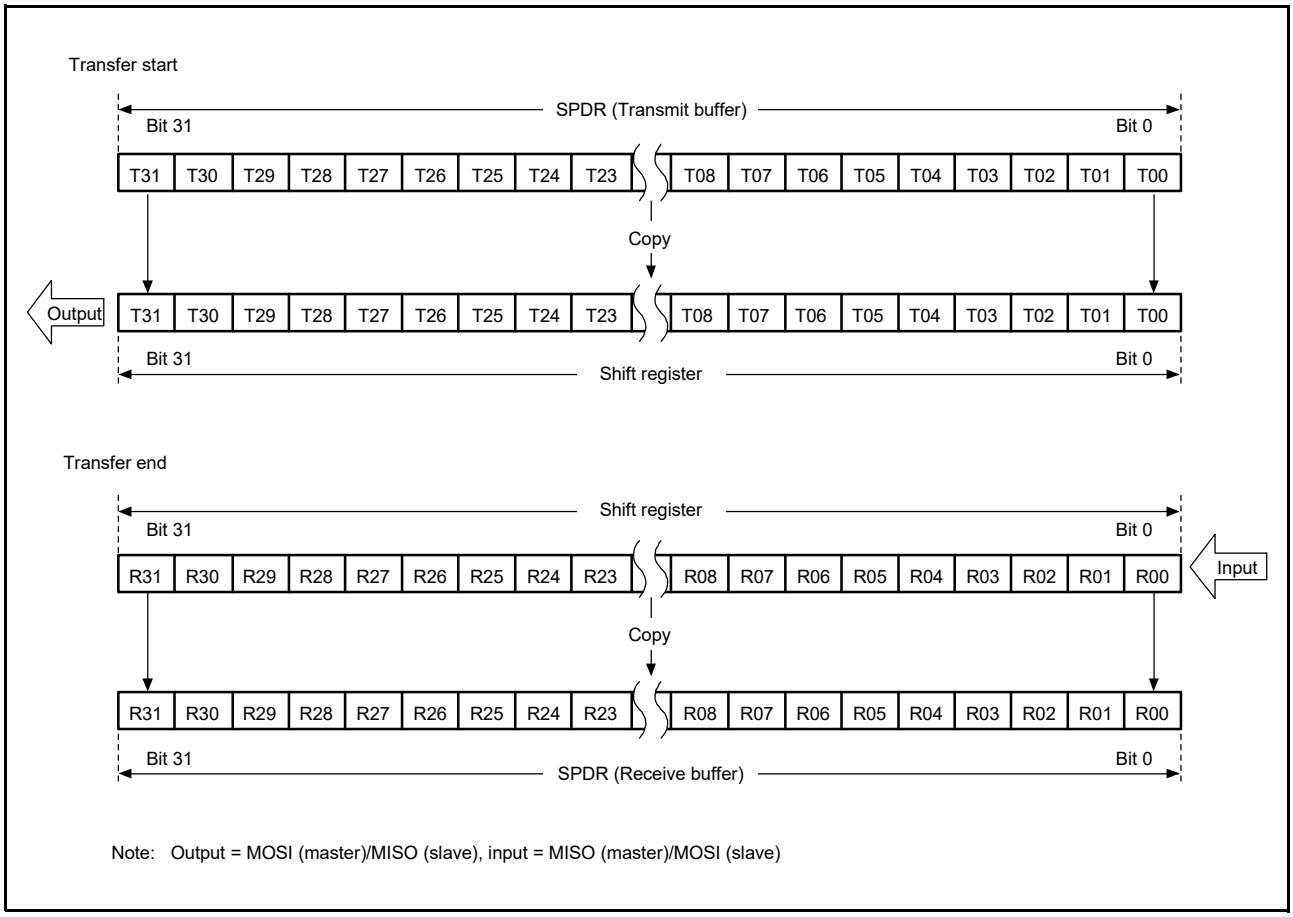


Figure 47.15 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 47.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

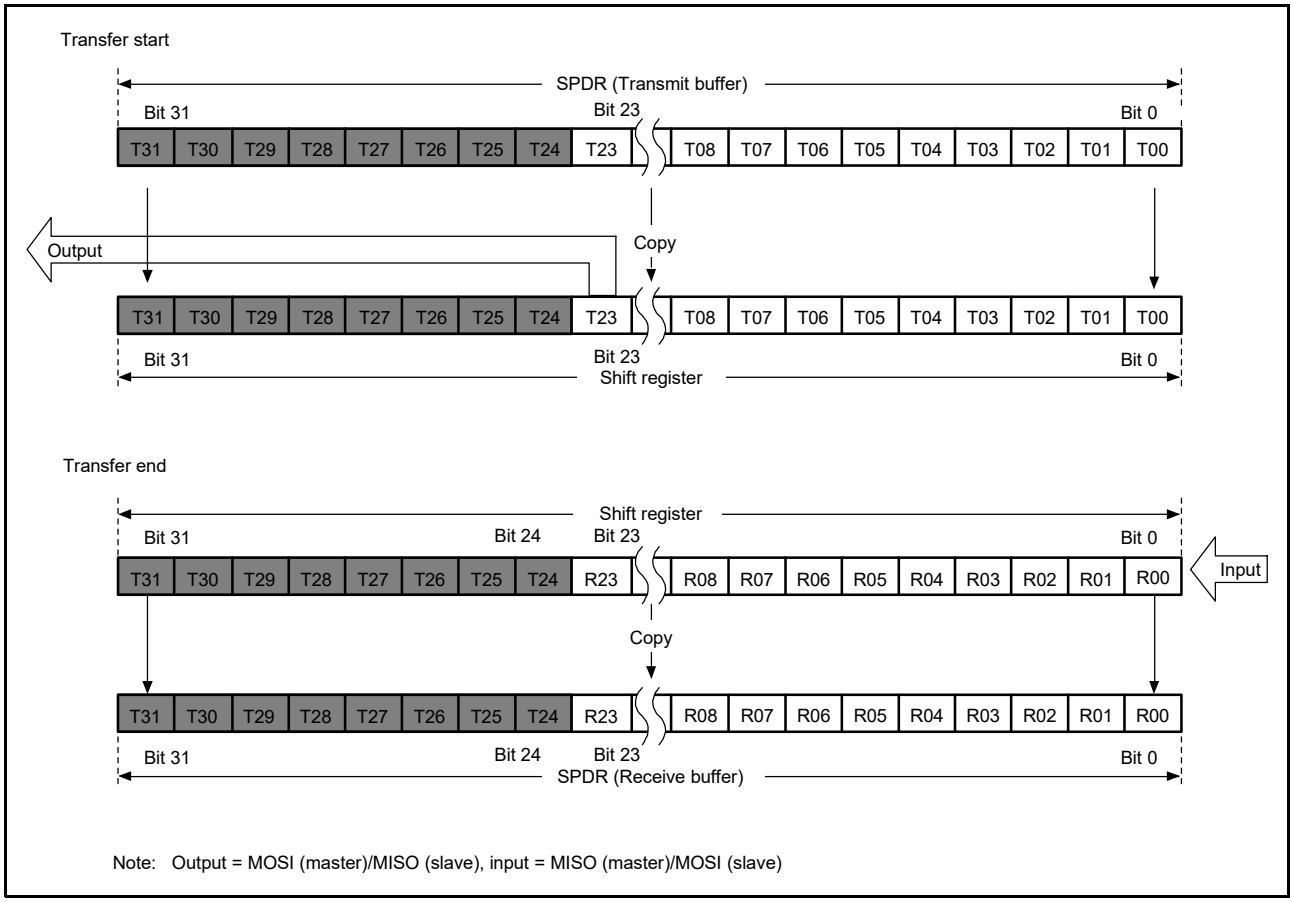


Figure 47.16 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 47.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order R31 to R00 for copying to the receive buffer.

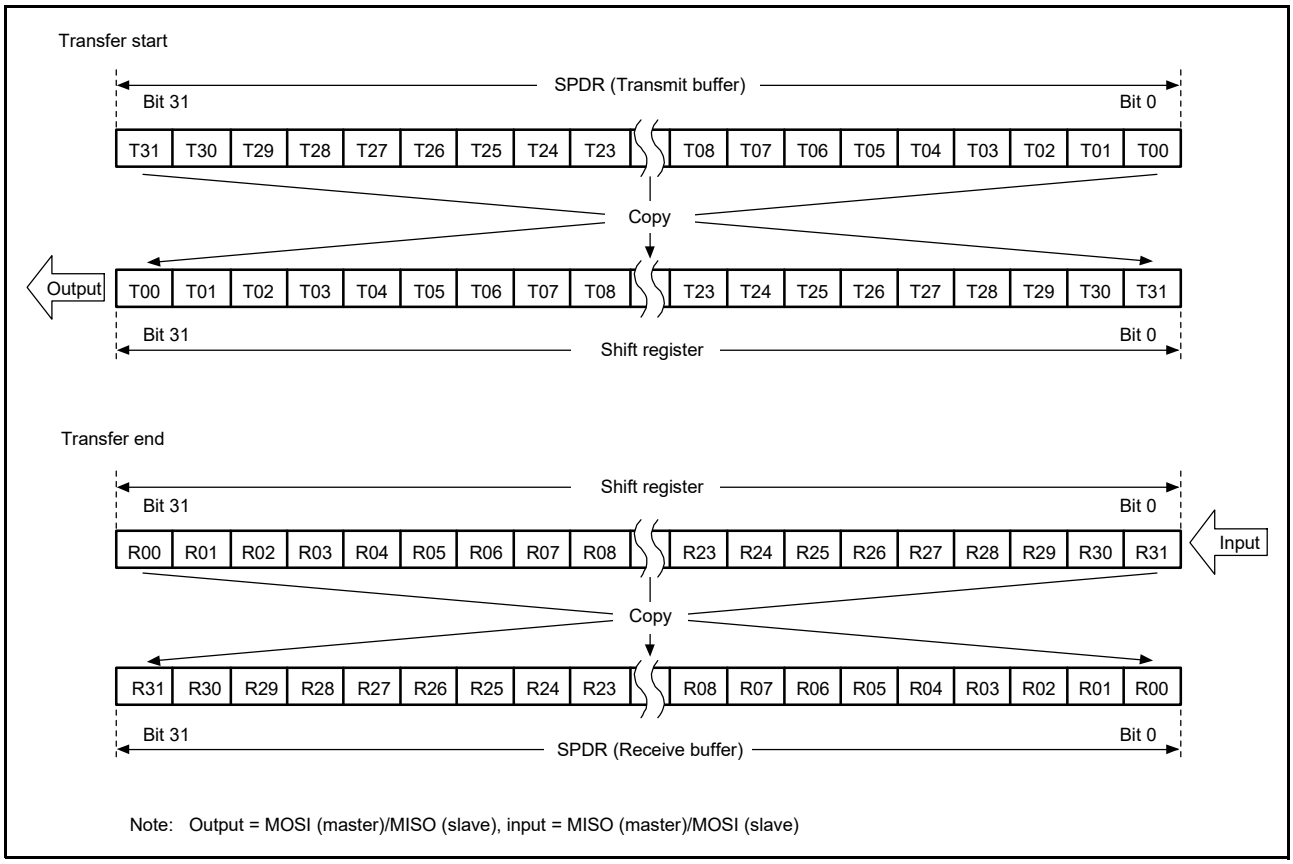


Figure 47.17 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 47.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order R23 to R00 for copying to the receive buffer.

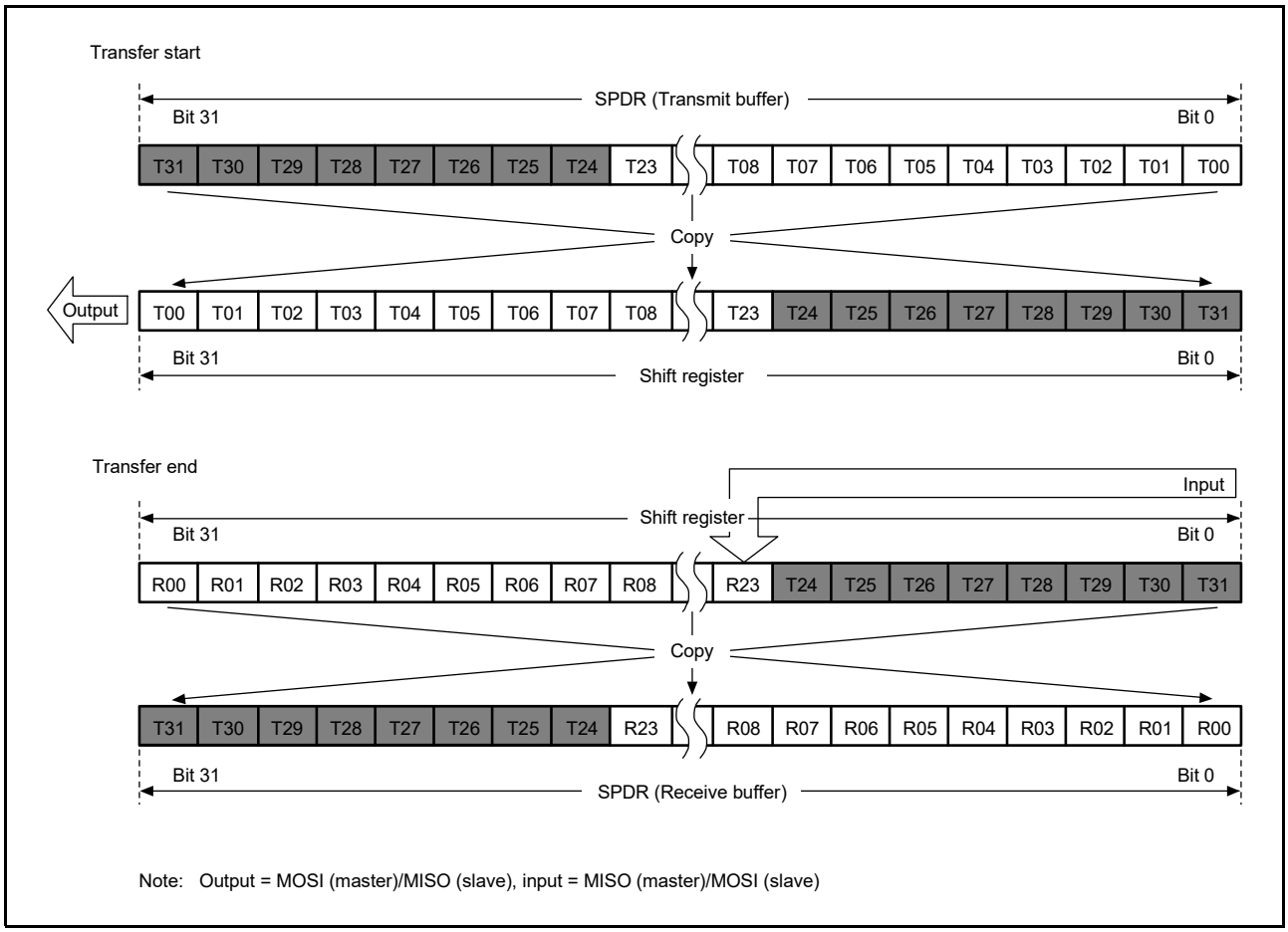


Figure 47.18 LSB First Transfer (24-Bit Data, Parity Disabled)

47.3.4.3 When Parity is Enabled (SPCR.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB First Transfer (32-Bit Data)

Figure 47.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the receive buffer, the data from R31 to P are checked to judge the parity error.

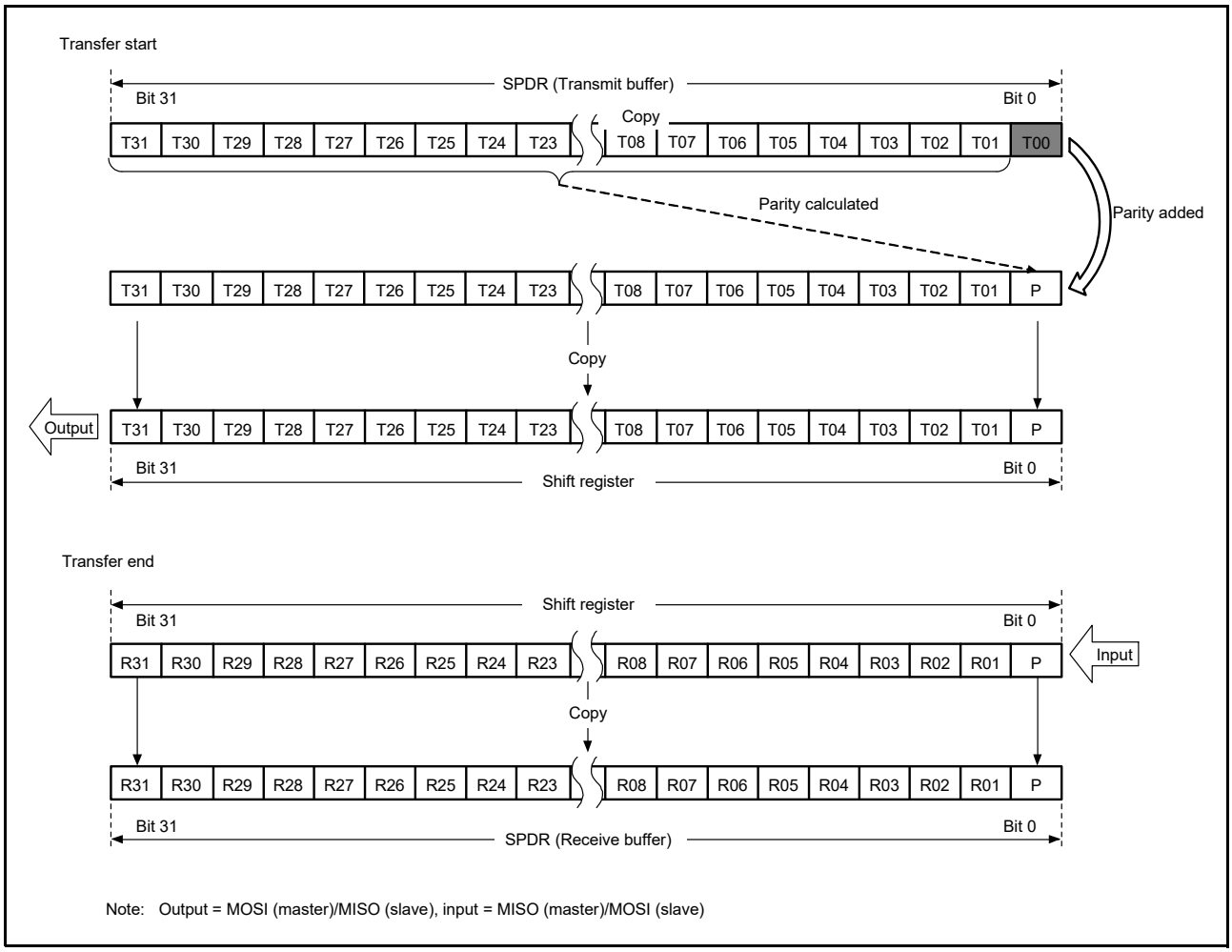


Figure 47.19 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 47.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the receive buffer, the data from R23 to P are checked to judge the parity error.

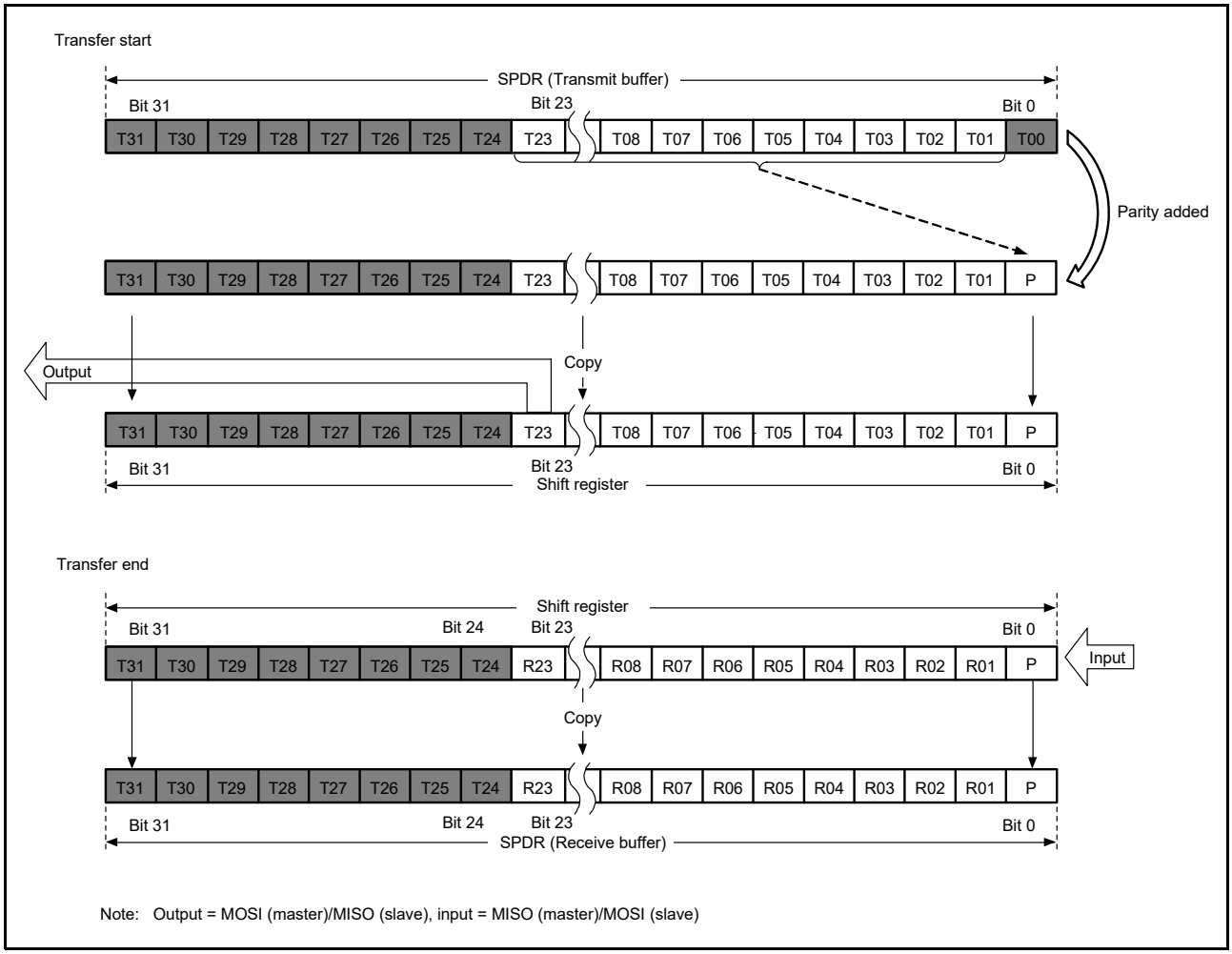


Figure 47.20 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 47.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P to R00 for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

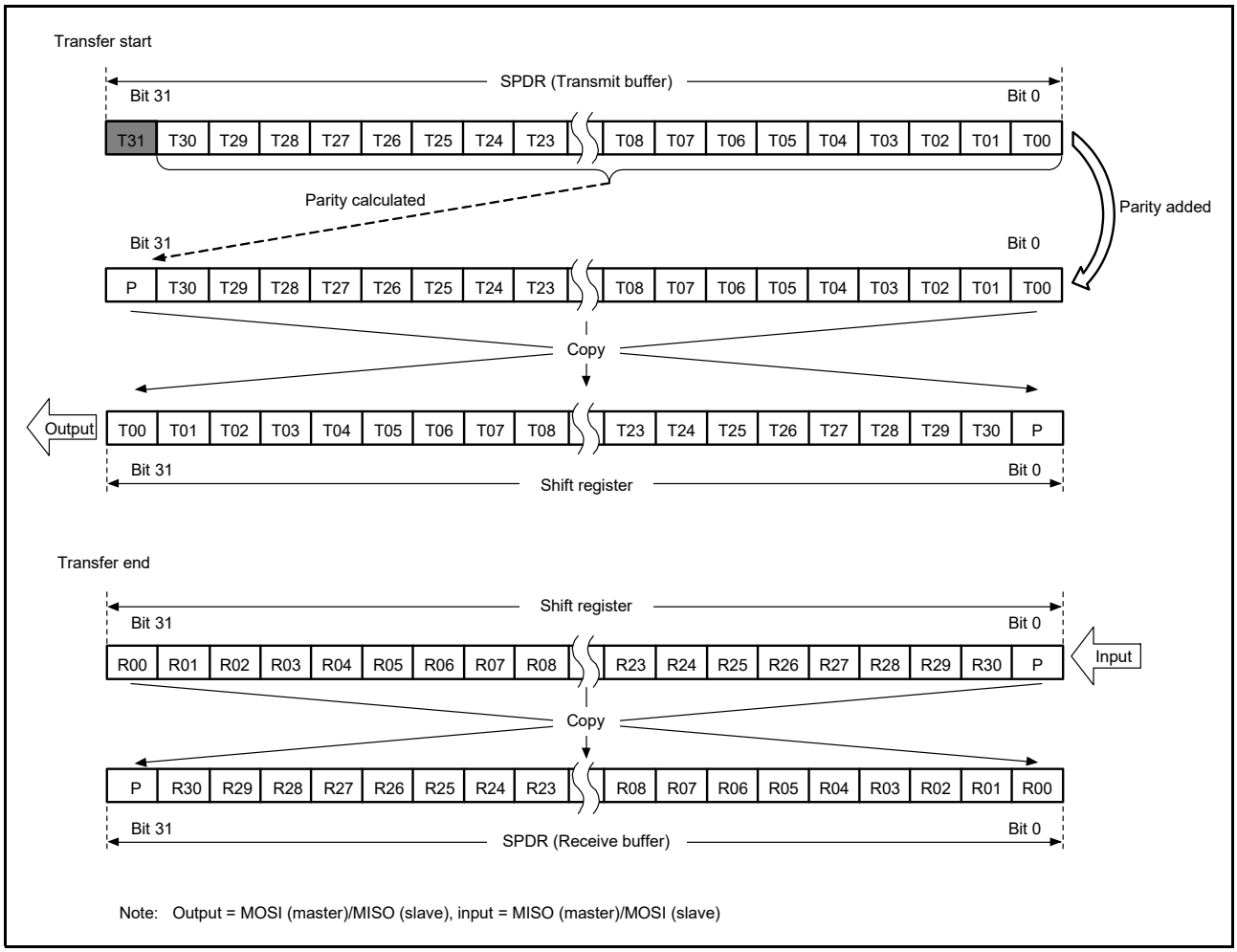


Figure 47.21 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 47.22 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P (bit 23) to R00 (bit 0) for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

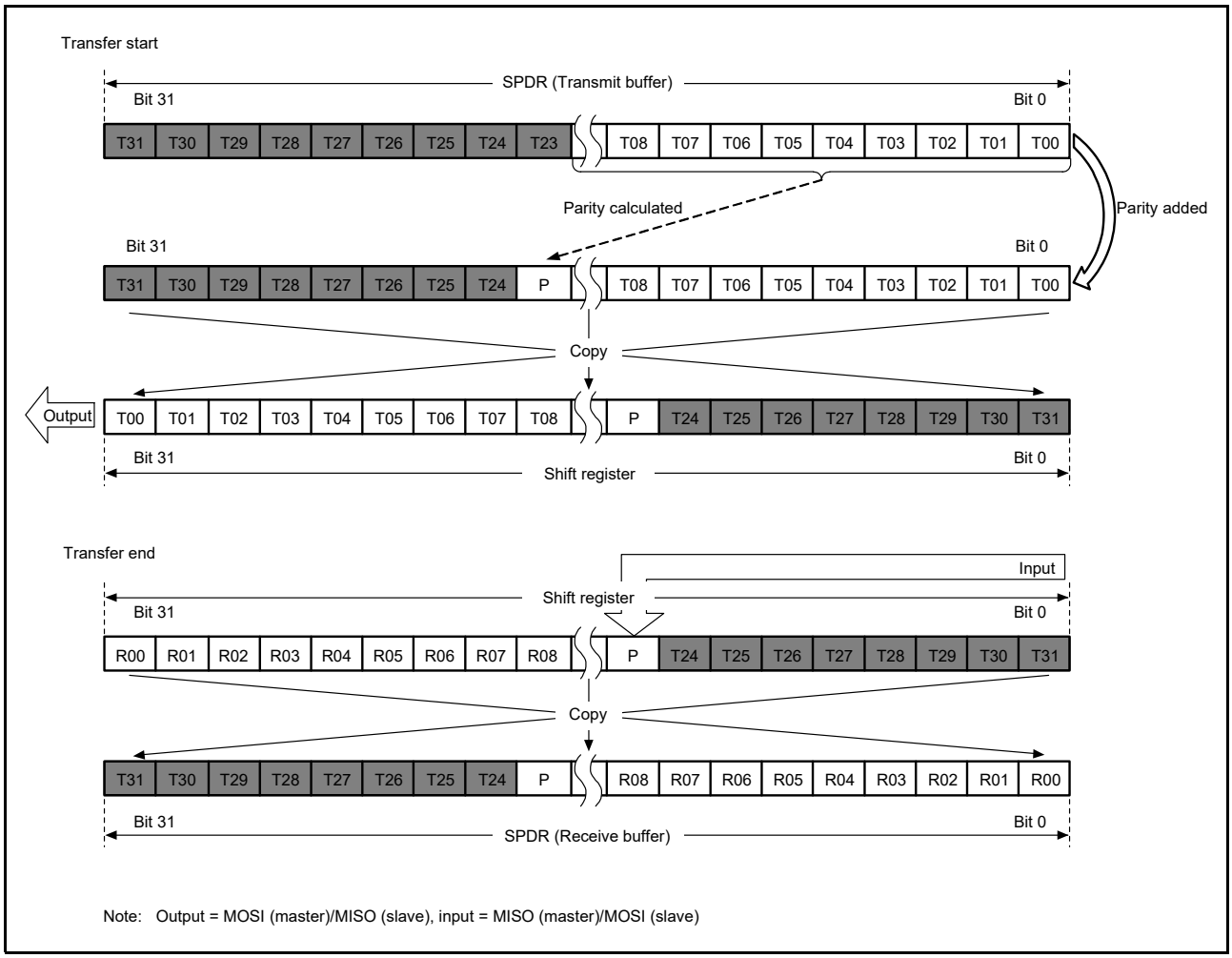


Figure 47.22 LSB First Transfer (24-Bit Data, Parity Enabled)

47.3.4.4 Byte Swap Transmission

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes written in the transmit buffer (SPDR) will be swapped (in 8 bit units) when the data is transferred to the shift register. Figure 47.23 shows data transfer between the SPDR register and the shift register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31, T30, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 3 [T31 to T24].
Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08, T23, T22, …, T16, T31, T30, …, and T24 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 3 [T24 to T31].
Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T31 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 3 [T24 to T31] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T24, T25, …, T31, T16, T17, …, T23, T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

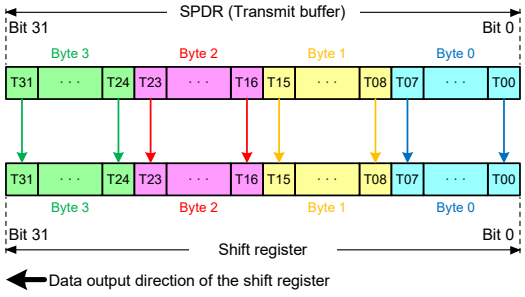
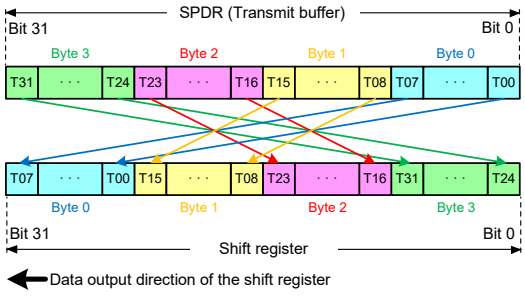
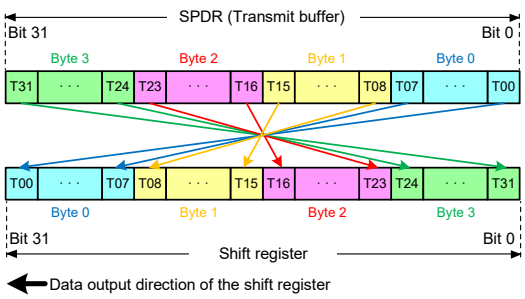
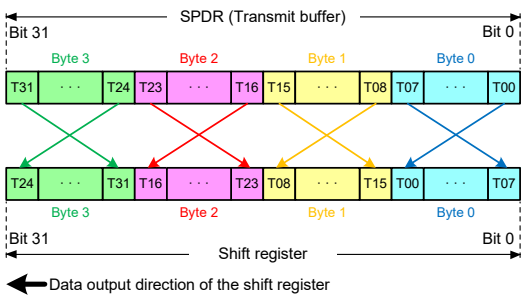
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 47.23 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 47.24 shows data transfer between the SPDR register and the shift register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte 1 [T15 to T08] to Byte 0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15, T14, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T15 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 1 [T08 to T15] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

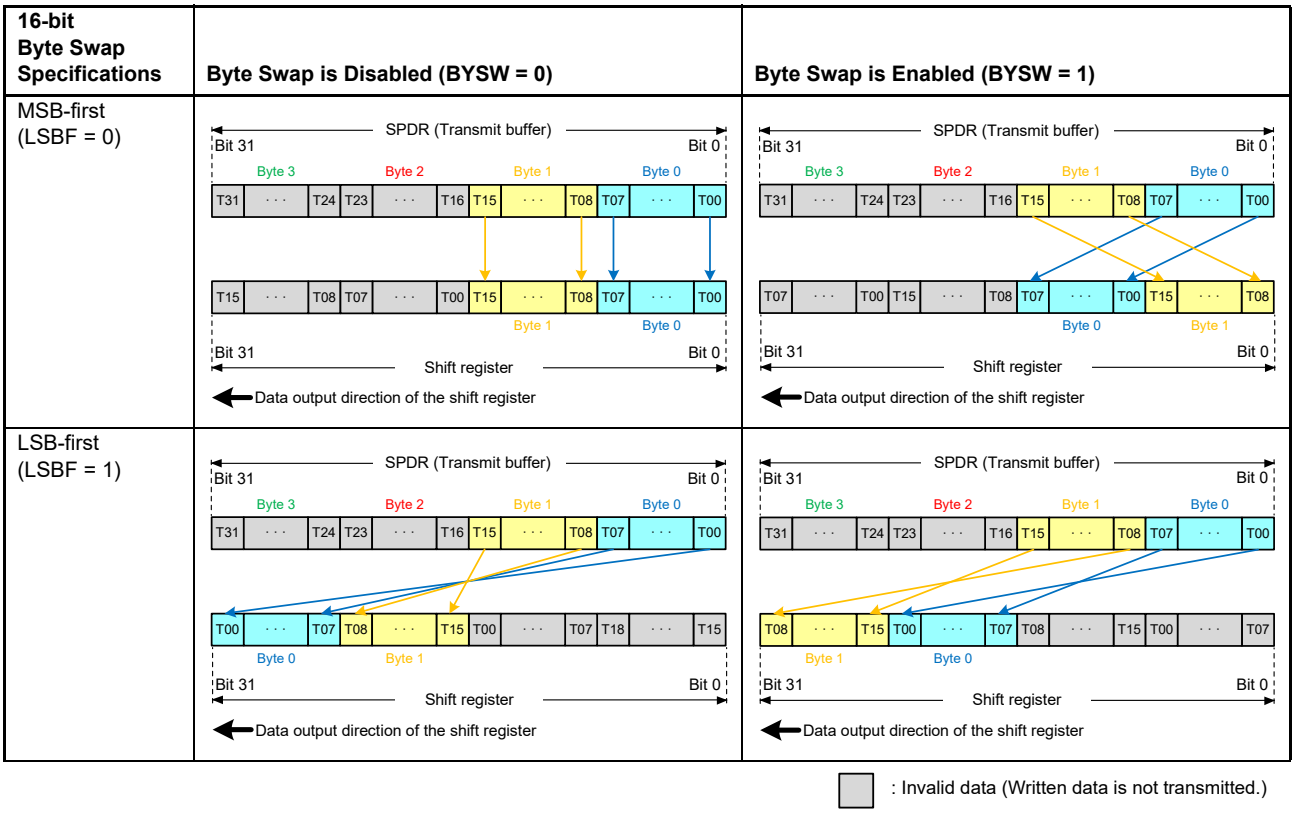


Figure 47.24 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.

Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation is not guaranteed.
Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

47.3.4.5 Byte Swap Reception

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes in the shift register will be swapped (in 8 bit units) when the data is transferred to the receive buffer (SPDR). Figure 47.25 shows data transfer between the shift register and the SPDR register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31, R30, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, R08, R23, R22, …, R16, R31, R30, …, and R24. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00, R01, …, and R31. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24, R25, …, R31, R16, R17, …, R23, R08, R09, …, R15, R00, R01, …, and R07. When necessary RSPCK cycles are input and data is stored from Byte 3 [R24 to R31] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

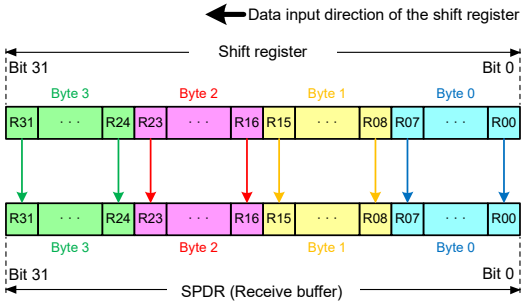
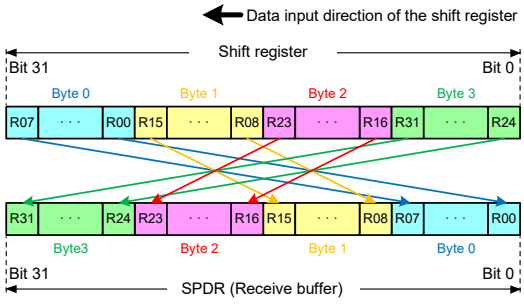
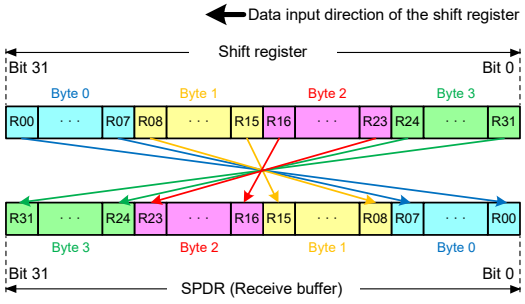
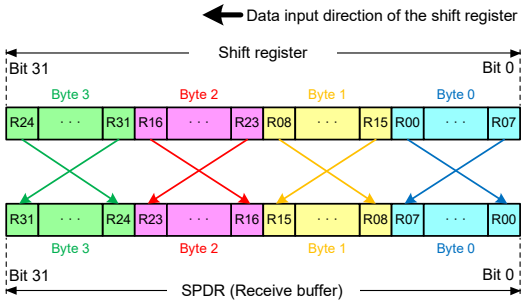
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 47.25 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 47.26 shows data transfer between the shift register and the SPDR register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15, R14, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, and R08. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00, R01, …, R07, R08, R09, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08, R09, …, R15, R00, R01, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 1 [R08 to R15] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

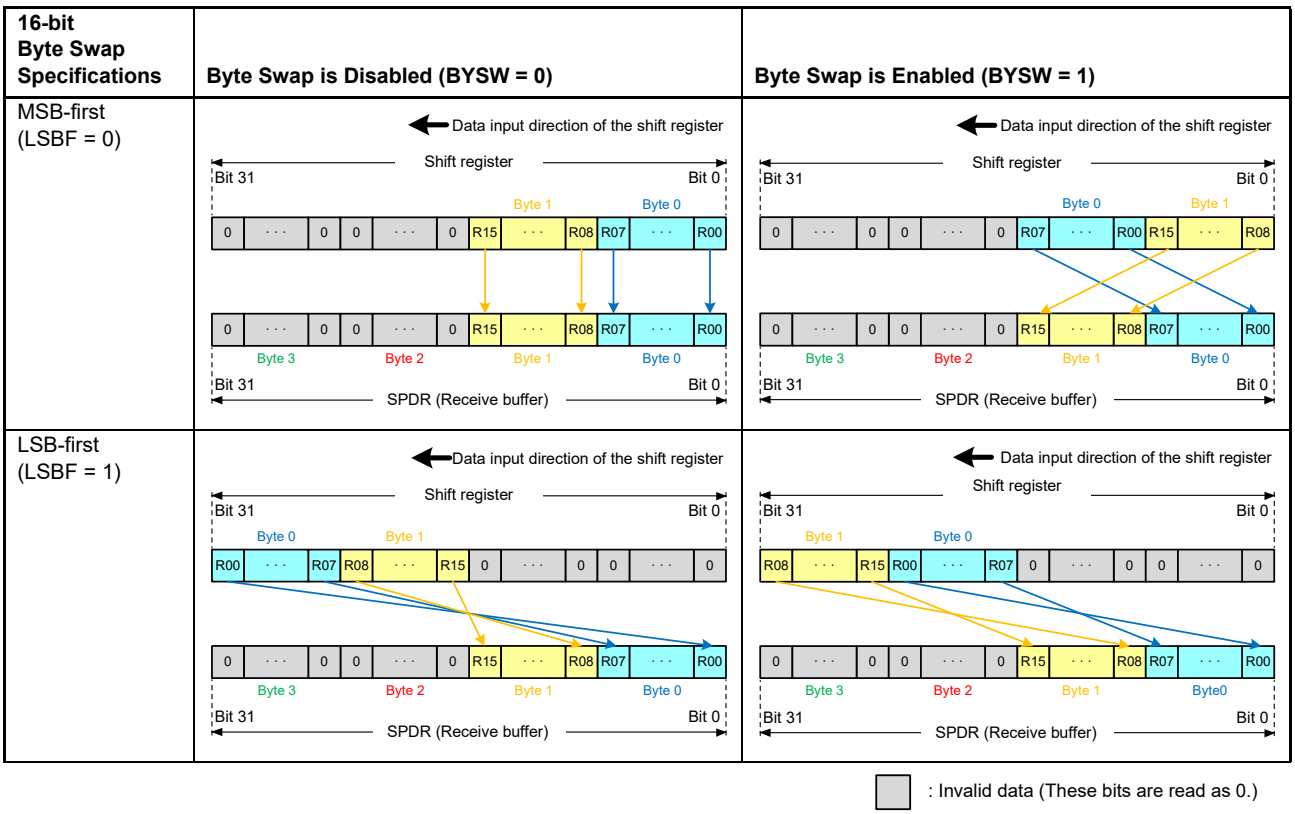


Figure 47.26 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.

Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation is not guaranteed.

Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, subsequent operation is not guaranteed.

47.3.5 Transfer Format (Frame Format)

47.3.5.1 When CPHA = 0

Figure 47.27 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPIA operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 47.27, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPIA settings. For details, refer to section 47.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals commences at an SSLni signal assertion timing. The first RSPCKn signal change timing that occurs after the SSLni signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKn signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLni signal assertion to RSPCKn oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKn oscillation to an SSLni signal negation (SSL negation delay). t3 denotes a period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPIA of this MCU is in master mode, refer to section 47.3.13.1, Master Mode Operation.

[Motorola SPI]

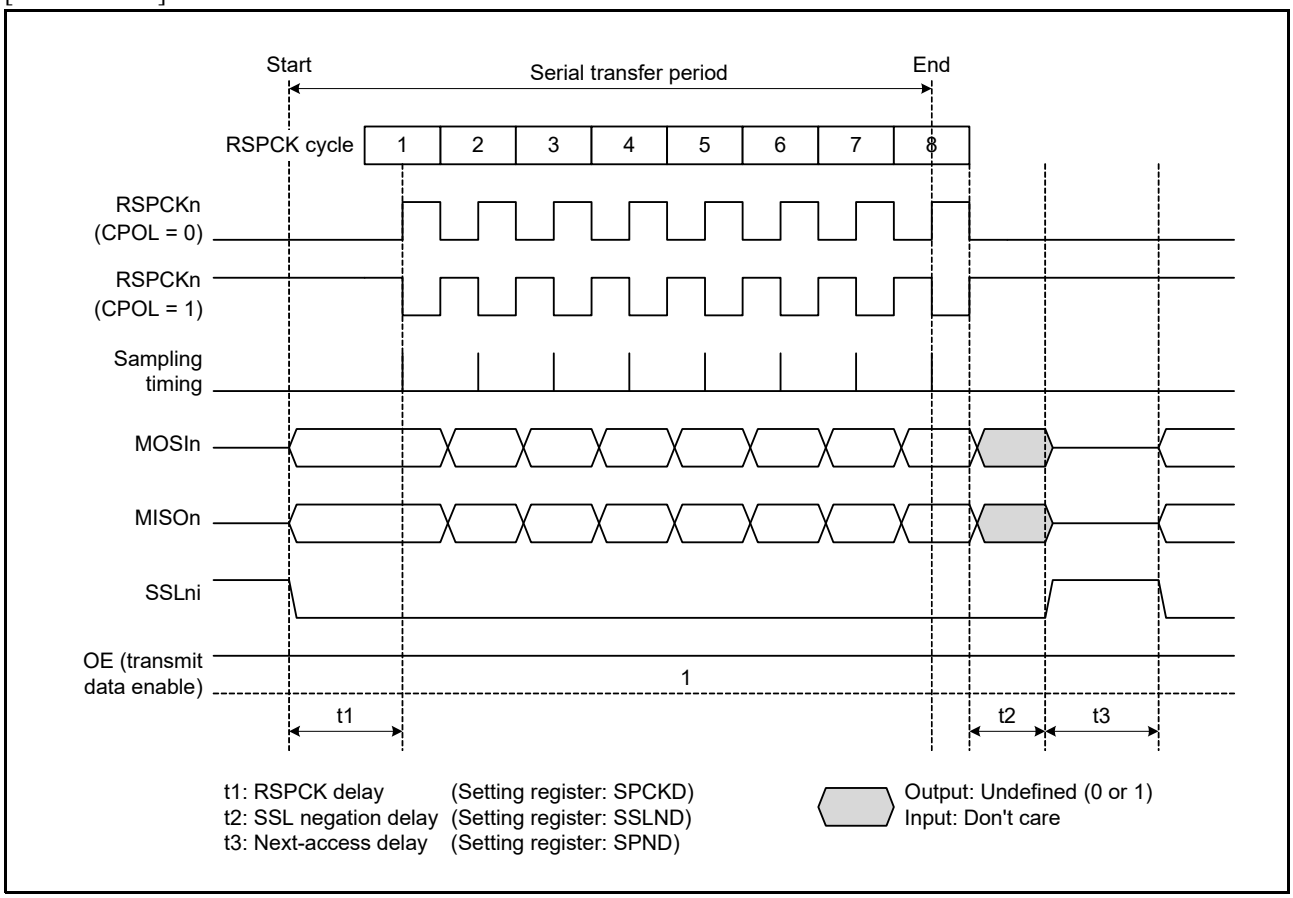


Figure 47.27 RSPI Transfer Format (CPHA = 0, FRFS = 0)

[TI SSP]

In TI SSP protocol, setting the CPHA bit to 0 is not effective.

47.3.5.2 When CPHA = 1

Figure 47.28 and Figure 47.29 show sample transfer formats for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 47.28, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 47.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal commences at an SSLni signal assertion timing. The output of valid data to the MOSIn and MISOn signals commences at the first RSPCKn signal change timing that occurs after the SSLni signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPIA of this MCU is in master mode, refer to section 47.3.13.1, Master Mode Operation.

[Motorola SPI]

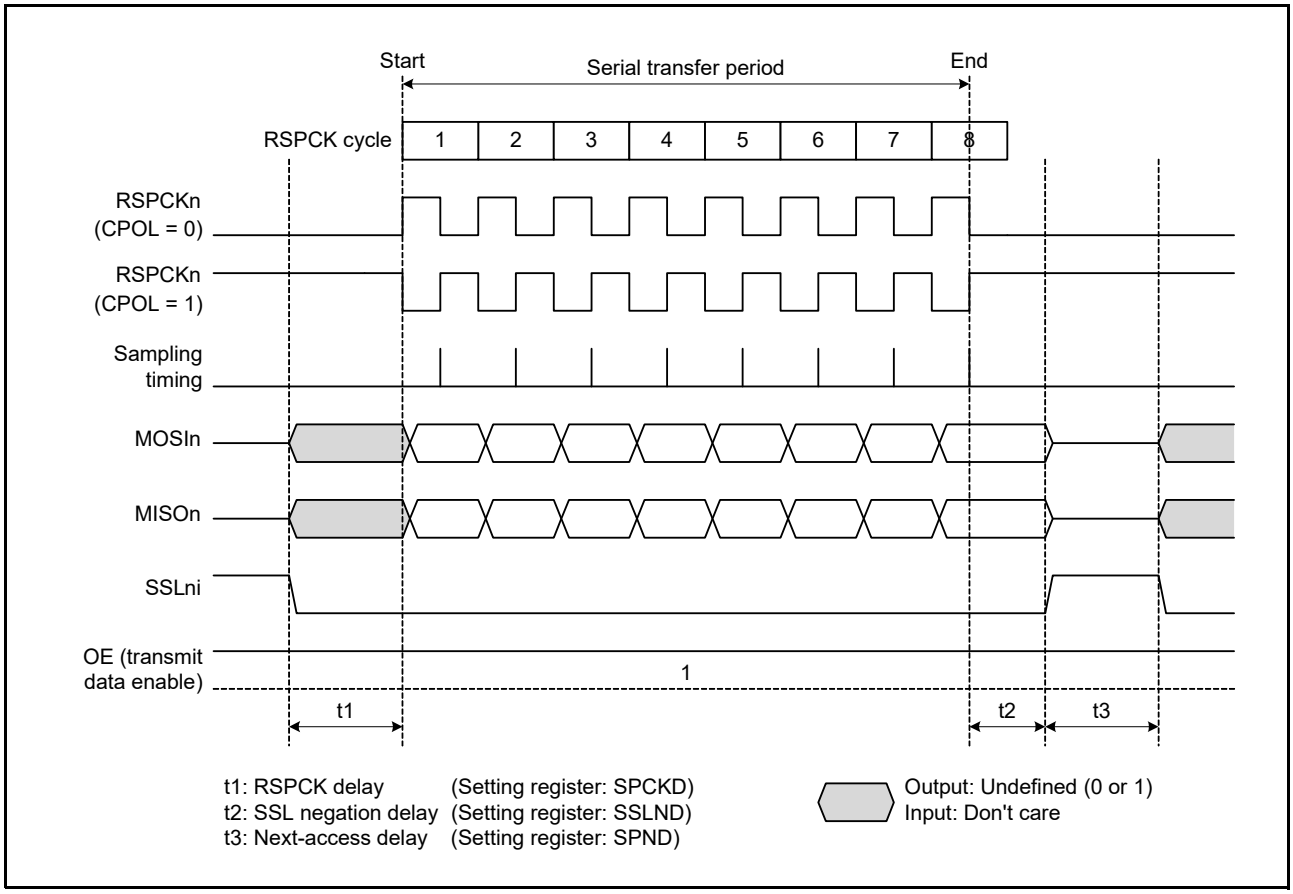


Figure 47.28 RSPI Transfer Format (CPHA = 1, FRFS = 0)

[TI SSP]

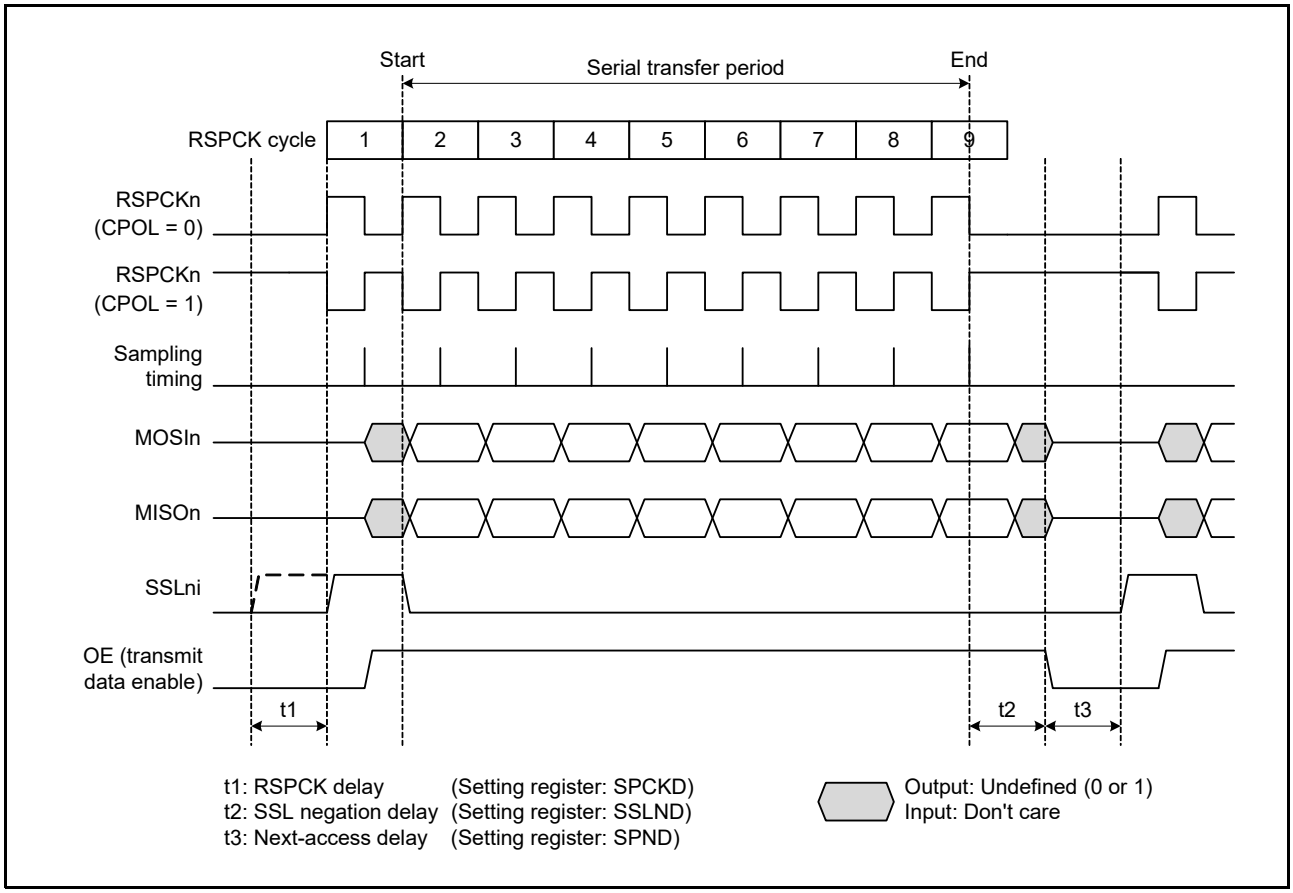


Figure 47.29 RSPI Transfer Format (CPHA = 1, FRFS = 1)

47.3.6 Communication Mode

Transmit-receive mode, transmit-only mode, and receive-only mode can be selected by the communication mode select bit (SPCR.CMMD[1:0]).

The SPDR register access shown in Figure 47.30, Figure 47.31 and Figure 47.32 indicate the condition of access to the SPDR register, where W denotes a write cycle.

47.3.6.1 Transmit-Receive Mode (SPCR.CMMD[1:0] = 00b)

Figure 47.30 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 00b. In the example in Figure 47.30, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

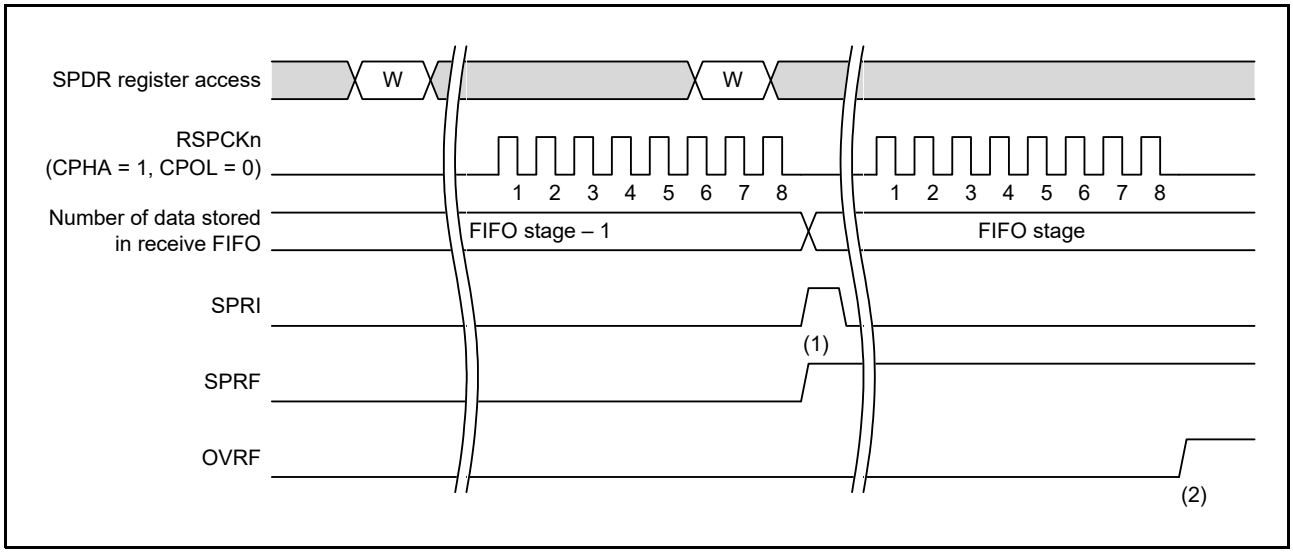


Figure 47.30 Operation Example of SPCR.CMMD[1:0] = 00b

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When transmit-receive mode (CMMD[1:0] = 00b) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

47.3.6.2 Transmit-Only Mode (SPCR.CMMD[1:0] = 01b)

Figure 47.31 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 01b. In the example in Figure 47.31, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = 00b, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

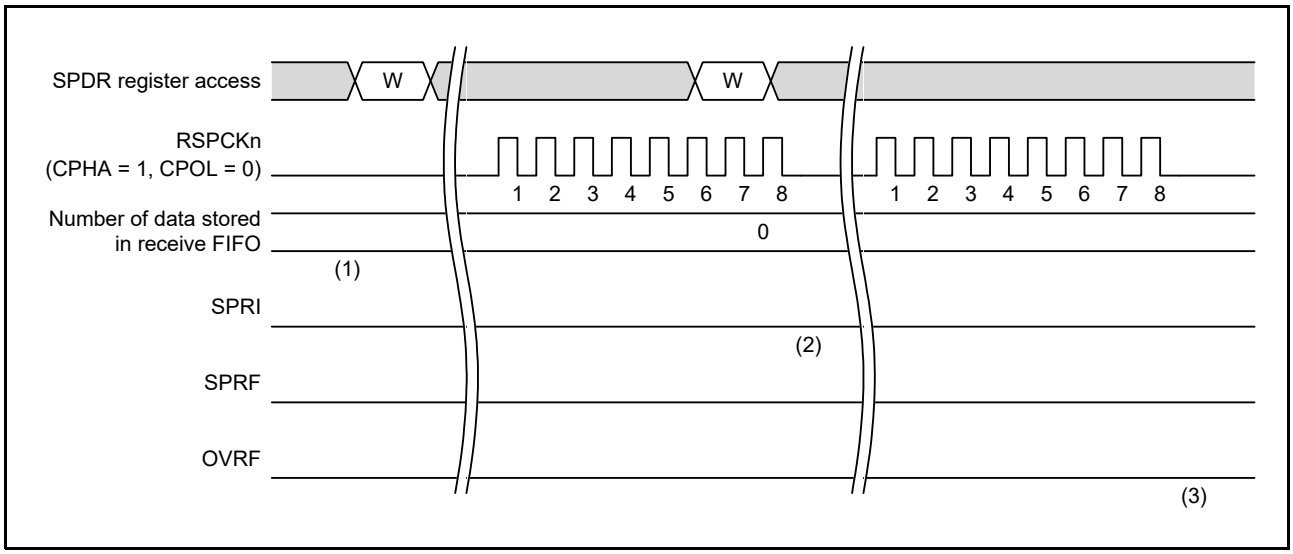


Figure 47.31 Operation Example of SPCR.CMMD[1:0] = 01b

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.CMMD[1:0] = 01b).

(2) When a serial transfer ends without received data in the receive FIFO of SPDR, if transmit-only mode is selected (SPCR.CMMD[1:0] = 01b), the SPRF flag remains 0 and the RSPIA does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When transmit-only mode (SPCR.CMMD[1:0] = 01b) is selected, the RSPIA transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

47.3.6.3 Receive-Only Mode (SPCR.CMMD[1:0] = 10b)

Figure 47.32 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 10b. In the example in Figure 47.32, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

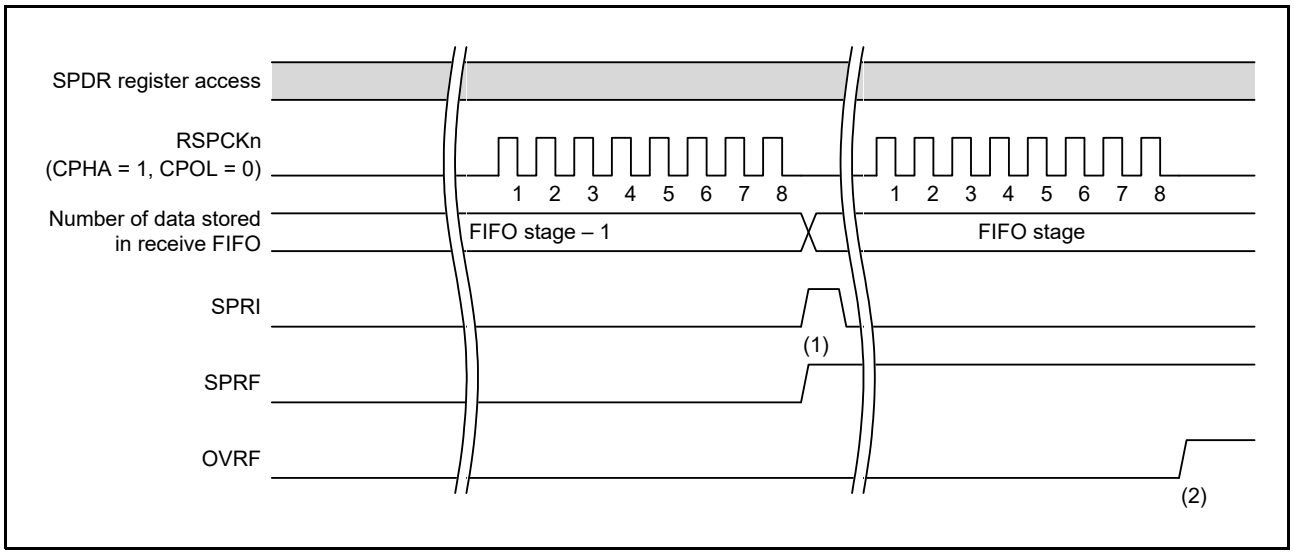


Figure 47.32 Operation Example of SPCR.CMMD[1:0] = 10b

The operation of the flags at timings shown in steps (1) to (2) in the figure is described below.

(1) When a serial transfer ends while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

47.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 47.33 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 47.33 indicate the conditions of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the examples in Figure 47.33, the RSPIA performs an 8-bit serial transfer in which the SPCR.CMMD[1:0] bits are 00b, SPFCR.TTRG[1:0] bits are 00b, the SPFCR.RTRG[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

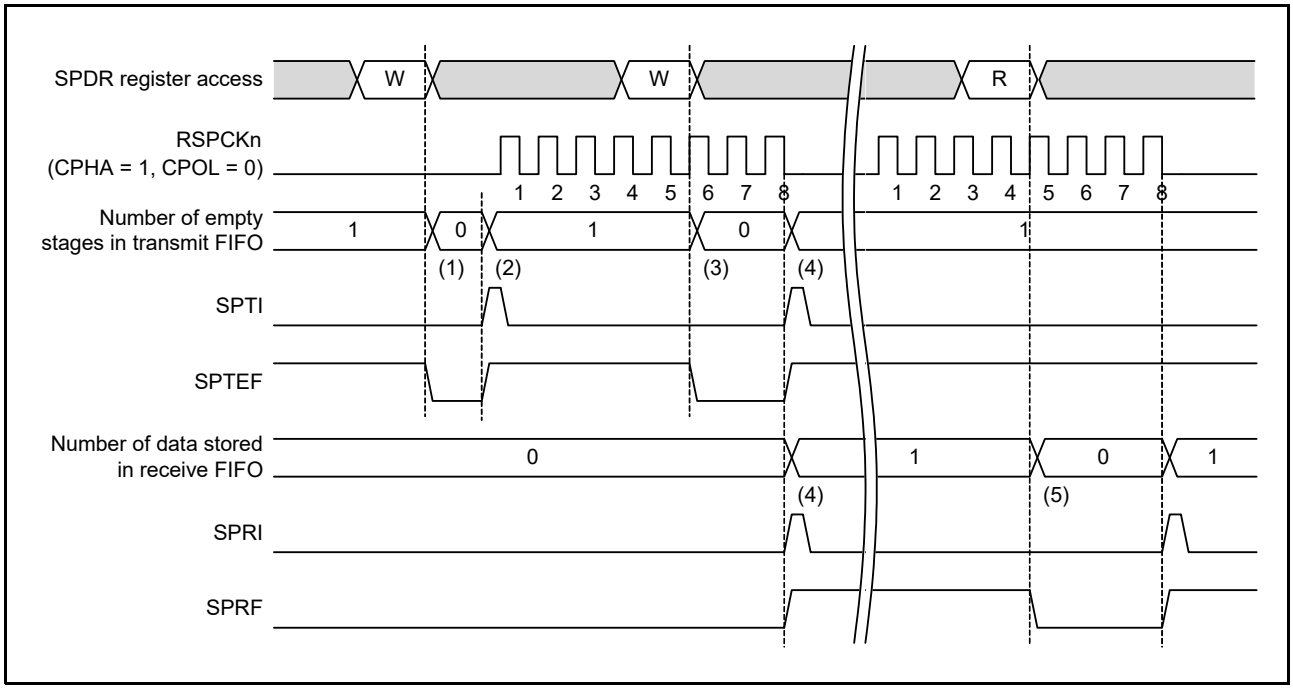


Figure 47.33 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) If transmit data is written to SPDR when the next transmit data is not set in the transmit FIFO, the RSPIA writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access.

(2) If the shift register is empty, the RSPIA copies the data from the transmit buffer to the shift register. If the number of empty stage of the transmit FIFO is greater than the threshold value, the RSPIA generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPIA. For details, refer to section 47.3.13, SPI Operation, and section 47.3.14, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access. Because the data being transmitted is stored in the shift register, the RSPIA does not copy the data from the transmit buffer to the shift register.

(4) If the serial transfer ends (data sampling clock edge in the final bit is detected) when the data stored in the receive FIFO of SPDR < the number of FIFO stages, the RSPIA copies the receive data from the shift register to the receive buffer. At this time, if the data stored in the receive FIFO is greater than the threshold value, the RSPIA generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of the serial transfer, when the next transmit data had been set in the transmit FIFO before the serial transfer ended, the RSPIA sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun

error status, upon completion of the serial transfer, the RSPIA determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt (SPRI) routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC/DMAC, the SPRF flag becomes 0 at the last access.

If transmit data is written to SPDR while no empty stages in the transmit FIFO, the RSPIA does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag.

When setting the SPCR.SPE bit to 0 (RSPI function disabled), if the SPCR.SPTIE is 1, a transmit buffer empty interrupt request will occur. However, transmit buffer empty interrupt is inhibited by disabling transmit buffer empty interrupt (the SPTIE bit is 0) at the same time as SPE bit setting.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the RSPIA does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 47.3.10, Error Detection). To prevent a receive data overrun error, read the received data before the next serial transfer ends.

Transmit buffer empty interrupt and receive buffer full interrupt or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

47.3.8 Idle Interrupts

When the SPSSR.SPCP[2:0] bits becomes 000b, if the next transmit data is not set, the SPSR.IDLNF flag is set to 0 and an idle interrupt request is generated. An interrupt request is also made by setting the SPCR.SPE bit to 0.

[Motorola SPI]

Figure 47.34 shows an example of idle interrupt operation during normal operation.

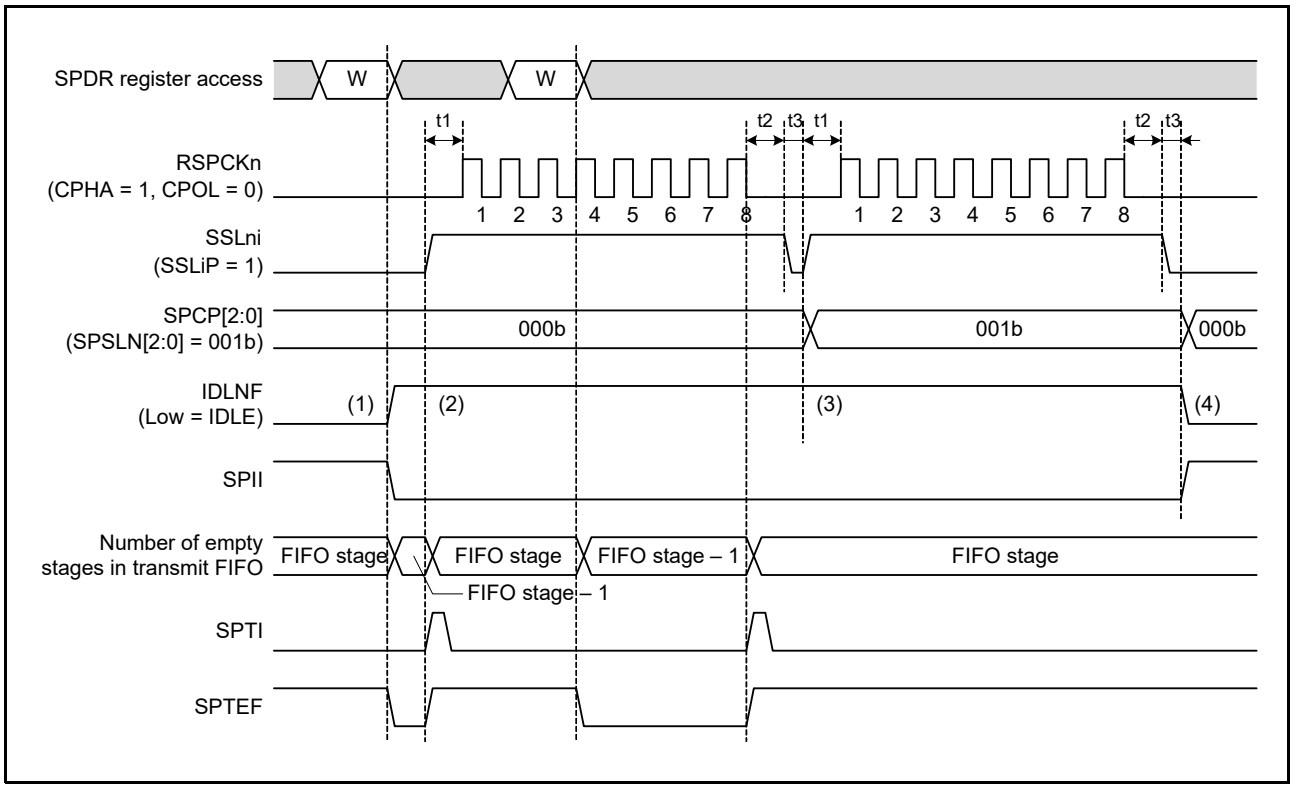


Figure 47.34 Operation Example of Idle Interrupts (Master Mode, Motorola SPI)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

[TI SSP]

Figure 47.35 shows an example of idle interrupt operation during normal operation.

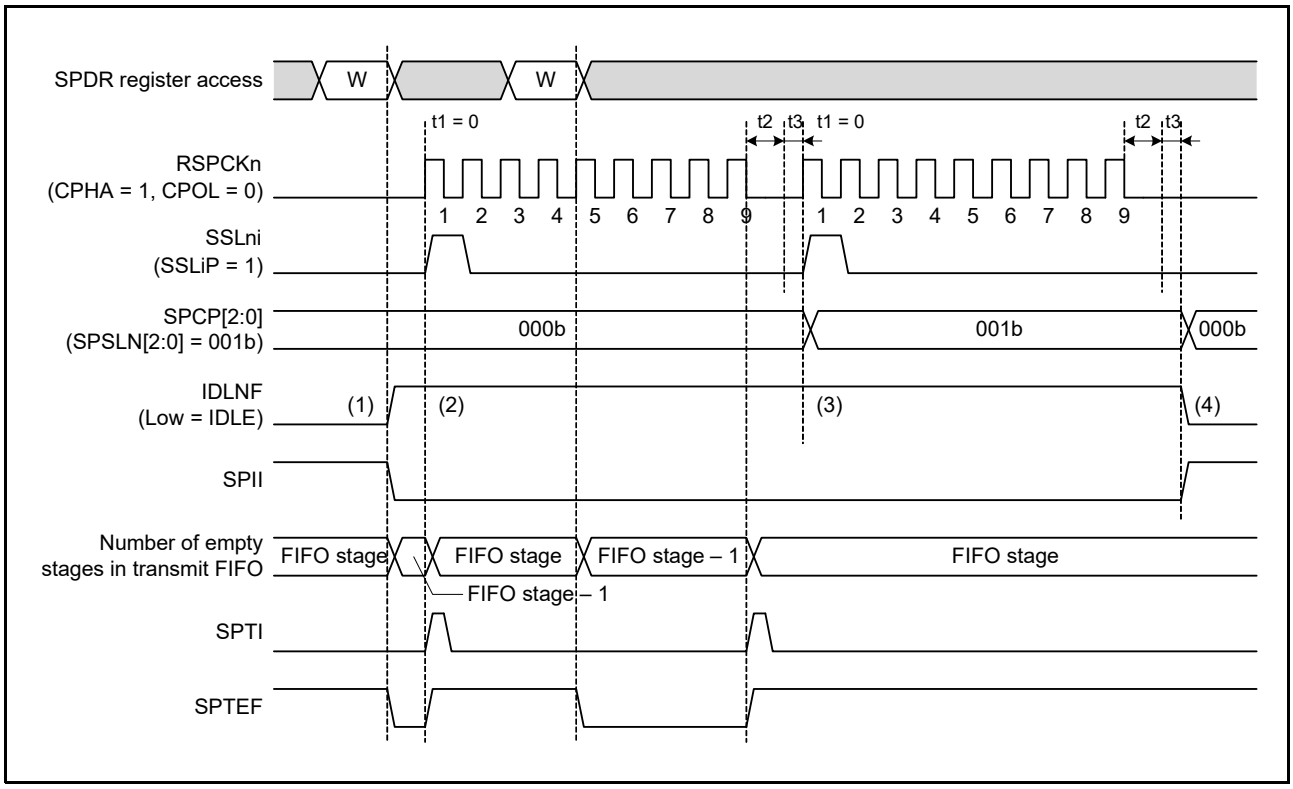


Figure 47.35 Operation Example of Idle Interrupts (Master Mode, TI SSP)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

An idle interrupt during slave mode operation depends only on the value of the SPCR.SPE bit.

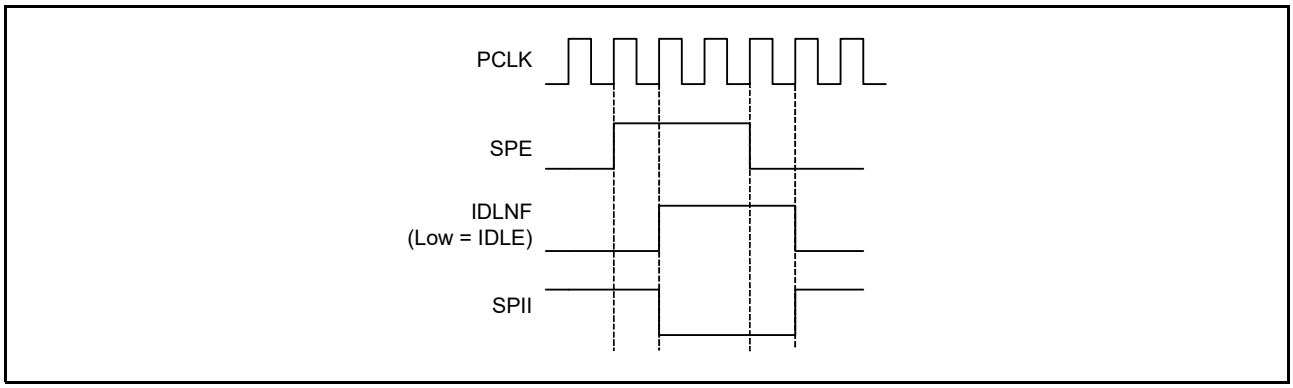


Figure 47.36 Operation Example of Idle Interrupts (Slave Mode)

47.3.9 Communication End Interrupts

47.3.9.1 Transmit-Receive Master Mode or Transmit-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive master mode or transmit-only master mode.

[Motorola SPI]

Figure 47.37 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

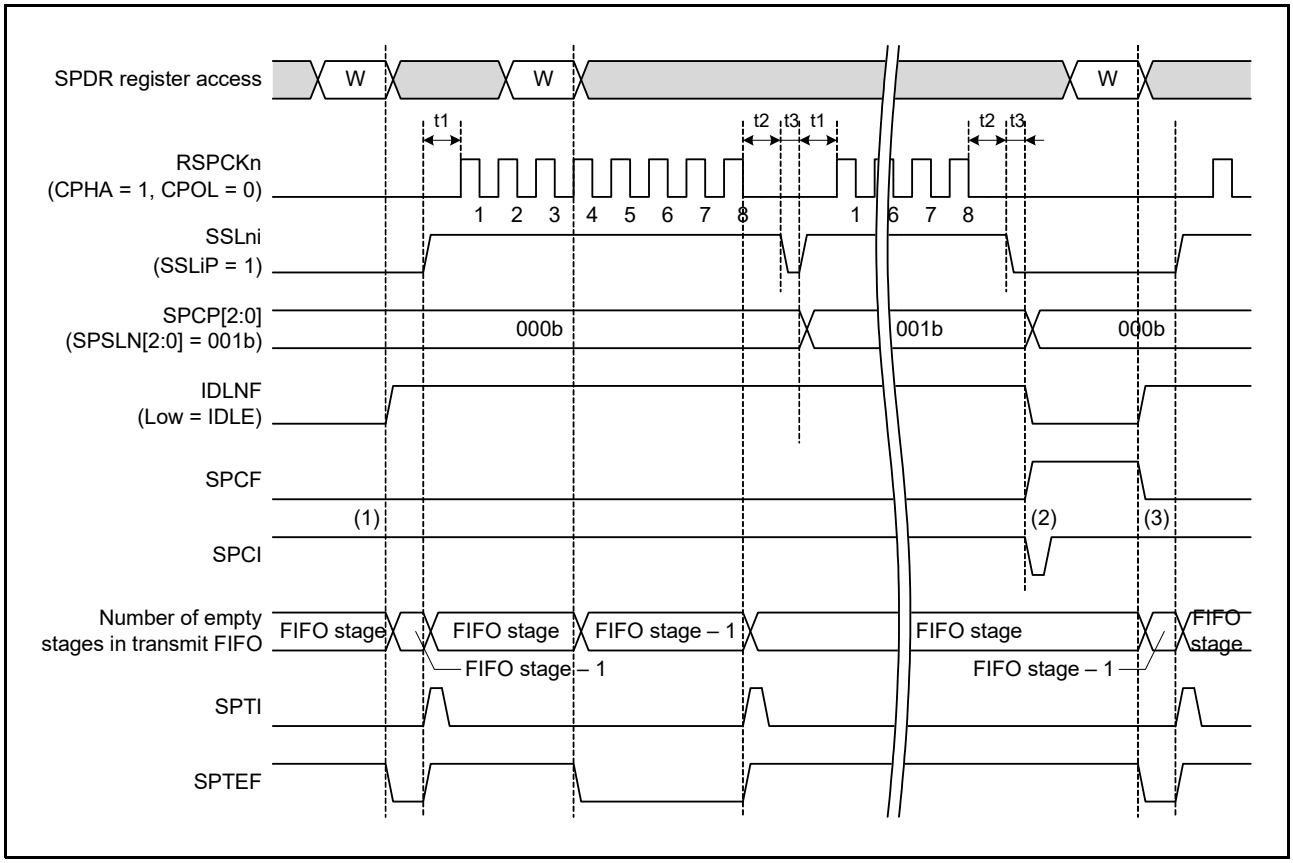


Figure 47.37 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 47.38 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

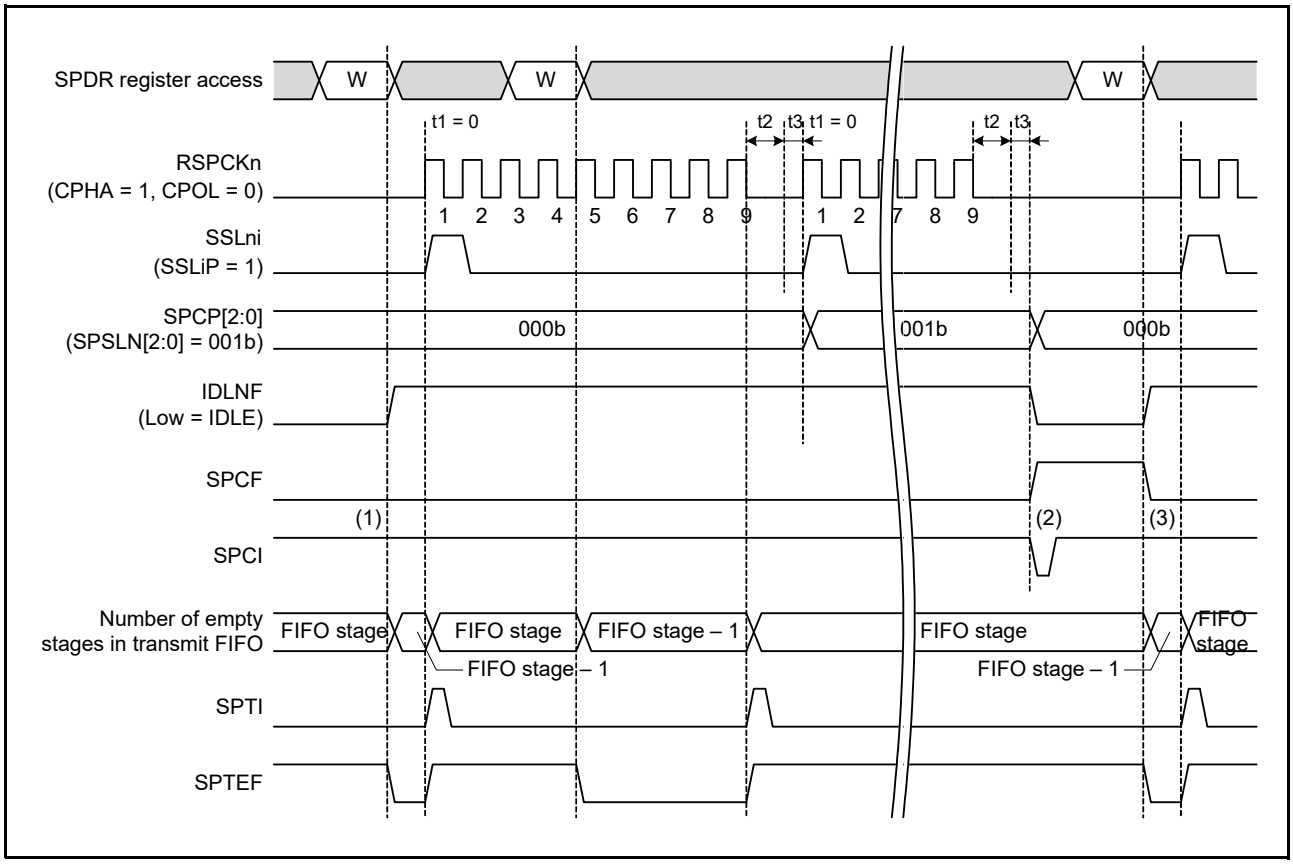


Figure 47.38 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

47.3.9.2 Receive-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only master mode.

Figure 47.39 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are 00000b.

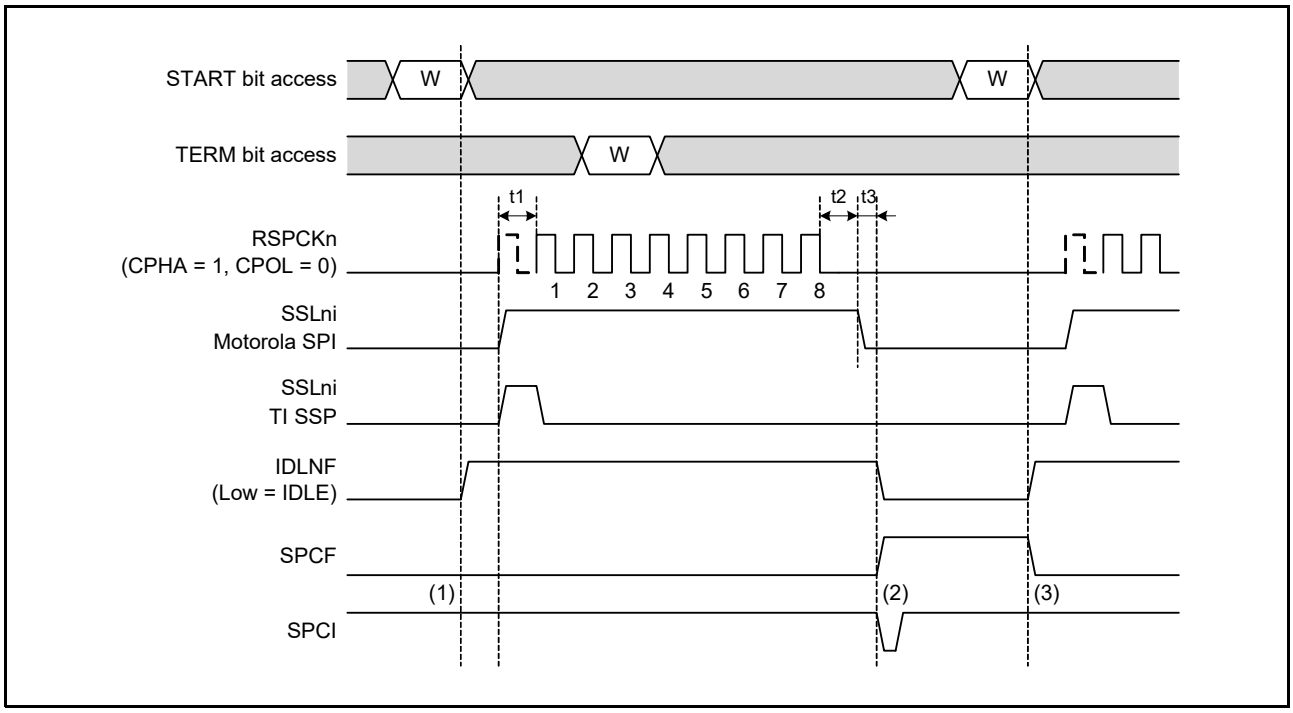


Figure 47.39 Operation Example of SPCI Interrupts (RFC[4:0] = 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) When the SPRMCR.TERM bit is set to 1 during communication, the SPCF flag becomes 1 (communication end) at the end of t3 cycle. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

Figure 47.40 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are not 00000b.

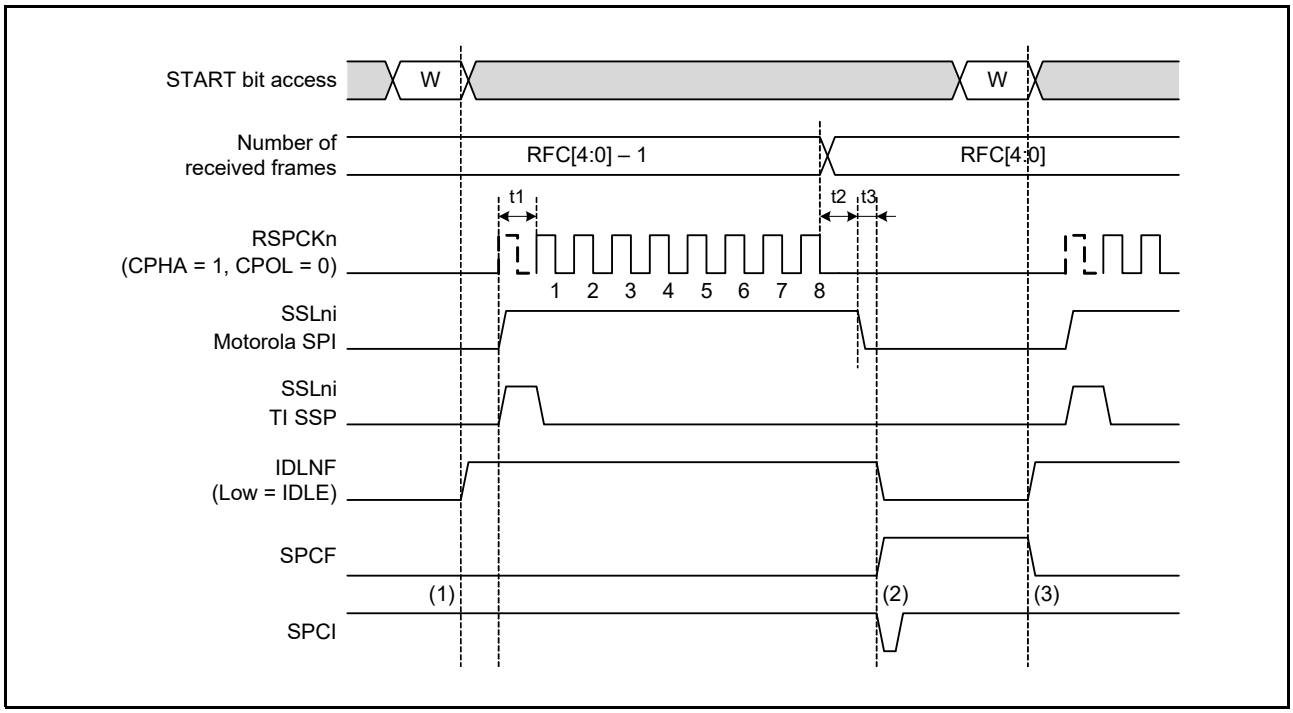


Figure 47.40 Operation Example of SPCI Interrupts (RFC[4:0] ≠ 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle after receiving the number of frames set by the SPRMCR.RFC[4:0] bits. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

In slave mode operation, the output timing of the SPCI interrupt is different due to the value of the SPCR.SPMS bit (RSPI mode select bit). And the clear timing of the communication end interrupt is different due to the communication mode (transmit-receive, transmit-only, or receive-only).

47.3.9.3 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on SPI Operation

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 47.41 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

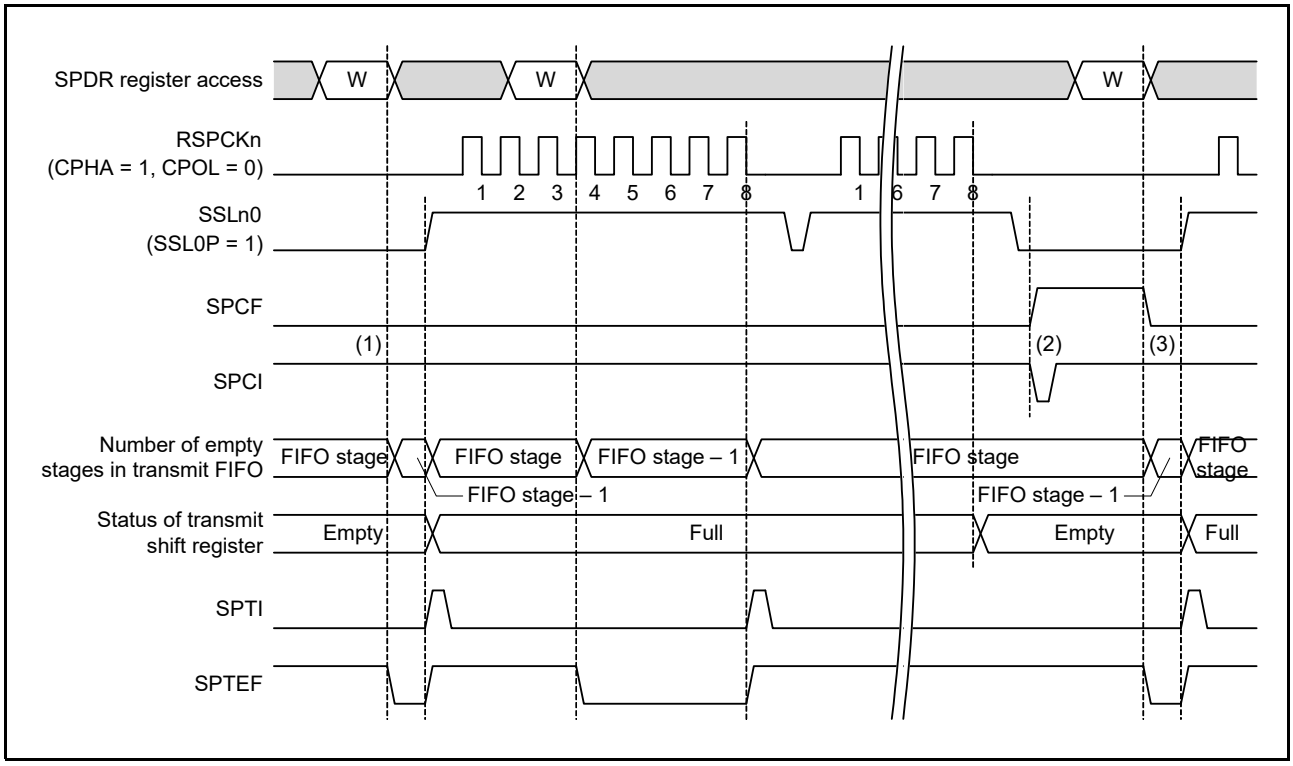


Figure 47.41 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the timing of SSLni negation, when the next transmit data is not set in the transmit FIFO and the transmit shift register is empty. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 47.42 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

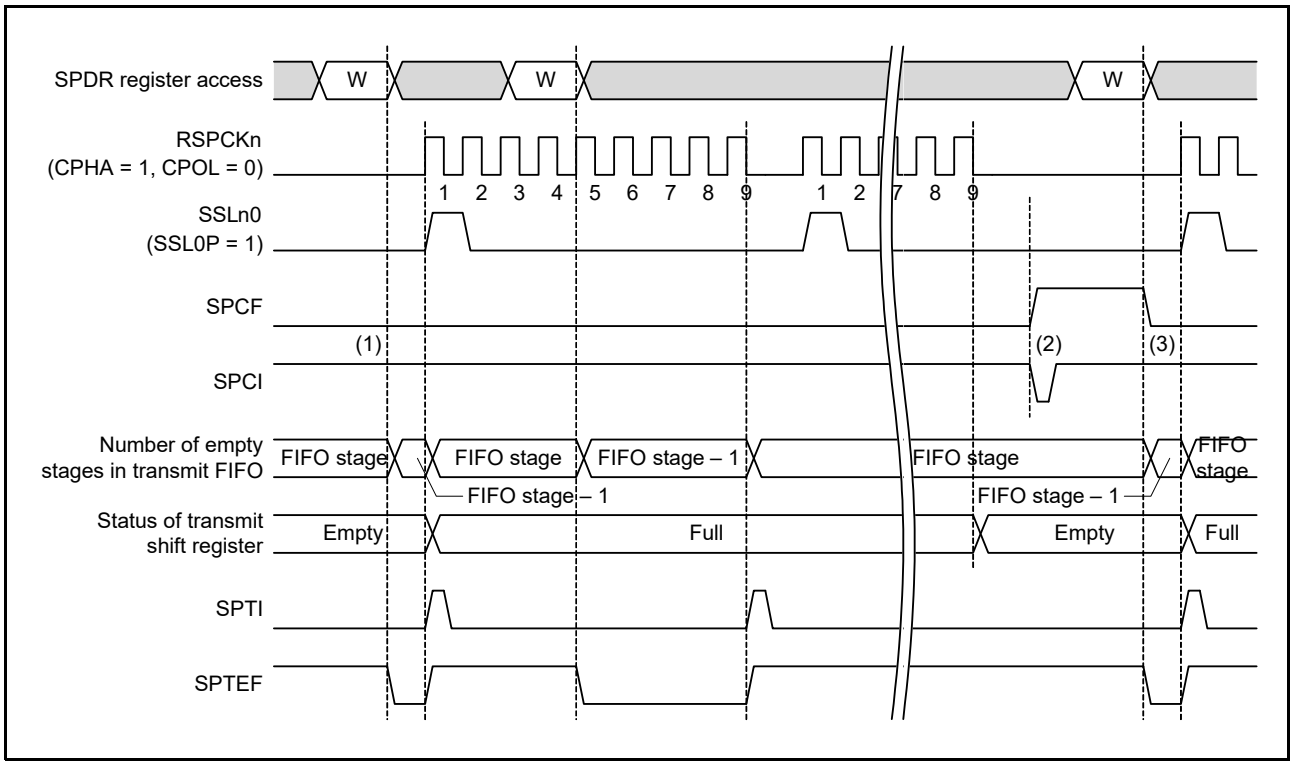


Figure 47.42 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) when the next transmit data is not set in the transmit FIFO, the transmit shift register is empty, and SSL negation delay is completed. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

47.3.9.4 Receive-Only Slave Mode on SPI Operation

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 47.43 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

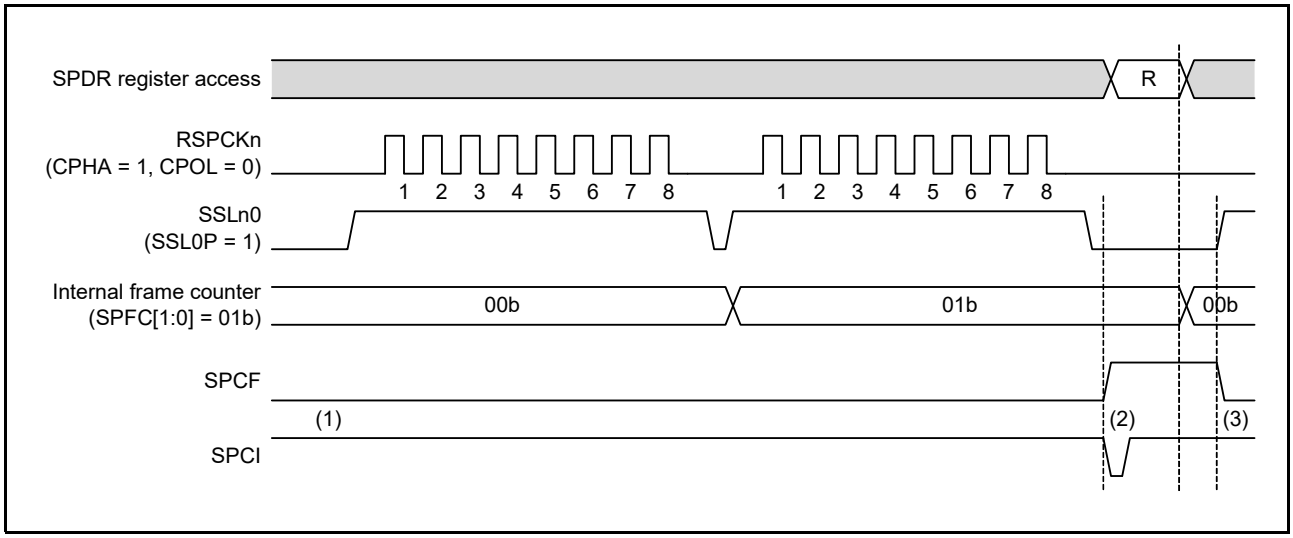


Figure 47.43 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing of SSLn0 negation. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSLn0 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 47.44 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

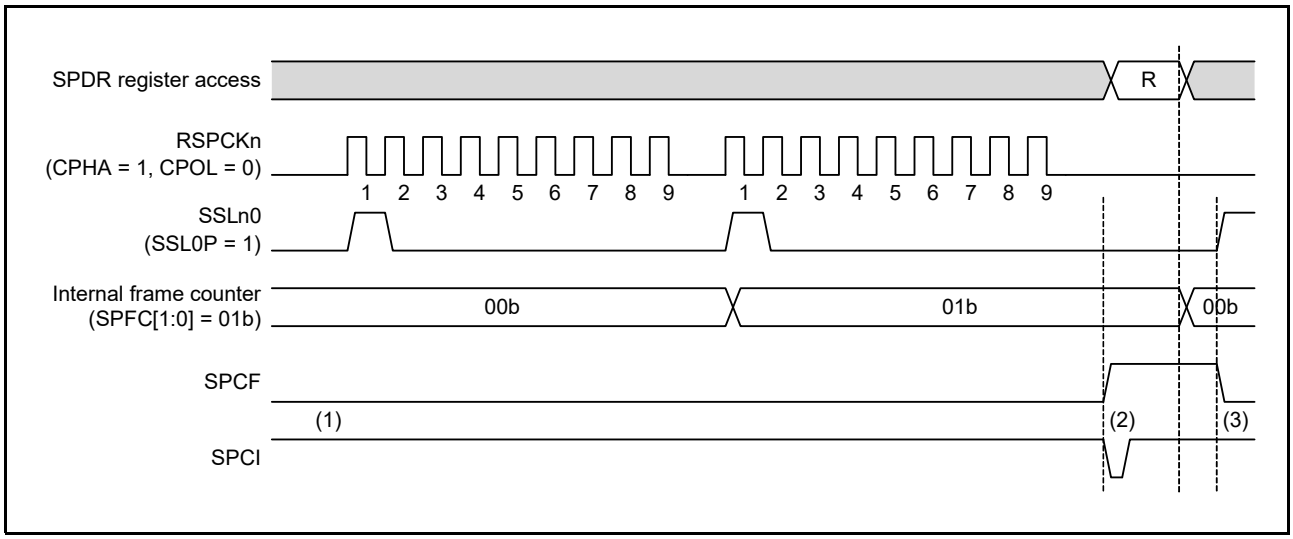


Figure 47.44 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing when SSLn0 negation delay is completed. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSLn0 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

47.3.9.5 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on Clock Synchronous Operation

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on clock synchronous operation (3-wire method).

Figure 47.45 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on clock synchronous operation.

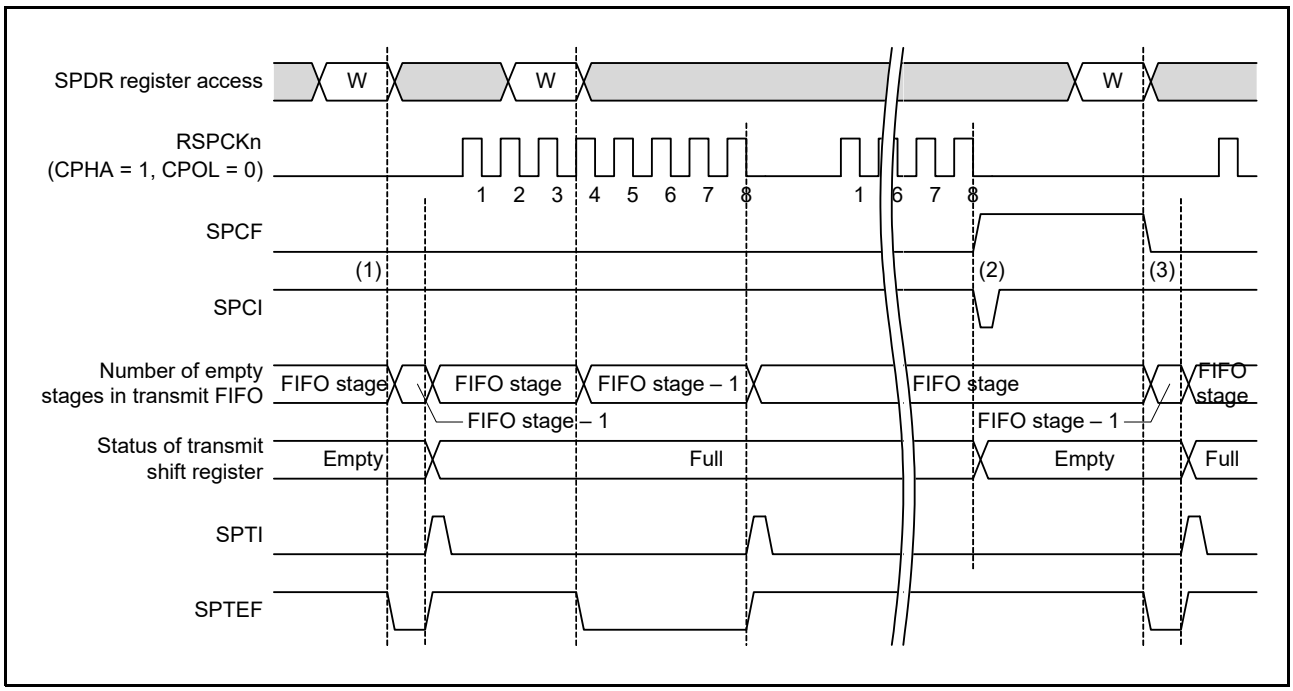


Figure 47.45 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the final data bit of RSPCKn, when the next transmit data is not set in the transmit buffer and the transmit shift register is empty. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

47.3.9.6 Receive-Only Slave Mode on Clock Synchronous Operation

Refer to the description of the SPCF Flag (Communication End Flag) in section 47.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on clock synchronous operation (3-wire method).

Figure 47.46 shows an example of the SPCI interrupt operation during receive-only slave mode on clock synchronous operation.

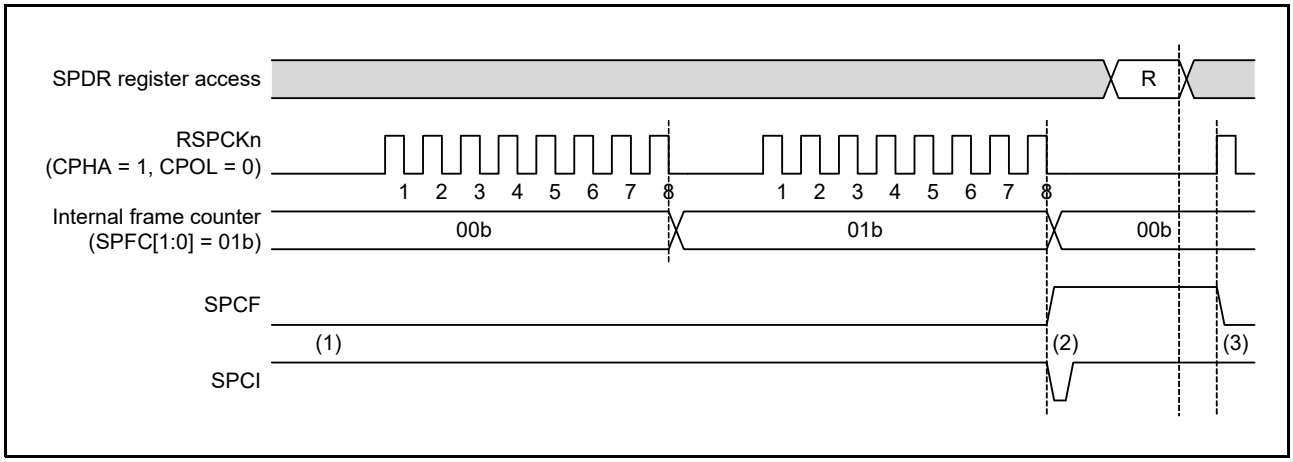


Figure 47.46 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the last data bit of RSPCKn in the last frame when the number of frames set in the SPDCR.SPFC[1:0] bits is received. And then the SPCI interrupt outputs with 1 PCLKA cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the first edge of RSPCKn for the next transmission. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

47.3.10 Error Detection

In the normal serial transfer of the RSPIA, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPIA at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPIA detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 47.9 lists the relationship between non-normal transfer operations and the RSPIA's error detection function.

Table 47.9 Relationship between Non-Normal Transfer Operations and Error Detection Function of RSPIA

	Occurrence Condition	RSPI Operation	Error Detection
1	The SPDR register is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	The SPDR register is read while no empty stages in the receive FIFO.	If reception has completed, then the received data is output to the bus. Otherwise, data received previously is output instead.	None
3	Serial transfer is started in transmit-receive slave mode or transmit-only slave mode when transmit data is not ready to output.	• Serial transfer is suspended • Transmit/receive data is missing • The MISOn output signal is disabled • RSPI function is disabled	Underrun error Refer to section 47.3.10.4, Underrun Error for underrun error.
4	Serial transfer terminates when data is stored in the receive FIFO with no empty FIFO stage.	• The contents of the receive FIFO are kept • Missing receive data	Overrun error Refer to section 47.3.10.1, Overrun Error for overrun error.
5	An incorrect parity bit is received with the parity function enabled in the transmit-receive mode or receive-only mode	The parity error flag is asserted.	Parity error Refer to section 47.3.10.2, Parity Error for parity error.
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	Mode fault error Refer to section 47.3.10.3, Mode Fault Error for mode fault error.
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped. • RSPI function is disabled.	
8	The SSLn0 input signal is negated during serial transfer in slave mode. [Motorola SPI]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOn output signal is stopped. • RSPI function is disabled.	
9	The SSLn0 input signal is asserted during serial transfer in slave mode. [TI SSP]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOn output signal is stopped. • RSPI function is disabled.	
10	After data is stored in the receive FIFO with RDRIS = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of the SPDRCSR register.	• The receive data ready flag is asserted.	Receive data ready Refer to section 47.3.11, Received Data Ready Detection for receive data ready

On operation 1 in Table 47.9, the RSPIA does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPIA does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when a receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

For the transmit and receive interrupts, refer to section 47.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

47.3.10.1 Overrun Error

If a serial transfer ends while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPIA does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the SPSR.OVRF flag to 0, issue a system reset or write 1 to the SPSCLR.OVRFC bit.

Figure 47.47 shows an example of operations of the SPSR.SPRF and OVRF flags. The SPSCLR register access and SPDR register access shown in Figure 47.47 indicate the condition of accesses to SPSCLR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 47.47, the RSPIA performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

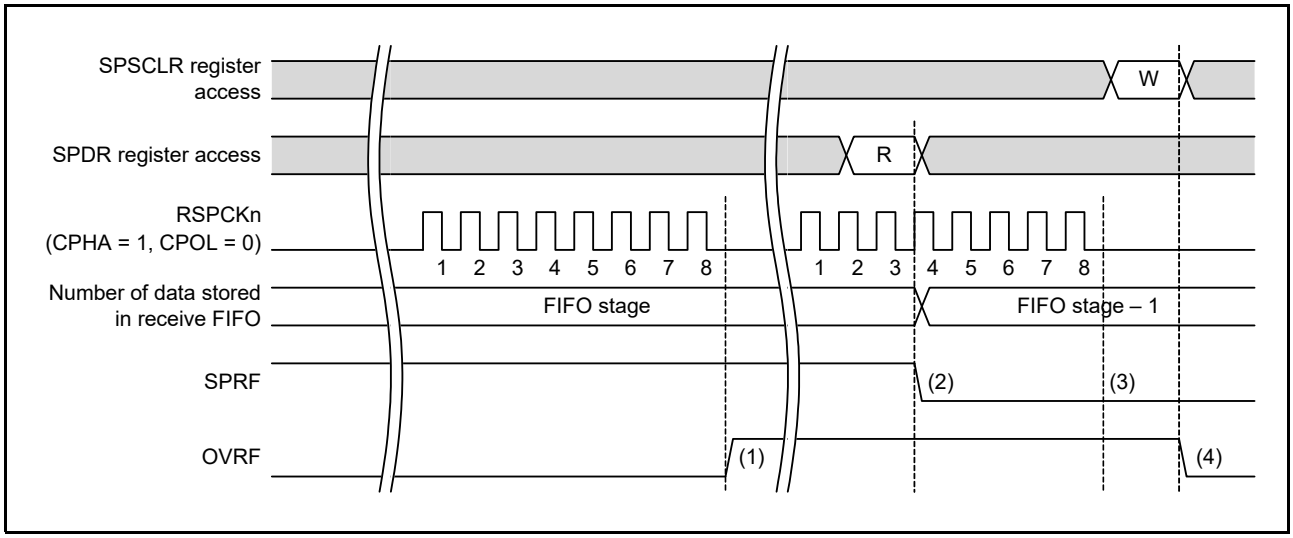


Figure 47.47 Operation Example of SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the OVRF flag to 1. The RSPIA does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPIA outputs the data in the receive buffer. At this time the SPRF flag becomes 0 at the last access when the received data is read from the SPDR register in one processing routine using DTC/DMAC. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPIA does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPIA does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPIA does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPIA determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) When 1 is written to the SPSCLR.OVRFC bit, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading the SPSR register immediately after SPDR is read. When the RSPIA is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits. If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 47.48 and Figure 47.49 show the clock stop waveform when a serial transfer continues while no empty stages in the receive FIFO in master mode.

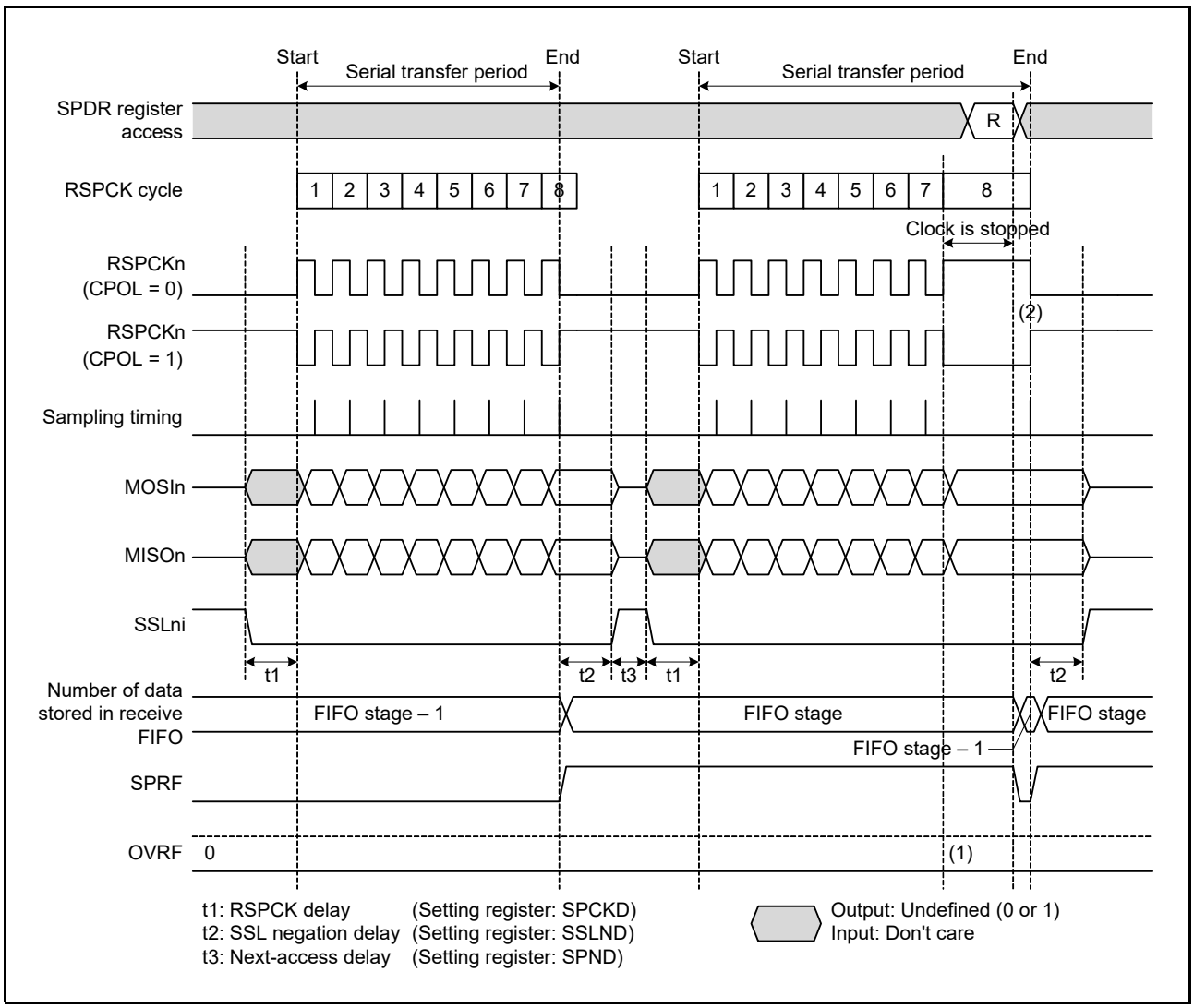


Figure 47.48 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 1)

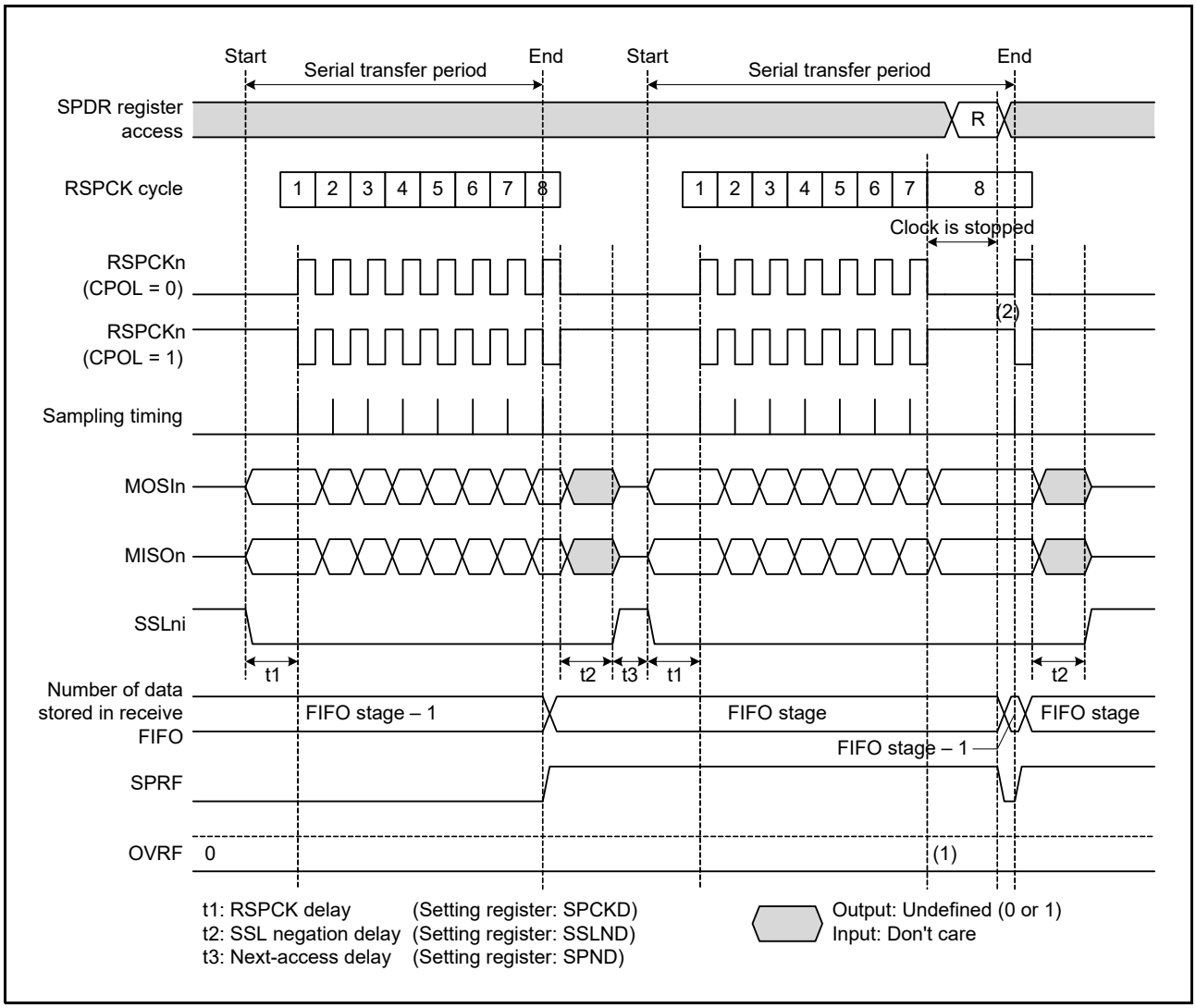


Figure 47.49 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCKn is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCKn restarts after reading the receive buffer.

When the RSPCK auto-stop function is enabled for transfer with no delay of between frames during burst transfer in master mode, an overrun error does not occur. Figure 47.50 and Figure 47.51 show the clock stop waveform when there is no delay between frames during burst transfer and a serial transfer continues while no empty stages in the receive FIFO in master mode.

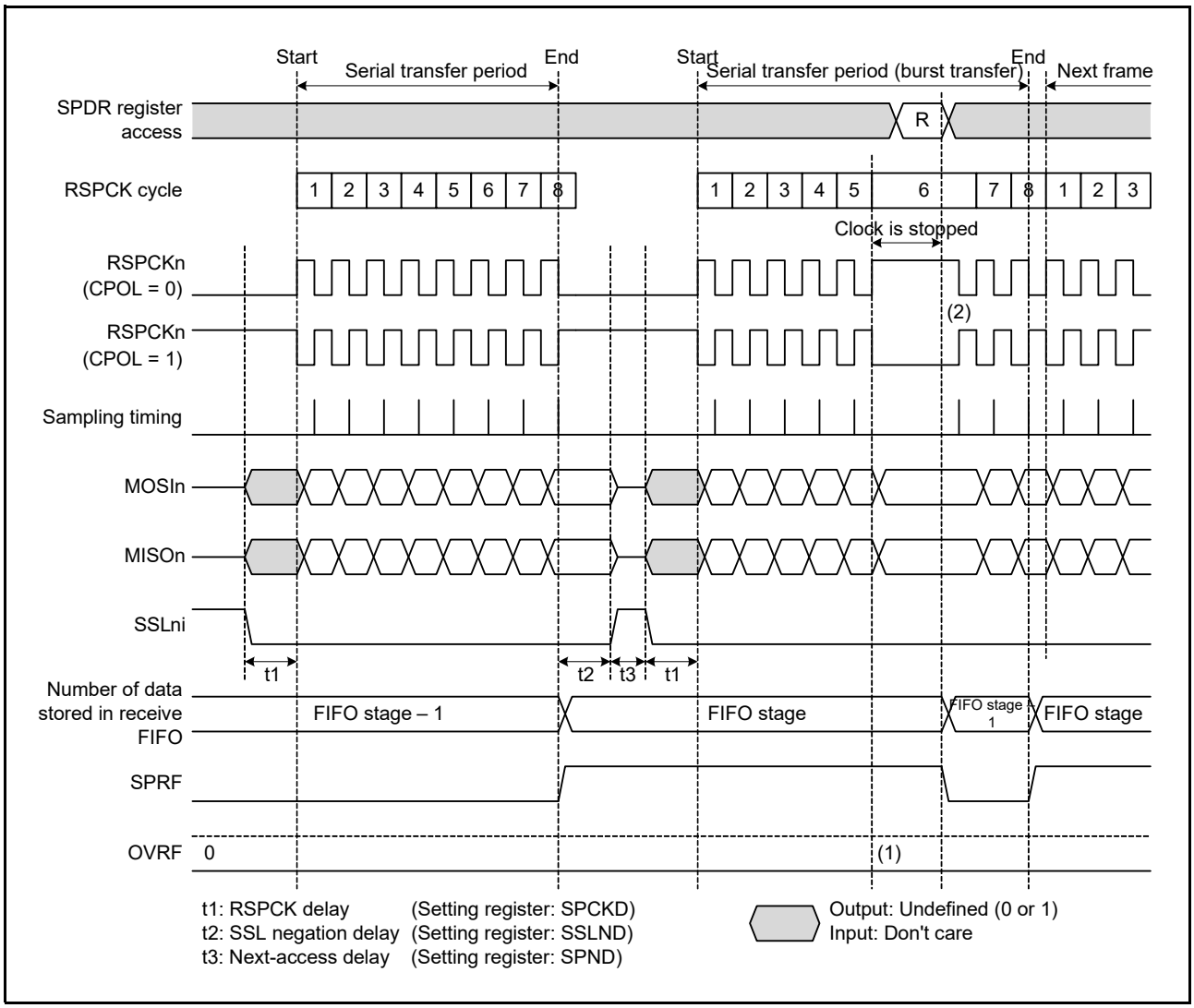


Figure 47.50 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 1)

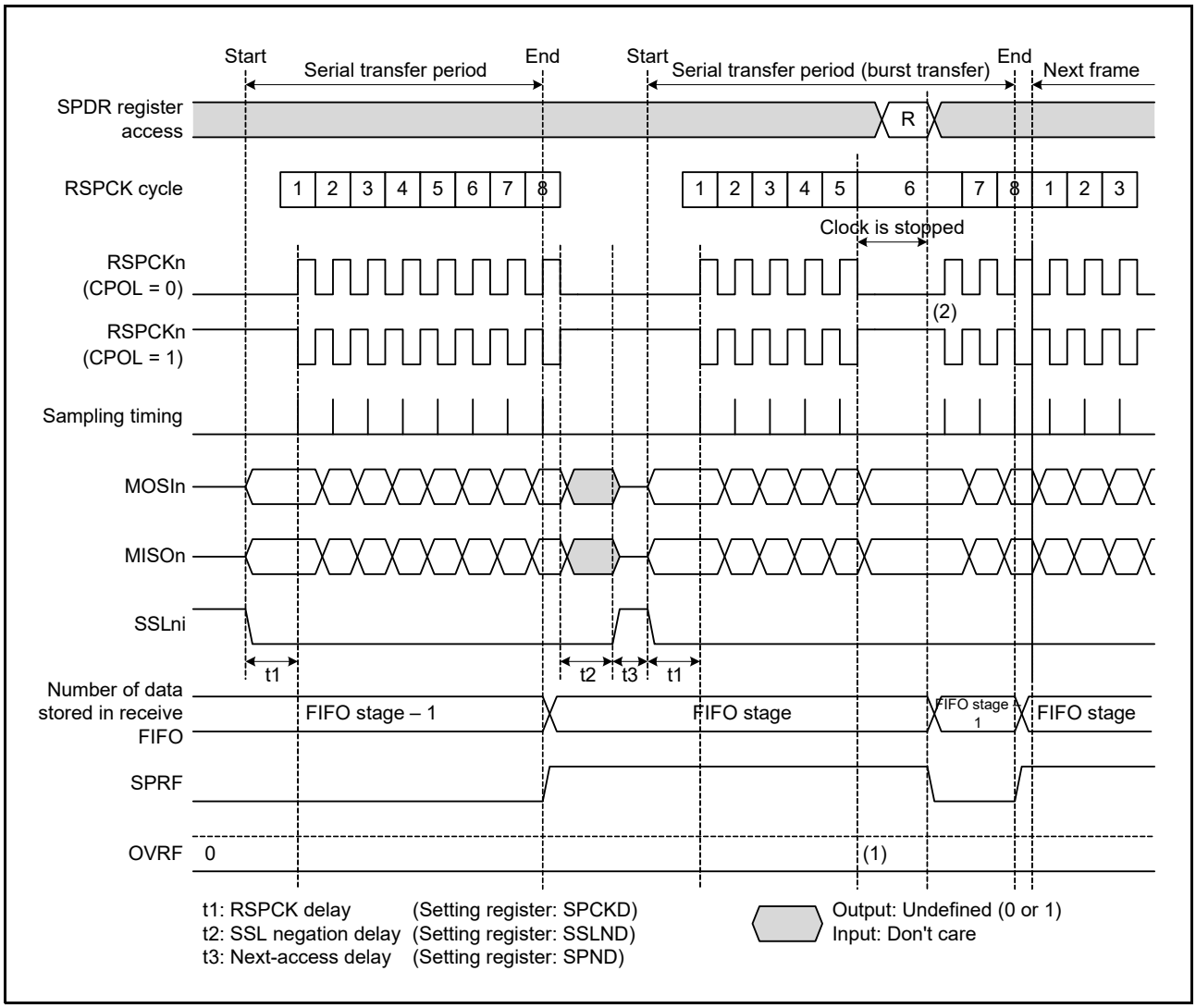


Figure 47.51 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCKn is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCKn restarts after reading the receive buffer.

47.3.10.2 Parity Error

If the SPCR.SPPE bit set to 1, when serial transfer in transmit-receive mode or receive-only mode terminates, the RSPIA checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPIA sets the SPSR.PERF flag to 1. Since the RSPIA does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, issue a system reset or write 1 to the SPSCLR.PERFC bit.

Figure 47.52 shows an example of operation of the OVRF and PERF flags. The SPSCLR register access shown in Figure 47.52 indicates the condition of access to SPSCLR register, where W denotes a write cycle. In the example of Figure 47.52, serial communications is performed while the SPCR.SPPE bit is 1. The RSPIA performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

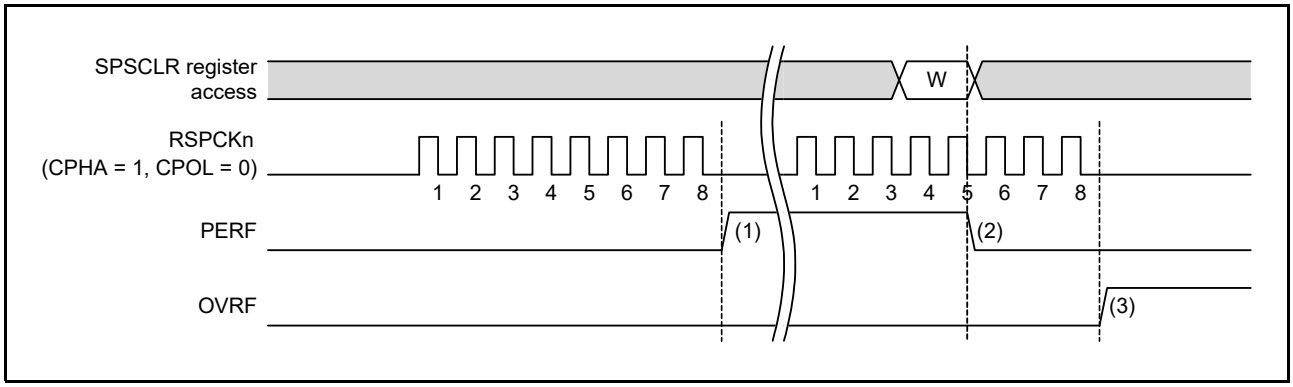


Figure 47.52 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPIA not detecting an overrun error, the RSPIA copies the data in the shift register to the receive buffer. The RSPIA judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When 1 is written to the SPSCLR.PERFC bit, the PERF flag is set to 0.

(3) When the RSPIA detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPIA does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. To use an error interrupt, set the SPCR.SPEIE bit to 1. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of parity errors, such as reading the SPSR register. When the RSPIA is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

47.3.10.3 Mode Fault Error

The RSPIA operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLn0 input signal of the RSPIA in multi-master mode, the RSPIA detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPIA copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLn0 signal is determined by the SSLP.SSL0P bit.
When the MSTR bit is 0, the RSPIA operates in slave mode. The RSPIA detects a mode fault error if the MODFEN bit of the RSPIA in slave mode is 1, and the SPMS bit is 0, and the following conditions are satisfied during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

- In the Motorola SPI protocol, when the SSLn0 input signal is negated during the serial transfer period
- In the TI SSP protocol, when the SSLn0 input signal is asserted during the serial transfer period
  However, during a burst transfer, no error is detected even if the SSLn0 input signal is asserted during the last bit of frame.

Upon detecting a mode fault error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. When the SPE bit is cleared, the RSPI function is disabled (refer to section 47.3.12, Initializing RSPIA). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.
The occurrence of a mode fault error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting mode fault errors without utilizing the error interrupt requires polling of the SPSR register. When using the RSPIA in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
When the MODF flag is set to 1, writing 1 to the SPE bit is ignored. To enable the RSPI function after a mode fault error has been detected, clear the MODF flag and then set the SPE bit to 1.

47.3.10.4 Underrun Error

If RSPIA operates in slave mode (the SPCR.MSTR bit is 0) and the communication mode select bits (CMMD[1:0]) in the RSPI control register (SPCR) is set to 00b or 01b, when serial transfer is started before transmit data output is ready with the SPCR.SPE bit is 1 (RSPI function is enabled), the RSPIA detects an underrun error, and sets the SPSR.MODF and SPSR.UDRF flags to 1.
Upon detecting an underrun error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. Clearing of the SPE bit disables the RSPI function. (Refer to section 47.3.12, Initializing RSPIA).
The occurrence of an underrun error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting underrun errors without utilizing the error interrupt requires polling of the SPSR register.
When the MODF flag is set to 1, writing 1 to the SPE bit is ignored. To enable the RSPI function after an underrun error has been detected, clear both the MODF and UDRF flags simultaneously, and then set the SPE bit to 1.

47.3.11 Received Data Ready Detection

When the SPCR.CMMD[1:0] bits are 00b, 01b or 11b, and the value of the SPDRCSR register is not 00h, after receiving data in the receive FIFO during communication (SPE = 1), the SPSR.RRDYF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than the threshold value and the value set in the SPDRCSR register has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPRI or SPEI with the SPCR.RDRIS bit.

Figure 47.53 shows an example of received data ready detection operation.

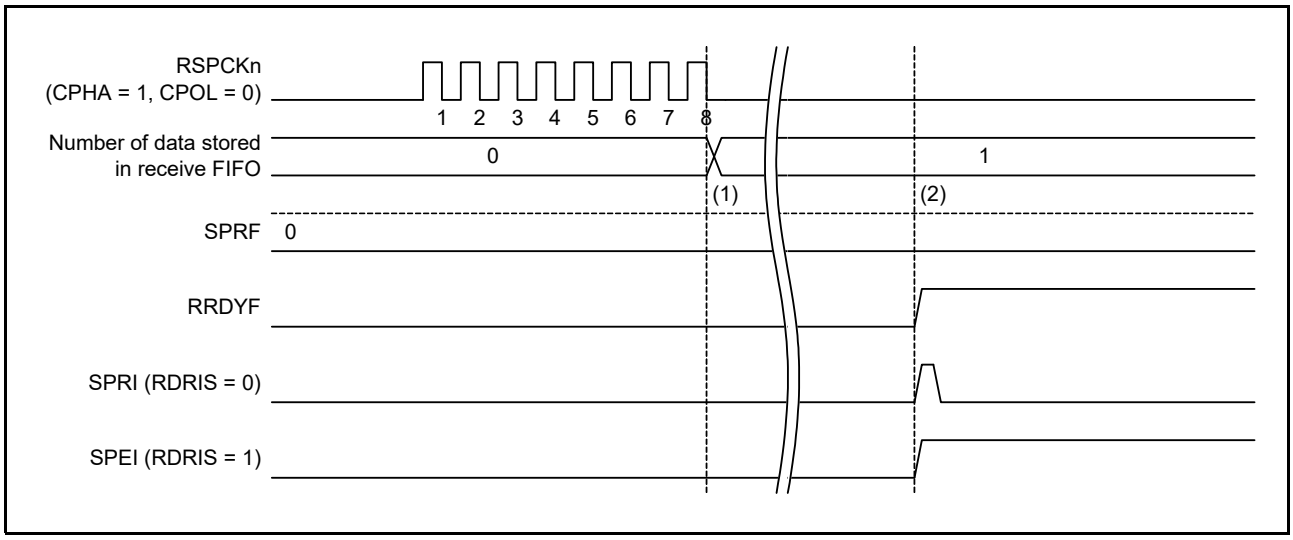


Figure 47.53 Received Data Ready Detection

The operation at the timing shown in steps (1) and (2) in the figure is described below.

(1) Store the received data to the receive FIFO. The SPSR.SPRF flag is 0, because the number of data stored in the receive FIFO is equal to or less than the value of the SPFCR.RTRG[1:0] bits.

(2) Set the RRDYF flag and assert SPRI or SPEI because data is not written to the receive FIFO for the value of the SPDRCSR register from (1).

47.3.12 Initializing RSPIA

If 0 is written to the SPCR.SPE bit or the RSPIA sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the RSPIA disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPIA initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

47.3.12.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPIA performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPIA
- Set the SPSR.SPTEF flag to 1

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPIA. For this reason, the RSPIA can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPCF, SPRF, OVRF, MODF, PERF, and UDRF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPIA is initialized, data from the receive buffer, the communication end status, and the error status during RSPI transfer can be read to check them.
The SPSR.SPTEF flag is initialized to 1. Therefore, if the SPCR.SPTIE bit is set to 1 after RSPIA initialization, a transmit buffer empty interrupt is generated. When the CPU initializes the RSPIA, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

47.3.12.2 System Reset

The initialization by a system reset completely initializes the RSPIA through the initialization of all bits for controlling the RSPIA, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 47.3.12.1, Initialization by Clearing the SPE Bit.

47.3.13 SPI Operation

47.3.13.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 47.3.10, Error Detection). When operating in single-master mode, the RSPIA does not detect mode fault errors whereas the RSPIA running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) when data is written to the RSPI data register (SPDR) with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty the RSPIA copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPIA changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format). The polarity of the SSLni output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

[Except receive-only operation in master mode]
Irrespective of the SPCMDm.CPHA bit, the RSPIA terminates the serial transfer after transmitting an RSPCKn edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, after the termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSLni output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).

[Receive-only operation in master mode]
The RSPIA terminates the serial transfer when any of the following conditions is satisfied.

- After the RSPIA detects the RSPCKn edge corresponding to the final sampling timing irrespective of the SPCMDm.CPHA bit
- When writing 1 to the SPRMCR.TERM bit during the serial transfer period

If the number of data stored in the receive FIFO is less the number of FIFO stages, after termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSLni output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).

(3) Sequence Control

The transfer format that is employed in master mode is determined by the following registers.

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: SSLni pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.

According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPIA loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPIA increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPIA sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

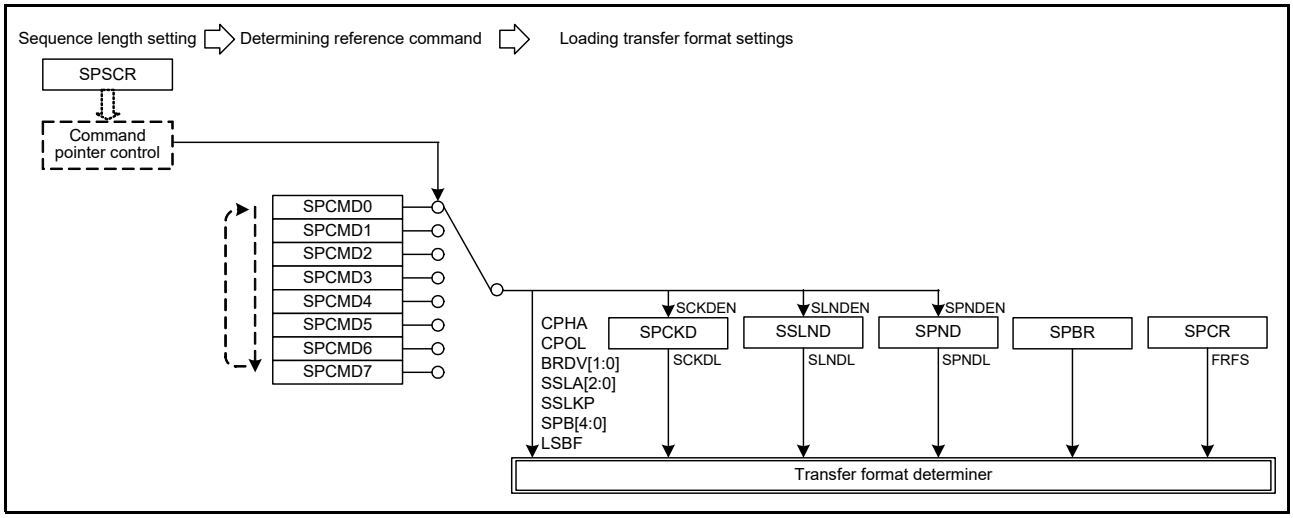


Figure 47.54 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

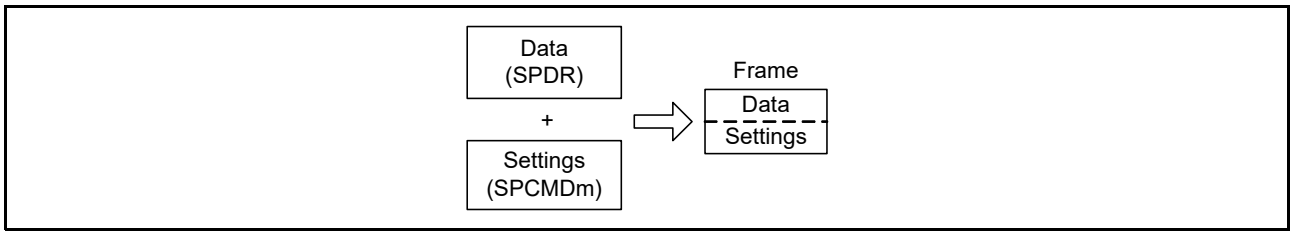


Figure 47.55 Concept of a Frame

Figure 47.56 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in sequence operations.

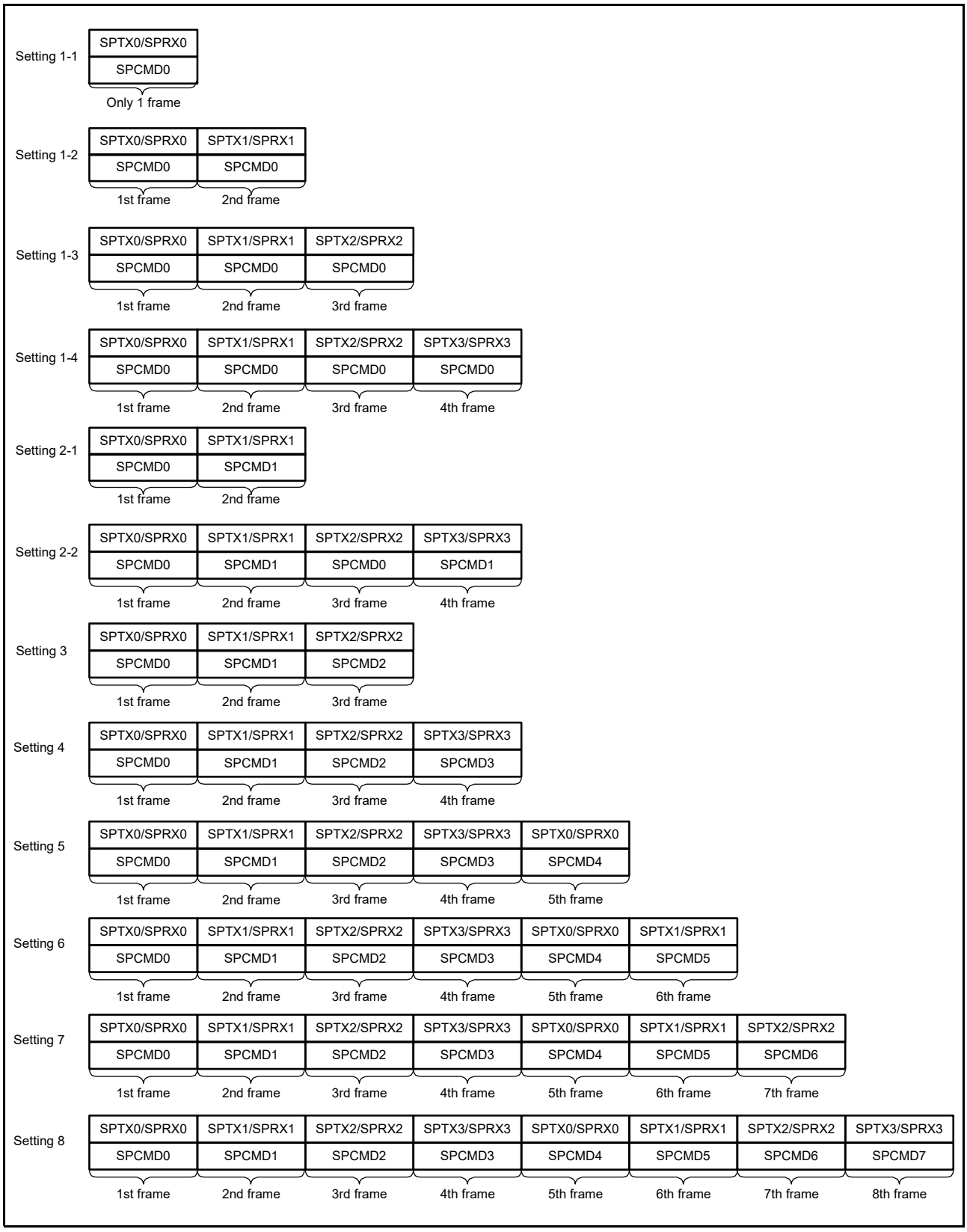


Figure 47.56 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

[Motorola SPI]

If the SPCMDm.SSLKP bit that the RSPIA references during the current serial transfer is 1, the RSPIA keeps the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the RSPIA can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 0

Figure 47.57 shows an example of an SSLni signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (8) as shown in Figure 47.57. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.

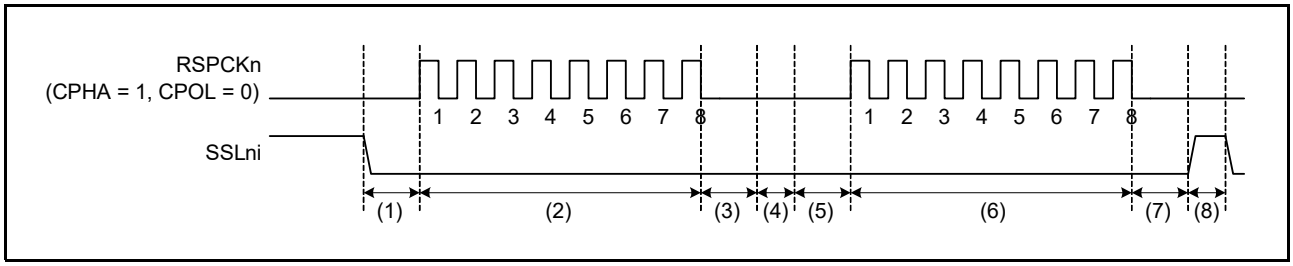


Figure 47.57 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 0, FRFS = 0)

(1) Based on SPCMD0, the RSPIA asserts the SSLni signal and inserts RSPCK delays.
(2) The RSPIA executes serial transfers according to SPCMD0.
(3) The RSPIA inserts SSL negation delays.
(4) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSLni signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0 and additional 5 TCLK cycles. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
In receive-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSLni signal value on SPCMD0. This period is sustained for a period equal to the next-access delay of SPCMD0 and additional 5 TCLK cycles.
(5) Based on SPCMD1, the RSPIA asserts the SSLni signal and inserts RSPCK delays.
(6) The RSPIA executes serial transfers according to SPCMD1.
(7) The RSPIA inserts SSL negation delays.
(8) Since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSLni signal. In addition, a next-access delay is inserted according to SPCMD1.

Note: If the SSLni signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit (to use for burst transfer) are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSLni signal status to SSLni signal assertion ((5) in Figure 47.57) corresponding to the command for the next transfer. Note that if such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels may occur.

The RSPIA in master mode references the SSLni signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPIA can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally. For this reason, burst transfer in master mode is enabled regardless of the CPHA bit setting. (Refer to section 47.3.13, SPI Operation.)

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 47.58 shows an example of an SSLni signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (6) as shown in Figure 47.58. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.

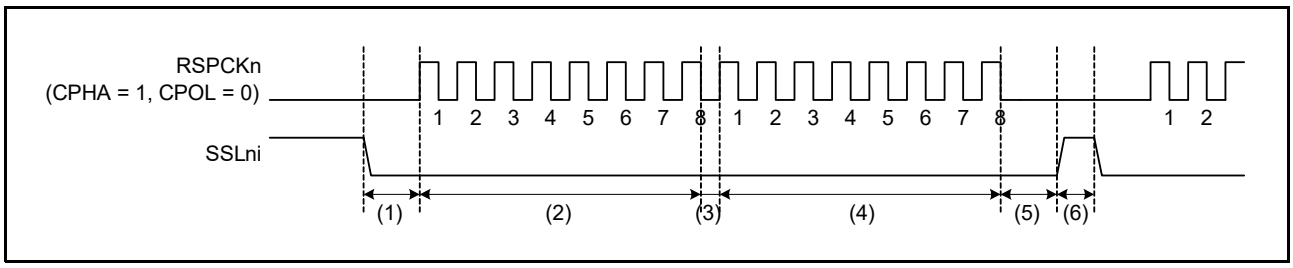


Figure 47.58 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 1, FRFS = 0)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSLni signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.

(2) In transmit-receive mode or transmit-only mode, the RSPIA executes serial transfers according to the SPCMD0 register. The RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
In receive-only mode, the RSPIA executes serial transfers according to the SPCMD0 register.

(3) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSLni signal value on the SPCMD0 register. If the shift register is not empty, RSPCK negate period between frames is 0.5 RSPCK.
In receive-only mode, since it is not the last frame*1, the RSPIA keeps the SSLni signal value on the SPCMD0 register. RSPCK negate period between frames is 0.5 RSPCK.

(4) The RSPIA executes serial transfers according to the SPCMD1 register.

(5) The RSPIA inserts SSL negation delays.

(6) In transmit-receive mode or transmit-only mode, since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSLni signal. In addition, a next-access delay is inserted according to the SPCMD1 register.
In receive-only mode, the RSPIA negates the SSLni signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.

[TI SSP]
The RSPIA asserts the SSLni signal for one cycle at the start of serial transfer.
The RSPIA can execute continuous serial transfers by asserting the SSLni signal for one cycle at the start of serial transfer (burst transfer).

- When the SPCMDm.SSLKP bit (SSL signal level keeping bit) is 1, and the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 47.59 shows an example of an SSLni signal operation and the serial data MISOn/MOSIn for the case where a burst transfer is implemented using the SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (6) as shown in Figure 47.59. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.

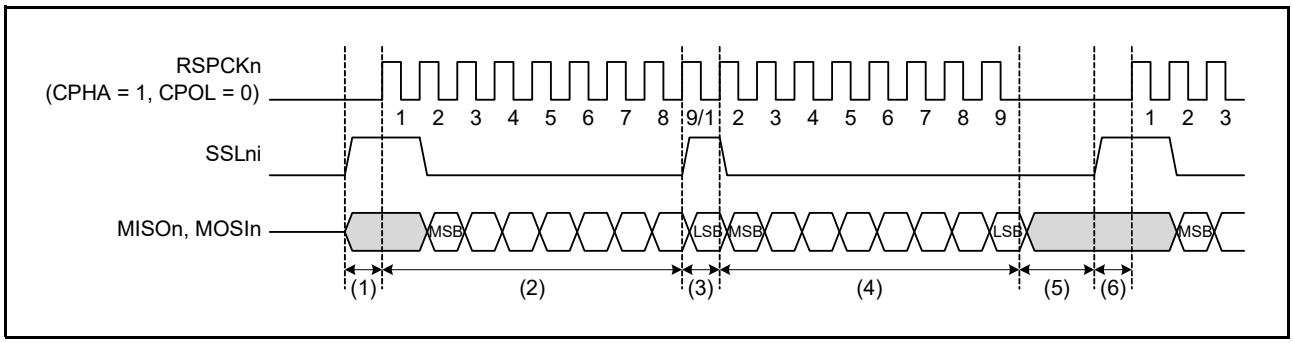


Figure 47.59 Example of Burst Transfer Operation (FRFS = 1)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSLni signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.
(2) The RSPIA executes serial transfers according to the SPCMD0 register.
(3) Final data transfer and the SSLni assertion are performed simultaneously.
During transmit-receive/transmit-only operation, the RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
(4) The RSPIA executes serial transfers according to the SPCMD1 register.
(5) The RSPIA inserts SSLni negation delays at the last frame*1.
(6) A next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.
Note: If the SSLni signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSLni signal status to SSLni signal assertion ((5) in Figure 47.59) corresponding to the command for the next transfer. Note that if such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels may occur.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPIA in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and the SPCKD.SCKDL[2:0] bits, as listed in Table 47.10. For a definition of RSPCK delay, refer to section 47.3.5, Transfer Format (Frame Format).

The RSPCK delay is inserted only the first frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 47.10 Relationship among SCKDEN Bit, SCKDL[2:0] Bits, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value	
		Motorola SPI	TI SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPIA in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND.SLNDL[2:0] bits, as listed in Table 47.11. For a definition of SSL negation delay, refer to section 47.3.5, Transfer Format (Frame Format).

The SSL negation delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 47.11 Relationship among SLNDEN Bit, SLNDL[2:0] Bits, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPIA in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and the SPND.SPNDL[2:0] bits, as listed in Table 47.12. For a definition of next-access delay, refer to section 47.3.5, Transfer Format (Frame Format).

The next-access delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 47.12 Relationship among SPNDEN Bit, SPNDL[2:0], and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 5 TCLK
1	000b	1 RSPCK + 5 TCLK
	001b	2 RSPCK + 5 TCLK
	010b	3 RSPCK + 5 TCLK
	011b	4 RSPCK + 5 TCLK
	100b	5 RSPCK + 5 TCLK
	101b	6 RSPCK + 5 TCLK
	110b	7 RSPCK + 5 TCLK
	111b	8 RSPCK + 5 TCLK

(8) Initialization Flowchart

Figure 47.60 is a flowchart illustrating an example of initialization in SPI operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

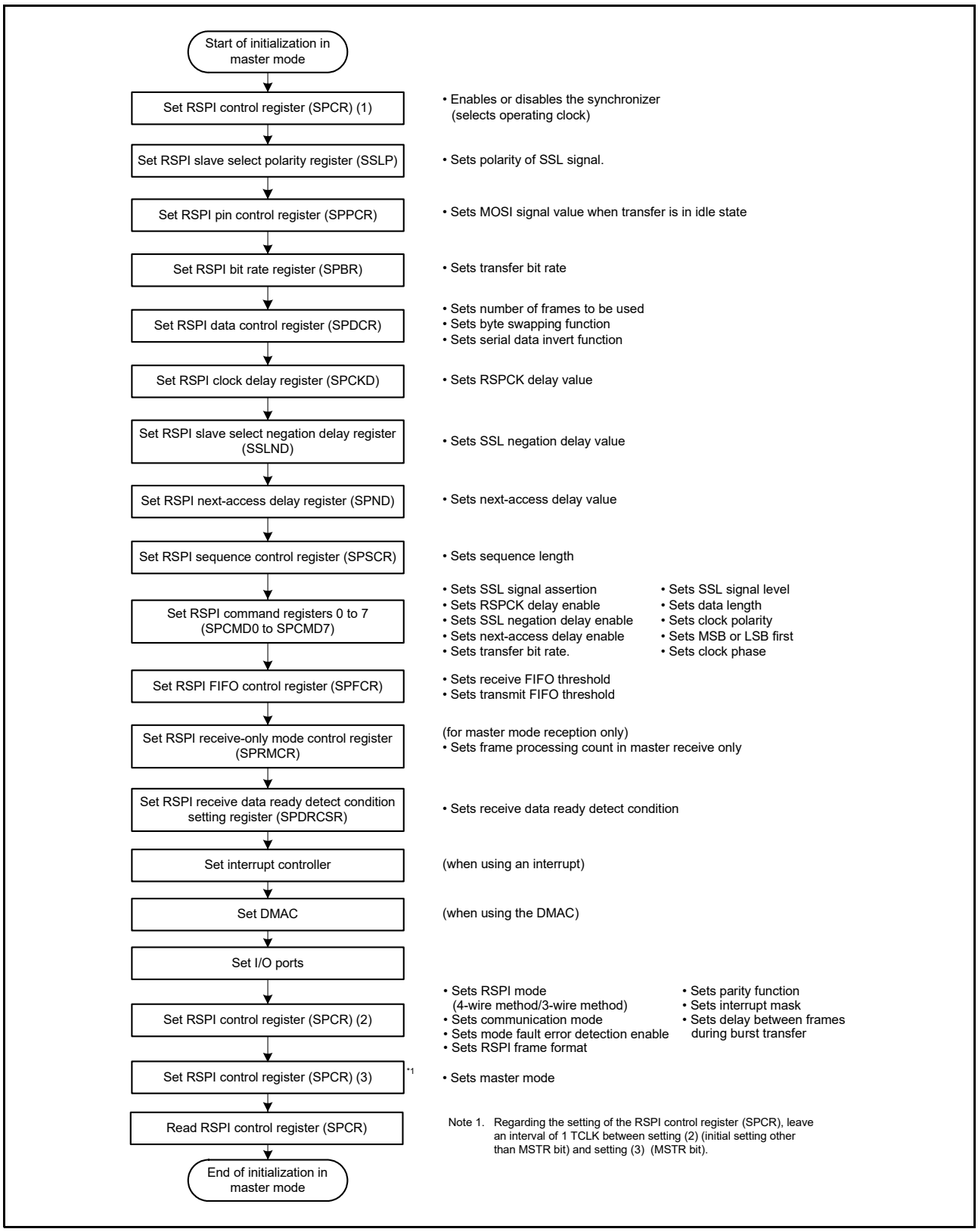


Figure 47.60 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 47.61 to Figure 47.63 show examples of the flow of software processing.

(a) Transmit Processing Flow

Enabling SPII or SPCI interrupt after final transmit data is written makes it possible to notify the CPU that transmission of all data has been completed.

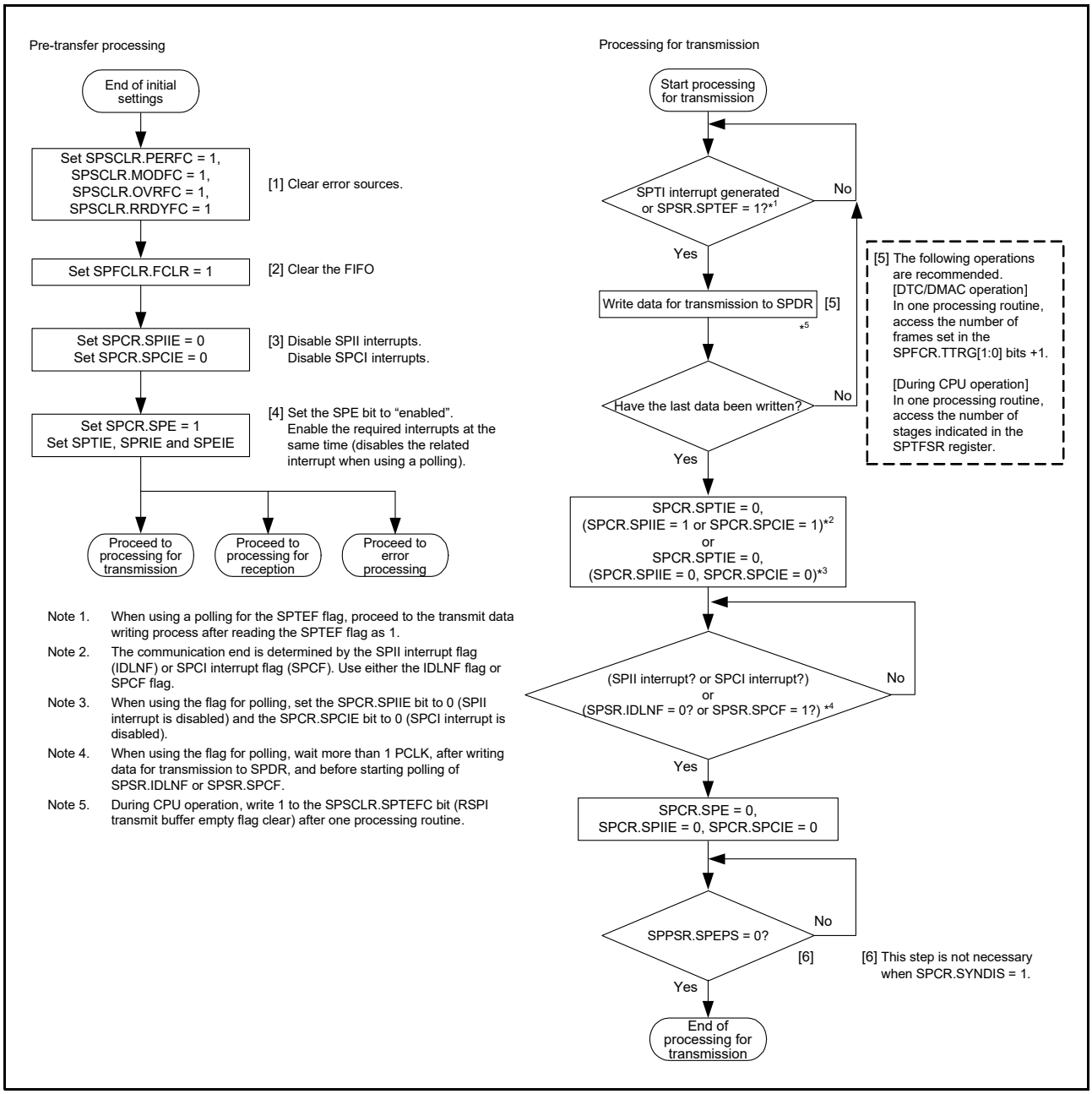


Figure 47.61 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

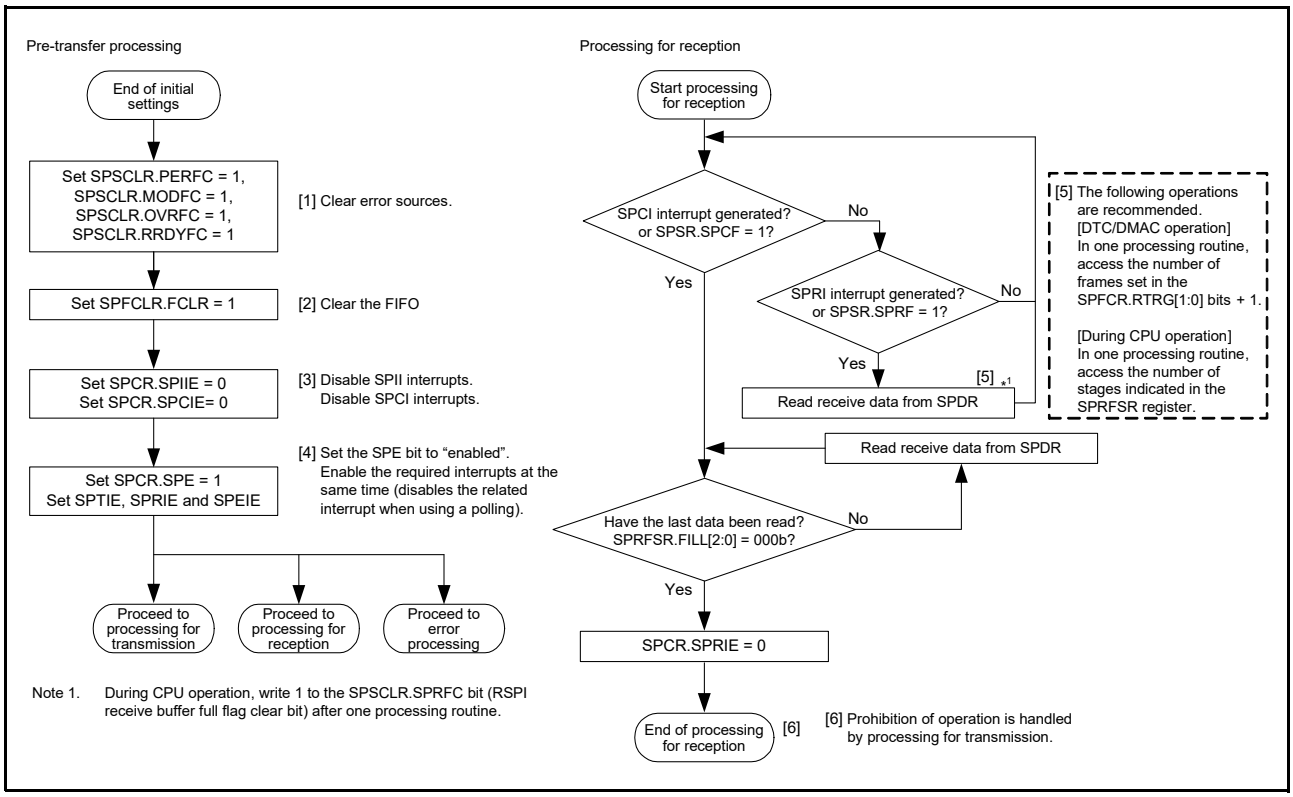


Figure 47.62 Flowchart in Master Mode (Reception)

(c) Receive-Only Processing Flow

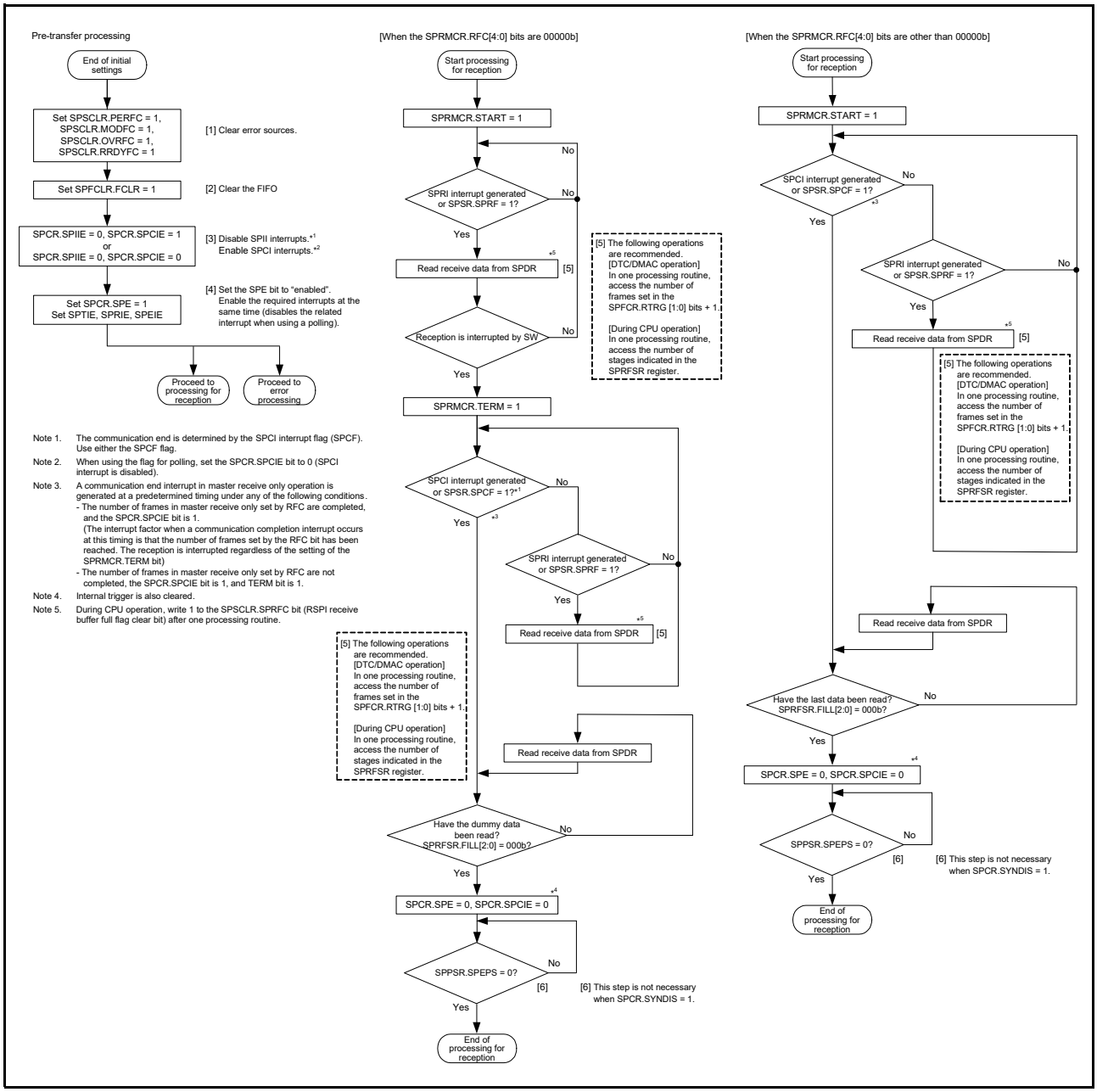


Figure 47.63 Flowchart in Master Mode (Receive-Only)

(d) Flow of Error Processing

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

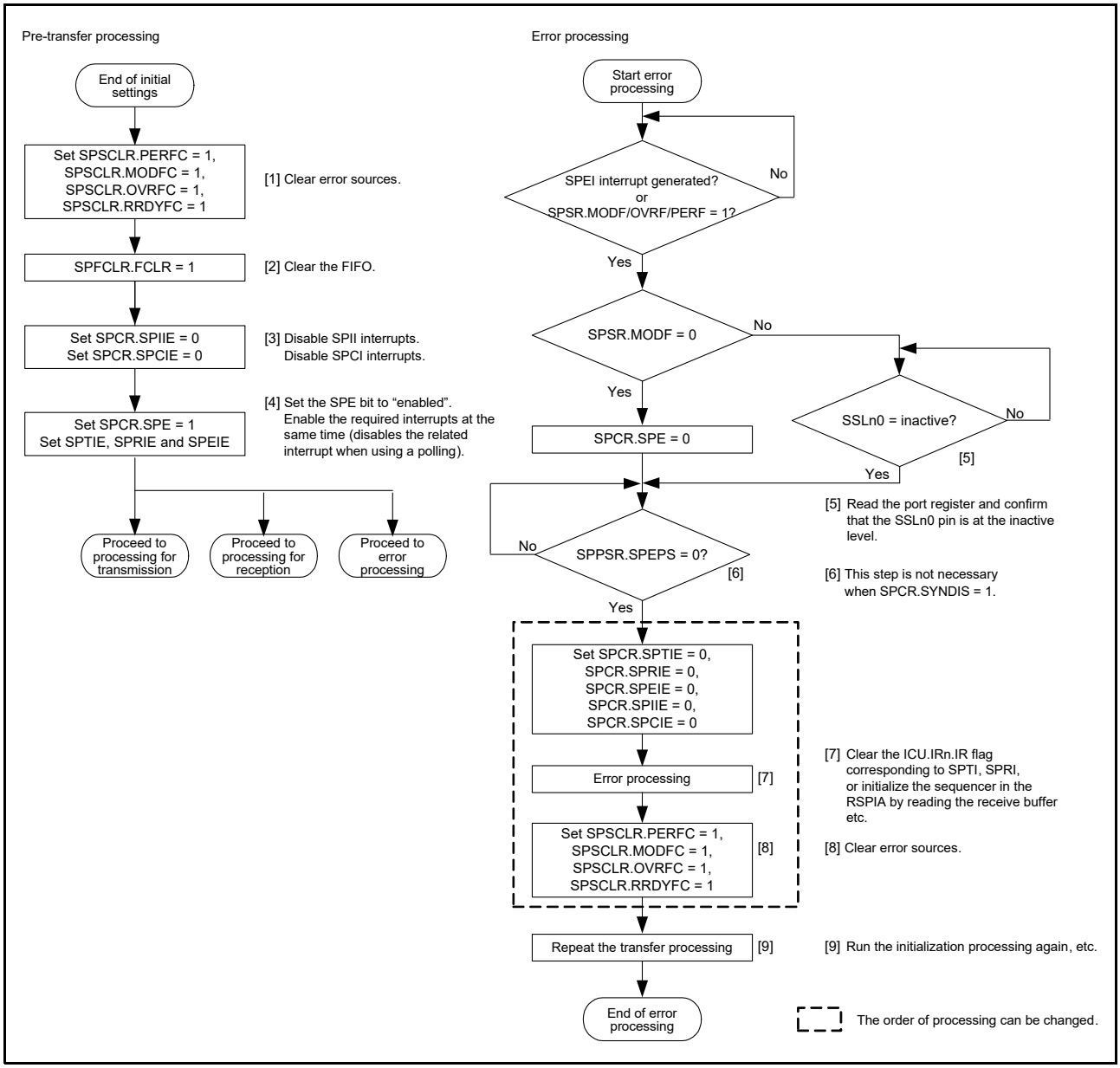


Figure 47.64 Flowchart for Master Mode (Error Processing)

47.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion, the RSPIA needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKn edge in an SSLn0 signal asserted condition, the RSPIA needs to start driving valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.
Irrespective of the CPHA bit setting, the timing at which the RSPIA starts driving of the MISOn output signal is the SSLn0 signal assertion timing. The data which is output by the RSPIA is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format). The polarity of the SSLn0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPIA terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPIA detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 47.3.10, Error Detection).
The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting. The polarity of the SSLn0 input signal depends on the SSLP.SSL0P bit setting. For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).

(3) Notes on Single-Slave Operation

[Motorola SPI]
If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion edge, the RSPIA starts a serial transfer. Upon the SSLn0 input signal is fixed to the active level when using the RSPIA in single slave mode in a configuration as shown in Figure 47.7, the RSPIA cannot start a serial transfer correctly while the CPHA bit is 0. To correctly perform transmission and reception of the RSPIA in slave mode in a configuration where the SSLn0 input signal is fixed to the active level, set the CPHA bit to 1. If the CPHA bit must be set to 0, do not fix the SSLn0 input signal to active level.

[TI SSP]
In the TI SSP protocol, upon the SSLn0 input signal is fixed to the inactive level when using the RSPIA in single slave mode in a configuration as shown in Figure 47.7, the RSPIA cannot start a serial transfer correctly.
When using the RSPIA in single slave mode, operate in a configuration as shown in Figure 47.6.

(4) Burst Transfer

[Motorola SPI]
When the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) is available while maintaining the SSLn0 input signal assertion state. When the CPHA bit is 1, the period (from the first RSPCKn edge in the SSLn0 input signal active state until the sampling timing for receiving the final bit) corresponds to the serial transfer period. Even if the SSLn0 input signal is always at the active level, start of an access can be detected, which allows burst transfer.
When the CPHA bit is 0, the second and subsequent serial burst transfers cannot be performed correctly for the same reason as (3).

[TI SSP]
In serial transfer, data transfer starts after the SSLni input signal is asserted for 1 RSPCK cycle. Since frame transfer starts from the SSLni input signal, SSL must be asserted between frames.

(5) Initialization Flowchart

Figure 47.65 shows an example of initialization flow when using the RSPIA in slave-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

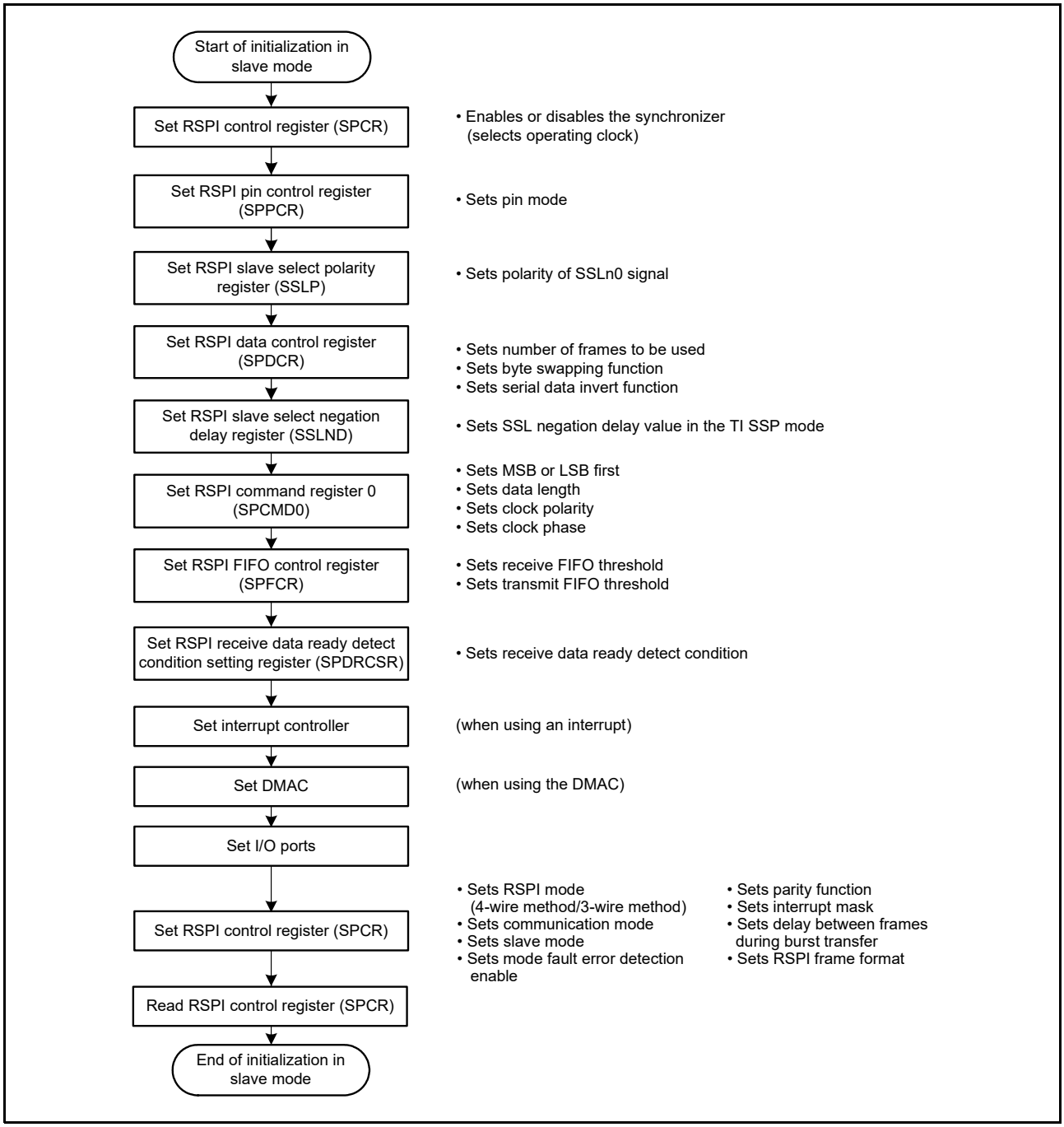
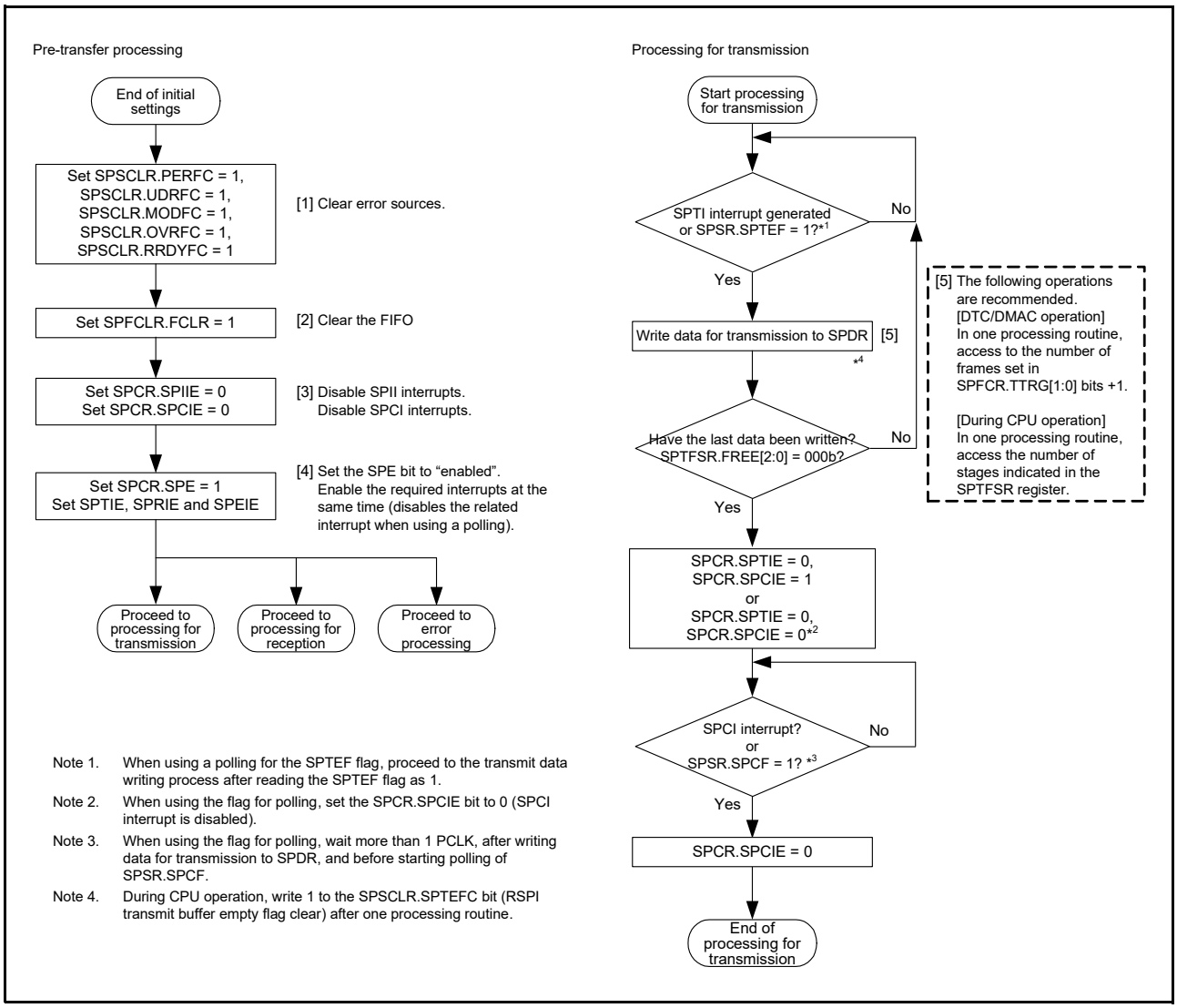


Figure 47.65 Example of Initialization Flow in Slave Mode

(6) Software Processing Flow

Figure 47.66 to Figure 47.69 show examples of the flow of software processing.

(a) Transmit Processing Flow



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Note 2. When using the flag for polling, set the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).

Note 3. When using the flag for polling, wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.SPCF.

Note 4. During CPU operation, write 1 to the SPSCLR.SPTEFC bit (RSPI transmit buffer empty flag clear) after one processing routine.

Figure 47.66 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

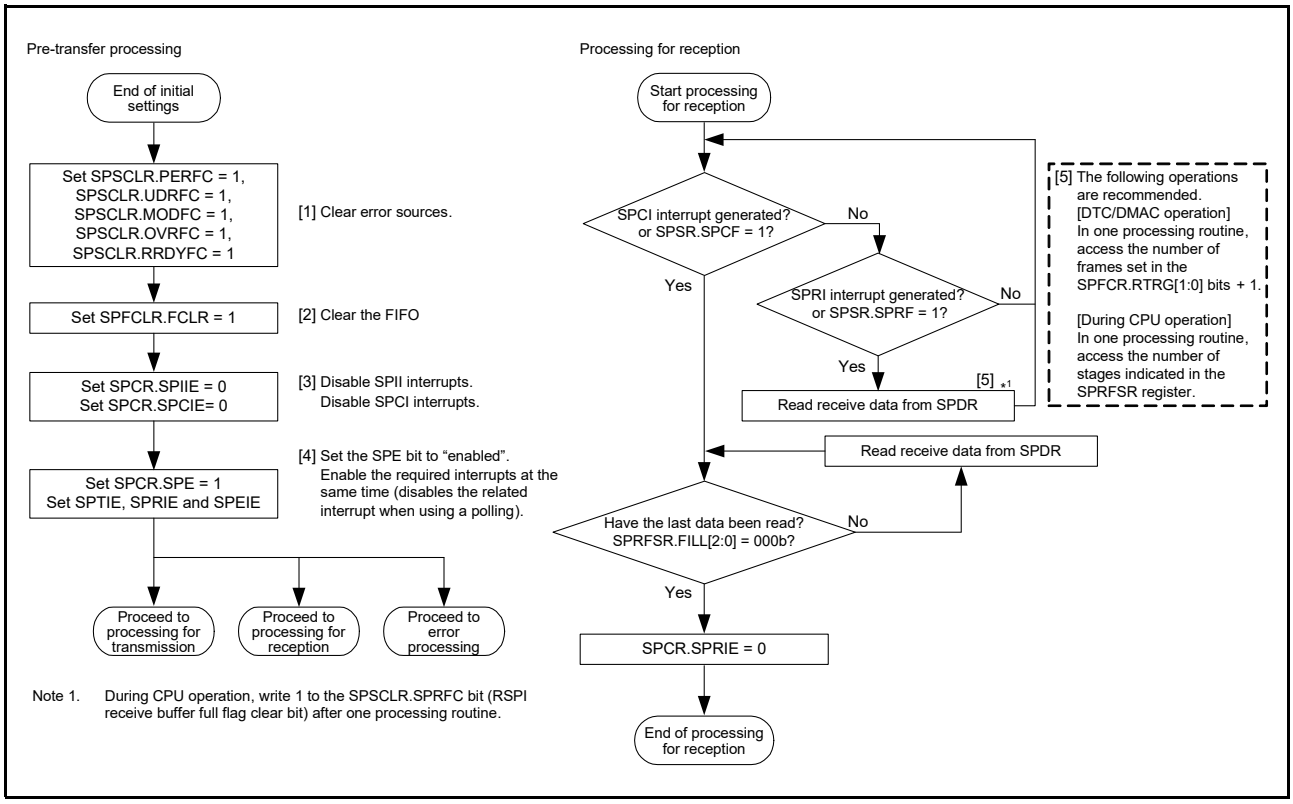
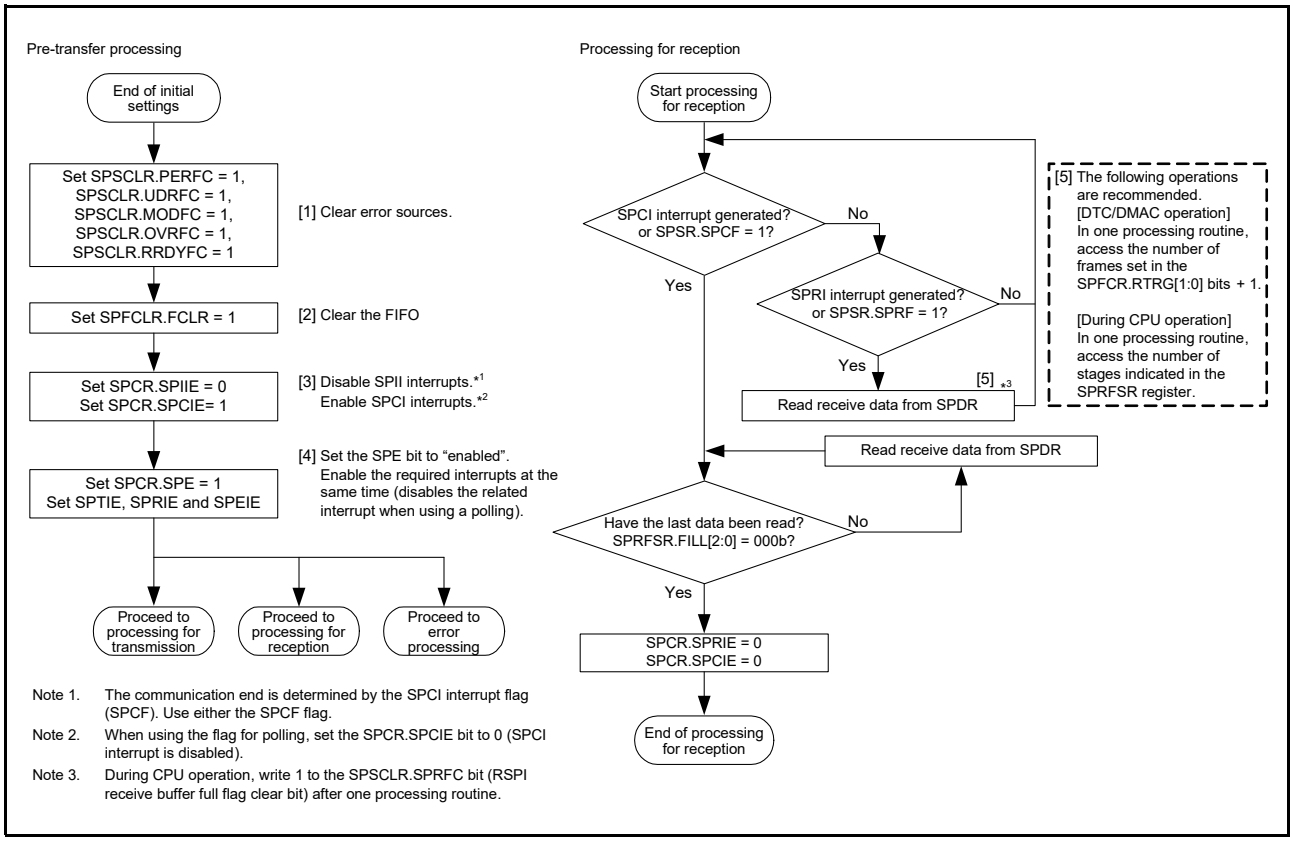


Figure 47.67 Flowchart in Slave Mode (Reception)

(c) Receive-Only Processing Flow



Note 1. The communication end is determined by the SPCI interrupt flag (SPCF). Use either the SPCF flag.
Note 2. When using the flag for polling, set the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).
Note 3. During CPU operation, write 1 to the SPSCLR.SPRFC bit (RSPI receive buffer full flag clear bit) after one processing routine.

Figure 47.68 Flowchart in Slave Mode (Receive-Only)

(d) Flow of Error Processing

In slave mode operation, the MODF flag can be cleared regardless of the SSLn0 pin status even when a mode fault error is generated.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

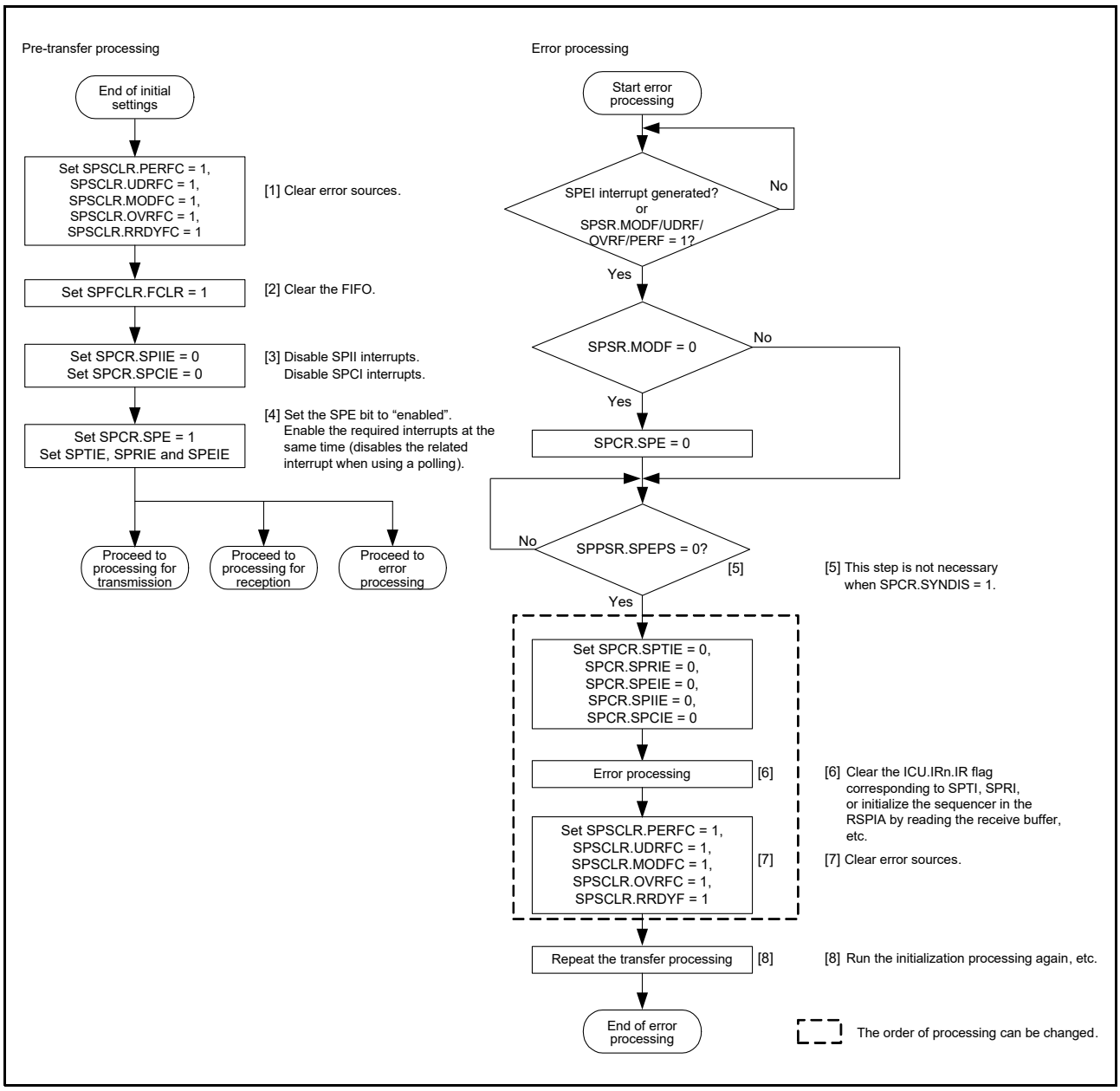


Figure 47.69 Flowchart in Slave Mode (Error Processing)

47.3.14 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPIA. In clock synchronous operation, the SSLni pin is not used, and the three pins of RSPCKn, MOSIn, and MISOn handle communications. The SSLni pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLni pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLni pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

47.3.14.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) of the SPDR register when data is written to the SPDR register with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty, the RSPIA copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPIA changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSLni output signal.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after transmitting an RSPCKn edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bit setting. For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSLni output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by the following registers:

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.
According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA contains a pointer to the SPCMDm register that makes up the sequence.

The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPIA loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPIA increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPIA sets the pointer in the SPCMD0 register, and in this manner the sequence is executed repeatedly.

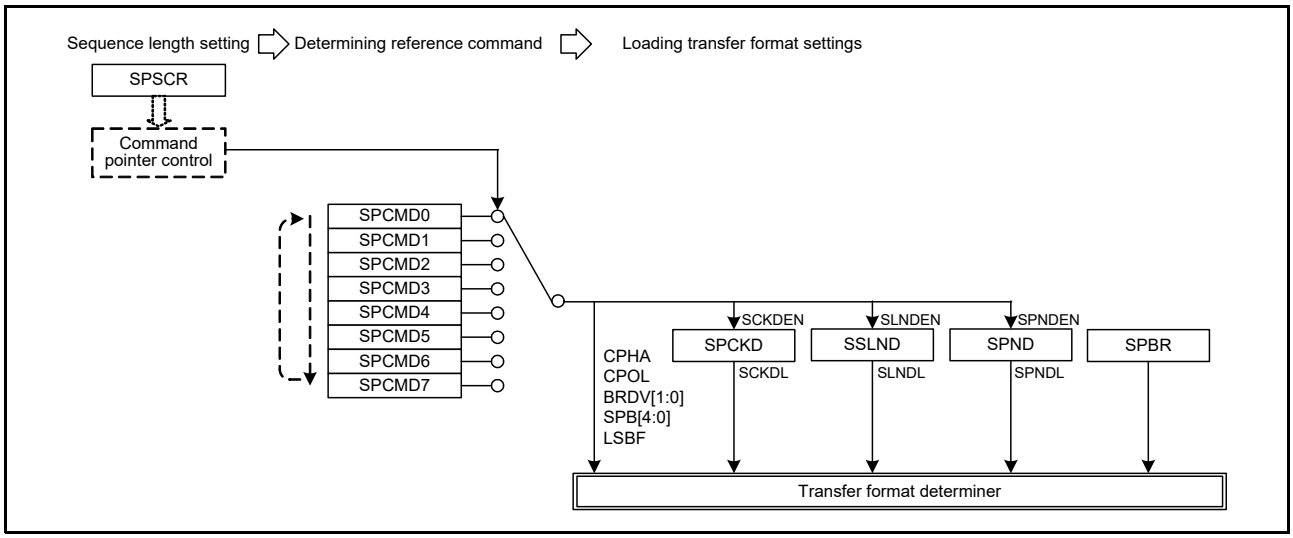


Figure 47.70 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

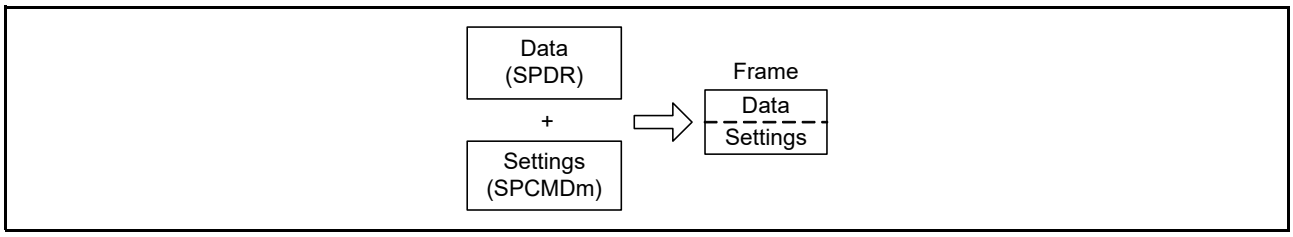


Figure 47.71 Concept of a Frame

Figure 47.72 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in the sequence of operations.

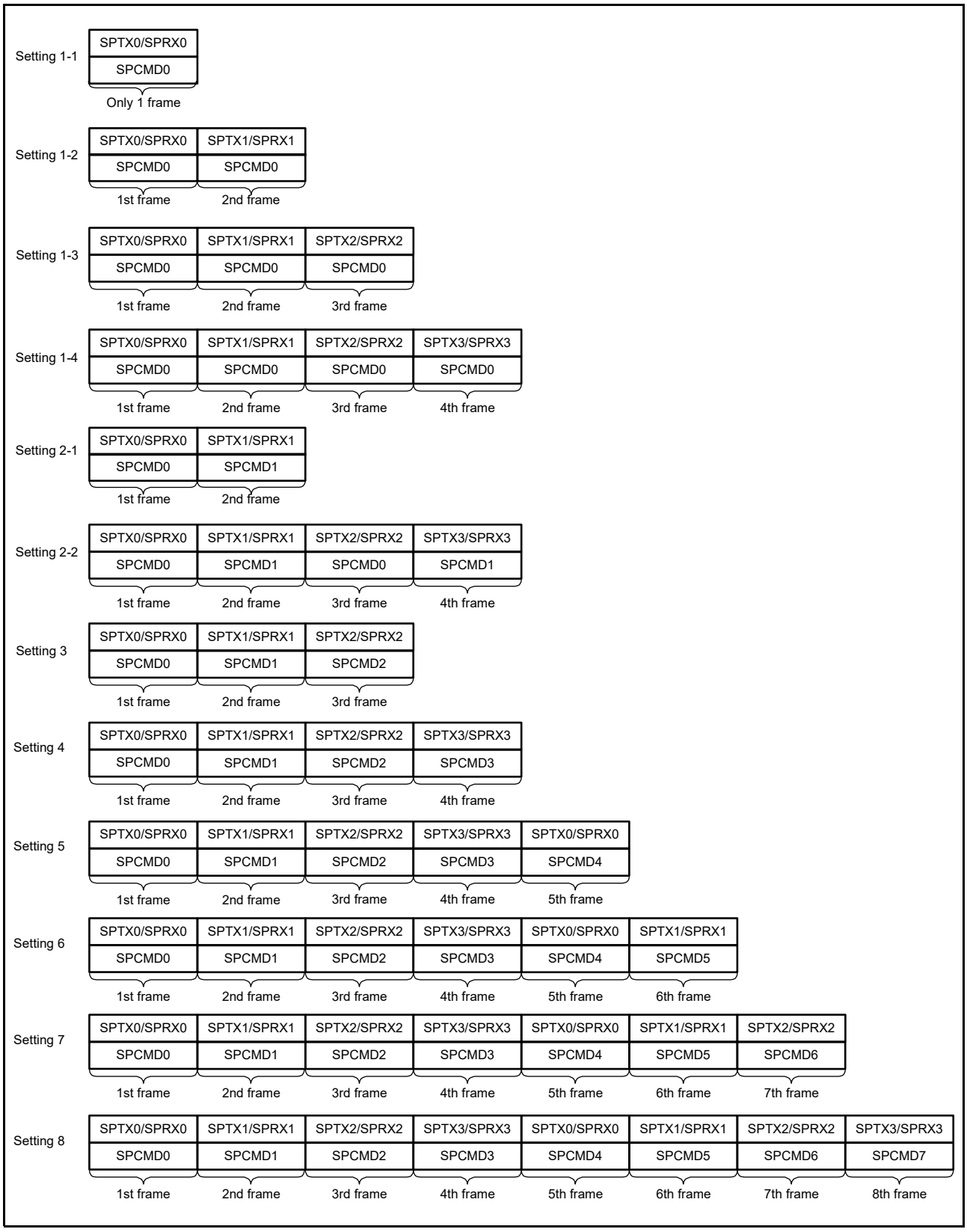


Figure 47.72 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 47.73 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

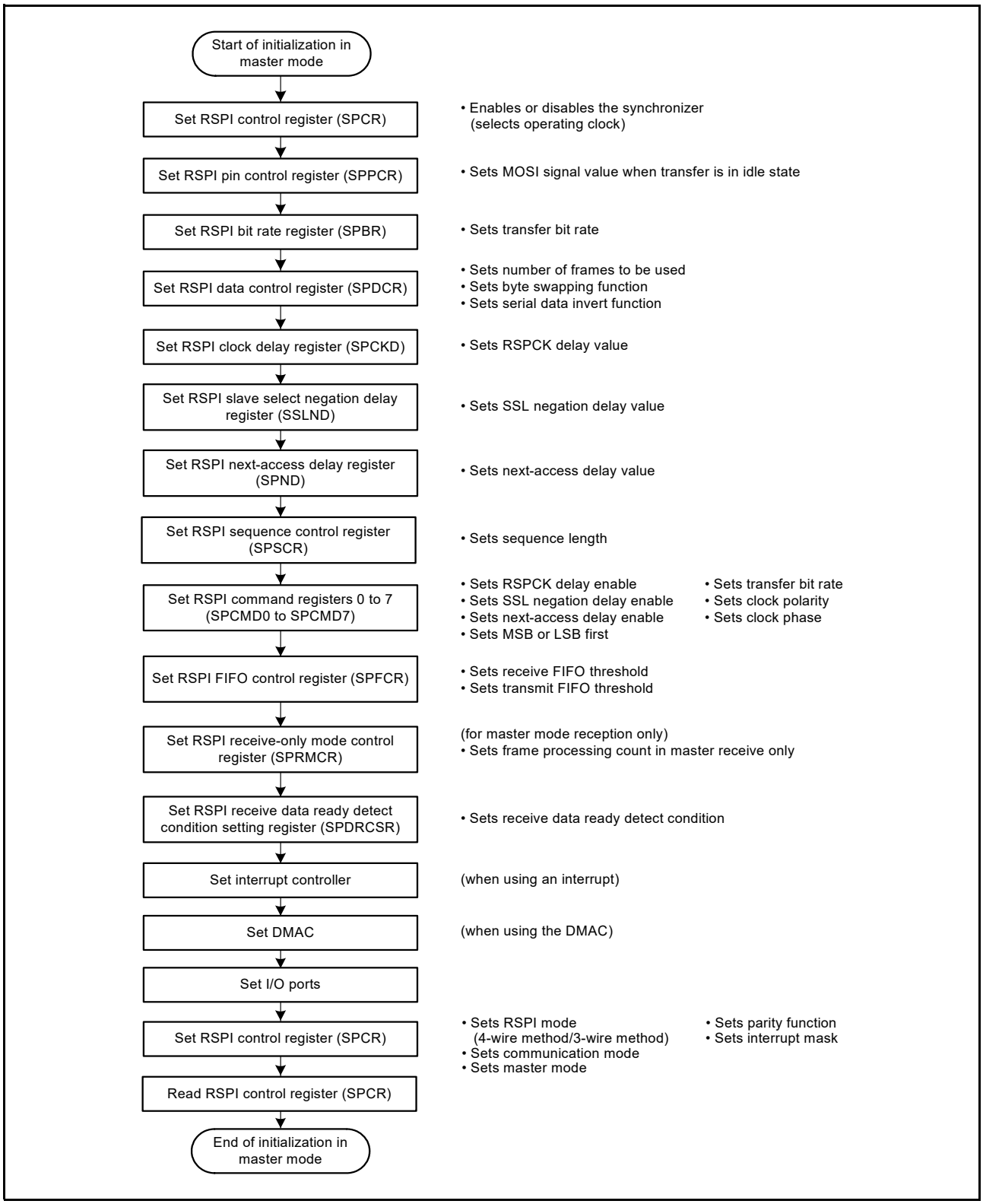


Figure 47.73 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in master mode is the same as that for SPI master mode operation. For details, refer to section 47.3.13.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

47.3.14.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the RSPIA.
When the SPMS bit is 1, the RSPIA drives the MISOn output signal.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).
It should be noted that the SSLn0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting.
For details on the RSPI transfer format, refer to section 47.3.5, Transfer Format (Frame Format).

(3) Initialization Flowchart

Figure 47.74 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

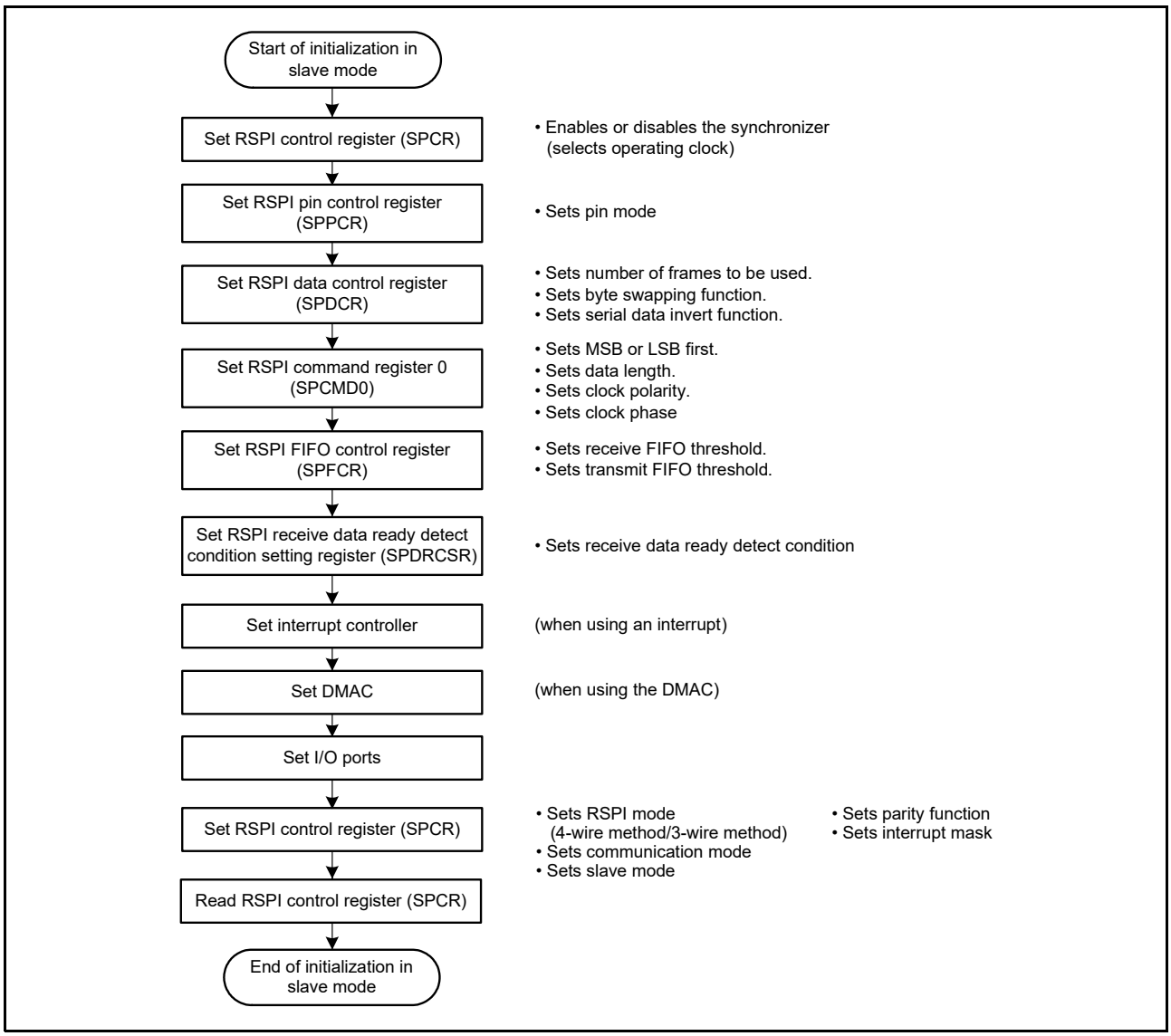


Figure 47.74 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in slave mode is the same as that for SPI slave mode operation. For details, refer to section 47.3.13.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

47.3.15 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPIA shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPIA does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode.
When a serial transfer is executed in loopback mode, the transmit data for the RSPIA or the reversed transmit data becomes the received data for the RSPIA.
Table 47.13 lists the relationship among the SPLP2 and SPLP bits and the received data.

Table 47.13 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	x	Transmit data

x: Don’t care.

Figure 47.75 shows the configuration of the shift register I/O paths in loopback mode.

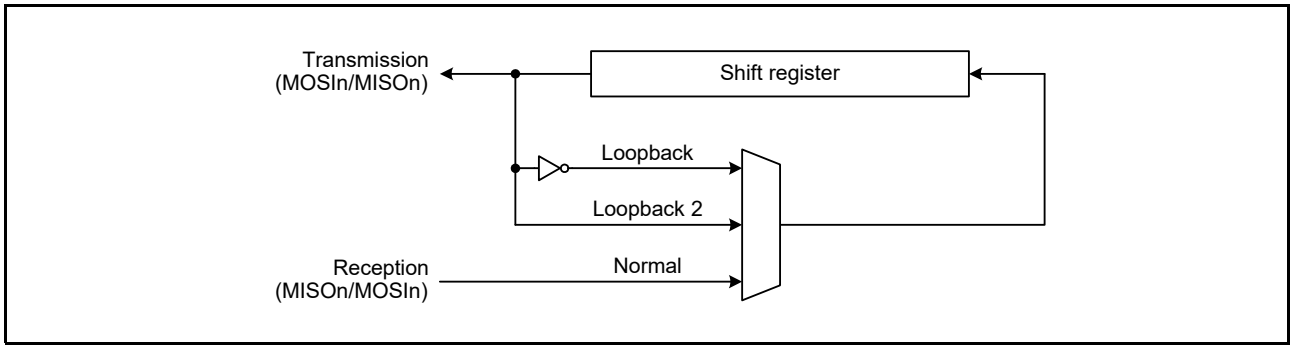


Figure 47.75 Configuration of Shift Register I/O Paths in Loopback Mode

47.3.16 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 47.76.

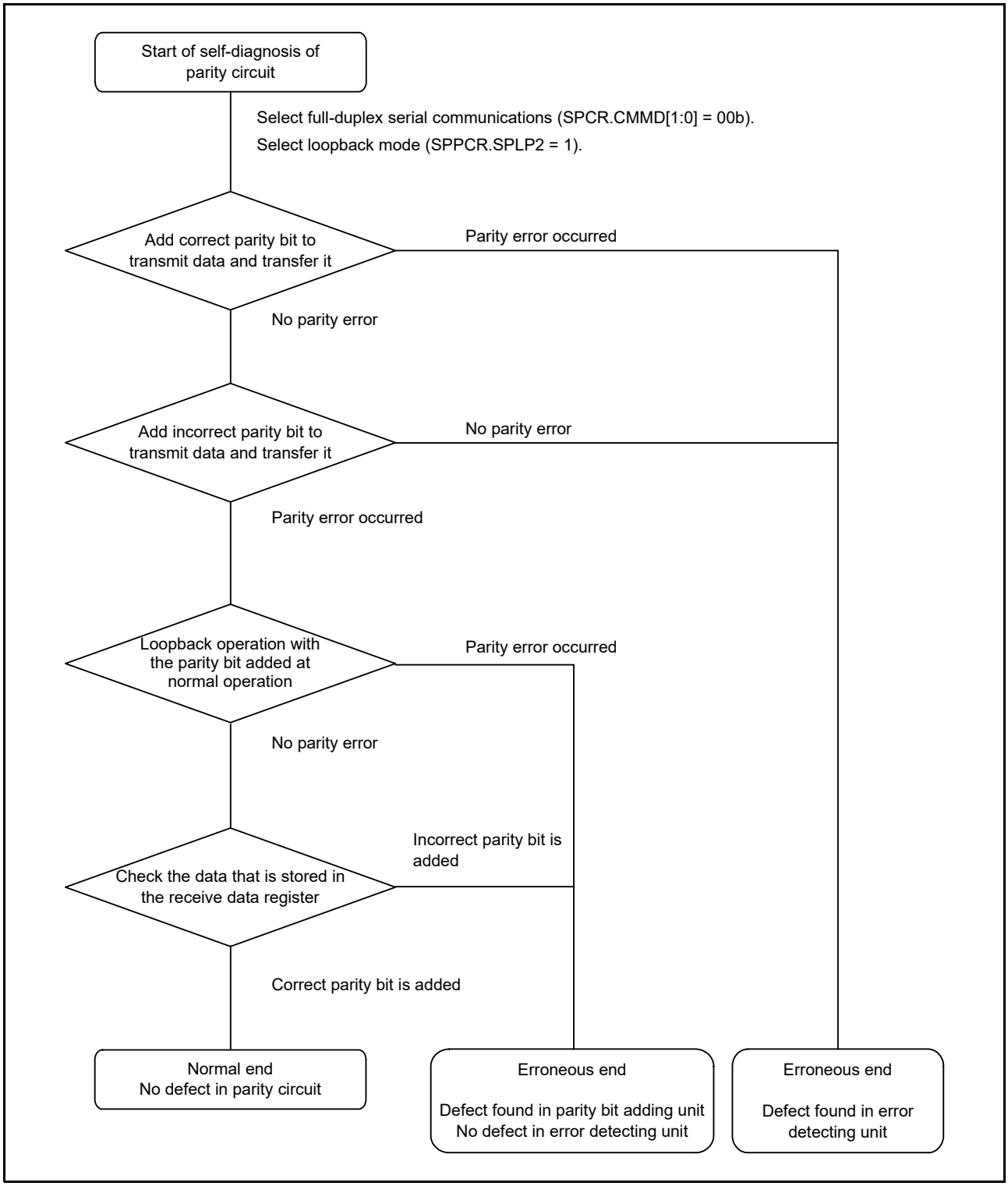


Figure 47.76 Flowchart for Self-Diagnosis of Parity Circuit

47.3.17 Interrupt Sources

The RSPIA has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), idle, communication end, and receive data ready. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, underrun, overrun, parity errors, and receive data ready (only when the RDRIS bit is 1), the actual interrupt source must be determined from the flags.

Interrupt sources for the RSPIA are listed in Table 47.14. An interrupt is generated on satisfaction of an interrupt condition in Table 47.14. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPIA settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACAa), or section 20, Data Transfer Controller (DTCA).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 47.14 Interrupt Sources of RSPIA

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full Receive data ready	SPRI	The SPSR.SPRF flag is set to 1 while the SPCR.SPRIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE is 1 and RDRIS bit is 0.	Possible
Transmit buffer empty	SPTI	The SPSR.SPTEF flag is set to 1 while the SPCR.SPTIE bit is 1.	Possible
Mode fault error Underrun error Overrun error Parity error Receive data ready	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE and RDRIS bits are 1.	Impossible
Idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR.SPIIE bit is 1.	Impossible
Communication end	SPCI	The SPSR.SPCF flag is set to 1 while the SPCR.SPCIE bit is 1.	Impossible

47.3.18 Link Operation by Event Linking

The RSPIA supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

47.3.18.1 Receive Buffer Full Event Output

This event signal is output when the number of data stored in the receive FIFO is greater than the threshold value, or when the SPCR.RDRIS bit = 0 and after writing to the receive FIFO, and the number of data stored in the receive FIFO is equal to or less than the threshold value, the value set in the SPDRCSR register has elapsed.
A receive buffer full event is output at the timing shown in Figure 47.77.

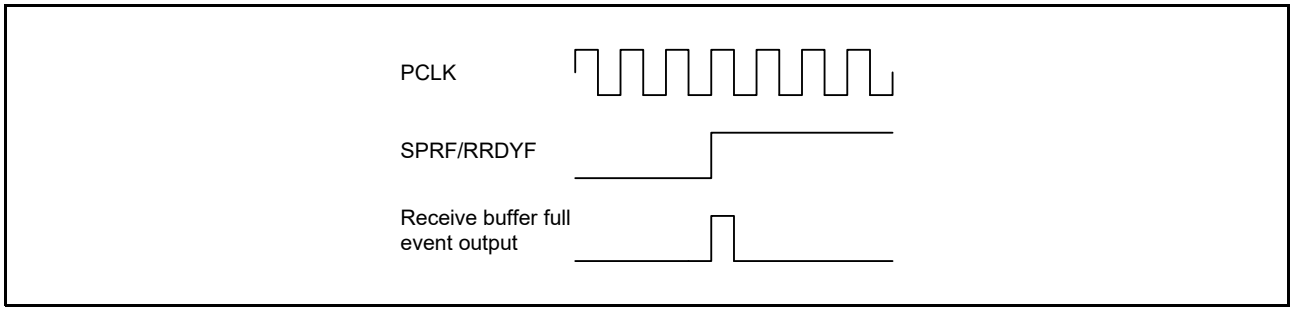


Figure 47.77 Event Output Timing of the Receive Buffer Full

47.3.18.2 Transmit Buffer Empty Event Output

This event signal is output when the number of empty stages in the transmit FIFO is greater than the threshold value or when the value of the SPCR.SPE bit has changed from 0 to 1.
A transmit buffer empty event is output at the timing shown in Figure 47.78.

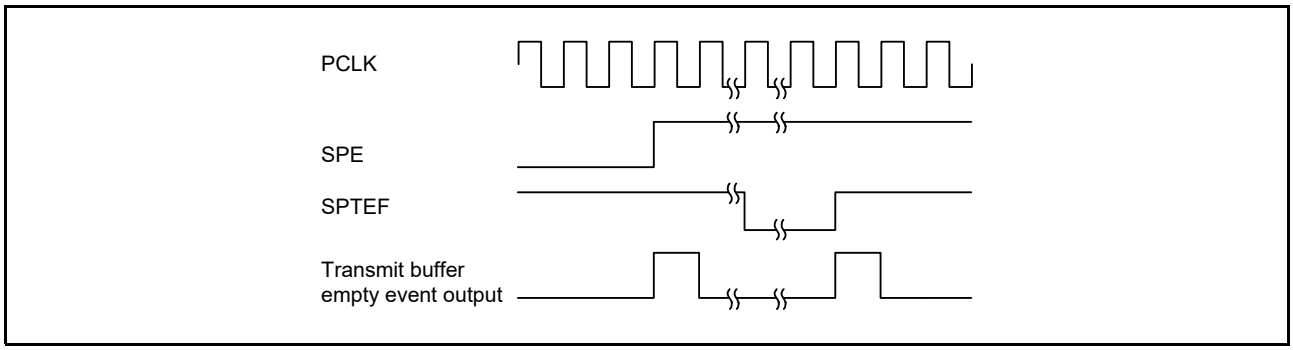


Figure 47.78 Event Output Timing of the Transmit Buffer Empty

47.3.18.3 Error Event Output

The error event signal is output due to five sources (mode fault, underrun, overrun, parity error, and receive data ready). For notes on error event output, see section 47.4.3, Note on Error Events Output.

(1) Mode Fault

A mode fault event occurs when all the following conditions are satisfied:

- SPI operation (SPCR.SPMS = 0)
- Slave mode (SPCR.MSTR = 0)
- The detection of mode fault error is enabled (SPCR.MODFEN = 1)
- Any of the following conditions are satisfied
  1. In the Motorola SPI protocol, the SSLn0 pin is deactivated during data transfer.
  2. In the TI SSP protocol, the SSLn0 pin is activated during data transfer.

(2) Underrun

The condition for this event signal being output in response to an underrun is the start of serial transfer while the value of the SPCR.MSTR bit is 0 (slave mode), the value of the SPCR.SPE bit is 1 (enables the RSPI function), and the transmit data are not ready for output.

(3) Overrun

The condition for this event signal being output in response to an overrun is completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the data is stored in the receive FIFO for the number of FIFO stages, in which case the SPSR.OVRF flag is set to 1.

(4) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the value of the SPCR.SPE bit is 1.

(5) Receive Data Ready

The condition for this event signal being output in response to a receive data ready is elapsing of the value set by the SPDRCSR register when the number of data stored in the receive FIFO is less than or equal to the receive FIFO threshold value after the data has stored in the receive FIFO, while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, the value of the SPCR.RDRIS bit is 1.

(6) Output Timing

Each error event signal is output at the timing shown in Figure 47.79.
Error event is output each time an error occurs.

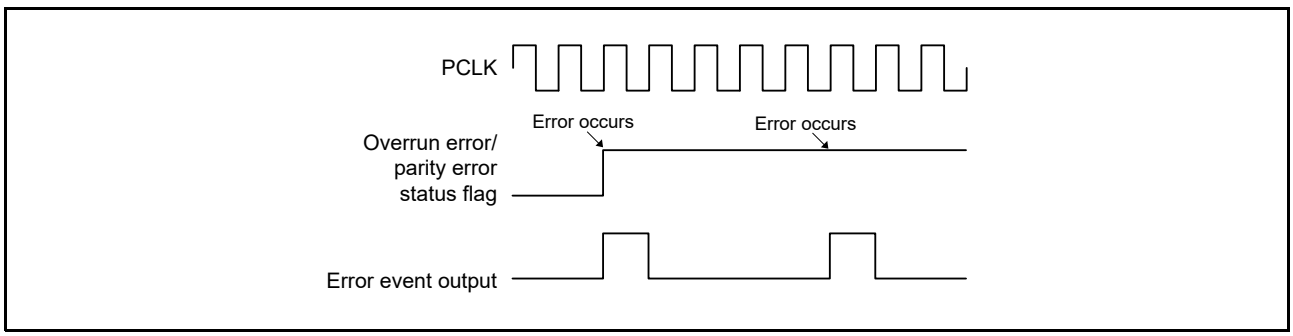


Figure 47.79 Error Event Output Timing

47.3.18.4 Idle Event Output

(1) In Master Mode

[Transmit-receive mode/transmit-only mode]

In transmit-receive master mode or transmit-only master mode, an event is output when the SPSR.IDLNF flag (idle flag) changes from 1 to 0.

The SPSR.IDLNF flag changes from 1 to 0 only when any of the following conditions are satisfied.

(a) The SPCR.SPE bit becomes 0 (RSPI is initialized) during transmission

(b) All the following conditions are satisfied:

- The next transmit data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

[Receive-only mode]

In receive-only master mode, an event is output when any of the following conditions are satisfied.

(a) The SPCR.SPE bit is set to 0 (RSPI is initialized).

(b) Any the following conditions are satisfied:

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

(3) Output Timing

An idle event is output at the timing shown in Figure 47.80.

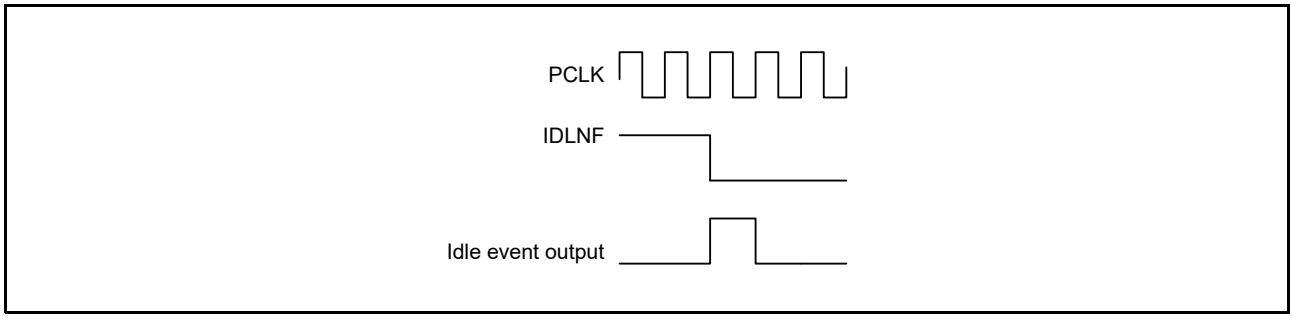


Figure 47.80 Idle Event Output Timing

47.3.18.5 Communication End Event Output

In master mode, an event is output under the condition for setting the SPSR.IDLNF flag (idle flag) from 1 to 0. In slave mode, an event is output under the conditions listed in Table 47.15 and Table 47.16.

Table 47.15 Generating Conditions of Communication End Event (Transmit-Receive Slave Mode/Transmit-Only Slave Mode)

RSPI Mode	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0, FRFS = 0)	Empty	Empty	Negation of the SSLn0 input
SPI operation (SPMS = 0, FRFS = 1)	Empty	Empty	Completion of the SSL negation delay
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCKn for the last data

Table 47.16 Generating Conditions of Communication End Event (Receive-Only Slave Mode)

RSPI Mode	Condition
SPI operation (SPMS = 0, FRFS = 0)	Negation of the SSLn0 input after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
SPI operation (SPMS = 0, FRFS = 1)	Completion of the SSL negation delay after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCKn for the last frame set by the SPDCR.SPFC[1:0] bits

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit during communication or the SPCR.SPE bit is cleared by the mode fault error or underrun error.
A communication end event is output at timings shown in Figure 47.81 to Figure 47.86. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

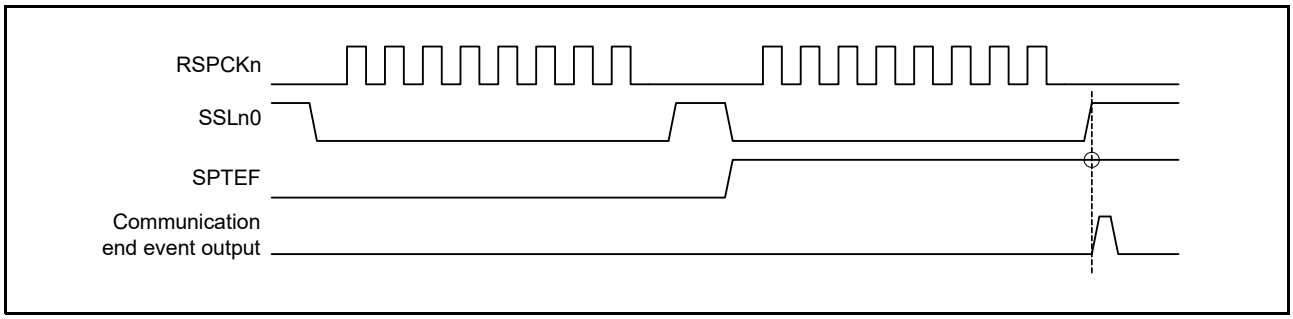


Figure 47.81 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI)

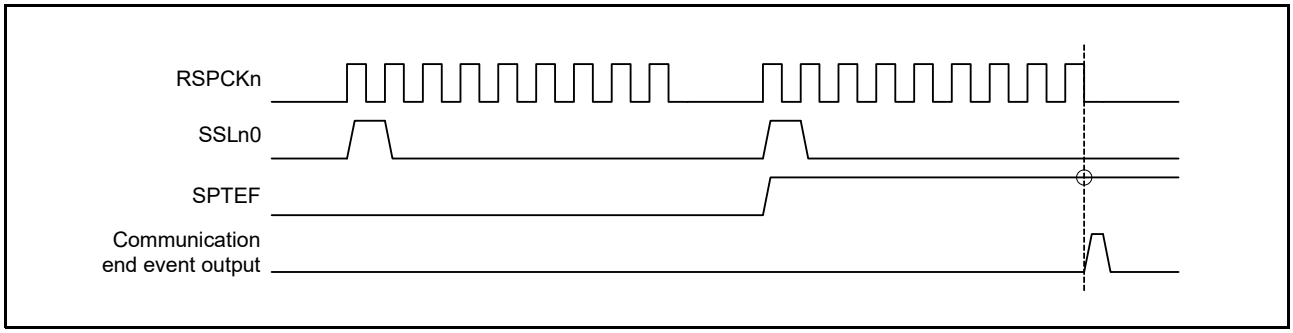


Figure 47.82 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP)

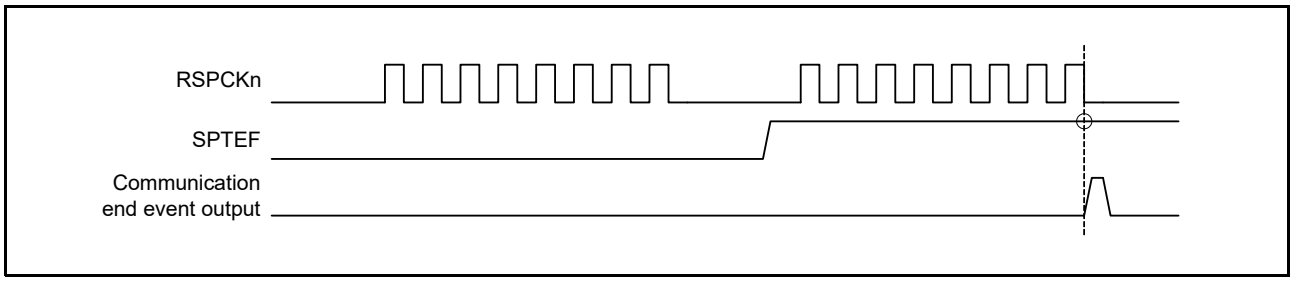


Figure 47.83 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

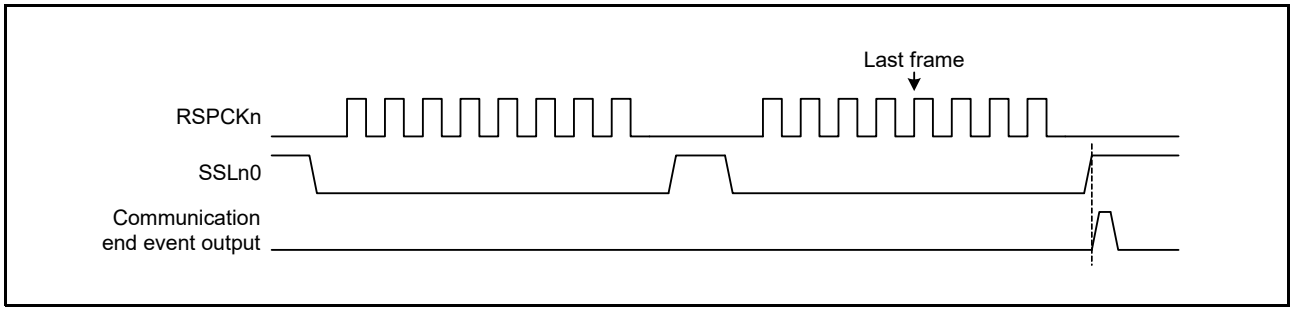


Figure 47.84 Communication End Event Output Timing (Receive-Only Slave Mode, Motorola SPI)

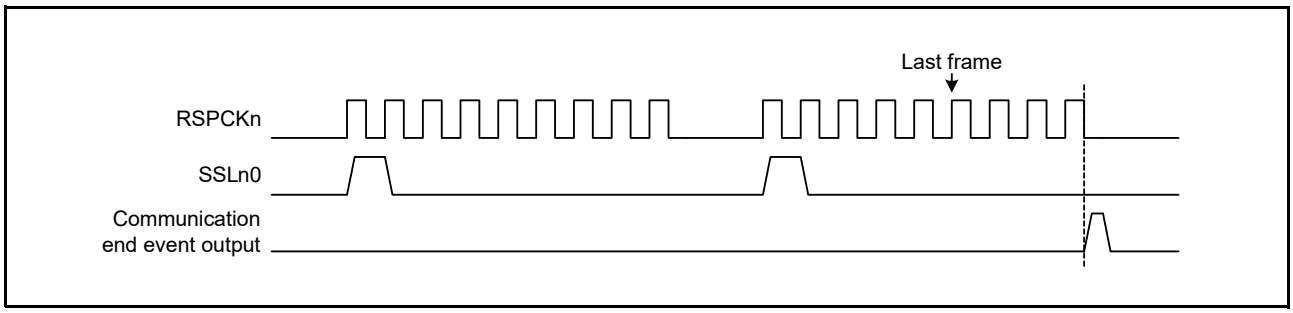


Figure 47.85 Communication End Event Output Timing (Receive-Only Slave Mode, TI SSP)

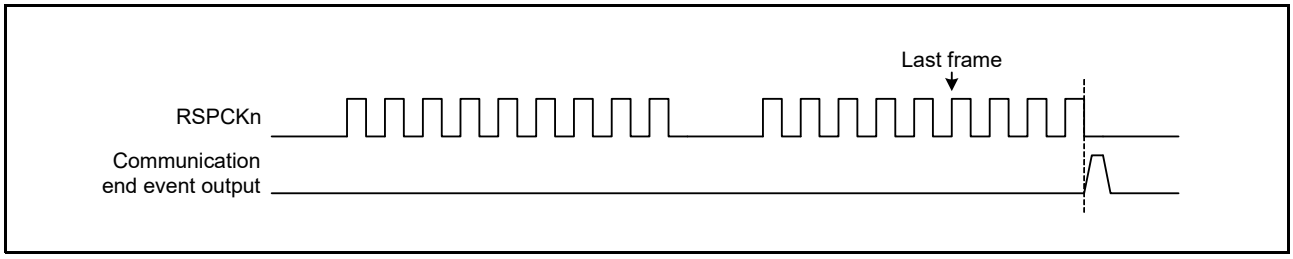


Figure 47.86 Communication End Event Output Timing (Receive-only slave mode, Clock Synchronous Operation)

47.3.19 Synchronizer Bypass Function

RSPIA has circuits that operate in synchronization with the peripheral module clock (PCLKA) and circuits that operate in synchronization with the operating clock (TCLK). A synchronizer is implemented to transfer signals across these two types of circuits, but the signals passing through the synchronizer are delayed by 2 to 3 PCLKA or 2 to 3 TCLK.
However, when the SPCR.SYNDIS bit is set to 1, the peripheral module clock is selected as the operating clock and the synchronizer is bypassed. In this case, the above delay is eliminated and responsiveness is improved.
Note that RSPIA also has a synchronizer between the circuits that operate in synchronization with the serial clock (RSPCK) and the circuits that operate in synchronization with the operating clock, but this synchronizer cannot be bypassed.

47.4 Usage Notes

47.4.1 Settings of the Module Stop Function

RSPIA operation can be disabled or enabled by the module stop control register B (MSTPCRB) for channels 0 and 1 or by the module stop control register C (MSTPCRC) for channel 2. After a reset, the operation of the RSPIA is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

47.4.2 Notes on Starting Transfer

When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

47.4.3 Note on Error Events Output

To enable SPI operation of the RSPIA in master mode in the multi-master environment (SPCR.SPMS bit = 0, SPCR.MSTR bit = 1, and SPCR.MODFEN bit = 1), do not use any output of error events.

47.4.4 Note on Low Power Consumption Functions

To reduce power consumption of this module by using the low power consumption functions, set the SPCR.SPE bit to 0 to terminate communication, and then use the low power consumption functions.

47.4.5 Note on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

47.4.6 Notes on Burst Transfer in Master Mode

During burst transfer, changing the following settings between the SPCMDm register with the SSLKP bit set to 1 and the SPCMDm register used in the next transfer is prohibited.

- SSL Signal Assertion setting (SSLA[2:0] bits)
- RSPCK output setting (CPHA, CPOL, BRDV[1:0] bits)

47.4.7 Notes on Operating in TI SSP Slave Mode

In the TI SSP slave mode, the delay between frames must observe the following intervals. Secure the following intervals on the master side.

[Master side: interval from the RSPCK edge of last bit to the RSPCK edge of next SSL] > [Slave side: OE delay time = TCLK × (1 or 2) + SLNDL[2:0] setting value]

47.4.8 Notes on Data Length and Frequency Ratio

Table 47.17 shows whether RSPIA can operate for each combination of the frequency ratio of PCLKA and TCLK and the data length.

Table 47.17 Operation Possibilities Based on Frequency Ratio of PCLKA and TCLK and Data Length (SPCMDm.BRDV[1:0] = 00b, SPBR = 00h)

		Data Length										
RSPI Mode	**PCLKA:TCLK**	**4 bits**	**5 bits**	**6 bits**	**7 bits**	**8 bits**	**9 bits**	**10 bits**	**11 bits**	**12 bits**	**13 bits**	**14 bits**
SPCR. SYNDIS = 0	1:4	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Possible
	1:2	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Possible	Possible	Possible	Possible
	2:3	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Possible	Possible	Possible	Possible
	1:1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Possible	Possible	Possible	Possible	Possible	Possible
	150 MHz : 32 kHz	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Conditionally possible *1	Possible	Possible	Possible	Possible	Possible	Possible	Possible
SPCR. SYNDIS = 1	1:1	Conditionally possible *2	Conditionally possible *2	Conditionally possible *2	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible

Possible: Operation is possible
Conditionally possible: Operation is possible under certain conditions.
Note 1. Operation is possible when the SPCR.SCKASE and SCKDDIS bits are 0.
Note 2. Operation is possible when the SPCR.SCKASE or SCKDDIS bit is 0.

47.4.9 Notes on Rewriting Registers

If the following registers or bits are rewritten while the SPCR.SPE bit is 1, subsequent operation is not guaranteed. When the SPCR.SPE bit is rewritten from 1 to 0 or the SPE bit becomes 0 because the SPSR.MODF flag is set to 1, while the SPCR.SYNDIS bit is 0, confirm that the SPPSR.SPEPS flag is set to 0, and then proceed to the next processing.

Table 47.18 List of Registers and Bits that Cannot Be Rewritten When the SPCR.SPE Bit Is 1 (1/2)

Register	**Bit**
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]

Table 47.18 List of Registers and Bits that Cannot Be Rewritten When the SPCR.SPE Bit Is 1 (2/2)

Register	Bit
SPCR	SYNDIS
	MSTR
	CMMD[1:0]
	FRFS
	SPMS
	MODFEN
	SCKDDIS
	SCKASE
	PTE
	SPOE
	SPPE
SPRMCR	RFC[4:0]
SPDRCSR	—
SPPCR	MOIFE
	MOIFV
	SPLP2
	SPLP
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	—
SPSCR	SPSLN[2:0]
SPCMD0*1	SSLA[2:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA
SPDCR	SPFC[1:0]
	DINV
	SPRDTD
	BYSW
SPFCR	TTRG[1:0]
	RTRG[1:0]
SPFCLR	FCLR

Note 1. Rewriting is prohibited in slave mode. In master mode, rewriting is possible only when there is no next transmit data in the transmit FIFO.

48. Expanded Serial Peripheral Interface (XSPI)

This MCU incorporates two units of expanded serial peripheral interface (XSPI0, XSPI1) compliant with the xSPI (eXpanded SPI) standard.
XSPI consists of a controller module and a target module, and the software can select and use either module.

48.1 Overview

XSPI complies with JEDEC standards JESD251 (Profile 1.0, 2.0), JESD251-1, and JESD252.
JESD251 Profile 1.0 supports Octal xSPI. JESD251 Profile 2.0 supports HyperBus™ (HyperRAM™ and HyperFlash™).
Table 48.1 lists the specifications of the XSPI controller module, Table 48.2 lists the specifications of the XSPI target module, Figure 48.1 shows the block diagram of the XSPI controller module, Figure 48.2 shows the block diagram of the XSPI target module, and Table 48.3 lists the I/O pins.

Table 48.1 XSPI Controller Module Specifications

Item	Description
Number of units	Two units
Supported standards	JESD251 (xSPI) Profile 1.0 (Octal xSPI), 2.0 (HyperBus™) JESD251-1 (x4 Quad I/O with Data Strobe) JESD252 (Serial Flash Reset Signaling Protocol)
Number of target	Up to two target devices can be connected.
Operating clock	Register access: PCLKB Communication clock: XSPIDIVCLK (up to 133 MHz), XSPICLK (up to 266 MHz)
Transfer speed	Supports xSPI-266: up to 133 MHz, up to 266 MT/s
IO protocol mode	Transaction phase: Command–Modifier–Data Transfer bit width: 1 bit, 2 bits, 4 bits, or 8 bits Data rate: SDR, DDR Address: 8 bits, 16 bits, 24 bits, 32 bits, or 45 bits Initial access latency: Configurable within the range of 0 to 31 for each command Supports XIP mode
Protocol mode	Supports the following seven protocol modes: • 1S-1S-1S (xSPI) • 1S-2S-2S (legacy SPI) • 2S-2S-2S (legacy SPI) • 1S-4S-4S (legacy SPI) • 4S-4S-4S (legacy SPI) • 4S-4D-4D (xSPI) • 8D-8D-8D (xSPI)
Access mode	Memory-mapped mode • Up to 256 Mbytes per area • Prefetch function: 128-byte prefetch buffer • 128-byte write buffer • Supports on-the-fly decryption of read data*1 Indirect access mode • Sequential mode: Up to four transactions can be issued consecutively • Periodic mode: Allows periodic checking of the status of the xSPI device
Others	Supports DS signal compatible with read data strobe/write data mask Automatic calibration function for read data strobe signal Supports In-band Reset
Interrupt sources	Error interrupt Completion interrupt
Module stop function	Module-stop state can be set to reduce power consumption
Protect Zone filter	Security attribute can be set for each unit.*2 Security attribute of the XSPI area used in memory-mapped mode is fixed to Non-secure.

Note 1. For details on on-the-fly decryption, refer to section 49, Decryption On The Fly (DOTF).
Note 2. When each unit is set to the secure attribute, registers in the unit are accessible by secure transactions. Any access by non-

secure transaction is ignored, and a Protect Zone access error is generated.
When each unit is set to the non-secure attribute, registers in the unit are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 48.2 XSPI Target Module Specifications

Item	Description
Number of units	Two units
Supported standards	JESD251 (xSPI) Profile 1.0 (Octal xSPI), 2.0 (HyperBus™) JESD251-1 (x4 Quad I/O with Data Strobe) JESD252 (Serial Flash Reset Signaling Protocol)
Operating clock	Register access: PCLKB Communication clock: XSPIn_CK (up to 100 MHz*1)
Transfer speed	Supports xSPI-200: up to 100 MHz, up to 200 MT/s
IO Protocol mode	Transaction phase: Command–Modifier–Data Transfer bit width: 1 bit, 4 bits, or 8 bits Data rate: SDR, DDR Address: 16 bits, 24 bits, or 32 bits Initial access latency: Configurable within the range of 11 to 31 for read and 0 to 31 for write
Protocol mode	Supports the following three protocol modes: • 1S-1S-1S (xSPI) • 4S-4D-4D (xSPI) • 8D-8D-8D (xSPI)
Access mode	Register access • The xSPI controller can access registers within the XSPI target module. Memory access • The xSPI controller can access SRAM0 and SRAM1 via the XSPI target module. • 1 Kbyte of write/read buffer
Module stop function	Module-stop state can be set to reduce power consumption
Protect Zone filter	Security attribute can be set for each unit.*2, *3

Note 1. The frequency of the XSPIn_CK should be 90% or less of the frequency of ICLK.
Note 2. When each unit is set to the secure attribute, registers in the unit are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each unit is set to the non-secure attribute, registers in the unit are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 3. The security attribute of the XSPI target module as a bus master is also the same as this setting.

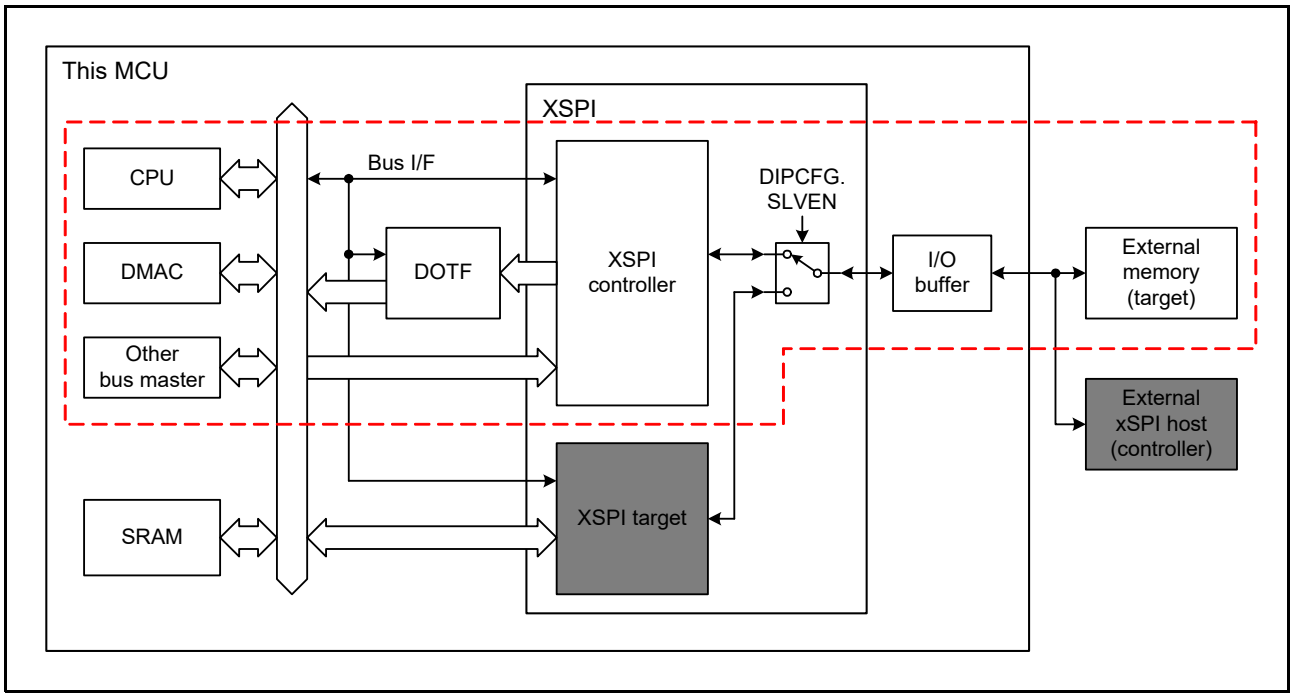


Figure 48.1 XSPI Controller Block Diagram

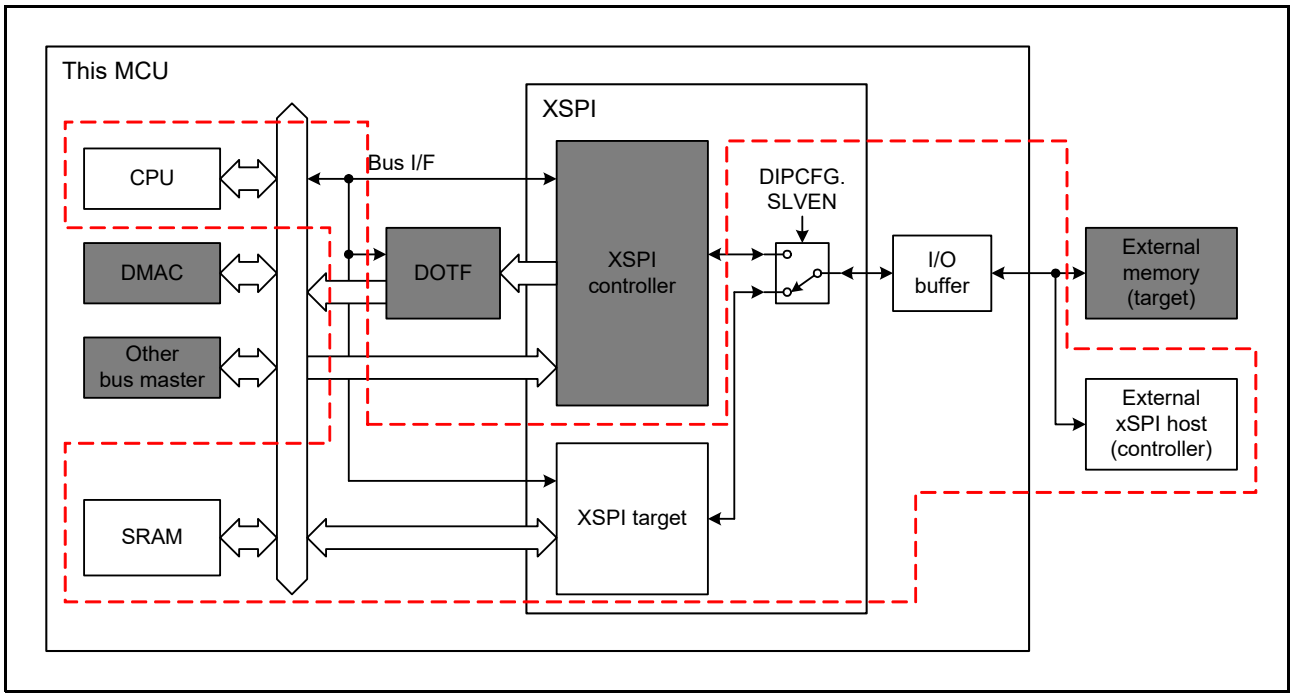


Figure 48.2 XSPI Target Block Diagram

Table 48.3 XSPI Pin Configuration

Unit	Pin Name	I/O		Description
		Controller	Target	
XSPI0	XSPI0_CK	Output	Input	Clock signal
	XSPI0_CKN	Output	—	Inverted clock signal when a differential clock is required
	XSPI0_CS0#	Output	Input	Chip select signal for target device 0
	XSPI0_CS1#	Output	—	Chip select signal for target device 1
	XSPI0_IO7 to XSPI0_IO0	I/O	I/O	Bidirectional data signal
	XSPI0_DS	I/O	I/O	Data strobe signal
	XSPI0_RESET0#	Output	Input	Reset signal for target device 0
	XSPI0_RESET1#	Output	—	Reset signal for target device 1
	XSPI0_RSTO0#	Input	Output	Signal indicating that target device 0 is resetting
	XSPI0_RSTO1#	Input	—	Signal indicating that target device 1 is resetting
	XSPI0_INT0#	Input	Output	Interrupt request signal from target device 0
	XSPI0_INT1#	Input	—	Interrupt request signal from target device 1
	XSPI0_ECS0#	Input	—	Signal indicating an ECC error occurred on target device 0
	XSPI0_ECS1#	Input	—	Signal indicating an ECC error occurred on target device 1
	XSPI0_WP0#	Output	—	Write protect signal for target device 0
	XSPI0_WP1#	Output	—	Write protect signal for target device 1
XSPI1	XSPI1_CK	Output	Input	Clock signal
	XSPI1_CKN	Output	—	Inverted clock signal when a differential clock is required
	XSPI1_CS0#	Output	Input	Chip select signal for target device 0
	XSPI1_CS1#	Output	—	Chip select signal for target device 1
	XSPI1_IO7 to XSPI1_IO0	I/O	I/O	Bidirectional data signal
	XSPI1_DS	I/O	I/O	Data strobe signal
	XSPI1_RESET0#	Output	Input	Reset signal for target device 0
	XSPI1_RESET1#	Output	—	Reset signal for target device 1
	XSPI1_RSTO0#	Input	Output	Signal indicating that target device 0 is resetting
	XSPI1_RSTO1#	Input	—	Signal indicating that target device 1 is resetting
	XSPI1_INT0#	Input	Output	Interrupt request signal from target device 0
	XSPI1_INT1#	Input	—	Interrupt request signal from target device 1
	XSPI1_ECS0#	Input	—	Signal indicating an ECC error occurred on target device 0
	XSPI1_ECS1#	Input	—	Signal indicating an ECC error occurred on target device 1
	XSPI1_WP0#	Output	—	Write protect signal for target device 0
	XSPI1_WP1#	Output	—	Write protect signal for target device 1

Note: Insert series termination registers of 30Ω ± 1% into the XSPIn_IO7 to XSPIn_IO0, XSPIn_CK, XSPIn_CKN, and XSPIn_DS pins (n = 0, 1).

48.2 Register Descriptions

48.2.1 Operating Mode Setting Register (DIPCFG)

Address(es): XSPI0.DIPCFG 8786 73FCh, XSPI1.DIPCFG 8786 77FCh, XSPI0_NS.DIPCFG 0786 73FCh, XSPI1_NS.DIPCFG 0786 77FCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SLVEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SLVEN	Target Mode Enable	0: Controller mode 1: Target mode	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the operating mode (controller/target) of the xSPI.

SLVEN Bit (Target Mode Enable)

This bit specifies whether XSPI is used in controller mode or target mode. It primarily switches the input/output direction of the pins.

Set this bit before an xSPI transaction begins. Changing this bit setting after an xSPI transaction has started is prohibited.

48.2.2 Controller Mode Registers

48.2.2.1 Timing Configuration Register (WRAPCFG)

Address(es): XSPI0.WRAPCFG 8786 8000h, XSPI1.WRAPCFG 8786 8400h, XSPI0_NS.WRAPCFG 0786 8000h, XSPI1_NS.WRAPCFG 0786 8400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	DSSFTCS1[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DSSFTCS0[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	DSSFTCS0[4:0]	DS Phase Shift Setting 0	Set the delay for the DS signal output by target device 0. b12 b8 00000: No delay 00001: Add a delay of 1 cell : : 11110: Add a delay of 30 cells 11111: Add a delay of 31 cells	R/W
b23 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	DSSFTCS1[4:0]	DS Phase Shift Setting 1	Set the delay for the DS signal output by target device 1. b28 b24 00000: No delay 00001: Add a delay of 1 cell : : 11110: Add a delay of 30 cells 11111: Add a delay of 31 cells	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

DSSFTCSi[4:0] Bits (DS Phase Shift Setting i) (i = 0, 1)

These bits configure the delay amount (phase shift amount) for the data strobe signal (XSPIn_DS signal (n = 0, 1)) output by target device i.

These bits are used to adjust the sampling timing in the protocol mode that uses a data strobe signal to sample read data.

Do not rewrite these bits when automatic calibration is enabled, as values are updated automatically.

48.2.2.2 Common Configuration Register (COMCFG)

Address(es): XSPI0.COMCFG 8786 8004h, XSPI1.COMCFG 8786 8404h, XSPI0_NS.COMCFG 0786 8004h, XSPI1_NS.COMCFG 0786 8404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OENEGEX	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	OEASTEX	Output Enable Assertion Delay	0: Does not delay the assertion timing of the IO[7:0] and DS signals. 1: Delays the assertion timing of IO[7:0] and DS signals by one cycle.	R/W
b17	OENEGEX	Output Enable Negation Delay	0: Does not delay the negation timing of the IO[7:0] and DS signals. 1: Delays the negation timing of IO[7:0] and DS signals by one cycle.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the basic operation of the XSPI controller module.

OEASTEX Bit (Output Enable Assertion Delay)

This bit specifies the assertion timing of the output buffer control signal (OE: Output Enable). Setting this bit to 1 delays the start of output from the XSPIn_IO7 to XSPIn_IO0 pins and the XSPIn_DS pin by one cycle (n = 0, 1). In this case, the LIOCFGCSi.CSASTEX bit must also be set to 1 to extend the assertion period of the XSPIn_CSi# signal (i = 0, 1). Do not set this bit to 1 if there are no latency cycles, as this may cause output data to conflict with input data.

OENEGEX Bit (Output Enable Negation Delay)

This bit specifies the negation timing of the output buffer control signal (OE: Output Enable). Setting this bit to 1 delays the stop of output from the XSPIn_IO7 to XSPIn_IO0 pins and the XSPIn_DS pin by one cycle.
Do not set this bit to 1 if there are no latency cycles, as this may cause output data to conflict with input data.

48.2.2.3 Memory-Mapped Mode Configuration Register (BMCFGCH0)

Address(es): XSPI0.BMCFGCH0 8786 8008h, XSPI1.BMCFGCH0 8786 8408h,
XSPI0_NS.BMCFGCH0 0786 8008h, XSPI1_NS.BMCFGCH0 0786 8408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMBTIM[7:0]								—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MWRSIZE[7:0]								MWRCOMB	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMD	Internal Bus Write Cycle Response Mode Select*1	0: Responds after storing data to the write buffer 1: Responds after issuing a write transaction on the xSPI bus	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MWRCOMB	Write Data Combination Mode Enable	0: Write data combination mode is disabled. 1: Write data combination mode is enabled.	R/W
b15 to b8	MWRSIZE[7:0]	Write Data Size Select	00h: 4 bytes 01h: 8 bytes : : 0Eh: 60 bytes 0Fh: 64 bytes FFh: 2 bytes Settings other than above are prohibited	R/W
b16	PREEN	Prefetch Function Enable	0: Prefetch function is disabled. 1: Prefetch function is enabled.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	CMBTIM[7:0]	Combination Timer Timeout Period Setting	These bits specify the timeout period for write data combination mode	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.
Note 1. When setting the WRMD bit to 1, set the MWRCOMB bit to 0.

This register configures the operation of the memory-mapped mode within the basic operation of the XSPI controller module.

WRMD Bit (Internal Bus Write Cycle Response Mode Select)

This bit selects the response timing for the write cycle from the internal bus in memory-mapped mode. When this bit is set to 1, a response is returned after one frame is transmitted on the xSPI bus.

MWRCOMB Bit (Write Data Combination Mode Enable)

This bit is used to combine multiple write cycles from the internal bus into a single xSPI frame in memory-mapped mode.
When this bit is set to 0, one xSPI frame is generated for each write cycle on the internal bus.
When this bit is set to 1, an xSPI frame of the data size specified by the MWRSIZE[7:0] bits is generated. The xSPI transaction is not issued and is held internally within the XSPI until data of the specified size is available.

MWRSIZE[7:0] Bits (Write Data Size Select)

These bits specify the data size of the xSPI frame to be generated in memory-mapped mode. While write cycles to consecutive addresses are issued, xSPI frames of the specified data size are transmitted. If a read cycle or a write cycle to non-consecutive addresses is issued before the specified size is reached, an xSPI frame is generated using the data that

was reserved up to that point and transmitted on the xSPI bus.

PREEN Bit (Prefetch Function Enable)

This bit enables the prefetch function for read transactions in memory-mapped mode, which has the effect of reducing the latency of xSPI transactions when reading from consecutive addresses.

CMBTIM[7:0] Bits (Combination Timer Timeout Period Setting)

This bit specifies the number of ICLK cycles for which frame transmission is suspended internally for data integration in write data combination mode.
When the timer value reaches 00h, an xSPI frame is generated using the data accumulated up to that point and transmitted onto the xSPI bus. When 00h is set, no timeout occurs.

48.2.2.4 CSi Area Memory-Mapped Mode Configuration Register 0 (CMCFG0CSi) (i = 0, 1)

Address(es): XSPI0.CMCFG0CS0 8786 8010h, XSPI0.CMCFG0CS1 8786 8020h, XSPI1.CMCFG0CS0 8786 8410h, XSPI1.CMCFG0CS1 8786 8420h, XSPI0_NS.CMCFG0CS0 0786 8010h, XSPI0_NS.CMCFG0CS1 0786 8020h, XSPI1_NS.CMCFG0CS0 0786 8410h, XSPI1_NS.CMCFG0CS1 0786 8420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ARYAMD	WPBSTMD	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FFMT[1:0]	Frame Format Setting	b1 b0 0 0: Normal format 0 1: 8D-8D-8D profile 1.0 format 1 0: 8D-8D-8D profile 2.0 command modifier format 1 1: 8D-8D-8D profile 2.0 extended command modifier format	R/W
b3, b2	ADDSIZE[1:0]	Address Size Setting	b3 b2 0 0: 1 byte (256-Byte address space) 0 1: 2 bytes (64-Kbyte address space) 1 0: 3 bytes (16-Mbyte address space) 1 1: 4 bytes (4-Gbyte address space)	R/W
b4	WPBSTMD	WRAP Burst Mode Setting	0: Separate xSPI transfer at the wrap boundary 1: Not separate xSPI transfer at the wrap boundary	R/W
b5	ARYAMD	Array Address Mode Setting	0: Normal address mode 1: Array address mode	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	ADDRPEN[7:0]	Address Replace Enable	These bits are used to specify, on a bit-by-bit basis, whether to replace the value of each of the 8 higher-order bits of the address output to the xSPI bus with the value specified by the corresponding ADDRPCD[7:0] bits. 0: Do not replace the corresponding address bit. 1: Replace the corresponding address bit.	R/W
b31 to b24	ADDRPCD[7:0]	Address to be Replaced	Set the value with which the 8 higher-order bits of the address output to the xSPI bus are to be replaced where this is selected by the corresponding ADDRPEN[7:0] bits.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the operation of the memory-mapped mode within the basic operation of the XSPI controller module.

FFMT[1:0] Bits (Frame Format Setting)

These bits configure the xSPI frame format in memory-mapped mode. Table 48.4 shows the frame format for each setting value.

Table 48.4 FFMT[1:0] Bit Settings and xSPI Frame

FFMT[1:0]	Protocol Mode	xSPI Frame Format
00b	Except for the following	Command (1 byte) + Address (1 to 4 bytes) + Data
01b	8D-8D-8D Profile 1.0	Command (1 byte) + Command extension (1 byte) + Address (1 to 4 bytes) + Data
10b	8D-8D-8D Profile 2.0	Command (1 byte) + Command modifier (5 bytes) + Data
11b	8D-8D-8D Profile 2.0	Command (1 byte) + Command modifier (5 bytes) + Command extension (1 byte) + Data

ADDSIZE[1:0] Bits (Address Size Setting)

These bits specify the number of bytes in the address field in memory-mapped mode.
When setting the FFMT[1:0] bits to 10b or 11b (8D-8D-8D Profile 2.0), set this bit to 11b (4 bytes).

WPBSTMD Bit (WRAP Burst Mode Setting)

This bit configures the separation of xSPI transfers at the wrap boundary.
When this bit is set to 1, the wrap boundary for internal bus access aligns with the wrap boundary for xSPI memory.

ARYAMD Bit (Array Address Mode Setting)

This bit is used for Profile 1.0 commands.
When this bit is set to 1, the CPU address bits A[31:26] are ignored and the bits sent are A[25:18], A[17:10], A[9:4] + 00b, and 0000b + A[3:0] in that order.

ADDRPEN[7:0] Bits (Address Replace Enable)

These bits are used to select which of the 8 higher-order bits of the address output to the xSPI bus are to be replaced with the corresponding bit values of the ADDRPCD[7:0] bits in memory-mapped mode.

ADDRPCD[7:0] Bits (Address to be Replaced)

The values of these bits are set as the 8 higher-order bits of the address output to the xSPI bus in memory-mapped mode when replacement is selected by the corresponding ADDRPEN[7:0] bits.
The address bits to be replaced also depend on the setting of the ADDSIZE[1:0] bits. For example, if the ADDSIZE[1:0] bits are set to 11b (for 4-byte addresses), A[31:24] of the CPU address are to be replaced; if the ADDSIZE[1:0] bits are set to 10b (for 3-byte addresses), A[23:16] of the CPU address are to be replaced.
The value of the address bit corresponding to any bit in the ADDRPEN[7:0] bits that has the value 1 is replaced with that of the corresponding bit of the ADDRPCD[7:0] bits.
The XSPI area is allocated to the address range from 2000 0000h to 5FFF FFFFh. Accordingly, when using 4-byte addresses, you can access any address in the 4-Gbyte space by setting the ADDRPEN[7:0] bits to F0h and varying the value of the ADDRPCD[7:4] bits within the range from 0h to Fh.

48.2.2.5 CSi Area Memory-Mapped Mode Configuration Register 1 (CMCFG1CSi) (i = 0, 1)

Address(es): XSPI0.CMCFG1CS0 8786 8014h, XSPI0.CMCFG1CS1 8786 8024h, XSPI1.CMCFG1CS0 8786 8414h, XSPI1.CMCFG1CS1 8786 8424h, XSPI0_NS.CMCFG1CS0 0786 8014h, XSPI0_NS.CMCFG1CS1 0786 8024h, XSPI1_NS.CMCFG1CS0 0786 8414h, XSPI1_NS.CMCFG1CS1 0786 8424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	RDLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RDCMD[15:0]	Read Command Setting	This field configures the value of the command field for read transactions issued in memory-mapped mode.	R/W
b20 to b16	RDLATE[4:0]	Read Latency Cycle Setting	b20 b16 00000: No latency 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the read command in the memory-mapped mode within the basic operation of the XSPI controller module.

RDCMD[15:0] Bits (Read Command Setting)

These bits are used to set the command code specified in the command field for read transactions that are automatically generated in memory-mapped mode.

Set the appropriate value according to the selected frame format. For the normal format and 8D-8D-8D Profile 2.0 format, set the command code in the RDCMD[15:8] bits and set the RDCMD[7:0] bits to 00h. For the 8D-8D-8D Profile 1.0 format, set the command code in the RDCMD[15:8] bits and the command extension in the RDCMD[7:0] bits.

RDLATE[4:0] Bits (Read Latency Cycle Setting)

These bits specify the number of latency cycles to insert during read transactions in memory-mapped mode.

48.2.2.6 CSi Area Memory-Mapped Mode Configuration Register 2 (CMCFG2CSi) (i = 0, 1)

Address(es): XSPI0.CMCFG2CS0 8786 8018h, XSPI0.CMCFG2CS1 8786 8028h, XSPI1.CMCFG2CS0 8786 8418h, XSPI1.CMCFG2CS1 8786 8428h, XSPI0_NS.CMCFG2CS0 0786 8018h, XSPI0_NS.CMCFG2CS1 0786 8028h, XSPI1_NS.CMCFG2CS0 0786 8418h, XSPI1_NS.CMCFG2CS1 0786 8428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	WRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WRCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WRCMD[15:0]	Write Command Setting	This field configures the value of the command field for write transactions issued in memory-mapped mode.	R/W
b20 to b16	WRLATE[4:0]	Write Latency Cycle Setting	b20 b16 00000: No latency 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the write command in the memory-mapped mode within the basic operation of the XSPI controller module.

WRCMD[15:0] Bits (Write Command Setting)

These bits are used to set the command code specified in the command field for write transactions that are automatically generated in memory-mapped mode.
Set the appropriate value according to the selected frame format. For the normal format and 8D-8D-8D Profile 2.0 format, set the command code in the WRCMD[15:8] bits and set the WRCMD[7:0] bits to 00h. For the 8D-8D-8D Profile 1.0 format, set the command code in the WRCMD[15:8] bits and the command extension in the WRCMD[7:0] bits.

WRLATE[4:0] Bits (Write Latency Cycle Setting)

These bits specify the number of latency cycles to insert during write transactions in memory-mapped mode.

48.2.2.7 CSi Area I/O Signal Configuration Register (LIOCFGCSi) (i = 0, 1)

Address(es): XSPI0.LIOCFGCS0 8786 8050h, XSPI0.LIOCFGCS1 8786 8054h, XSPI1.LIOCFGCS0 8786 8450h, XSPI1.LIOCFGCS1 8786 8454h, XSPI0_NS.LIOCFGCS0 0786 8050h, XSPI0_NS.LIOCFGCS1 0786 8054h, XSPI1_NS.LIOCFGCS0 0786 8450h, XSPI1_NS.LIOCFGCS1 0786 8454h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DDRSMPEX[3:0]				SDRSMPSFT[3:0]				SDRSMPMD	SDRDRV	CSNEGEX	CSASTEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRMSKMD	LATEMD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PRTMD[9:0]	Protocol Mode Setting	000h: 1S-1S-1S 3B2h: 4S-4D-4D 3FFh: 8D-8D-8D 048h: 1S-2S-2S 049h: 2S-2S-2S 090h: 1S-4S-4S 092h: 4S-4S-4S Settings other than above are prohibited.	R/W
b10	LATEMD	Latency Mode Select	0: Configurable latency 1: Variable latency	R/W
b11	WRMSKMD	Write Data Mask Enable	0: Write data mask id disabled 1: Write data mask is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	CSMIN[3:0]	CS Minimum High Width Setting	b19 b16 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles : : 0 1 1 1: 8 cycles (initial value) : : 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W
b20	CSASTEX	CS Assertion Delay	0: No delay 1: 1 cycle delay	R/W
b21	CSNEGEX	CS Negation Delay	0: No delay 1: 1 cycle delay	R/W
b22	SDRDRV	SDR Data Output Timing Setting	0: Data is output on the falling edge of the XSPIDIVCLK. 1: Data is output on the rising edge of the XSPIDIVCLK.	R/W
b23	SDRSMPMD	SDR Data Sampling Timing Setting	0: Data is sampled on the falling edge of the CK or DS signal. 1: Data is sampled on the rising edge of the CK or DS signal.	R/W
b27 to b24	SDRSMPSFT[3:0]	SDR Data Sampling Timing Delay	b27 b24 0 0 0 0: No delay 0 0 0 1: Delay sampling timing by one cycle : : 0 1 1 0: Delay sampling timing by six cycles 0 1 1 1: Delay sampling timing by seven cycles Settings other than above are prohibited.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	DDRSMPEX[3:0]	DDR Data Sampling Period Extension	b31 b28 0 0 0 0: No extension 0 0 0 1: Expand by one cycle : : 0 1 1 0: Expand by six cycles 0 1 1 1: Expand by seven cycles Settings other than above are prohibited.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the timing of input and output signals within the basic operation of the XSPI controller module.

PRTMD[9:0] Bits (Protocol Mode Setting)

These bits specify the IO protocol mode.
In 4S-4D-4D protocol mode and 8D-8D-8D protocol mode, the DS signal is used for data sampling. In other protocol modes, the CK signal is used.

LATEMD Bit (Latency Mode Select)

This bit selects the behavior of the initial access latency phase. Setting this bit to 0 inserts the specified number of latency cycles after the address field is transmitted.
Setting this bit to 1 causes the latency cycle count to start from the middle of the address field. Furthermore, the inserted latency cycles are doubled when the XSPIn_DS signal is high during the command & command modifier phase (n = 0, 1). This bit is used only during the 8D-8D-8D Profile 2.0 format.

WRMSKMD Bit (Write Data Mask Enable)

This bit configures whether to use the XSPIn_DS signal as a mask signal for each byte in the write data.
The write data mask function is only used in 8D-8D-8D protocol mode.

CSMIN[3:0] Bits (CS Minimum High Width Setting)

These bits are used to set the minimum high width of the XSPIn_CSi# signal (the shortest number of cycles between xSPI frames) (n = 0, 1; i = 0, 1).

CSASTEX Bit (CS Assertion Delay)

This bit is used to delay the assertion timing of the XSPIn_CSi# signal by one cycle.

CSNEGEX Bit (CS Negation Delay)

This bit is used to delay the negation timing of the XSPIn_CSi# signal by one cycle.

SDRDRV Bit (SDR Data Output Timing Setting)

This bit is used to delay the data output timing by 0.5 cycles during SDR transfer.
To prevent potential collisions between output and input data, do not set this bit to 1 if there is no latency cycle.

SDRSMPMD Bit (SDR Data Sampling Timing Setting)

This bit is used to select the input data sampling edge during SDR transfer. When this bit is set to 1, sampling occurs half a cycle earlier than when it is set to 0.

SDRSMPSFT[3:0] Bits (SDR Data Sampling Timing Delay)

This bit is used to delay the sampling timing during SDR transfer in one cycle units.
Set this bit to 0 during DDR transfer.

DDRSMPEX[3:0] Bits (DDR Data Sampling Period Extension)

This bit is used to extend the data phase during DDR transfer.

During DDR transfer, data is sampled within the data phase following the latency cycle. XSPI determines the timing to end the data phase by estimating the number of read data bits. However, depending on the delay of the signal output by the target device, valid data may fall outside the data phase range and risk not being sampled. In such cases, use this bit to extend the data phase.

48.2.2.8 Memory Access Extension Configuration Register (MEMEXTCFG)

Address(es): XSPI0.MEMEXTCFG 8786 805Ch, XSPI1.MEMEXTCFG 8786 845Ch, XSPI0_NS.MEMEXTCFG 0786 805Ch, XSPI1_NS.MEMEXTCFG 0786 845Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DSVLDMDCS1	—	—	—	DSEXTMDCS1	—	—	—	DSVLDMDCS0	—	—	—	DSEXTMDCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSEXTMDCS0	DS Receive Wait Extension Mode 0	Extends the acceptance time for the DS signal driven by target device 0 0: DS receive wait extension is disabled 1: DS receive wait extension is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DSVLDMDCS0	DS Valid Cycle Setting 0	Delays the start position for accepting the DS signal driven by target device 0 0: DS is valid from the first cycle 1: DS is valid from the second cycle	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DSEXTMDCS1	DS Receive Wait Extension Mode 1	Extends the acceptance time for the DS signal driven by target device 1 0: DS receive wait extension is disabled 1: DS receive wait extension is enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	DSVLDMDCS1	DS Valid Cycle Setting 1	Delays the start position for accepting the DS signal driven by target device 1 0: DS is valid from the first cycle 1: DS is valid from the second cycle	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions.

This register configures the basic operation of the XSPI controller module.

DSEXTMDCSi Bit (DS Receive Wait Extension Mode i) (i = 0, 1)

This bit is used to configure whether the XSPI accepts DS signals after the scheduled receive cycle has elapsed in 8D-8D-8D Profile 2.0 mode.

When this bit is set to 1, the XSPI waits for DS input for each frame until the expected number of data is reached.

DSVLDMDCSi Bit (DS Valid Cycle Setting i) (i = 0, 1)

When in 8D-8D-8D Profile 1.0 mode, this bit configures the DS valid cycle.

When this bit is 0, the DS input is enabled starting from the first cycle.

When this bit is 1, the DS signal's first cycle (both rising and falling edges) is ignored, and it is used starting from the second cycle.

48.2.2.9 Memory-Mapped Mode Control Register 0 (BMCTL0)

Address(es): XSPI0.BMCTL0 8786 8060h, XSPI1.BMCTL0 8786 8460h, XSPI0_NS.BMCTL0 0786 8060h, XSPI1_NS.BMCTL0 0786 8460h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CH0CS1ACC[1:0]		CH0CS0ACC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CH0CS0ACC[1:0]	Target Device 0 Memory Access Enable	b1 b0 0 0: Read/Write disabled 0 1: Read enabled, Write disabled 1 0: Read disabled, Write enabled 1 1: Read/Write enabled	R/W
b3, b2	CH0CS1ACC[1:0]	Target Device 1 Memory Access Enable	b3 b2 0 0: Read/Write disabled 0 1: Read enabled, Write disabled 1 0: Read disabled, Write enabled 1 1: Read/Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI transaction, stop all write/read transactions before rewriting it.

This register configures the operation of the memory-mapped mode within the basic operation of the XSPI controller module.

CH0CSiACC[1:0] Bits (Target Device i Memory Access Enable) (i = 0, 1)

These bits are used to configure protection when accessing the CSi area using memory-mapped mode. Protection should be disabled at the end of the initialization process.

48.2.2.10 Memory-Mapped Mode Control Register 1 (BMCTL1)

Address(es): XSPI0.BMCTL1 8786 8064h, XSPI1.BMCTL1 8786 8464h, XSPI0_NS.BMCTL1 0786 8064h, XSPI1_NS.BMCTL1 0786 8464h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PBUFCLRCH0	—	MWRPUSHCH0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	The write value should be 0.	W
b8	MWRPUSHCH0	Write Data Push*1	Writing 1 to this bit will generate an xSPI frame with the previously pending data.	W
b9	—	Reserved	The write value should be 0.	W
b10	PBUFCLRCH0	Prefetch Buffer Clear*1	Writing 1 to this bit clears the prefetch buffer.	W
b31 to b11	—	Reserved	The write value should be 0.	W

Note 1. If an access from the internal bus occurs at the same time that this bit is written to 1, it is undefined whether a prefetch or write data combination will be performed.

This register controls the operation of the XSPI controller module in memory-mapped mode.

MWRPUSHCH0 Bit (Write Data Push)

In write data combination mode, if there is data reserved without generating an xSPI frame, writing 1 to this bit will force a transaction to be issued.

PBUFCLRCH0 Bit (Prefetch Buffer Clear)

This bit is used to request clearing of the prefetch buffer when the prefetch function is enabled.

48.2.2.11 Memory-Mapped Mode Command Control Register (CMCTLCH0)

Address(es): XSPI0.CMCTLCH0 8786 8068h, XSPI1.CMCTLCH0 8786 8468h, XSPI0_NS.CMCTLCH0 0786 8068h, XSPI1_NS.CMCTLCH0 0786 8468h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XIPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	XIPEXCODE[7:0]								XIPENCODE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	XIPENCODE[7:0]	XIP Mode Enter Code Setting	Set the code to enter XIP mode in these bits.	R/W
b15 to b8	XIPEXCODE[7:0]	XIP Mode Exit Code Setting	Set the code to exit XIP mode in these bits.	R/W
b16	XIPEN	XIP Mode Enable*1	0: Disable XIP mode 1: Enable XIP mode	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not set this bit to 1 when the 8D-8D-8D Profile 2.0 format is selected.

This register controls the operation of the XSPI controller module in memory-mapped mode.

XIPEN Bit (XIP Mode Enable)

This bit is used to enable XIP mode.

When this bit is set to 1, the XIP enter code specified in the XIPENCODE[7:0] bits is inserted into the latency field of the read transaction. The command field is omitted from subsequent read transactions.

When writing 0 to this bit, the XIP exit code specified in the XIPEXCODE[7:0] bits is inserted into the latency field of the read transaction, and then this bit is set to 0. While this bit is set to 1, XIP mode has not ended.

48.2.2.12 Indirect Access Mode Control Register 0 (CDCTL0)

Address(es): XSPI0.CDCTL0 8786 8070h, XSPI1.CDCTL0 8786 8470h, XSPI0_NS.CDCTL0 0786 8070h, XSPI1_NS.CDCTL0 0786 8470h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	PERREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]		CSSEL	—	PERMD	TRREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRREQ	Transaction Request	0: Transaction aborted or transaction completed 1: Request transaction issuance	R/W
b1	PERMD	Periodic Mode Setting	0: Sequential mode 1: Periodic mode	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CSSEL	Chip Select	0: Target device 0 (CS0) 1: Target device 1 (CS1)	R/W
b5, b4	TRNUM[1:0]	Transaction Number Setting	b5 b4 0 0: One (using command buffer 0) 0 1: Two (using command buffers 0 and 1) 1 0: Three (using command buffers 0 to 2) 1 1: Four (using command buffers 0 to 3)	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	PERITV[4:0]	Transaction Interval Setting	b20 b16 00000: 2^1 cycles (2 cycles) 00001: 2^2 cycles (4 cycles) : : 11110: 2^{31} cycles (2,147,483,648 cycles) 11111: 2^{32} cycles (4,294,967,296 cycles)	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	PERREP[3:0]	Transaction Repetition Count Setting	b27 b24 0 0 0 0: 2^0 time (1 time) 0 0 0 1: 2^1 times (2 times) : : 1 1 1 0: 2^{14} times (16,384 times) 1 1 1 1: 2^{15} times (32,768 times)	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the operation of the XSPI controller module in indirect access mode.

TRREQ Bit (Transaction Request)

This bit controls the issuance of transactions.

When this bit is set to 1, the transactions specified in the command buffers are issued. When all transactions have been issued, this bit is automatically set to 0.

Writing 0 to this bit while a transaction is being issued aborts the transaction.

PERMD Bit (Periodic Mode Setting)

This bit is used to enable periodic mode.

When this bit is set to 1, the read transaction specified in command buffer 0 is repeated periodically and the read value is compared with the expected value. This can be used for status polling of external memory.

CSSEL Bit (Chip Select)

This bit selects the target device to issue the transaction.

TRNUM[1:0] Bits (Transaction Number Setting)

This bit is used to set the number of transactions to issue in sequential mode.
In periodic mode, this bit should normally be set to 00b. If a value other than 00b is set, the read data of the last transaction is compared with the expected value.

PERITV[4:0] Bits (Transaction Interval Setting)

These bits are used to set the transaction issuance interval in periodic mode.
If this interval is too short compared to the CPU bus cycle, the read value may not be stored in the data buffer. Ensure the interval is at least four times longer than the CPU bus cycle.

PERREP[3:0] Bits (Transaction Repetition Count Setting)

These bits are used to set the number of times to repeat a specified transaction in periodic mode.
If the read value does not match the expected value even after the number of times set in these bits have been reached, it is considered a timeout.

48.2.2.13 Indirect Access Mode Control Register 1 (CDCTL1)

Address(es): XSPI0.CDCTL1 8786 8074h, XSPI1.CDCTL1 8786 8474h, XSPI0_NS.CDCTL1 0786 8074h, XSPI1_NS.CDCTL1 0786 8474h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the expected value for comparison with the read value in the periodic mode of indirect access mode. Set the expected value right-justified. For example, when performing a 1-byte comparison, set the value in bits b7 to b0.

48.2.2.14 Indirect Access Mode Control register 2 (CDCTL2)

Address(es): XSPI0.CDCTL2 8786 8078h, XSPI1.CDCTL2 8786 8478h, XSPI0_NS.CDCTL2 0786 8078h, XSPI1_NS.CDCTL2 0786 8478h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the mask value for comparing the read value with the expected value in the periodic mode of indirect access mode. Bits set to 1 in this register will not be compared.
For example, this register is used to compare only one byte when the protocol mode that cannot read in units of one byte is selected.

48.2.2.15 Command Buffer m Transaction Type Register (CDTBUFm) (m = 0 to 3)

Address(es): XSPI0.CDTBUF0 8786 8080h, XSPI0.CDTBUF1 8786 8090h, XSPI0.CDTBUF2 8786 80A0h, XSPI0.CDTBUF3 8786 80B0h, XSPI1.CDTBUF0 8786 8480h, XSPI1.CDTBUF1 8786 8490h, XSPI1.CDTBUF2 8786 84A0h, XSPI1.CDTBUF3 8786 84B0h, XSPI0_NS.CDTBUF0 0786 8080h, XSPI0_NS.CDTBUF1 0786 8090h, XSPI0_NS.CDTBUF2 0786 80A0h, XSPI0_NS.CDTBUF3 0786 80B0h, XSPI1_NS.CDTBUF0 0786 8480h, XSPI1_NS.CDTBUF1 0786 8490h, XSPI1_NS.CDTBUF2 0786 84A0h, XSPI1_NS.CDTBUF3 0786 84B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRTYPE	—	LATE[4:0]					DATASIZE[3:0]				ADDSIZE[2:0]			CMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMDSIZE[1:0]	Command Size Setting	b1 b0 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W
b4 to b2	ADDSIZE[2:0]	Address Size Setting	b4 b2 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Settings other than above are prohibited.	R/W
b8 to b5	DATASIZE[3:0]	Data Size Setting	b8 b5 0 0 0 0: 0 bytes (No data phase) 0 0 0 1: 1 byte : : 0 1 1 1: 7 bytes 1 0 0 0: 8 bytes Settings other than above are prohibited.	R/W
b13 to b9	LATE[4:0]	Latency Cycle Setting	b13 b9 00000: No latency 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	TRTYPE	Transaction Type Select	0: Read transaction 1: Other than read transaction	R/W
b31 to b16	CMD[15:0]	Command Field Setting	Set the command field (1 to 2 bytes)	R/W

This register configures the transaction to be issued in indirect access mode. This register specifies the command type and the size of each field, etc.

CMDSIZE[1:0] Bits (Command Size Setting)

These bits are used to set the size of the command field.

If 8D-8D-8D protocol mode is selected, these bits must be set to 10b.

Setting both the command size and address size to 0 bytes is prohibited.

ADDSIZE[2:0] Bits (Address Size Setting)

These bits are used to set the size of the address field.

DATASIZE[3:0] Bits (Data Size Setting)

These bits are used to set the size of the data field.
When 8D-8D-8D protocol mode is selected, data bytes are transferred only in byte pairs (16-bit units). For example, even if an odd-numbered byte is set to read, an even-numbered byte of data is received. In this case, the last byte is ignored.
Do not set 0 bytes for a read transaction.

LATE[4:0] Bits (Latency Cycle Setting)

These bits are used to set the latency cycle.

TRTYPE Bit (Transaction Type Select)

These bits are used to set the type of the transaction.

CMD[15:0] Bits (Command Field Setting)

These bits are used to set the command field as one or two bytes.
The command with the number of bytes specified by CMDSIZE[1:0] bits is transferred.

Table 48.5 CMD[15:0] Bit Setting Value for Each Frame Format

Frame Format	CMD[15:8] Bits	CMD[7:0] Bits
1S-1S-1S	Command code	Not used
4S-4D-4D	Command code	Not used
8D-8D-8D Profile 1.0 format	Command code	Command extension
8D-8D-8D Profile 2.0 command modifier format	Command code	First byte of the command modifier
8D-8D-8D Profile 2.0 extended command modifier format	Command code + command extension	First byte of the command modifier

48.2.2.16 Command Buffer m Address Register (CDABUFm) (m = 0 to 3)

Address(es): XSPI0.CDABUF0 8786 8084h, XSPI0.CDABUF1 8786 8094h, XSPI0.CDABUF2 8786 80A4h, XSPI0.CDABUF3 8786 80B4h, XSPI1.CDABUF0 8786 8484h, XSPI1.CDABUF1 8786 8494h, XSPI1.CDABUF2 8786 84A4h, XSPI1.CDABUF3 8786 84B4h, XSPI0_NS.CDABUF0 0786 8084h, XSPI0_NS.CDABUF1 0786 8094h, XSPI0_NS.CDABUF2 0786 80A4h, XSPI0_NS.CDABUF3 0786 80B4h, XSPI1_NS.CDABUF0 0786 8484h, XSPI1_NS.CDABUF1 0786 8494h, XSPI1_NS.CDABUF2 0786 84A4h, XSPI1_NS.CDABUF3 0786 84B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the transaction to be issued in indirect access mode. Set the address field in this register. The address of the number of bytes specified by the CDTBUFm.ADDSIZE[2:0] bits is transferred in order from the most significant byte. If the address size is less than 4 bytes, set the lower byte to 00h.

Table 48.6 Address Field Setting Value for Each Frame Format

Frame Format	Address Field
1S-1S-1S	Address
4S-4D-4D	Address
8D-8D-8D Profile 1.0 format	Address
8D-8D-8D Profile 2.0 command modifier format	Second to fifth bytes of the command modifier
8D-8D-8D Profile 2.0 extended command modifier format	Second to fifth bytes of the command modifier

48.2.2.17 Command Buffer m Data Register 0 (CDD0BUFm) (m = 0 to 3)

Address(es): XSPI0.CDD0BUF0 8786 8088h, XSPI0.CDD0BUF1 8786 8098h, XSPI0.CDD0BUF2 8786 80A8h, XSPI0.CDD0BUF3 8786 80B8h, XSPI1.CDD0BUF0 8786 8488h, XSPI1.CDD0BUF1 8786 8498h, XSPI1.CDD0BUF2 8786 84A8h, XSPI1.CDD0BUF3 8786 84B8h, XSPI0_NS.CDD0BUF0 0786 8088h, XSPI0_NS.CDD0BUF1 0786 8098h, XSPI0_NS.CDD0BUF2 0786 80A8h, XSPI0_NS.CDD0BUF3 0786 80B8h, XSPI1_NS.CDD0BUF0 0786 8488h, XSPI1_NS.CDD0BUF1 0786 8498h, XSPI1_NS.CDD0BUF2 0786 84A8h, XSPI1_NS.CDD0BUF3 0786 84B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the transaction to be issued in indirect access mode.
For a write transaction, set the first 4 bytes of the data field in this register. The data of the number of bytes specified by the CDTBUFm.DATASIZE[3:0] bits is transferred in order from the most significant byte. If the data size is less than 4 bytes, set the lower byte to 00h.
For a read transaction, the first 4 bytes of the read data are stored in this register.

48.2.2.18 Command Buffer m Data Register 1 (CDD1BUFm) (m = 0 to 3)

Address(es): XSPI0.CDD1BUF0 8786 808Ch, XSPI0.CDD1BUF1 8786 809Ch, XSPI0.CDD1BUF2 8786 80ACh, XSPI0.CDD1BUF3 8786 80BCh, XSPI1.CDD1BUF0 8786 848Ch, XSPI1.CDD1BUF1 8786 849Ch, XSPI1.CDD1BUF2 8786 84ACh, XSPI1.CDD1BUF3 8786 84BCh, XSPI0_NS.CDD1BUF0 0786 808Ch, XSPI0_NS.CDD1BUF1 0786 809Ch, XSPI0_NS.CDD1BUF2 0786 80ACh, XSPI0_NS.CDD1BUF3 0786 80BCh, XSPI1_NS.CDD1BUF0 0786 848Ch, XSPI1_NS.CDD1BUF1 0786 849Ch, XSPI1_NS.CDD1BUF2 0786 84ACh, XSPI1_NS.CDD1BUF3 0786 84BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to configure the transaction to be issued in indirect access mode.
For a write transaction, set the last 4 bytes of the data field in this register. The data of the number of bytes specified by the CDTBUFm.DATASIZE[3:0] bits is transferred in order from the most significant byte. If the data size is less than 8 bytes, set the lower byte to 00h.
For a read transaction, the last 4 bytes of the read data are stored in this register.

48.2.2.19 Signal Pattern Control Register 0 (LPCTL0)

Address(es): XSPI0.LPCTL0 8786 8100h, XSPI1.LPCTL0 8786 8500h, XSPI0_NS.LPCTL0 0786 8100h, XSPI1_NS.LPCTL0 0786 8500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	XD2VAL	—	—	XD2LEN[4:0]					XD1VAL	—	—	XD1LEN[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	XDPIN[1:0]		CSSEL	—	—	PATREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PATREQ	XIP Disable Pattern Generation Request	0: Do not generate XIP disable pattern/Generation complete 1: Request generation of XIP disable pattern	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	CSSEL	Chip Select	0: Target device 0 (CS0) 1: Target device 1 (CS1)	R/W
b5, b4	XDPIN[1:0]	XIP Disable Pattern Output Pin Select	b5 b4 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	XD1LEN[4:0]	XIP Disable Pattern 1st Phase Length Setting*1	b20 b16 00000: 0 cycles 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b22, b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	XD1VAL	XIP Disable Pattern 1st Phase Output Level Select	0: Low 1: High	R/W
b28 to b24	XD2LEN[4:0]	XIP Disable Pattern 2nd Phase Length Setting*1	b28 b24 00000: 0 cycles 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b30 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	XD2VAL	XIP Disable Pattern 2nd Phase Output Level Select	0: Low 1: High	R/W

Note 1. Setting both of these bits to 00000b is prohibited.

This register controls the generation of the XIP disable pattern by the XSPI controller module.

PATREQ Bit (XIP Disable Pattern Generation Request)

This bit is used to request generation of the XIP disable pattern.

Writing 1 to this bit initiates generation of the XIP disable pattern. It automatically returns to 0 when pattern generation is complete.

CSSEL Bit (Chip Select)

This bit selects the target device for which the XIP disable pattern is generated.

XDPIN[1:0] Bits (XIP Disable Pattern Output Pin Select)

These bits select the number of data pins to be used to generate the XIP disable pattern.

XD1LEN[4:0] Bits (XIP Disable Pattern 1st Phase Length Setting)

These bits set the length of the first phase of the XIP disable pattern in number of XSPIDIVCLK cycles.

XD1VAL Bit (XIP Disable Pattern 1st Phase Output Level Select)

These bits select the signal level output during the first phase of the XIP disable pattern.

XD2LEN[4:0] Bits (XIP Disable Pattern 2nd Phase Length Setting)

These bits set the length of the second phase of the XIP disable pattern in number of XSPIDIVCLK cycles.

XD2VAL Bit (XIP Disable Pattern 2nd Phase Output Level Select)

These bits select the signal level output during the second phase of the XIP disable pattern.

48.2.2.20 Signal Pattern Control Register 1 (LPCTL1)

Address(es): XSPI0.LPCTL1 8786 8104h, XSPI1.LPCTL1 8786 8504h, XSPI0_NS.LPCTL1 0786 8104h, XSPI1_NS.LPCTL1 0786 8504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSEL	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1 to b0	PATREQ[1:0]	Signal Pattern Generation Request	b1 b0 0 0: Do not generate signal pattern/Generation complete 0 1: Request generation of reset pattern 1 0: Request generation of CS only pattern 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CSSEL	Chip Select	0: Target device 0 (CS0) 1: Target device 1 (CS1)	R/W
b5 to b4	RSTREP[1:0]	Reset Pattern Repetition Count Select	b5 b4 0 0: 4 times (compliant with the Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
b7 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	RSTWID[2:0]	Reset Pattern Width Select	b10 b8 0 0 0: 2^1 cycles (2 cycles) 0 0 1: 2^2 cycles (4 cycles) : : 1 1 0: 2^7 cycles (128 cycles) 1 1 1: 2^8 cycles (256 cycles)	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	RSTSU[2:0]	Reset Pattern Data Output Hold Time Setting	b14 b12 0 0 0: 1 cycle 0 0 1: 2 cycles : : 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the generation of special signal patterns by the XSPI controller module.

PATREQ[1:0] Bits (Signal Pattern Generation Request)

These bits are used to request generation of a signal pattern.

Writing 01b or 10b to these bits initiates generation of the corresponding signal pattern. It automatically returns to 00b when pattern generation is complete.

CSSEL Bit (Chip Select)

This bit selects the target device for which the signal pattern is generated.

RSTREP[1:0] Bits (Reset Pattern Repetition Count Select)

These bits select how many times the CS# signal toggles. The number specified in JESD252 is four times.

RSTWID[2:0] Bits (Reset Pattern Width Select)

These bits select the high width (tCSH) and low width (tCSL) of the CS# signal in the reset pattern or the low width (tCSL) of the CS# signal in the CS only pattern, in number of XSPIDIVCLK cycles.
Set these widths to at least 500 ns.

RSTSU[2:0] Bits (Reset Pattern Data Output Hold Time Setting)

These bits select the hold time of the IO[0] signal relative to the rising edge of the CS# signal in the reset pattern, in number of XSPIDIVCLK cycles.
Set these bits to ensure a hold time of at least 5 ns, including output buffer delay*1. Also, set these bits so that the hold time does not exceed the high width of the CS# signal specified by the RSTWID[2:0] bits.

Note 1. Set these bits so that the hold time also meets the specifications required by the target device.

48.2.2.21 I/O Signal Control Register (LIOCTL)

Address(es): XSPI0.LIOCTL 8786 8108h, XSPI1.LIOCTL 8786 8508h, XSPI0_NS.LIOCTL 0786 8108h, XSPI1_NS.LIOCTL 0786 8508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTCS1	RSTCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS1	WPCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	WPCS0	WP0# Signal Output Control	0: Output low 1: Output high	R/W
b1	WPCS1	WP1# Signal Output Control	0: Output low 1: Output high	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	RSTCS0	RESET0# Signal Output Control	0: Output low 1: Output high	R/W
b17	RSTCS1	RESET1# Signal Output Control	0: Output low 1: Output high	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the output signal of the XSPI controller module.

WPCSi Bit (WPi# Signal Output Control) (i = 0, 1)

This bit controls the signal level output from the XSPIn_WPi# pin (n = 0, 1).

This bit setting is only effective when a target device with a write protect pin is connected.

RSTCSi Bit (RESETi# Signal Output Control) (i = 0, 1)

This bit controls the signal level output from the XSPIn_RESETi# pin.

This bit setting is only effective when a target device with a reset pin is connected.

48.2.2.22 CSi Area Calibration Control Register 0 (CCCTL0CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL0CS0 8786 8130h, XSPI0.CCCTL0CS1 8786 8150h, XSPI1.CCCTL0CS0 8786 8530h, XSPI1.CCCTL0CS1 8786 8550h, XSPI0_NS.CCCTL0CS0 0786 8130h, XSPI0_NS.CCCTL0CS1 0786 8150h, XSPI1_NS.CCCTL0CS0 0786 8530h, XSPI1_NS.CCCTL0CS1 0786 8550h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CASFTEND[4:0]					—	—	—	CASFTSTA[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CAITV[4:0]					—	—	—	—	—	—	CANOWR	CAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CAEN	Automatic Calibration Enable	0: Automatic calibration is disabled. 1: Automatic calibration is enabled.	R/W
b1	CANOWR	Read-Only Calibration Mode	0: Write-read calibration 1: Read-only calibration	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	CAITV[4:0]	Calibration Interval Select	b12 b8 00000: 2^1 cycle (2 cycles) 00001: 2^2 cycle (4 cycles) : : 11110: 2^{31} cycles (2,147,483,648 cycles) 11111: 2^{32} cycles (4,294,967,296 cycles)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	CASFTSTA[4:0]	Initial DS Delay Setting	These bits configure the initial value for the DS signal delay.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CASFTEND[4:0]	Maximum DS Delay Setting	These bits configure the maximum value for the DS signal delay.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to automatically adjust the delay time of the DS signal for the XSPI controller module.

CAEN Bit (Automatic Calibration Enable)

This bit enables the automatic calibration.

Setting this bit to 1 periodically transmits a calibration sequence to adjust the delay time of the DS signal using delay elements. If this bit is written to 0 during execution of the calibration sequence, it is set to 0 after the currently executing calibration sequence completes.

CANOWR Bit (Read-Only Calibration Mode)

This bit selects the calibration method. Set this bit to 1 when calibration can be performed without writing to the target device.

When this bit is set to 0, calibration is performed to ensure values written to the target device can be read correctly. When this bit is set to 1, calibration is performed to ensure the calibration pattern originally written to the target device can be read correctly.

This bit can be used when the target device has a calibration pattern.

CAITV[4:0] Bits (Calibration Interval Select)

These bits select the calibration interval.

CASFTSTA[4:0] Bits (Initial DS Delay Setting)

These bit specify the initial value for calibration.
XSPI begins calibration using the DS signal delay time (number of delay cells) set in these bits as the initial delay time.

CASFTEND[4:0] Bits (Maximum DS Delay Setting)

These bit specify the maximum value for calibration.
When the delay time of the DS signal reaches the delay time (number of delay cells) set in these bits, the calibration sequence for that cycle ends.

48.2.2.23 CSi Area Calibration Control Register 1 (CCCTL1CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL1CS0 8786 8134h, XSPI0.CCCTL1CS1 8786 8154h, XSPI1.CCCTL1CS0 8786 8534h, XSPI1.CCCTL1CS1 8786 8554h, XSPI0_NS.CCCTL1CS0 0786 8134h, XSPI0_NS.CCCTL1CS1 0786 8154h, XSPI1_NS.CCCTL1CS0 0786 8534h, XSPI1_NS.CCCTL1CS1 0786 8554h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CARDLATE[4:0]					—	—	—	CAWRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CADATASIZE[3:0]				CAADDSIZE[2:0]			CACMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CACMDSIZE[1:0]	Command Size Setting	b1 b0 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W
b4 to b2	CAADDSIZE[2:0]	Address Size Setting	b4 b2 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Settings other than above are prohibited.	R/W
b8 to b5	CADATASIZE[3:0]	Data Size Setting	b8 b5 0 0 0 0: 1 byte 0 0 0 1: 2 bytes : : 1 1 1 0: 15 bytes 1 1 1 1: 16 bytes	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	CAWRLATE[4:0]	Write Latency Cycle Setting	b20 b16 00000: No latency 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CARDLATE[4:0]	Read Latency Cycle Setting	b28 b24 00000: No latency 00001: 1 cycle : : 11110: 30 cycles 11111: 31 cycles	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the frame format for automatic calibration in the XSPI controller module.

CACMDSIZE[1:0] Bits (Command Size Setting)

These bits configures the size of the command field.

When selecting 8D-8D-8D protocol mode, set these bit to 10b.

Setting both the command size and address size to 0 bytes is prohibited.

CAADDSIZE[2:0] Bits (Address Size Setting)

These bits configures the size of the address field.

CADATASIZE[3:0] Bits (Data Size Setting)

These bits configures the size of the data field.

When selecting 8D-8D-8D protocol mode, ensure the byte count is even.

CAWRLATE[4:0] Bits (Write Latency Cycle Setting)

These bits configures the latency cycle during write operation.

CARDLATE[4:0] Bits (Read Latency Cycle Setting)

These bits configures the latency cycle during read operation.

48.2.2.24 CSi Area Calibration Control Register 2 (CCCTL2CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL2CS0 8786 8138h, XSPI0.CCCTL2CS1 8786 8158h, XSPI1.CCCTL2CS0 8786 8538h, XSPI1.CCCTL2CS1 8786 8558h, XSPI0_NS.CCCTL2CS0 0786 8138h, XSPI0_NS.CCCTL2CS1 0786 8158h, XSPI1_NS.CCCTL2CS0 0786 8538h, XSPI1_NS.CCCTL2CS1 0786 8558h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CARDCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CAWRCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CAWRCMD[15:0]	Write Command Setting	Sets the value of the command field in the write transaction issued during automatic calibration.	R/W
b31 to b16	CARDCMD[15:0]	Read Command Setting	Sets the value of the command field in the read transaction issued during automatic calibration.	R/W

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the command code used for automatic calibration in the XSPI controller module.

48.2.2.25 CSi Area Calibration Control Register 3 (CCCTL3CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL3CS0 8786 813Ch, XSPI0.CCCTL3CS1 8786 815Ch, XSPI1.CCCTL3CS0 8786 853Ch, XSPI1.CCCTL3CS1 8786 855Ch, XSPI0_NS.CCCTL3CS0 0786 813Ch, XSPI0_NS.CCCTL3CS1 0786 815Ch, XSPI1_NS.CCCTL3CS0 0786 853Ch, XSPI1_NS.CCCTL3CS1 0786 855Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the address used for automatic calibration in the XSPI controller module.

48.2.2.26 CSi Area Calibration Control Register 4 (CCCTL4CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL4CS0 8786 8140h, XSPI0.CCCTL4CS1 8786 8160h, XSPI1.CCCTL4CS0 8786 8540h, XSPI1.CCCTL4CS1 8786 8560h, XSPI0_NS.CCCTL4CS0 0786 8140h, XSPI0_NS.CCCTL4CS1 0786 8160h, XSPI1_NS.CCCTL4CS0 0786 8540h, XSPI1_NS.CCCTL4CS1 0786 8560h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the data used for automatic calibration in the XSPI controller module.
Set data bytes 0 through 3 to this register.

48.2.2.27 CSi Area Calibration Control Register 5 (CCCTL5CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL5CS0 8786 8144h, XSPI0.CCCTL5CS1 8786 8164h, XSPI1.CCCTL5CS0 8786 8544h, XSPI1.CCCTL5CS1 8786 8564h, XSPI0_NS.CCCTL5CS0 0786 8144h, XSPI0_NS.CCCTL5CS1 0786 8164h, XSPI1_NS.CCCTL5CS0 0786 8544h, XSPI1_NS.CCCTL5CS1 0786 8564h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the data used for automatic calibration in the XSPI controller module.
Set data bytes 4 through 7 to this register.

48.2.2.28 CSi Area Calibration Control Register 6 (CCCTL6CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL6CS0 8786 8148h, XSPI0.CCCTL6CS1 8786 8168h, XSPI1.CCCTL6CS0 8786 8548h, XSPI1.CCCTL6CS1 8786 8568h, XSPI0_NS.CCCTL6CS0 0786 8148h, XSPI0_NS.CCCTL6CS1 0786 8168h, XSPI1_NS.CCCTL6CS0 0786 8548h, XSPI1_NS.CCCTL6CS1 0786 8568h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the data used for automatic calibration in the XSPI controller module.
Set data bytes 8 through 11 to this register.

48.2.2.29 CSi Area Calibration Control Register 7 (CCCTL7CSi) (i = 0, 1)

Address(es): XSPI0.CCCTL7CS0 8786 814Ch, XSPI0.CCCTL7CS1 8786 816Ch, XSPI1.CCCTL7CS0 8786 854Ch, XSPI1.CCCTL7CS1 8786 856Ch, XSPI0_NS.CCCTL7CS0 0786 814Ch, XSPI0_NS.CCCTL7CS1 0786 816Ch, XSPI1_NS.CCCTL7CS0 0786 854Ch, XSPI1_NS.CCCTL7CS1 0786 856Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register should be set when the CCCTL0CSi.CAEN bit is 0 (automatic calibration is disabled).

This register is used to configure the data used for automatic calibration in the XSPI controller module.
Set data bytes 12 through 15 to this register.

48.2.2.30 Common Status Register (COMSTT)

Address(es): XSPI0.COMSTT 8786 8184h, XSPI1.COMSTT 8786 8584h, XSPI0_NS.COMSTT 0786 8184h, XSPI1_NS.COMSTT 0786 8584h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	RSTOCS1	INTCS1	ECSCS1	—	RSTOCS0	INTCS0	ECSCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	WRBUFNECH0	—	PBUFNECH0	—	—	—	MEMACCCH0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	MEMACCCH0	Memory Access Status Flag	0: XSPI controller is not accessing memory. 1: XSPI controller is accessing memory.	R
b1	—	Reserved	This bit is undefined when read.	R
b3, b2	—	Reserved	These bits are read as 0.	R
b4	PBUFNECH0	Prefetch Buffer Not Empty Flag	0: Prefetch buffer is empty. 1: Data is present in the prefetch buffer.	R
b5	—	Reserved	This bit is read as 0.	R
b6	WRBUFNECH0	Write Buffer Not Empty Flag	0: Write buffer is empty. 1: Data is present in the write buffer.	R
b15 to b7	—	Reserved	These bits are read as 0.	R
b16	ECSCS0	ECS0# Signal Monitor Flag	0: Low 1: High	R
b17	INTCS0	INT0# Signal Monitor Flag	0: Low 1: High	R
b18	RSTOCS0	RSTO0# Signal Monitor Flag	0: Low 1: High	R
b19	—	Reserved	This bit is read as 0.	R
b20	ECSCS1	ECS1# Signal Monitor Flag	0: Low 1: High	R
b21	INTCS1	INT1# Signal Monitor Flag	0: Low 1: High	R
b22	RSTOCS1	RSTO1# Signal Monitor Flag	0: Low 1: High	R
b31 to b23	—	Reserved	These bits are read as 0.	R

This register indicates the status of the XSPI controller module.

MEMACCCH0 Flag (Memory Access Status Flag)

This flag indicates the memory access state.

This flag is only valid in the (2) Communication Stop Flow and (6) Stop Flow for Memory-Mapped Mode described in section 48.3.1.7, Operation Flows.

PBUFNECH0 Flag (Prefetch Buffer Not Empty Flag)

This flag indicates that data is present in the prefetch buffer.

WRBUFNECH0 Flag (Write Buffer Not Empty Flag)

This flag indicates that data remains in the write buffer.
This flag is only valid in the (2) Communication Stop Flow and (6) Stop Flow for Memory-Mapped Mode described in section 48.3.1.7, Operation Flows.

ECSCSi Flag (ECSi# Signal Monitor Flag) (i = 0, 1)

This flag indicates the input level of the XSPIn_ECSi# pin (n = 0, 1).

INTCSi Flag (INTi# Signal Monitor Flag) (i = 0, 1)

This flag indicates the input level of the XSPIn_INTi# pin.

RSTOCSi Flag (RSTOi# Signal Monitor Flag) (i = 0, 1)

This flag indicates the input level of the XSPIn_RSTOi# pin.

48.2.2.31 CSi Area Calibration Status Register (CASTTCSi) (i = 0, 1)

Address(es): XSPI0.CASTTCS0 8786 8188h, XSPI0.CASTTCS1 8786 818Ch, XSPI1.CASTTCS0 8786 8588h, XSPI1.CASTTCS1 8786 858Ch, XSPI0_NS.CASTTCS0 0786 8188h, XSPI0_NS.CASTTCS1 0786 818Ch, XSPI1_NS.CASTTCS0 0786 8588h, XSPI1_NS.CASTTCS1 0786 858Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CASUC31	CASUC30	CASUC29	CASUC28	CASUC27	CASUC26	CASUC25	CASUC24	CASUC23	CASUC22	CASUC21	CASUC20	CASUC19	CASUC18	CASUC17	CASUC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CASUC15	CASUC14	CASUC13	CASUC12	CASUC11	CASUC10	CASUC9	CASUC8	CASUC7	CASUC6	CASUC5	CASUC4	CASUC3	CASUC2	CASUC1	CASUC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CASUC0	Read Comparison Result Flag 0	0: No match 1: Match	R
b1	CASUC1	Read Comparison Result Flag 1		R
b2	CASUC2	Read Comparison Result Flag 2		R
b3	CASUC3	Read Comparison Result Flag 3		R
b4	CASUC4	Read Comparison Result Flag 4		R
b5	CASUC5	Read Comparison Result Flag 5		R
b6	CASUC6	Read Comparison Result Flag 6		R
b7	CASUC7	Read Comparison Result Flag 7		R
b8	CASUC8	Read Comparison Result Flag 8		R
b9	CASUC9	Read Comparison Result Flag 9		R
b10	CASUC10	Read Comparison Result Flag 10		R
b11	CASUC11	Read Comparison Result Flag 11		R
b12	CASUC12	Read Comparison Result Flag 12		R
b13	CASUC13	Read Comparison Result Flag 13		R
b14	CASUC14	Read Comparison Result Flag 14		R
b15	CASUC15	Read Comparison Result Flag 15		R
b16	CASUC16	Read Comparison Result Flag 16		R
b17	CASUC17	Read Comparison Result Flag 17		R
b18	CASUC18	Read Comparison Result Flag 18		R
b19	CASUC19	Read Comparison Result Flag 19		R
b20	CASUC20	Read Comparison Result Flag 20		R
b21	CASUC21	Read Comparison Result Flag 21		R
b22	CASUC22	Read Comparison Result Flag 22		R
b23	CASUC23	Read Comparison Result Flag 23		R
b24	CASUC24	Read Comparison Result Flag 24		R
b25	CASUC25	Read Comparison Result Flag 25		R
b26	CASUC26	Read Comparison Result Flag 26		R
b27	CASUC27	Read Comparison Result Flag 27		R
b28	CASUC28	Read Comparison Result Flag 28		R
b29	CASUC29	Read Comparison Result Flag 29		R
b30	CASUC30	Read Comparison Result Flag 30		R
b31	CASUC31	Read Comparison Result Flag 31		R

This register indicates the result of automatic calibration in the XSPI controller module.

CASUCm Flag (Read Comparison Result Flag m) (m = 0 to 31)

This flag indicates that the written and read values match when the DS signal delay (number of delay cells) is m. This flag is updated each time the calibration sequence is performed.
If any one of the CASUCm flags is set to 1 at the end of the calibration sequence, the INTS.CASUCCSi flag is set to 1; if all are 0, the INTS.CAFAILCSi flag is set to 1.

48.2.2.32 Interrupt Status Register (INTS)

Address(es): XSPI0.INTS 8786 8190h, XSPI1.INTS 8786 8590h, XSPI0_NS.INTS 0786 8190h, XSPI1_NS.INTS 0786 8590h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CASUCCS1	CASUCCS0	CAFAILCS1	CAFAILCS0	—	—	—	—	—	—	—	BUSERRCH0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	INTCS1	INTCS0	—	—	ECSCS1	ECSCS0	—	—	DSTOCS1	DSTOCS0	PERTO	—	PATCMP	CMDCMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDCMP	Command Issuance Completion Flag	0: Command issuance not complete 1: Command issuance completed	R
b1	PATCMP	Signal Pattern Generation Completion Flag	0: Signal pattern generation not completed 1: Signal pattern generation completed	R
b2	—	Reserved	This bit is read as 0.	R
b3	PERTO	Periodic Transaction Timeout Flag	0: No timeout detected 1: Timeout detected	R
b4	DSTOCS0	DS Timeout Flag 0	0: No DS timeout on target device 0 1: DS timeout occurred on target device 0	R
b5	DSTOCS1	DS Timeout Flag 1	0: No DS timeout on target device 1 1: DS timeout occurred on target device 1	R
b7, b6	—	Reserved	These bits are read as 0.	R
b8	ECSCS0	ECC Error Detection Flag 0	0: No ECC errors on target device 0 1: ECC error occurred on target device 0	R
b9	ECSCS1	ECC Error Detection Flag 1	0: No ECC errors on target device 1 1: ECC error occurred on target device 1	R
b11, b10	—	Reserved	These bits are read as 0.	R
b12	INTCS0	INT0 Detection Flag	0: No interrupt request from target device 0 1: An interrupt is requested from target device 0	R
b13	INTCS1	INT1 Detection Flag	0: No interrupt request from target device 1 1: An interrupt is requested from target device 1	R
b19 to b14	—	Reserved	These bits are read as 0.	R
b20	BUSERRCH0	Internal Bus Error Flag	0: No bus error 1: Bus error occurred	R
b27 to b21	—	Reserved	These bits are read as 0.	R
b28	CAFAILCS0	Calibration Failure Flag 0	0: Calibration for target device 0 has not failed. 1: Calibration for target device 0 has failed.	R
b29	CAFAILCS1	Calibration Failure Flag 1	0: Calibration for target device 1 has not failed. 1: Calibration for target device 1 has failed.	R
b30	CASUCCS0	Calibration Success Flag 0	0: Calibration for target device 0 has not succeeded. 1: Calibration for target device 0 has succeeded.	R
b31	CASUCCS1	Calibration Success Flag 1	0: Calibration for target device 1 has not succeeded. 1: Calibration for target device 1 has succeeded.	R

This register indicates the interrupt status of the XSPI controller module.
Each flag can be cleared by writing 1 to the corresponding bit in the INTC register.

CMDCMP Flag (Command Issuance Completion Flag)

This flag indicates that the issuance of the specified number of transactions has been completed in indirect access mode.

In sequential mode, it means that the issuance of all transactions specified by the CDCTL0.TRNUM[1:0] bits has been completed. In periodic mode, it means that the read value matched the expected value within the number of times specified by the CDCTL0.PERREP[3:0] bits.

PATCMP Flag (Signal Pattern Generation Completion Flag)

This flag indicates that generation of the requested signal pattern (XIP disable pattern, reset pattern, CS only pattern) has been completed.

PERTO Flag (Periodic Transaction Timeout Flag)

This flag indicates that, in periodic mode, the read value did not match the expected value even if the count specified by the CDCTL0.PERREP[3:0] bits is reached.

DSTOCSi Flag (DS Timeout Flag i) (i = 0, 1)

This flag indicates that the DS signal was not input in a read transaction that uses the DS signal.
This means that data was not received within the expected data phase. In this case, the XSPI aborts this read transaction and any subsequent transactions.
This error may issue an error response to the internal bus.

ECSCSi Flag (ECC Error Detection Flag i) (i = 0, 1)

This flag indicates that an ECC error has occurred on target device i.
This flag becomes 1 when the falling edge of the XSPIn_ECSi# signal is detected (n = 0, 1).
This flag is used for target devices that have ECC error detection capability.

INTCSi Flag (INTi Detection Flag) (i = 0, 1)

This flag indicates that an interrupt request has been received from target device i.
This flag becomes 1 when the falling edge of the XSPIn_INTi# signal is detected.
This flag is used for target devices that have interrupt functionality.

BUSERRCH0 Flag (Internal Bus Error Flag)

This flag indicates that a internal bus error has occurred.

CAFAILCSi Flag (Calibration Failure Flag i) (i = 0, 1)

This flag indicates that calibration has failed.
When the calibration sequence ends, this flag becomes 1 if all CASTTCSi.CASUCm flags are 0.

CASUCCSi Flag (Calibration Success Flag i) (i = 0, 1)

This flag indicates that calibration has succeeded.
When the calibration sequence ends, this flag becomes 1 if any of the CASTTCSi.CASUCm flags are 1.

48.2.2.33 Interrupt Clear Register (INTC)

Address(es): XSPI0.INTC 8786 8194h, XSPI1.INTC 8786 8594h, XSPI0_NS.INTC 0786 8194h, XSPI1_NS.INTC 0786 8594h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CASUCCS1C	CASUCCS0C	CAFAILCS1C	CAFAILCS0C	—	—	—	—	—	—	—	BUSERRCH0C	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	INTCS1C	INTCS0C	—	—	ECSCS1C	ECSCS0C	—	—	DSTOCS1C	DSTOCS0C	PERTOC	—	PATCMPC	CMDCMPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDCMPC	Command Issuance Completion Flag Clear	Writing 1 to this bit clears the INTS.CMDCMP flag. Writing 0 has no effect.	W
b1	PATCMPC	Signal Pattern Generation Completion Flag Clear	Writing 1 to this bit clears the INTS.PATCMP flag. Writing 0 has no effect.	W
b2	—	Reserved	The write value should be 0.	W
b3	PERTOC	Periodic Transaction Timeout Flag Clear	Writing 1 to this bit clears the INTS.PERTO flag. Writing 0 has no effect.	W
b4	DSTOCS0C	DS Timeout Flag 0 Clear	Writing 1 to this bit clears the INTS.DSTOCS0 flag. Writing 0 has no effect.	W
b5	DSTOCS1C	DS Timeout Flag 1 Clear	Writing 1 to this bit clears the INTS.DSTOCS1 flag. Writing 0 has no effect.	W
b7, b6	—	Reserved	The write value should be 0.	W
b8	ECSCS0C	ECC Error Detection Flag 0 Clear	Writing 1 to this bit clears the INTS.ECSCS0 flag. Writing 0 has no effect.	W
b9	ECSCS1C	ECC Error Detection Flag 1 Clear	Writing 1 to this bit clears the INTS.ECSCS1 flag. Writing 0 has no effect.	W
b11, b10	—	Reserved	The write value should be 0.	W
b12	INTCS0C	INT0 Detection Flag Clear	Writing 1 to this bit clears the INTS.INTCS0 flag. Writing 0 has no effect.	W
b13	INTCS1C	INT1 Detection Flag Clear	Writing 1 to this bit clears the INTS.INTCS1 flag. Writing 0 has no effect.	W
b19 to b14	—	Reserved	The write value should be 0.	W
b20	BUSERRCH0C	Internal Bus Error Flag Clear	Writing 1 to this bit clears the INTS.BUSERRCH0 flag. Writing 0 has no effect.	W
b27 to b21	—	Reserved	The write value should be 0.	W
b28	CAFAILCS0C	Calibration Failure Flag 0 Clear	Writing 1 to this bit clears the INTS.CAFAILCS0 flag. Writing 0 has no effect.	W
b29	CAFAILCS1C	Calibration Failure Flag 1 Clear	Writing 1 to this bit clears the INTS.CAFAILCS1 flag. Writing 0 has no effect.	W
b30	CASUCCS0C	Calibration Success Flag 0 Clear	Writing 1 to this bit clears the INTS.CASUCCS0 flag. Writing 0 has no effect.	W
b31	CASUCCS1C	Calibration Success Flag 1 Clear	Writing 1 to this bit clears the INTS.CASUCCS1 flag. Writing 0 has no effect.	W

This register clears the interrupt status flags of the XSPI controller module.
Writing 1 to each bit clears the corresponding flag in the INTS register.

48.2.2.34 Interrupt Enable Register (INTE)

Address(es): XSPI0.INTE 8786 8198h, XSPI1.INTE 8786 8598h, XSPI0_NS.INTE 0786 8198h, XSPI1_NS.INTE 0786 8598h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CASUCCS1E	CASUCCS0E	CAFAILCS1E	CAFAILCS0E	—	—	—	—	—	—	—	BUSERRCH0E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	INTCS1E	INTCS0E	—	—	ECSCS1E	ECSCS0E	—	—	DSTOCS1E	DSTOCS0E	PERTOE	—	PATCMPE	CMDCMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDCMPE	Command Issuance Completion Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b1	PATCMPE	Signal Pattern Generation Completion Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PERTOE	Periodic Transaction Timeout Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b4	DSTOCS0E	DS Timeout Interrupt 0 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b5	DSTOCS1E	DS Timeout Interrupt 1 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ECSCS0E	ECC Error Detection Interrupt 0 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b9	ECSCS1E	ECC Error Detection Interrupt 1 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	INTCS0E	INT0 Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b13	INTCS1E	INT1 Detection Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b19 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	BUSERRCH0E	Internal Bus Error Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b27 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	CAFAILCS0E	Calibration Failure Interrupt 0 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b29	CAFAILCS1E	Calibration Failure Interrupt 1 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b30	CASUCCS0E	Calibration Success Interrupt 0 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b31	CASUCCS1E	Calibration Success Interrupt 1 Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W

This register enables each interrupt generated by the XSPI controller module.
If the corresponding flag in the INTS register becomes 1 while a bit in this register is 1, an interrupt request is generated.

48.2.3 Target Mode Registers

48.2.3.1 L1.5 Configuration Register 0 (L15CFG0)

Address(es): XSPI0.L15CFG0 8786 7000h, XSPI1.L15CFG0 8786 7400h, XSPI0_NS.L15CFG0 0786 7000h, XSPI1_NS.L15CFG0 0786 7400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRMSKMD	—	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PRTMD[9:0]	Protocol Mode Setting	000h: 1S-1S-1S 3B2h: 4S-4D-4D 3FFh: 8D-8D-8D Settings other than above are prohibited.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	WRMSKMD	Write Data Mask Enable	0: Write data mask is disabled. 1: Write data mask is enabled.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	OEASTEX	Output Enable Signal Assertion Period Extension	0: Do not extend the output enable signal 1: Extend the output enable signal by one cycle	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI memory write/read transaction, stop all write/read transactions before rewriting it.

This register specifies the communication protocol for the XSPI target module.

PRTMD[9:0] Bits (Protocol Mode Setting)

These bits specify the IO protocol mode.
In 1S-1S-1S protocol modes, the CK signal is used for data sampling. In 4S-4D-4D protocol mode and 8D-8D-8D protocol mode, the DS signal is used for data sampling.

WRMSKMD Bit (Write Data Mask Enable)

This bit configures whether to use the XSPIn_DS signal as a mask signal for each byte in the write data (n = 0, 1).
The write data mask function is only used in 8D-8D-8D protocol mode.

OEASTEX Bit (Output Enable Signal Assertion Period Extension)

This bit specifies the negation timing of the output buffer control signal (OE: Output Enable). When this bit is set to 1, the output of the XSPIn_IO7 to XSPIn_IO0 pins and the XSPIn_DS pin is extended by one cycle (n = 0, 1).
To prevent potential collisions between output and input data, do not set this bit to 1 if there is no latency cycle.

48.2.3.2 L1.5 Configuration Register 1 (L15CFG1)

Address(es): XSPI0.L15CFG1 8786 7004h, XSPI1.L15CFG1 8786 7404h, XSPI0_NS.L15CFG1 0786 7004h, XSPI1_NS.L15CFG1 0786 7404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDLATE[7:0]								WRLATE[7:0]							
Value after reset:	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	WRLATE[7:0]	Write Latency Cycle Setting	00h: No latency 01h: 1 cycle : : 1Eh: 30 cycles 1Fh: 31 cycles Settings other than above are prohibited.	R/W
b15 to b8	RDLATE[7:0]	Read Latency Cycle Setting	Set these bits to 0Fh (15 cycles).	R/W
b17, b16	FFMT[1:0]	Frame Format Setting	b17 b16 0 0: Normal format 0 1: 8D-8D-8D profile 1.0 format 1 0: 8D-8D-8D profile 2.0 command modifier format 1 1: 8D-8D-8D profile 2.0 extended command modifier format	R/W
b19, b18	ADDSIZE[1:0]	Address Size	b19 b18 0 1: 2 bytes (64-Kbyte address space) 1 0: 3 bytes (16-Mbyte address space) 1 1: 4 bytes (4-Gbyte address space) Settings other than above are prohibited.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI memory write/read transaction, stop all write/read transactions before rewriting it.

This register specifies the frame format for the XSPI target module.

WRLATE[7:0] Bits (Write Latency Cycle Setting)

These bits specify the number of latency cycles to insert during write transactions when accessing memory.

RDLATE[7:0] Bits (Read Latency Cycle Setting)

These bits specify the number of latency cycles to insert during read transactions when accessing memory.

FFMT[1:0] Bits (Frame Format Setting)

These bits configure the xSPI frame format when accessing memory. Table 48.4 shows the frame format for each setting value.

ADDSIZE[1:0] Bits (Address Size)

These bits configure the number of bytes in the address field when accessing memory.
When setting the FFMT[1:0] bits to 10b or 11b (8D-8D-8D Profile 2.0), set this bit to 11b (4 bytes).

48.2.3.3 L1.5 Configuration Register 2 (L15CFG2)

Address(es): XSPI0.L15CFG2 8786 7008h, XSPI1.L15CFG2 8786 7408h, XSPI0_NS.L15CFG2 0786 7008h, XSPI1_NS.L15CFG2 0786 7408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	MRDFSIZE[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MWRFSIZE[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	MWRFSIZE[10:0]	Maximum Write Frame Size Setting	These bits specify the maximum data size for the write frame.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	MRDFSIZE[10:0]	Maximum Read Frame Size Setting	These bits specify the maximum data size for the read frame.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI memory write/read transaction, stop all write/read transactions before rewriting it.

This register specifies the frame format for the XSPI target module.

MWRFSIZE[10:0] Bits (Maximum Write Frame Size Setting)

These bits are used to set the maximum data size in bytes for the write frame when accessing memory.
When the xSPI controller writes data to SRAM, the data size must match this size except for the last write data frame.
The value set for these bits must be less than or equal to the buffer size indicated by the TBI.BUFSIZE[12:0] bits.
The possible values are 8, 16, 32, 64, 128, 256, and 512.

MRDFSIZE[10:0] Bits (Maximum Read Frame Size Setting)

These bits are used to set the maximum data size in bytes for the read frame when accessing memory.
When the xSPI controller reads data from SRAM, the data size must match this size except for the last read data frame.
The value set for these bits must be less than or equal to the buffer size indicated by the TBI.BUFSIZE[12:0] bits.
The possible values are 8, 16, 32, 64, 128, 256, and 512.

48.2.3.4 L2 Configuration Register 0 (L2CFG0)

Address(es): XSPI0.L2CFG0 8786 7040h, XSPI1.L2CFG0 8786 7440h, XSPI0_NS.L2CFG0 0786 7040h, XSPI1_NS.L2CFG0 0786 7440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI memory write/read transaction, stop all write/read transactions before rewriting it.

This register specifies the address space of the XSPI target module.

Set the highest address (= memory size – 1) of the memory space exposed externally as an xSPI target. Set the lower 12 bits to 1.

Only the nine values shown in Table 48.7 are configurable.

Table 48.7 xSPI Target Address Space and Register Setting

xSPI Address Space	Memory Size	L2CFG0 Register Setting Value
0000 0000h to 0000 0FFFh	4 Kbytes	00000FFFh
0000 0000h to 0000 1FFFh	8 Kbytes	00001FFFh
0000 0000h to 0000 3FFFh	16 Kbytes	00003FFFh
0000 0000h to 0000 7FFFh	32 Kbytes	00007FFFh
0000 0000h to 0000 FFFFh	64 Kbytes	0000FFFFh
0000 0000h to 0001 FFFFh	128 Kbytes	0001FFFFh
0000 0000h to 0003 FFFFh	256 Kbytes	0003FFFFh
0000 0000h to 0007 FFFFh	512 Kbytes	0007FFFFh
0000 0000h to 000F FFFFh	1 Mbyte	000FFFFFh

48.2.3.5 L2 Configuration Register 1 (L2CFG1)

Address(es): XSPI0.L2CFG1 8786 7044h, XSPI1.L2CFG1 8786 7444h, XSPI0_NS.L2CFG1 0786 7044h, XSPI1_NS.L2CFG1 0786 7444h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WRMCMD[7:0]								WRRCMD[7:0]							
Value after reset:	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDMCMD[7:0]								RDRCMD[7:0]							
Value after reset:	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RDRCMD[7:0]	Command Code for Read Register Linear*1	Set the command code for the Read Register Linear command.	R/W
b15 to b8	RDMCMD[7:0]	Command Code for Read Memory Linear*1	Set the command code for the Read Memory Linear command.	R/W
b23 to b16	WRRCMD[7:0]	Command Code for Write Register Linear*1	Set the command code for the Write Register Linear command.	R/W
b31 to b24	WRMCMD[7:0]	Command Code for Write Memory Linear*1	Set the command code for the Write Memory Linear command.	R/W

Note: This register should be set during the initialization phase before issuing xSPI transactions. If you need to change the settings of this register after starting an xSPI memory write/read transaction, stop all write/read transactions before rewriting it.

Note 1. For Profile 2.0 commands with extended command modifier, lower 5 bits are ignored.

This register specifies the commands processed by the XSPI target module.

48.2.3.6 Target Write Address Register (TWADR)

Address(es): XSPI0.TWADR 8786 7100h, XSPI1.TWADR 8786 7500h, XSPI0_NS.TWADR 0786 7100h, XSPI1_NS.TWADR 0786 7500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register specifies the absolute address when the xSPI controller writes data to the memory of this MCU.
When using DDR mode, specify an even address.

48.2.3.7 Target Write Length Register (TWLEN)

Address(es): XSPI0.TWLEN 8786 7104h, XSPI1.TWLEN 8786 7504h, XSPI0_NS.TWLEN 0786 7104h, XSPI1_NS.TWLEN 0786 7504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register specifies the total data size when the xSPI controller writes data to the memory of this MCU.
The configurable value range is 0000 0001h to 0010 0000h. When using DDR mode, specify an even value.

48.2.3.8 Target Write Control Register (TWCNT)

Address(es): XSPI0.TWCNT 8786 7108h, XSPI1.TWCNT 8786 7508h, XSPI0_NS.TWCNT 0786 7108h, XSPI1_NS.TWCNT 0786 7508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WEN	Write Enable	Writing 1 to this bit starts a write transfer. Writing 0 has no effect. The read value has no meaning.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables write transfers to the XSPI target module.

WEN Bit (Write Enable)

Writing 1 to this bit starts a write transfer, using the value in the TWADR register as the start address and the value in the TWLEN register as the data size. Do not set this bit to 1 at the same time as the TRCNT.REN bit.

48.2.3.9 Target Read Address Register (TRADR)

Address(es): XSPI0.TRADR 8786 7140h, XSPI1.TRADR 8786 7540h, XSPI0_NS.TRADR 0786 7140h, XSPI1_NS.TRADR 0786 7540h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register specifies the absolute address when the xSPI controller reads data from the memory of this MCU.
When using DDR mode, specify an even address.

48.2.3.10 Target Read Length Register (TRLEN)

Address(es): XSPI0.TRLEN 8786 7144h, XSPI1.TRLEN 8786 7544h, XSPI0_NS.TRLEN 0786 7144h, XSPI1_NS.TRLEN 0786 7544h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register specifies the total data size when the xSPI controller read data from the memory of this MCU.
The configurable value range is 0000 0001h to 0010 0000h. When using DDR mode, specify an even value.

48.2.3.11 Target Read Control Register (TRCNT)

Address(es): XSPI0.TRCNT 8786 7148h, XSPI1.TRCNT 8786 7548h, XSPI0_NS.TRCNT 0786 7148h, XSPI1_NS.TRCNT 0786 7548h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REN	Read Enable	Writing 1 to this bit starts a read transfer. Writing 0 has no effect. The read value has no meaning.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables write transfers to the XSPI target module.

REN Bit (Read Enable)

Writing 1 to this bit starts a read transfer, using the value in the TRADR register as the start address and the value in the TRLEN register as the data size. Do not set this bit to 1 at the same time as the TWCNT.WEN bit.

48.2.3.12 Transfer Control Register (TRNSC)

Address(es): XSPI0.TRNSC 8786 7180h, XSPI1.TRNSC 8786 7580h, XSPI0_NS.TRNSC 0786 7180h, XSPI1_NS.TRNSC 0786 7580h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSMSTB	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	RSMSTB	Software Standby Mode Transition Setting	0: Operation enabled 1: Operation disabled (transition to software standby mode possible)	R/W

This register controls the data transfer of the XSPI target module.

RSMSTB Bit (Software Standby Mode Transition Setting)

This bit is used to transition to and return from software standby mode.

Before transitioning to software standby mode, set this bit to 1 while no communication is being performed. When this bit is 1, communication functions will stop. After returning from software standby mode, set this bit to 0 before resuming operation.

48.2.3.13 Target Write Buffer Status Register (TWBS)

Address(es): XSPI0.TWBS 8786 7200h, XSPI1.TWBS 8786 7600h, XSPI0_NS.TWBS 0786 7200h, XSPI1_NS.TWBS 0786 7600h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VACSIZE[12:0]												
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	VACSIZE[12:0]	Write Buffer Free Size	These bits indicate the free space in the write buffer. The xSPI controller can only write data up to the value indicated by these bits.	R
b30 to b13	—	Reserved	These bits are read as 0.	R
b31	ERR	Error Flag	This flag indicates that any of the error flags in the TS register (SBERR, MFSEERR, UNKCERR, BUDFERR, BOVFERR) is set to 1.	R

This register indicates the status of write transfer in the XSPI target module.

48.2.3.14 Target Write Transfer Status Register (TWTS)

Address(es): XSPI0.TWTS 8786 7204h, XSPI1.TWTS 8786 7604h, XSPI0_NS.TWTS 0786 7204h, XSPI1_NS.TWTS 0786 7604h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	LWDSIZE[20:16]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LWDSIZE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b20 to b0	LWDSIZE[20:0]	Remaining Write Data Size	These bits indicate the remaining data size for the write transfer. Immediately after setting the TWCNT.WEN bit to 1, these bits indicate the data size set in the TWLEN register. After the write transfer completes, these bits indicate 000000h.	R
b31 to b21	—	Reserved	These bits are read as 0.	R

This register indicates the status of write transfer in the XSPI target module.

48.2.3.15 Target Read Buffer Status Register (TRBS)

Address(es): XSPI0.TRBS 8786 7240h, XSPI1.TRBS 8786 7640h, XSPI0_NS.TRBS 0786 7240h, XSPI1_NS.TRBS 0786 7640h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DATSIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	DATSIZE[12:0]	Read Buffer Data Size	These bits indicate the data size stored in the read buffer. The xSPI controller can only read data up to the value indicated by these bits.	R
b30 to b13	—	Reserved	These bits are read as 0.	R
b31	ERR	Error Flag	This flag indicates that any of the error flags in the TS register (SBERR, MFSEERR, UNKCERR, BUDFERR, BOVFERR) is set to 1.	R

This register indicates the status of read transfer in the XSPI target module.

48.2.3.16 Target Read Transfer Status Register (TRTS)

Address(es): XSPI0.TRTS 8786 7244h, XSPI1.TRTS 8786 7644h, XSPI0_NS.TRTS 0786 7244h, XSPI1_NS.TRTS 0786 7644h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	LRDSIZE[20:16]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LRDSIZE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b20 to b0	LRDSIZE[20:0]	Remaining Read Data Size	These bits indicate the remaining data size for the read transfer. Immediately after setting the TRCNT.REN bit to 1, these bits indicate the data size set in the TRLEN register. After the read transfer completes, these bits indicate 000000h.	R
b31 to b21	—	Reserved	These bits are read as 0.	R

This register indicates the status of read transfer in the XSPI target module.

48.2.3.17 Target Status Register (TS)

Address(es): XSPI0.TS 8786 7280h, XSPI1.TS 8786 7680h, XSPI0_NS.TS 0786 7280h, XSPI1_NS.TS 0786 7680h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SBERR	—	—	MFSEERR	UNKCERR	—	—	BUDFERR	BOVFERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EV1	EV0	—	—	SRSTO	SRST	—	—	—	TCMP	—	—	—	TIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TIP	Transfer Status Flag	0: No transfer 1: Transfer exist	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	TCMP	Transfer Complete Flag	0: Transfer not complete 1: Transfer complete	R
b7 to b5	—	Reserved	These bits are read as 0.	R
b8	SRST	SPI Reset Detection Flag	0: No reset detected 1: Reset detected	R
b9	SRSTO	SPI Reset Indication Output Flag	0: No reset indication is output 1: Reset indication is output	R
b11, b10	—	Reserved	These bits are read as 0.	R
b12	EV0	Event 0 Flag	0: No event 1: Event present	R
b13	EV1	Event 1 Flag	0: No event 1: Event present	R
b15, b14	—	Reserved	These bits are read as 0.	R
b16	BOVFERR	Buffer Overflow Flag	0: No buffer overflow 1: Buffer overflow	R
b17	BUDFERR	Buffer Underflow Flag	0: No buffer underflow 1: Buffer underflow	R
b19, b18	—	Reserved	These bits are read as 0.	R
b20	UNKCERR	Unknown Command Error Flag	0: No unknown command error 1: Unknown command error	R
b21	MFSEERR	Maximum Frame Size Exceeded Flag	0: No maximum frame size exceeded 1: Maximum frame size exceeded	R
b23, b22	—	Reserved	These bits are read as 0.	R
b24	SBERR	Internal Bus Error Flag	0: No internal bus error 1: Internal bus error present	R
b31 to b25	—	Reserved	These bits are read as 0.	R

This register indicates the status the XSPI target module.

TIP Flag (Transfer Status Flag)

This flag indicates the transfer status, including internal bus transfers. If an error is detected, wait until this flag becomes 1 before starting the next transfer.

SRSTO Flag (SPI Reset Indication Output Flag)

This flag indicates the status of a reset output. Setting this flag to 1 drives the RSTO0# signal low.

EV0 Flag (Event 0 Flag)

This flag indicates the status of a software event. This flag assumes events from the controller to the target.

EV1 Flag (Event 1 Flag)

This flag indicates the status of a software event. This flag assumes events from the target to the controller.

BOVFERR Flag (Buffer Overflow Flag)

This flag indicates the presence of a buffer overflow. This flag is set to 1 when the xSPI controller writes the data larger than the free size indicated by the TWBS.VACSIZE[12:0] bits.

BUDFERR Flag (Buffer Underflow Flag)

This flag indicates the presence of a buffer underflow. This flag is set to 1 when the xSPI controller reads the data larger than the data size indicated by the TRBS.DATSIZE[12:0] bits.

UNKCERR Flag (Unknown Command Error Flag)

This flag indicates that the XSPI target received an unknown command as listed below.

- When the L2CFG1 register is not configured.
- When a Read Memory command is issued while the TRCNT.REN bit is 0.
- When a Write Memory command is issued while the TWCNT.WEN bit is 0.
- When the inverted command is not matched with the expected value in 8D-8D-8D profile1.0 protocol mode.

MFSEERR Flag (Maximum Frame Size Exceeded Flag)

This flag indicates whether the maximum frame size has been exceeded. The maximum frame size is specified as follows: 4 bytes for register accesses, the L15CFG2.MRDFSIZE[10:0] bits for memory read accesses, and the MWRFSIZE[10:0] bits for memory write accesses.

SBERR Flag (Internal Bus Error Flag)

This flag indicates that the internal bus returned an error response or an unexpected response.

48.2.3.18 Target Status Clear Register (TSC)

Address(es): XSPI0.TSC 8786 7284h, XSPI1.TSC 8786 7684h, XSPI0_NS.TSC 0786 7284h, XSPI1_NS.TSC 0786 7684h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SBERRC	—	—	MFSEERRC	UNKCERRC	—	—	BUDFERRC	BOVFERRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EV1C	EV0C	—	—	SRSTOC	SRSTC	—	—	—	TCMPC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	The write value should be 0.	W
b4	TCMPC	Transfer Complete Flag Clear	Writing 1 to this bit clears the TS.TCMP flag. Writing 0 has no effect.	W
b7 to b5	—	Reserved	The write value should be 0.	W
b8	SRSTC	SPI Reset Detection Flag Clear	Writing 1 to this bit clears the TS.SRST flag. Writing 0 has no effect.	W
b9	SRSTOC	SPI Reset Indication Output Flag Clear	Writing 1 to this bit clears the TS.SRSTO flag. Writing 0 has no effect.	W
b11, b10	—	Reserved	The write value should be 0.	W
b12	EV0C	Event 0 Flag Clear	Writing 1 to this bit clears the TS.EV0 flag. Writing 0 has no effect.	W
b13	EV1C	Event 1 Flag Clear	Writing 1 to this bit clears the TS.EV1 flag. Writing 0 has no effect.	W
b15, b14	—	Reserved	The write value should be 0.	W
b16	BOVFERRC	Buffer Overflow Flag Clear	Writing 1 to this bit clears the TS.BOVFERR flag. Writing 0 has no effect.	W
b17	BUDFERRC	Buffer Underflow Flag Clear	Writing 1 to this bit clears the TS.BUDFERR flag. Writing 0 has no effect.	W
b19, b18	—	Reserved	The write value should be 0.	W
b20	UNKCERRC	Unknown Command Error Flag Clear	Writing 1 to this bit clears the TS.UNKCERR flag. Writing 0 has no effect.	W
b21	MFSEERRC	Maximum Frame Size Exceeded Flag Clear	Writing 1 to this bit clears the TS.MFSEERR flag. Writing 0 has no effect.	W
b23, b22	—	Reserved	The write value should be 0.	W
b24	SBERRC	Internal Bus Error Flag Clear	Writing 1 to this bit clears the TS.SBERR flag. Writing 0 has no effect.	W
b31 to b25	—	Reserved	The write value should be 0.	W

This register clears the status flags of the XSPI target module.
Writing 1 to each bit clears the corresponding flag in the TS register.

48.2.3.19 Target Status Set Register (TSS)

Address(es): XSPI0.TSS 8786 7288h, XSPI1.TSS 8786 7688h, XSPI0_NS.TSS 0786 7288h, XSPI1_NS.TSS 0786 7688h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EV1S	EV0S	—	—	SRSTOS	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	The write value should be 0.	W
b9	SRSTOS	SRSTO Flag Set	Writing 1 to this bit sets the TS.SRSTO flag to 1. Writing 0 has no effect.	W
b11, b10	—	Reserved	The write value should be 0.	W
b12	EV0S	EV0 Flag Set	Writing 1 to this bit sets the TS.EV0 flag to 1. Writing 0 has no effect.	W
b13	EV1S	EV1 Flag Set	Writing 1 to this bit sets the TS.EV1 flag to 1. Writing 0 has no effect.	W
b31 to b14	—	Reserved	The write value should be 0.	W

This register sets the status flags of the XSPI target module.
Writing 1 to each bit sets the corresponding flag in the TS register.

48.2.3.20 Target Status xSPI Interrupt Enable Register (TSXIE)

Address(es): XSPI0.TSXIE 8786 728Ch, XSPI1.TSXIE 8786 768Ch, XSPI0_NS.TSXIE 0786 728Ch, XSPI1_NS.TSXIE 0786 768Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SBERRXIE	—	—	MFSEERRXIE	UNKCERRXIE	—	—	BUDFERRXIE	BOVFERRXIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EV1XIE	EV0XIE	—	—	SRSTOXIE	—	—	—	—	TCMPXIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TCMPXIE	Transfer Complete Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b8 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	SRSTOXIE	SPI Reset Output Enable	0: RSTO0# output is disabled. 1: RSTO0# output is enabled.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	EV0XIE	Event 0 Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b13	EV1XIE	Event 1 Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	BOVFERRXIE	Buffer Overflow Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b17	BUDFERRXIE	Buffer Underflow Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	UNKCERRXIE	Unknown Command Error Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b21	MFSEERRXIE	Maximum Frame Size Exceed Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	SBERRXIE	Internal Bus Error Interrupt Enable	0: INT0# output is disabled. 1: INT0# output is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables each interrupt generated by the XSPI target module.
If the corresponding flag in the TS register is set to 1 while a bit in this register is 1, an interrupt request (reset status signal) is output from the INT0# pin (RSTO0# pin).

48.2.3.21 Target Status System Interrupt Enable Register (TSSIE)

Address(es): XSPI0.TSSIE 8786 7290h, XSPI1.TSSIE 8786 7690h, XSPI0_NS.TSSIE 0786 7290h, XSPI1_NS.TSSIE 0786 7690h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SBERRSIE	—	—	MFSEERRSIE	UNKCERRSIE	—	—	BUDFERRSIE	BOVFERRSIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EV1SIE	EV0SIE	—	—	—	SRSTSIE	—	—	—	TCMPSIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TCMPSIE	Transfer Complete Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SRSTSIE	SPI Reset Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	EV0SIE	Event 0 Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b13	EV1SIE	Event 1 Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	BOVFERRSIE	Buffer Overflow Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b17	BUDFERRSIE	Buffer Underflow Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	UNKCERRSIE	Unknown Command Error Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b21	MFSEERRSIE	Maximum Frame Size Exceeded Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	SBERRSIE	Internal Bus Error Interrupt Enable	0: Interrupt is disabled. 1: Interrupt is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables each interrupt generated by the XSPI target module.
If the corresponding flag in the TS register becomes 1 while a bit in this register is 1, an interrupt request is generated.

48.2.3.22 Target Basic Information Register (TBI)

Address(es): XSPI0.TBI 8786 72C0h, XSPI1.TBI 8786 76C0h, XSPI0_NS.TBI 0786 72C0h, XSPI1_NS.TBI 0786 76C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VER[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BUFSIZE[12:0]												
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	BUFSIZE[12:0]	Buffer Size	These bits indicates the size of the write/read buffer installed. This product has a 1-Kbyte buffer installed.	R
b15 to b13	—	Reserved	These bits are read as 0.	R
b31 to b16	VER[15:0]	Version Register	These bits indicates the version of the IP installed.	R

This register indicates information about the XSPI target module.

48.3 Operation

48.3.1 XSPI Controller Module

The XSPI controller has the function to issue transactions to external memory devices equipped with an xSPI target interface.

The XSPI controller has two modes for issuing transactions. One is indirect access mode, which accesses external memory via XSPI registers. After setting all fields of the xSPI frame, the transaction is initiated by a software request. The other is memory-mapped mode. Reading or writing any address within the XSPI area (addresses 2000 0000h to 5FFF FFFFh) is automatically converted into an xSPI transaction. Each access occurs without needing to manipulate XSPI registers, allowing access to external memory as if it were directly connected to the internal bus.

48.3.1.1 xSPI Bus

Figure 48.3 shows an example of the connection between the XSPI controller and the memory devices.
In this figure, the XSPI0_CS0 pin is connected to the RAM device, and the XSPI0_CS1 pin is connected to the Flash memory device.
Specify the pull-up resistor values according to the instructions for each device.

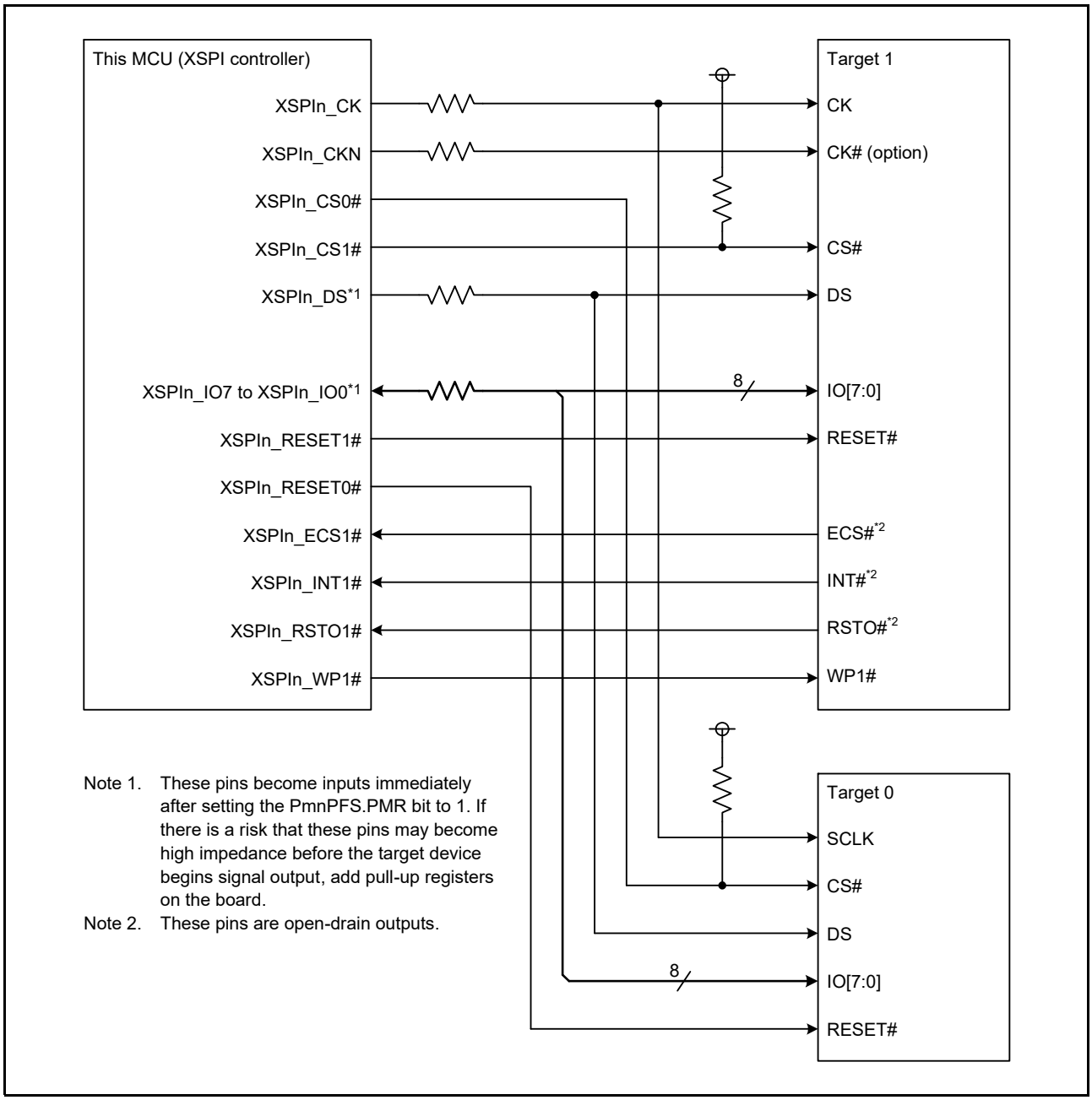


Figure 48.3 Example of Connection between XSPI Controller and Memory Devices (n = 0, 1)

(1) Supported Protocol Mode

The XSPI controller supports various protocol modes. Protocol mode is set by the LIOCFGCSi.PRTMD[9:0] bits (i = 0, 1). Table 48.8 provides an overview of each protocol mode.

Table 48.8 Supported Protocol Mode (n = 0, 1; i = 0, 1)

Protocol Mode	Function	LIOCFGCSi.PRTMD[9:0] Bit Setting	Note
1S-1S-1S	Command, Address and Data field are transferred at SDR with using 1data input pin and 1 data output pin. Read data is sampled with the XSPIn_CK signal.	000h	Specified by JESD251
4S-4D-4D	Command field is transferred at SDR with using 4 data pins. Address and Data field are transferred at DDR with using 4 data pins. Read data is sampled with the XSPIn_DS signal.	3B2h	Specified by JESD251-1
8D-8D-8D	Command, Address and Data fields are transferred at DDR with using 8 data pins. Read data is sampled with the XSPIn_DS signal.	3FFh	Specified by JESD251
1S-2S-2S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with the XSPIn_CK signal.	048h	—
2S-2S-2S	Command, Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with the XSPIn_CK signal.	049h	—
1S-4S-4S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 4 data pins. Read data is sampled with the XSPIn_CK signal.	090h	—
4S-4S-4S	Command, Address and Data field are transferred at SDR with using 4 data pins. Read data is sampled with the XSPIn_CK signal.	092h	—

Note: When XIP mode is enabled in memory-mapped mode, XIP code is inserted into Latency field.

Bytes in the command field and address field are transferred in highest order to lowest order sequence. Sequential bytes in the data field are transferred in lowest address to highest address order. When using multiple pins, the least significant bit of each byte is assigned to the XSPIn_IO0 pin with each higher order bit on the successively higher numbered XSPIn_IOm pins (n = 0, 1).

Figure 48.4 shows the timing chart for the 1S-1S-1S protocol mode. The XSPIn_IO0 signal is used for output data and the XSPIn_IO1 signal is used for input data (n = 0, 1).

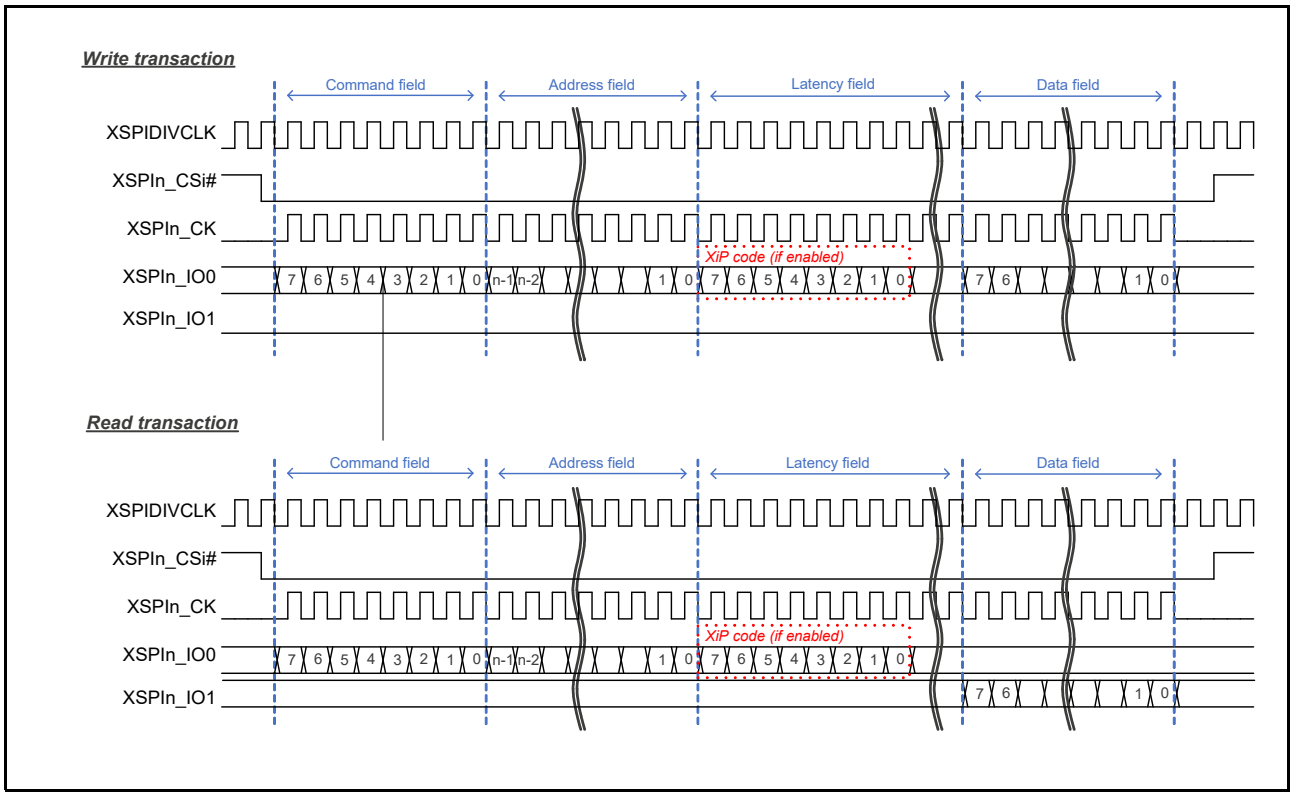


Figure 48.4 1S-1S-1S Timing Chart (n = 0, 1; i = 0, 1)

Figure 48.5 shows the timing chart for 1S-2S-2S protocol mode.

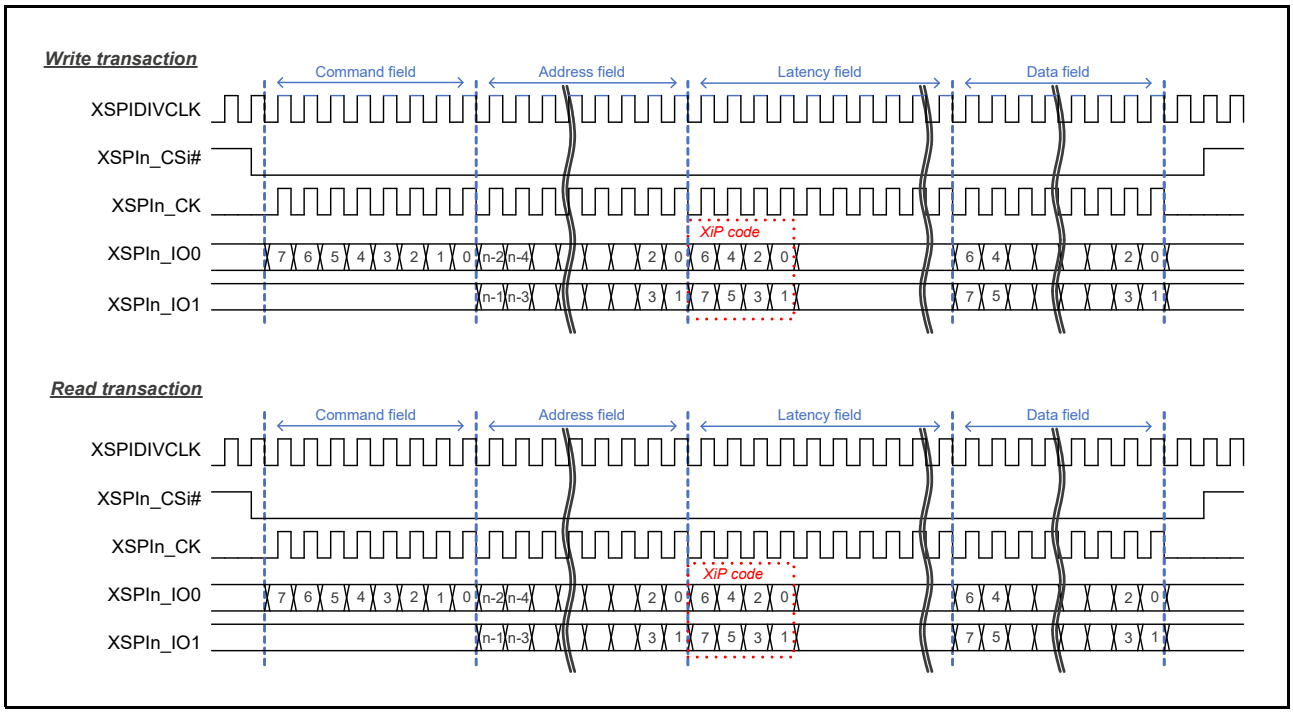


Figure 48.5 1S-2S-2S Timing Chart (n = 0, 1; i = 0, 1)

Figure 48.6 shows the timing chart for 4S-4D-4D protocol mode.

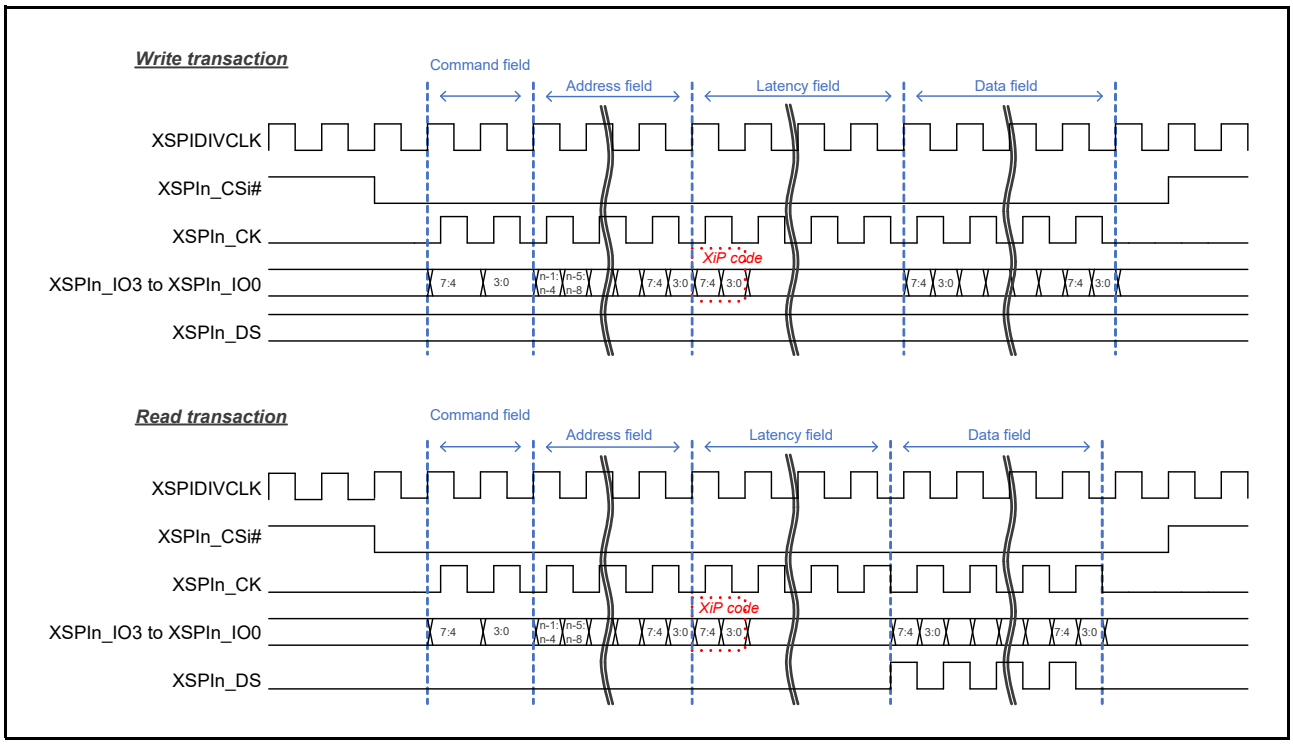


Figure 48.6 4S-4D-4D Timing Chart (n = 0, 1; i = 0, 1)

Figure 48.7 shows the timing chart for 8D-8D-8D profile1.0 protocol mode. In 8D-8D-8D mode, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

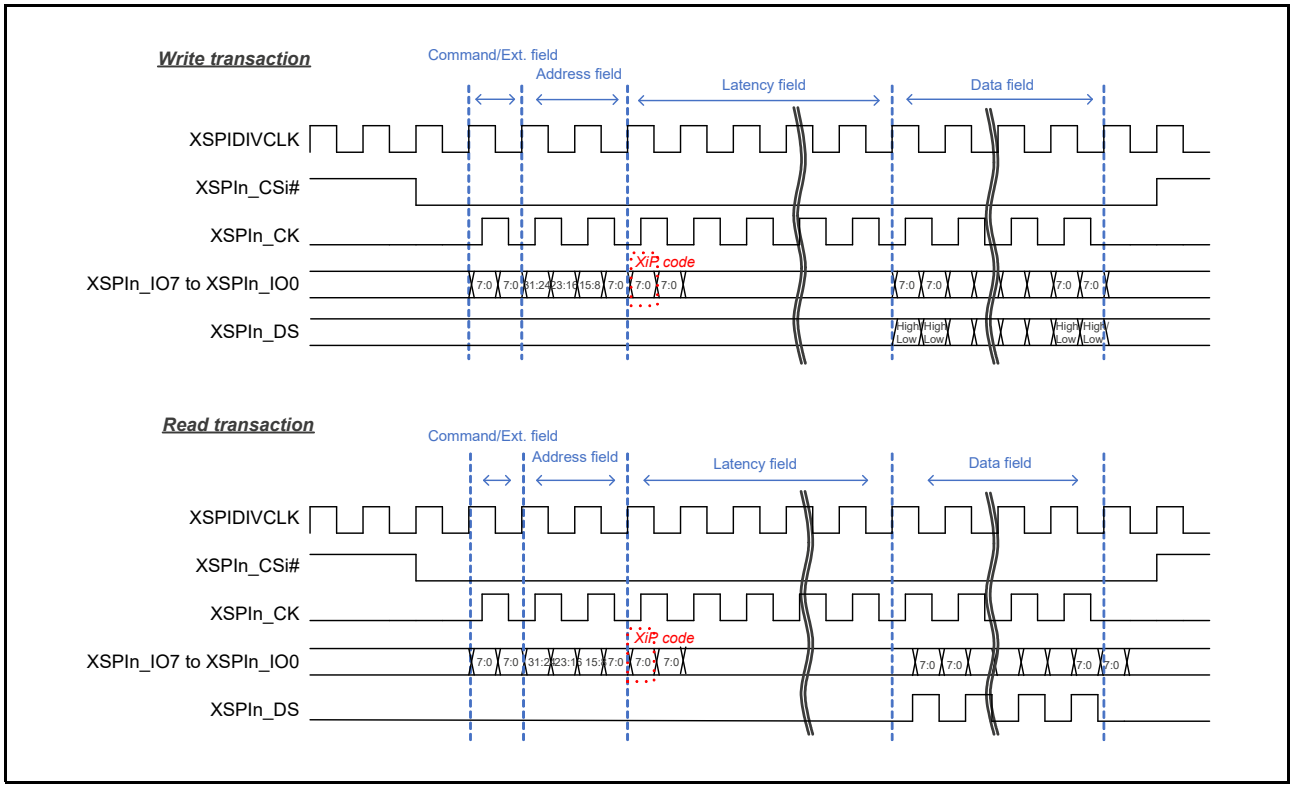


Figure 48.7 8D-8D-8D Profile 1.0 Timing Chart (n = 0, 1; i = 0, 1)

Figure 48.8 shows the timing chart for 8D-8D-8D profile2.0 protocol mode. In 8D-8D-8D mode, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

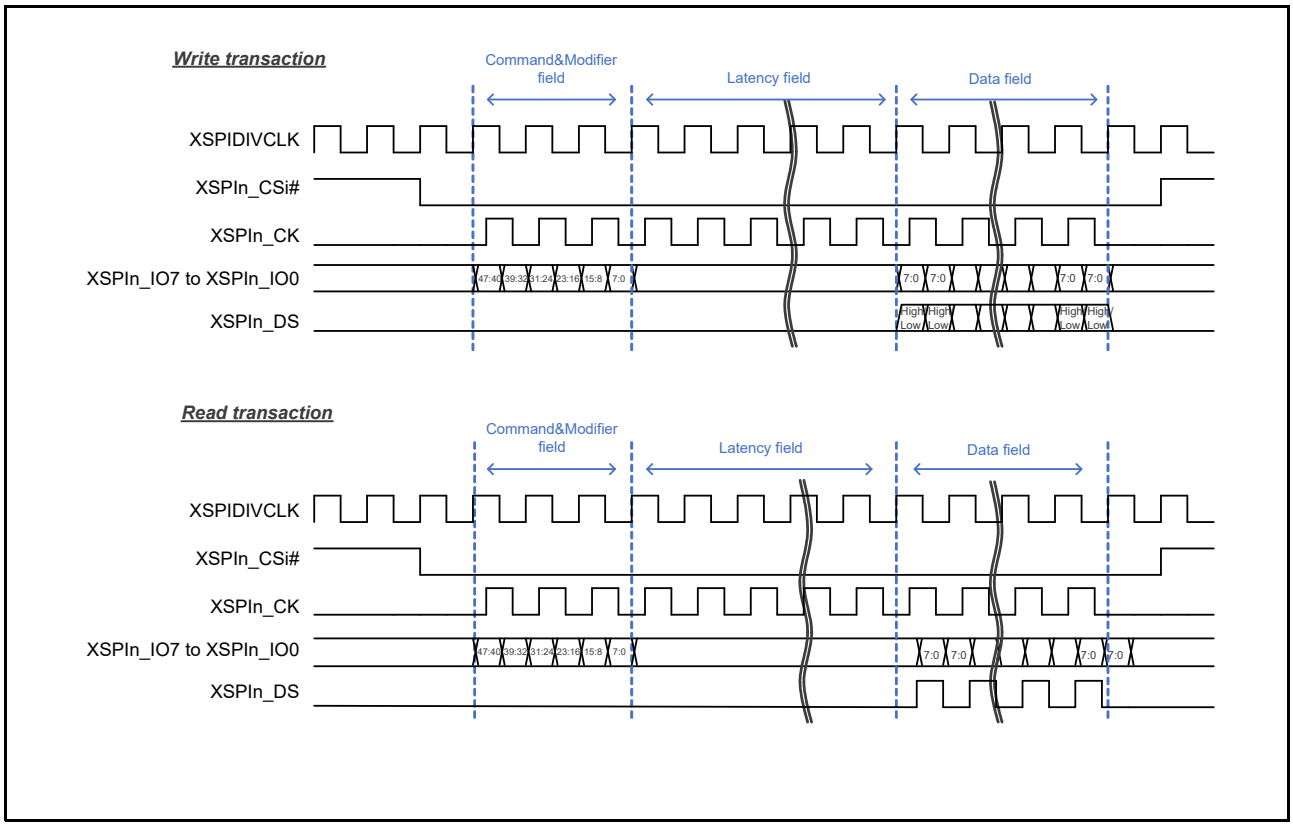


Figure 48.8 8D-8D-8D Profile 2.0 Timing Chart (n = 0, 1; i = 0, 1)

(2) xSPI Frame Interval

The interval between xSPI frames is set by the LIOCFGCSi.CSMIN[3:0] bits (i = 0, 1). The setting value depends on the specifications of the xSPI target device.

(3) Timing Control of xSPI Signals

The XSPI controller supports both SDR transfer and DDR transfer. For SDR transfers, input data is sampled with the XSPIn_CK signal, and for DDR transfers, it is sampled with the XSPIn_DS signal (n = 0, 1). The XSPI controller can statically adjust the output timing/sampling timing of the xSPI interface signals. Table 48.9 summarizes the timing control for the xSPI interface signals.

Table 48.9 Summary of Timing Control for the xSPI Signals (n = 0, 1; i = 0, 1)

Signal	I/O	Mode	Default Operation	Timing Control
XSPIn_CS0, XSPIn_CS1	Output	—	Asserted one cycle before the rising edge of the first XSPIn_CK signal	Assertion timing can be delayed one cycle by the LIOCFGCSi.CSASTEX bit.
			Negated 1.5 cycles after the falling edge of the last XSPIn_CK signal	Negation timing can be delayed one cycle by the LIOCFGCSi.CSNEGEX bit.
XSPIn_CK		SDR transfer	Reference signal	—
		DDR transfer	Reference signal	—
XSPIn_IO0 to XSPIn_IO7		SDR transfer	Output on the falling edge of XSPIDIVCLK	Output timing can be changed to the rising edge of XSPIDIVCLK (0.5 cycles delay) by the LIOCFGCSi.SDRDRV bit.
		DDR transfer	Output on both edges of XSPIDIVCLK	—
XSPIn_IO0 to XSPIn_IO7	Input	SDR transfer	Sampled at the falling edge of the XSPIn_CK signal within the expected data size	Sampling timing can be delayed 0 to 7 cycles by LIOCFGCSi.SDRSMPSFT[3:0] bits. Sampling timing can be changed to the rising edge of the XSPIn_CK signal (0.5 cycle earlier) by the LIOCFGCSi.SDRSMPMD bit.
		DDR transfer	Sampled at both edges of the XSPIn_DS signal within the expected data size	The phase of the XSPIn_DS signal can be shifted 0 to 1 cycle by the WRAPCFG.DSSFTCSi[4:0] bits. Sampling period can be extended from 0 to 7 cycles by the LIOCFGCSi.DDRSMPEX[3:0] bits.

Figure 48.9 shows the default operation and timing control for SDR transfer.

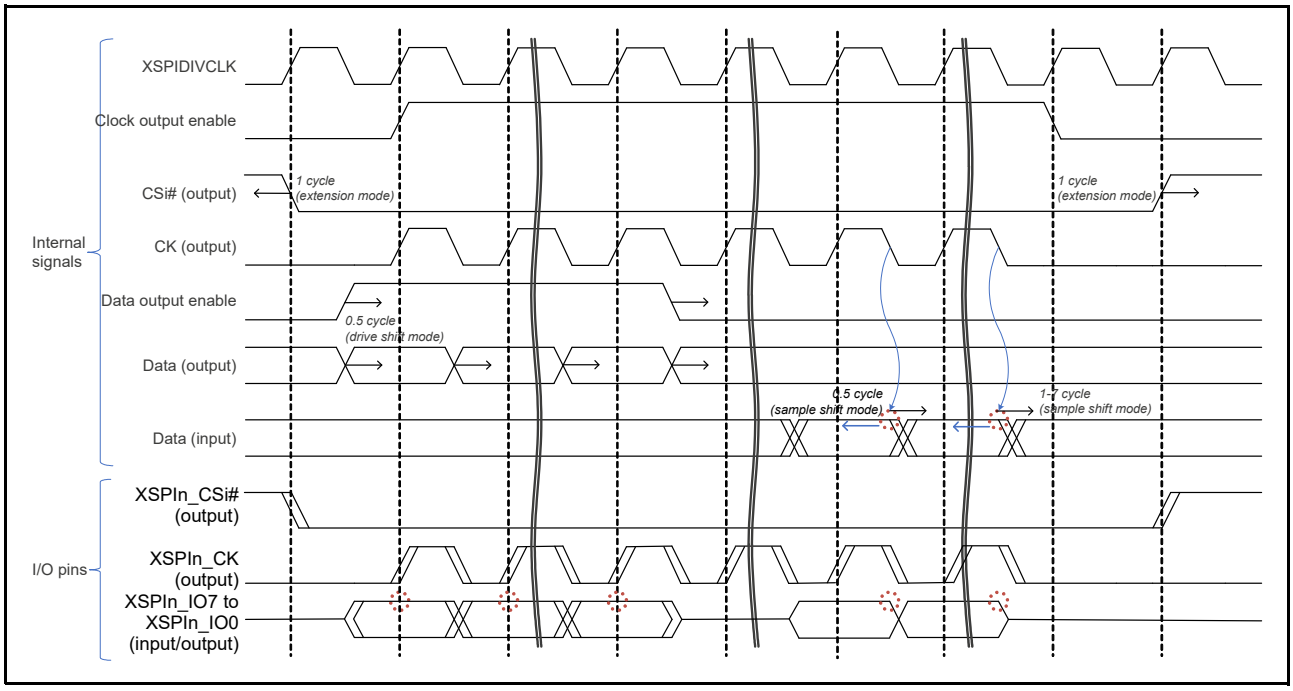


Figure 48.9 Timing Control for SDR Transfer (n = 0, 1; i = 0, 1)

Figure 48.10 shows the default operation and timing control for DDR transfer.

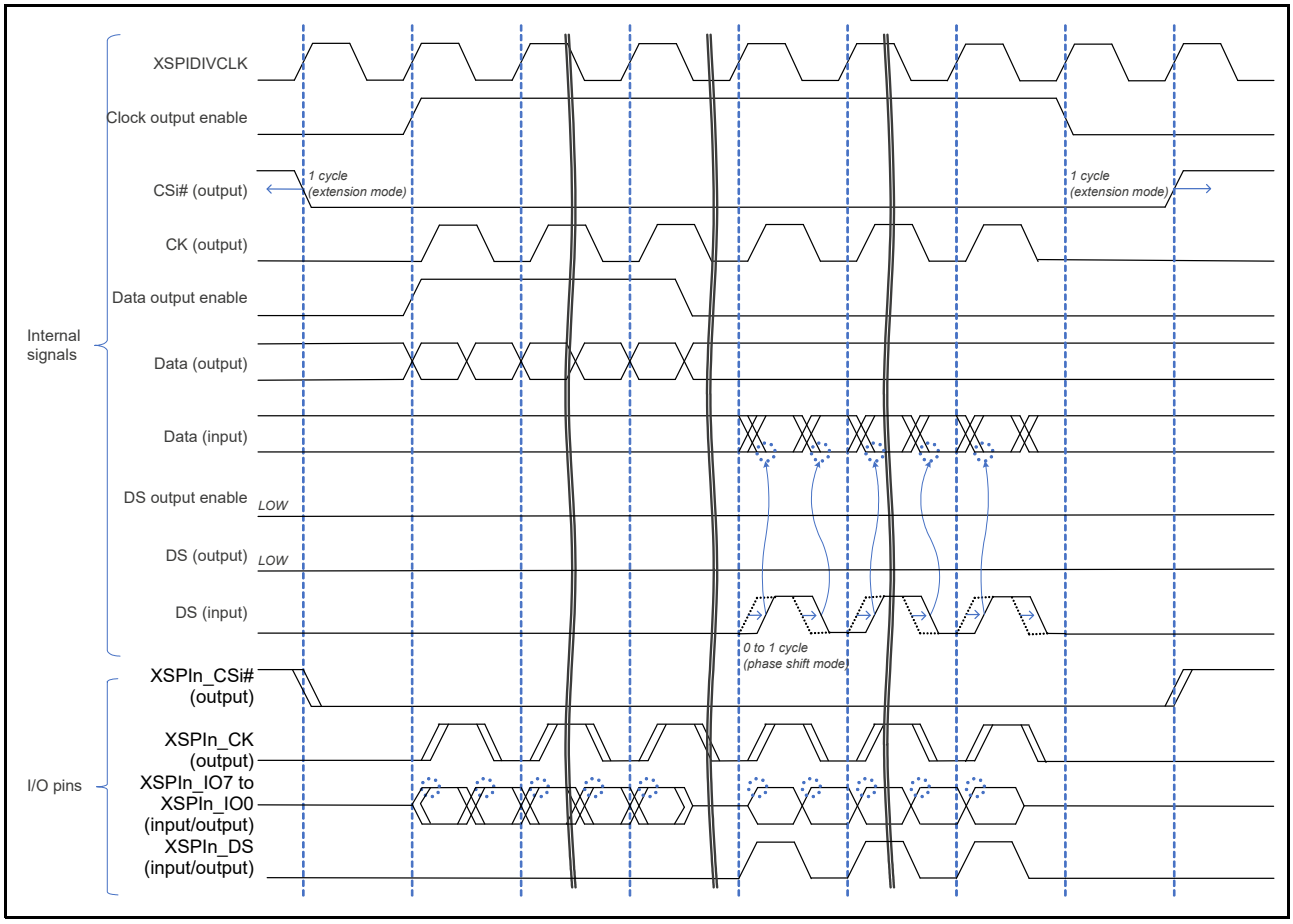


Figure 48.10 Timing Control for DDR Transfer (n = 0, 1; i = 0, 1)

(4) Automatic Calibration

The XSPI controller has a function to automatically adjust the phase shift amount of the DS signal.
This function is enabled by setting the CCCTL0CSi.CAEN bit to 1, and the XSPI controller periodically transmits calibration sequences and automatically adjusts the value of the WRAPCFG.DSSFTCSi[4:0] bits (phase shift amount) (i = 0, 1).
During read transactions in the calibration sequence, if all read comparison results are mismatched, the INTS.CAFAILCSi flag is set to 1 (calibration failed), and the value of the DSSFTCSi[4:0] bits is not updated. If at least one read comparison result matches, the INTS.CASUCCSi flag is set to 1 (calibration success), and the value of the DSSFTCSi[4:0] bits is updated. The success or failure of each phase shift amount can be confirmed by the corresponding flag in the CASTTCSi register.
Figure 48.11 shows the operation of automatic calibration.

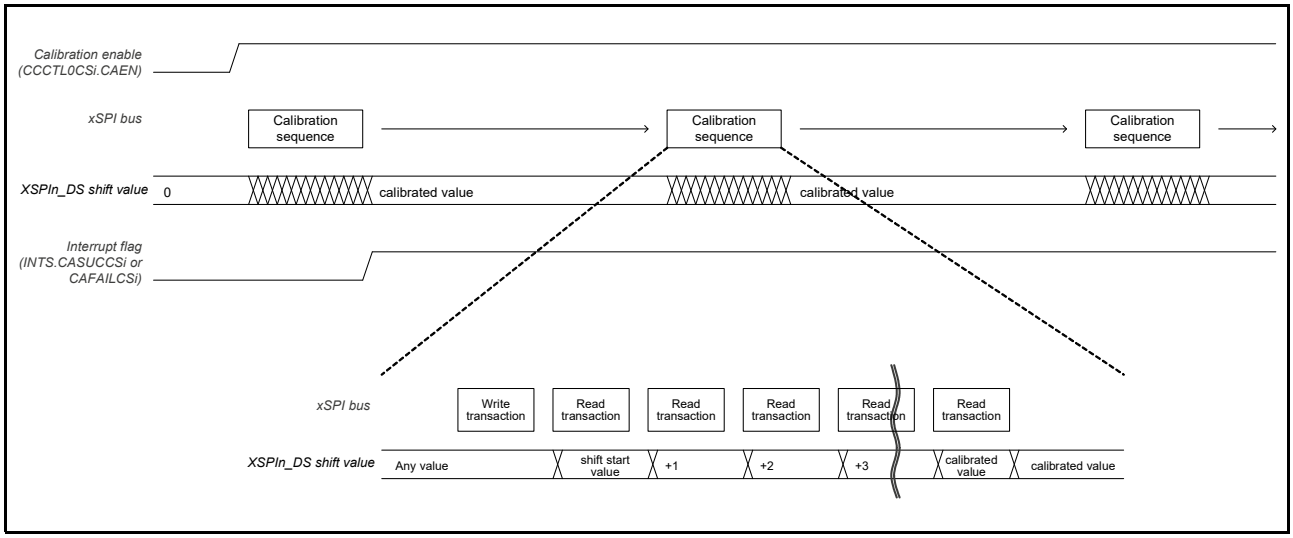


Figure 48.11 Automatic Calibration (n = 0, 1; i = 0, 1)

48.3.1.2 Indirect Access Mode

The indirect access mode has two functional modes: sequential mode and periodic mode.

(1) Sequential Mode

This mode sequentially issues up to four xSPI transactions as configured and requested by software. It can be used to change the mode or to read the status of the xSPI target device. This mode is enabled by setting the CDCTL0.PERMD bit to 0.

Set each transaction in the CDTBUFm, CDABUFm, CDD0BUFm, and CDD1BUFm registers (m = 0 to 3), and set the number of transactions to be issued in the CDCTL0.TRNUM[1:0] bits. After that, when the CDCTL0.TRREQ bit is set to 1, the specified number of transactions are issued sequentially starting from buffer number 0.

Table 48.10 lists the registers and bits used to set the transactions.

Table 48.10 Transaction Settings for Sequential Mode

Transaction	Transaction Type	Command	Command Size	Address	Address Size	Data	Data Size	Latency Cycle
1st Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE[1:0]	CDABUF0	CDTBUF0. ADDSIZE[2:0]	CDD0BUF0, CDD1BUF0	CDTBUF0. DATASIZE[3:0]	CDTBUF0. LATE[4:0]
2nd Transaction	CDTBUF1. TRTYPE	CDTBUF1. CMD[15:0]	CDTBUF1. CMDSIZE[1:0]	CDABUF1	CDTBUF1. ADDSIZE[2:0]	CDD0BUF1, CDD1BUF1	CDTBUF1. DATASIZE[3:0]	CDTBUF1. LATE[4:0]
3rd Transaction	CDTBUF2. TRTYPE	CDTBUF2. CMD[15:0]	CDTBUF2. CMDSIZE[1:0]	CDABUF2	CDTBUF2. ADDSIZE[2:0]	CDD0BUF2, CDD1BUF2	CDTBUF2. DATASIZE[3:0]	CDTBUF2. LATE[4:0]
4th Transaction	CDTBUF3. TRTYPE	CDTBUF3. CMD[15:0]	CDTBUF3. CMDSIZE[1:0]	CDABUF3	CDTBUF3. ADDSIZE[2:0]	CDD0BUF3, CDD1BUF3	CDTBUF3. DATASIZE[3:0]	CDTBUF3. LATE[4:0]

(2) Periodic Mode

This mode periodically issues a single xSPI read transaction configured and requested by software, and compares up to 4 bytes of received data with the expected value. It can be used for status polling of the xSPI target device. This mode is enabled by setting the CDCTL0.PERMD bit to 1.

Set the transaction to be issued in the CDTBUF0, CDABUF0, CDD0BUF0, and CDD1BUF0 registers, the expected value of the received data in the CDCTL1 and CDCTL2 registers, the transaction interval in the CDCTL0.PERITV[4:0] bits, and the repetition count in the PERREP[3:0] bits. After that, when the CDCTL0.TRREQ bit is set to 1, the specified transaction is repeated up to the specified number of times at the specified interval. When the received data matches the expected value, the INTS.CMDCMP flag is set to 1 and transaction issuance stops.

Table 48.11 lists the registers and bits used to set the transactions.

Table 48.11 Transaction Settings for Periodic Mode

Transaction	Transaction Type	Command	Command Size	Address	Address Size	Data	Data Size	Latency Cycle
Read Transaction	CDTBUF0. TRTYPE = 0	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE[1:0]	CDABUF0	CDTBUF0. ADDSIZE[2:0]	CDD0BUF0, CDD1BUF0	CDTBUF0. DATASIZE[3:0]	CDTBUF0. LATE[4:0]

48.3.1.3 Memory-Mapped Mode

This mode automatically converts internal bus accesses to predefined memory regions into xSPI transactions.

(1) Configuration

In memory-mapped mode, the payload for the address and data fields is obtained from the internal bus signals, and information such as the command field and sizes is obtained from pre-configured registers.
Table 48.12 lists the registers and bits used in memory-mapped mode.

Table 48.12 Memory-Mapped Mode Configuration (i = 0, 1)

Internal Bus Transaction	Frame Format	Command	Command Size	Address*1	Address Size	Data	Latency Cycle
Writing to the memory area of target device i	Normal	CMCFG2CSi.WRCMD[15:8]	1 byte	A[x:0]	CMCFG0CSi.ADDSIZE[1:0]	D[x:0]	CMCFG2CSi.WRLATE[4:0]
	8D-8D-8D Profile 1.0	CMCFG2CSi.WRCMD[15:0]	2 bytes				
	8D-8D-8D Profile 2.0 Command Modifier	CMCFG2CSi.WRCMD[15:8]	1 byte	A[27:20], A[19:12], A[11:4], 00h, 00000b+A[3:1]	5 bytes		
	8D-8D-8D Profile 2.0 Extended Command Modifier	CMCFG2CSi.WRCMD[15:13]	3 bits	0b+A[31:28], A[27:20], A[19:12], A[11:4], 00h, 00000b+A[3:1]	45 bits (5 bits + 5 bytes)		
Reading from the memory area of target device i	Normal	CMCFG1CSi.RDCMD[15:8]	1 byte	A[x:0]	CMCFG0CSi.ADDSIZE[1:0]	D[x:0]	CMCFG1CSi.RDLATE[4:0]
	8D-8D-8D Profile 1.0	CMCFG1CSi.RDCMD[15:0]	2 bytes				
	8D-8D-8D Profile 2.0 Command Modifier	CMCFG1CSi.RDCMD[15:8]	1 byte	A[27:20], A[19:12], A[11:4], 00h, 00000b+A[3:1]	5 bytes		
	8D-8D-8D Profile 2.0 Extended Command Modifier	CMCFG1CSi.RDCMD[15:13]	3 bits	0b+A[31:28], A[27:20], A[19:12], A[11:4], 00h, 00000b+A[3:1]	45 bits (5 bits + 5 bytes)		

Note 1. The most significant byte of the address (A[31:24]) can be replaced using the CMCFG0CSi.ADDRPEN[7:0] and CMCFG0CSi.ADDRPCD[7:0] bits.

(2) Write Access

When accepting a write access from the internal bus to the XSPI area, the XSPI controller stores all payload data in an internal bridge buffer and then issues a write transaction to the XSPI target.
Figure 48.12 shows an overview of the write access operation.

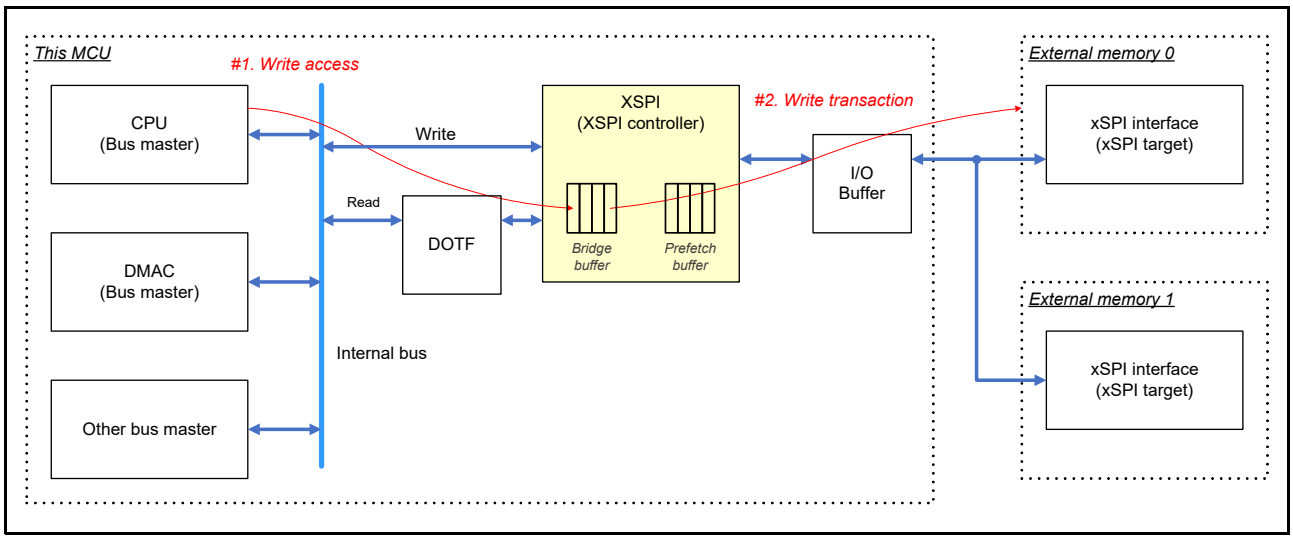


Figure 48.12 Write Access to Memory Area

The operation of xSPI bus changes depending on internal bus burst type. When the burst type is single type or incremental type, one internal bus transaction triggers for one xSPI frame. When the burst type is wrap type and the CMCFG0CSi.WPBSTMD is 0, one internal bus transaction triggers two xSPI frames (i = 0, 1).
Figure 48.13 shows the relationship between AXI and xSPI frame.

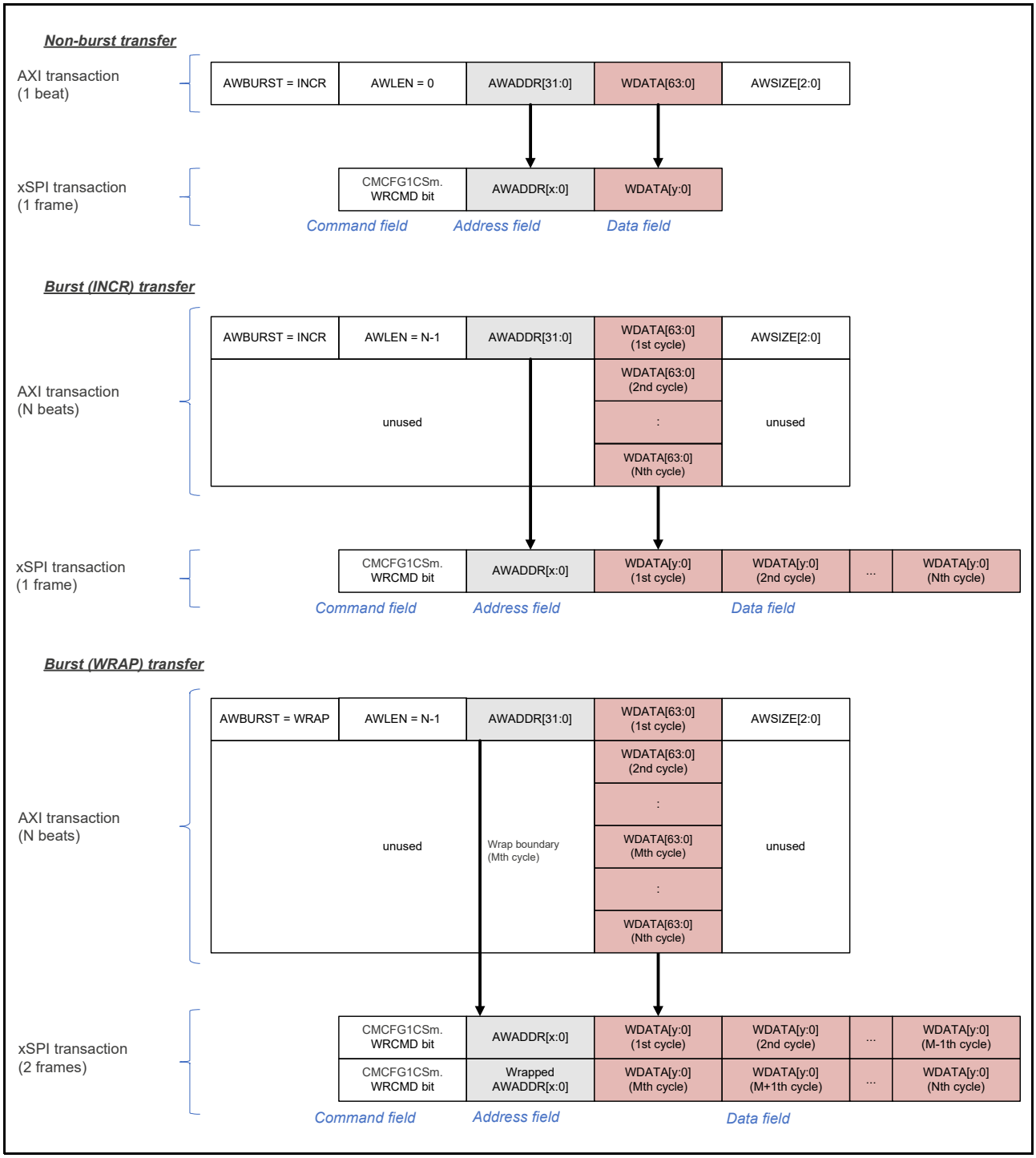


Figure 48.13 xSPI Frame Format in Write Access (Normal Format) (i = 0, 1)

(3) Write Data Combination Function

To achieve high throughput on the xSPI bus during internal bus write accesses to the XSPI area, the XSPI controller incorporates a function to combine write data. When the BMCFGCH0.MWRCOMB bit is set to 1, this function is enabled and the XSPI controller transmits xSPI frames at the selected size while the addresses are incrementally consecutive*1.

Note 1. The access which comply all below condition is considered as incremental address.
- Transaction type is INCR.
- The starting address of the access is consecutive with the last write address.
  - The starting position of WSTRB access is treated as the starting address
  - The last WSTRB of the previous access is treated as the last write address

When any of the following conditions is detected, the XSPI controller transmits pending data to the xSPI bus even if the target size set in the BMCFGCH0.MWRSIZE[7:0] bits has not been reached.

- Non-incremental address is detected.
- Different burst type is detected.
- Read transaction is detected.
- Access for different target is detected.
- Memory Write Data Push bit (BMCTL1.MWRPUSHCH0) is set to 1.

This function is useful when the target device requests large amount of data at once. In this case, the internal bus master must continue to provide fixed-size data blocks at incremental addresses.
Figure 48.14 shows the operation when enable the combination function.

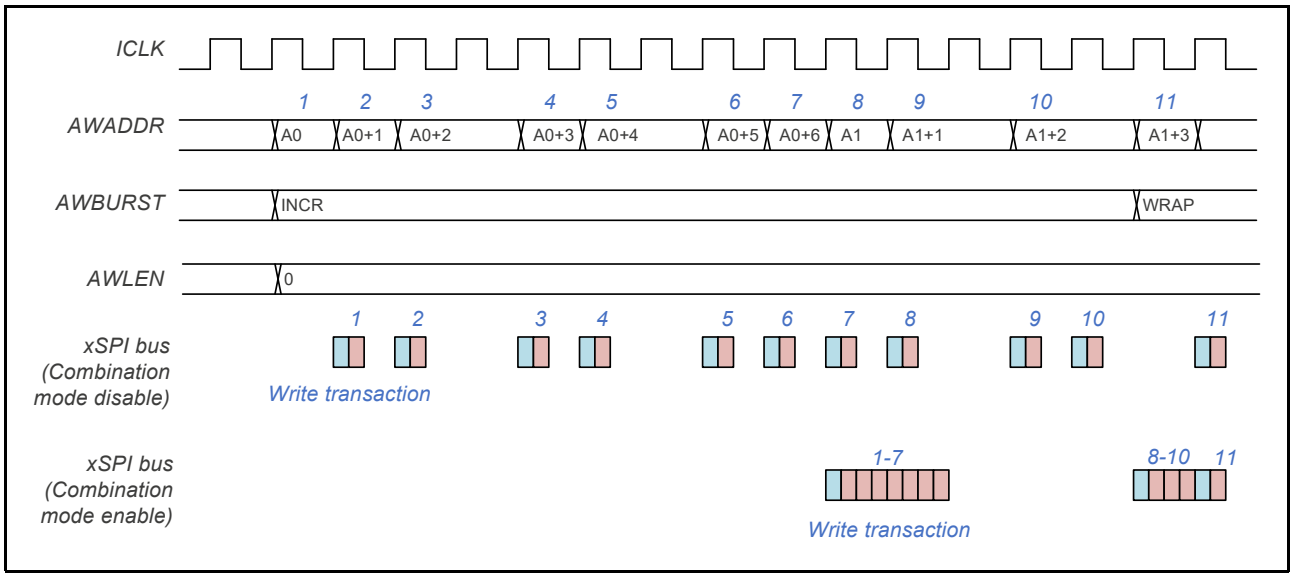


Figure 48.14 Write Data Combination Function

Figure 48.15 shows an example of write data combination in AXI access.

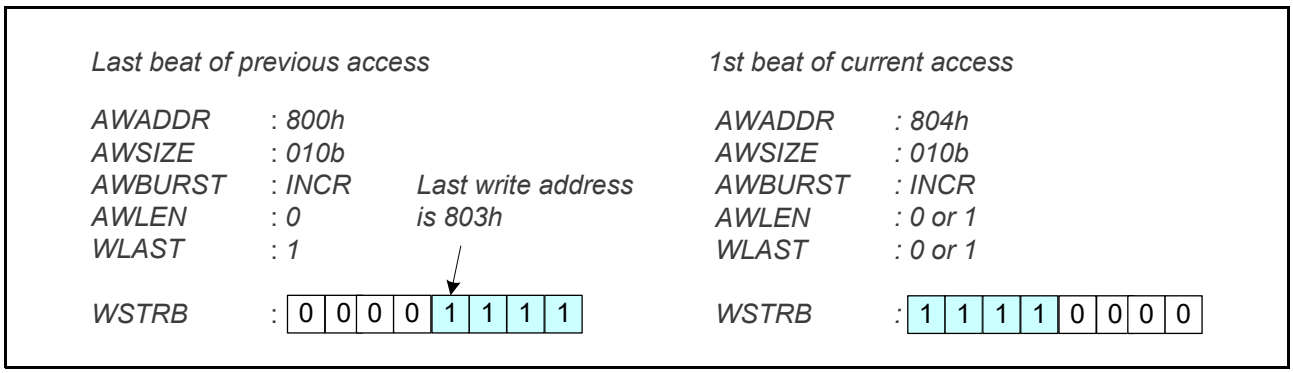


Figure 48.15 Example of Write Data Combination in AXI Access

(4) Read Access

When a read access to the XSPI area from the internal bus is detected, the XSPI controller immediately issues a read transaction to the xSPI target.

Figure 48.16 shows an overview of the read access operation.

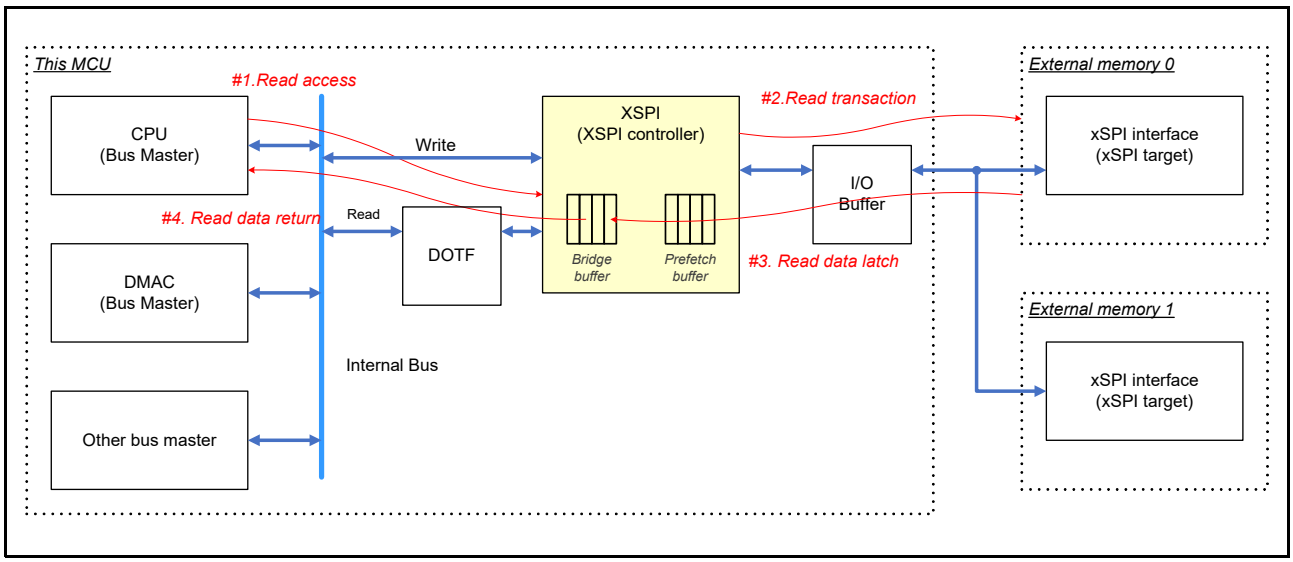


Figure 48.16 Read Access to Memory Area

The operation of xSPI bus changes depending on burst type. When the type is single or increment type, one internal bus's read transaction triggers one xSPI frame. When the type is wrap type and the CMCFG0CSi.WPBSTMD is 0, one system bus read transactions triggers two xSPI frame (i = 0, 1).
Figure 48.17 shows the relationship between AXI and xSPI frame.

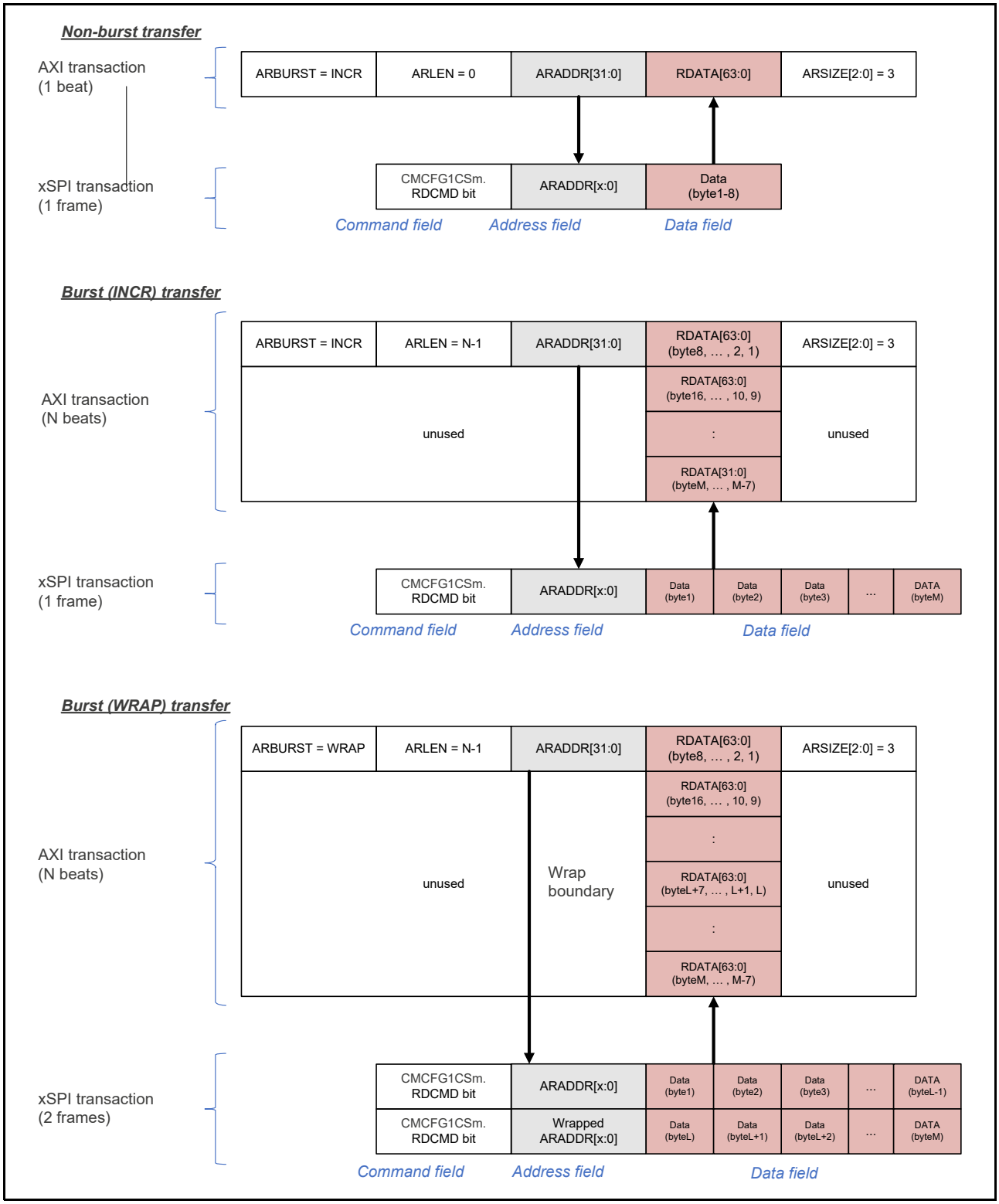


Figure 48.17 xSPI Frame Format in Read Access (Normal Format) (i = 0, 1)

(5) Prefetch Function

The XSPI controller has a function that prefetches read data to reduce latency for read accesses to the XSPI area from the internal bus. Setting the BMCFGCHi.PREEN bit to 1 enables this function, and the XSPI controller continuously reads data from subsequent addresses from the xSPI target and stores it in the internal prefetch buffer (i = 0, 1). When the next read access from the internal bus is detected, the XSPI controller searches the prefetch buffer. If the desired read data is found in the prefetch buffer, the XSPI controller returns the data from the prefetch buffer. If not found, the prefetch buffer is cleared and a new read transaction is issued to the xSPI target.

This function is effective in applications that read adjacent areas consecutively. It is ineffective if the addresses to be read are not consecutive.

Figure 48.18 shows the operation when enable the prefetch function.

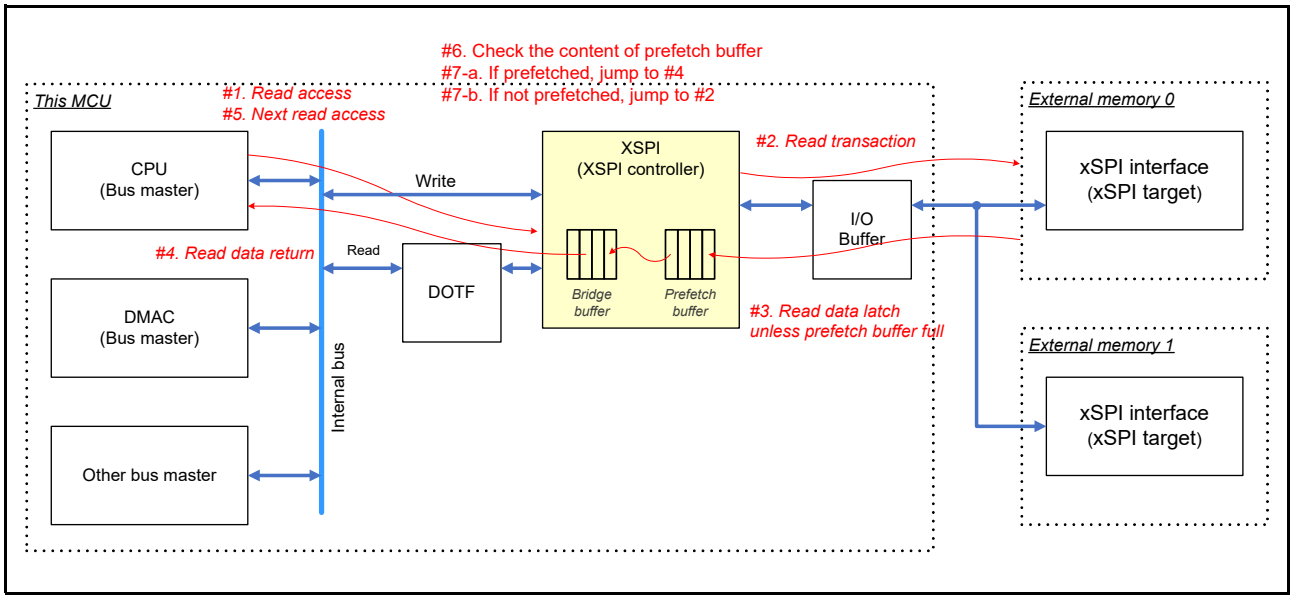


Figure 48.18 Read Access to Memory Area with Prefetch Function Enabled

(6) XIP Mode

Some target devices can reduce latency by omitting the command phase. To support such devices, the XSPI controller supports XIP mode.
When the CMCTLCH0.XIPEN bit is set to 1 (XIP mode enabled), the XSPI controller inserts the XIP enter code specified by the CMCTLCH0.XIPENCODE[7:0] bits into the latency field. When the XIPEN bit is set to 0 (XIP mode disabled), the XSPI controller inserts the XIP exit code specified by the CMCTLCH0.XIPEXCODE[7:0] bits into the latency field. This function is only available in memory-mapped mode.
The XIPEN bit also becomes 0 and XIP mode is disabled when the XSPI controller sends an XIP disable pattern by setting the LPCTL0.PATREQ bit to 1. It is not possible to disable XIP mode on only one channel using the XIP disable pattern.

Note: If there are not enough latency cycles for the XIP code to be inserted, the XIP code cannot be inserted.

48.3.1.4 Signal Pattern Control

This xSPI controller has the function to transmit three types of signal patterns that are not in xSPI frame format. Transmission of signal patterns is triggered by setting the signal pattern generation request bits (LPCTL0.PATREQ, LPCTL1.PATREQ[1:0]).

(1) XIP Disable Pattern

The XIP disable pattern is a signal pattern with the length and level set by the XD1LEN[4:0], XD1VAL, XD2LEN[4:0], and XD2VAL bits in the LPCTL0 register. The XSPIn_CK and XSPIn_IO7 to XSPIn_IO0 signals are used, and the number of output pins is specified by the LPCTL0.XDPIN[1:0] bits (n = 0, 1).
The XIP disable pattern can be used to disable XIP mode in Legacy SPI.
Figure 48.19 shows the timing chart for the XIP disable pattern.

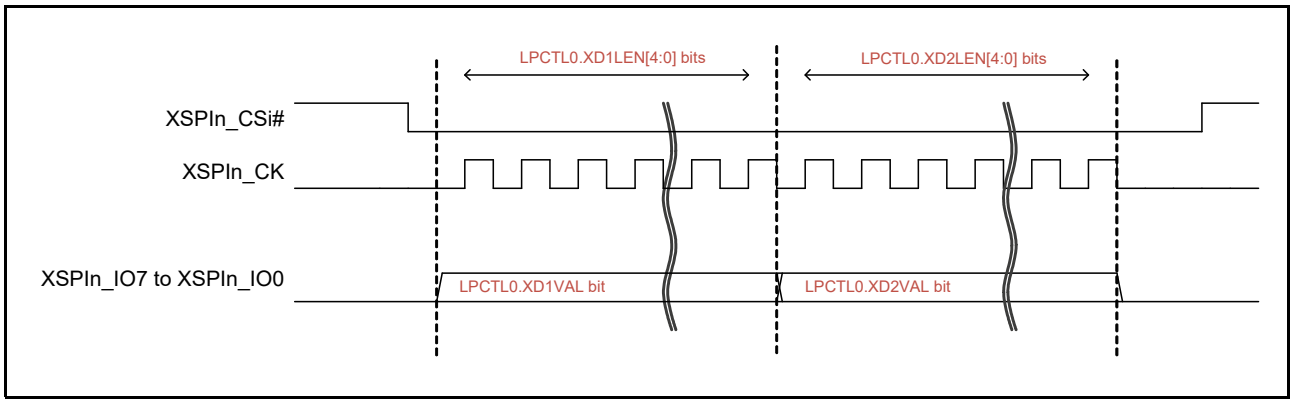


Figure 48.19 XIP Disable Pattern (n = 0, 1; i = 0, 1)

(2) Reset Pattern

The Reset pattern is a signal pattern specified in JESD252.
Figure 48.20 shows the timing chart for the reset pattern.
The low and high width of the XSPIn_CSi# signal is configured with the LPCTL1.RSTWID[2:0] bits (n = 0, 1; i = 0, 1).
The xSPI target samples data on the rising edge of the XSPIn_CSi# signal. The setup time for data output is configured with the LPCTL1.RSTSU[2:0] bits. The setup time must be shorter than the reset pattern width.

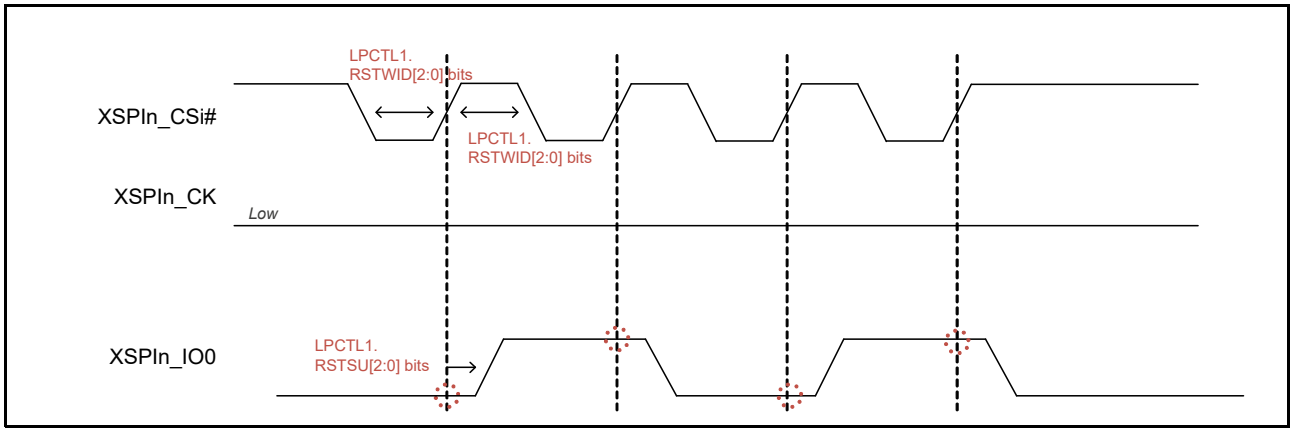


Figure 48.20 Reset Pattern (n = 0, 1; i = 0, 1)

(3) CS Only Pattern

CS only pattern asserts the XSPIn_CSi# pin with the pulse width configured with the LPCTL1.RSTWID[2:0] bits (n = 0, 1; i = 0, 1). It can be used to resume from deep power-down state.

Figure 48.21 shows the timing chart for the CS only pattern.

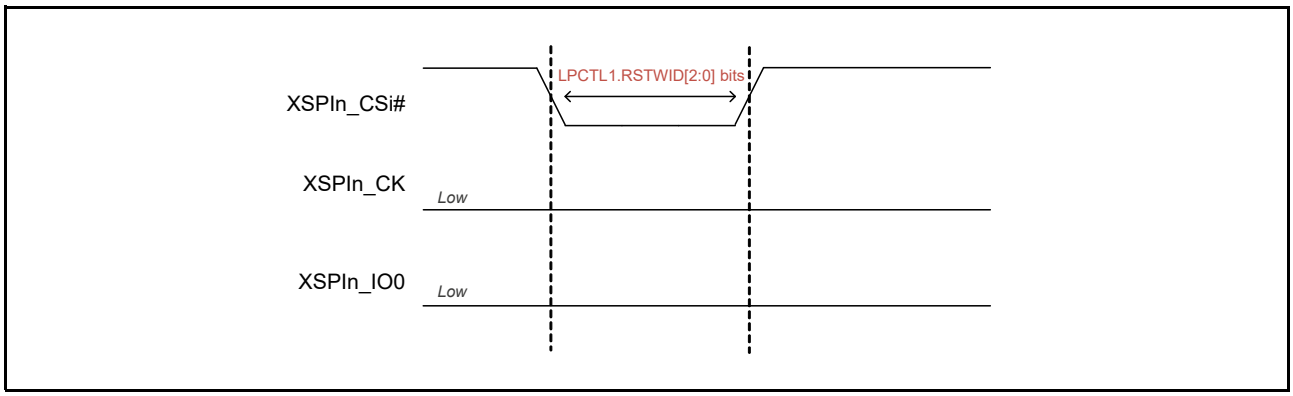


Figure 48.21 CS Only Pattern (n = 0, 1; i = 0, 1)

48.3.1.5 Error Detection

The xSPI controller can detect the errors shown in Table 48.13.

Table 48.13 Error List (n = 0, 1; i = 0, 1)

Name	Error Source	Detection Flag	Remarks
Calibration failed	The read data did not match the expected value during automatic calibration.	INTS.CAFAILCSi	Unexpected data is written to the xSPI target.
Internal bus error	An error occurred on the internal bus during memory-mapped mode.	INTS.BUSERRCH0	Due to a fatal error, the XSPI controller must be reset.
ECC error detection	The XSPIn_ECSi# pin changed from high to low (valid only if the xSPI target has ECC detection capability).	INTS.ECSCSi	Just a notification of an error event from the xSPI target
DS timeout	The XSPIn_DS signal did not toggle during a read transaction that uses the XSPIn_DS signal.	INTS.DSTOCSi	Due to a fatal error, both the XSPI controller and the xSPI target must be reset.
Periodic transaction timeout	The read value did not match the expected value in periodic mode of indirect access mode.	INTS.PERTO	

48.3.1.6 Interrupt Sources

This XSPI controller has two types of interrupt sources: CMP interrupt and ERR interrupt as shown in Table 48.14. Each interrupt can be monitored using the INTS register, and its enable/disable setting can be changed using the INTE register.

Note: Even if the corresponding bit in the INTE register is set to 1 after the flag in the INTS register is set to 1, the interrupt signal will not be output.

Table 48.14 XSPI Controller Interrupt Sources

Name	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Interrupt Clear Bit	DTC/DMAC Triggerable
CMP*1	Target 1 calibration success	INTS.CASUCCS1	INTE.CASUCCS1E	INTC.CASUCCS1C	No
	Target 0 calibration success	INTS.CASUCCS0	INTE.CASUCCS0E	INTC.CASUCCS0C	
	Signal pattern generation completion	INTS.PATCMP	INTE.PATCMPE	INTC.PATCMPC	
	Command issuance completion	INTS.CMDCMP	INTE.CMDCMPE	INTC.CMDCMPC	
ERR*1	Target 1 calibration failure	INTS.CAFAILCS1	INTE.CAFAILCS1E	INTC.CAFAILCS1C	No
	Target 0 calibration failure	INTS.CAFAILCS0	INTE.CAFAILCS0E	INTC.CAFAILCS0C	
	Internal bus error	INTS.BUSERRCH0	INTE.BUSERRCH0E	INTC.BUSERRCH0C	
	Target 1 interrupt detection	INTS.INTCS1	INTE.INTCS1E	INTC.INTCS1C	
	Target 0 interrupt detection	INTS.INTCS0	INTE.INTCS0E	INTC.INTCS0C	
	Target 1 ECC error detection	INTS.ECSCS1	INTE.ECSCS1E	INTC.ECSCS1C	
	Target 0 ECC error detection	INTS.ECSCS0	INTE.ECSCS0E	INTC.ECSCS0C	
	Target 1 DS timeout	INTS.DSTOCS1	INTE.DSTOCS1E	INTC.DSTOCS1C	
	Target 0 DS timeout	INTS.DSTOCS0	INTE.DSTOCS0E	INTC.DSTOCS0C	
	Periodic transaction timeout	INTS.PERTO	INTE.PERTOE	INTC.PERTOC	

Note 1. The interrupts from the XSPI target are each logically ORed and output to the interrupt controller (ICU).

48.3.1.7 Operation Flows

(1) Initial Configuration Flow

Figure 48.22 shows the initial configuration flow.

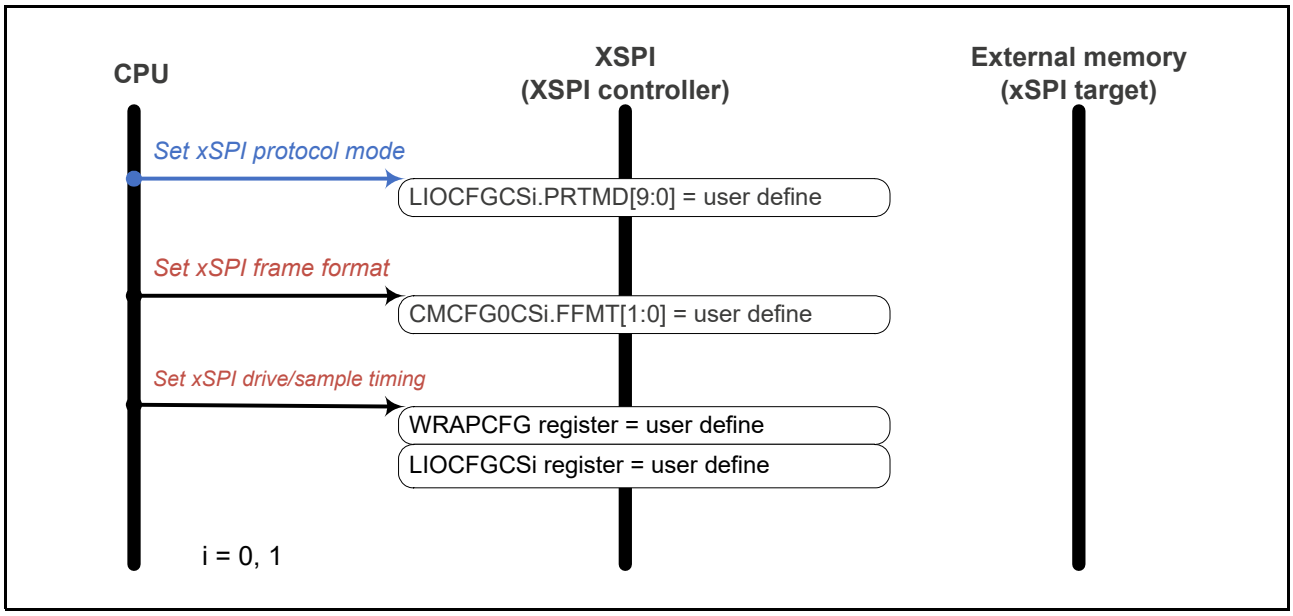


Figure 48.22 Initial Configuration Flow

(2) Communication Stop Flow

Figure 48.23 shows communication stop flow.

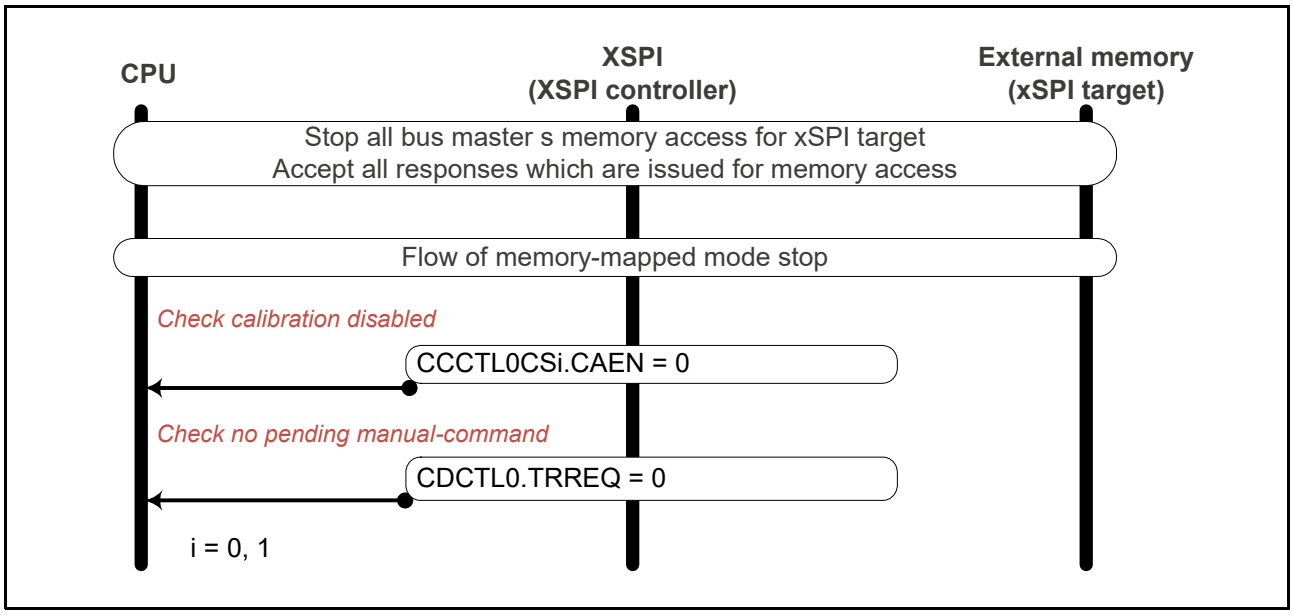


Figure 48.23 Communication Stop Flow

Note: When reconfiguring any of the configuration registers, all communication with the xSPI target must be stopped to avoid conflicts between register settings and memory accesses. This means automatic calibration is disabled, there are no pending commands in indirect access mode, and there are no accesses in memory-mapped mode.

(3) Automatic Calibration Flow

Figure 48.24 shows automatic calibration flow.

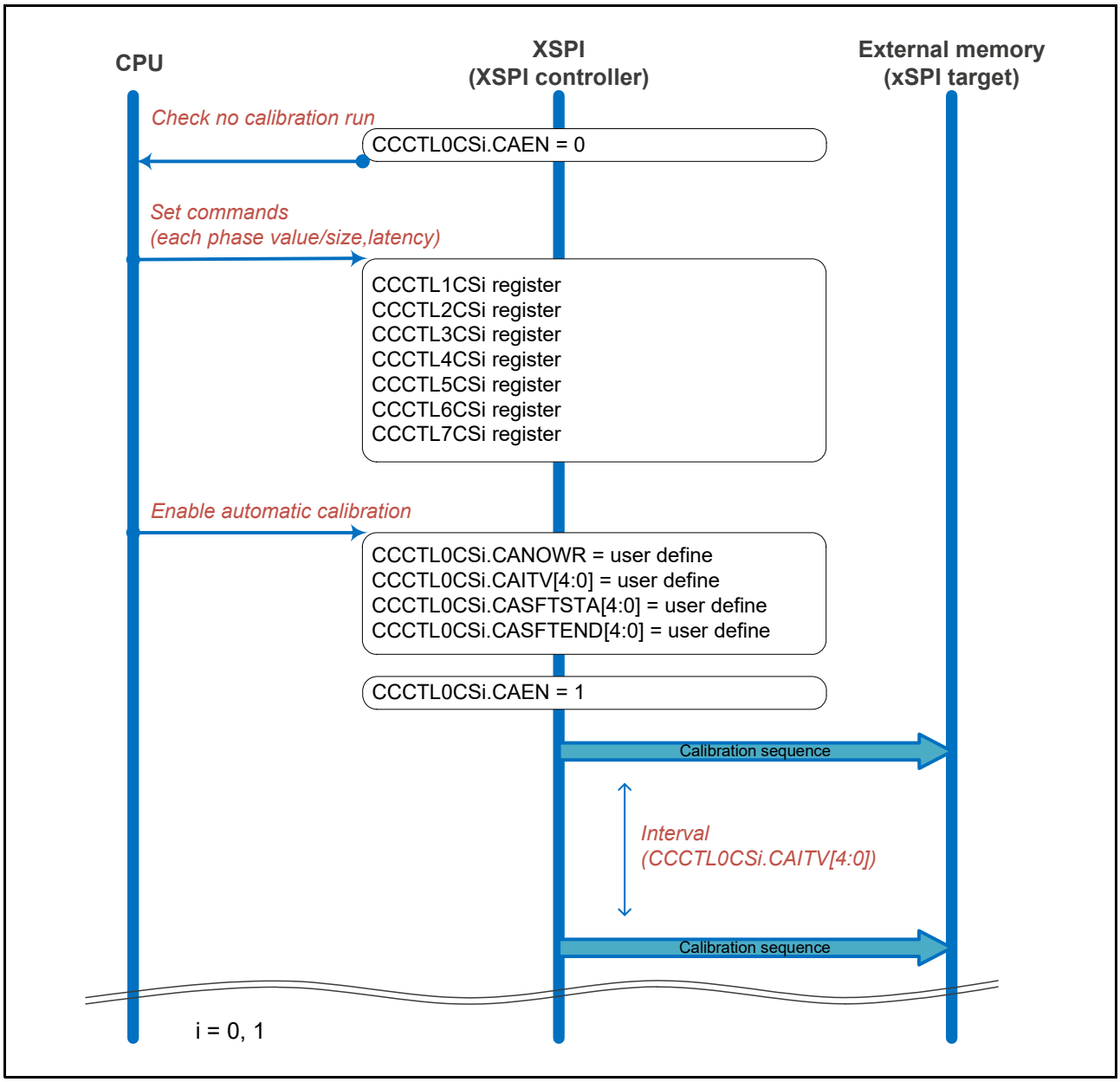


Figure 48.24 Automatic Calibration Flow

(4) Command Issuance Flow in Indirect Access Mode

Figure 48.25 shows the flow of command issuance procedure in sequential mode.

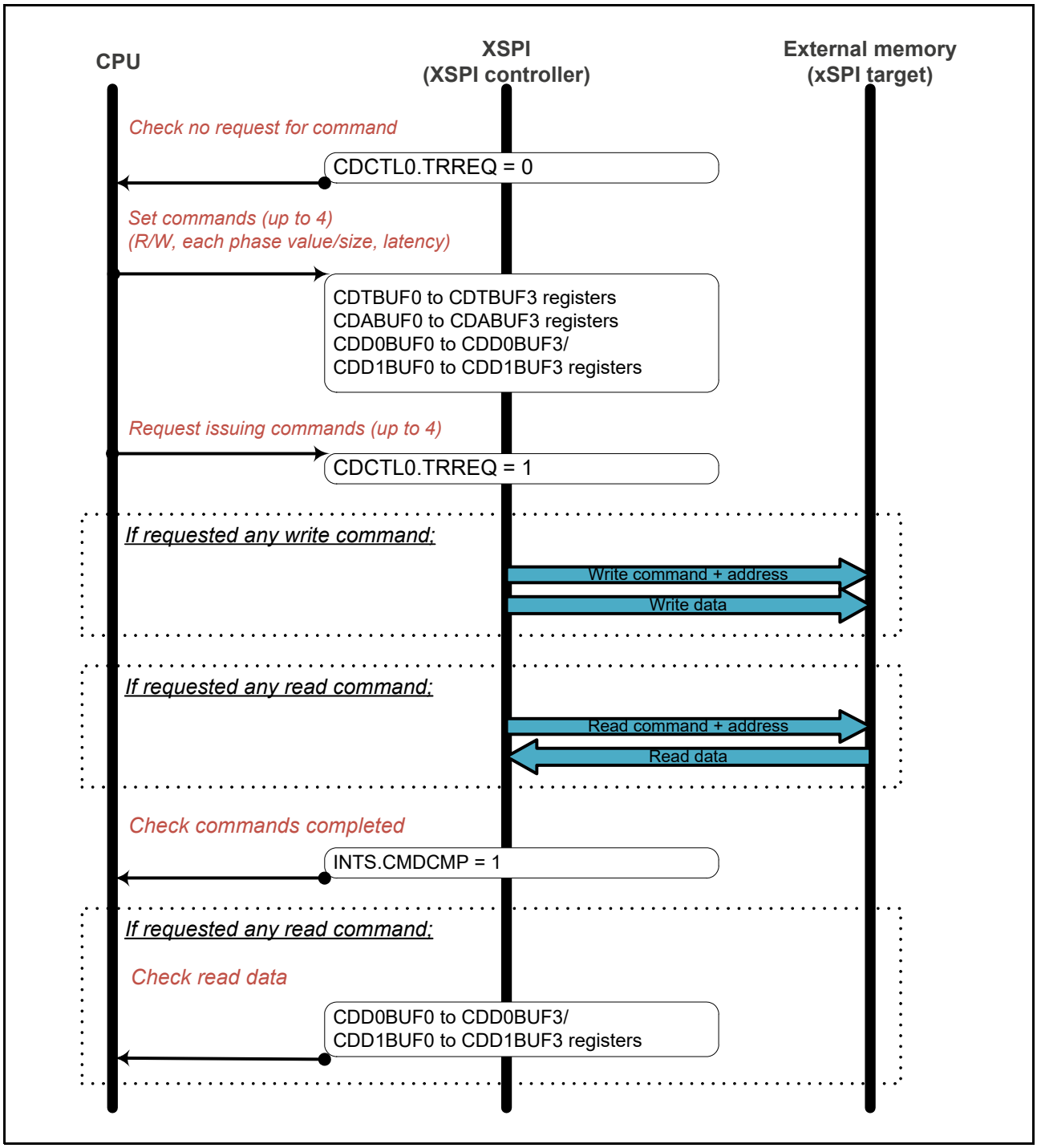


Figure 48.25 Command Issuance Procedure in Indirect Access Mode (Sequential Mode)

Figure 48.26 shows the flow of command issuance procedure in periodic mode.

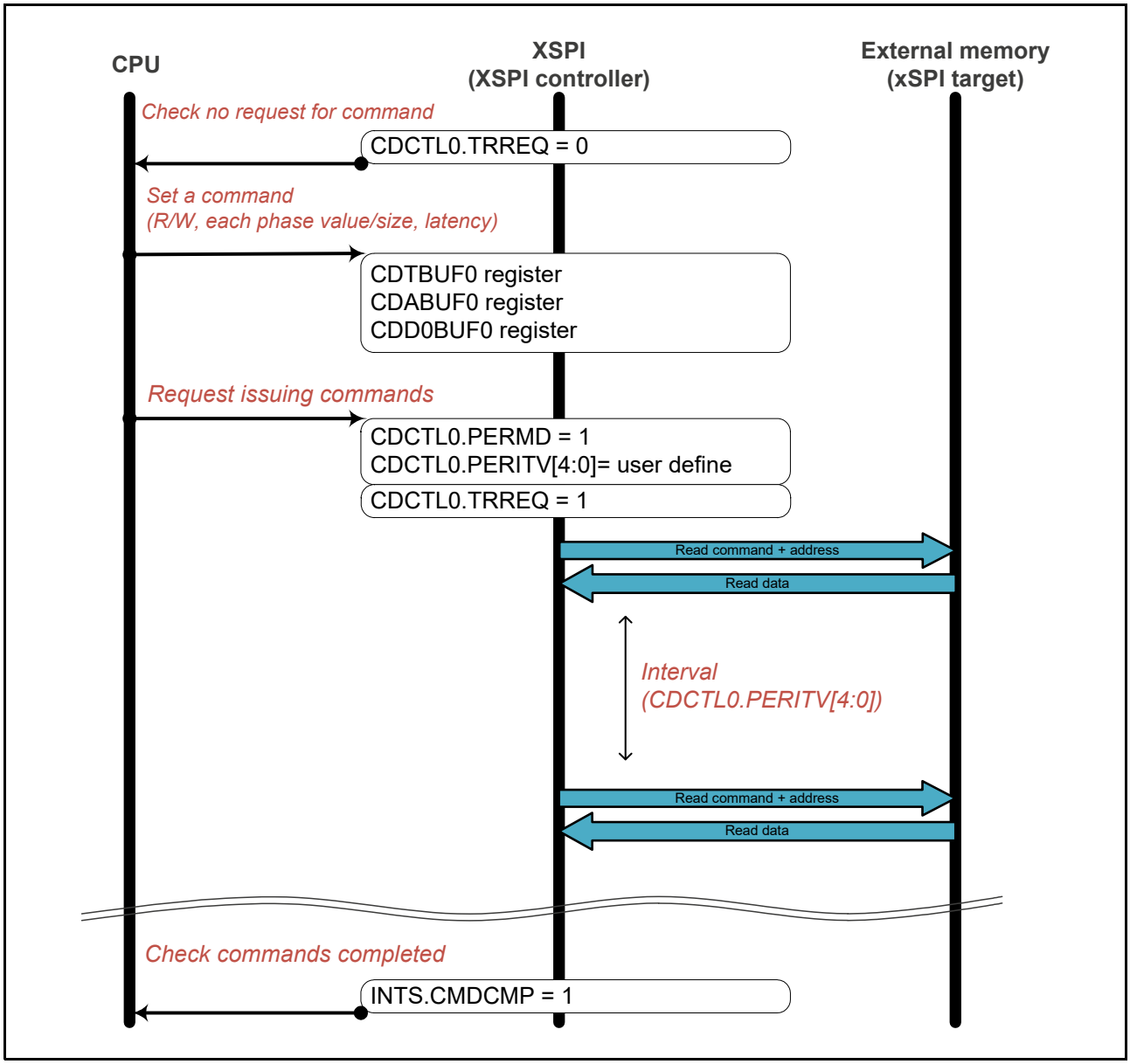


Figure 48.26 Command Issuance Procedure in Indirect Access Mode (Periodic Mode)

(5) Execution Flow in Memory-Mapped Mode

Figure 48.27 shows the execution flow in memory-mapped mode.

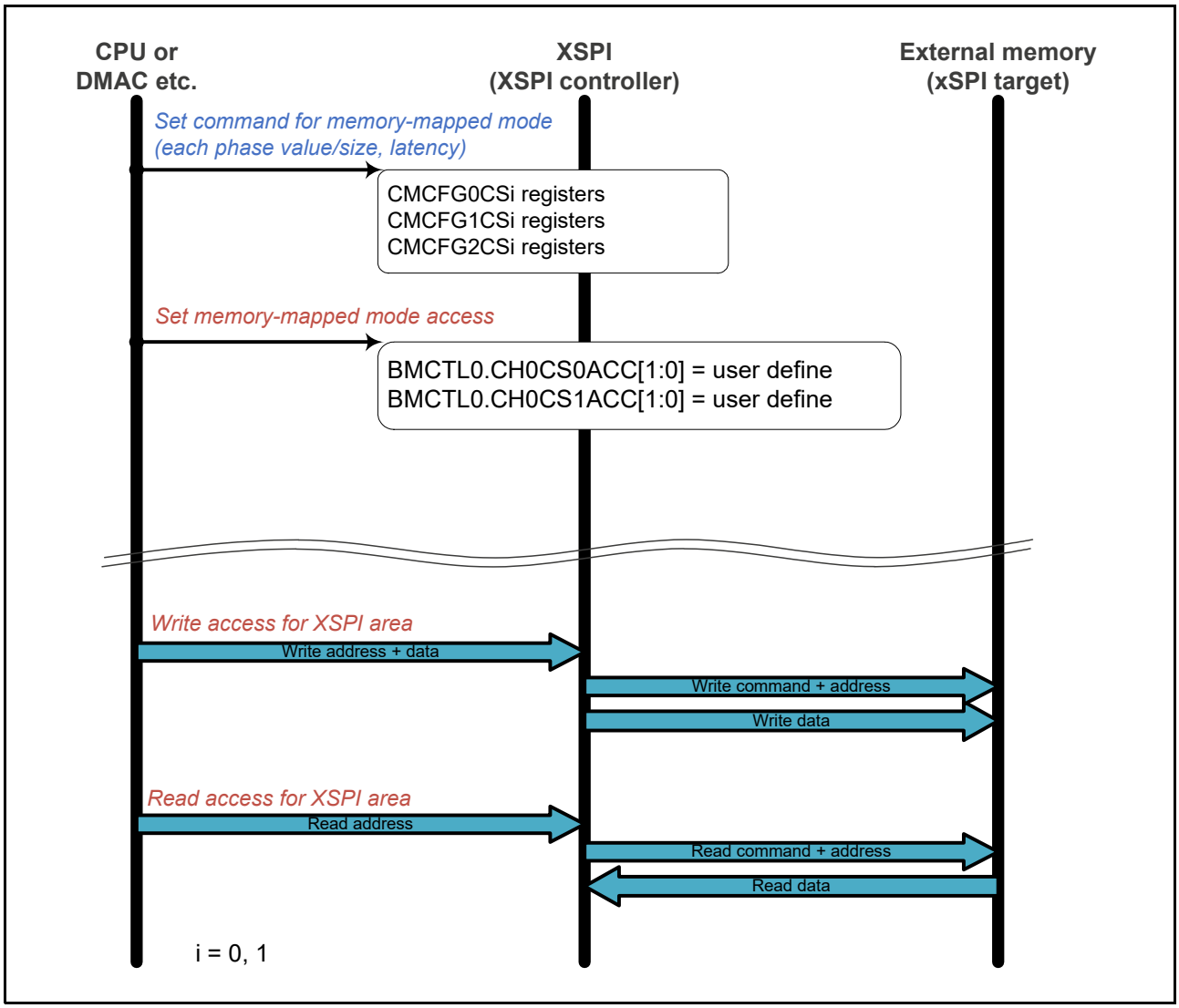


Figure 48.27 Memory-Mapped Mode Execution Flow

(6) Stop Flow for Memory-Mapped Mode

Figure 48.28 shows the stop flow for memory-mapped mode.

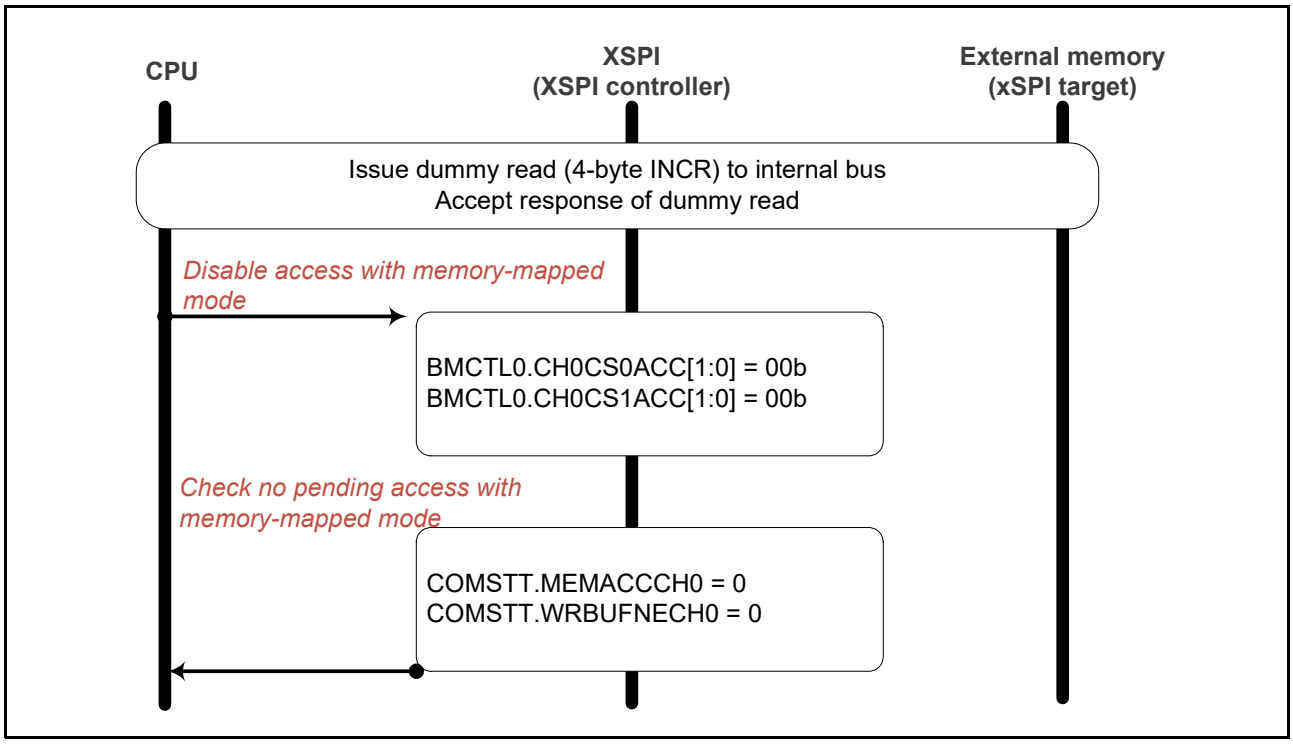


Figure 48.28 Memory-Mapped Mode Stop Flow

(7) Signal Pattern Generation Request Flow

Figure 48.29 shows the flow for requesting signal pattern generation. Before requesting a pattern, complete or cancel any commands that are being executed.

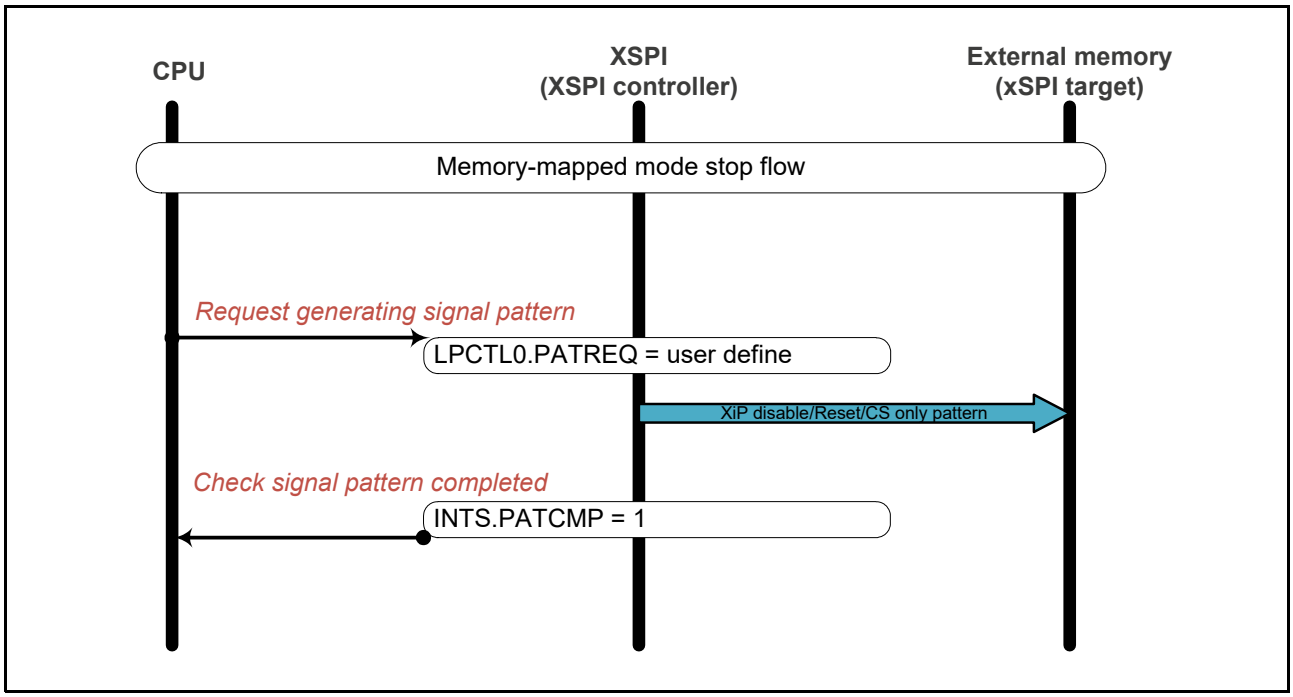


Figure 48.29 Signal Pattern Generation Request Flow

(8) XIP Mode Enable/Disable Flow

Figure 48.30 shows the flow to enable/disable XIP mode.

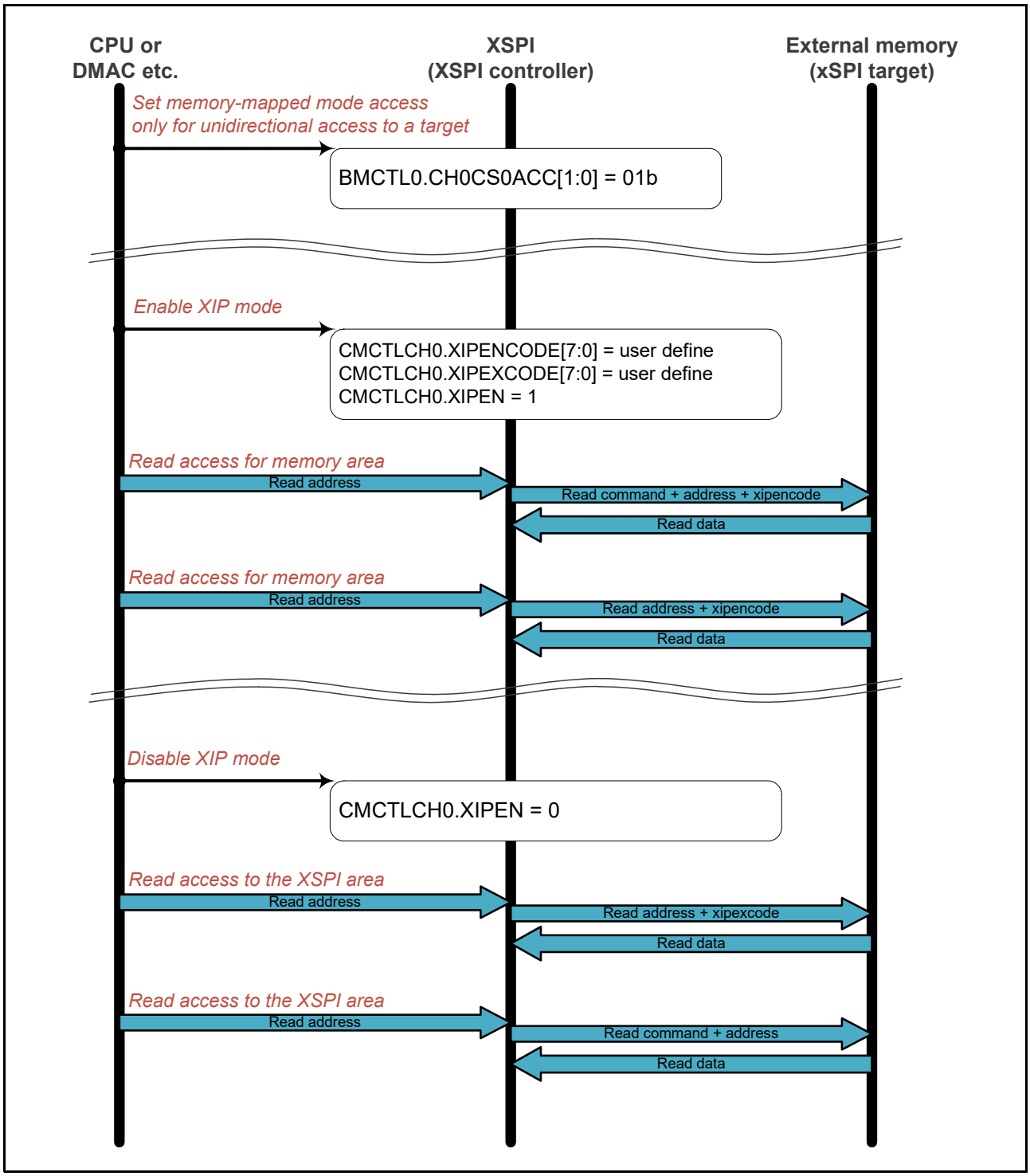


Figure 48.30 XIP Mode Enable/Disable Flow

(9) XSPI Module Stop Flow

Figure 48.31 shows the stop flow of the XSPI module.

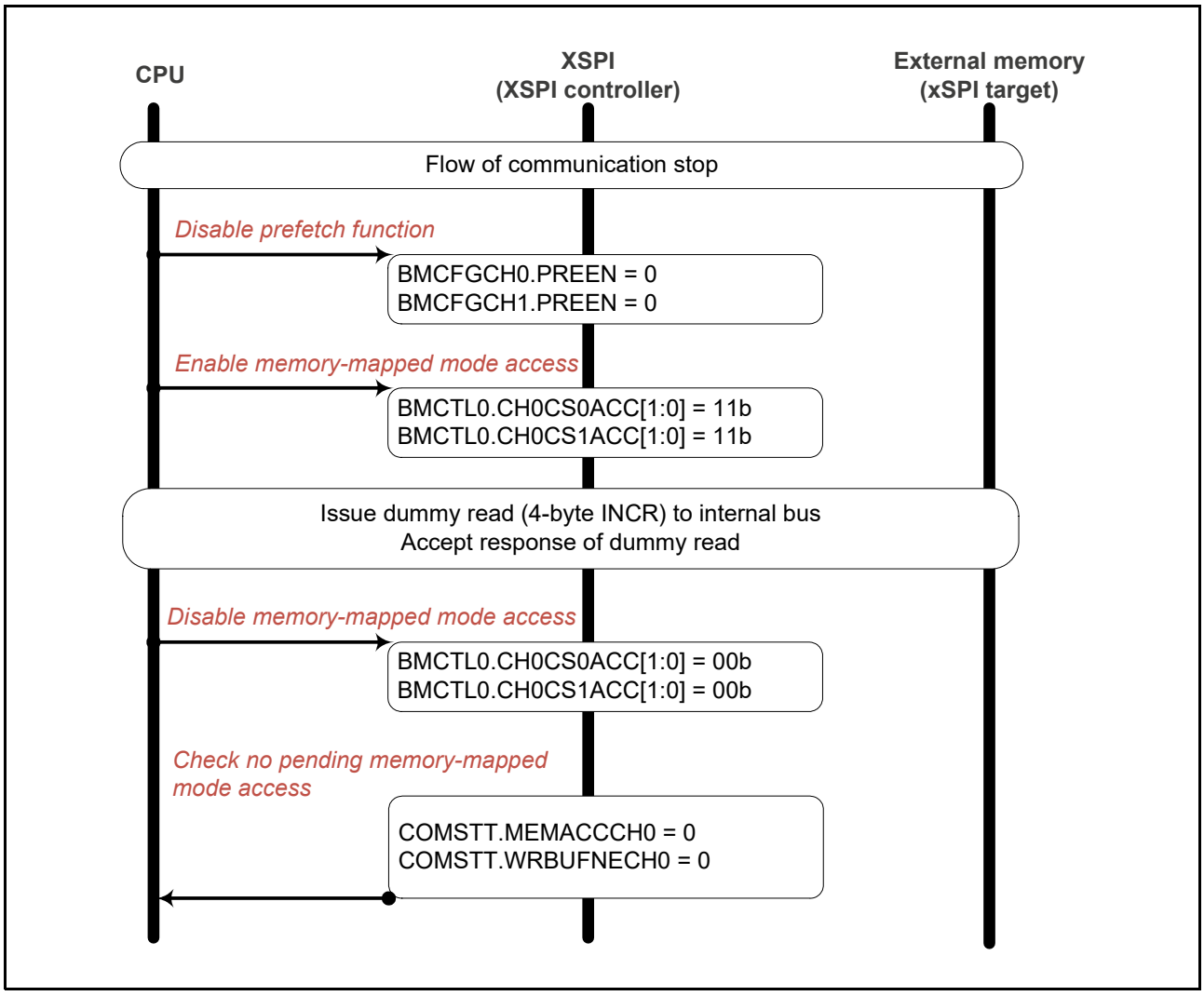


Figure 48.31 XSPI Module Stop Flow

48.3.2 XSPI Target Module

The XSPI target converts write/read accesses from the xSPI controller into write/read accesses to the internal bus. The xSPI controller must set the transfer address and transfer size in the registers of the XSPI target. The xSPI controller can then perform the intended data transfer.
The XSPI target operates in 8D-8D-8D protocol mode after reset is released. The xSPI controller can specify other protocol modes (1S-1S-1S or 4S-4D-4D) by register settings.
This section describes register access, memory write/read transfers, and protocol mode change.

48.3.2.1 XSPI Target Configuration

To enable the XSPI target function, set the DIPCFG.SLVEN bit to 1. After enabling the target function, access to the XSPI area and the registers of the XSPI controller is prohibited. Accessing these areas after setting the target mode will cause the XSPI operation to become undefined.

48.3.2.2 Register Access

The xSPI controller can access all registers of the XSPI target described in section 48.2.3, Target Mode Registers using the xSPI register write/read commands.
The register write command is defined by the L2CFG1.WRRCMD[7:0] bits, and the register read command is defined by the L2CFG1.RDRCMD[7:0] bits. The default frame format is 8D-8D-8D Profile 2.0 (specified by the L15CFG0.PRTMD[9:0] bits and L15CFG1.ADDSIZE[1:0], FFMT[1:0] bits). The register access size is 4 bytes only; other access sizes are prohibited.

Note: From this point on, register writes from the xSPI controller will be denoted as RegWrite(RegName, Value), and register reads as Value = RegRead(RegName). RegName can be L15CFG0, L15CFG1, L15CFG2, L2CFG0, L2CFG1, TWADR, TWLEN, TWCNT, TRADR, TRLEN, TRCNT, TRNSC, TWBS, TWTS, TRBS, TRTS, TS, TSC, TSS, TSXIE, TSSIE, or TBI. Value is the write or read value.

48.3.2.3 Memory Access

(1) Memory Map Configuration

Before initiating memory access from the xSPI controller, it is necessary to configure the address space of the XSPI target.
The xSPI controller must issue xSPI frames targeting the address space configured in the L2CFG0 register. Additionally, the maximum write frame size and maximum read frame size must be set to the MWRFSIZE[10:0] bits and MRDFSIZE[10:0] bits in the L15CFG2 register, respectively.
When accessing the XSPI target, the xSPI controller must ensure a minimum interval of 4 cycles between CS and the next CS.

(2) Write Access

The xSPI controller must set the transfer start address and transfer data size in the TWADR and TWLEN registers, respectively. The maximum transfer size is 1 Mbytes. Then, set the TWCNT.WEN bit to 1 to start the transfer.
Next, the xSPI controller polls the TWBS.VACSIZE[12:0] bits, and when the free space in the write buffer reaches or exceeds the size specified by the L15CFG2.MWRFSIZE[10:0] bits, it issues a write transaction of the size specified by the MWRFSIZE[10:0] bits. Only the final write frame will be a write transaction with a size equal to or less than the MWRFSIZE[10:0] bits.
The xSPI controller must transfer all write data of the size set in the TWLEN register, even if the TWBS.ERR flag is set to 1.
Figure 48.32 shows the memory write transfer from xSPI controller.

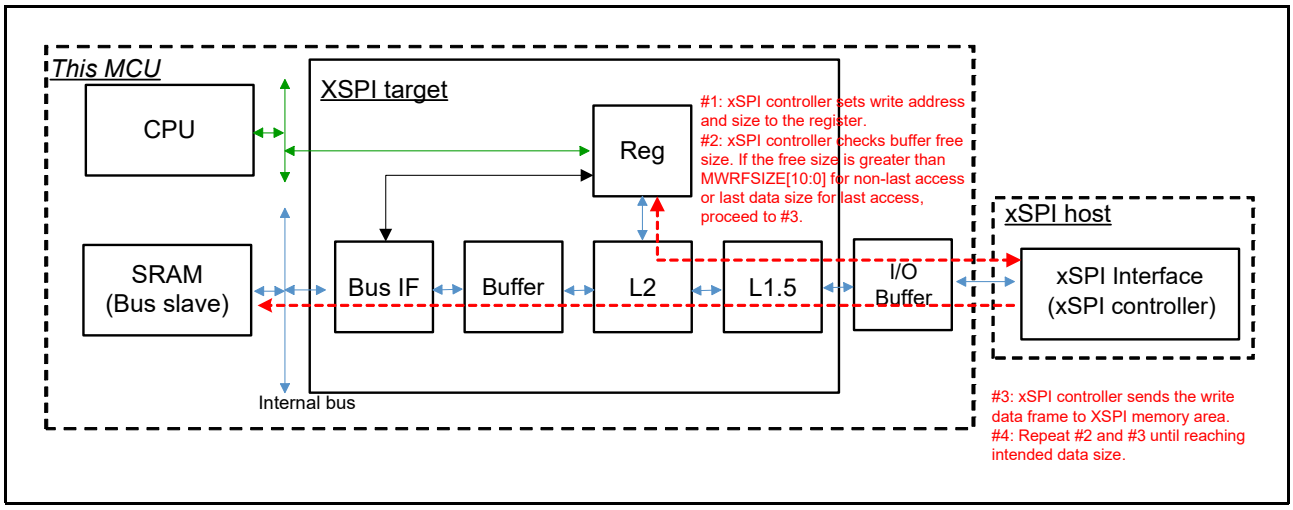


Figure 48.32 Memory Write Transfer from xSPI Controller

Figure 48.33 shows the flow of a write transfer as seen from the xSPI controller. The variables and functions used in the figure are defined as follows.

ValTargetWriteAddress: Write address in system memory of the XSPI target.
ValTransferSize: Total size of write data.
ValWrSize: Write size per frame.
ValVac: Variable to hold the read value of the TWBS register.
MemWrite: A function that issues a write transaction, with address, data size, and write data as arguments.
ValWrAdr: Address of the write frame. This value must be less than or equal to the value in the L2CFG0 register.
WriteData: Write data. This size must be less than or equal to ValVac.
ValTs: Variable to hold the read value of the TS register.

Note: Write access from the xSPI controller must be performed consecutively using frames with the data size specified by the L15CFG2.MWRFSIZE[10:0] bits. The XSPI target writes the received data sequentially starting from the address specified in the TWADR register, regardless of the address specified in the received frame. The XSPI target cannot rewind the address and rewrite data.

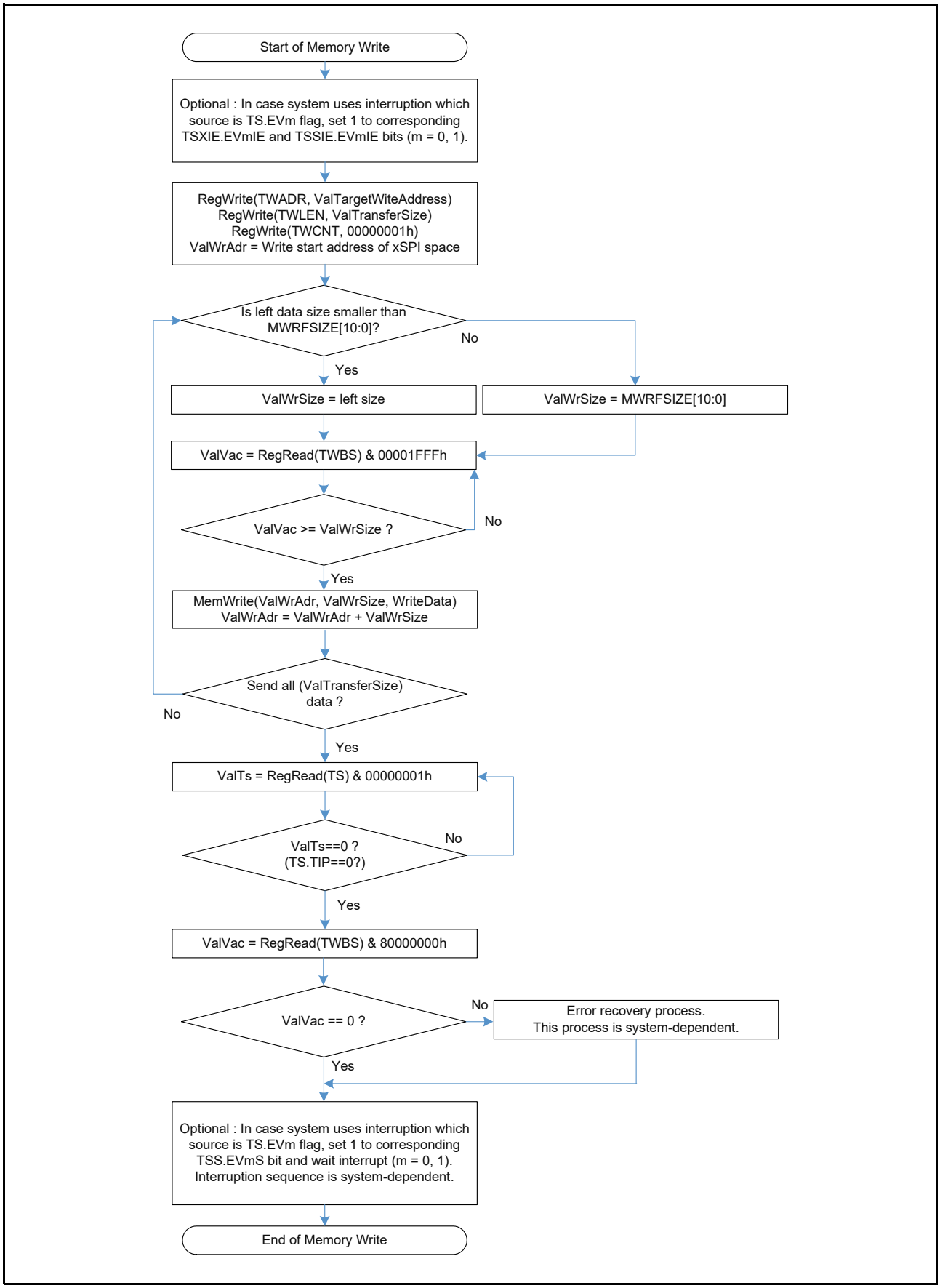


Figure 48.33 Memory Write Transfer Flow as Seen from xSPI Controller

(3) Read Access

The xSPI controller must set the transfer start address and transfer data size in the TRADR and TRLEN registers, respectively. The maximum transfer size is 1 Mbytes. Then, set the TRCNT.REN bit to 1 to start the transfer.
Next, the xSPI controller polls the TRBS.DATSIZE[12:0] bits, and when data equal to or more than the size specified by the L15CFG2.MRDFSIZE[10:0] bits is stored in the read buffer, it issues a read transaction of the size specified by the MRDFSIZE[10:0] bits. Only the final read frame will be a read transaction with a size equal to or less than the MRDFSIZE[10:0] bits (i.e., the size of the DATSIZE[12:0] bits).
The xSPI controller must transfer all data of the size set in the TRLEN register, even if the TRBS.ERR flag is set to 1.
Figure 48.34 shows the memory read transfer from xSPI controller.

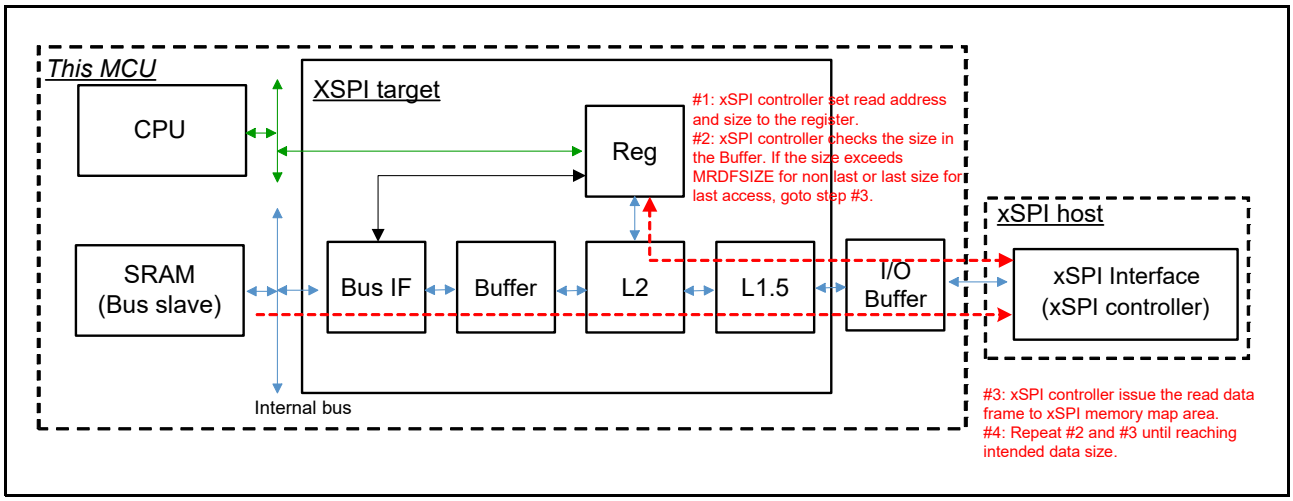


Figure 48.34 Memory Read Transfer from xSPI Controller

Figure 48.35 shows the flow of a read transfer as seen from the xSPI controller. The variables and functions used in the figure are defined as follows.

ValTargetReadAddress: Read address in the system memory of the XSPI target.
ValTransferSize: Total size of read data.
ValRdSize: Read size per frame.
ValDat: Variable to hold the read value of the TRBS register.
MemRead: A function that issues a read transaction, with address, data size, and storage address of read data as arguments.
ValRdAdr: Address of the read frame. This value must be less than or equal to the value in the L2CFG0 register.
&ReadData: Area for storing read data.

Note: Read access from the xSPI controller must be performed consecutively using frames with the data size specified by the L15CFG2.MRDFSIZE[10:0] bits. The XSPI target reads data sequentially starting from the address specified in the TRADR register, regardless of the address specified in the received frame. The XSPI target cannot rewind the address and reread data.

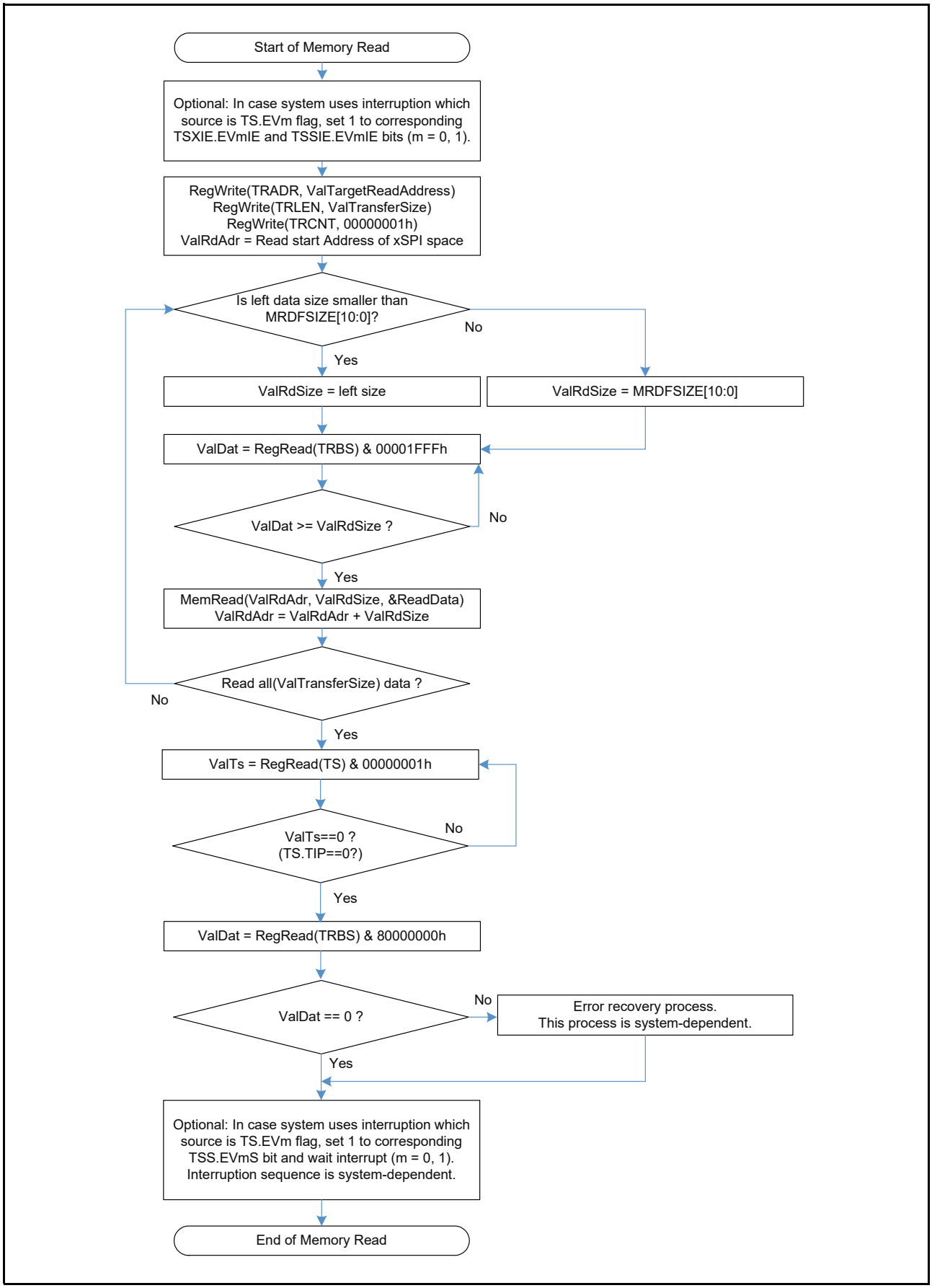


Figure 48.35 Memory Read Transfer Flow as seen from xSPI Controller

48.3.2.4 Changing IO Protocol Mode

Parameters related to the IO protocol mode of the XSPI target (such as protocol mode, address size, frame format, latency, etc.) can be changed. Before changing the IO protocol mode, the xSPI controller must ensure that no transfers are currently in progress.

Figure 48.36 shows the flow for changing these parameters. After sending the xSPI frame corresponding to the RegWrite(L15CFG1, ValL15CFG1Setting) in the flow below, the IO protocol for subsequent xSPI frames must follow the new settings. The definitions of the variables in the figure are as follows.

ValL15CFG0Setting: Setting value for the L15CFG0 register (user-defined)

ValL15CFG1Setting: Setting value for the L15CFG1 register (user-defined)

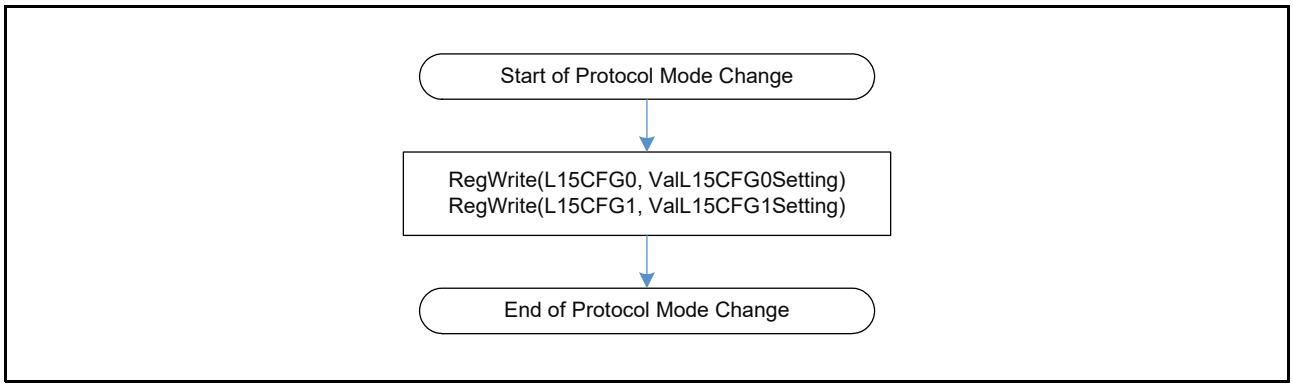


Figure 48.36 IO Protocol Mode Change Flow from xSPI Controller

48.3.2.5 xSPI Reset

The XSPI target has xSPI reset pins corresponding to the RESET# and RSTO# signals defined in the xSPI standard. The XSPI target has a reset information propagation mechanism shown in Figure 48.37.

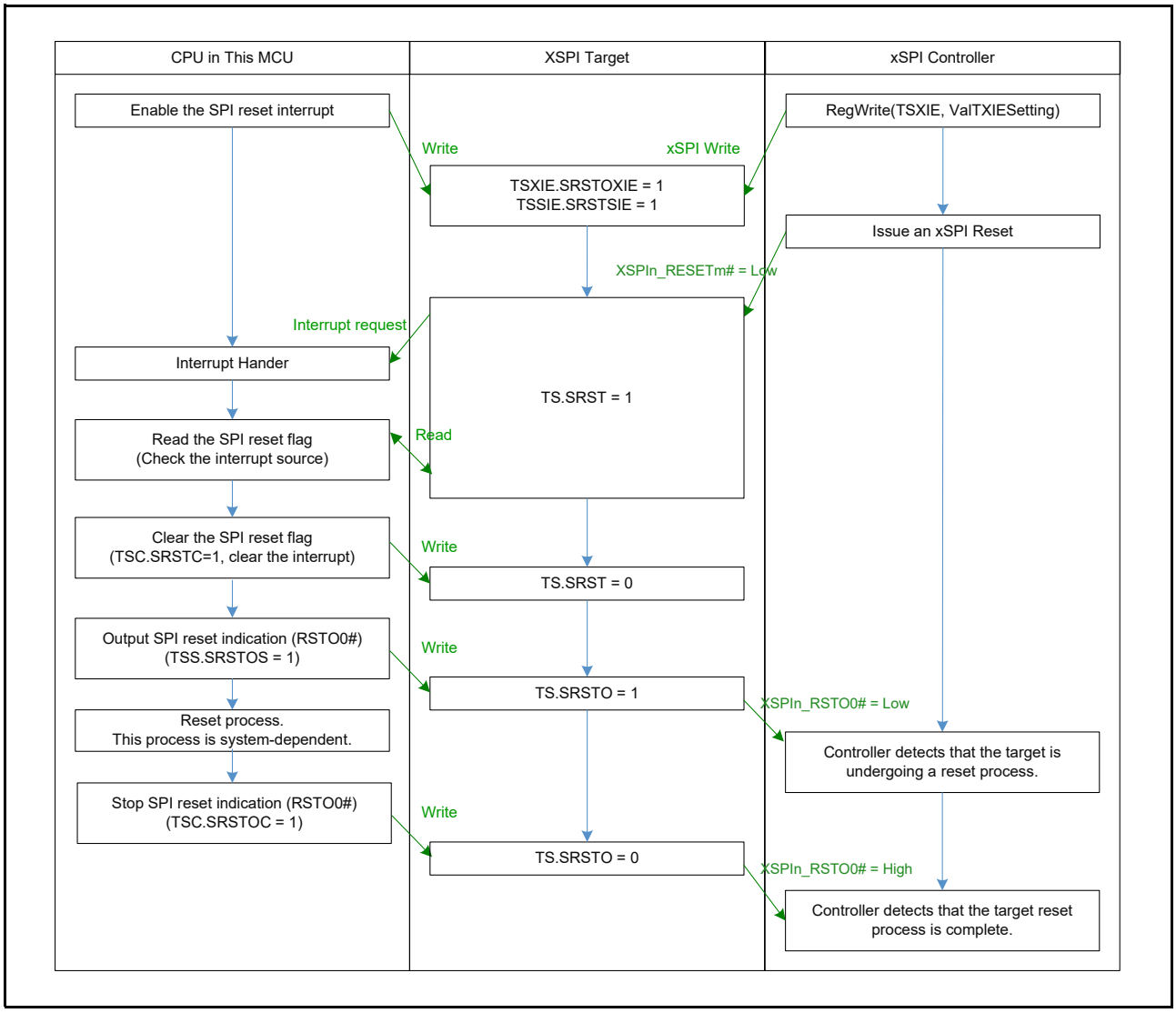


Figure 48.37 XSPI Target Reset Flow

48.3.2.6 Error Detection

The xSPI target can detect the errors shown in Table 48.15.

Table 48.15 Error List

Name	Error Source	Detection Flag	Remarks
Internal bus error	An error occurred on the internal bus when the XSPI target accessed an internal resource of this MCU.	TS.SBERR	Fatal error. This IP needs reset for recovery after this error.
Maximum frame size exceeded	The xSPI controller issued a transaction with a data size exceeding 4 bytes when accessing a register, or issued a transaction with a data size exceeding the size specified in the L15CFG2 register when accessing memory.	TS.MFSEERR	
Unknown command error	The command code in the transaction issued by the xSPI controller was undefined, or the xSPI controller issued an unauthorized transaction, or there was an error in the command extension.	TS.UNKCERR	
Buffer overflow	The xSPI controller issued a write transaction with a data size that exceeds the free space in the write buffer.	TS.BOVFERR	
Buffer underflow	The xSPI controller issued a read transaction with a data size that exceeds the data stored in the read buffer.	TS.BUDFERR	

48.3.2.7 Interrupt Sources

The XSPI target has two types of interrupts: internal bus interrupt that is notified to the CPU, and xSPI interrupt that is notified to the xSPI controller. The internal bus interrupt include the CMP interrupt and ERR interrupt, as shown in Table 48.16, and the xSPI interrupt has eight types of interrupt sources, as shown in Table 48.17.
Each interrupt can be monitored using the TS register, and the enable/disable setting can be changed using the TSSIE register (internal bus interrupt) or TSXIE register (xSPI interrupt).

Table 48.16 Internal Bus Interrupt Sources for XSPI Target

Name	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Interrupt Clear Bit	DTC/DMAC Triggerable
CMP*1	Event 0 detection	TS.EV0	TSSIE.EV0SIE	TSC.EV0C	No
	Event 1 detection	TS.EV1	TSSIE.EV1SIE	TSC.EV1C	
	SPI reset detection	TS.SRST	TSSIE.SRSTSIE	TSC.SRSTC	
	Transfer complete	TS.TCMP	TSSIE.TCMPSIE	TSC.TCMPC	
ERR*1	Internal bus error	TS.SBERR	TSSIE.SBERRSIE	TSC.SBERRC	No
	Maximum frame size exceeded	TS.MFSEERR	TSSIE.MFSEERRSIE	TSC.MFSEERRC	
	Unknown command error	TS.UNKCERR	TSSIE.UNKCERRSIE	TSC.UNKCERRC	
	Buffer underflow	TS.BUDFERR	TSSIE.BUDFERRSIE	TSC.BUDFERRC	
	Buffer overflow	TS.BOVFERR	TSSIE.BOVFERRSIE	TSC.BOVFERRC	

Note 1. The interrupts from the XSPI controller are each logically ORed and output to the interrupt controller (ICU).

Table 48.17 xSPI Interrupt Sources for XSPI Target (n = 0, 1)

Pin	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Interrupt Clear Bit
XSPIn_INT0#	Event 0 detection	TS.EV0	TSXIE.EV0XIE	TSC.EV0C
	Event 1 detection	TS.EV1	TSXIE.EV1XIE	TSC.EV1C
	Transfer complete	TS.TCMP	TSXIE.TCMPXIE	TSC.TCMPC
	Internal bus error	TS.SBERR	TSXIE.SBERRXIE	TSC.SBERRC
	Maximum frame size exceeded	TS.MFSEERR	TSXIE.MFSEERRXIE	TSC.MFSEERRC
	Unknown command error	TS.UNKCERR	TSXIE.UNKCERRXIE	TSC.UNKCERRC
	Buffer underflow	TS.BUDFERR	TSXIE.BUDFERRXIE	TSC.BUDFERRC
	Buffer overflow	TS.BOVFERR	TSXIE.BOVFERRXIE	TSC.BOVFERRC

48.4 Usage Notes

48.4.1 Settings of the Module Stop Function

XSPI operation can be disabled or enabled by the module stop control register B (MSTPCRB). After a reset, the operation of the XSPI is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

48.4.2 Notes on the XSPI Controller

48.4.2.1 Software Standby Mode

To transition to software standby mode, stop communication according to the communication stop flow and then stop the XSPI module according to the module stop flow.
Place the programs before and after the transition to software standby mode and the interrupt processing program used to return from software standby mode in the memory in this MCU, not on the xSPI target.
After returning from software standby mode, enable the settings that were disabled before transitioning to software standby mode, and if using automatic calibration, perform calibration before accessing the xSPI target.

48.4.2.2 Restrictions Regarding the Prefetch Function

When the prefetch function is enabled, do not access the last 128 bytes of the memory device. Accessing this area will issue transactions to addresses that do not exist on the memory device.

48.4.2.3 Note on the Prefetch Function

If all of the following conditions are met, prefetching does not proceed even if the prefetch function is enabled.

- Instruction caching is enabled and the XSPI area is subject to instruction caching, or data caching is enabled and the XSPI area is subject to data caching.
- Access by the CPU is to an XSPI area outside the decryption area for use with the DOTF unit.

48.4.2.4 Note on Using Special Memory Devices

In 8D-8D-8D protocol mode, the XSPI controller writes to and reads from memory in the order Data 0 – Data 1 – Data 2 – Data 3 –…. However, some memory devices input or output data in the order Data 1 – Data 0 – Data 3 – Data 2 –….
When reading data, which were written other than in 8D-8D-8D protocol mode, from such a device, in 8D-8D-8D protocol mode, change the order of writing the data to the device so that they are to be read in the order Data 0 – Data 1 – Data 2 – Data 3 –….

48.4.3 Notes on the XSPI Target

48.4.3.1 Software Standby Mode

When transitioning to software standby mode, ensure all operations on the XSPI target have stopped. Then, set the TRNSC.RSMSTB bit to 1 to stop communication functions.
After resuming from software standby mode, set the TRNSC.RSMSTB bit to 0.

48.4.3.2 Endian Setting

Regardless of the CPU's endian setting, the XSPI target is fixed to little endian. If the CPU is big endian, the xSPI controller must issue transactions with data sizes that are multiples of 4, starting from addresses that are multiples of 4.

49. Decryption On The Fly (DOTF)

49.1 Overview

This MCU has two units of on-the-fly decryption circuits (DOTF0, DOTF1) built into the expanded serial peripheral interfaces (XSPI0, XSPI1).
DOTF can decrypt data read by XSPI in memory-mapped mode at high speed using the AES core.
Table 49.1 lists the specifications of the DOTF.

Table 49.1 DOTF Specifications

Item	Description
Number of units	1 unit for each XSPI*1, total 2 units
Operating clock	Register clock: PCLKB AES core clock: ICLK
AES core function	• Utilize for on-the-fly decryption of encrypted software stored in external memory • Block size: 128 bits • Key sizes: 128, 192, or 256 bits • Block cipher mode of operation CTR: Compliant with NIST SP800-38A • Supports side-channel attack countermeasures • Supports self-test function
Tamper Resistance	Countermeasures against side-channel attacks, including SPA/DPA and timing attacks, are available.
Low power consumption	Module stop state can be set for each unit (simultaneously with XSPI)
ProtectZone filter	Security attribute can be set for each unit (same attribute with XSPI)*2

Note 1. For more information on XSPI, refer to section 48, Expanded Serial Peripheral Interface (XSPI).
Note 2. When each unit is set to the secure attribute, registers in the unit are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each unit is set to the non-secure attribute, registers in the unit are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 49.1 shows a block diagram of the DOTF. For a block diagram including XSPI, refer to section 48, Expanded Serial Peripheral Interface (XSPI).

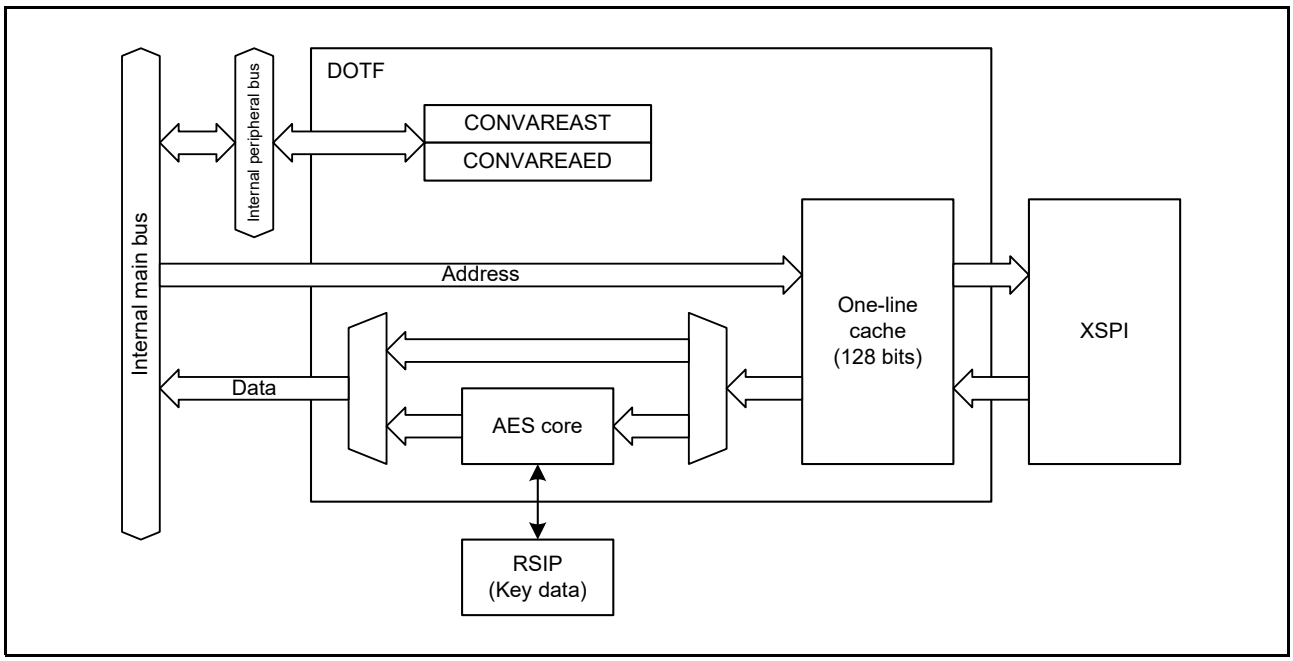


Figure 49.1 DOTF Block Diagram

49.2 Register Descriptions

49.2.1 Decryption Area Start Address Register (CONVAREAST)

Address(es): DOTF0.CONVAREAST 8786 8800h, DOTF1.CONVAREAST 8786 8900h, DOTF0_NS.CONVAREAST 0786 8800h, DOTF1_NS.CONVAREAST 0786 8900h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CONVAREAST register specifies the start address of the decryption area. The lower 12 bits are fixed at 000h. Set this register to a value less than the value of the CONVAREAED register.
Set this register before reading data from the XSPI area and do not change it thereafter.

49.2.2 Decryption Area End Address Register (CONVAREAED)

Address(es): DOTF0.CONVAREAED 8786 8804h, DOTF1.CONVAREAED 8786 8904h, DOTF0_NS.CONVAREAED 0786 8804h, DOTF1_NS.CONVAREAED 0786 8904h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

The CONVAREAED register specifies the end address of the decryption area. The lower 12 bits are fixed at FFFh. Set this register to a value larger than the value of the CONVAREAST register.
Set this register before reading data from the XSPI area and do not change it thereafter.

49.3 Operation

When a bus master such as the CPU or DMAC performs a read access to XSPI in memory-mapped mode, the data read from the xSPI target device is transferred to the bus master via DOTF.
DOTF determines whether decryption is required based on the address issued by the bus master, and if necessary, it decrypts the data using the AES core. If decryption is not required, the read data is output as is.
The area where the decryption process by DOTF is performed (decryption area) is limited to the range from the address set in the CONVAREAST register to the address set in the CONAREAED register.

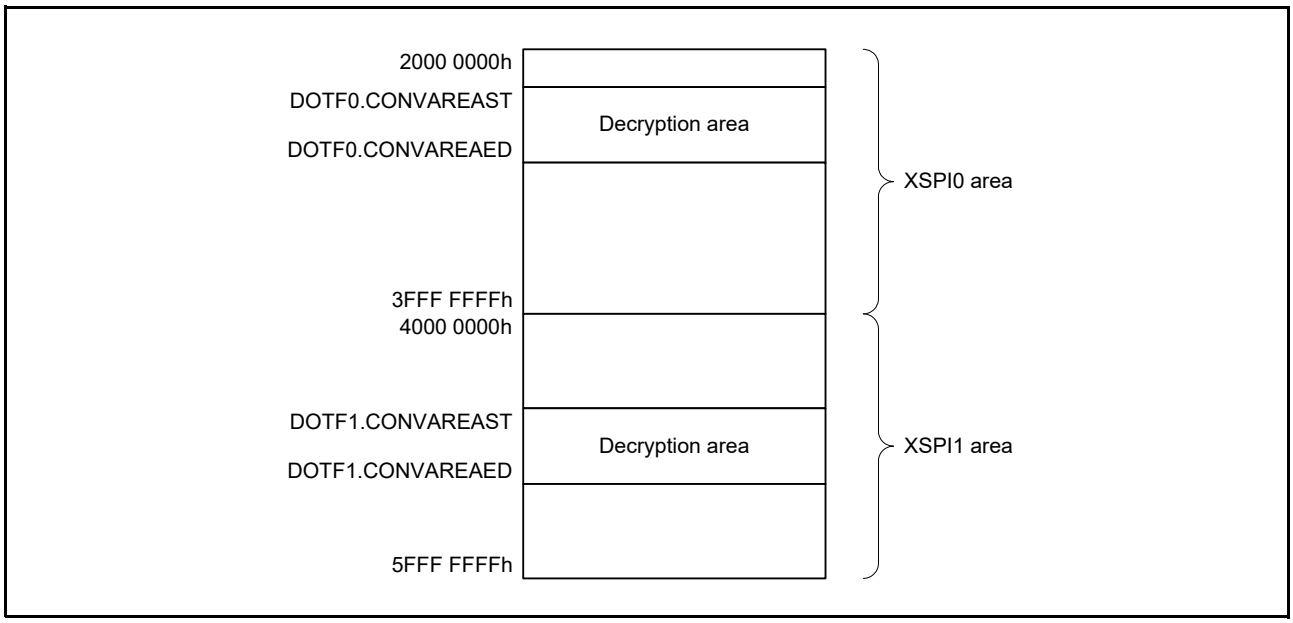


Figure 49.2 Decryption Area Setting

49.3.1 Initialization Flow

Figure 49.3 shows the initialization flow after hardware reset.

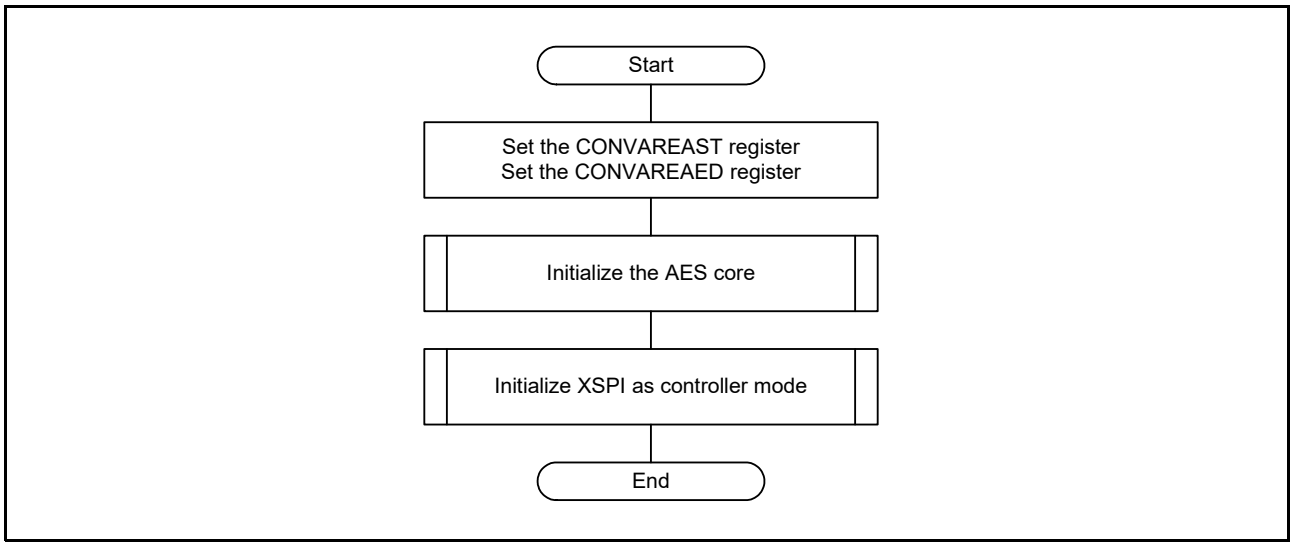


Figure 49.3 Initialization Flow

49.4 Usage Notes

49.4.1 Module Stop Function Setting

Operation of the DOTF module can be disabled or enabled using the module stop control register B (MSTPCRB). The DOTF is initially stopped after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

50. CRC Calculator (CRCA)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

50.1 Overview

Table 50.1 lists the specifications of the CRC calculator, and Figure 50.1 shows a block diagram of the CRC calculator.

Table 50.1 CRC Specifications

Item	Description	
Data size	8 bits	32 bits
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)	CRC codes are generated for any desired data in 32n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing	32-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$	One of two generating polynomials is selectable • 32-bit CRC $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$
CRC calculation switching	The order of the bits produced by CRC calculation can be switched for LSB first or MSB first communication	
Low power consumption	Module stop state can be set.	
Protect Zone filter	Security attribute can be set.*2	

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit or 32-bit units.
Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

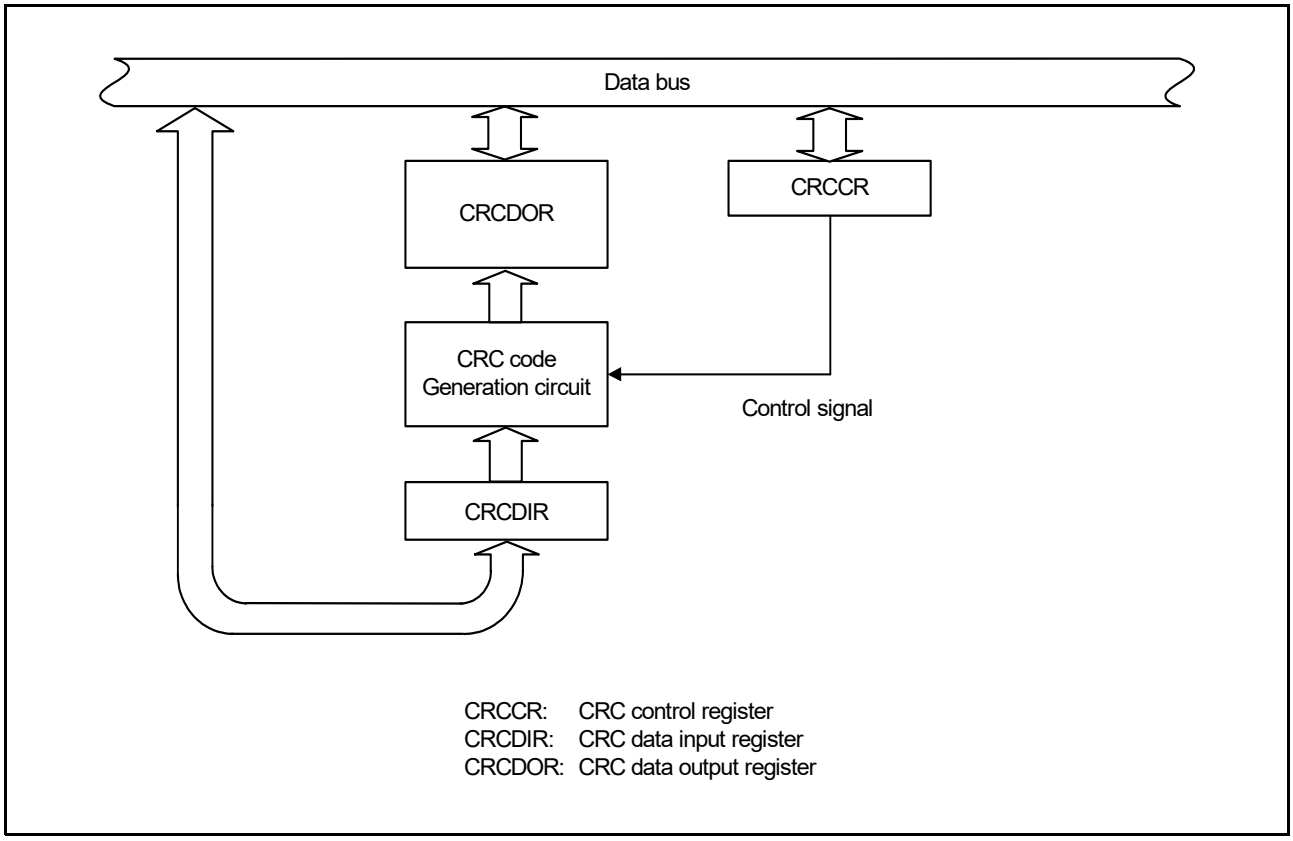


Figure 50.1 CRC Block Diagram

50.2 Register Descriptions

50.2.1 CRC Control Register (CRCCR)

Address(es): CRC.CRCCR 8741 0000h, CRC_NS.CRCCR 0741 0000h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GPS[2:0]	CRC Generating Polynomial Switching	b2 b1 b0 0 0 0: No calculation is executed. 0 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) 1 1 0: No calculation is executed. 1 1 1: No calculation is executed.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

The setting this bit selects the order of the bits of generated CRC codes. The bit selects transmission of the lower-order byte of the CRC code first for LSB first communication, or the higher-order byte first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 50.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000 0000h.
This bit is read as 0. Writing 0 to this bit has no effect.

50.2.2 CRC Data Input Register (CRCDIR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDIR 8741 0004h, CRC_NS.CRCDIR 0741 0004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 16-bit or 8-bit CRC is selected

Address(es): CRC.CRCDIR 8741 0004h, CRC_NS.CRCDIR 0741 0004h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CRCDIR register is a readable and writable register. Write data for CRC calculation to this register.
When generating a 32-bit CRC, the CRCDIR register should be accessed in longword units. When generating a 8-bit or 16-bit CRC, the CRCDIR register should be accessed in byte units.

50.2.3 CRC Data Output Register (CRCDOR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDOR 8741 0008h, CRC_NS.CRCDOR 0741 0008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 16-bit CRC is selected

Address(es): CRC.CRCDOR 8741 0008h, CRC_NS.CRCDOR 0741 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 8-bit CRC is selected

Address(es): CRC.CRCDOR 8741 0008h, CRC_NS.CRCDOR 0741 0008h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CRCDOR register is a readable and writable register.
The value after a reset is 0000 0000h. When calculating with the initial value set to a value other than 0000 0000h, rewrite the CRCDOR register.
Writing data to the CRCDIR register stores result of calculation in the CRCDOR register. In addition, following communication data, when a CRC code is written to the CRCDIR register and if the calculation result is 0000 0000h, there is no error in the communication data.
When a 32-bit CRC is selected, the CRCDOR register should be accessed in longword units. When a 16-bit CRC is selected, the CRCDOR register should be accessed in word units. When an 8-bit CRC is selected, the CRCDOR register should be accessed in byte units.

50.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first communication.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared by setting the DORCLR bit to 1 to set the initial value for CRC calculation to 0000 0000h.

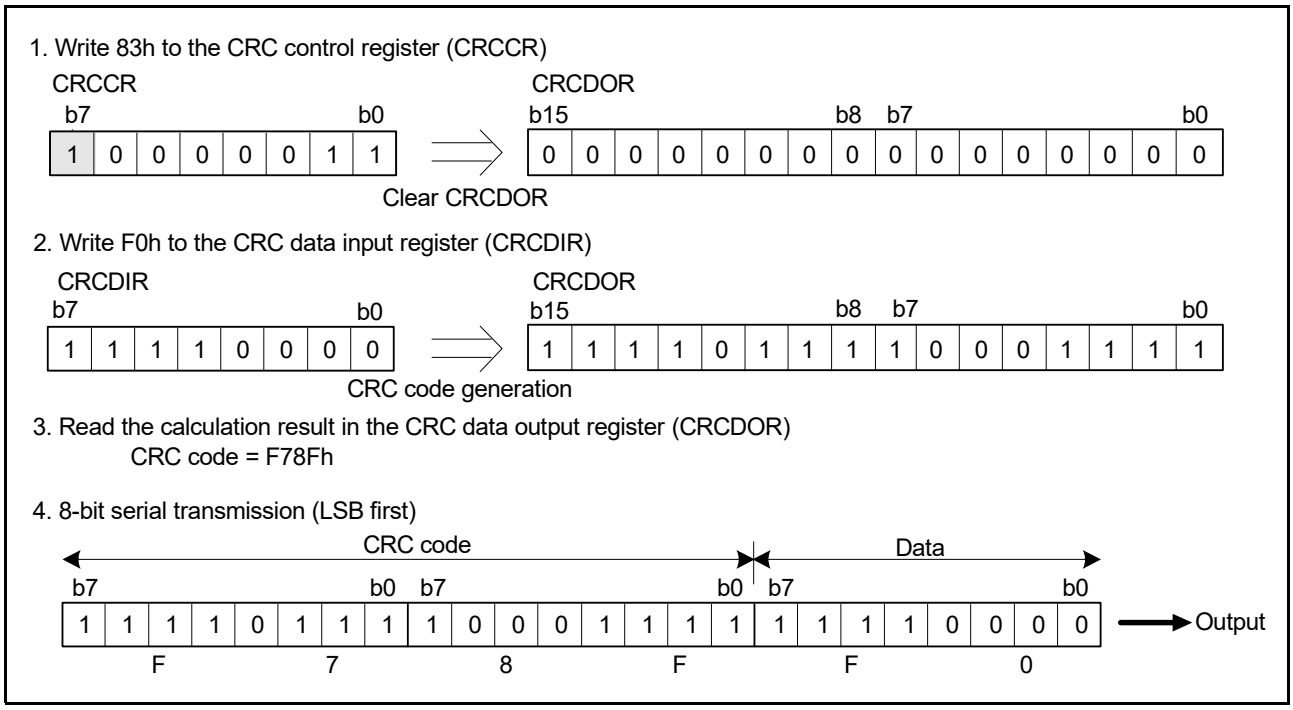


Figure 50.2 LSB First Data Transmission

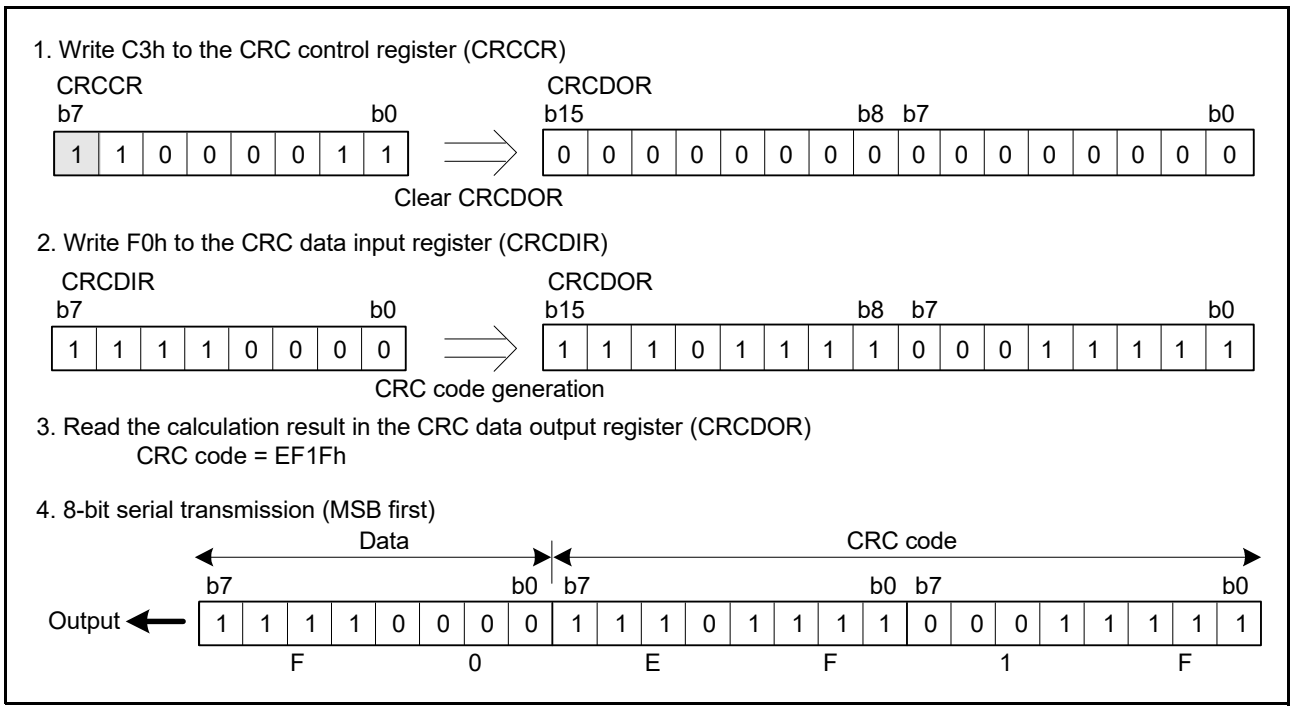


Figure 50.3 MSB First Data Transmission

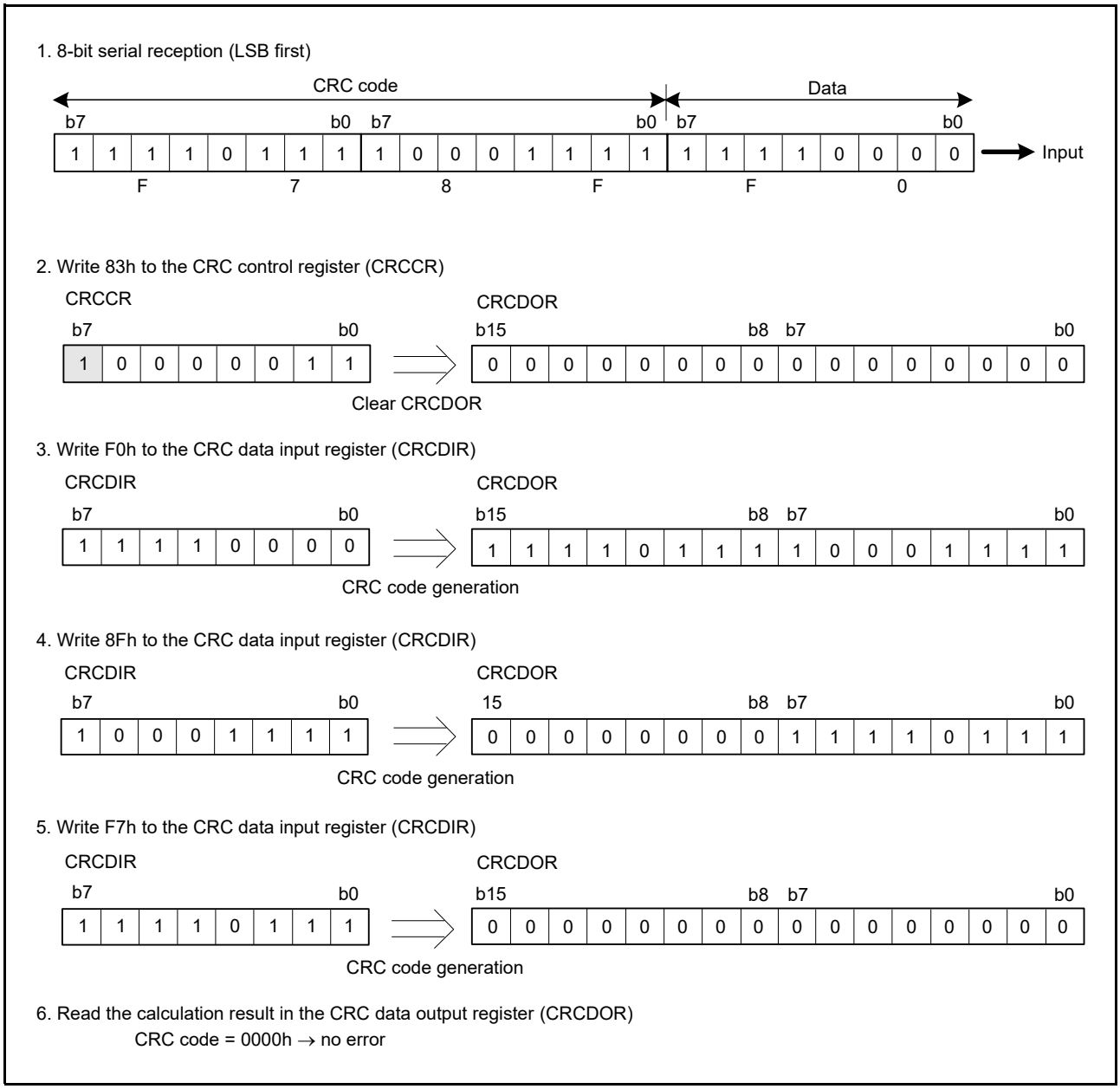


Figure 50.4 LSB First Data Reception

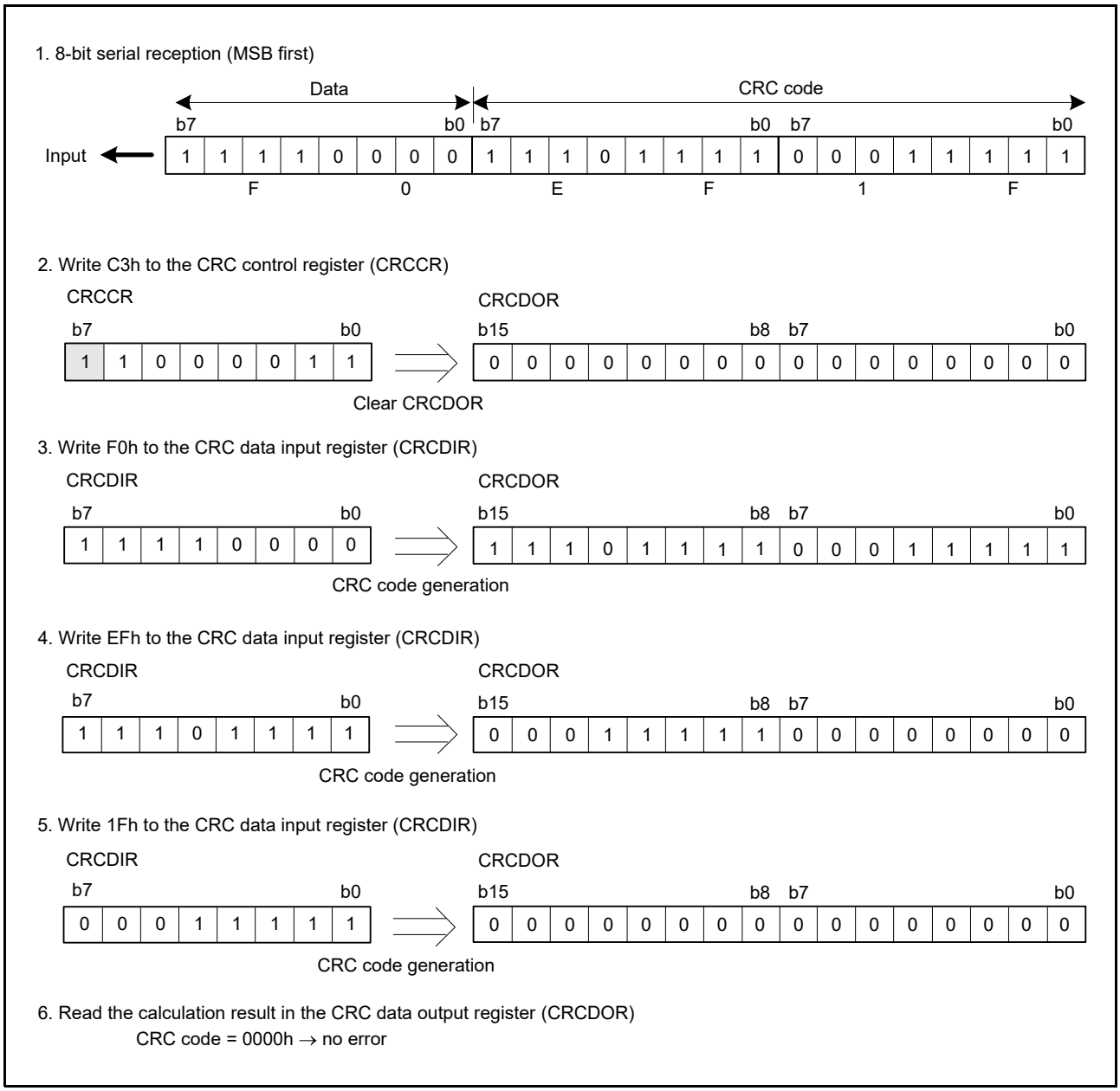


Figure 50.5 MSB First Data Reception

50.4 Usage Notes

50.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register C (MSTPCRC). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state.
For details, refer to section 11, Low Power Consumption.

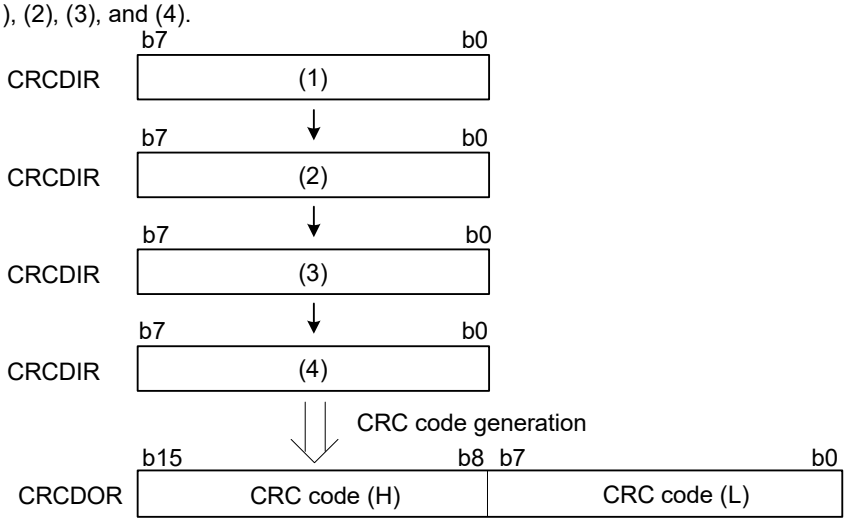
50.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

When transmitting 32-bit data

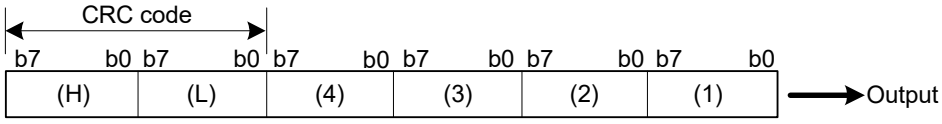
1. CRC code

After specifying the method for generation calculation, write data to the CRCDIR register in the order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB first



(ii) When transmission is MSB first

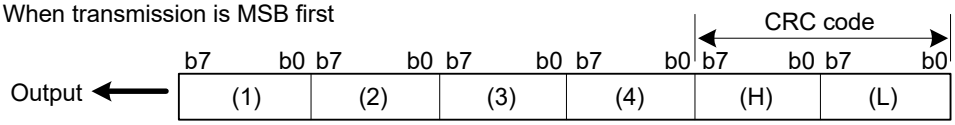


Figure 50.6 LSB First and MSB First Data Transmission

51. Serial Sound Interface (SSIE)

This MCU has two serial sound interfaces (SSIE0 and SSIE1). Each SSIE supports monaural and time-division multiplexed (TDM) formats in addition to standard audio data formats such as I²S, left-justified, and right-justified.

51.1 Overview

Table 51.1 lists the specifications of the SSIE. Table 51.2 lists and Figure 51.1 shows the terms and the definitions used in this chapter. Figure 51.2 shows a block diagram of the SSIE.

Table 51.1 Features of SSIE

Item		Description
Number of channels		Two channels, SSIE0 and SSIE1
Transfer mode		• Master/slave • Transmission, reception, or transception
Data format		• I²S format • Left-justified format • Right-justified format • Monaural format • TDM format
Serial data		• Fixed to MSB first • System word length: Selectable from among 8, 16, 24, 32, 48, 64, 128, or 256 bits • Data word length: Selectable from among 8, 16, 18, 20, 22, 24, or 32 bits • The level of padding bits is programmable. • Mute function
Master clock (MCK)		Clock source: AUDIO_CLK or GTIOC14A
Bit clock (BCK)	In master mode	• Frequency: Selectable from among 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, or 1/128 of the MCK frequency • Continued supply or stopping while data transfer is halted is selectable.
	In master and slave modes	• Polarity is selectable as rising edges or falling edges.
LR clock (LRCK)		• Polarity is selectable as low or high. • Continued supply or stopping while data transfer is halted is selectable.
FIFO	Capacity	• Transmit FIFO: 4 bytes × 32 stages • Receive FIFO: 4 bytes × 32 stages
	Data alignment	• Alignment of data (left-justified or right-justified) in the FIFO is selectable.
Interrupts		• Data transfer error/idle state • Receive data full • Transmit data empty
Low power consumption function		• Module stop function • Master clock (MCK) supply stopping function
Protect Zone filter		Security attribute can be set for each channel.*1, *2

Note 1. If the security attributes for each channel are different, there are some notes regarding the input of AUDIO_CLK. Refer to section 51.5.2, Notes on AUDIO_CLK for details.

Note 2. When each channel is set to the secure attribute, registers in the channel are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each channel is set to the non-secure attribute, registers in the channel are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 51.2 Definitions of Terms

Term	Definition
Start trigger	First edge of the type specified by the SSICR.LRCKP bit of the signal on the SSILRCK pin after the start of data transfer
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame
Frame word length	Number of system words per frame (number of channels)
System word length	Number of transferred bits per channel
Data word length	Number of effective bits per channel

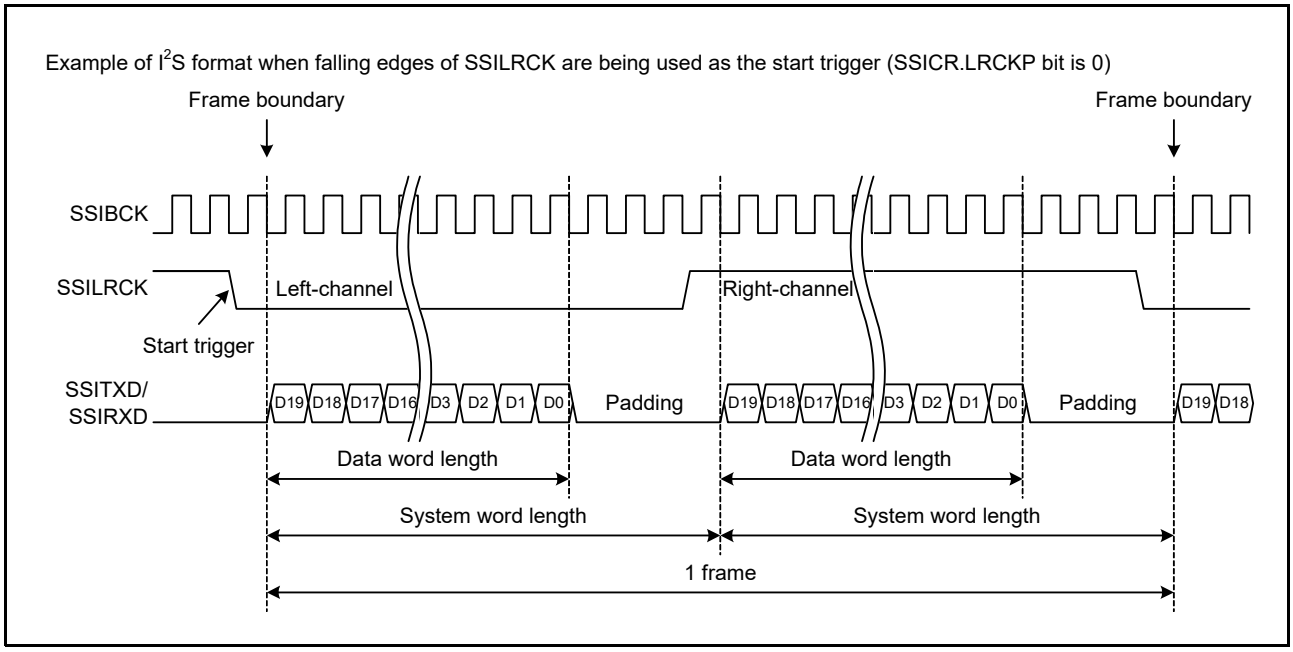


Figure 51.1 Definitions of Terms Related to Data Format

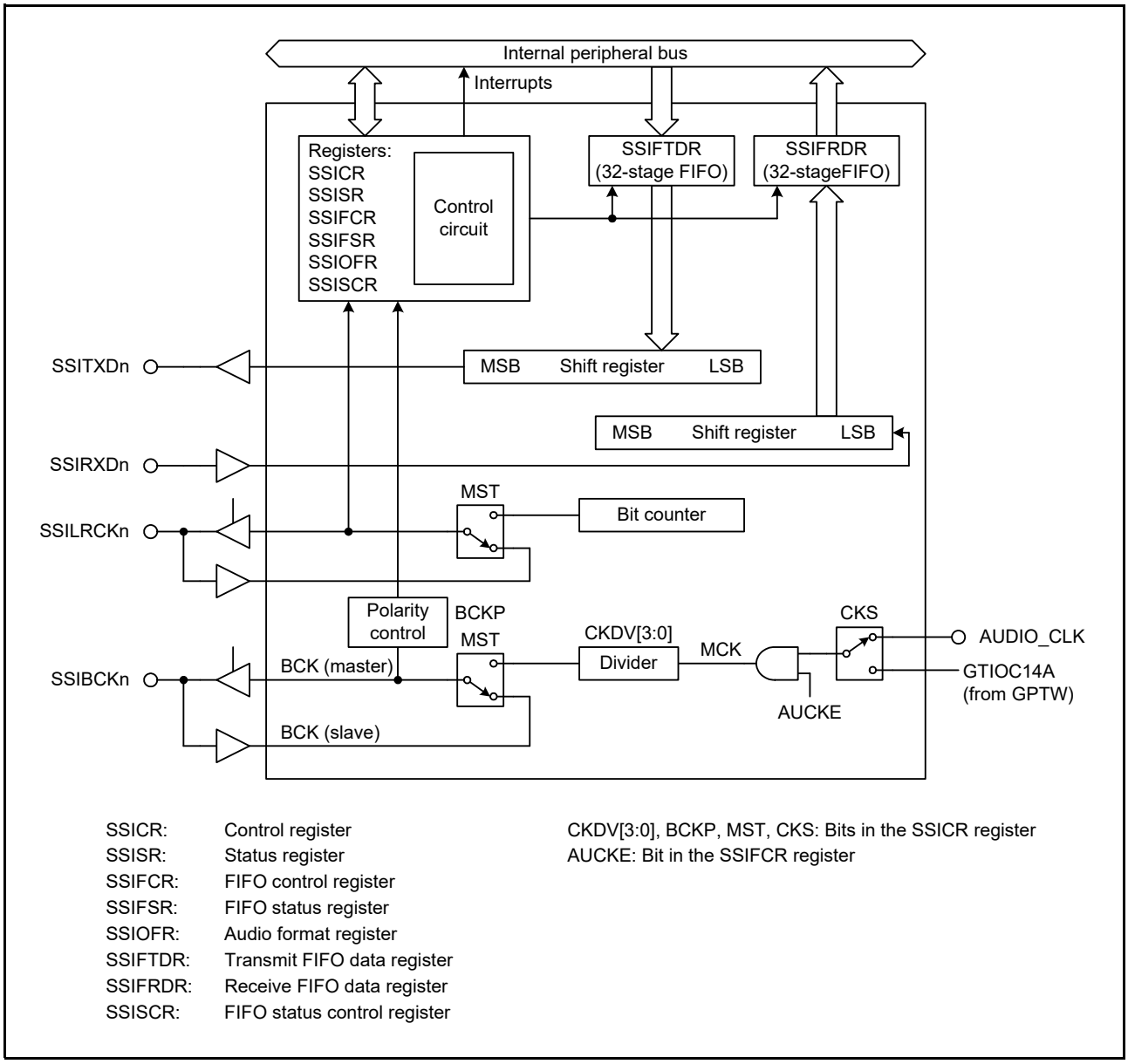


Figure 51.2 SSIE Block Diagram (n = 0, 1)

Table 51.3 lists the I/O pins of the SSIEs.

Table 51.3 I/O pins of SSIE

Pin Name	I/O	Function
AUDIO_CLK	Input	Audio clock input
SSIBCK0	I/O	Channel 0 bit-clock input/output
SSILRCK0	I/O	Channel 0 LR clock input/output
SSITXD0	Output	Channel 0 data output
SSIRXD0	Input	Channel 0 data input
SSIBCK1	I/O	Channel 1 bit-clock input/output
SSILRCK1	I/O	Channel 1 LR clock input/output
SSITXD1	Output	Channel 1 data output
SSIRXD1	Input	Channel 1 data input

51.2 Register Descriptions

51.2.1 Control Register (SSICR)

Address(es): SSIE0.SSICR 8785 D000h, SSIE1.SSICR 8785 D100h, SSIE0_NS.SSICR 0785 D000h, SSIE1_NS.SSICR 0785 D100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REN	Reception Enable*1	0: Reception disabled 1: Reception enabled	R/W
b1	TEN	Transmission Enable*1	0: Transmission disabled 1: Transmission enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute	0: Unmute 1: Mute	R/W
b7 to b4	CKDV[3:0]	Bit Clock Division Ratio Select *2, *3	Set the division ratio to generate a bit clock from MCK. b7 b4 0 0 0 0: No division 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 6 1 0 0 1: Divided by 12 1 0 1 0: Divided by 24 1 0 1 1: Divided by 48 1 1 0 0: Divided by 96 Settings other than above are prohibited.	R/W
b8	DEL	Serial Data Delay Select*2	Set the delay between the edges of SSILRCK and the input and output of data. 0: Delay of 1 cycle of SSIBCK 1: No delay	R/W
b9	PDTA	Data Alignment Select*2	Set the data alignment of the SSIFTDR and SSIFRDR registers. 0: Left-justified 1: Right-justified	R/W
b10	SDTA	Serial Data Alignment Select *2	0: I2S and left-justified format (data bits are transmitted and received before the padding bits) 1: Right-justified format (data bits are transmitted and received after the padding bits)	R/W
b11	SPDP	Padding Data Select*2	0: Padding bits are at the low level. 1: Padding bits are at the high level.	R/W
b12	LRCKP	LR Clock Polarity Select*2	0: SSILRCK is low for left channel. The frame boundary is a falling edge of SSILRCK. 1: SSILRCK is high for left channel. The frame boundary is a rising edge of SSILRCK.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13	BCKP	Bit Clock Polarity Select*2, *3	0: Output signals change at a falling edge of bit clock and input signals are sampled at a rising edge of bit clock. 1: Output signals change at a rising edge of bit clock and input signals are sampled at a falling edge of bit clock.	R/W
b14	MST	Master Mode*2, *3	0: Slave mode 1: Master mode	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	SWL[2:0]	System Word Length Select*2	b18 b16 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 24 bits 0 1 1: 32 bits 1 0 0: 48 bits 1 0 1: 64 bits 1 1 0: 128 bits 1 1 1: 256 bits	R/W
b21 to b19	DWL[2:0]	Data Word Length Select*2	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits 1 1 0: 32 bits 1 1 1: Setting prohibited.	R/W
b23, b22	FRM[1:0]	Frame Word Length Select*2	Set a number of system words per a frame. b23 b22 0 0: 2 system words (I^2S, left-justified, or right-justified) or 1 system word (monaural) 0 1: 4 system words (TDM format only) 1 0: 6 system words (TDM format only) 1 1: 8 system words (TDM format only)	R/W
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25	IIEN	Idle State Interrupt Enable	0: Idle state interrupt disabled 1: Idle state interrupt enabled	R/W
b26	ROIEN	Receive Overflow Interrupt Enable	0: Receive overflow interrupt disabled 1: Receive overflow interrupt enabled	R/W
b27	RUIEN	Receive Underflow Interrupt Enable	0: Receive underflow interrupt disabled 1: Receive underflow interrupt enabled	R/W
b28	TOIEN	Transmit Overflow Interrupt Enable	0: Transmit overflow interrupt disabled 1: Transmit overflow interrupt enabled	R/W
b29	TUIEN	Transmit Underflow Interrupt Enable	0: Transmit underflow interrupt disabled 1: Transmit underflow interrupt enabled	R/W
b30	CKS	Master Clock Select*2, *3	0: Input signal from the AUDIO_CLK pin 1: The GTIOC14A signal from GPTW	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. After the TEN or REN bit is rewritten, confirm that the SSISR.IIRQ flag is in the desired status (when transmission or reception is enabled, confirm that the IIRQ flag is 0; when transmission and reception are disabled, confirm that the IIRQ flag is 1).
Note 2. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.
Note 3. Rewrite these bits while the SSIFCR.AUCKE bit is 0.

TEN Bit (Transmission Enable) and REN Bit (Reception Enable)

These bits enable/disable transmission and reception. When one of these bits is set to 1, the given type of data transfer starts in synchronization with a start trigger of the SSILRCK signal. When the bit is set to 0, the data transfer stops at the next frame boundary. To use an SSIE for both transmission and reception, set both the TEN and REN bits to 1 simultaneously. To stop the data transfer, set both the TEN and REN bits to 0 simultaneously.
To switch among transmission, reception, and transception, do so in the idle state.

MUEN Bit (Mute)

This bit is used to mute audio output by driving the SSITXD pin low. When this bit is changed during a frame, the setting takes effect at the next frame boundary.
The MUEN bit only controls fixing of the SSITXD pin. Status flags are set and interrupt signals are generated regardless of the setting of the MUEN bit.
Changing the value of this bit must be performed after setting the data format to be used.

CKDV[3:0] Bits (Bit Clock Division Ratio Select)

These bits specify the divisor for the generation of a bit clock (BCK) from the master clock (MCK). This bit is disabled in slave mode.
Set these bits while the SSIFCR.AUCKE bit is 0.

DEL Bit (Serial Data Delay Select)

When this bit is 0, a delay of one cycle of the bit clock is inserted between edges of SSILRCK and SSITXD and SSIRXD. When 1, no delay is inserted.
Set this bit to 0 for the I^{2}S format and to 1 for the left-justified or right-justified format. Set this bit in conformance with the other party device for the monaural and TDM formats.When the monaural format is used, setting of this bit changes the high width of SSILRCK.

PDTA Bit (Data Alignment Select)

This bit specifies the alignment of the effective bits in the SSIFTDR and SSIFRDR registers. This bit is valid when the data word length 18 to 24 bits (the DWL[2:0] bits = 010b to 101b). Set this bit to 0, when the other data word length is selected.
Figure 51.3 shows the data alignment of the SSIFTDR and SSIFRDR registers.

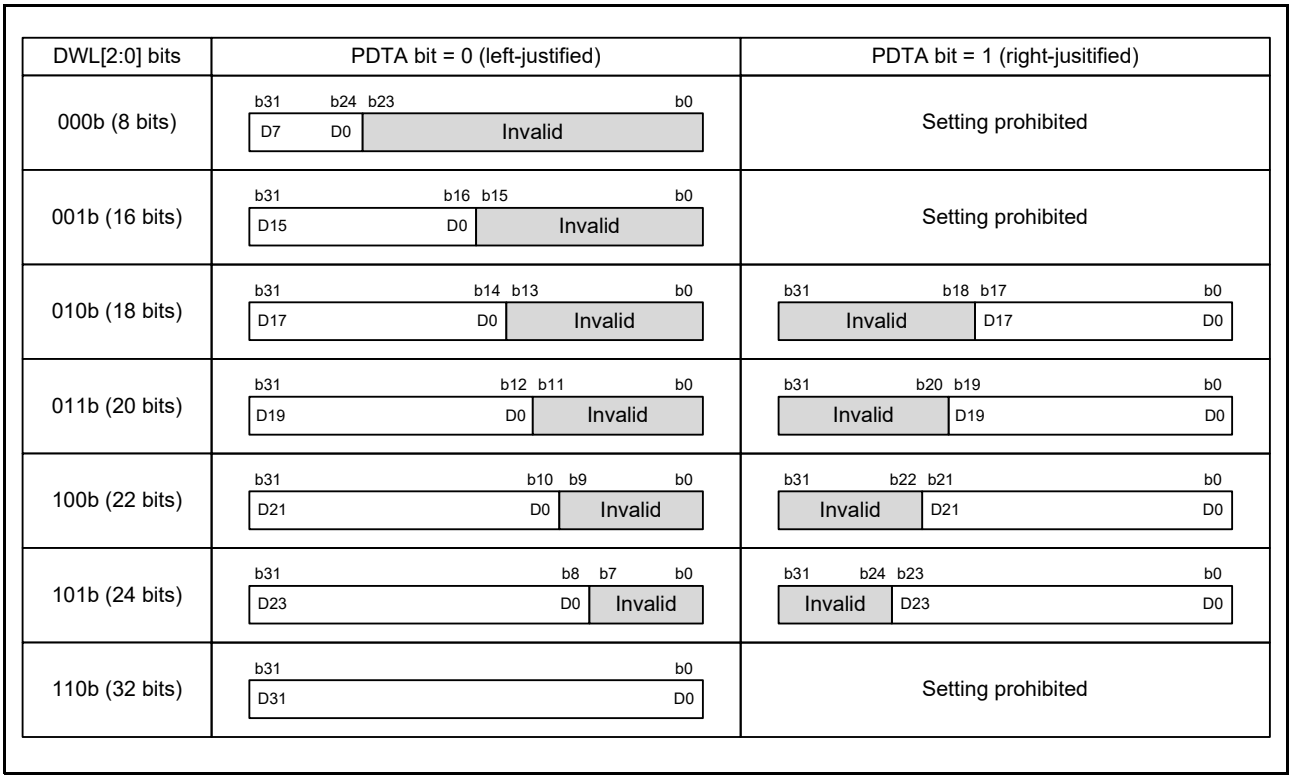


Figure 51.3 Data Alignment in the SSIFTDR and SSIFRDR Registers

SDTA Bit (Serial Data Alignment Select)

This bit specifies the order of data bits and padding bits for transception in system words. SSIE transmits and receives data bits first and then padding bits when this bit is 0 and padding bits first and then data bits when this bit is 1. This bit is invalid for data format without padding bits.

LRCKP Bit (LR Clock Polarity Select)

This bit specifies the initial status and polarity of the SSILRCK pin. Table 51.4 lists the settings of this bit and the corresponding states of the LR clock. SSIE uses the SSILRCK signal as a start trigger in slave mode.
Set this bit to 0 when the I^2S format is selected and 1 when the other data formats are selected.

Table 51.4 The Value of the LRCKP Bit and Responses to the States of the LR clock

LRCKP bit	Initial State of SSILRCK pin	Left Channel	Right Channel	Frame Boundary
0	High	Low	High	Falling edge
1	Low	High	Low	Rising edge

BCKP Bit (Bit Clock Polarity Select)

This bit specifies the polarity of the bit clock. Table 51.5 lists the settings of this bit and the corresponding timings of the respective signals.
Set this bit while the SSIFCR.AUCKE bit is 0.

Table 51.5 The Value of the BCKP bit and the Bit Clock Polarity

BCKP bit	Output of SSILRCK SSITXD	Input of SSILRCK SSIRXD
0	The signals change on falling edges of SSIBCK.	The signals are sampled on rising edges of SSIBCK.
1	The signals change on rising edges of SSIBCK.	The signals are sampled on falling edges of SSIBCK.

SWL[2:0] Bits (System Word Length Select)

These bits specify the number of bits in a system word. The setting of these bits determines the frequency ratio of the bit clock to the LR clock. Then number of the padding bits is obtained from the difference in the SWL[2:0] and DWL[2:0] bits. Refer to Table 51.8 for details.

DWL[2:0] Bits (Data Word Length Select)

These bits specify the number of bits of valid data in a system word. The data word length must not exceed the system word length. For details, refer to Table 51.8.

FRM[1:0] Bits (Frame Word Length Select)

These bits specify the number of system words in a frame.
Set these bits to 00b when the I^{2}S, left-justified, right-justified, or monaural format is selected. Do not set these bits to 00b when the TDM format is selected.

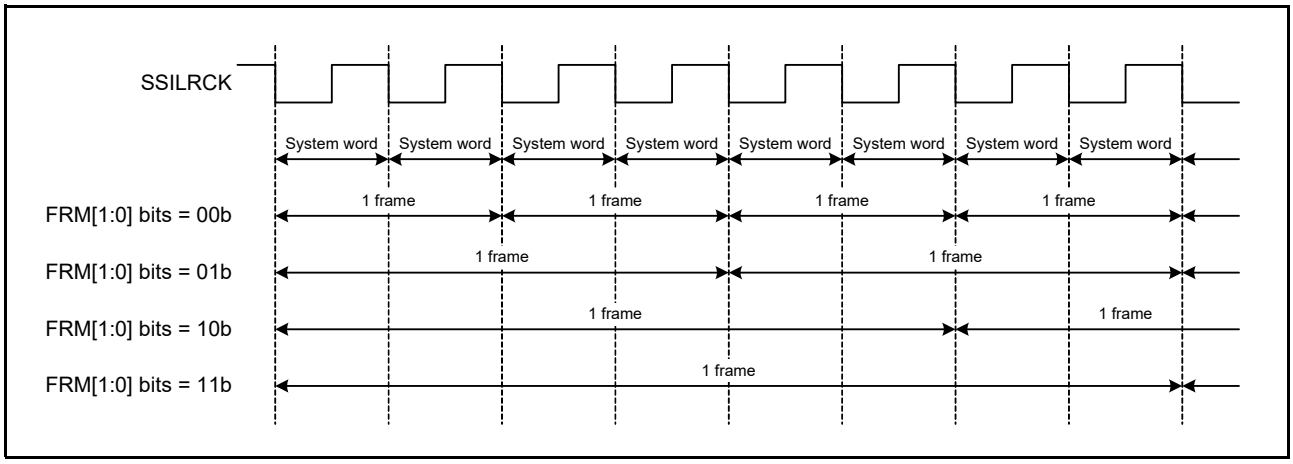


Figure 51.4 Frame Word Length

IIEN Bit (Idle State Interrupt Enable)

This bit enables or disables the idle state interrupts.
While this bit is 1, an interrupt request is output when the SSISR.IIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the IIRQ flag is 1.

ROIEN Bit (Receive Overflow Interrupt Enable)

This bit enables or disables interrupts receive overflow interrupts.
While this bit is 1, an interrupt request is output when the SSISR.ROIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the ROIRQ flag is 1.

RUIEN Bit (Receive Underflow Interrupt Enable)

This bit enables/disables the receive underflow interrupts.
While this bit is 1, an interrupt request is output when the SSISR.RUIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the RUIRQ flag is 1.

TOIEN Bit (Transmit Overflow Interrupt Enable)

This bit enables/disables the transmit overflow interrupts.
While this bit is 1, an interrupt request is output when the SSISR.TOIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the TOIRQ flag is 1.

TUIEN Bit (Transmit Underflow Interrupt Enable)

This bit enables/disables the transmit underflow interrupts.
While this bit is 1, an interrupt request is output when the SSISR.TUIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the TUIRQ flag is 1.

51.2.2 Status Register (SSISR)

Address(es): SSIE0.SSISR 8785 D004h, SSIE1.SSISR 8785 D104h, SSIE0_NS.SSISR 0785 D004h, SSIE1_NS.SSISR 0785 D104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b24 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle State Flag	0: Transfer state 1: Idle state	R
b26	ROIRQ	Receive Overflow Flag	0: No receive overflows occurred. 1: A receive overflow has occurred.	R/W
b27	RUIRQ	Receive Underflow Flag	0: No receive underflows occurred. 1: A receive underflow has occurred.	R/W
b28	TOIRQ	Transmit Overflow Flag	0: No transmit overflows occurred. 1: A transmit overflow has occurred.	R/W
b29	TUIRQ	Transmit Underflow Flag	0: No transmit underflows occurred. 1: A transmit underflow has occurred.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the SSIE status.

IIRQ Flag (Idle State Flag)

This flag indicates whether SSIE is in the idle state or transfer state.

[Setting condition]

- When a transmission of a frame is completed after changing the SSICR.TEN bit from 1 to 0.
- When a reception of a frame is completed after changing the SSICR.REN bit from 1 to 0.
- When a transmission of a frame is completed after changing the SSICR.TEN and SSICR.REN bits from 1 to 0.

[Clearing condition]

- When a start trigger is generated by writing a transmit data in the SSIFTDR register, while the SSICR.TEN bit is 1.
- When a start trigger is detected while the SSICR.REN bit is 1.
- When a start trigger is generated by writing a transmit data to the SSIFTDR register, while the SSICR.TEN and SSICR.REN bits are 1.

ROIRQ Flag (Receive Overflow Flag)

This flag indicates a receive overflow. This flag indicates that the data processing of the received data is delayed. The received data is not transferred to the receive FIFO if a receive overflow occurs. This flag is not cleared by resetting a receive FIFO.

[Priority order for setting and clearing]

Setting is prioritized.

[Setting condition]

- When a reception of the next data is completed while the receive FIFO is full.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.REN bit is changed from 0 to 1.

RUIRQ Flag (Receive Underflow Flag)

This flag indicates a receive underflow. This flag indicates that the SSIFRDR register is read while the receive FIFO is empty. Data read from the SSIFRDR register is invalid if a receive underflow occurs. This flag is not cleared by resetting a receive FIFO. Reading the SSIFRDR register while resetting the receive FIFO does not lead to setting of this flag.

[Priority order for setting and clearing]

Setting is prioritized.

[Setting condition]

- When the SSIFRDR register is read while the receive FIFO is empty.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.REN bit is changed from 0 to 1.

TOIRQ Flag (Transmit Overflow Flag)

This flag indicates a transmit overflow. This flag indicates that a data is written to the SSIFTDR register while the transmit FIFO is full. The written data is discarded. This flag is not cleared by resetting a transmit FIFO.

[Priority order for setting and clearing]

Setting is prioritized.

[Setting condition]

- When a data is written to the SSIFTDR register while the transmit FIFO is full.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.TEN bit is changed from 0 to 1.

TUIRQ Flag (Transmit Underflow Flag)

This flag indicates a transmit underflow. This flag indicates that supply of the transmit data was not enough to the transmission of the frame. The SSITXD pin becomes low if a transmit underflow occurs, and remain low even if the flag is cleared. This flag is not cleared by resetting a transmit FIFO.

[Priority order for setting and clearing]

Setting is prioritized.

[Setting condition]

- When a data transfer to the transmit shift register occurs while the transmit FIFO is empty.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.TEN bit is changed from 0 to 1.

51.2.3 FIFO Control Register (SSIFCR)

Address(es): SSIE0.SSIFCR 8785 D010h, SSIE1.SSIFCR 8785 D110h, SSIE0_NS.SSIFCR 0785 D010h, SSIE1_NS.SSIFCR 0785 D110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFRST	Receive FIFO Reset*1	0: The receive FIFO is released from the reset state. 1: The receive FIFO is reset.	R/W
b1	TFRST	Transmit FIFO Reset*1	0: The transmit FIFO is released from the reset state. 1: The transmit FIFO is reset.	R/W
b2	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b3	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b10 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	BSW	Byte Swap*1	0: Byte swapping is disabled 1: Byte swapping is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SSIRST	Software Reset	0: The software reset is released. 1: A software reset is applied.	R/W
b30 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	AUCKE	MCK Supply Enable*1	0: MCK is stopped. 1: MCK is supplied.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.

RFRST Bit (Receive FIFO Reset)

This bit is used to reset the receive FIFO. Writing 1 to this bit resets the receive FIFO and initializes the RDC[5:0] bits and RDF flag in the SSIFSR register. Since this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.
This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, this bit cannot be set to 1 when the SSIRST bit is 1.

TFRST Bit (Transmit FIFO Reset)

This bit is used to reset the transmit FIFO. Writing 1 to this bit resets the transmit FIFO and initializes the TDC[5:0] bits and TDE flag in the SSIFSR register. Since this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.
This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, this bit cannot be set to 1 when the SSIRST bit is set.

RIE Bit (Receive Data Full Interrupt Enable)

This bit enables or disables a receive data full interrupt. Set this bit to 1 after specifying the condition for generating a receive data full interrupt in the SSISCR.RDFS[4:0] bits.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit enables or disables a transmit data empty interrupt. Set this bit to 1 after specifying the condition for generating a transmit data empty interrupt in the SSISCR.TDES[4:0] bits.

BSW Bit (Byte Swap)

The BSW bit enables or disables byte swapping when the SSIFTDR is written and the SSIFRDR register is read. This bit is valid only when the SSIFTDR and SSIFRDR register are accessed in 16-bit or 32-bit units. For details, refer to Figure 51.5.

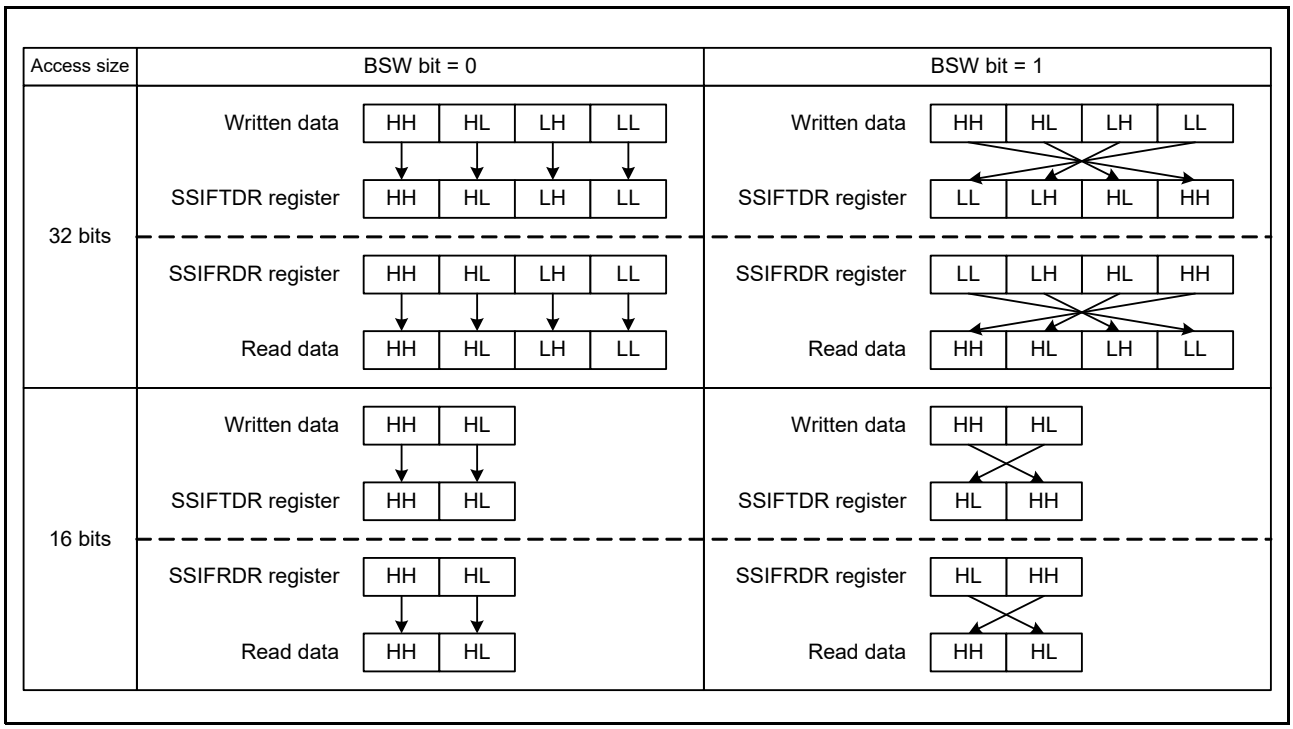


Figure 51.5 Examples of Byte-Swapping Operations

SSIRST Bit (Software Reset)

This bit is used to reset SSIE. Writing 1 to this bit initializes the internal state of SSIE. The registers and bits to be reset by setting this bit are listed below.

- SSICR.MUEN, TEN, and REN bits
- SSISR register
- SSIFCR.TFRST and RFRST bits
- SSIFSR register
- SSIFTDR register
- SSIFRDR register

Because this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.

Because writing 1 to this bit during data transfer leads to initialization by software reset regardless of the bit clock, be careful to stop data transfer by using this bit.

AUCKE Bit (MCK Supply Enable)

This bit enables/disables supply of MCK while in master-mode. The setting value of this bit is reflected at the timing shown in Figure 51.6.

Changes to settings related to MCK (the settings of the CKS, MST, BCKP, and CKDV[3:0] bits in the SSICR register) should only proceed while this bit is 0.

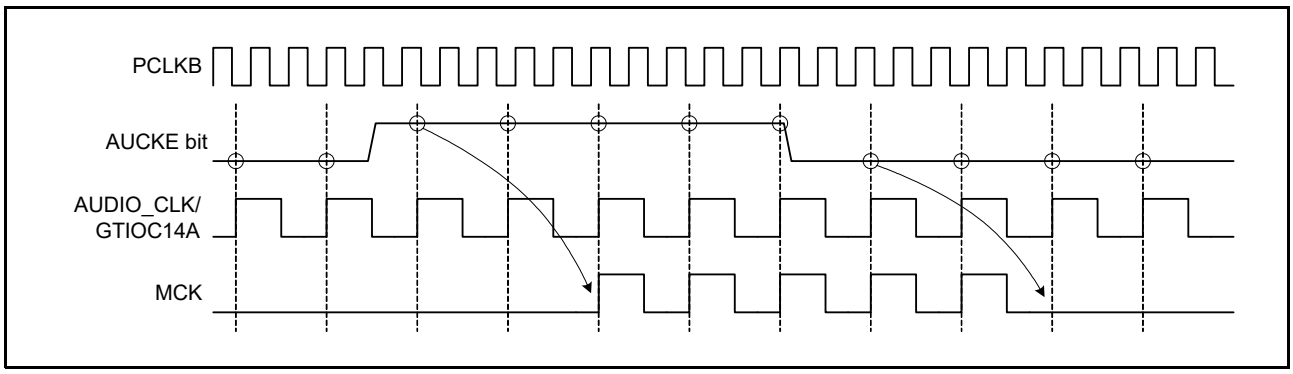


Figure 51.6 Supplying and Stopping MCK

51.2.4 FIFO Status Register (SSIFSR)

Address(es): SSIE0.SSIFSR 8785 D014h, SSIE1.SSIFSR 8785 D114h, SSIE0_NS.SSIFSR 0785 D014h, SSIE1_NS.SSIFSR 0785 D114h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RDF	Receive Data Full Flag	0: The number of received data stored in the receive FIFO does not exceed the value of the SSISCR.RDFS[4:0] bits. 1: The number of received data stored in the receive FIFO exceeds the value of the RDFS[4:0] bits.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	RDC[5:0]	Receive FIFO Data Count	Number of data stored in the receive FIFO	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: The number of spaces available in the transmit FIFO does not exceed the value of the SSISCR.TDES[4:0] bits 1: The number of spaces available in the transmit FIFO exceeds the value of TDES[4:0] bits.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29 to b24	TDC[5:0]	Transmit FIFO Data Count	Number of data stored in the transmit FIFO	R
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the status of the transmit FIFO and the receive FIFO.

RDF Flag (Receive Data Full Flag)

This flag indicates that the number of unread data in the receive FIFO is more than the value of the SSISCR.RDFS[4:0] bits.

[Priority order for setting and clearing]

Clearing is prioritized.

[Setting condition]

- When the number of data stored in the receive FIFO exceeds the value of the SSISCR.RDFS[4:0] bits.

[Clearing condition]*1

- When 0 is written to this flag after confirming this flag is 1.
- When a received data is read from the SSIFRDR register by DMA/DTC transfer (the last transfer of the block if using block transfer).

Note 1. This flag is cleared by a software reset and receive FIFO reset.

RDC[5:0] Bits (Receive FIFO Data Count)

These bits indicate the number of valid data that are stored in the receive FIFO. When this bit is 00h, there is no received data. When 20h, the receive FIFO is full and there is no space available.

TDE Flag (Transmit Data Empty Flag)

This flag indicates that the number of spaces available in the transmit FIFO is more than the value of the SSISCR.TDES[4:0] bits.
[Priority order for setting and clearing]
Clearing is prioritized.
[Setting condition]

- When the number of spaces available in the transmit FIFO exceeds the value of the SSISCR.TDES[4:0] bits.

[Clearing condition]*1

- When 0 is written to this flag after confirming this flag is 1.
- When a transmit data is written to the SSIFTDR register by DMA/DTC transfer (the last transfer of the block if using block transfer).

Note 1. This flag is cleared by a software reset and transmit FIFO reset.

TDC[5:0] Bits (Transmit FIFO Data Count)

These bits indicate the number of valid data that are stored in the transmit FIFO. When this bit is 00h, there is no data to be transmitted. When 20h, the transmit FIFO is full and there is no space available.

51.2.5 Transmit FIFO Data Register (SSIFTDR)

- When accessing in longword size

Address(es): SSIE0.SSIFTDR 8785 D018h, SSIE1.SSIFTDR 8785 D118h, SSIE0_NS.SSIFTDR 0785 D018h, SSIE1_NS.SSIFTDR 0785 D118h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): SSIE0.SSIFTDR.H 8785 D018h, SSIE1.SSIFTDR.H 8785 D118h,
SSIE0_NS.SSIFTDR.H 0785 D018h, SSIE1_NS.SSIFTDR.H 0785 D118h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
									H							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): SSIE0.SSIFTDR.HH 8785 D018h, SSIE1.SSIFTDR.HH 8785 D118h,
SSIE0_NS.SSIFTDR.HH 0785 D018h, SSIE1_NS.SSIFTDR.HH 0785 D118h

	b7	b6	b5	b4	b3	b2	b1	b0
					HH			
Value after reset:	0	0	0	0	0	0	0	0

This register is a write-only register used to write a transmit data to the 32-stage and 4-byte-wide transmit FIFO. Access to this register according to the access size listed in Table 51.6.

Table 51.6 Access Size to the SSIFTDR and SSIFRDR Registers

		Access Size		
SSICR.DWL[2:0] Bits	**Data Word Length**	**Byte**	**Word**	**Longword**
000b	8	Available	Not available	Not available
001b	16	Not available	Available	Not available
010b	18	Not available	Not available	Available
011b	20	Not available	Not available	Available
100b	22	Not available	Not available	Available
101b	24	Not available	Not available	Available
110b	32	Not available	Not available	Available

51.2.6 Receive FIFO Data Register (SSIFRDR)

- When accessing in longword size

Address(es): SSIE0.SSIFRDR 8785 D01Ch, SSIE1.SSIFRDR 8785 D11Ch, SSIE0_NS.SSIFRDR 0785 D01Ch, SSIE1_NS.SSIFRDR 0785 D11Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): SSIE0.SSIFRDR.H 8785 D01Ch, SSIE1.SSIFRDR.H 8785 D11Ch,
SSIE0_NS.SSIFRDR.H 0785 D01Ch, SSIE1_NS.SSIFRDR.H 0785 D11Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): SSIE0.SSIFRDR.HH 8785 D01Ch, SSIE1.SSIFRDR.HH 8785 D11Ch,
SSIE0_NS.SSIFRDR.HH 0785 D01Ch, SSIE1_NS.SSIFRDR.HH 0785 D11Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	HH							
Value after reset:	0	0	0	0	0	0	0	0

This register is a read-only register used to read a received data from the 32-stage and 4-byte-wide receive FIFO. Access to this register according to the access size listed in Table 51.6.

51.2.7 Audio Format Register (SSIOFR)

Address(es): SSIE0.SSIOFR 8785 D020h, SSIE1.SSIOFR 8785 D120h, SSIE0_NS.SSIOFR 0785 D020h, SSIE1_NS.SSIOFR 0785 D120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMOD[1:0]	Data Format Select*1	b1 b0 0 0: I²S, left-justified, or right-justified format 0 1: TDM format 1 0: Monaural format 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	LRCONT	LRCK Continuous Output Enable*2, *3	0: LRCK stops in the idle state. 1: Output of LRCK continues in the idle state.	R/W
b9	BCKASTP	BCK Continuous Output Disable*2, *3	0: BCK is always output. 1: BCK stops in idle state.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.
Note 2. This bit is valid only in master-mode. The setting is invalid in slave-mode.
Note 3. Do not set both of the BCKASTP and LRCONT bits to 1.

This register is used to set an audio format (which includes the settings of data format, LR clock continuous output mode, and BCK continuous output disable).

LRCONT Bit (LRCK Continuous Output Enable)

This bit enables or disables the output from SSILRCK pin when SSIE is in the idle state. This bit is effective only in master mode.

BCKASTP Bit (BCK Continuous Output Disable)

This bit enables or disables the output from SSIBCK pin when SSIE is in the idle state. This bit is effective only in master mode.

Set this bit to 0 to output BCK from the SSIBCK pin before starting data transfer. If this bit is set to 1 during data transfer, BCK output from the SSIBCK pin stops when the SSIE enters the idle state.

Set this bit after the data format to be used is specified. Do not rewrite this bit while the supply of MCK stops.

51.2.8 FIFO Status Control Register (SSISCR)

Address(es): SSIE0.SSISCR 8785 D024h, SSIE1.SSISCR 8785 D124h, SSIE0_NS.SSISCR 0785 D024h, SSIE1_NS.SSISCR 0785 D124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RDFS[4:0]	Receive Data Full Condition Setting*1	Specifies the condition for setting the RDF flag to 1. b4 b0 00000: When receive FIFO has unread received data. 00001: When receive FIFO has two or more unread received data. ... 11110: When receive FIFO has 31 or more unread received data. 11111: When receive FIFO is full.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	TDES[4:0]	Transmit Data Empty Condition Setting*1	Specifies the condition for setting the TDE flag to 1. b12 b8 00000: When transmit FIFO has space available. 00001: When transmit FIFO has at least two stages available. ... 11110: When transmit FIFO has at least 31 stages available. 11111: When transmit FIFO is empty.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.

51.3 Operation

51.3.1 Data Formats

SSIE supports five data formats. Table 51.7 lists the supported data formats.

Table 51.7 Supported Data Formats and Bit Setting

Data Format	SSIOFR.OMOD[1:0] Bits	SSICR.DEL Bit	SSICR.LRCKP Bit	SSICR.SDTA Bit
I²S format	00b	0	0	0
Left-justified format	00b	1	1	0
Right-justified format	00b	1	1	1
TDM format	01b	0 or 1	1	0 or 1
Monaural format	10b	0 or 1	1	0 or 1

The following describes the serial data structure shared by data formats.
A serial data structure is defined by the system word length (set in the SSICR.SWL[2:0] bits) and the data word length (set in the SSICR.DWL[2:0] bits). If the data word length is shorter than the system word length, padding bits are inserted (Figure 51.7).

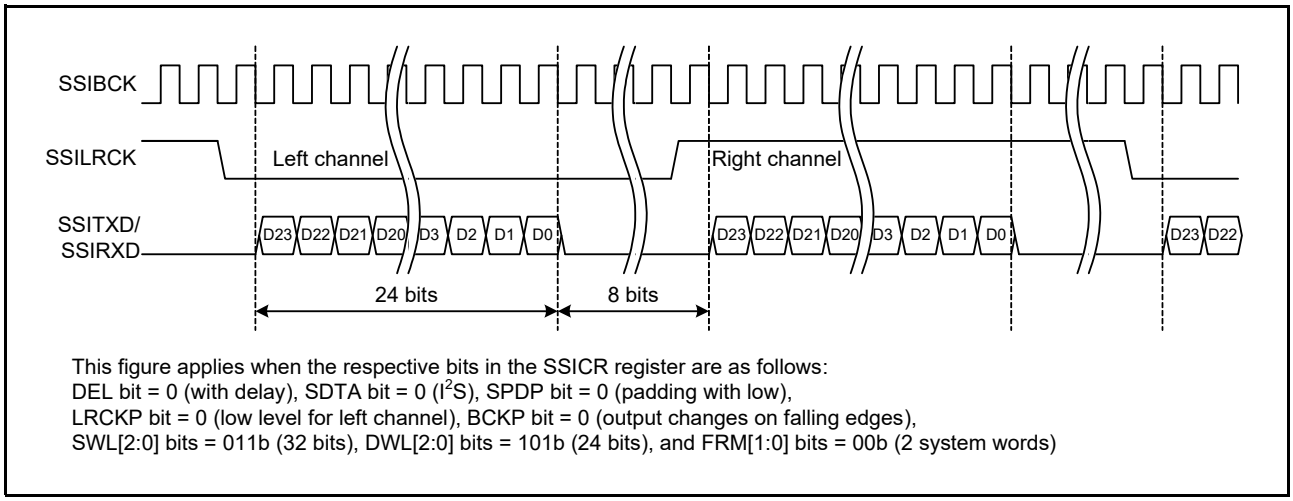


Figure 51.7 Example of Padding Bit Insertion (I²S Format, System Word Length > Data Word Length)

The number of padding bits is calculated from the system word length minus the data word length.
Table 51.8 lists the number of padding bits to be inserted with each combination of system word length (SWL[2:0] bits) and data word length (DWL[2:0] bits).

Table 51.8 Number of Padding Bits

System Word Length (SSICR.SWL[2:0] Bits)	Data Word Length (SSICR.DWL[2:0] Bits)						
	8 (000b)	16 (001b)	18 (010b)	20 (011b)	22 (100b)	24 (101b)	32 (110b)
8 (000b)	0	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting
16 (001b)	8	0	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting
24 (010b)	16	8	6	4	2	0	Prohibited setting
32 (011b)	24	16	14	12	10	8	0
48 (100b)	40	32	30	28	26	24	16
64 (101b)	56	48	46	44	42	40	32
128 (110b)	120	112	110	108	106	104	96
256 (111b)	248	240	238	236	234	232	224

51.3.1.1 I²S Format

The I²S format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes low for left channel and high for right channel. The effective data are output one bit clock period after the SSILRCK changes and then the padding bits are output.
Figure 51.8 shows an example of I²S format.

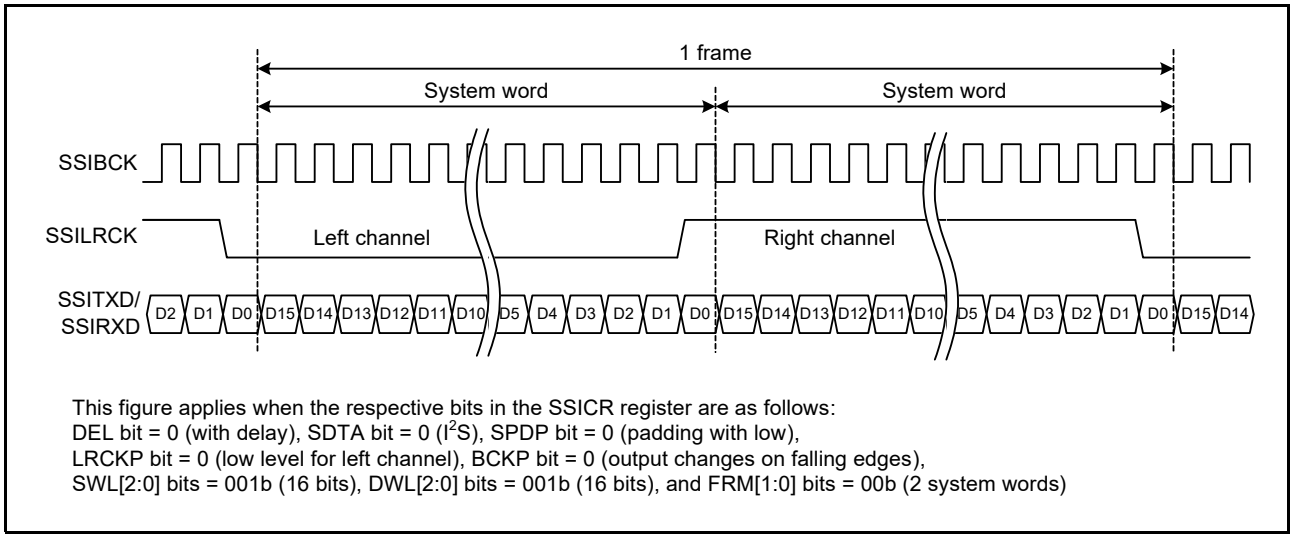


Figure 51.8 Example of I²S Format

51.3.1.2 Left-Justified Format

The left-justified format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes high for left channel and low for right channel. The effective data are output at the same time as the SSILRCK changes and then the padding bits are output.
Figure 51.9 shows an example of left-justified format.

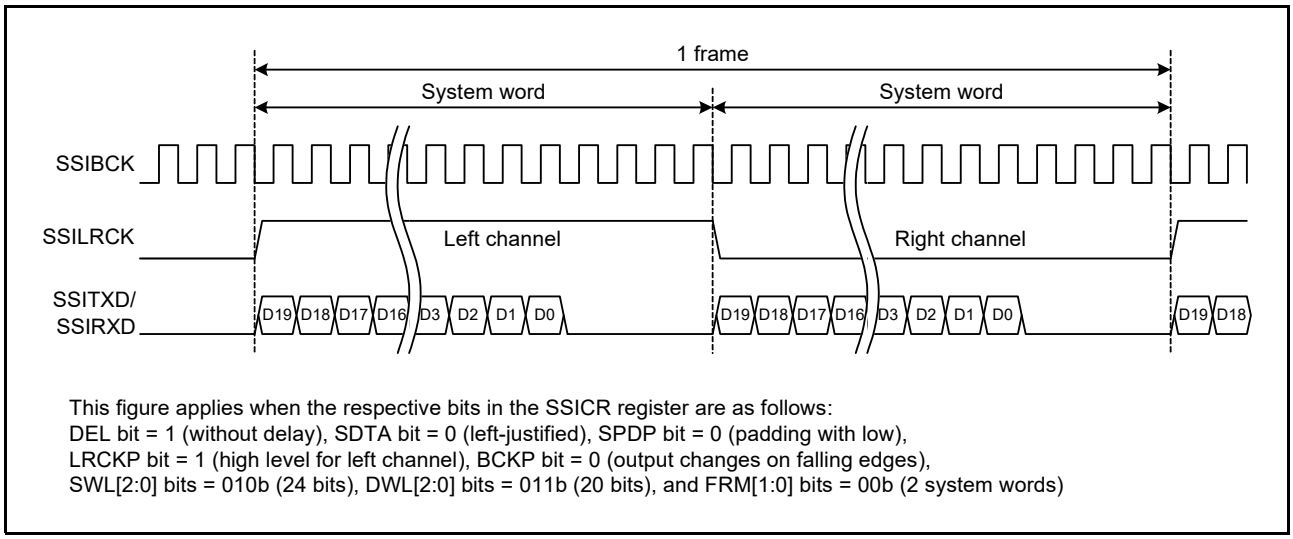


Figure 51.9 Example of Left-Justified Format

51.3.1.3 Right-Justified Format

The right-justified format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes high for left channel and low for right channel. The padding bits are output at the same time as the SSILRCK changes and then the effective data are output.
Figure 51.10 shows an example of right-justified format.

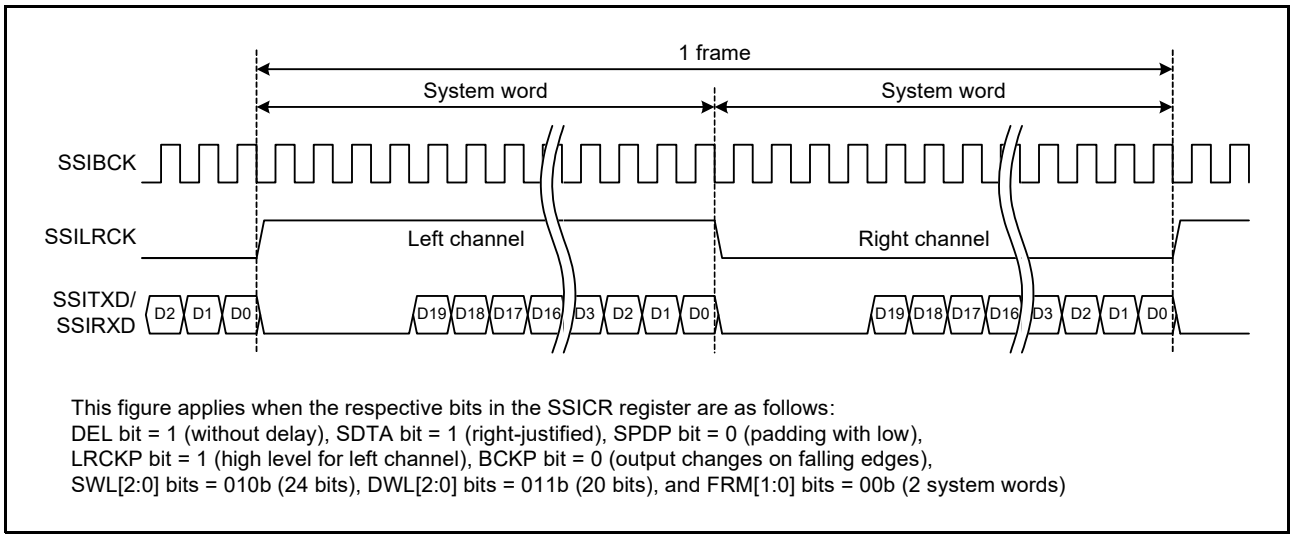


Figure 51.10 Example of Right-Justified Format

51.3.1.4 Monaural Format

The monaural format configures a frame with one system word. The SSILRCK signal is used as a transfer start trigger. Figure 51.11, Figure 51.12, and Figure 51.13 show examples of monaural formats with 0 as the value of the SSICR.DEL bit and no padding bits, with 0 as the value of the DEL bit and padding bits, and with 1 as the value of the DEL bit and no padding bits, respectively.

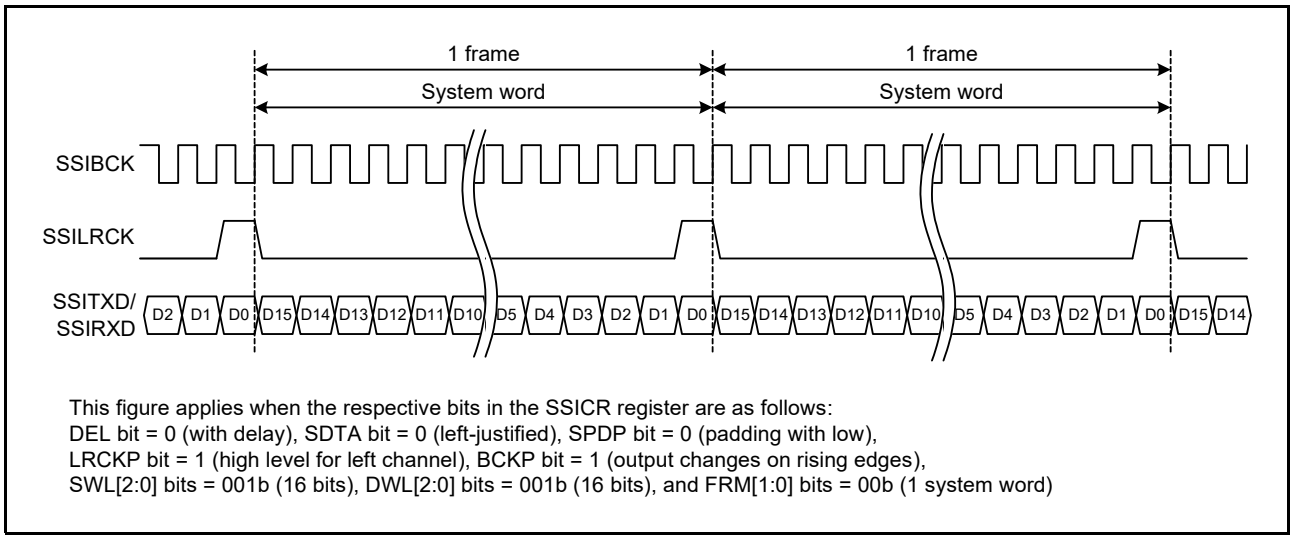


Figure 51.11 Example of Monaural Format (1): SSICR.DEL Bit = 0 and No Padding

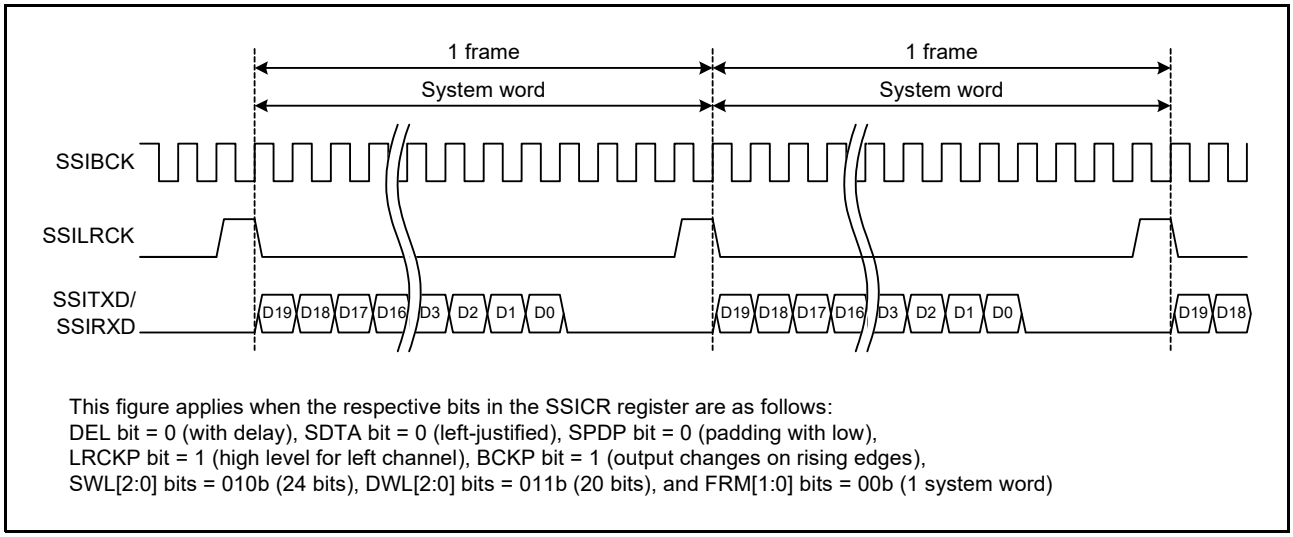


Figure 51.12 Example of Monaural Format (2): SSICR.DEL Bit = 0 and Padding Bits

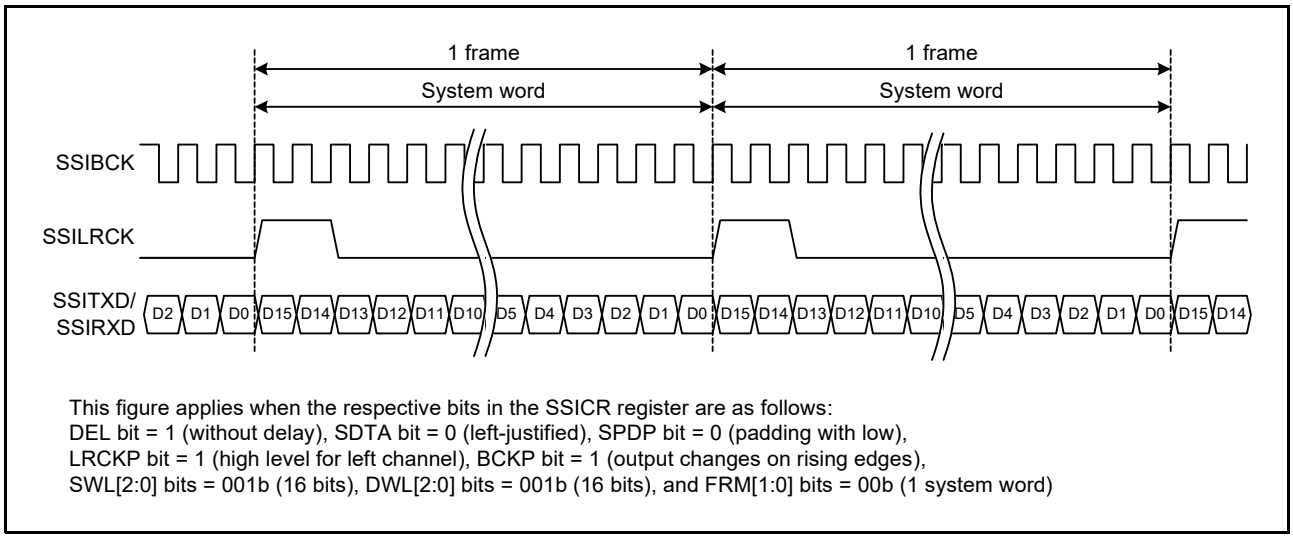


Figure 51.13 Example of Monaural Format (3): SSICR.DEL Bit = 1 and No Padding

51.3.1.5 TDM Format

The TDM format configures a frame with four to eight system words. The SSILRCK signals become high for the first system word and low for the other system words. The SSILRCK signal is defined as the synchronous pulse and its rising edge means a start of a frame.

Figure 51.14 and Figure 51.15 show examples of the TDM formats with 0 as the value of the SSICR.DEL bit and no padding bits, and with 1 as the value of the SSICR.DEL bit and padding bits, respectively.

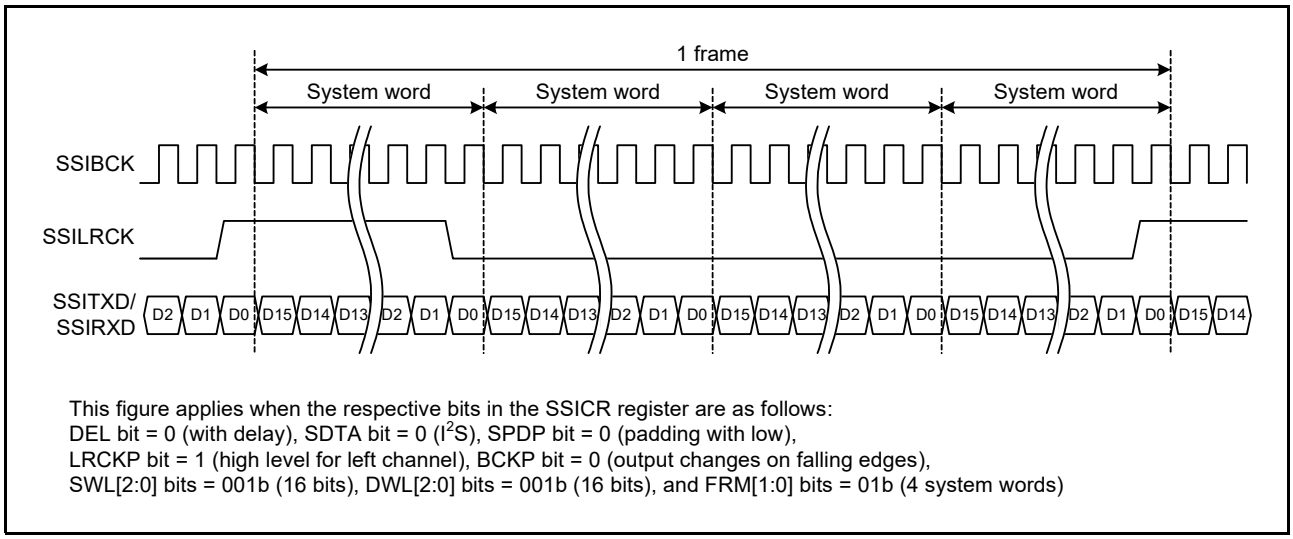
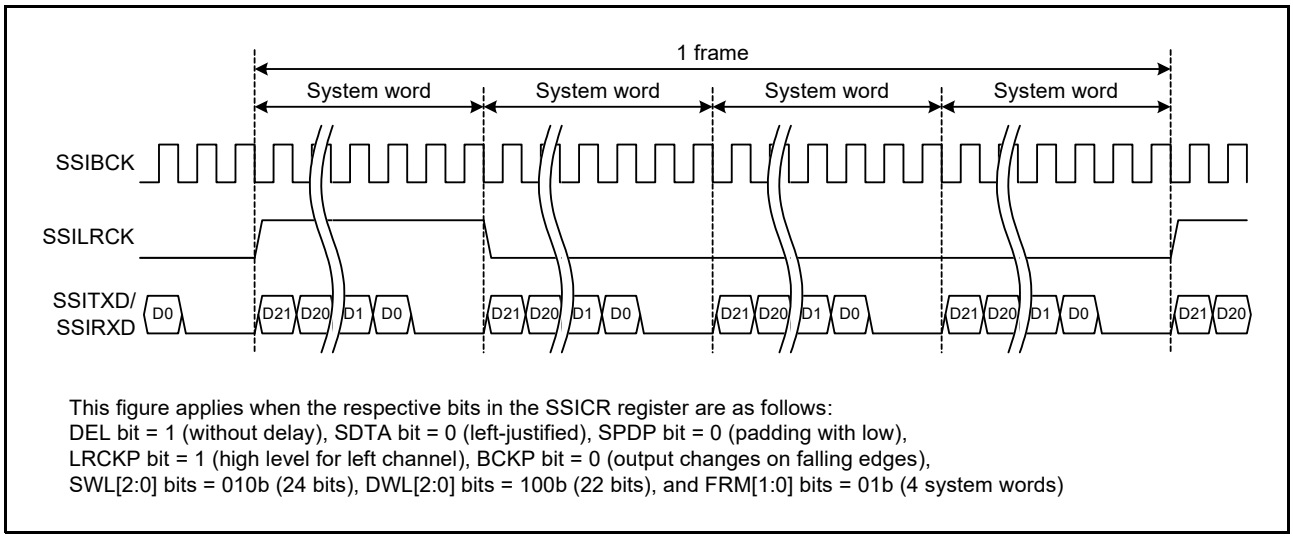


Figure 51.14 Example of TDM Format (1): SSICR.DEL Bit = 0 and No Padding



This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 1 (without delay), SDTA bit = 0 (left-justified), SPDP bit = 0 (padding with low),
LRCKP bit = 1 (high level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 010b (24 bits), DWL[2:0] bits = 100b (22 bits), and FRM[1:0] bits = 01b (4 system words)

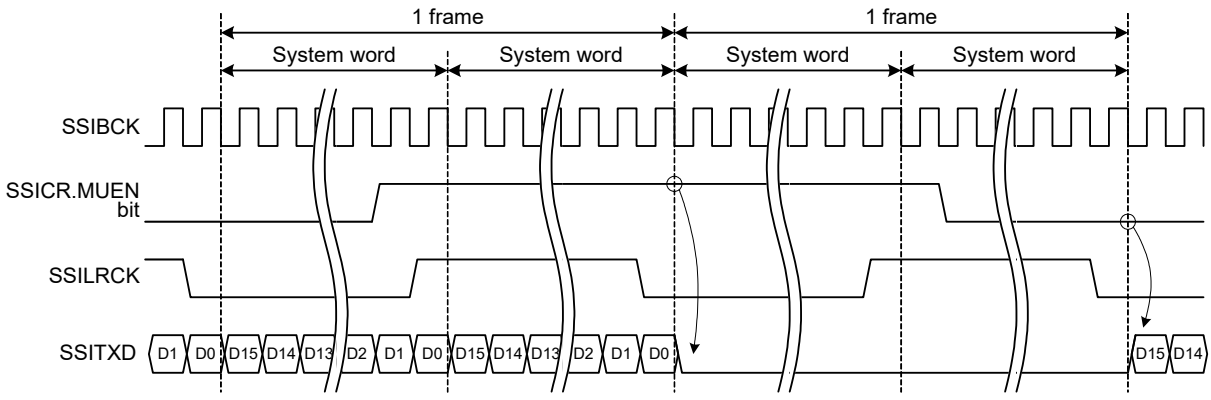
Figure 51.15 Example of TDM Format (2): SSICR.DEL Bit = 1 and Padding Bits

51.3.2 Output Control

SSIE has functions to control the outputs of data and clocks such as mute function, LRCK continuous output function, and BCK continuous output disable function.

51.3.2.1 Mute Function

The output from the SSITXD pin can be fixed to the low level by using the mute function. If the SSICR.MUEN bit is changed during a frame, the setting takes effect at the next frame boundary. Even while the MUEN bit is set to 1, status flags are updated and interrupt request signals are generated.
Figure 51.16 shows an example of the mute operation.



This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 001b (16 bits), DWL[2:0] bits = 001b (16 bits), and FRM[1:0] bits = 00b (2 system words)

Figure 51.16 Example of Mute Operation

51.3.2.2 LRCK Continuous Output Function

LRCK can be output from the SSILRCK pin even in the idle state by using the LRCK continuous output function. It is controlled by the SSIOFR.LRCONT bit.

Figure 51.17 shows an example of the operation with LRCK continuous output function.

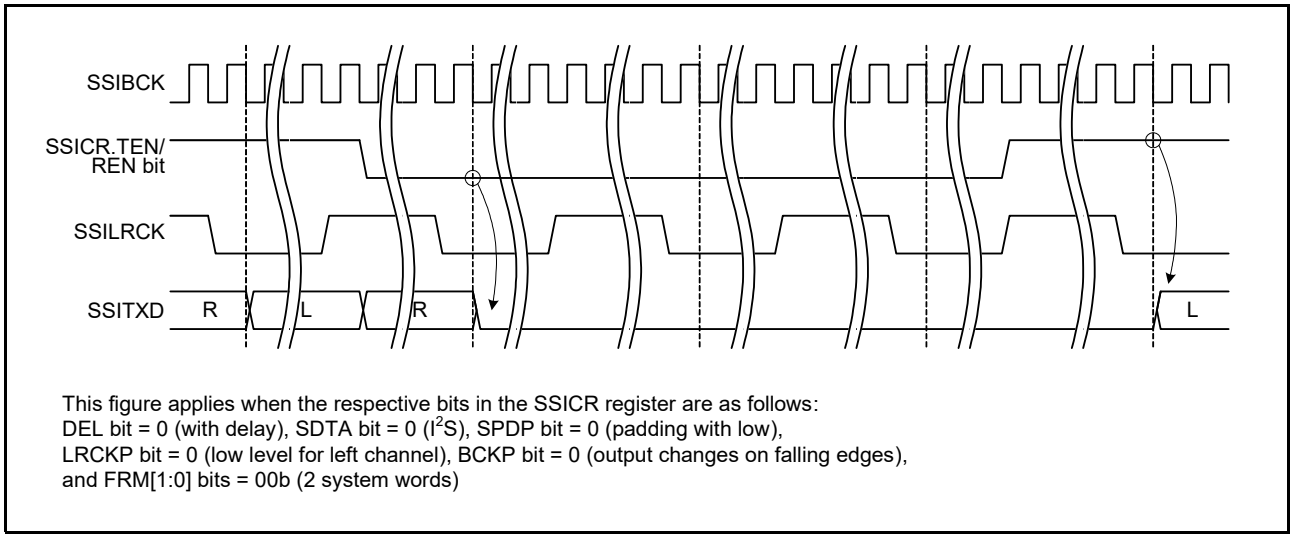


Figure 51.17 Example of Operation with LRCK Continuous Output Function

51.3.2.3 BCK Continuous Output Disable Function

BCK output from the SSIBCK pin can be stopped in the idle state by using the BCK continuous output disable function. It is controlled by the SSIOFR.BCKASTP bit. When the BCKASTP bit is set to 1, BCK stops in the idle state.

Set the BCKASTP bit to 0 and then start data transfer. If this bit is set to 1 during data transfer, BCK stops when SSIE enters the idle state. To resume the data transfer, set the BCKASTP bit to 0 to output BCK and then set the SSICR.TEN/REN bit to 1.

Figure 51.18 shows an example of the operation with BCK continuous output disable function.

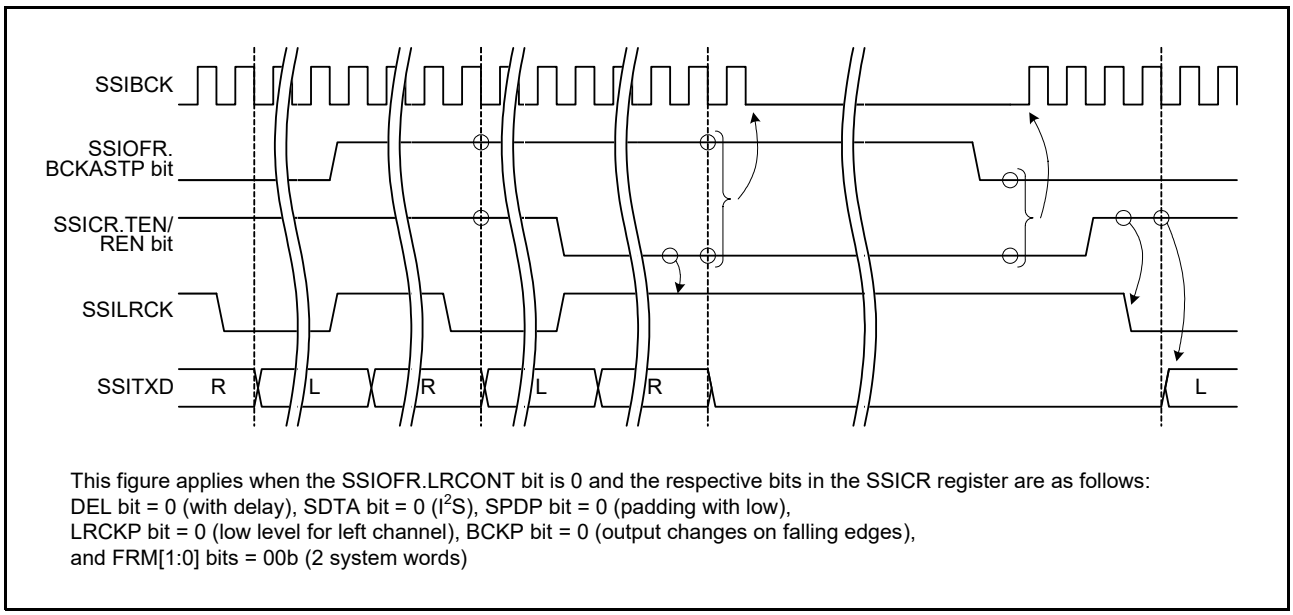


Figure 51.18 Example of Operation with BCK Continuous Output Disable Function

51.3.3 Transfer Modes

SSIE supports the transfer modes listed in Table 51.9. Table 51.10 lists the control bits valid in each transfer mode.

Table 51.9 Transfer Modes

Transfer Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transception	1	1	1

Table 51.10 Control Bits Valid in Each Transfer Mode*1

Control Bit		Transfer Mode					
Register	Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transception	Master-mode Reception	Master-mode Transmission	Master-mode Transception
SSICR	CKS	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	CKDV[3:0]	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	MUEN	Ineffective	Effective	Effective	Ineffective	Effective	Effective
SSIFCR	AUCKE	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	TIE	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RIE	Effective	Ineffective	Effective	Effective	Ineffective	Effective
	TFRST	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RFRST	Effective	Ineffective	Effective	Effective	Ineffective	Effective
SSIOFR	BCKASTP	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	LRCONT	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	OMOD[1:0]	Effective	Effective	Effective	Effective	Effective	Effective
SSISCR	TDES[4:0]	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RDFS[4:0]	Effective	Ineffective	Effective	Effective	Ineffective	Effective

Note 1. “Ineffective” means that the setting of the bits does not affect operation.

51.3.3.1 Slave Mode

The SSIE operates in slave mode when the SSICR.MST bit is 0. The SSIBCK and SSILRCK signals must be supplied from an external device.

51.3.3.2 Master Mode

The SSIE operates in master mode when the SSICR.MST bit is 1. The SSIBCK and SSILRCK signals are internally generated from the master clock (MCK).

51.3.3.3 Transmission

The SSIE operates as a transmitter when the SSICR.TEN bit is 1 and the SSICR.REN bit is 0. Receive circuits do not operate.

51.3.3.4 Reception

The SSIE operates as a receiver when the SSICR.TEN bit is 0 and the SSICR.REN bit is 1. Transmit circuits do not operate.

51.3.3.5 Transception

The SSIE operates as a transceiver when both of the SSICR.TEN and SSICR.REN bits are 1.

51.3.4 State Transition

SSIE has the two main operation states: idle state and transfer state. The SSICR.TEN and SSICR.REN bits switch between these states. The SSISR.IIRQ flag can be used to confirm in which state SSIE is.
Figure 51.19 shows the state transition of the SSIE.

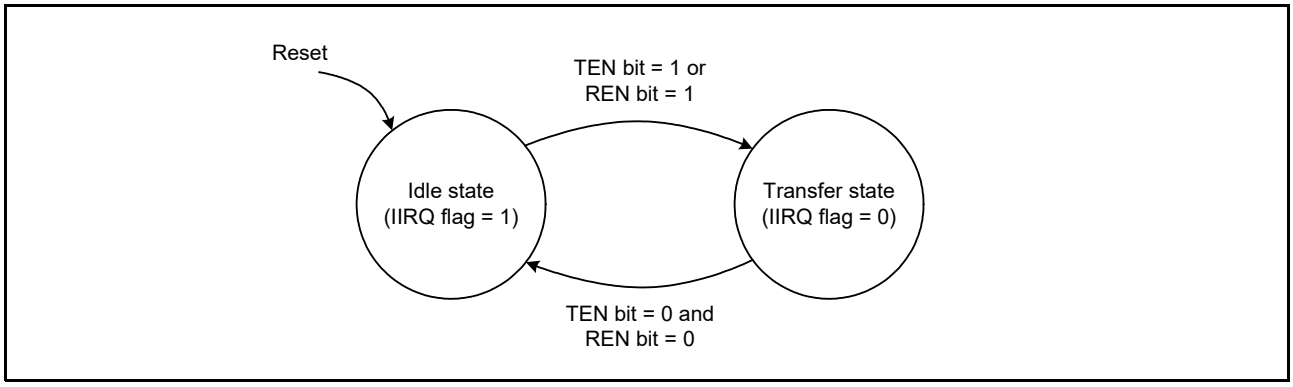


Figure 51.19 State Transition of the SSIE

51.3.4.1 Idle State

In the idle state, SSIE stops data transfer. When both or either of the SSICR.TEN and SSICR.REN bits are/is set to 1, the SSIE enters the transfer state.
Outputs of the BCK and LRCK in master mode can be controlled by the SSIOFR.BCKASTP and SSIOFR.LRCONT bits, respectively. For details, refer to Table 51.11.

Table 51.11 Output of Clock/Data in the Idle State

SSICR.MST Bit	SSIOFR.BCKASTP Bit	SSIOFR.LRCONT Bit	SSIBCK	SSILRCK	SSITXD
0	Invalid	Invalid	— (input)	— (input)	Stop
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop
1	1	1	Stop	Supply	Stop

51.3.4.2 Transfer States

In this state, SSIE continues data transfer. When the SSICR.TEN and SSICR.REN bit is set to 0, SSIE enters the idle state.

For details, refer to Figure 51.20.

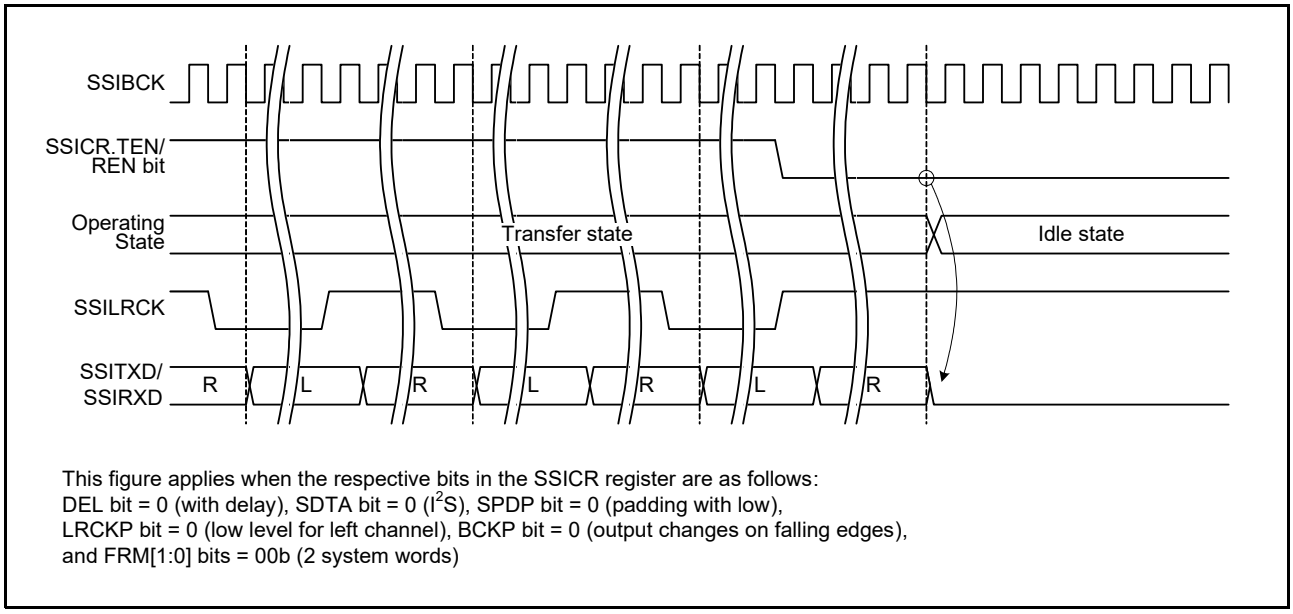


Figure 51.20 Transfer State and Transition to Idle State

51.3.5 SSIE Initialization

Figure 51.21 shows an example of the flow to initialize SSIE.

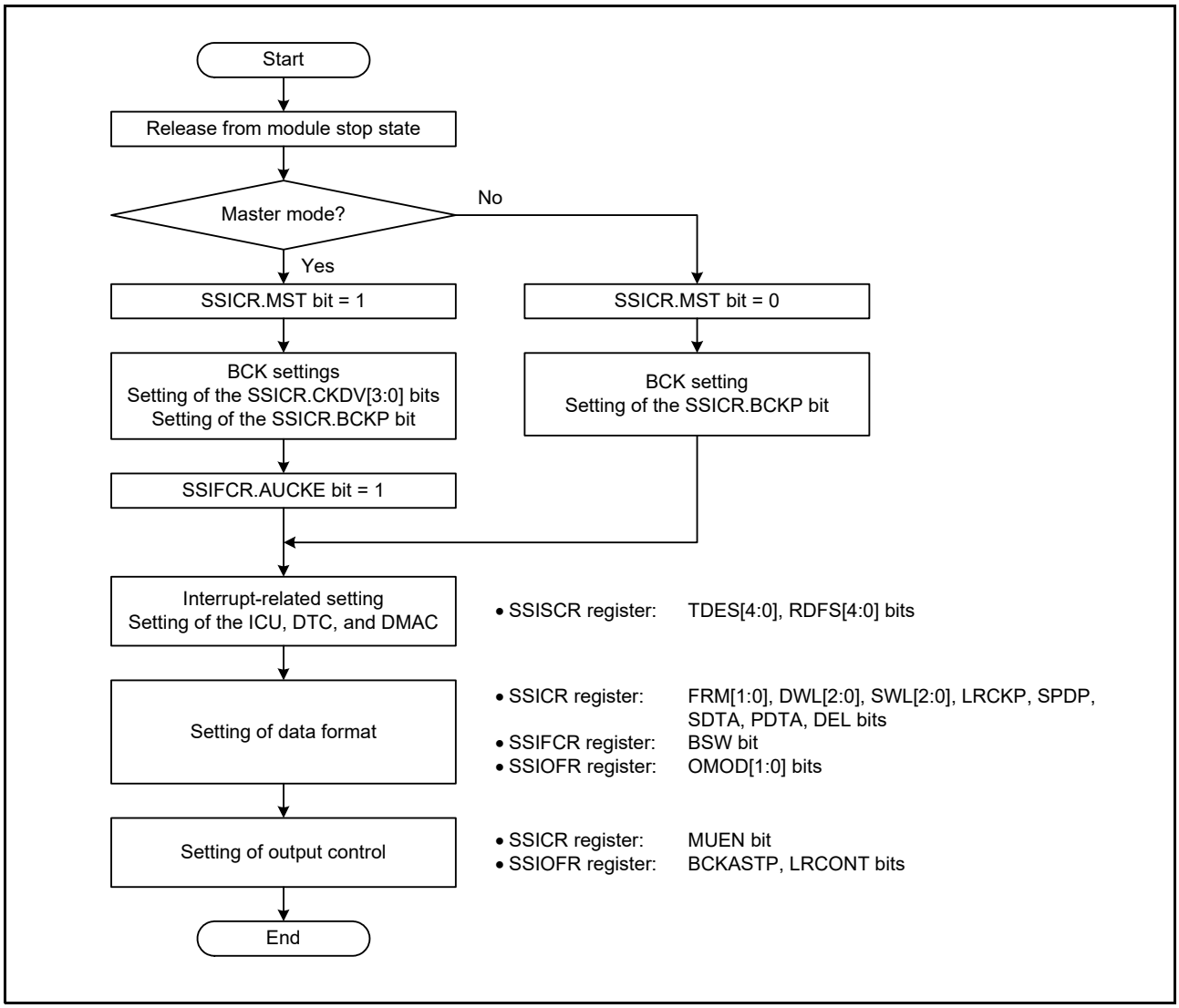


Figure 51.21 Example Flowchart of SSIE Initialization

51.3.6 Transmission

Figure 51.22 shows an example of the flow to transmit data.
When the SSICR.TEN bit is set to 1 after setting the data for one frame in the SSIFTDR register, the SSIE starts transmission. When the number of spaces available in the transmit FIFO reaches the value specified by the SSISCR.TDES[4:0] bits while the SSIFCR.TIE bit is 1, a transmit data empty interrupt occurs. The number of data to be written to the SSIFTDR register should be determined in accordance with the space available in the transmit FIFO specified by the TDES[4:0] bits.

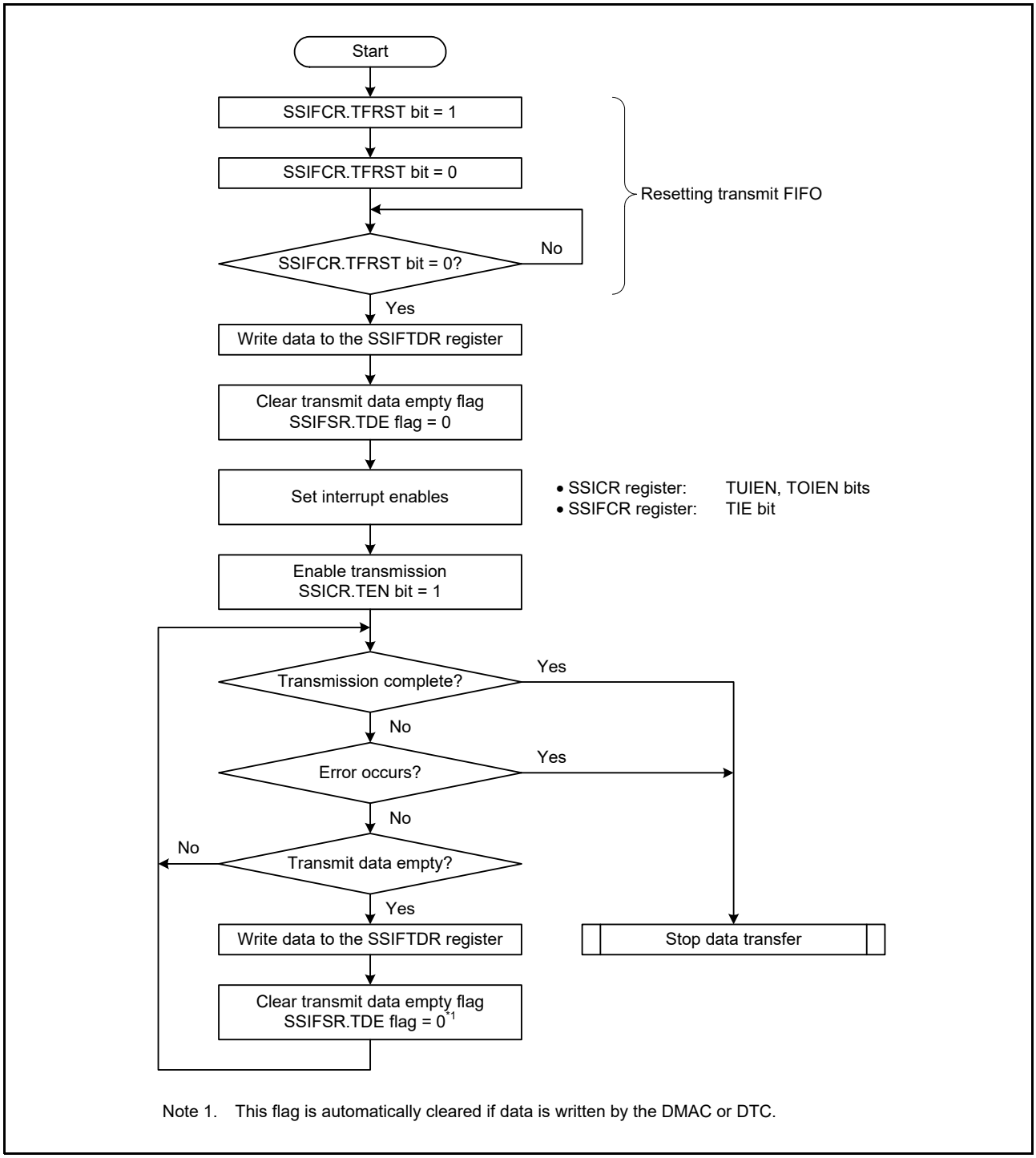
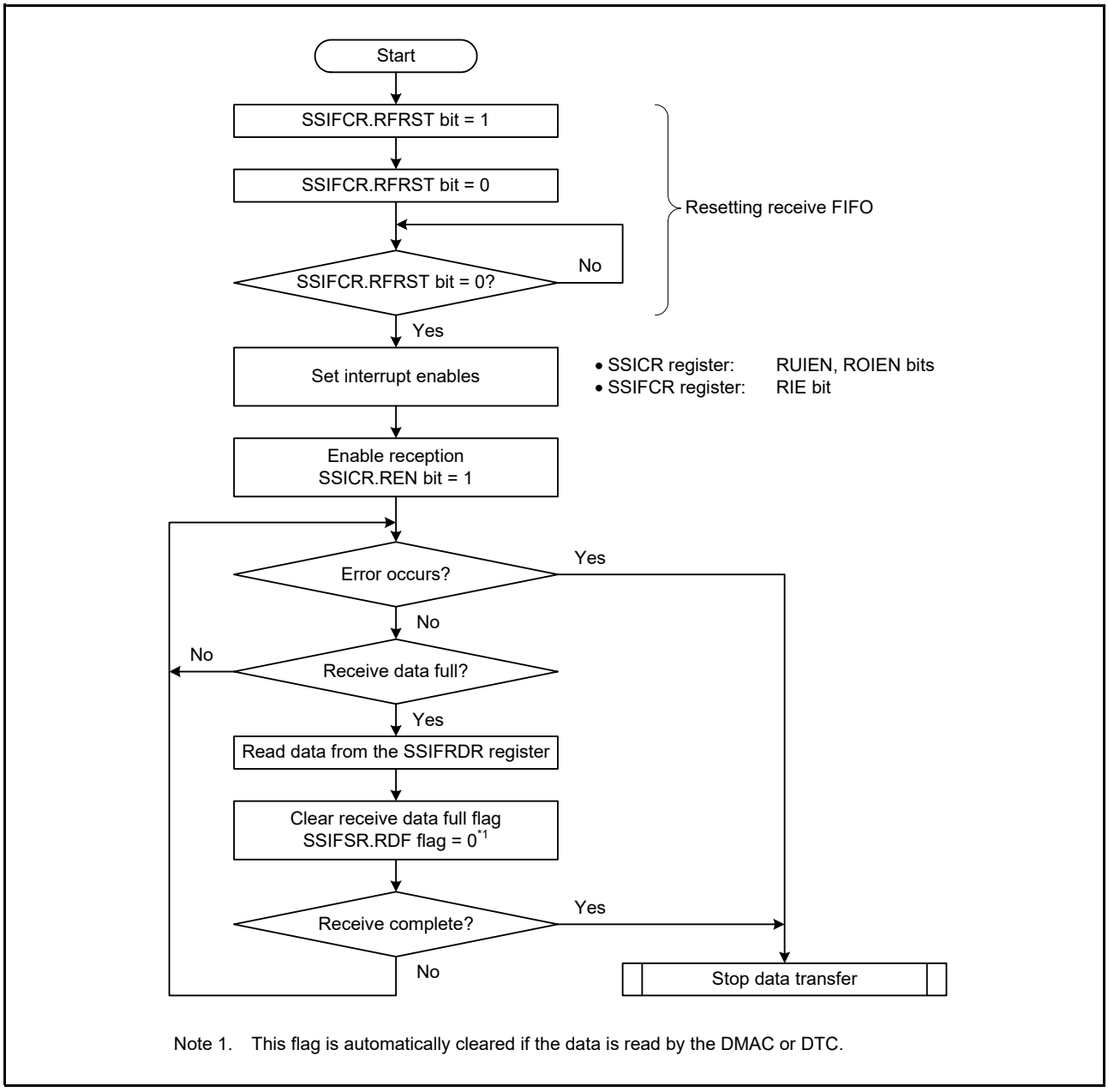


Figure 51.22 Example Flowchart of Transmission

51.3.7 Reception

Figure 51.23 shows an example of the flow to receive data.
When the SSICR.REN bit is set to 1, SSIE starts reception. When the number of data stored in the receive FIFO reaches to the value specified by the SSISCR.RDFS[4:0] bits while the SSIFCR.RIE bit is 1, a receive data full interrupt occurs. The number of data to be read from the SSIFRDR register should be determined in accordance with the number of data stored in the receive FIFO specified by the RDFS[4:0] bits.



Note 1. This flag is automatically cleared if the data is read by the DMAC or DTC.

Figure 51.23 Example Flowchart of Reception

51.3.8 Stop Data Transfer

Figure 51.24 shows an example of the flow to stop data transfer.

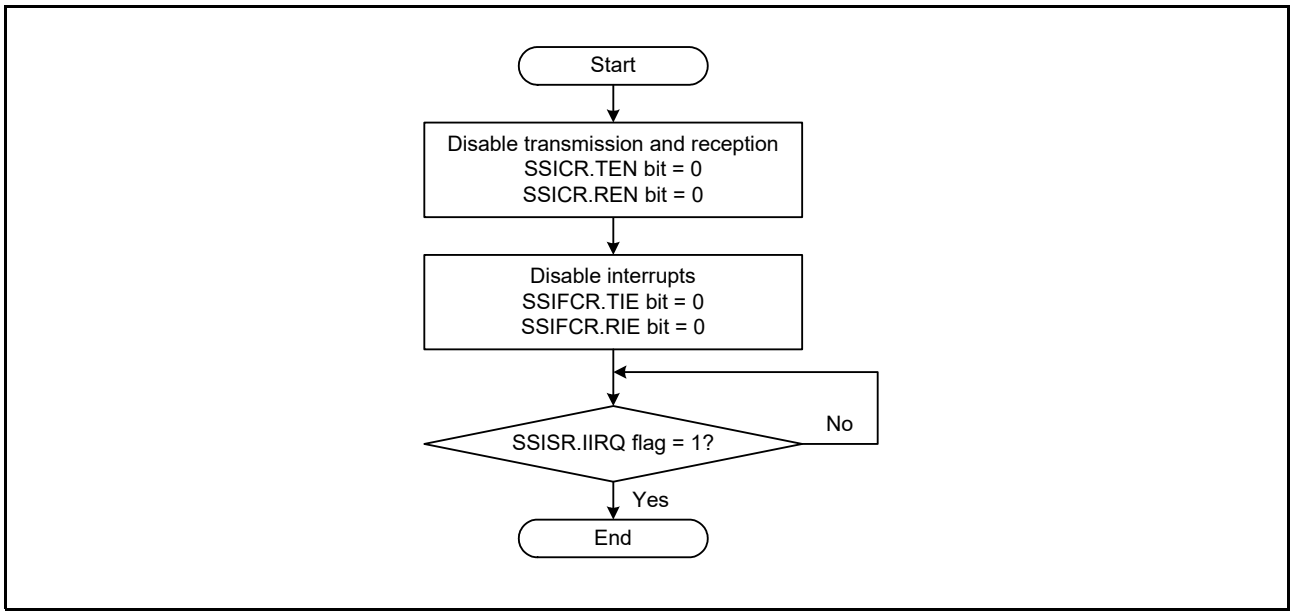


Figure 51.24 Example Flowchart of Stopping Data Transfer

To resume data transfer, conform with the example flowcharts of transmission and reception in Figure 51.22 and Figure 51.23, respectively. When changing the settings of clocks or switching a transfer mode between master and slave, start from initialization.

51.3.9 Error Processing

SSIE has the following four errors.

- Transmit underflow
- Transmit overflow
- Receive underflow
- Receive overflow

If an error has occurred, follow the procedure to stop data transfer shown in Figure 51.24 and then clear the corresponding flag. When an error has occurred while the corresponding interrupt enable bit in the SSICR register is 1, an interrupt is generated.

(1) Transmit Underflow

When a data transfer from the transmit FIFO to transmit shift register has performed while the transmit FIFO is empty, a transmit underflow is informed. If a transmit underflow has occurred, the SSITXD pin is driven low.

(2) Transmit Overflow

When a data is written to the SSIFTDR register while the transmit FIFO is full, a transmit overflow is informed. The data to be written is discarded.

(3) Receive Underflow

When a data is read from the SSIFRDR register while the receive FIFO is empty, a receive underflow is informed. The read data is undefined.

(4) Receive Overflow

When a data transfer from receive shift register to the receive FIFO has performed while the receive FIFO is full, a receive overflow is informed. The data to be transferred is discarded and is not stored in the receive FIFO.

51.4 Interrupts

Table 51.12 lists the interrupt sources. Figure 51.25 shows an block diagram for interrupts.

Table 51.12 SSIE Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Request Flag	Interrupt Enable Bit	DMAC/DTC Triggerable
SSIE0	SSIF0	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle state interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	No
	SSIRXI0	Receive data full interrupt	SSIFSR.RDF	SSIFCR.RIE	Yes
	SSITXI0	Transmit data empty interrupt	SSIFSR.TDE	SSIFCR.TIE	Yes
SSIE1	SSIF1	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle state interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	No
	SSIRXI1	Receive data full interrupt	SSIFSR.RDF	SSIFCR.RIE	Yes
	SSITXI1	Transmit data empty interrupt	SSIFSR.TDE	SSIFCR.TIE	Yes

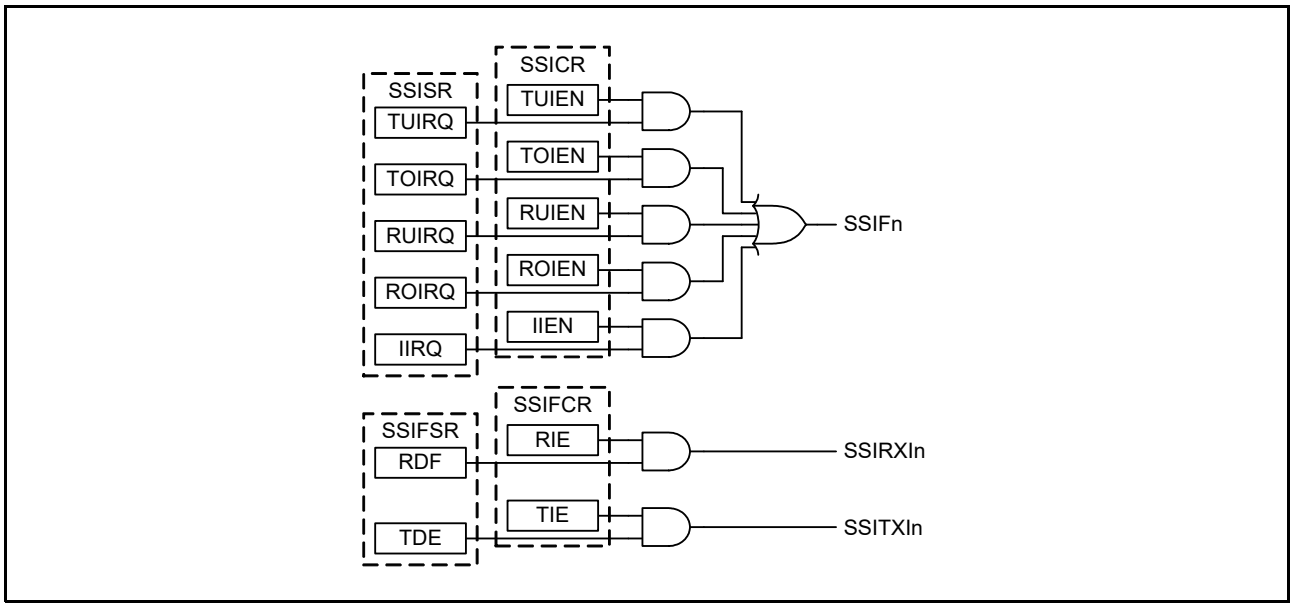


Figure 51.25 Interrupt System Block Diagram (n = 0, 1)

51.4.1 SSIFn Interrupt

Each SSIFn interrupt combines five interrupt sources (n = 0, 1). Each interrupt source has a status flag and an interrupt enable bit, so the interrupt can be enabled or disabled individually.

If an SSIFn interrupt occurs, read the SSISR register in the interrupt handling routine to check the source of the interrupt request, and then perform proper operation. To clear the interrupt request, set the corresponding interrupt enable bit to 0 or clear the interrupt request flag.

51.4.2 SSITXIn Interrupt

The SSITXIn interrupt (channel n transmit data empty interrupt) is an edge detection interrupt. This interrupt request is generated when either of the following conditions is satisfied.

- The SSIFSR.TDE flag has changed from 0 to 1 while the SSIFCR.TIE bit is 1.
- The SSIFCR.TIE bit has been changed from 0 to 1 while the SSIFSR.TDE flag is 1.

Figure 51.26 shows a timing chart when an interrupt request is generated when the TDE flag changes.

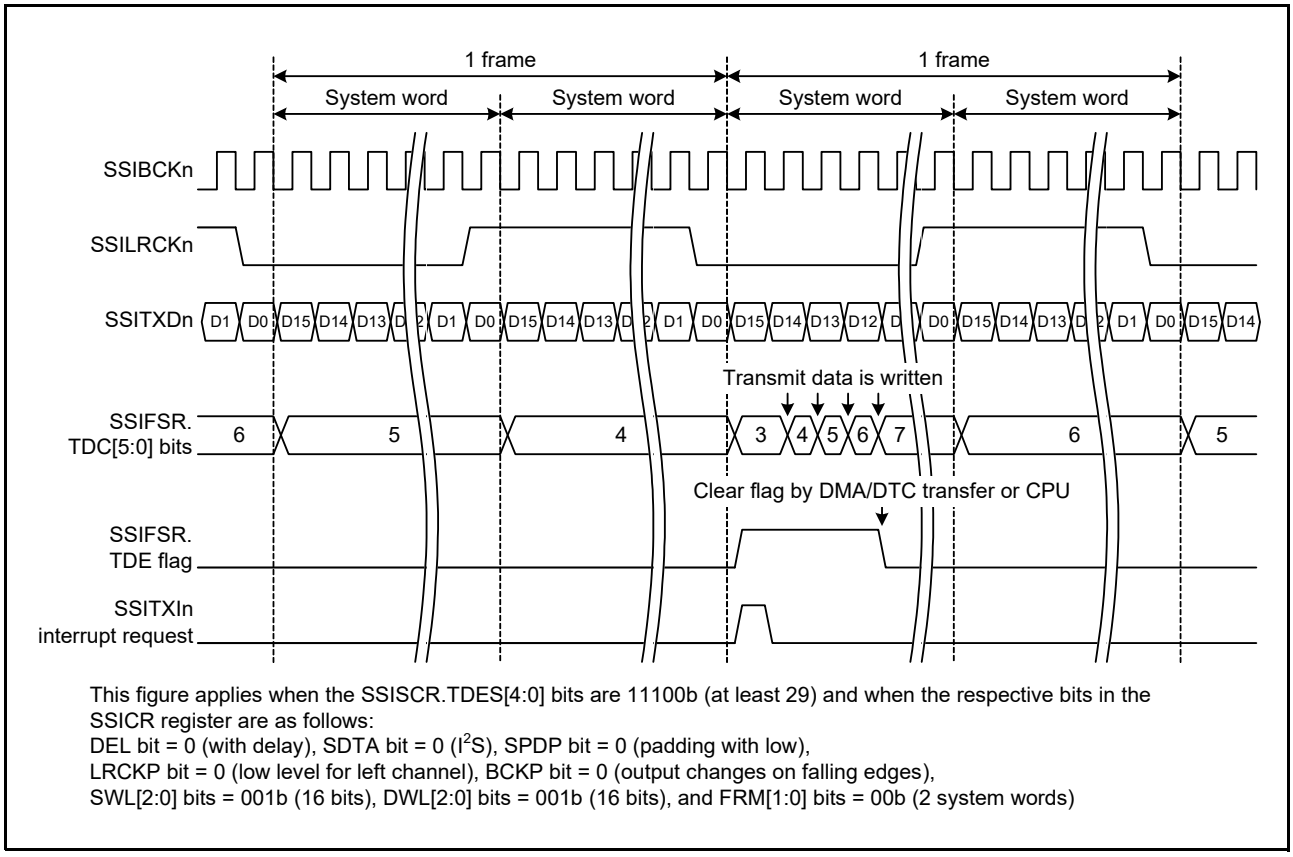


Figure 51.26 SSITXIn Interrupt Timing Chart (n = 0, 1)

51.4.3 SSIRXIn Interrupt

The SSIRXIn interrupt (channel n receive data full interrupt) is an edge detection interrupt. This interrupt request is generated when either of the following conditions is satisfied.

- The SSIFSR.RDF flag has changed from 0 to 1 while the SSIFCR.RIE bit is 1.
- The SSIFCR.RIE bit has been changed from 0 to 1 while the SSIFSR.RDF flag is 1.

Figure 51.27 shows a timing chart when an interrupt request is generated when the RDF flag changes.

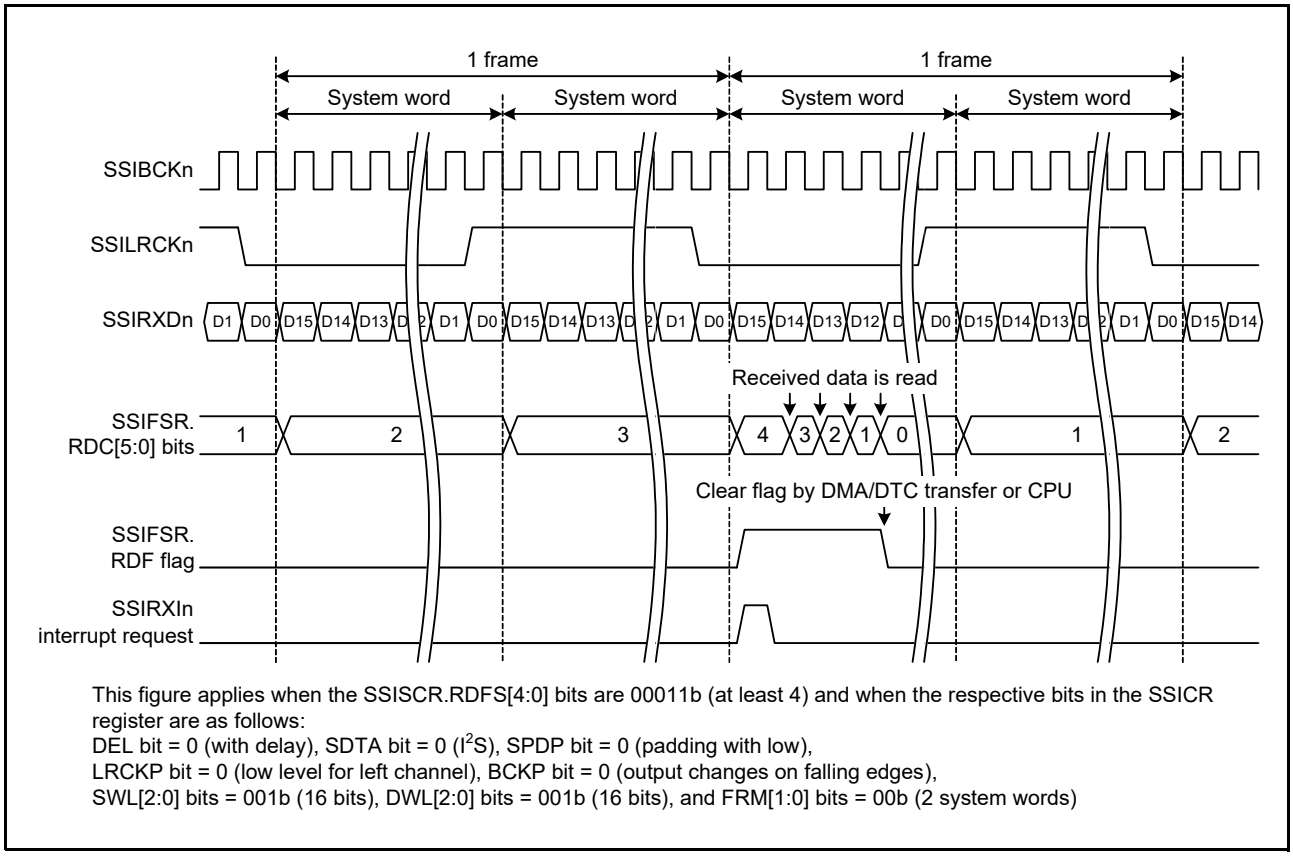


Figure 51.27 SSIRXIn Interrupt Timing Chart (n = 0, 1)

51.5 Notes

51.5.1 Setting the Module Stop Function

Operation of the SSIE can be enabled or disabled by setting a bit in the module stop control register C (MSTPCRC). The SSIE is initially disabled after a reset. Registers in the SSIE only become accessible after it has been released from the module stop state. For details, refer to section 11, Low Power Consumption.

51.5.2 Notes on AUDIO_CLK

In this MCU, the ProtectZone function allows signal input and output only when the security attributes of the peripheral function and the I/O pins match. The SSIE can change the security attribute for each channel. However, if the security attributes of SSIE0 and SSIE1 are different, using AUDIO_CLK on both channels results in the following specification mismatch.

- When the security attribute of the I/O pin is secure, if the security attribute of one channel is secure, AUDIO_CLK is input even if the other channel is non-secure.
- When the security attribute of the I/O pin is non-secure, even if the security attribute of one channel is non-secure, AUDIO_CLK is not input if the other channel is secure.

When setting different security attributes for SSIE0 and SSIE1, use them with the above understanding.
If there is a channel that is not used or does not use AUDIO_CLK, the above problem can be avoided by setting the SSICR.CKS bit of that channel to 1.

51.5.3 Note on LRCK in Slave Mode

The SSIE uses the SSILRCK signal as a start trigger in slave mode. Neither valid edges after the start trigger nor signal levels are used, so if a bit slippage occurs by a noise, the bit slippage cannot be repaired even at a frame boundary.

51.5.4 Transition to Software Standby Mode

Before transitioning to software standby mode, stop communication by following the procedure described in section 51.3.8, Stop Data Transfer. Additionally, if the SSIE is set to master mode, stop the clock output by setting the SSIOFR.BCKASTP bit to 1 and the SSIOFR.LRCONT bit to 0. After returning from software standby mode, set the BCKASTP and LRCONT bits before resuming communication.

52. SD/MMC Host Interface (SDHIc)

This MCU incorporates an SD/MMC host interface (SDHI) which is compliant with the SD Specifications and the *e*•MMC standard (JESD84-B451). When developing host devices that are compliant with the SD Specifications, the user must enter into the SD Host/Ancillary Product License Agreement (SD HALA).

52.1 Overview

Table 52.1 lists the SDHI specifications.

Table 52.1 SDHI Specifications

Item	Description
SD bus interface	• Compatible with SD memory card and SDIO card*1 • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats • Selectable from default speed mode, high-speed mode, SDR12, or SDR25
MMC bus interface*2	• Supports *e*•MMC device • Transfer bus width selectable from 1, 4, or 8 bits • Selectable from backward compatible mode or high-speed SDR mode
SDHI clock	The SDHI clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512
Error check functions	• CRC7 (command/response) • CRC16 (transfer data)
Interrupt sources	• Card access interrupt (CACI) • SDIO access interrupt (SDACI) • Card detection interrupt (CDETI) • SD buffer access interrupt (SBFAI)
DMA transfer sources	• DMAC and DTC triggerable by the SBFAI interrupt • SD buffer is read and write accessible using the DMAC and DTC
Other functions	• Card detection • Write protection
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attributes can be set.*3

Note 1. SPI bus interface, embedded SDIO shared bus, 8-bit SD bus, and SDIO suspend/resume functions are not supported.
Note 2. Stream read/write, high priority interrupt (HPI), boot operation, alternative boot operation, and open-ended multi block read/write are not supported.
Note 3. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Figure 52.1 shows a block diagram of the SDHI.

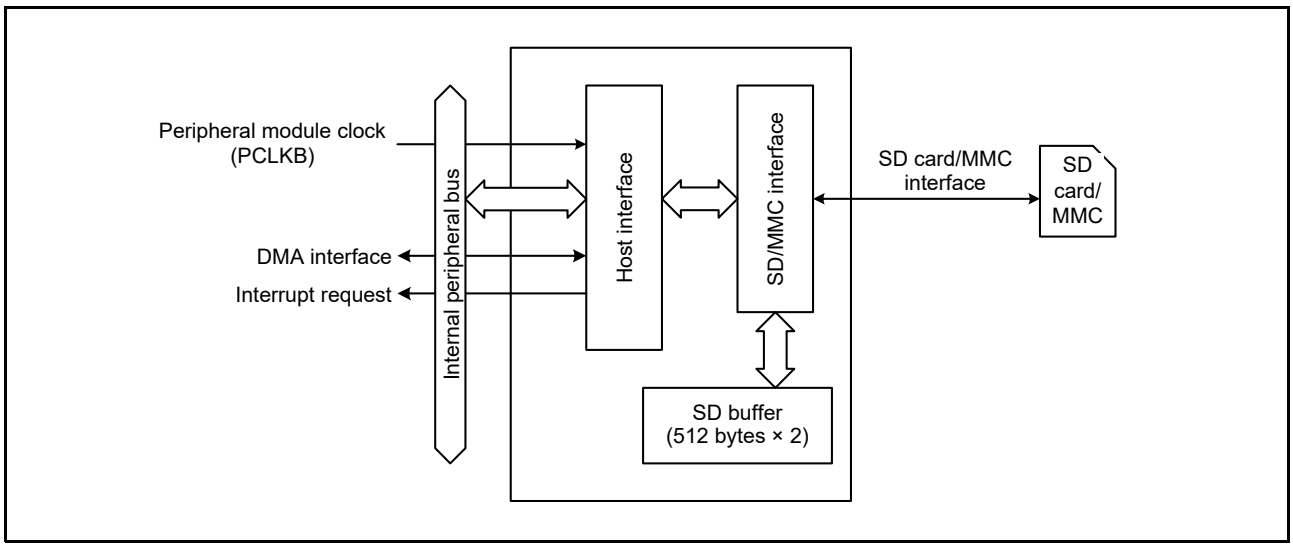


Figure 52.1 Block Diagram of the SDHI

Table 52.2 lists the pin configuration of the SDHI.

Table 52.2 Pin Configuration of the SDHI

Pin Name	I/O	Description
SD0_CLK	Output	SDHI clock
SD0_CMD	I/O	Command output, response input
SD0_DAT0	I/O	Data 0 (DAT0)
SD0_DAT1	I/O	Data 1 (DAT1), SDIO access interrupt
SD0_DAT2	I/O	Data 2 (DAT2), read wait
SD0_DAT3	I/O	Data 3 (DAT3), SD card detection
SD0_DAT4 to SD0_DAT7	I/O	Data 4 (DAT4) to Data 7 (DAT7) for MMC
SD0_CD	Input	SD card/MMC detection
SD0_WP	Input	SD card/MMC write protection

52.2 Register Descriptions

52.2.1 Command Register (SDCMD)

Address(es): SDHI.SDCMD 8785 2000h, SDHI_NS.SDCMD 0785 2000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The examples below include the bit values for the ACMD[1:0] bits. b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD Only set the values listed above.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select*1	b10 b8 0 0 0: Normal mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid. 0 1 1: Expansion mode and no response 1 0 0: Expansion mode and R1, R5, R6, or R7 response 1 0 1: Expansion mode and R1b response 1 1 0: Expansion mode and R2 response 1 1 1: Expansion mode and R3 or R4 response Only set the values listed above.	R/W
b11	CMDTP	Data Transfer Select*2	0: Command does not include data transfer (bc, bcr, or ac) 1: Command includes data transfer (adtc)	R/W
b12	CMDRW	Data Transfer Direction Select*3	0: Write data to the SD card/MMC 1: Read data from the SD card/MMC	R/W
b13	TRSTP	Block Transfer Select*3	0: Single block transfer 1: Multiple blocks transfer	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select*4	b15 b14 0 0: CMD12 is automatically issued during multi-block transfer 0 1: CMD12 is not automatically issued during multi-block transfer Only set the values listed above.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Some commands cannot be used in normal mode. Refer to Table 52.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SDCMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. Refer to Table 52.3 and Table 52.4 for setting examples. Do not write to the SDCMD register when the

SDSTS2.CBSY flag is 1.

Table 52.3 and Table 52.4 list examples of SDCMD register settings.

Table 52.3 Examples of SDCMD Register Settings (SD Card)

Type	Command Symbol	SDCMD Register Setting	Remarks
CMD	CMD0	0000 0000h	
	CMD2	0000 0002h	
	CMD3	0000 0003h	
	CMD4	0000 0004h	
	CMD5	0000 0705h or 0000 0005h	
	CMD6	0000 1C06h or 0000 0006h	
	CMD7	0000 0007h	When the card is deselected, the SD card does not return a response, so the SDSTS2.RSPTO flag becomes 1.
	CMD8	0000 0408h or 0000 0008h	
	CMD9	0000 0009h	
	CMD10	0000 000Ah	
	CMD11	0000 040Bh or 0000 000Bh	
	CMD12	0000 000Ch	
	CMD13	0000 000Dh	
	CMD15	0000 000Fh	
	CMD16	0000 0010h	
	CMD17	0000 0011h	
	CMD18	0000 0012h	Auto CMD12 enabled
	CMD20	0000 0514h or 0000 0014h	
	CMD24	0000 0018h	
	CMD25	0000 0019h	Auto CMD12 enabled
	CMD27	0000 001Bh	
	CMD28	0000 001Ch	
	CMD29	0000 001Dh	
	CMD30	0000 001Eh	
	CMD32	0000 0020h	
	CMD33	0000 0021h	
	CMD38	0000 0026h	
	CMD42	0000 002Ah	
	CMD52	0000 0434h or 0000 0034h	
	CMD53	0000 1C35h	Single block read
		0000 0C35h	Single block write
		0000 7C35h	Multi-block read
		0000 6C35h	Multi-block write
		0000 0035h	The setting to the left can be used regardless of the transfer being single block or multi-block. However, the MSB in the SDARG register (RW flag) must be set to 0 when reading and 1 when writing.
	CMD55	0000 0037h	
	CMD56	0000 0038h	
ACMD	ACMD6	0000 0046h	
	ACMD13	0000 004Dh	
	ACMD22	0000 0056h	
	ACMD23	0000 0057h	
	ACMD41	0000 0069h	
	ACMD42	0000 006Ah	
	ACMD51	0000 0073h	

Table 52.4 Examples of SDCMD Register Settings (MMC)

Type	Command Symbol	SDCMD Register Setting	Remarks
CMD	CMD0	0000 0000h	
	CMD1	0000 0701h	
	CMD2	0000 0002h	
	CMD3	0000 0003h	
	CMD4	0000 0004h	
	CMD5	0000 0505h	
	CMD6	0000 0506h or 0000 0406h	
	CMD7	0000 0007h	When the card is deselected, the MMC does not return a response, so the SDSTS2.RSPTO flag becomes 1.
	CMD8	0000 1C08h	
	CMD9	0000 0009h	
	CMD10	0000 000Ah	
	CMD12	0000 000Ch	
	CMD13	0000 000Dh	
	CMD14	0000 1C0Eh	Set the SDIFMD.NOCHKCR bit to 1 (CRC check is disabled).
	CMD15	0000 000Fh	
	CMD16	0000 0010h	
	CMD17	0000 0011h	
	CMD18	0000 7C12h	Pre-defined block count
	CMD19	0000 0C13h	Set the SDIFMD.NOCHKCR bit to 1 (CRC check is disabled).
	CMD21	0000 1C15h	
	CMD23	0000 0017h	
	CMD24	0000 0018h	
	CMD25	0000 6C19h	Pre-defined block count
	CMD26	0000 0C1Ah	
	CMD27	0000 001Bh	
	CMD28	0000 001Ch	
	CMD29	0000 001Dh	
	CMD30	0000 001Eh	
	CMD31	0000 1C1Fh	
	CMD35	0000 0423h	
	CMD36	0000 0424h	
	CMD38	0000 0026h	
	CMD39	0000 0427h	
	CMD40	0000 0428h	
	CMD42	0000 002Ah	
	CMD49	0000 0C31h	
	CMD53	0000 7C35h	
	CMD54	0000 6C36h	
	CMD55	0000 0037h	
	CMD56	0000 0038h	

52.2.2 Argument Register (SDARG)

Address(es): SDHI.SDARG 8785 2008h, SDHI_NS.SDARG 0785 2008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SDARG register is used for setting the argument field value. Set the SDARG register before setting the SDCMD register. The argument field value of the automatically issued CMD12 is 0000 0000h regardless of the SDARG register value.

52.2.3 Data Stop Register (SDSTOP)

Address(es): SDHI.SDSTOP 8785 2010h, SDHI_NS.SDSTOP 0785 2010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDBLKCNTEN	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1.	R/W
b7 to b1	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDBLKCNTEN	Block Count Register Value Select*1	0: SDBLKCNT register value is invalid 1: SDBLKCNT register value is valid	R/W
b31 to b9	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSTOP register stops data transfer. During a multi-block transfer sequence, the SDBLKCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SDSTOP register.

STP Bit (Transfer Stop)

When setting the STP bit to 1, set it after the SDSTS1.RSPEND flag becomes 1; when setting the STP bit to 0, set it after the SDSTS1.ACEND flag becomes 1. After a command sequence is complete, the SDHI does not issue CMD12 and the SDSTS1.ACEND flag does not become 1 even if the STP bit is set to 1. When the SDHI is in the busy state after receiving the R1b response, the SDHI does not issue CMD12 even if the STP bit is 1, and after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1.

- Performing a multi-block transfer
  When the STP bit is set to 1, the SDHI issues CMD12, and the command sequence is stopped. The SD buffer can be accessed even after the STP bit is set to 1, but a buffer access error occurs and the SDSTS2.ILW flag or SDSTS2.ILR flag becomes 1. If the command sequence stops due to a communication error or a timeout, the SDHI does not issue CMD12.
- Performing a single block transfer
  When the STP bit is set to 1 during a single block write access, if there is no data in the SD buffer, the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer, after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1. When the STP bit is set to 1 during a single block read access, the SDSTS1.ACEND flag becomes 1. Also, CMD12 is not issued even if the STP bit is set to 1 during a single block read access or single block write access.

SDBLKCNTEN Bit (Block Count Register Value Select)

If the SDBLKCNTEN bit is 1 during a multi-block transfer sequence, the SDHI automatically issues CMD12. When the SDCMD.RSPTP[2:0] bits are set to 000b and CMD18 or CMD25 is issued, or if the SDCMD.RSPTP[2:0] bits are set to 011b, 100b, 101b, 110b, or 111b and the SDCMD.TRSTP bit is 1 (multiple blocks transferred), if the SDCMD.CMD12AT[1:0] bits are 00b (CMD12 is automatically issued during multi-block transfer), and the number of transfer blocks reaches the value set in the SDBLKCNT register, the SDHI automatically issues CMD12.
If the command sequence is stopped by a communication error or a timeout, CMD12 is not automatically issued.

52.2.4 Block Count Register (SDBLKCNT)

Address(es): SDHI.SDBLKCNT 8785 2014h, SDHI_NS.SDBLKCNT 0785 2014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When performing a multi-block transfer, SDBLKCNT is a readable/writable register used to set the number of blocks to be transferred. For example, when the register value is 0000 0001h, 1 block is transferred; when the register value is 0000 FFFFh, 65,535 blocks are transferred; and when the register value is FFFF FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000 0000h. Do not rewrite the SDBLKCNT register when the SDSTS2.CBSY flag is 1.

52.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76)

- SDRSP10, SDRSP32, SDRSP54

Address(es): SDHI.SDRSP10 8785 2018h, SDHI.SDRSP32 8785 2020h, SDHI.SDRSP54 8785 2028h, SDHI_NS.SDRSP10 0785 2018h, SDHI_NS.SDRSP32 0785 2020h, SDHI_NS.SDRSP54 0785 2028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- SDRSP76

Address(es): SDHI.SDRSP76 8785 2030h, SDHI_NS.SDRSP76 0785 2030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Registers SDRSP10, SDRSP32, SDRSP54, and SDRSP76 are read-only registers that store the response from the SD card/MMC. Depending on the type of response from the SD card/MMC, the SDHI divides and stores the response among the four registers.
Table 52.5 lists the correspondence between the response type and its storage destination.

Table 52.5 Correspondence Between the Response Type and Its Storage Destination

Response Type	SDRSP76 Register	SDRSP54 Register	SDRSP32 Register	SDRSP10 Register
R1	—	[39:8] *1	—	[39:8]
R1b	—	[39:8] *1	—	[39:8]
R2	[127:104]	[103:72]	[71:40]	[39:8]
R3	—	—	—	[39:8]
R4	—	—	—	[39:8]
R5	—	—	—	[39:8]
R6	—	—	—	[39:8]
R7	—	—	—	[39:8]

Note 1. The responses for CMD18 and CMD25 are stored in registers SDRSP10 and SDRSP54. Therefore, even if the SDRSP10 register is overwritten with the response for the automatically transmitted CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SDRSP54 register.

52.2.6 SD Status Register 1 (SDSTS1)

Address(es): SDHI.SDSTS1 8785 2038h, SDHI_NS.SDSTS1 0785 2038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	0	0	0	0	0	x	0	0	x	0	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end is not detected 1: Response end is detected	R/(W) *1
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACEND	Access End Detection Flag	0: Access end is not detected 1: Access end is detected	R/(W) *1
b3	SDCDRM	SD0_CD Removal Flag	0: SD card/MMC removal not detected by the SD0_CD pin 1: SD card/MMC removal detected by the SD0_CD pin	R/(W) *1
b4	SDCDIN	SD0_CD Insertion Flag	0: SD card/MMC insertion not detected by the SD0_CD pin 1: SD card/MMC insertion detected by the SD0_CD pin	R/(W) *1
b5	SDCDMON	SD0_CD Pin Monitor Flag	0: SD0_CD pin level is high *2 1: SD0_CD pin level is low *2	R
b6	—	Reserved	This bit is 0 when read and cannot be modified.	R
b7	SDWPMON	SD0_WP Pin Monitor Flag	0: SD0_WP pin level is high 1: SD0_WP pin level is low	R
b8	SDD3RM	SD0_DAT3 Removal Flag	0: SD card/MMC removal not detected by the SD0_DAT3 pin 1: SD card/MMC removal detected by the SD0_DAT3 pin	R/(W) *1
b9	SDD3IN	SD0_DAT3 Insertion Flag	0: SD card/MMC insertion not detected by the SD0_DAT3 pin 1: SD card/MMC insertion detected by the SD0_DAT3 pin	R/(W) *1
b10	SDD3MON	SD0_DAT3 Pin Monitor Flag	0: SD0_DAT3 pin level is low 1: SD0_DAT3 pin level is high	R
b31 to b11	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 2. The flag changes when the pin level continues for the period set in the SDOPT.CTOP[3:0] bits or longer.

The SDSTS1 register indicates the detection of a response end or access end for a command sequence. The SDSTS1 register also indicates the detection SD card/MMC insertion/removal, and indicates the write protection status.
During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.
If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.
After a reset is released, the SDD3MON flag, SDD3IN flag, and SDD3RM flag values are changed according to the status of the SD0_DAT3 pin, and their values are changed when data is being transferred in wide bus mode.
Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

RSPEND Flag (Response End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A response is received.
- A command that does not have a response is issued.
- After the R1b response is received, the SDHI is released from the busy state.
- During a multi-block transmission, after the SDIOMD.C52PUB bit is set to 1, the CMD52 response is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: When a command is issued that is absent of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND Flag (Access End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block read sequence, the SD buffer read access is completed.
- During a multi-block read sequence, the last block is read from the SD buffer.
- During a multi-block read sequence, if CMD12 is automatically issued, data is read from the SD buffer, and the response for CMD12 is received.
- During a single block write sequence, after a CRC status token is received, the SDHI is released from the busy state.
- During a multi-block write sequence, after a CRC status token is received for the last block, the SDHI is released from the busy state.
- During a multi-block write sequence, when CMD12 is automatically issued, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response of the automatically issued CMD12 is received.
- During a multi-block write sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- During a multi-block write sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM Flag (SD0_CD Removal Flag)

— This flag becomes 1 under the following condition:

- The SD0_CD pin changes from low to high, and the high period is the period set in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDCDIN Flag (SD0_CD Insertion Flag)

— This flag becomes 1 under the following condition:

- The SD0_CD pin changes from high to low, and the low period is the period specified in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3RM Flag (SD0_DAT3 Removal Flag)

— This flag becomes 1 under the following condition:

- The SD0_DAT3 pin changes from high to low, and the low period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3IN Flag (SD0_DAT3 Insertion Flag)

— This flag becomes 1 under the following condition:

- The SD0_DAT3 pin changes from low to high, and the high period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

52.2.7 SD Status Register 2 (SDSTS2)

Address(es): SDHI.SDSTS2 8785 203Ch, SDHI_NS.SDSTS2 0785 203Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILA	CBSY	SDCLKCREN	—	—	—	BWE	BRE	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	0	0	1	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected	R/(W) *1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected	R/(W) *1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected	R/(W) *1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected	R/(W) *1
b4	ILW	SDBUFR Illegal Write Access Detection Flag	0: Illegal write access to the SDBUFR register not detected 1: Illegal write access to the SDBUFR register detected	R/(W) *1
b5	ILR	SDBUFR Illegal Read Access Detection Flag	0: Illegal read access to the SDBUFR register not detected 1: Illegal read access to the SDBUFR register detected	R/(W) *1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected	R/(W) *1
b7	SDD0MON	SD0_DAT0 Pin Status Flag	0: SD0_DAT0 pin is low 1: SD0_DAT0 pin is high	R
b8	BRE	SDBUFR Read Enable Flag	0: Read access to the SDBUFR register disabled 1: Read access to the SDBUFR register enabled	R/(W) *1
b9	BWE	SDBUFR Write Enable Flag	0: Write access to the SDBUFR register disabled 1: Write access to the SDBUFR register enabled	R/(W) *1
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 0 when read. Set it to 1 when writing.	R/W
b12	—	Reserved	This bit is 0 when read and cannot be modified.	R
b13	SDCLKCREN	SDCLKCR Write Enable Flag	0: SD/MMC bus (CMD and DAT lines) is busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is disabled. 1: SD/MMC bus (CMD and DAT lines) is not busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence completed 1: Command sequence in progress (busy)	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SDSTS2 register indicates the status of the SD buffer and the status of the SD card/MMC. Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

CMDE Flag (Command Error Detection Flag)

The command sequence is stopped when a command error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 52.3.6.9 or section 52.3.6.10 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The command index field value for the command transmitted differs from the command index field value for the response received.
- The command index field value for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) differs from the command index field value for the response received.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CRCE Flag (CRC Error Detection Flag)

The command sequence is stopped when a CRC error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 52.3.6.9 or section 52.3.6.10 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The received CRC status token is in error (the value of the CRC status is a value other than 010b).
- The read data contains a CRC error.
- The response contains a CRC error.
- The response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains a CRC error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ENDE Flag (End Bit Error Detection Flag)

The command sequence is stopped when an end bit error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 52.3.6.9 or section 52.3.6.10 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The response length is in error (the end bit could not be detected).
- The read data length is in error (the end bit of the enabled bit could not be detected).
- The CRC status token length is in error (the end bit could not be detected).
- The response length for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains an error (the end bit could not be detected).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

DTO Flag (Data Timeout Detection Flag)

This flag indicates that the data expected to be received during the period specified (set in the SDOPT.TOP[3:0] bits) was not received. However, response timeouts are excluded. The command sequence stops when a data timeout occurs.

— This flag becomes 1 under any of the following conditions:

- After the R1b response is received, the SDHI is busy for the period specified or longer.
- After the CRC status token is received, the SDHI is busy for the period specified or longer.
- After data is written, the CRC status token is not received even after the period specified elapsed.
- After a read command is issued, the read data is not received even after the period specified elapsed.
- After CMD12 is issued during a command sequence, the SDHI is busy for the period specified or longer.
- After the read data is received, the next read data is not received even after the period specified elapsed.
- After the SDHI exits the read wait state, the next read data is not received even after the period specified elapsed.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILW Flag (SDBUFR Illegal Write Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDBUFR register while the SDHI is not in the data read or data write command state.
- A value is written to the SDBUFR register while the SD buffer is full.
- A value is written to the SDBUFR register while the CRC status token or CRC status token length is in error.
- After the CRC status token is received, a value is written to the SDBUFR register if the SDHI is busy for the period set in bits SDOPT.TOP[3:0] or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILR Flag (SDBUFR Illegal Read Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- The SDBUFR register is read while the SD buffer is empty.
- The value read from the SDBUFR register includes a CRC error or end bit error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

RSPTO Flag (Response Timeout Detection Flag)

The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 52.3.6.9 or section 52.3.6.10 and complete the command sequence.

— This flag becomes 1 under the following condition:

- A response is not received even after 640 SDHI clock cycles or more have elapsed (including the response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer)).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD0MON Flag (SD0_DAT0 Pin Status Flag)

This flag indicates the status of the SD0_DAT0 pin. After an erase command is issued, if the DTO flag is 1 and the RSPTO flag is 0, polling can be used to monitor the SDD0MON flag change from 0 to 1, and check that the erase command sequence is complete. If a communication error or timeout occurs during the write sequence, the SD0_DAT0 pin may remain low. When the SDHI clock is stopped, the value before the SDHI clock was stopped is retained.

BRE Flag (SDBUFR Read Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in the SD buffer.
- During a multi-block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in one of the two SD buffers.

— This flag becomes 0 under any of the following conditions:

- The bit is set to 0.
- DMA transfer is used to read 1 block of data from the SD buffer.

If the CPU is used to read data from the SDBUFR register, set the BRE flag to 0 before reading the data size *1 set in the SDSIZE.LEN[9:0] bits. Even if the block of data read contains a CRC error or end bit error, the data is stored in the SD buffer and the BRE flag becomes 1.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the remaining byte is ignored. Refer to section 52.5.2, SDBUFR Register Illegal Write Error for details.

BWE Flag (SDBUFR Write Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the SD buffer is empty.
- During a multi-block transfer, bank 1 or bank 2 of the SD buffer is empty.

— This flag becomes 0 under any of the following conditions:

- The flag is set to 0.
- DMA transfer is used to write 1 block of data to the SD buffer.

If the CPU is used to write data to the SDBUFR register, set the BWE flag to 0 before writing the data size *1 set in the SDSIZE.LEN[9:0] bits.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the remaining byte is ignored. Refer to section 52.5.2, SDBUFR Register Illegal Write Error for details.

SDCLKCREN Flag (SDCLKCR Write Enable Flag)

When a value is written to the SDCMD register, the SDHI starts the command sequence, the SDSTS2.CBSY flag becomes 1, and the SDSTS2.SDCLKCREN flag becomes 0. When the command sequence is complete, after the SDSTS2.CBSY flag becomes 0, eight cycles of the SDHI clock elapse and then the SDSTS2.SDCLKCREN flag becomes 1.

ILA Flag (Illegal Access Error Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDCMD register when the SDSTS2.CBSY flag is 1.
- The SDCMD.CMDTP bit is set to 1 (command accompanying data transfer), the SDCMD.ACMD[1:0] bits are set to 00b, and the SDCMD.CMDIDX[5:0] bits are set to 001100b (CMD12).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

52.2.8 SD Interrupt Mask Register 1 (SDIMSK1)

Address(es): SDHI.SDIMSK1 8785 2040h, SDHI_NS.SDIMSK1 0785 2040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SDD3INM	SDD3RMM	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Response end interrupt request is not masked 1: Response end interrupt request is masked	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACENDM	Access End Interrupt Request Mask	0: Access end interrupt request is not masked 1: Access end interrupt request is masked	R/W
b3	SDCDRMM	SD0_CD Removal Interrupt Request Mask	0: SD card/MMC removal interrupt request by the SD0_CD pin not masked 1: SD card/MMC removal interrupt request by the SD0_CD pin masked	R/W
b4	SDCDINM	SD0_CD Insertion Interrupt Request Mask	0: SD card/MMC insertion interrupt request by the SD0_CD pin not masked 1: SD card/MMC insertion interrupt request by the SD0_CD pin masked	R/W
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDD3RMM	SD0_DAT3 Removal Interrupt Request Mask	0: SD card/MMC removal interrupt request by the SD0_DAT3 pin not masked 1: SD card/MMC removal interrupt request by the SD0_DAT3 pin masked	R/W
b9	SDD3INM	SD0_DAT3 Insertion Interrupt Request Mask	0: SD card/MMC insertion interrupt request by the SD0_DAT3 pin not masked 1: SD card/MMC insertion interrupt request by the SD0_DAT3 pin masked	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIMSK1 register enables and disables the interrupt requests from the status flags in the SDSTS1 register. Refer to Table 52.9, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

52.2.9 SD Interrupt Mask Register 2 (SDIMSK2)

Address(es): SDHI.SDIMSK2 8785 2044h, SDHI_NS.SDIMSK2 0785 2044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILAM	—	—	—	—	—	BWEM	BREM	—	RSPTOM	ILRM	ILWM	DTTOM	ENDEM	CRCEM	CMDEM
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Command error interrupt request not masked 1: Command error interrupt request masked	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: CRC error interrupt request not masked 1: CRC error interrupt request masked	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: End bit detection error interrupt request not masked 1: End bit detection error interrupt request masked	R/W
b3	DTTOM	Data Timeout Interrupt Request Mask	0: Data timeout interrupt request not masked 1: Data timeout interrupt request masked	R/W
b4	ILWM	SDBUFR Register Illegal Write Interrupt Request Mask	0: Illegal write detection interrupt request for the SDBUFR register not masked 1: Illegal write detection interrupt request for the SDBUFR register masked	R/W
b5	ILRM	SDBUFR Register Illegal Read Interrupt Request Mask	0: Illegal read detection interrupt request for the SDBUFR register not masked 1: Illegal read detection interrupt request for the SDBUFR register masked	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Response timeout interrupt request not masked 1: Response timeout interrupt request masked	R/W
b7	—	Reserved	This bit is 0 when read and cannot be modified.	R
b8	BREM	BRE Interrupt Request Mask*1	0: Read enable interrupt request for the SDBUFR register not masked 1: Read enable interrupt request for the SDBUFR register masked	R/W
b9	BWEM	BWE Interrupt Request Mask*1	0: Write enable interrupt request for the SDBUFR register not masked 1: Write enable interrupt request for the SDBUFR register masked	R/W
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R/W
b14 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Illegal access error interrupt request not masked 1: Illegal access error interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDIMSK2 register enables and disables the interrupt requests from the status flags in the SDSTS2 register. Refer to Table 52.9 for details on the relationship between the status flags and the requested interrupt source.

52.2.10 SDHI Clock Control Register (SDCLKCR)

Address(es): SDHI.SDCLKCR 8785 2048h, SDHI_NS.SDCLKCR 0785 2048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CLKCTRLEN	CLKEN	CLKSEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select	b7 b0 0 0 0 0 0 0 0 0: PCLKB divided by 2 0 0 0 0 0 0 0 1: PCLKB divided by 4 0 0 0 0 0 0 1 0: PCLKB divided by 8 0 0 0 0 0 1 0 0: PCLKB divided by 16 0 0 0 0 1 0 0 0: PCLKB divided by 32 0 0 0 1 0 0 0 0: PCLKB divided by 64 0 0 1 0 0 0 0 0: PCLKB divided by 128 0 1 0 0 0 0 0 0: PCLKB divided by 256 1 0 0 0 0 0 0 0: PCLKB divided by 512 1 1 1 1 1 1 1 1: PCLKB *2 Only set the values listed above.	R/W*1
b8	CLKEN	SDHI Clock Output Control	0: SDHI clock output is disabled (SD0_CLK signal fixed low) 1: SDHI clock output enabled	R/W*1
b9	CLKCTRLEN	SDHI Clock Output Automatic Control Select	0: Automatic control of SDHI clock output disabled 1: Automatic control of SDHI clock output enabled	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SDSTS2.SDCLKCREN flag is 0.
Note 2. When setting the CLKSEL[7:0] bits to 11111111b or when changing the CLKSEL[7:0] bit values from 11111111b to another value, perform the following steps:
(1) Set the CLKEN bit to 0. Do not change the other bit values.
(2) Change the CLKSEL[7:0] bit values. Do not change the other bit values.
(3) Set the CLKEN bit to 1. Do not change the other bit values.

The SDCLKCR register controls the SDHI clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SDCMD register to start a command sequence. Do not write access the SDCLKCR register when the SDSTS2.SDCLKCREN flag is 0.

CLKCTRLEN Bit (SDHI Clock Output Automatic Control Select)

The SDHI clock output automatic control is a function for starting and stopping SDHI clock output only during a command sequence. When this function is enabled, the SDHI starts outputting the SDHI clock after a value is set to the SDCMD register. After the command sequence is complete and eight cycles of the SDHI clock elapse, the SDHI stops outputting the SDHI clock. When the SDCLKCR.CLKEN bit is 0, output from the SD0_CLK pin becomes low regardless of the CLKCTRLEN bit setting.

52.2.11 Transfer Data Size Register (SDSIZE)

Address(es): SDHI.SDSIZE 8785 204Ch, SDHI_NS.SDSIZE 0785 204Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LEN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting*1	Set the transfer data size.	R/W
b11 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b31 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SDSIZE register is used to set the transfer data size.

LEN[9:0] Bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

52.2.12 Card Access Option Register (SDOPT)

Address(es): SDHI.SDOPT 8785 2050h, SDHI_NS.SDOPT 0785 2050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WIDTH	—	WIDTH8	—	—	—	—	TOUTMASK	TOP[3:0]				CTOP[3:0]			
Value after reset:	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTOP[3:0]	Card Detection Time Counter*1	b3 b0 0 0 0 0: PCLKB × 2^{10} 0 0 0 1: PCLKB × 2^{11} 0 0 1 0: PCLKB × 2^{12} 0 0 1 1: PCLKB × 2^{13} 0 1 0 0: PCLKB × 2^{14} 0 1 0 1: PCLKB × 2^{15} 0 1 1 0: PCLKB × 2^{16} 0 1 1 1: PCLKB × 2^{17} 1 0 0 0: PCLKB × 2^{18} 1 0 0 1: PCLKB × 2^{19} 1 0 1 0: PCLKB × 2^{20} 1 0 1 1: PCLKB × 2^{21} 1 1 0 0: PCLKB × 2^{22} 1 1 0 1: PCLKB × 2^{23} 1 1 1 0: PCLKB × 2^{24} 1 1 1 1: Do not set this value.	R/W
b7 to b4	TOP[3:0]	Timeout Counter*1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20} 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Do not set this value.	R/W
b8	TOUTMASK	Timeout Mask	0: Timeout detection is enabled. 1: Timeout detection is disabled.	R/W
b12 to b9	—	Reserved	These bits are 0 when read and cannot be modified.	R
b13	WIDTH8	MMC Bus Width Select*1	0: Wide bus mode (4 bits) 1: Wide bus mode (8 bits)	R/W
b14	—	Reserved	This bit is 1 when read and cannot be modified.	R
b15	WIDTH	SD Bus Width Select*1	0: Selected by the WIDTH8 bit 1: Default bus mode (1 bit)	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SD/MMC bus width and timeout counter are set in the SDOPT register.

TOUTMASK Bit (Timeout Mask)

This bit is used to disable the timeout detect function.
When this bit is set to 1, the RSPTO and DTO flags in the SDSTS2 register and each flag in the SDERSTS2 register are not set to 1.

WIDTH8 Bit (MMC Bus Width Select)

This bit is used to select the bus width of the MMC bus in combination with the WIDTH bit.

WIDTH Bit	WIDTH8 Bit	Bus Width
0	0	4 bits
0	1	8 bits
1	0	1 bit
1	1	1 bit

52.2.13 SD Error Status Register 1 (SDERSTS1)

Address(es): SDHI.SDERSTS1 8785 2058h, SDHI_NS.SDERSTS1 0785 2058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: Command index field value for a command *1 response is error free 1: Command index field value for a command *1 response is in error	R
b1	CMDE1	Command Error Flag 1	0: Command index field value for a command *2 response is error free 1: Command index field value for a command *2 response is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag)	R
b2	RSPLENE0	Response Length Error Flag 0	0: Command *1 response length is error free 1: Command *1 response length is in error	R
b3	RSPLENE1	Response Length Error Flag 1	0: Command *2 response length is error free 1: Command *2 response length is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag)	R
b4	RDLENE	Read Data Length Error Flag	0: Read data length error did not occur 1: Read data length error occurred	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: CRC status token length error did not occur 1: CRC status token length error occurred	R
b7, b6	—	Reserved	These bits are 0 when read.	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command *1 response 1: CRC error detected in command *1 response	R
b9	RSPCRCE1	Response CRC Error Flag 1	0: No CRC error detected in command *2 response 1: CRC error detected in command *2 response (the error that occurs for CMD12 with the setting of the SDCMD.CMDIDX[5:0] bits is indicated by the RSPCRCE0 flag)	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token	R
b14 to b12	CRCTK[2:0]	CRC Status Token	Store the CRC status token value (normal value is 010b)	R
b15	—	Reserved	This bit is 0 when read.	R
b31 to b16	—	Reserved	These bits are undefined when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop data transfer.

The SDERSTS1 register indicates the CRC status token, CRC error, end bit error, and command error.

52.2.14 SD Error Status Register 2 (SDERSTS2)

Address(es): SDHI.SDERSTS2 8785 205Ch, SDHI_NS.SDERSTS2 0785 205Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After a command *1 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *1 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed.	R
b1	RSPTO1	Response Timeout Flag 1	0: After a command *2 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *2 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag).	R
b2	BSYTO0	Busy Timeout Flag 0	0: After the R1b response was received, the SDHI was released from the busy state during the specified period *3. 1: After the R1b response was received, the SDHI was in the busy state even after the specified period *3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, the SDHI was released from the busy state during the specified period *3. 1: After CMD12 was automatically issued, the SDHI was in the busy state even after the specified period *3 elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	After a read command is issued, this flag becomes 1 when read data is not received even after the specified period *3 elapses. After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses. After the SDHI exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After data was written to the SD card/MMC, a CRC status token was received during the specified period *3. 1: After CRC data was written to the SD card/MMC, a CRC status token was not received even after the specified period *3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SDHI was released from the busy state during the specified period *3. 1: After a CRC status token was received, the SDHI is in the busy state even after the specified period *3 elapsed.	R
b31 to b7	—	Reserved	These bits are 0 when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. Set the SDOPT.TOP[3:0] bits to select the number of *n* cycles.

The SDERSTS2 register indicates the timeout status.

52.2.15 SD Buffer Register (SDBUFR)

Address(es): SDHI.SDBUFR 8785 2060h, SDHI_NS.SDBUFR 0785 2060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

The SDBUFR register is used when writing data to the SD card/MMC and when reading data from the SD card/MMC. The SDBUFR register is connected to the SDHI's internal SD buffer. Refer to section 52.3.1, Data Block Format of the SD Card/MMC for details on the configuration of the SDBUFR register and the SD buffer.

52.2.16 SDIO Mode Control Register (SDIOMD)

Address(es): SDHI.SDIOMD 8785 2068h, SDHI_NS.SDIOMD 0785 2068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable*1	0: SDIO interrupt accept disabled 1: SDIO interrupt accept enabled	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	RWREQ	Read Wait Request	0: SDHI exits read wait state 1: Request for SDHI to enter read wait state	R/W
b7 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued before the transfer process is complete, and the command sequence is completed.	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIOMD register controls reception of the SDIO interrupt, controls CMD52 issuance during multi-block transfer, and controls the read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ Bit (Read Wait Request)

If the RWREQ bit is set to 1 during a multi-block read sequence triggered by issuing CMD53, when the current block is done being read, the SDHI enters the read wait state. The method for exiting the read wait state is as follows.

- If the RWREQ bit is set to 0 while the SDHI is in the read wait state, the SDHI exits the read wait state.
- If the IOABT bit is set to 1 while the SDHI is in the read wait state, after CMD52 is issued, the RWREQ bit becomes 0 and the SDHI exits the read wait state.
- If bits C52PUB and RWREQ are simultaneously set to 1 during a multi-block read sequence triggered by issuing CMD53 *1, the SDHI does not automatically exit the read wait state, so after receiving the CMD52 response, set the RWREQ bit to 0.

Note 1. Set bits RWREQ and C52PUB to 1 simultaneously.

If the RWREQ bit is set to 1 while the last block is being transferred during a multi-block read sequence triggered by issuing CMD53, the SDHI will not enter the read wait state, the SDSTS1.ACEND flag becomes 1, and the RWREQ bit becomes 0. Set the RWREQ bit to 1 after the SDSTS1.RSPEND flag becomes 1.

IOABT Bit (SDIO Abort)

- If the IOABT bit is set to 1 during a multi-block transfer sequence triggered by issuing CMD53, the SDHI stops the CMD53 command sequence, and CMD52 is issued. If the command sequence is stopped due to a communication error or timeout, the SDHI does not issue CMD52. The SD buffer can be accessed even after the IOABT bit is set to 1, but the SDSTS2.ILR flag or ILW flag becomes 1, and a buffer access error occurs. Write a value to the SDARG register before setting the IOABT bit to 1.
- During a single block write, if there is no data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 during a single block read, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag immediately becomes 1.
- If the SDHI is in the busy state after the R1b response is received and the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 after the command sequence is completed, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag does not become 1.
- Set the IOABT bit to 1 after the SDSTS1.RSPEND flag becomes 1.
- Set the IOABT bit to 0 after the SDSTS1.ACEND flag becomes 1.

C52PUB Bit (SDIO None Abort)

- If the C52PUB bit is set to 1 during a multi-block write sequence triggered by issuing CMD53, CMD52 is automatically issued when the SD buffer is empty and the current block write access is complete. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- If the C52PUB bit and RWREQ bit are set to 1 during a multi-block read sequence triggered by issuing CMD53, the SDHI enters the read wait state after the current block read access is complete, and the SDHI automatically issues CMD52. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is set to 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- During a multi-block read sequence triggered by issuing CMD53, if the C52PUB bit is set to 1, also set the RWREQ bit to 1.
- Write a value to the SDARG register before setting the C52PUB bit to 1.
- Set the C52PUB bit to 1 after the SDSTS1.RSPEND flag becomes 1.

52.2.17 SDIO Status Register (SDIOSTS)

Address(es): SDHI.SDIOSTS 8785 206Ch, SDHI_NS.SDIOSTS 0785 206Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWT	EXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: SDIO interrupt not accepted 1: SDIO interrupt accepted	R/(W) *1
b2, b1	—	Reserved	These bits are undefined when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52	R/(W) *1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag value does not change even when set to 1. If 0 is written to this flag, it becomes 0.

The SDIOSTS register indicates the status of the SDIO card access. When clearing a flag, bits to be cleared should be set to 0; bits not being cleared should be set to 1.

IOIRQ Flag (SDIO Interrupt Status Flag)

— This flag becomes 1 under the following condition:

- The SDIO interrupt from the SDIO card is accepted while the SDIOMD.INTEN bit is 1.

— This flag becomes 0 under the following condition:

- The flag is set to 0. *1

Note 1. Access the SDIO card, negate the SDIO interrupt from the SDIO card, and then set the IOIRQ flag to 0. If the SDIO interrupt from the SDIO card is not negated, the IOIRQ flag might become 1 again.

EXPUB52 Flag (EXPUB52 Status Flag)

— This flag becomes 1 under any of the following conditions:

- When multi-block transfer is triggered by CMD53 being issued, the SDIOMD.C52PUB bit is set to 1 while the last block is being transferred.
- When multi-block write is triggered by CMD53 being issued, the SDIOMD.C52PUB bit remains set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

EXWT Flag (EXWT Status Flag)

— This flag becomes 1 under the following condition:

- During a multi-block read sequence triggered by CMD53 being issued, the SDIOMD.RWREQ bit is set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

52.2.18 SDIO Interrupt Mask Register (SDIOIMSK)

Address(es): SDHI.SDIOIMSK 8785 2070h, SDHI_NS.SDIOIMSK 0785 2070h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWTM	EXPUB52M	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQM
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: IOIRQ interrupt not masked 1: IOIRQ interrupt masked	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: EXPUB52 interrupt request not masked 1: EXPUB52 interrupt request masked	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: EXWT interrupt request not masked 1: EXWT interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIOIMSK register enables and disables the interrupt requests from the status flags in the SDIOSTS register. Refer to Table 52.9, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

52.2.19 DMA Transfer Enable Register (SDDMAEN)

Address(es): SDHI.SDDMAEN 8785 21B0h, SDHI_NS.SDDMAEN 0785 21B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b1	DMAEN	DMA Transfer Enable*1 *2	0: Using DMAC and DTC to access the SDBUFR register is disabled 1: Using DMAC and DTC to access the SDBUFR register is enabled	R/W
b3, b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b4	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b5	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are 0 when read and cannot be modified.	R
b9, b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b11, b10	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b31 to b13	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY bit is 1.
Note 2. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDDMAEN register enables and disables DMA transfer.

DMAEN Bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SDCMD register.

52.2.20 SDHI Software Reset Register (SDRST)

Address(es): SDHI.SDRST 8785 21C0h, SDHI_NS.SDRST 0785 21C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDRST	SDHI Software Reset Control	0: SDHI software reset 1: SDHI software reset released	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R
b31 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R

Table 52.6 lists the bits and flags initialized by the SDHI software reset.

Table 52.6 Bits and Flags Initialized by the SDHI Software Reset

Register	Bit/Flag
SDSTOP	SDBLKCNTEN
SDSTS1	RSPEND, ACEND
SDSTS2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SDCLKCREN, ILA
SDCLKCR	CLKEN
SDOPT	CTOP[3:0], TOP[3:0], TOUTMASK, WIDTH8, WIDTH
SDERSTS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SDERSTS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIOSTS	IOIRQ, EXPUB52, EXWT
SDIFMD	NOCHKCR

52.2.21 SD/MMC Interface Mode Register (SDIFMD)

Address(es): SDHI.SDIFMD 8785 21CCh, SDHI_NS.SDIFMD 0785 21CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NOCHKCR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7 to b1	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	NOCHKCR	CRC Check Mask*1	0: CRC check is enabled 1: CRC check is disabled	R/W
b9	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIFMD register is used to specify the operation mode of the SD/MMC interface.

NOCHKCR Bit (CRC Check Mask)

This bit is used to support MMC test commands.

When this bit is set to 1, the CRC16 field in the received data is not checked on read commands and the CRC status token is not detected on write commands.

52.2.22 Swap Control Register (SDSWAP)

Address(es): SDHI.SDSWAP 8785 21E0h, SDHI_NS.SDSWAP 0785 21E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read and cannot be modified.	R
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b2	—	Reserved	This bit is 0 when read and cannot be modified.	R
b4, b3	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b5	—	Reserved	This bit is 0 when read and cannot be modified.	R
b6	BWSWP	SDBUFR Swap Write*1	0: Normal write operation 1: Swap the byte endian before writing to the SDBUFR register	R/W
b7	BRSWP	SDBUFR Swap Read*1	0: Normal read operation 1: Swap the byte endian before reading the SDBUFR register	R/W
b10 to b8	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12, b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b14, b13	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSWAP register is used to select whether or not the byte endian is swapped when accessing the SDBUFR register. Refer to section 52.3.1 for details on the differences in accessing the SDBUFR register based on the SDSWAP register value.

52.3 SDHI Operation

52.3.1 Data Block Format of the SD Card/MMC

The SDHI has a default bus mode (1-bit width) that uses just the SD0_DAT0 pin as a data line, a wide bus mode (4-bit width) that uses pins SD0_DAT0 to SD0_DAT3, and an MMC-only wide bus mode (8-bit width) that uses pins SD0_DAT0 to SD0_DAT7. Figure 52.2 shows the transfer format in default bus mode, Figure 52.3 shows the transfer format in wide bus mode (4-bit width), and Figure 52.4 shows the transfer format in wide bus mode (8-bit width).

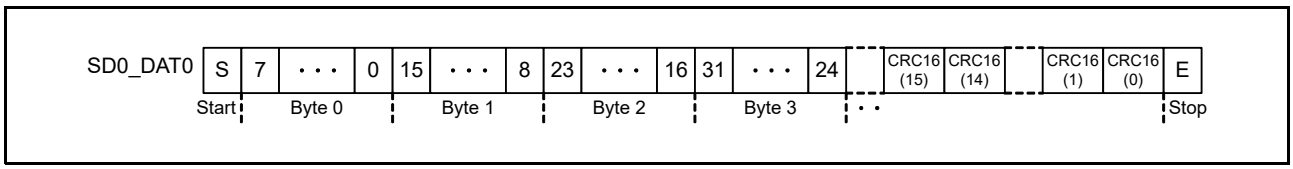


Figure 52.2 Transfer Format in Default Bus Mode (When the SDOPT.WIDTH Bit Is 1)

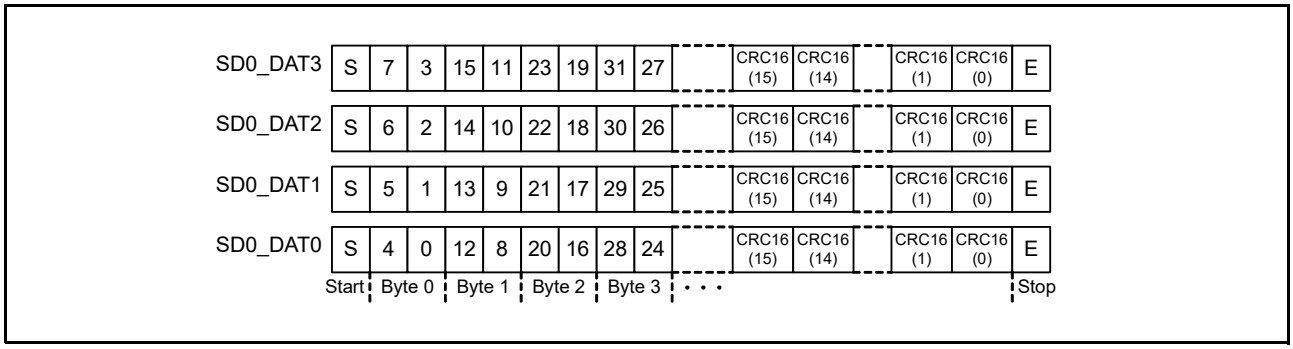


Figure 52.3 Transfer Format in Wide Bus Mode (4-bit Width) (When the SDOPT.WIDTH and WIDTH8 Bits Are 0)

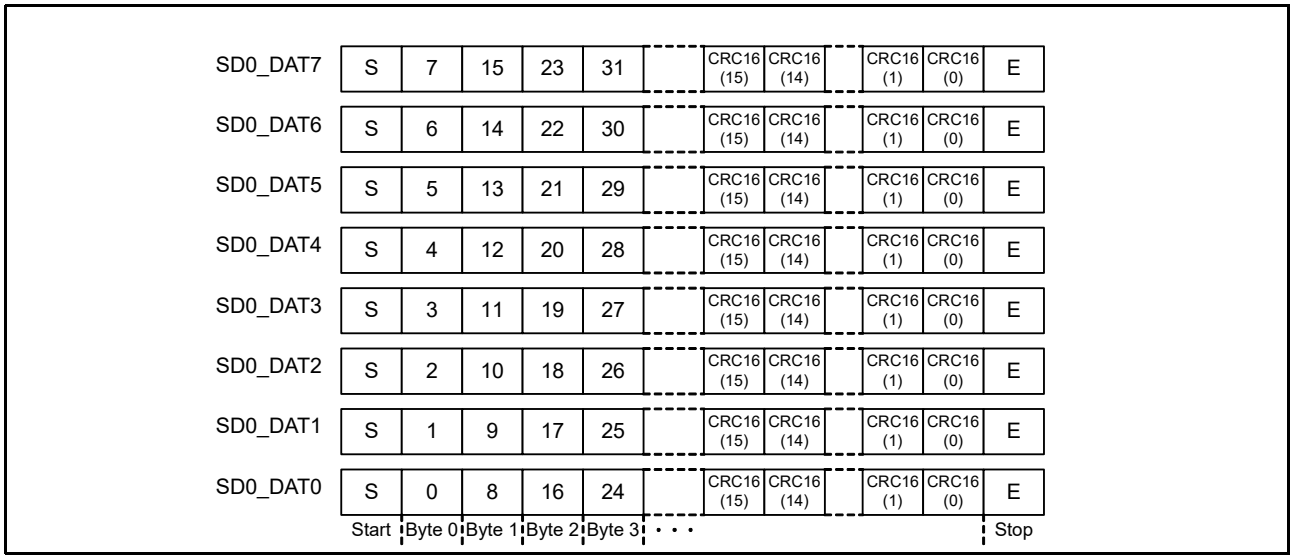


Figure 52.4 Transfer Format in Wide Bus Mode (8-bit Width) (When the SDOPT.WIDTH Bit Is 0 and WIDTH8 Bit Is 1)

52.3.2 SD Buffer and the SDBUFR Register

The SDHI transfers data to an SD card/MMC via its internal SD buffer. The SD buffer is comprised of a double buffer, and each buffer is 512 bytes. Figure 52.5 shows the data configuration of a single buffer of the SD buffer's double buffer.

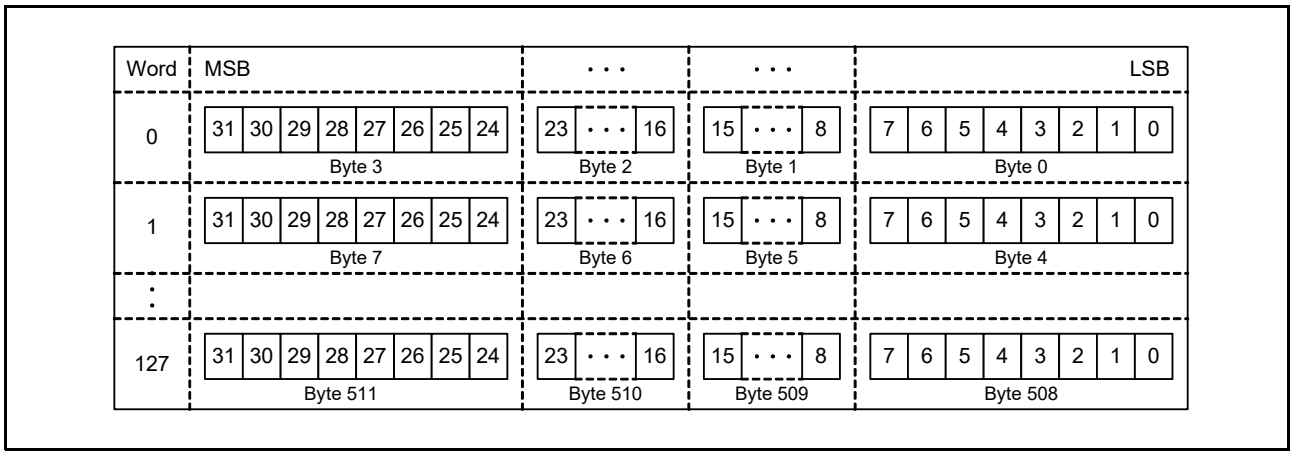


Figure 52.5 Data Configuration of Single Buffer in the SD Buffer

Access to the SD buffer is done via the SDBUFR register. If data is written to the SDBUFR register while the SDSWAP.BWSWP bit is 1, the SDHI swaps the endian for the byte, and stores the data in the SDBUFR register. If data is read from the SDBUFR register while the SDSWAP.BRSWP bit is 1, the data of the endian for the byte that was swapped can be read. Figure 52.6 shows the data alignment when reading the SDBUFR register.

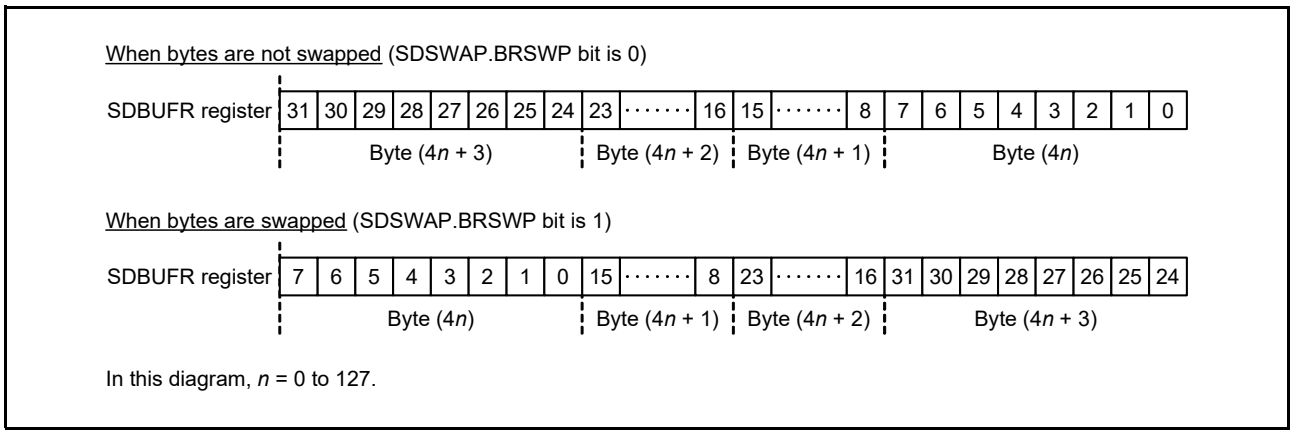


Figure 52.6 Data Alignment When Reading the SDBUFR Register

52.3.3 SD Card/MMC Detection

The SDHI can detect an SD card/MMC using either the SD0_CD pin or SD0_DAT3 pin.

52.3.3.1 Using the SD0_CD Pin to Detect an SD Card/MMC

Figure 52.7 shows the timing chart for SD card/MMC detection using the SD0_CD pin. The SD0_CD pin is connected to the card detection switch of the SD card/MMC connector, and is pulled-up by the MCU. The pull-up resistance value is determined by the specifications of the host device. Note that there are some SD card/MMC sockets whose card detection switches become open when the card is inserted.

- Detecting SD card/MMC insertion
  The signal from the SD0_CD pin becomes low when an SD card/MMC is inserted. This causes the SDSTS1.SDCDIN flag to become 1 if the SD0_CD pin is low for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDIN flag is cleared by setting it to 0.
- Detecting SD card/MMC removal
  The signal from the SD0_CD pin becomes high when the SD card/MMC is removed. This causes the SDSTS1.SDCDRM flag to become 1 if the SD0_CD pin is high for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDRM flag is cleared by setting it to 0.

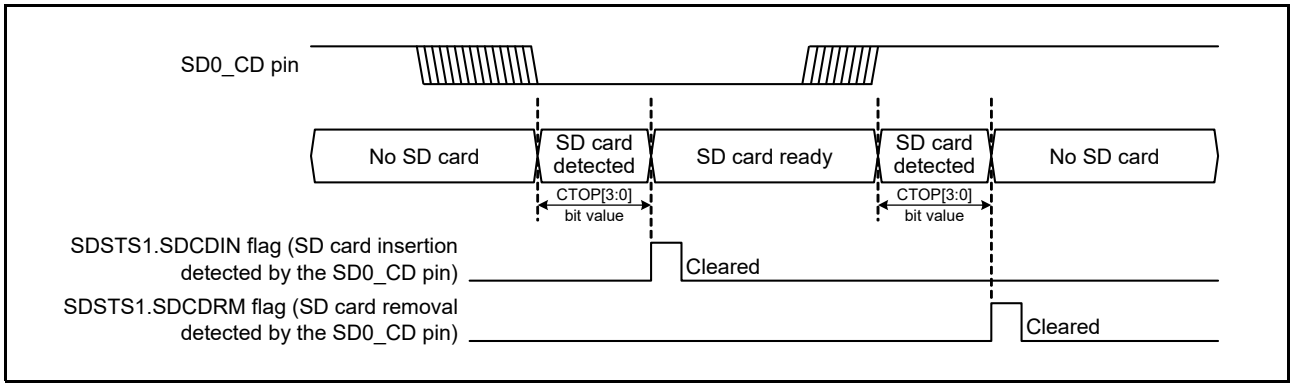


Figure 52.7 SD Card/MMC Detection Using the SD0_CD Pin

52.3.3.2 Using the SD0_DAT3 Pin to Detect an SD Card

Figure 52.8 shows the timing chart for SD card detection using the SD0_DAT3 pin. The SD0_DAT3 pin is pulled-down by the MCU. The pull-down resistance value is determined by the specifications of the host device.

- Detecting SD card insertion
  The signal from the SD0_DAT3 pin becomes high when an SD card is inserted. This causes the SDSTS1.SDD3IN flag to become 1. The SDSTS1.SDD3IN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SD0_DAT3 pin becomes low when the SD card is removed. This causes the SDSTS1.SDD3RM flag to become 1. The SDSTS1.SDD3RM flag is cleared by setting it to 0.

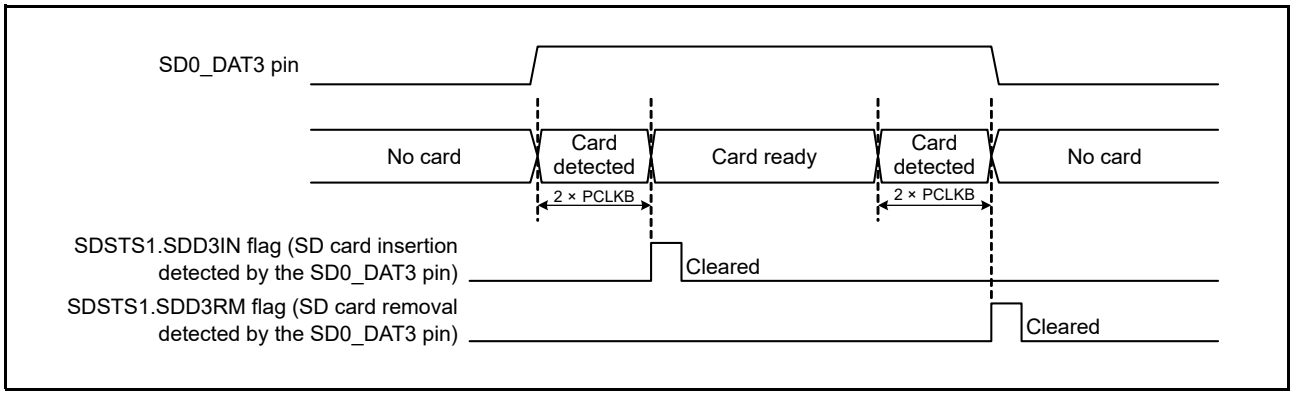


Figure 52.8 SD Card Detection Using the SD0_DAT3 Pin

52.3.4 SD Card/MMC Write Protection

The SDHI can disable writing to an SD card/MMC via the SD0_WP pin or a command.

52.3.4.1 Using the SD0_WP Pin to Enable Write Protection

The SD0_WP pin is connected to the WP detection switch of the SD card/MMC connector, and the SD0_WP pin is pulled-down or pulled-up when a card is inserted. The resistance value and whether the SD0_WP pin is pulled-up or pulled-down are determined by the specifications of the host device. The status of the SD0_WP pin is reflected in the SDSTS1.SDWPMON flag. After a card is inserted, read the SDSTS1.SDWPMON flag to check if write protection is enabled or disabled.

52.3.4.2 Using a Command to Enable Write Protection

The SDHI uses the write protect command or the lock command to disable writing to the SD card/MMC.

52.3.5 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SDSTS2 register becomes 1. Also, depending on the source of the error, the corresponding flag in the SDERSTS1 or SDERSTS2 register becomes 1.

The status flags in registers SDERSTS1 and SDERSTS2 become 0 by writing to the SDCMD register, or by setting the SDRST.SDRST bit to 0.

Table 52.7 Communication Errors

Communication Error	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register	Bit	Register	Bit	
End bit error	SDSTS2	ENDE	SDERSTS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error *1
				RSPLENE0	The response length is in error *2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response *1
				RSPCRCE0	There is a CRC error in the response *2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match *1
				CMDE0	The command index field value for the transmitted command and received response do not match *2

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.

Table 52.8 Timeouts

Timeout	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register	Bit	Register	Bit	
Response timeout	SDSTS2	RSPTO	SDERSTS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse *1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse *2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set *3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set *3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set *3 elapses
					After the read data is received, the next block read data is not received even after at least the period set *3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set *3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set *3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set *3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. The period is set in the SDOPT.TOP[3:0] bits.

52.3.6 Examples of Issuing a Command

52.3.6.1 Command Absent of Response Reception and Data Transfer

Figure 52.9 shows an example of no response being received and no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value to the SDARG register, write the command information to be sent to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a command is issued, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0.

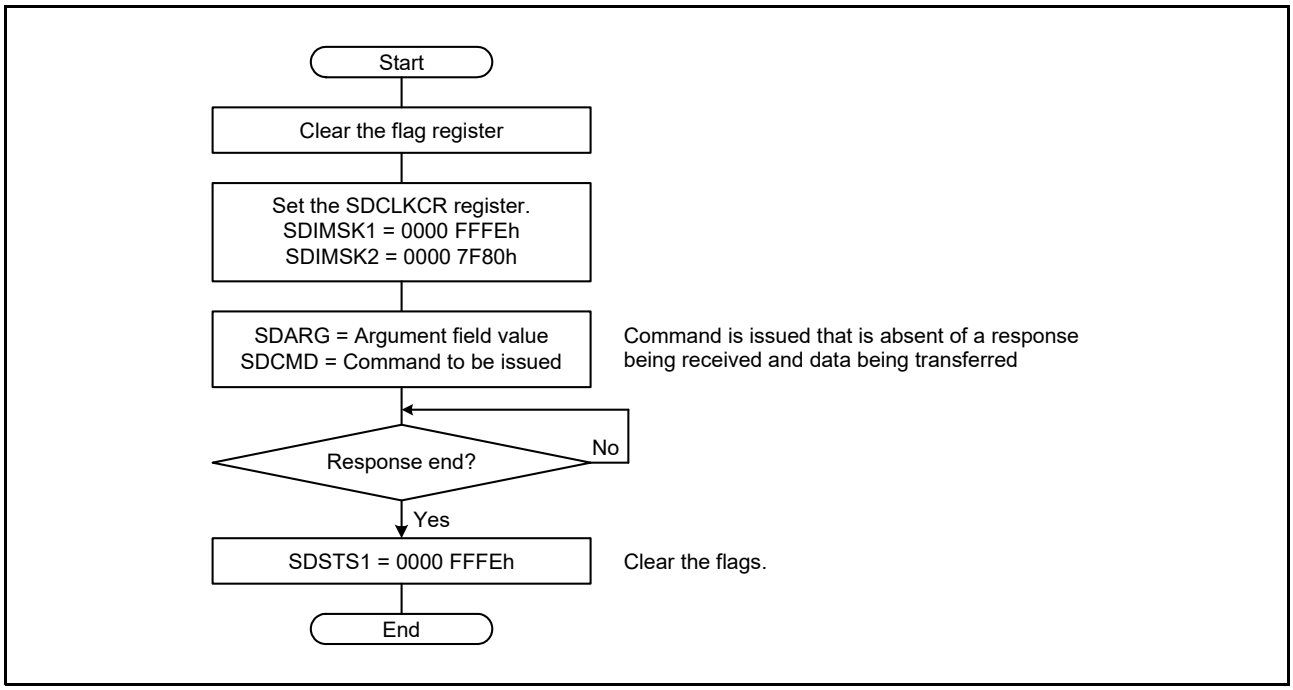


Figure 52.9 Command Issued That Is Absent of Response Reception and Data Transfer

52.3.6.2 Command Absent of Data Transfer

Figure 52.10 shows an example of no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value in the SDARG register, write the information of the command to be issued to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

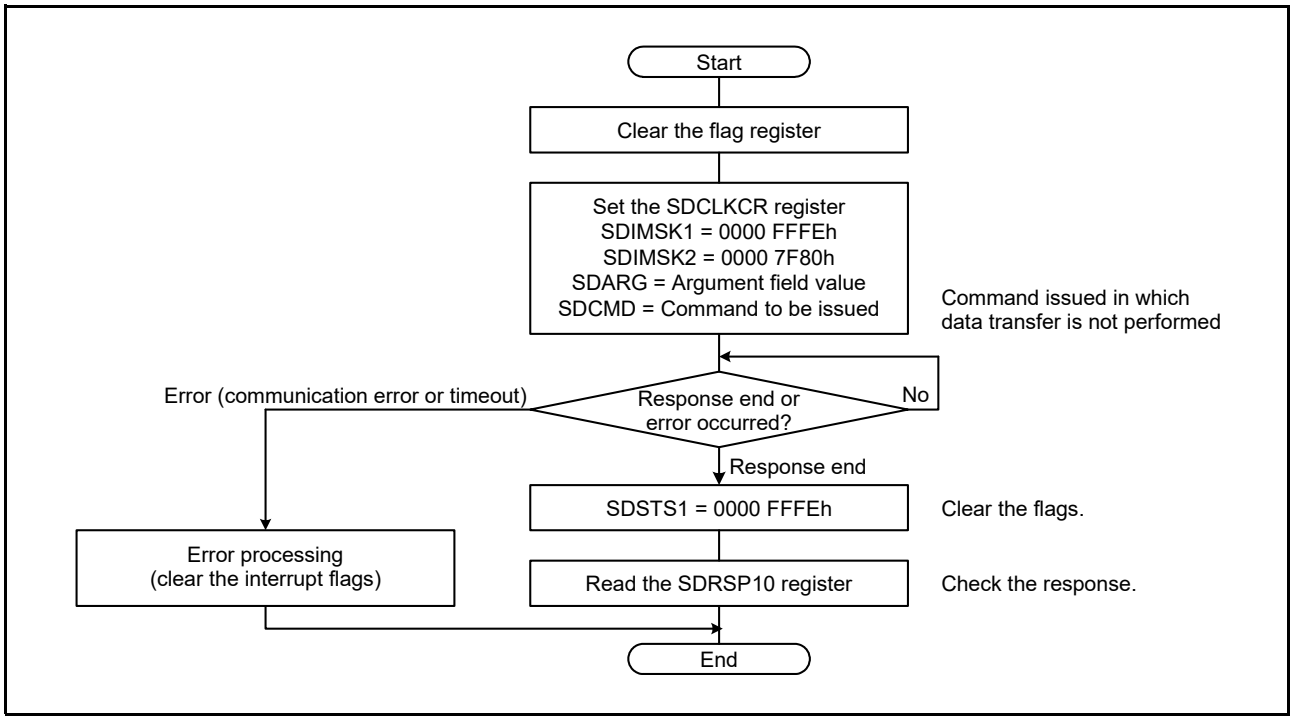


Figure 52.10 Command Issued That Is Absent of Data Transfer

52.3.6.3 Single Block Read Command (CMD17)

Figure 52.11 shows an example of issuing the single block read command (CMD17).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD17 to the SDARG register, write 0000 0011h to the SDCMD register. The SDHI issues CMD17 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register. If the response read is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that CMD12 and CMD52 are not automatically issued by stopping this command sequence.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0 and set the SDIMSK2.BREM bit to 0.
7. After the amount of data set in the SDSIZE.LEN[9:0] bits is received from the SD card/MMC, the SDSTS2.BRE flag becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register.
9. After data has been read from the SDBUFR register, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

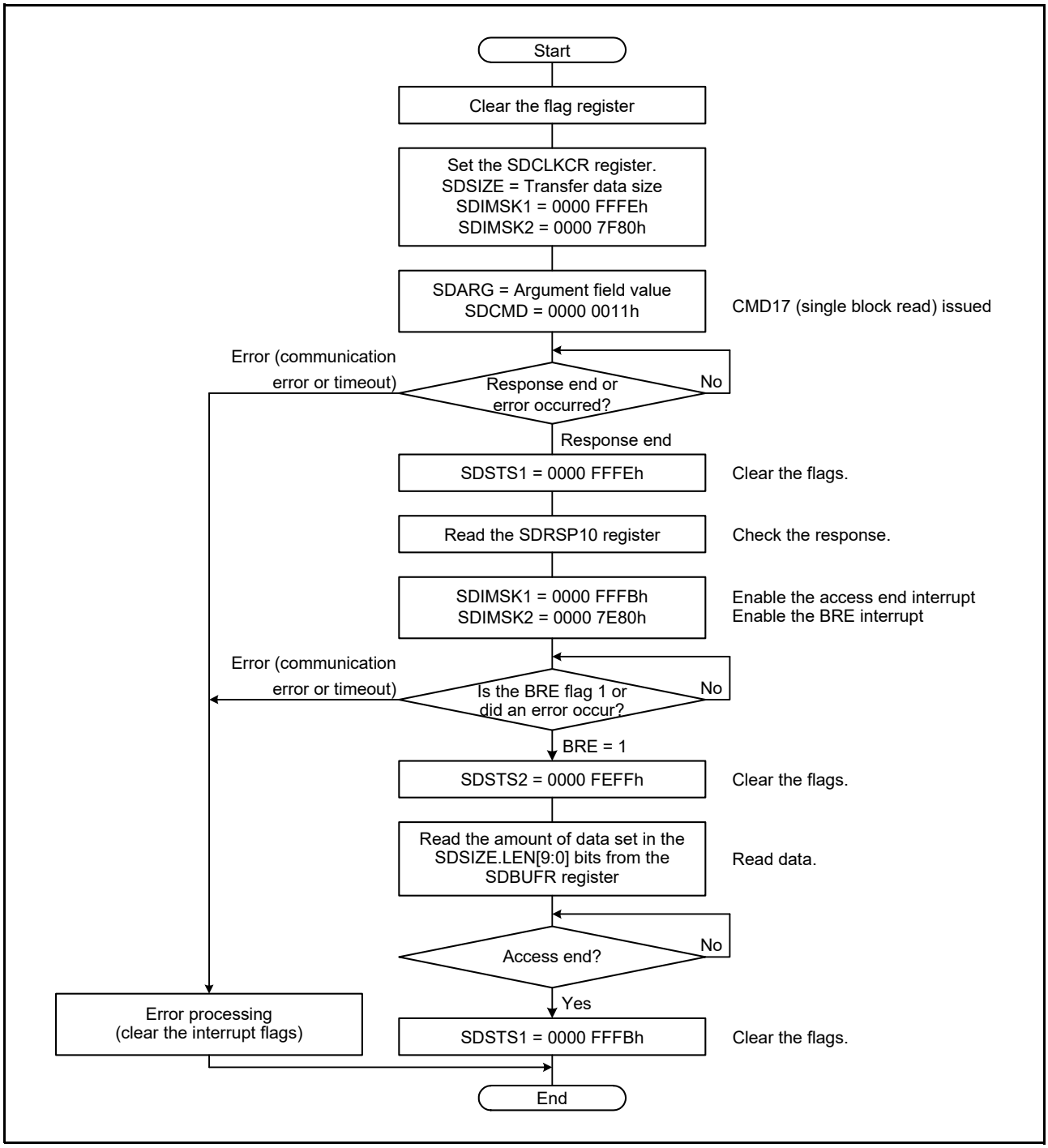


Figure 52.11 Issuing the Single Block Read Command

52.3.6.4 Single Block Write Command (CMD24)

Figure 52.12 shows an example of issuing the single block write command (CMD24).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD24 to the SDARG register, write 0000 0018h to the SDCMD register. The SDHI issues CMD24 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP10 register. If the read response is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that when this command sequence is stopped, CMD12 and CMD52 are not automatically issued.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BWEM bit to 0.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, the SDHI transmits write data to the SD card/MMC. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
9. After all data has been written to the SD card/MMC, the SDHI receives the CRC status token, and the SD0_DAT0 pin line becomes busy (low). Then, when the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

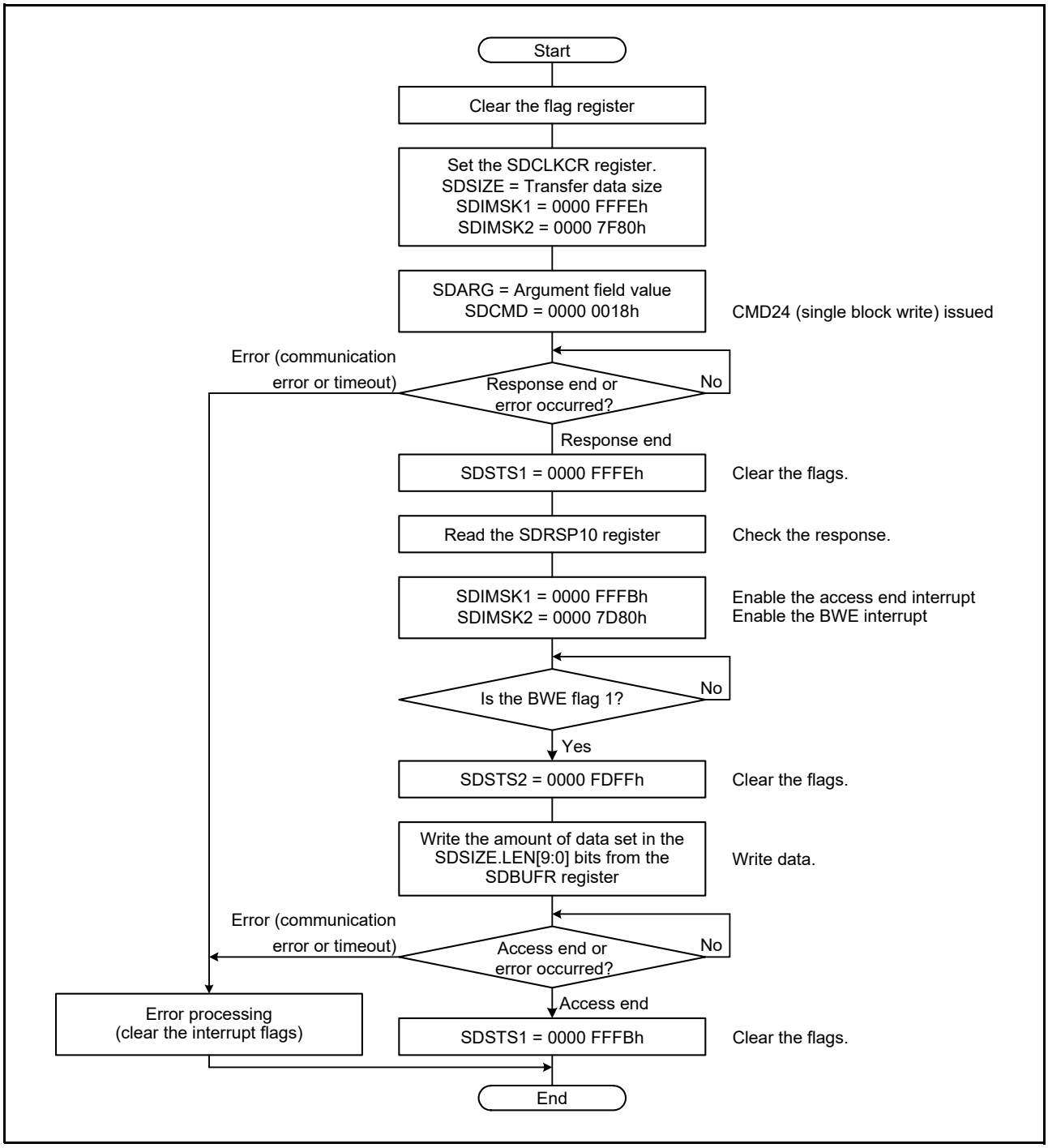


Figure 52.12 Issuing the Single Block Write Command

52.3.6.5 Multi-Block Read Command (CMD18)

Figure 52.13 shows an example of issuing the multi-block read command (CMD18).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD18 to the SDARG register, write 0000 0012h to the SDCMD register. The SDHI issues CMD18 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BREM bit to 0.
7. After receiving one block of data from the SD card/MMC, the SDSTS2.BRE bit becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register. The read access to the SDBUFR register repeats for the amount of transfer blocks set in the SDBLKCNT register. Also, while reading the SDBUFR register, data reception may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been read, the SDHI automatically issues CMD12, and the response is received. At this time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been received and the CMD12 response is received, the SDSTS1.ACEND flag becomes 1 and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

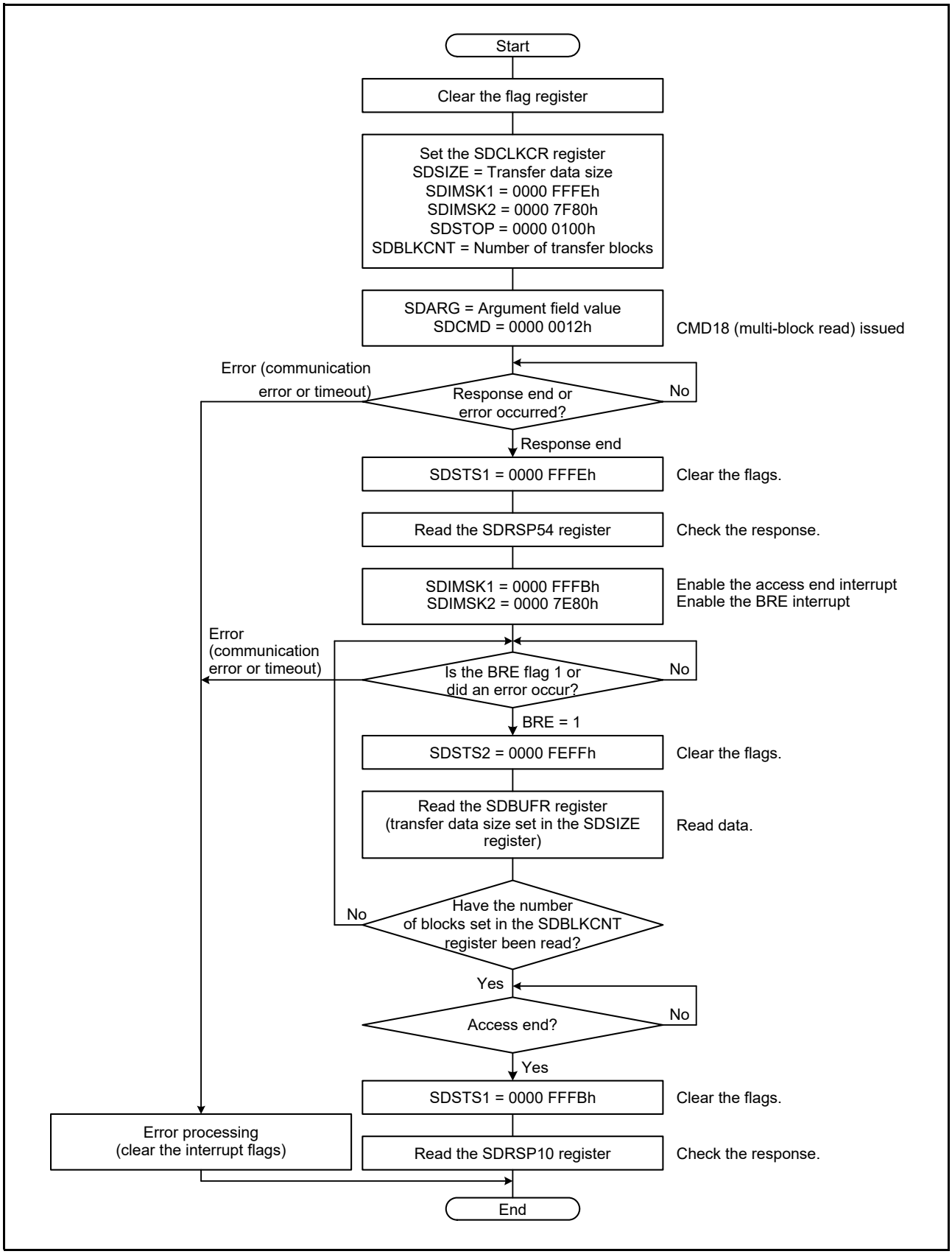


Figure 52.13 Issuing the Multi-Block Read Command

52.3.6.6 Multi-Block Write Command (CMD25)

Figure 52.14 shows an example of issuing the multi-block write command (CMD25).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD25 to the SDARG register, write 0000 0019h to the SDCMD register. The SDHI issues CMD25 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the SD card/MMC, the CRC status token is received, and the SD0_DAT0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been written, the SDHI automatically issues CMD12, and the response is received. At that time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

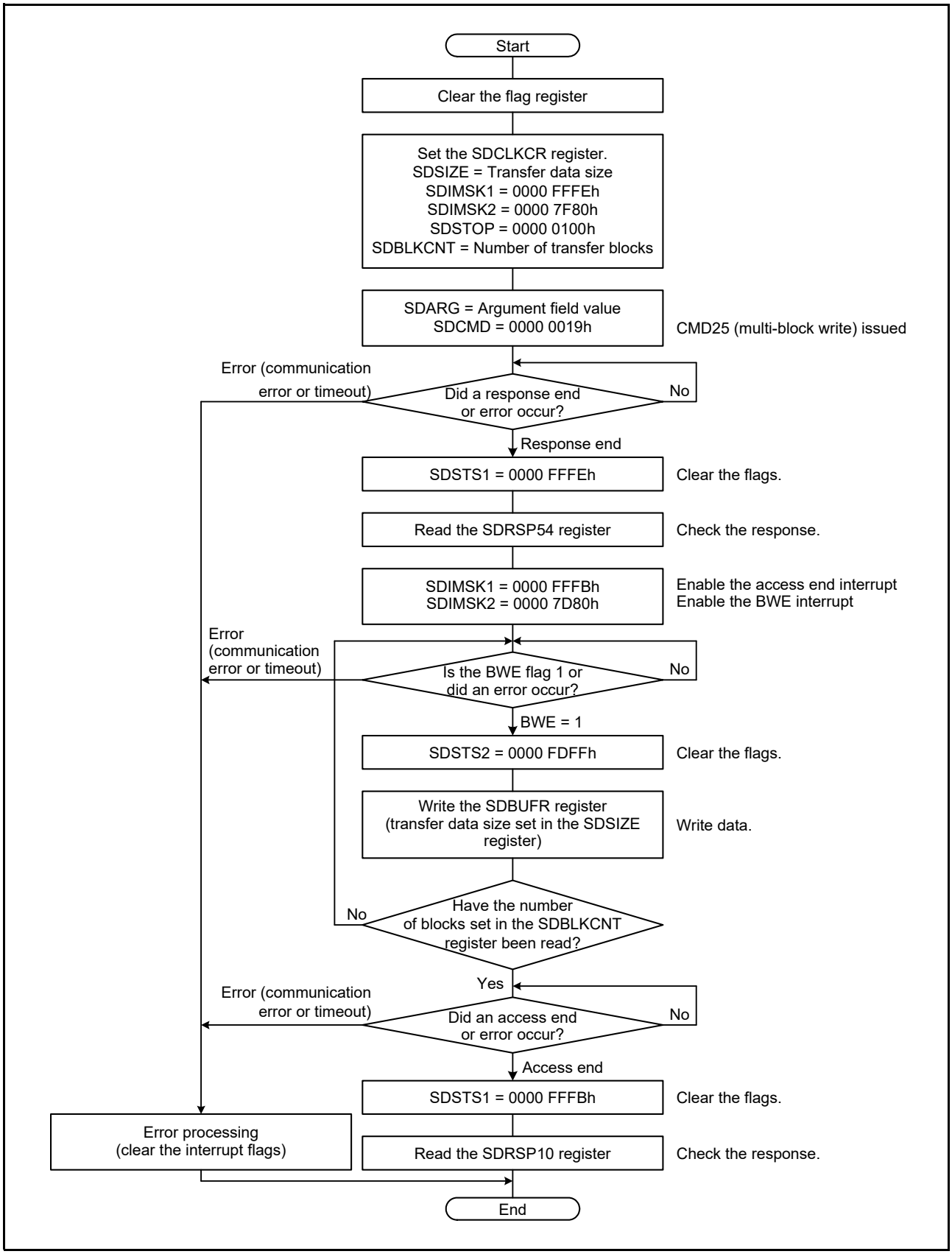


Figure 52.14 Issuing the Multi-Block Write Command

52.3.6.7 Multi-Block Write Command (CMD54)

Figure 52.15 shows an example of issuing the multi-block write command (CMD54).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 52.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD54 to the SDARG register, write 0000 6C36h to the SDCMD register. The SDHI issues CMD54 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. Set the SDOPT.TOUTMASK bit to 1 to disable timeout detection by the internal timer and start a timer outside the SDHI to be used for timeout measurement.
8. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
9. Set the SDSTS2.BWE flag to 0, and write the 512 bytes of data to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the MMC, the CRC status token is received, and the SD0_DAT0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
10. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
11. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error occurs. Perform a software reset if a timeout occurs.

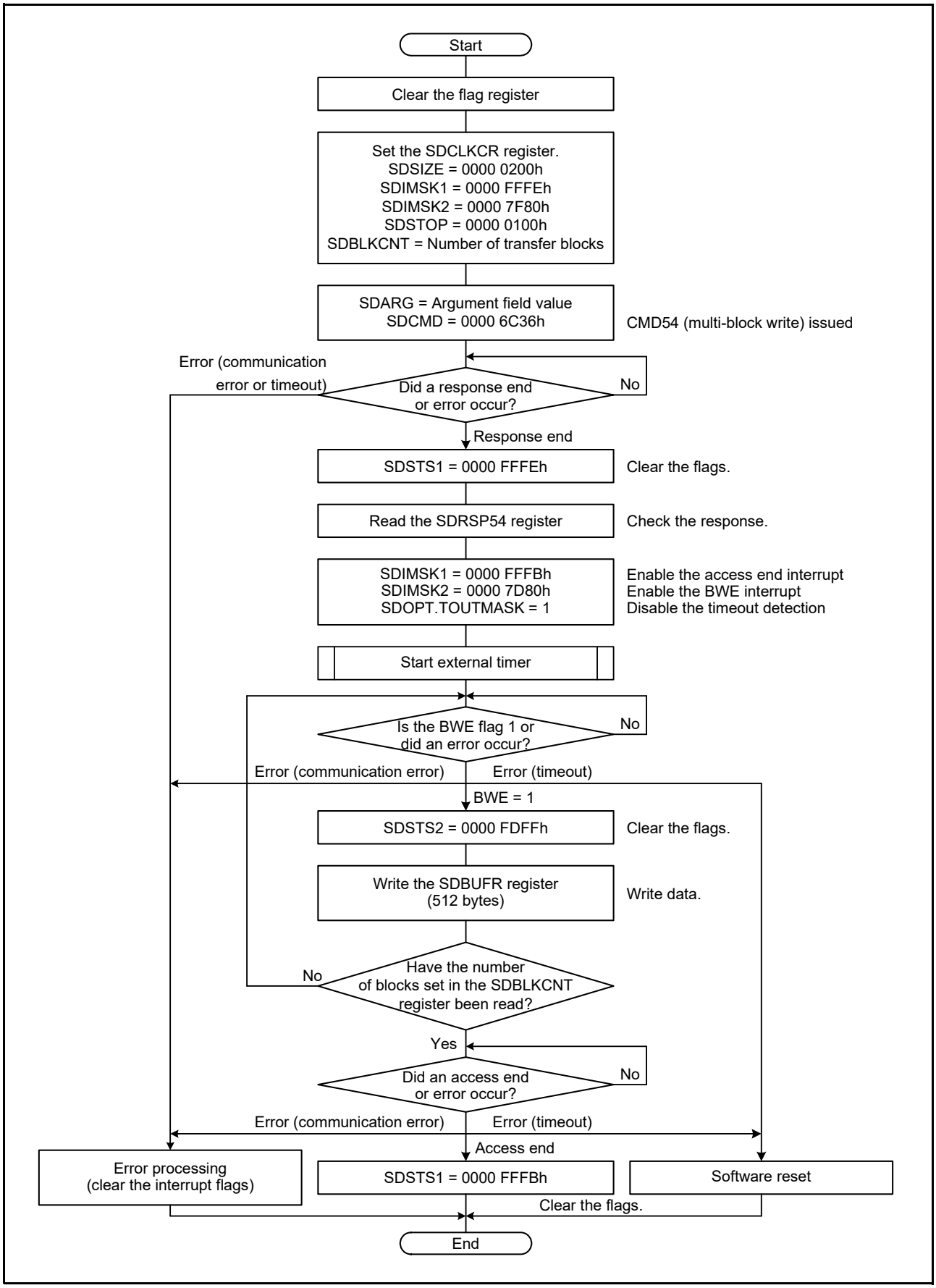


Figure 52.15 Issuing the Multi-Block Write Command

52.3.6.8 IO_RW_DIRECT Command (CMD52)

Figure 52.16 shows an example of issuing the IO_RW_DIRECT command (CMD52).

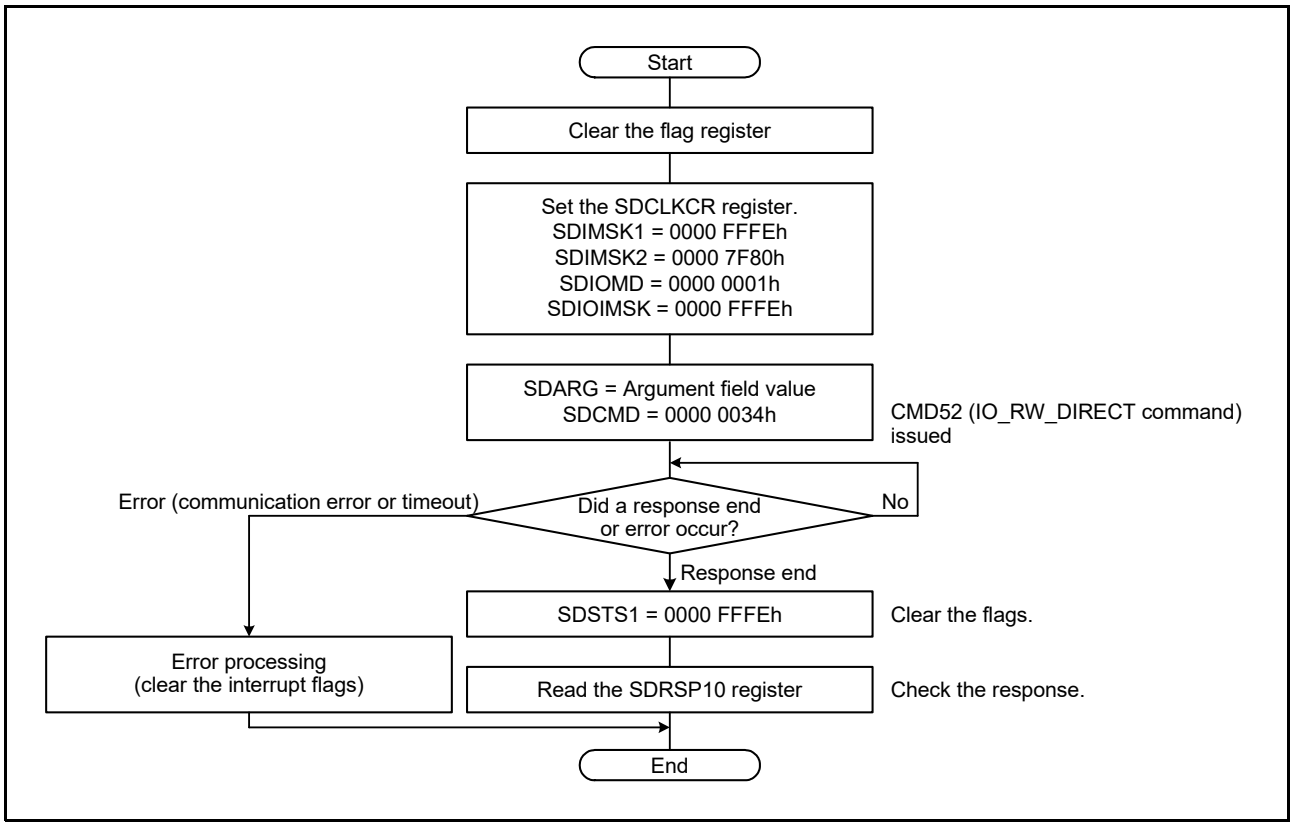


Figure 52.16 Issuing the IO_RW_DIRECT Command

52.3.6.9 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read))

Figure 52.17 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block read).

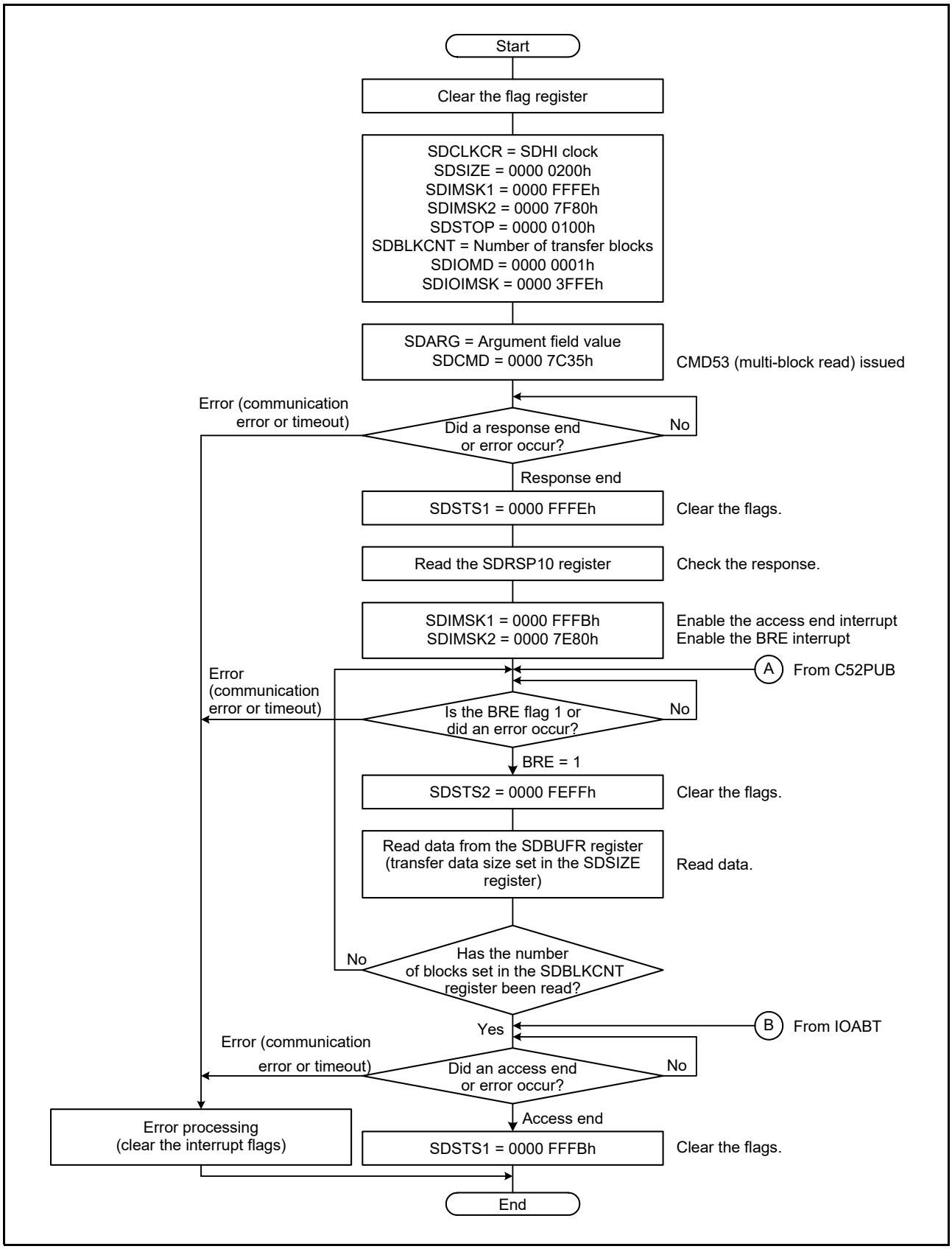


Figure 52.17 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 52.18 shows an example of SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block read) sequence.

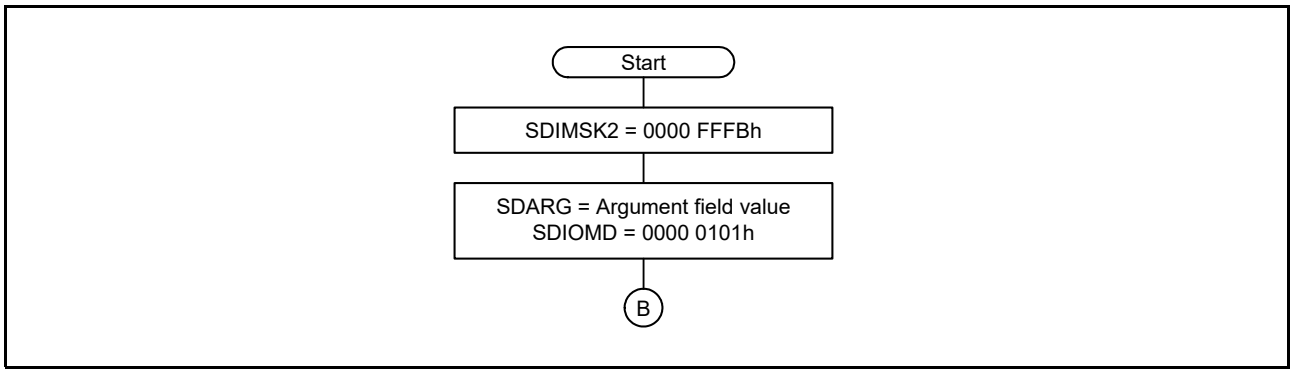


Figure 52.18 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Read) Sequence

Figure 52.19 shows an example of entering the read wait state and then issuing the SDIO none abort command (CMD52) during the IO_RW_EXTENDED command (CMD53/Multi-block read).

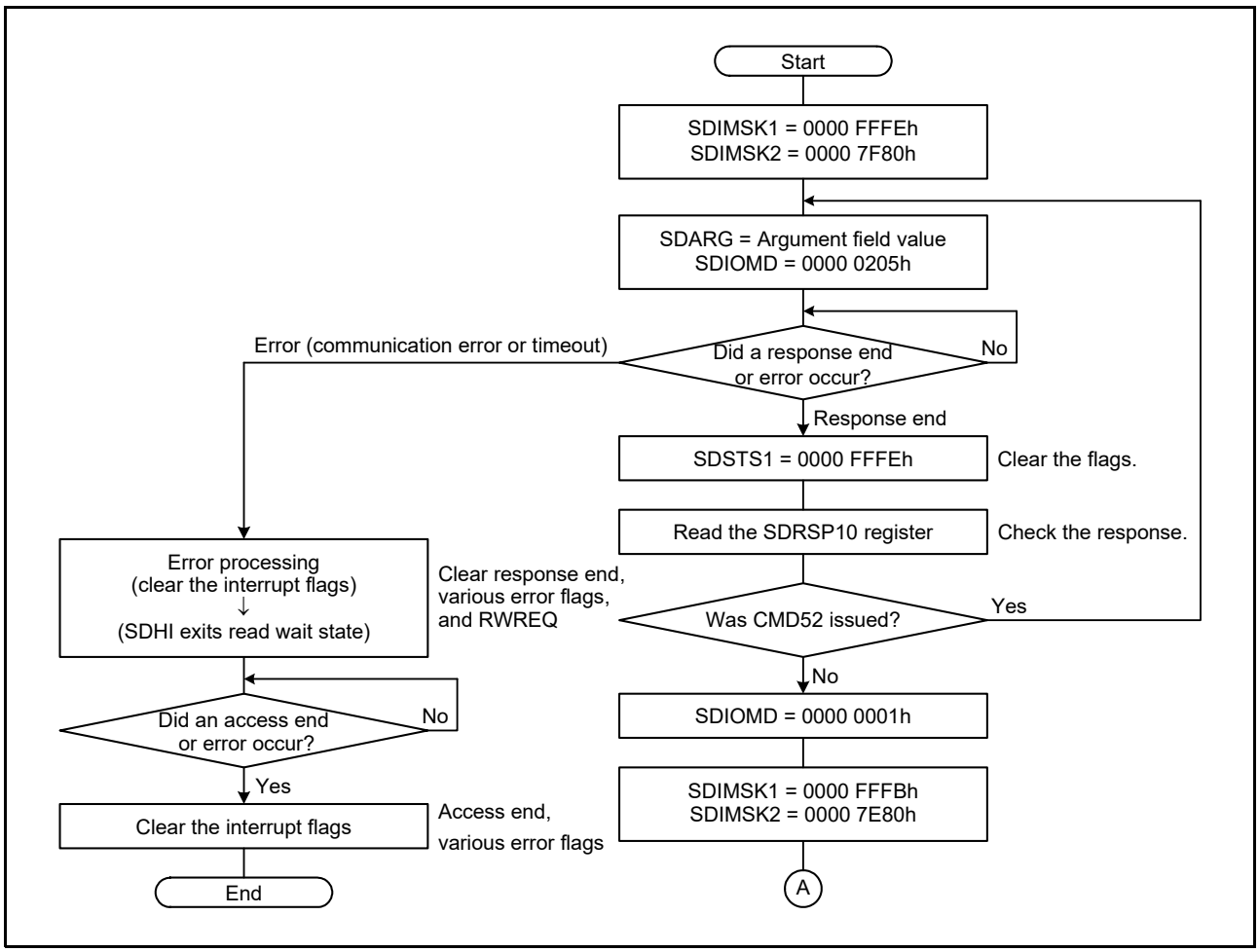


Figure 52.19 Entering the Read Wait State and Then Issuing the SDIO None Abort Command (CMD52) During the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

52.3.6.10 IO_RW_EXTENDED (CMD53 Multi-Block Write)

Figure 52.20 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block write).

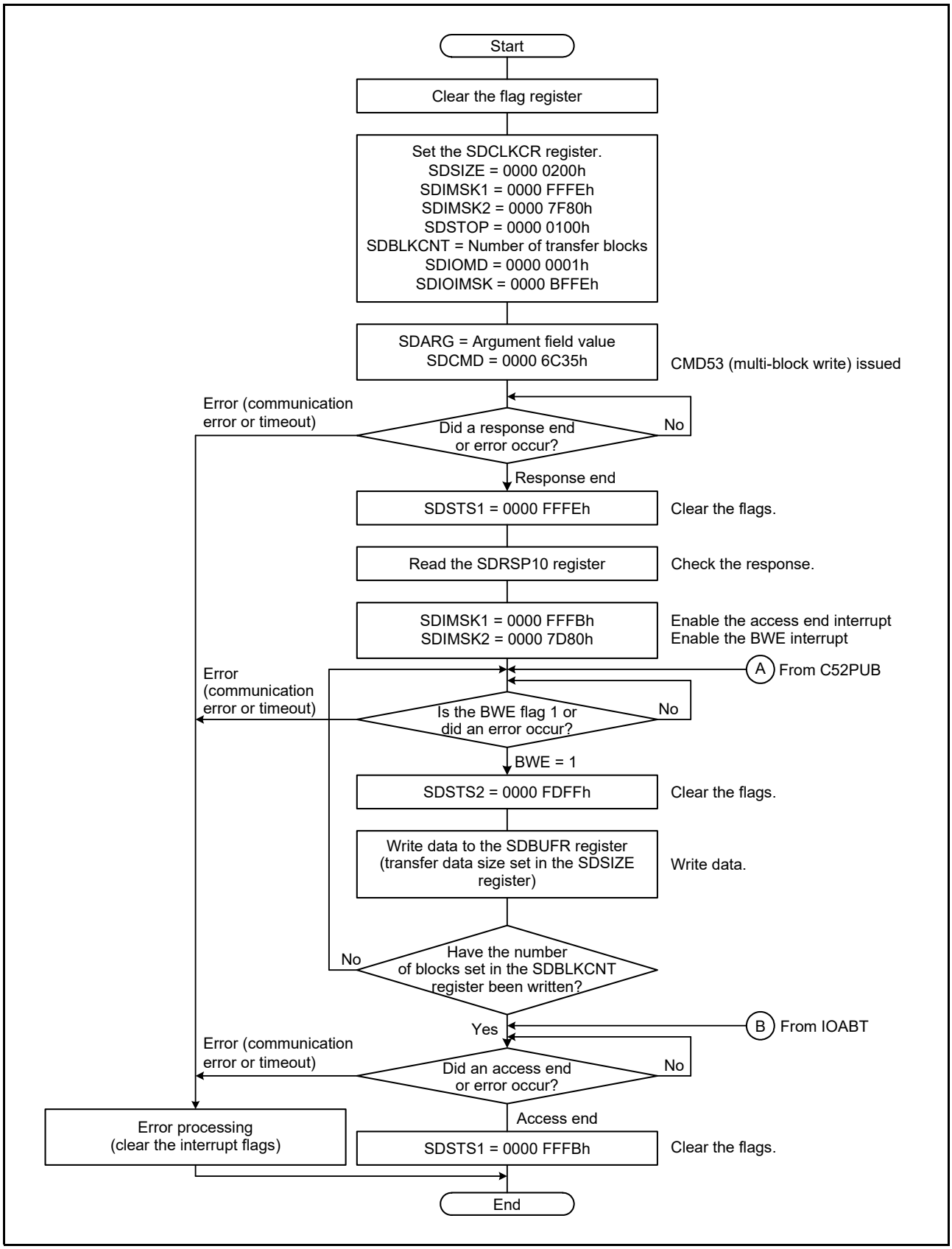


Figure 52.20 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Write)

Figure 52.21 shows SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53) sequence.

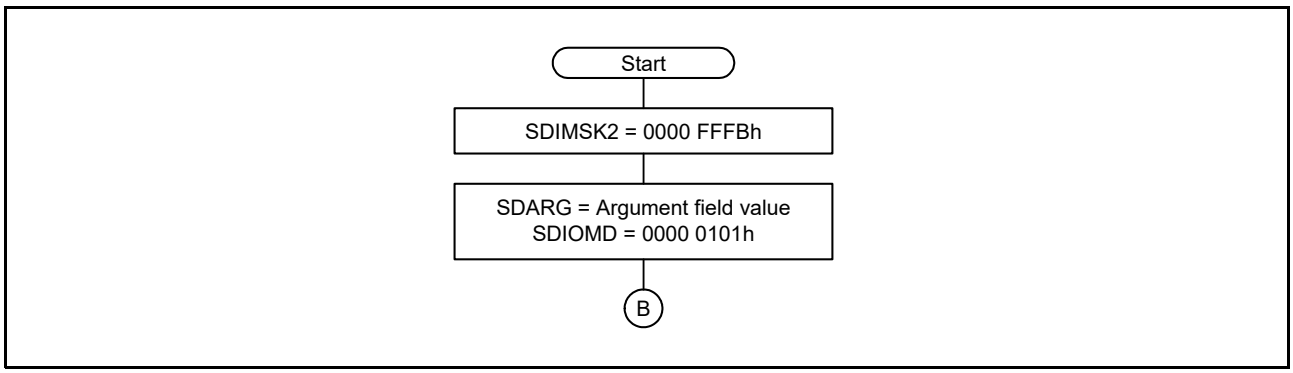


Figure 52.21 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53) Sequence

Figure 52.22 shows SDIO none abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block write) sequence.

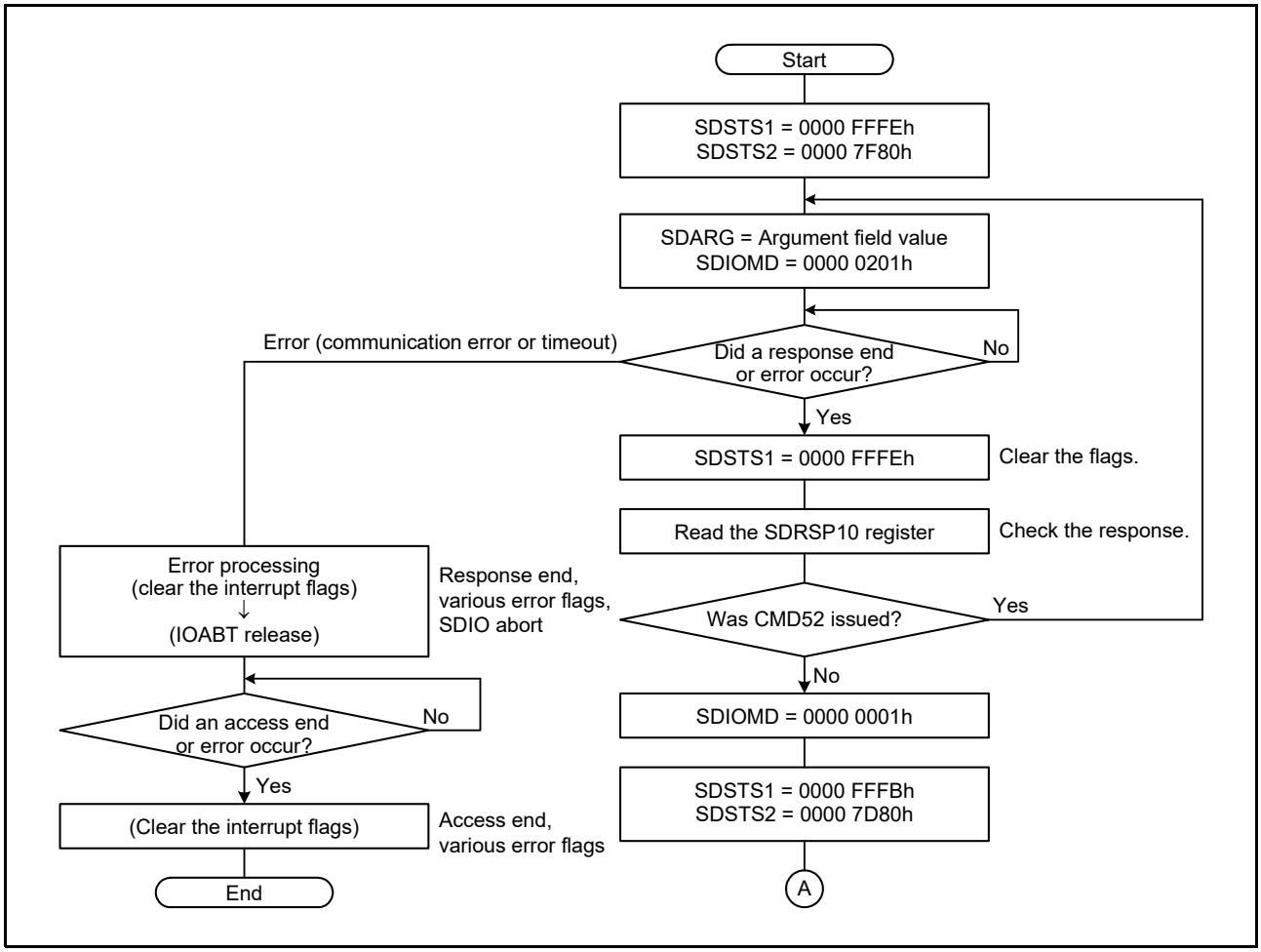


Figure 52.22 SDIO None Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Write) Sequence

52.3.6.11 DMA Transfer

Figure 52.23 shows an example of data being transferred from the SDBUFR register after the CMD18 multi-block read command is issued.

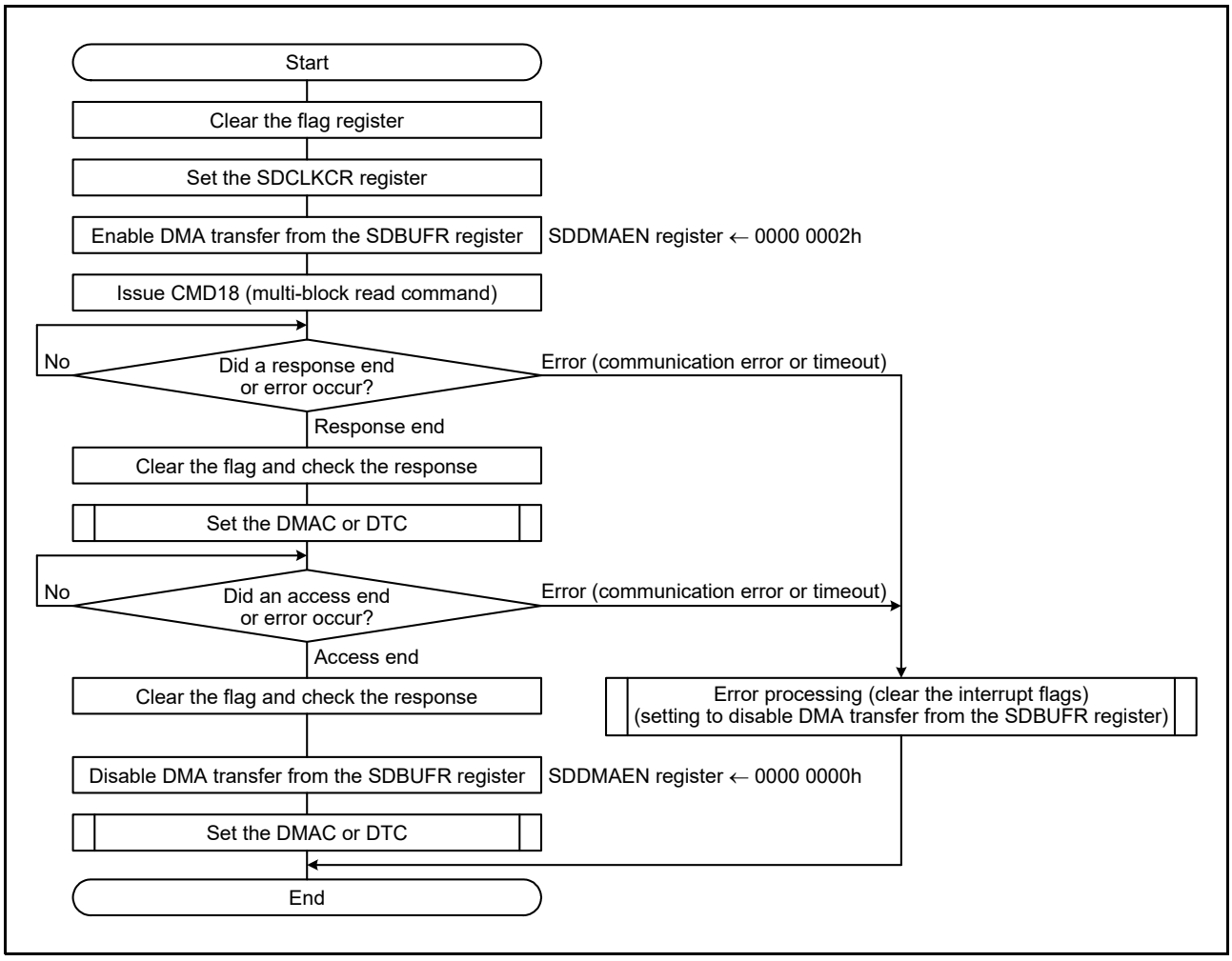


Figure 52.23 DMA Transfer After CMD18 is Issued

Figure 52.24 shows an example of data being DMA transferred to the SDBUFR register after the CMD25 multi-block write command is issued.

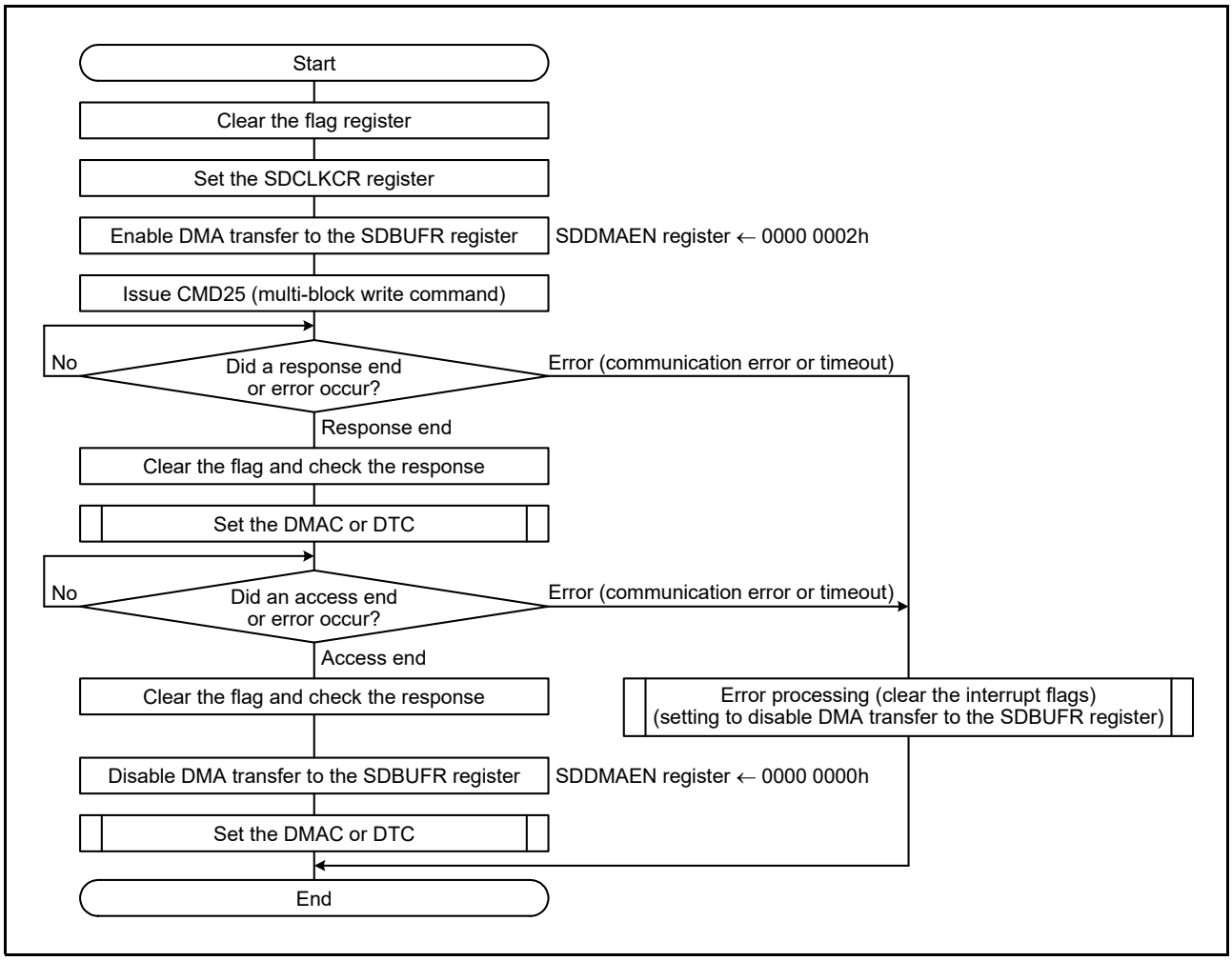


Figure 52.24 DMA Transfer After CMD25 is Issued

52.4 Interrupts

Table 52.9 lists the SDHI interrupt sources. When the status flags in registers SDSTS1, SDSTS2, and SDIOSTS become 1, if the corresponding bits in registers SDIMSK1, SDIMSK2, and SDIOIMSK are 0, the SDHI requests an interrupt. When clearing the status flags in registers SDSTS1, SDSTS2, and SDIOSTS, write 0 to the status flags to be cleared and write 1 to the states flags not being cleared.

Table 52.9 Interrupt Sources

Interrupt Source	Status Flag Register		Interrupt Mask/Enable Register		DMAC/DTC Triggerable
	Register	Bit	Register	Bit	
Card access interrupt (CACI)	SDSTS1	ACEND	SDIMSK1	ACENDM	No
		RSPEND		RSPENDM	
	SDSTS2	ILA	SDIMSK2	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDIO access interrupt (SDACI)	SDIOSTS	EXWT	SDIOIMSK	EXWTM	No
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
Card detection interrupt (CDETI)	SDSTS1	SDD3IN	SDIMSK1	SDD3INM	No
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	
SD buffer access interrupt (SBFAI)	SDSTS2	BWE	SDDMAEN	DMAEN	Yes
		BRE			

52.4.1 DMA Transfer Triggered by Interrupt Requests

When the SBFAI interrupt is requested, DMA/DTC transfer can be used to write to or read the SDBUFR register. When using the SBFAI interrupt, set the SDDMAEN.DMAEN bit to 1, the SDIMSK2.BWEM bit to 1, and SDIMSK2.BREM bit to 1.

When the SDDMAEN.DMAEN bit is 1, if a write command is issued, the SDSTS2.BWE flag becomes 1; if a read command is issued, the SDSTS2.BRE flag becomes 1. At this point, the SBFAI interrupt request is output. When the last data of a block is transferred (one block is the transfer data size set in the SDSIZE.LEN[9:0] bits), the SBFAI interrupt request is cleared, and the SDSTS2.BWE flag or the SDSTS2.BRE flag becomes 0.

The SBFAI interrupt request is also cleared by following:

- The SDRST.SDRST bit is set to 0 (SDHI software reset).
- The SDSTOP.STP bit is set to 1.
- The SDIOMD.IOABT bit is set to 1.
- The SDDMAEN.DMAEN bit is set to 0.
  However, if the DMAEN bit is set to 1 again before the next command is written to the SDCMD register, the SBFAI interrupt request is output again.

The SBFAI interrupt request will not be cleared when a communication error or timeout occurs during DMA transfer. Perform error processing by software.

The SDSTS2.BWE flag and BRE flag will not become 0 when a communication error or timeout occurs, nor will they become 0 when the SDSTOP.STP bit and SDIOMD.IOABT bit are set to 1. If the SDSTS2.BWE flag or BRE flag remain set to 1, even if a write command or read command is issued, the SBFAI interrupt request will not be output, and the SDSTS2.BWE flag and BRE flag must be set to 0 before issuing the next command.

Table 52.10 lists the DMAC and DTC settings when performing DMA transfer.

Table 52.10 DMAC and DTC Settings When Performing DMA Transfer

Item		Setting Description
Transfer mode		Block transfer mode
Transfer data	1 data	32 bits
	Block size	Size set in the SDSIZE.LEN[9:0] bits divided by 4
Number of block transfers		Number of transfers set in the SDBLKCNT register

52.5 Notes on Using the SDHI

52.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It

When the multi-block read command (CMD18) is issued to read one or two blocks, if the response to CMD18 stored in the SDRSP10 register is read, the timing of the read access may cause the response to be read incorrectly.
Figure 52.25 shows examples of a normal read and an illegal read when using CMD18 to read two blocks.

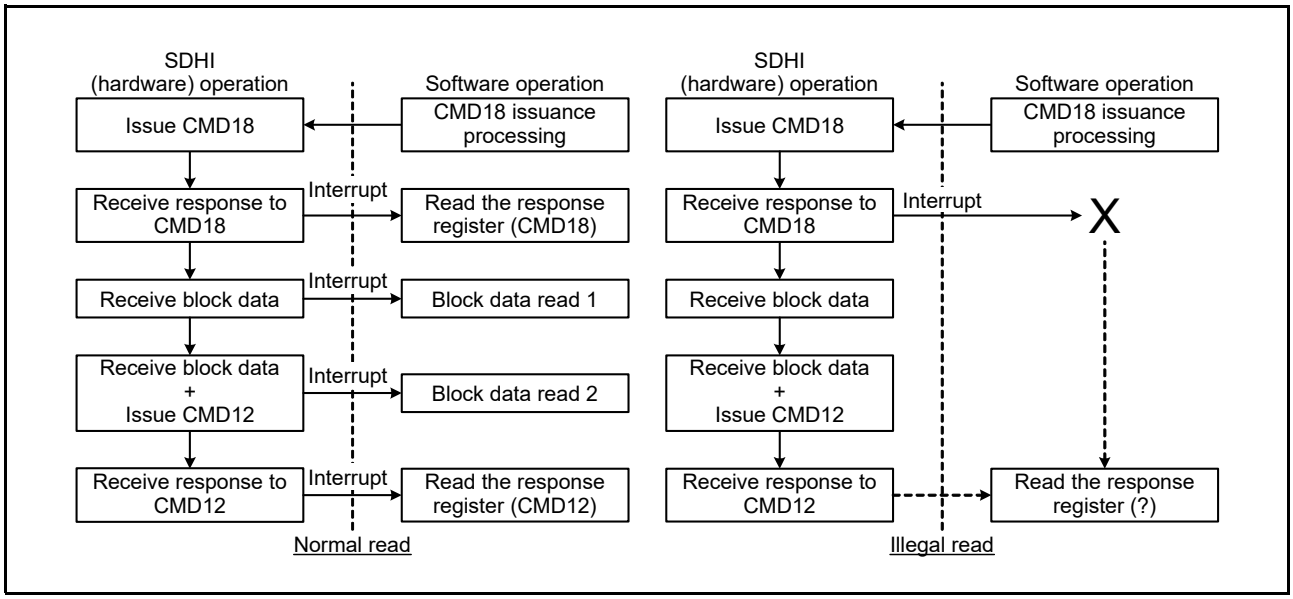


Figure 52.25 Multi-Block Read Processing When Reading Two Blocks

In the example of the illegal read, when the interrupt is generated by receiving the response to CMD18, the timing for reading the SDRSP10 register during the interrupt handling is delayed, and instead of the response for CMD18 being read, the response for CMD12 or the data during the response for CMD12 may be read.
This problem can be avoided by performing either of the following:

- When reading one block or two blocks, use a single block read command instead of a multi-block read command.
- When reading the response for CMD18, read the SDRSP54 register instead of the SDRSP10 register.

When using a multi-block read command to read three or more blocks, CMD12 will not be issued if no block data is read, which will prevent this problem from happening. Also, when using a multi-block write command, the above problem can be avoided by transmitting block data after reading the response to CMD25.

52.5.2 SDBUFR Register Illegal Write Error

When writing to the SDBUFR register after issuing a single block write command or a multi-block write command, write data for the size set by the SDSIZE.LEN[9:0] bits. If the amount of data written to the SDBUFR register is greater than the size set by the SDSIZE.LEN[9:0] bits, an illegal write error occurs in the SDBUFR register, and the SDSTS2.ILW flag becomes 1. However, the padding data included in the data being written to the SDBUFR register is ignored, so this error does not occur. For example, if the data size set by the SDSIZE.LEN[9:0] bits is an odd number, of the data written to the SDBUFR register, there is a remainder of 1 byte or 3 bytes. Although the extra data is written to the register, no error occurs. If the data size set by the SDSIZE.LEN[9:0] bits is an even number, and there is a remainder of 2 bytes, no error occurs even if these 2 bytes are written to the SDBUFR register.
If data written to the SDBUFR register is not transmitted, the SDSTS2.SDCLKCREN flag may remain set to 0. In this case, in order to set the SDSTS2.SDCLKCREN flag to 1, the SDRST.SDRST bit must be set to 0 and then set back to 1.

52.5.3 Automatic Control of the SDHI Clock Output

As per the SD card/MMC specifications, after MCU power-on, 74 cycles of the SDHI clock must be output from the host to the SD card/MMC before the card initialization command (CMD0) can be issued. Therefore, 74 cycles of the SDHI clock should be output from the SDHI to the SD card/MMC before enabling automatic control of the SDHI clock output. When automatic control of the SDHI clock output is enabled, clock output from the SD0_CLK pin stops if the command sequence is ended by a communication error or timeout. Therefore, if it is necessary to change the internal status of the SD card/MMC even after the command sequence ends, disable automatic control of the SDHI clock output and output the SDHI clock to the SD card/MMC.

52.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence

During a CMD53 multi-block write sequence, if the SDIOMD.C52PUB bit is set to 1, the SDHI issues CMD52 after the SD buffer becomes empty. To immediately issue CMD52, perform one of the procedures below to suspend writing to the SD buffer, and set the C52PUB bit to 1.

Procedure to suspend writing to the SD buffer when not performing DMA transfer (interrupt used)

1. Set the SDIMSK2.BWEM bit to 1 to disable the interrupt, and suspend writing to the SDBUFR register.
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, set the SDIMSK2.BWEM bit to 0 to enable the interrupt, and resume writing to the SDBUFR register.

Procedure to suspend writing to the SD buffer when performing DMA transfer

1. Configure settings to perform DMA transfer every [SDSIZE register setting value × *n* blocks], and suspend writing to the SDBUFR register before setting the SDIOMD.C52PUB bit (*n* = 1, 2, ...).
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, resume DMA transfer to the SDBUFR register.

52.5.5 Note on Setting the SDCLKCR Register

The SDCLKCR register cannot be written when the SDSTS2.SDCLKCREN flag is 0. Set the SDSTS2.SDCLKCREN flag to 1 before writing to the SDCLKCR register.

52.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence

When the SDSTOP.SDBLKCNTEN bit is 1 during a multi-block read sequence, if the SDSTOP.STP bit is set to 1 and the command sequence is stopped, the command sequence may not be completed depending on when the SDSTOP.STP bit is set to 1. To avoid this problem, set the SDSTOP.STP bit and the SDSTOP.SDBLKCNTEN bit to 0 simultaneously. Note that at this time, the SDSTOP.SDBLKCNTEN bit should be set to 0 even if the SDSTS2.SDCLKCREN bit is 0. If the SDSTOP.SDBLKCNTEN bit is not set to 0, the command sequence can be completed by setting the SDRST.SDRST bit to 0.
During a CMD53 multi-block transfer, when stopping data transfer by setting the SDIOMD.IOABT bit to 1, the SDSTOP.SDBLKCNTEN bit should remain set to 1.

52.5.7 Controlling Module Operation

SDHI operation is controlled by setting the MSTPCRC.MSTPC12 bit. Setting the MSTPC12 bit to 0 enables the SDHI; setting the MSTPC12 bit to 1 disables the SDHI. The SDHI is disabled after a reset. Registers in the SDHI can be accessed by setting the MSTPC12 bit to 0. Refer to section 11, Low Power Consumption for details.

53. Parallel Data Capture Unit (PDC)

53.1 Overview

This MCU includes a single parallel data capture unit (PDC).
The PDC has the function of communicating with the external I/O devices including image sensors, and specifically transferring parallel data such as an image output from the external I/O devices via the DTC or DMAC to the on-chip RAM and external address spaces (the CS, SDRAM and XSPIareas).

Table 53.1 PDC Specifications

Item	Description
Capture Range	• Desired amounts of parallel data within the following ranges in the vertical and horizontal directions. Vertical direction: 1 to 4095 lines Horizontal direction: 4 to 4095 bytes
Parallel data transfer clock (PIXCLK)	• Operating frequency: 1 to 27 MHz*1
Interrupt Sources	• Receive data ready • Frame end • Overrun • Underrun • Error in the setting for the number of lines • Error in setting for the number of bytes per line
Startup of DTC/DMAC	• The receive data ready interrupt is capable of starting the DTC or DMAC
Parallel data transfer clock output (PCKO)	• Operating frequency: 1 to 37.5 MHz*2 • Clock source: Peripheral module clock B (PCLKB) • Frequency division ratio: Selectable from 2, 4, 6, 8, 10, 12, 14, and 16.
Others	• PDC reset function • Selectable active sense for the VSYNC and HSYNC signals • Monitoring of the VSYNC and HSYNC signals • Endian selection
Reducing Power Consumption	• The PDC can be placed in the module stop state.
Protect Zone filter	• Security attribution can be set.*3

Note 1. The frequency of the parallel data transfer clock (PIXCLK) should be equal to or less than 0.6 × PCLKB (peripheral module clock).
Note 2. The operating frequency is 37.5 MHz when that of the peripheral module clock B (PCLKB) is 75 MHz and the frequency division ratio is 2.
Note 3. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

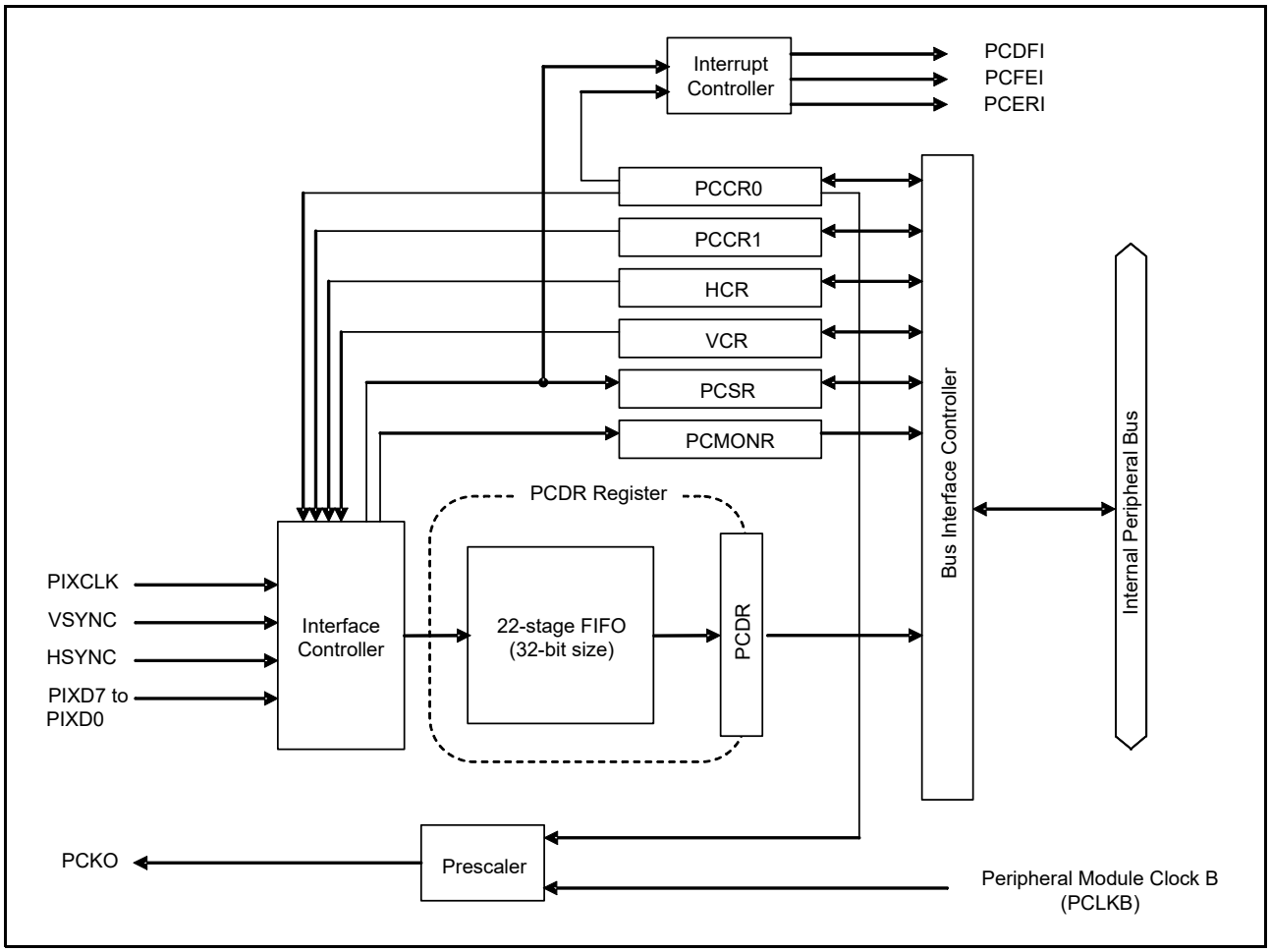


Figure 53.1 Block Diagram of the PDC

Table 53.2 Input and Output Pins of the PDC

Pin Name	Input/Output	Description
PIXCLK	Input	Parallel data transfer clock
VSYNC	Input	Vertical synchronization signal
HSYNC	Input	Horizontal synchronization signal
PIXD7 to PIXD0	Input	8-bit data
PCKO	Output	Output of the parallel data transfer clock

53.2 Register Descriptions

53.2.1 PDC Control Register 0 (PCCR0)

Address(es): PDC.PCCR0 8784 1000h, PDC_NS.PCCR0 0784 1000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EDS	PCKDIV[2:0]			PCKOE	HERIE	VERIE	UDRIE	OVIE	FEIE	DFIE	PRST	HPS	VPS	PCKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCKE	PIXCLK Input Enable	0: PIXCLK input is disabled. 1: PIXCLK input is enabled.	R/W
b1	VPS	VSYNC Signal Polarity Select	0: VSYNC signal is active high. 1: VSYNC signal is active low.	R/W
b2	HPS	HSYNC Signal Polarity Select	0: HSYNC signal is active high. 1: HSYNC signal is active low.	R/W
b3	PRST	PDC Reset	0: PDC reset is not applied. 1: PDC is reset.	R/(W) *1
b4	DFIE	Receive Data Ready Interrupt Enable	0: Generation of receive data ready interrupt requests is disabled. 1: Generation of receive data ready interrupt requests is enabled.	R/W
b5	FEIE	Frame End Interrupt Enable	0: Generation of frame end interrupt requests is disabled. 1: Generation of frame end interrupt requests is enabled.	R/W
b6	OVIE	Overrun Interrupt Enable	0: Generation of overrun interrupt requests is disabled. 1: Generation of overrun interrupt requests is enabled.	R/W
b7	UDRIE	Underrun Interrupt Enable	0: Generation of underrun interrupt requests is disabled. 1: Generation of underrun interrupt requests is enabled.	R/W
b8	VERIE	Vertical Line Number Setting Error Interrupt Enable	0: Generation of vertical line number setting error interrupt requests is disabled. 1: Generation of vertical line number setting error interrupt requests is enabled.	R/W
b9	HERIE	Horizontal Byte Number Setting Error Interrupt Enable	0: Generation of horizontal byte number setting error interrupt requests is disabled. 1: Generation of horizontal byte number setting error interrupt requests is enabled.	R/W
b10	PCKOE	PCKO Output Enable	0: PCKO output is disabled (fixed to the high level) 1: PCKO output is enabled.	R/W
b13 to b11	PCKDIV[2:0]	PCKO Frequency Division Ratio Select	b13 b11 0 0 0: PCKO/2 0 0 1: PCKO/4 0 1 0: PCKO/6 0 1 1: PCKO/8 1 0 0: PCKO/10 1 0 1: PCKO/12 1 1 0: PCKO/14 1 1 1: PCKO/16	R/W
b14	EDS	Endian Select	0: Little endian 1: Big endian	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

The PCCR0 register should only be set while the PCE bit in the PCCR1 register is 0.

PCKE Bit (PIXCLK Input Enable)

This bit enables and disables the input through the PIXCLK pin.
Set this bit to 1 before enabling reception. After enabling input through the PIXCLK pin, use the PRST bit to initialize the PDC.
When setting this bit to 0, do so after disabling reception operations.

VPS Bit (VSYNC Signal Polarity Select)

This bit selects the active sense of the VSYNC signal.

HPS Bit (HSYNC Signal Polarity Select)

This bit selects the active sense of the HSYNC signal.

PRST Bit (PDC Reset)

This bit initializes the internal status of the PDC and the target registers of the PDC reset. Refer to section 53.3.11, Reset State for the target registers.
The PDC should be reset after setting the PCKE bit to 1.
When 1 is written to the PRST bit, initialization is started in synchronization with the PIXCLK. After completion of the initialization, the PRST bit is automatically cleared to 0. When the PDC is reset, ensure that the PIXCLK pin has an input signal. Also, after 1 has been written to the PRST bit, do not proceed to the next step until verifying that the bit has returned to 0.
In the case of consecutive PDC resets, wait for at least one cycle of PIXCLK after verifying that the PRST bit has returned to 0.

DFIE Bit (Receive Data Ready Interrupt Enable)

This bit enables and disables the generation of receive data ready interrupt requests.

FEIE Bit (Frame End Interrupt Enable)

This bit enables and disables the generation of frame end interrupt requests.

OVIE Bit (Overrun Interrupt Enable)

This bit enables and disables the generation of overrun interrupt requests.

UDRIE Bit (Underrun Interrupt Enable)

This bit enables and disables the generation of underrun interrupt requests.

VERIE Bit (Vertical Line Number Setting Error Interrupt Enable)

This bit enables and disables the generation of vertical line number setting error interrupt requests.

HERIE Bit (Horizontal Byte Number Setting Error Interrupt Enable)

This bit enables and disables the generation of horizontal byte number setting error interrupt requests.

PCKOE Bit (PCKO Output Enable)

This bit enables and disables an output from PCKO.
When the PCKOE bit is cleared to 0 during low output of PCKO, it may cause high output at the time of clearing, resulting in spoiling the duty factor.

PCKDIV[2:0] Bits (PCKO Frequency Division Ratio Select)

This bit selects the frequency division ratio of the PCKO.
The PCKO output is a clock signal derived by dividing the PCLKB clock signal by the value from 2 to 16 corresponding to the setting of the PCKDIV[2:0] bits.
The operating frequency should be set in accord with that of PCLKB to a value within the range from 1 to 37.5 MHz.

EDS Bit (Endian Select)

This bit selects an endian for the captured data.

53.2.2 PDC Control Register 1 (PCCR1)

Address(es): PDC.PCCR1 8784 1004h, PDC_NS.PCCR1 0784 1004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCE	PDC Operation Enable	0: Operations for reception are disabled. 1: Operations for reception are enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PCE Bit (PDC Operation Enable)

This bit enables and disables operations for reception.
When the PCE bit is set to 1 during assertion of the VSYNC signal, the PDC starts operations for reception from the next effective edge of the VSYNC signal.
The PCE bit should only be cleared to 0 while operations for reception or continued reception are stopped, including the frame end interrupt and so on. Regarding operations for continued reception, refer to section 53.3.6, Operations for Continued Reception at the Time of Frame End.

53.2.3 PDC Status Register (PCSR)

Address(es): PDC.PCSR 8784 1008h, PDC_NS.PCSR 0784 1008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	HERF	VERF	UDRF	OVRF	FEF	FEMPF	FBSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	FBSY	Frame Busy Flag	0: Operations for reception are stopped. 1: Operations for reception are ongoing.	R
b1	FEMPF	FIFO Empty Flag	0: FIFO is not empty. 1: FIFO is empty.	R
b2	FEF	Frame End Flag	0: Frame end has not been generated. 1: Frame end has been generated.	R/(W) *1
b3	OVRF	Overrun Flag	0: FIFO overrun has not been generated. 1: FIFO overrun has been generated.	R/(W) *1
b4	UDRF	Underrun Flag	0: Underrun has not been generated. 1: Underrun has been generated.	R/(W) *1
b5	VERF	Vertical Line Number Setting Error Flag	0: Vertical line number setting error has not been generated. 1: Vertical line number setting error has been generated.	R/(W) *1
b6	HERF	Horizontal Byte Number Setting Error Flag	0: Horizontal byte number setting error has not been generated. 1: Horizontal byte number setting error has been generated.	R/(W) *1
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. 0 can only be written to these bits to clear these flags after they have been read out as 1.

FBSY Flag (Frame Busy Flag)

This flag indicates the state of PDC operations.

[Setting Condition]

- Detection of the effective edge of the VSYNC signal after the enabling of operations for reception.

[Clearing Conditions]

- Reception of one-frame of data according to the settings of the VCR and HCR registers being completed.*1
- When an error (overrun, underrun, vertical line number setting error, or horizontal byte number setting error) occurs
- When the PCCR1.PCE bit is 0

Note 1. This flag is 0 during operations for continued reception.

FEMPF Flag (FIFO Empty Flag)

This flag indicates the state of the FIFO.

This flag indicates the state of the FIFO at the time when a vertical line number setting error or a horizontal byte number setting error occurs and is 0 following an overrun and undefined following an underrun.

[Setting Conditions]*1

- Reading of the PCDR register while the FIFO is empty.

- Detection of an effective edge of the VSYNC signal.
- The PDC being reset.

[Clearing Condition]*1

- Storage of the data captured in the FIFO.

Note 1. This flag is undefined following an underrun, and is set to either 0 or 1.

FEF Flag (Frame End Flag)

This flag indicates the end of a frame.

[Setting Condition]

- Reception of one frame of data in accord with the settings of the VCR and HCR registers.*1

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

Note 1. In the case of operations for continued reception, this flag will be 1 after their completion.

OVRF Flag (Overrun Flag)

This flag indicates an overrun.

[Setting Condition]

- Data for reception arriving while the FIFO is full.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

UDRF Flag (Underrun Flag)

This flag indicates an underrun.

[Setting Condition]

- Reading of the PCDR register while the FIFO is empty.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

VERF Flag (Vertical Line Number Setting Error Flag)

This flag indicates an error in the setting for the number of lines.

[Setting Condition]

- The VSYNC signal is negated because fewer lines have been captured than the value in the VCR register.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

HERF Flag (Horizontal Byte Number Setting Error Flag)

This flag indicates an error in the setting for the number of bytes in a line.

[Setting Condition]

- The HSYNC signal is negated because fewer bytes in a line have been captured than the value in the HCR register.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

53.2.4 PDC Pin Monitor Register (PCMONR)

Address(es): PDC.PCMONR 8784 100Ch, PDC_NS.PCMONR 0784 100Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSYNC	VSYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VSYNC	VSYNC Signal Status Flag	0: VSYNC signal is at the low level. 1: VSYNC signal is at the high level.	R
b1	HSYNC	HSYNC Signal Status Flag	0: HSYNC signal is at the low level. 1: HSYNC signal is at the high level.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

VSYNC Flag (VSYNC Signal Status Flag)

This flag indicates the state of the VSYNC signal.

HSYNC Flag (HSYNC Signal Status Flag)

This flag indicates the state of the HSYNC signal.

53.2.5 PDC Receive Data Register (PCDR)

Address(es): PDC.PCDR 8784 1010h, PDC_NS.PCDR 0784 1010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The PDC includes a 32-bit, 22-stage FIFO buffer for the storage of captured data. The FIFO buffer is mapped to the 4-byte PCDR register, and captured data are read from this register in 4-byte units. The receive data ready flag is set for every 32 bytes of received data, and this also leads to a receive data ready interrupt if the DFIE bit in the PCCR0 register is set to 1. When a receive data ready interrupt is generated, read the PCDR register eight times. Figure 53.2 shows a schematic view of the PCDR register.

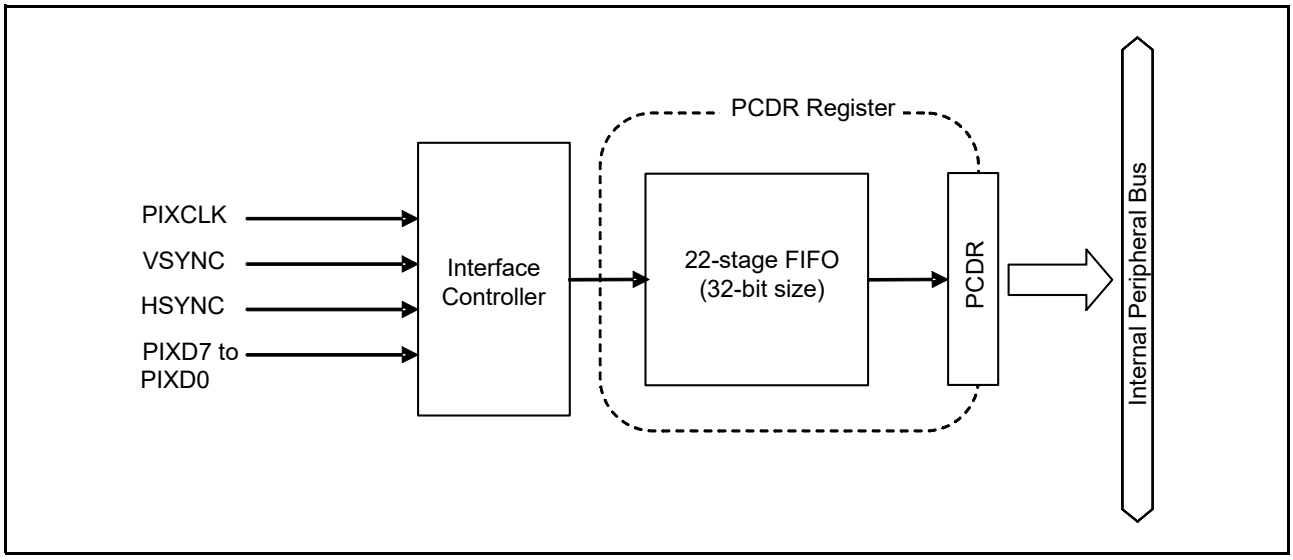


Figure 53.2 Schematic View of the PCDR Register

For the format of the captured data, either big or little endian can be selected by the EDS bit of the PCCR0 register. Figure 53.3 shows the arrangements of data according to the endian formats.

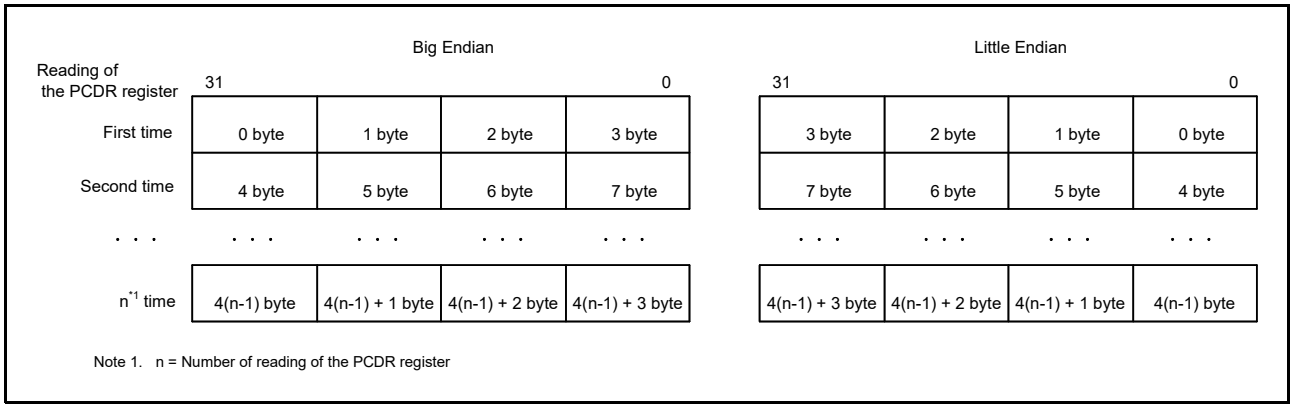


Figure 53.3 Endian Formats

53.2.6 Vertical Capture Register (VCR)

Address(es): PDC.VCR 8784 1014h, PDC_NS.VCR 0784 1014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	VSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	VST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	VST[11:0]	Vertical Capture Start Line Position	Number of the line where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	VSZ[11:0]	Vertical Capture Size	Number of lines to be captured.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the VCR register setting and the captured range, refer to section 53.3.3, Settings of the VCR and HCR Registers and the Captured Range.
The VCR register should be set while the PCE bit in the PCCR1 register is 0.

VST[11:0] Bits (Vertical Capture Start Line Position)

These bits set the number of the line where capture is to start.
To set the first line, these bits should be 000h; to set the 4095th line, they should be FFEh.
The setting of the VST[11:0] bits should be within the range from 000h to FFEh and, in combination with that of the VSZ[11:0] bits, satisfy the following relation:
Setting range of the VST[11:0] bits: $1 \leq VST[11:0] + VSZ[11:0] \leq FFFh$.

VSZ[11:0] Bits (Vertical Capture Size)

These bits set the number of lines to be captured.
To set one line, these bits should be 001h; to set 4095 lines, they should be FFFh.
The setting of the VSZ[11:0] bits should be within the range from 001h to FFFh and, in combination with that of the VST[11:0] bits, satisfy the following relation:
Setting range of the VSZ[11:0] bits: $1 \leq VST[11:0] + VSZ[11:0] \leq FFFh$.

53.2.7 Horizontal Capture Register (HCR)

Address(es): PDC.HCR 8784 1018h, PDC_NS.HCR 0784 1018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	HSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	HST[11:0]	Horizontal Capture Start Byte Position	Horizontal position in bytes where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	HSZ[11:0]	Horizontal Capture Size	Number of bytes to be captured horizontally.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the HCR register setting and the captured range, refer to section 53.3.3, Settings of the VCR and HCR Registers and the Captured Range.
The HCR register should be set while the PCE bit in the PCCR1 register is 0.

HST[11:0] Bits (Horizontal Capture Start Byte Position)

These bits set the horizontal position in bytes where capture is to start.
To set the first byte, these bits should be 000h; to set the 4092th byte, they should be FFBh.
The setting of the HST[11:0] bits should be within the range from 000h to FFBh and, in combination with that of the HSZ[11:0] bits, satisfy the following relation:
Setting range of the HST[11:0] bits: $4 \leq HST[11:0] + HSZ[11:0] \leq FFFh$.

HSZ[11:0] Bits (Horizontal Capture Size)

These bits set the number of bytes to be captured per line.
To set four bytes, these bits should be 004h; to set 4095 bytes, they should be FFFh.
The setting of the HSZ[11:0] bits should be within the range from 004h to FFFh and, in combination with that of the HST[11:0] bits, satisfy the following relation:
Setting range of the HSZ[11:0] bits: $4 \leq HST[11:0] + HSZ[11:0] \leq FFFh$.

53.3 Operation

53.3.1 Transfer Formats

The PDC supports the four transfer formats shown in Figure 53.4 to Figure 53.7. The format is determined by the setting of the VPS and HPS bits in the PCCR0 register.

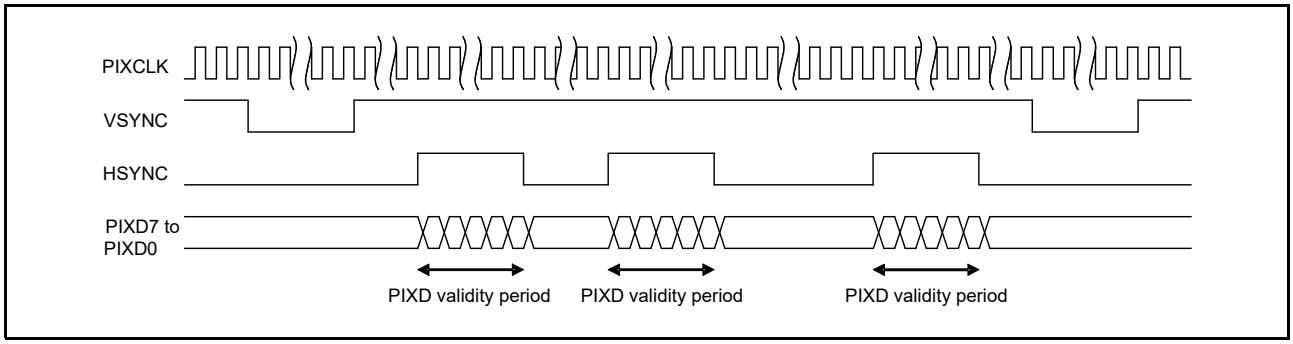


Figure 53.4 PDC Transfer Format (for VPS=0, HPS=0)

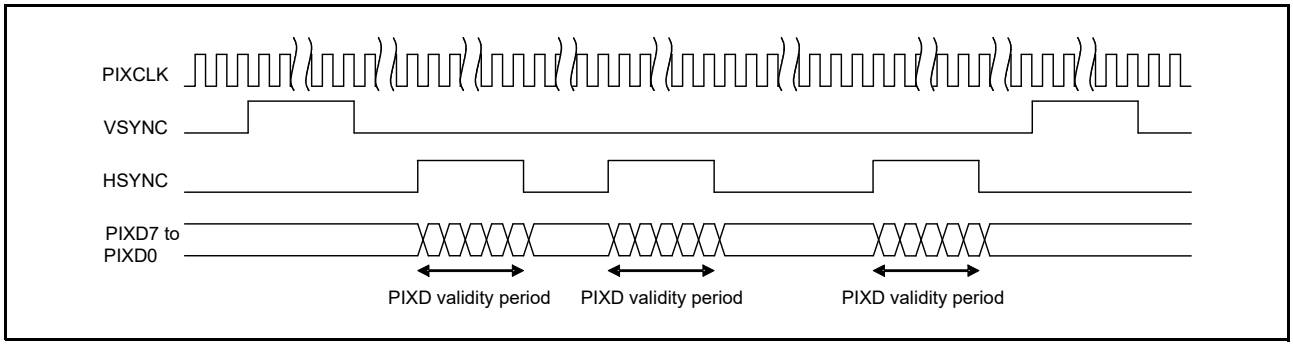


Figure 53.5 PDC Transfer Format (for VPS=1, HPS=0)

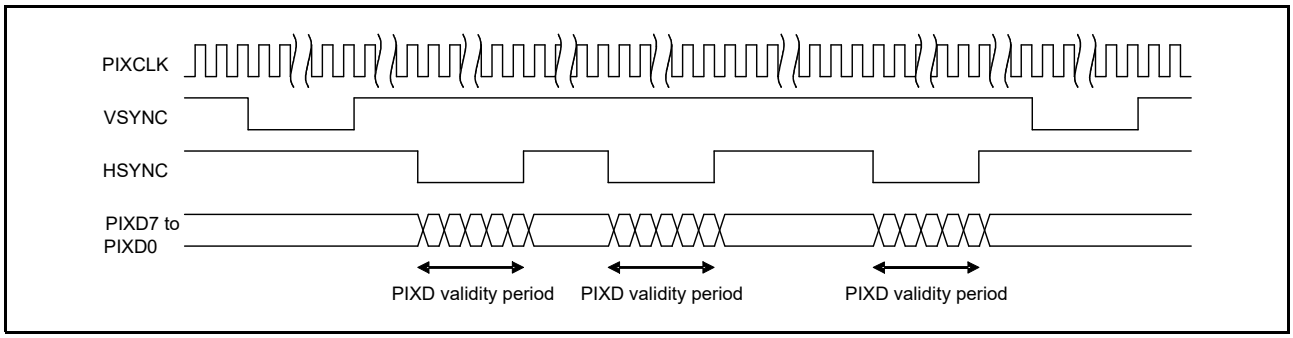


Figure 53.6 PDC Transfer Format (for VPS=0, HPS=1)

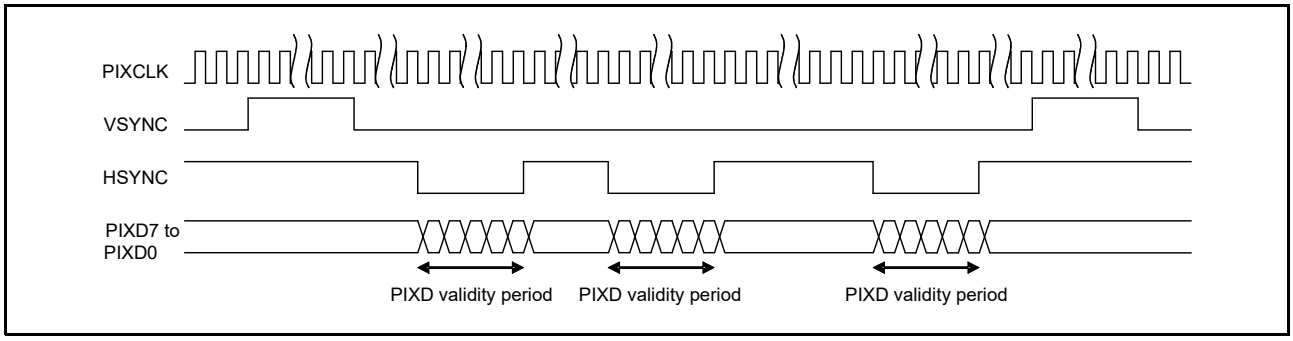


Figure 53.7 PDC Transfer Format (for VPS=1, HPS=1)

53.3.2 Transfer Timing

Figure 53.8 and Table 53.3 show the timing of transfer by the PDC.

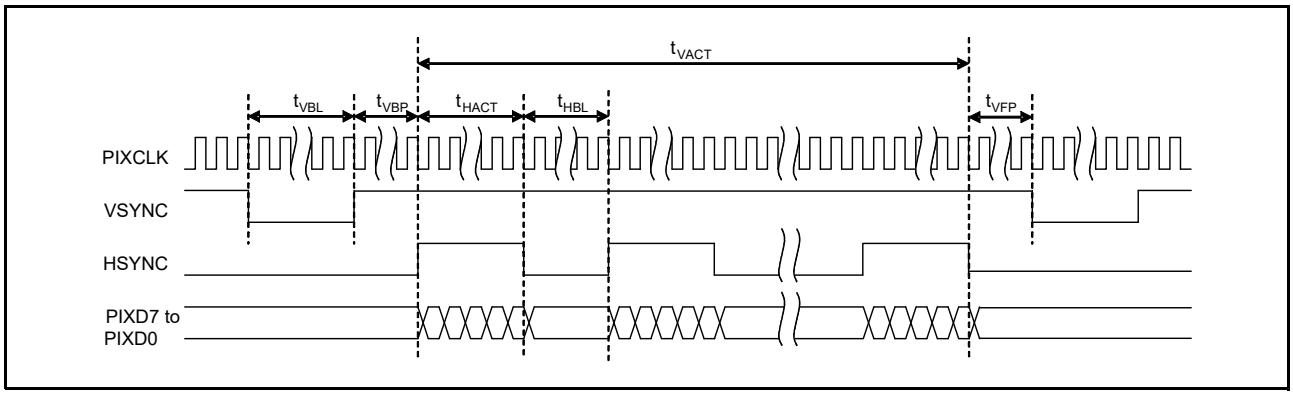


Figure 53.8 PDC Transfer Timing

Table 53.3 PDC Transfer Timing

Item	Symbol	Min*1	Max	Unit
Vertical Blanking Period	t_{VBL}	128	—	PIXCLK
Vertical Backporch	t_{VBP}	10	—	PIXCLK
Horizontal Valid Period	t_{HACT}	4	4095	PIXCLK
Horizontal Blanking Period	t_{HBL}	128	—	PIXCLK
Vertical Frontporch	t_{VFP}	10	—	PIXCLK
Vertical Valid Period	t_{VACT}	1	4095	Line

Note 1. The minimum values are the lowest of which this MCU circuit is capable. They are not values which guarantee the avoidance of overruns, vertical line number setting errors, or horizontal byte number setting errors.

53.3.3 Settings of the VCR and HCR Registers and the Captured Range

Figure 53.9 and Figure 53.10 show the relationship between the settings of the VCR and HCR registers and the captured range.

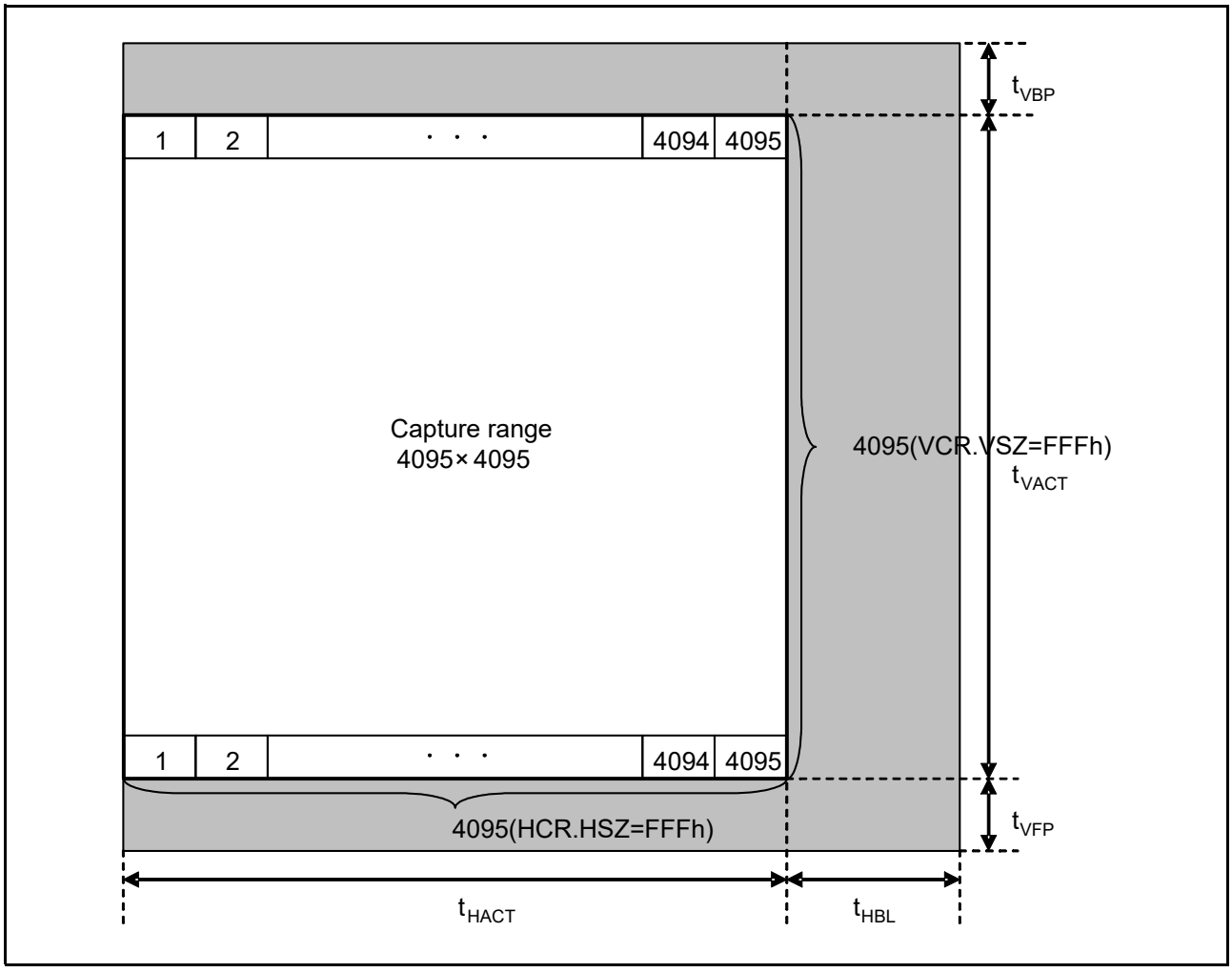


Figure 53.9 Settings of the VCR and HCR Registers and the Captured Range (VCR = 0FFF 0000h, HCR = 0FFF 0000h)

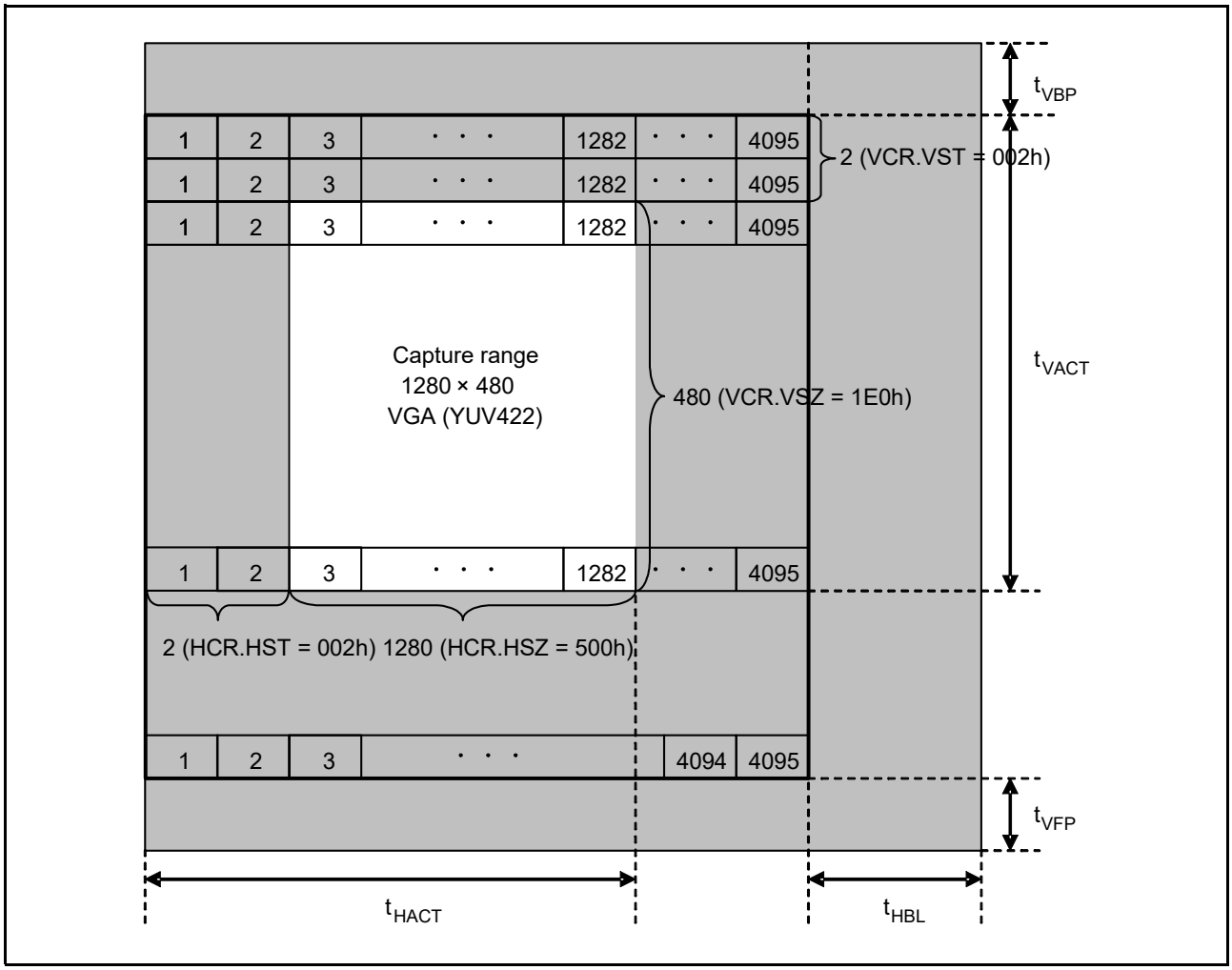


Figure 53.10 Settings of the VCR and HCR Registers and the Captured Range (VCR = 01E0 0002h, HCR = 0500 0002h)

53.3.4 Operations for Reception

Figure 53.11 shows an example of operations for reception when receive data ready interrupts (to start up the DTC or DMAC) and frame end interrupts are in use.

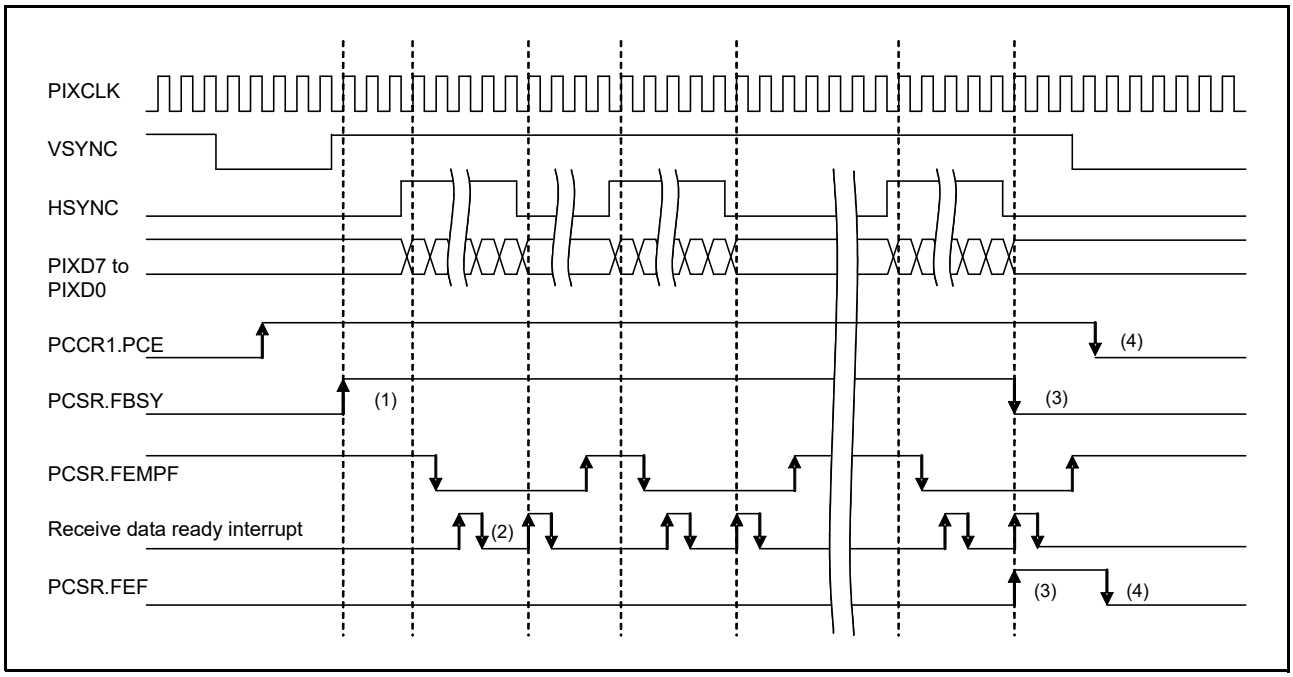


Figure 53.11 Example of Operations for Reception

The actual operations at the times indicated by (1), (2), (3), and (4) in Figure 53.11 are described below.

(1) When an effective edge of the VSYNC signal is detected after the PCE bit in the PCCR1 register is set to 1, the FEMPF flag in the PCSR register is set to 1 and the FIFO is initialized. Concurrently, the FBSY flag in the PCSR register is set to 1 and operations for reception start.

(2) When data within the range for capture set in the VCR and HCR registers is received, the data are stored in the FIFO. The PDC generates a receive data ready interrupt every time it receives 32 bytes of data, and the interrupt starts transfer of the captured data by the DTC or DMAC to the on-chip RAM or an external address space. The FIFO is likely to overrun if reading of the PCDR register takes more time than reception of the data. Check the OVRF flag in the PCSR register to verify an overrun.

(3) After reception of the last byte of data is completed, the FBSY flag in the PCSR register is cleared to 0 and the FEF flag in the PCSR register is set to 1 so that a receive data ready interrupt and frame end interrupt are generated.

(4) The FEMPF flag in the PCSR register is polled by the frame end interrupt, after which the completion of data transfer by the DTC or DMAC should be verified. After the PCE bit in the PCCR1 register is cleared to 0, the FEF flag in the PCSR register is also cleared to 0, and the reception of one frame of data is complete.
If an underrun occurs while the PCSR.FEMPF flag is being polled, setting of the PCSR.FEMPF flag may not proceed. Accordingly, also check the PCSR.UDRF flag during polling, and if an underrun occurs, run appropriate error processing.

If the FEF flag in the PCSR register has been set to 1 before the PCE bit in the PCCR1 register is set to 1, effective edges of the VSYNC signal are not detected and operations for reception will not be started. Clear the FEF flag in the PCSR register to 0 to start operations for data reception.

53.3.5 Operation during the Horizontal Blanking Period

If the horizontal blanking period begins but the number of the received data bytes has not reached 32 bytes since the previous receive data ready, the count of the number of received data bytes is retained and carried over to the next horizontal valid period. Figure 53.12 shows an example of operation during the horizontal blanking period.

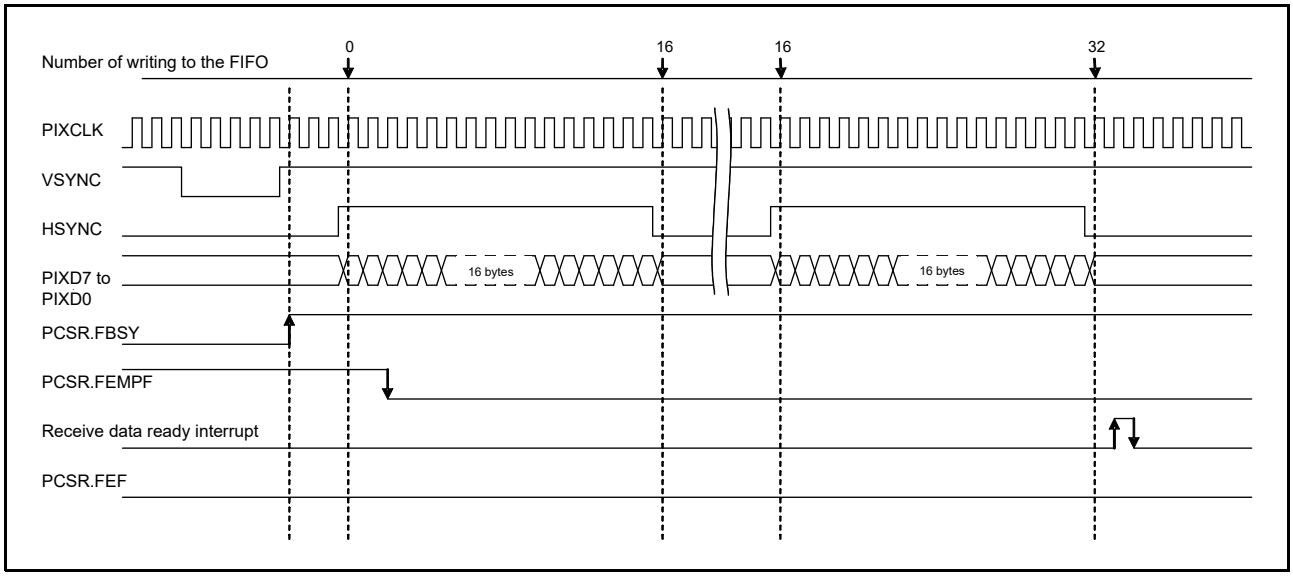


Figure 53.12 Example of Operation during the Horizontal Blanking Period

53.3.6 Operations for Continued Reception at the Time of Frame End

When the last of the data have been received but the number of bytes of data received since the previous receive data ready has not reached 32, the PDC continues to receive data (hereinafter referred to as “continued reception”) until the number reaches 32. After that, it generates a receive data ready and a frame end. The PIXCLK should be input during operations for continued reception. If the data stored in the FIFO are read out during operations for continued reception, the values read are undefined. Figure 53.13 shows an example of operations at the time of frame end.

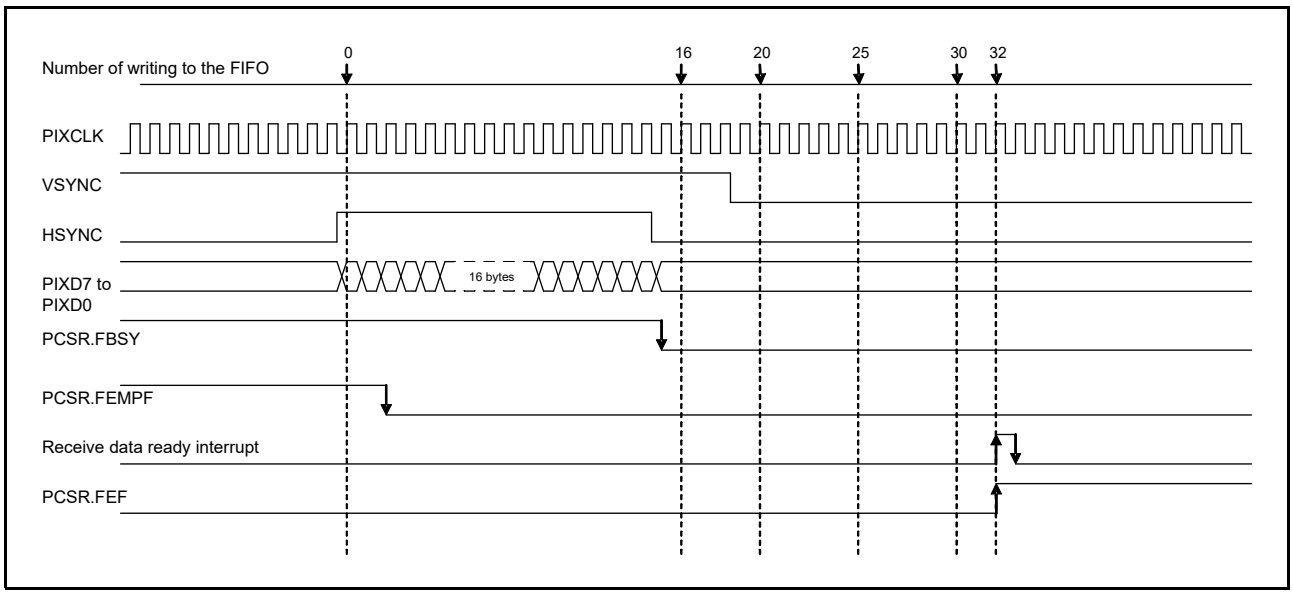


Figure 53.13 Example of Continued Reception at the Time of Frame End

53.3.7 Error Detection

The PDC has functions for error detection so that the software can apply control in response to errors during operations for reception. Table 53.4 summarizes the conditions under which each type of error is detected and the interrupt flags set in response.

Table 53.4 Error Detection

Error Factor	Conditions of Error Detection	Interrupt Flag	Example of Operation
Overrun	Data for reception arriving while the FIFO is full.*1	PCSR.OVRF	Figure 53.14
Underrun	The PCDR register being read while the FIFO is empty.	PCSR.UDRF	Figure 53.15
Vertical line number setting error	Negation of the VSYNC signal when the number of captured lines is less than the value set in the VCR register.	PCSR.VERF	Figure 53.16
Horizontal byte number setting error	Negation of the HSYNC signal when the number of bytes captured in a line is less than the value set in the HCR register.	PCSR.HERF	Figure 53.17

Note 1. This includes data reception during operations for continued reception.

When an error is detected, the PDC sets the corresponding interrupt flag to 1 to stop operations for reception. While the interrupt flag is set to 1, the PDC does not detect effective edges of the VSYNC signal and does not start operations for reception. Clear all error source interrupt flags to 0 to start operations for reception.
In addition, when an error occurs, data stored in the FIFO is disabled.

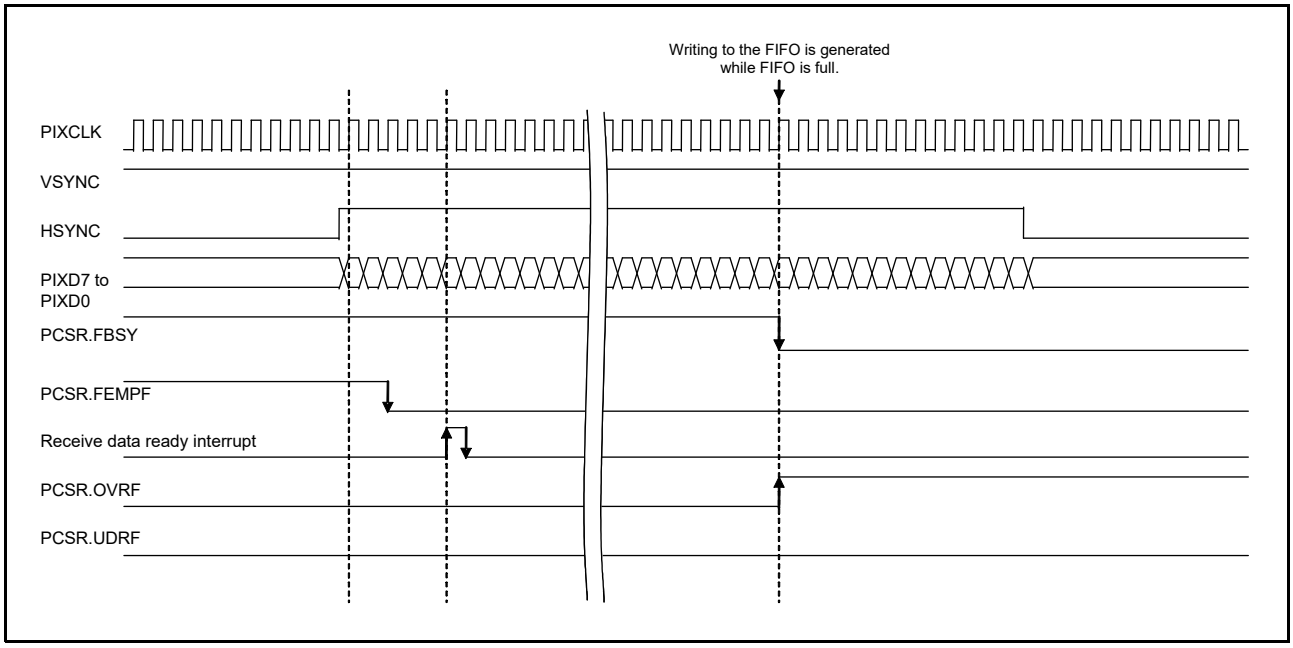


Figure 53.14 Overrun Detected

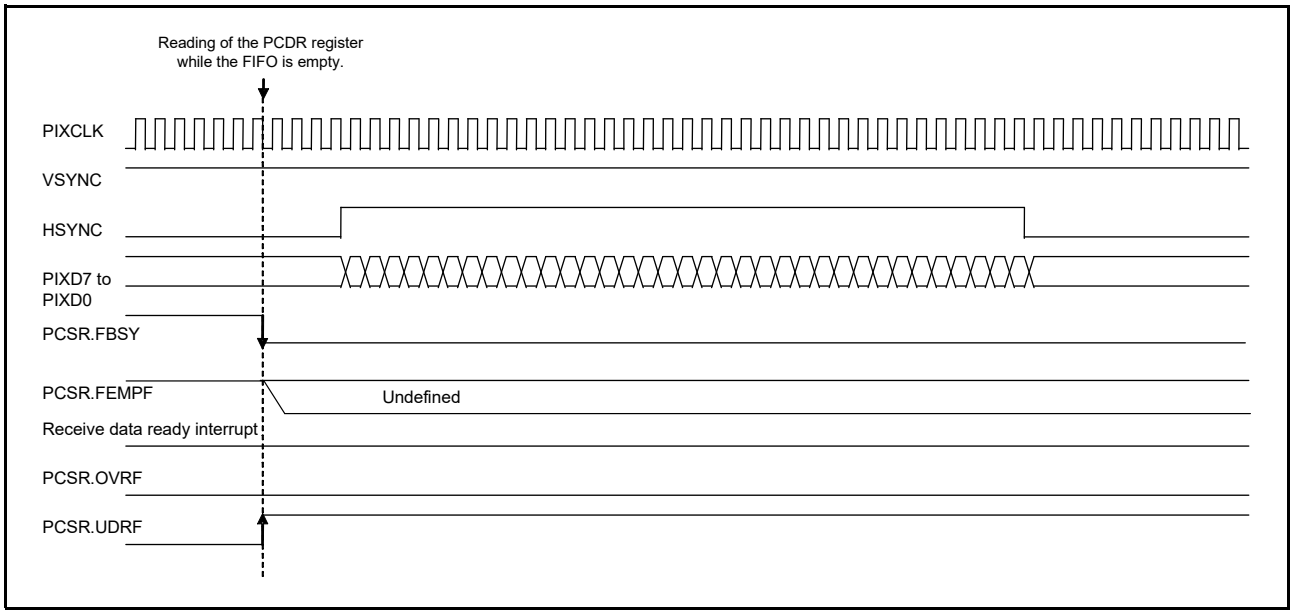


Figure 53.15 Underrun Detected

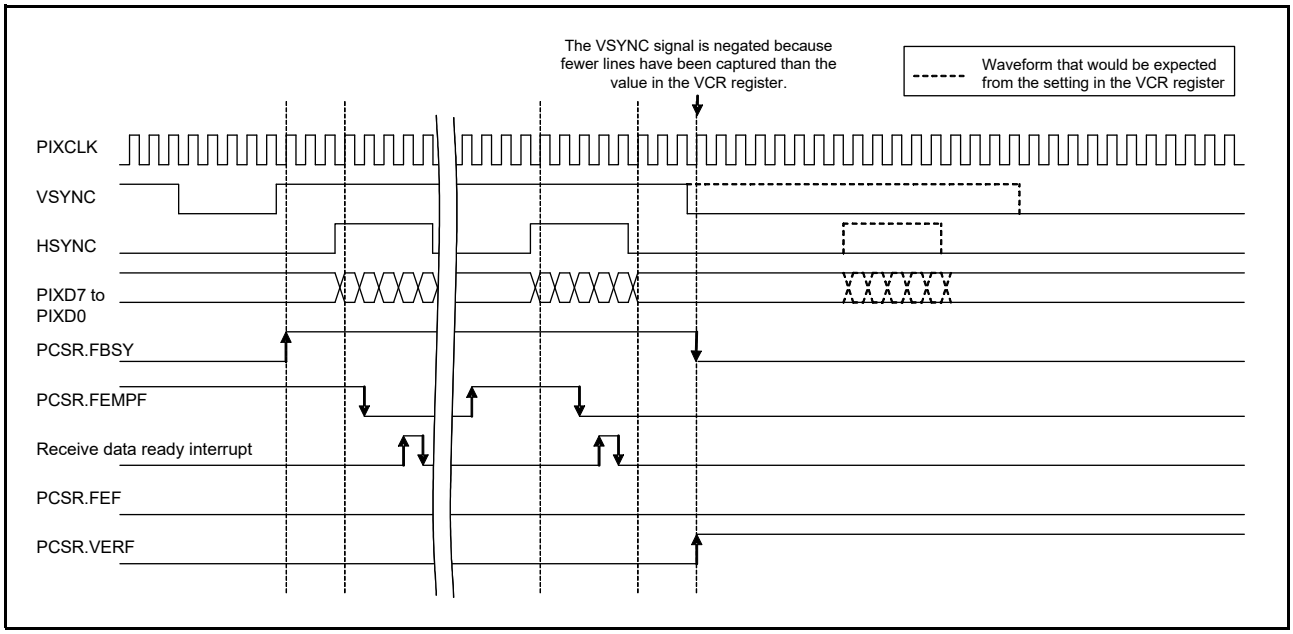


Figure 53.16 Vertical Line Number Setting Error Detected

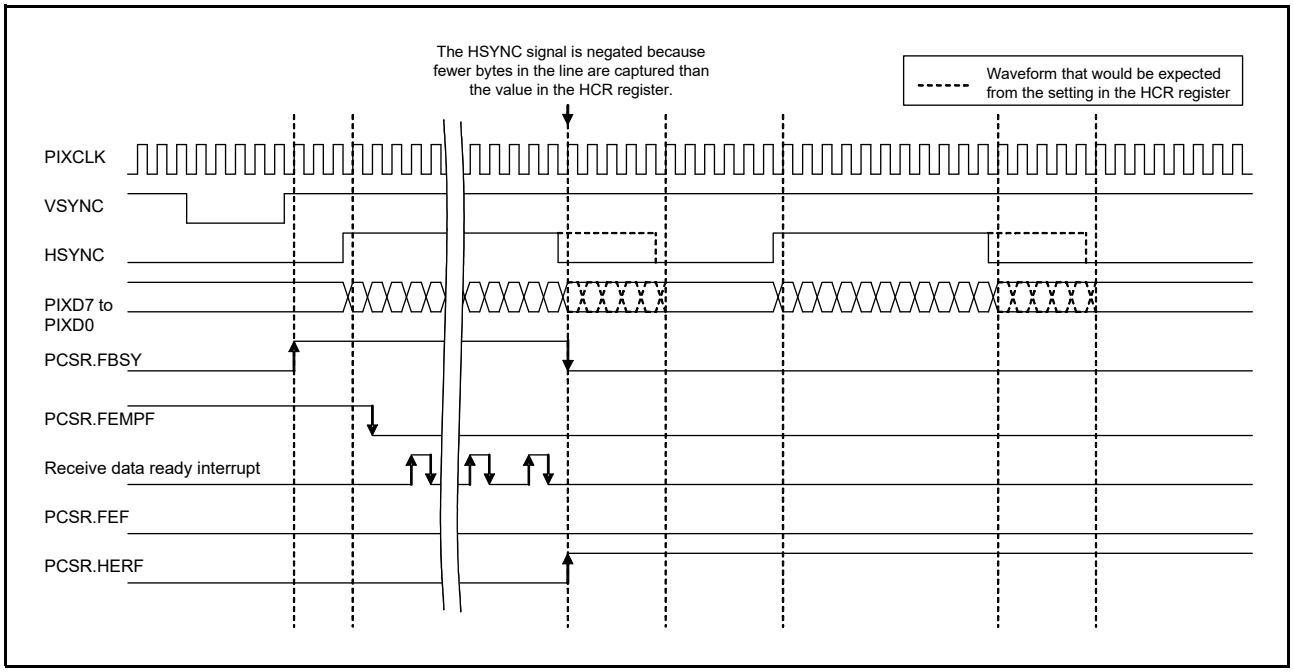


Figure 53.17 Horizontal Byte Number Setting Error Detected

53.3.8 Initial Settings

Figure 53.18 is a sample flowchart for initial settings. For a description of how to set up the input and output ports and the interrupt controller, refer to the descriptions given in the sections on the relevant blocks.

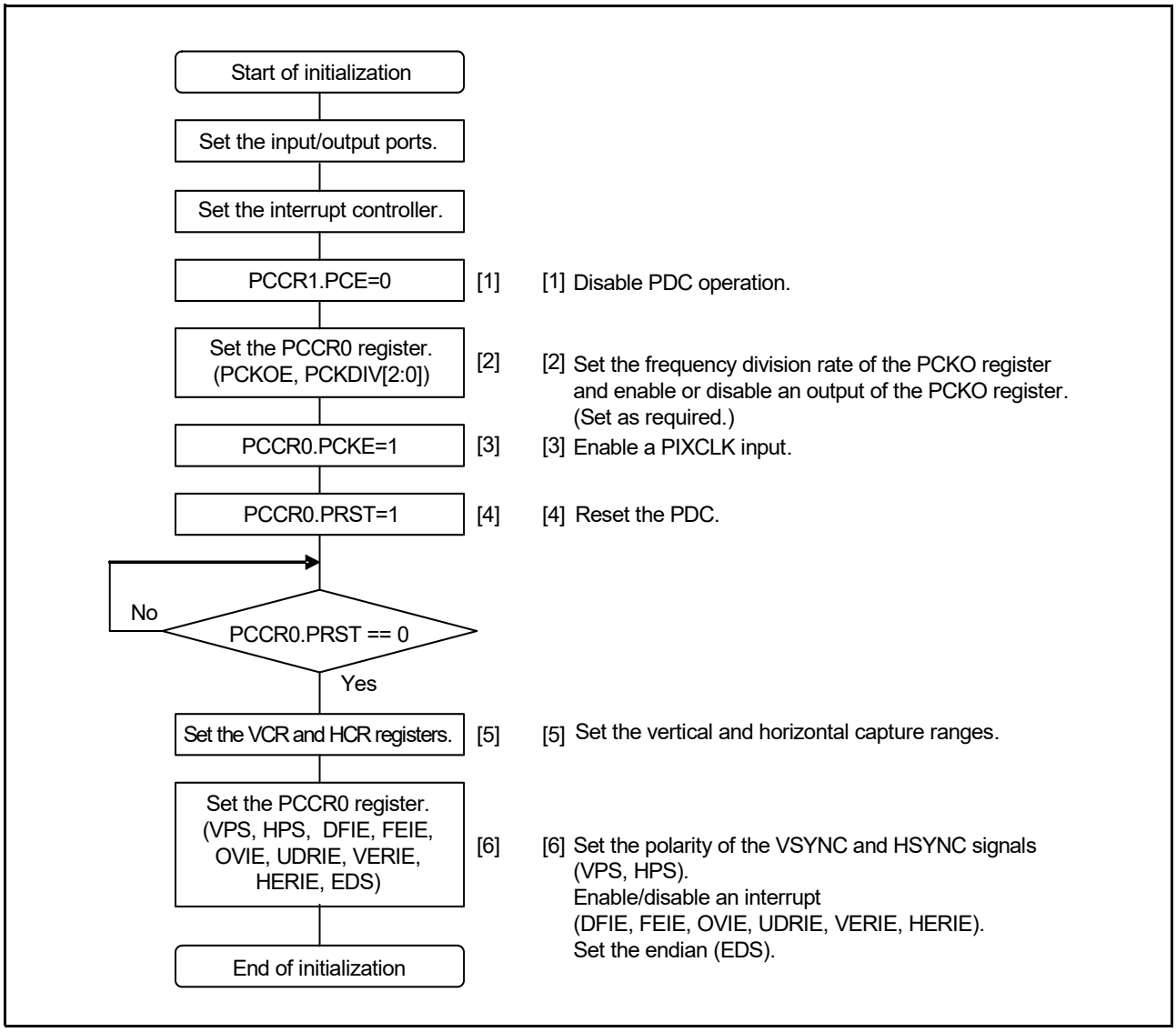
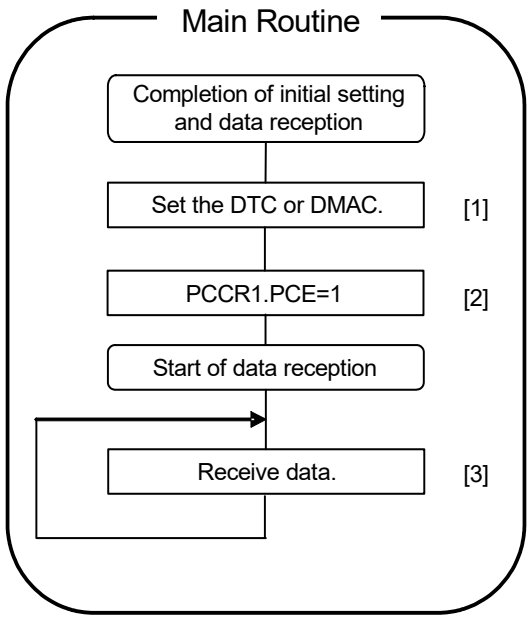
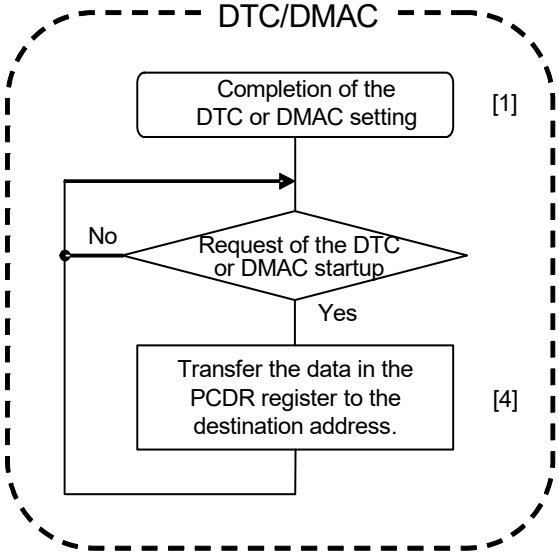


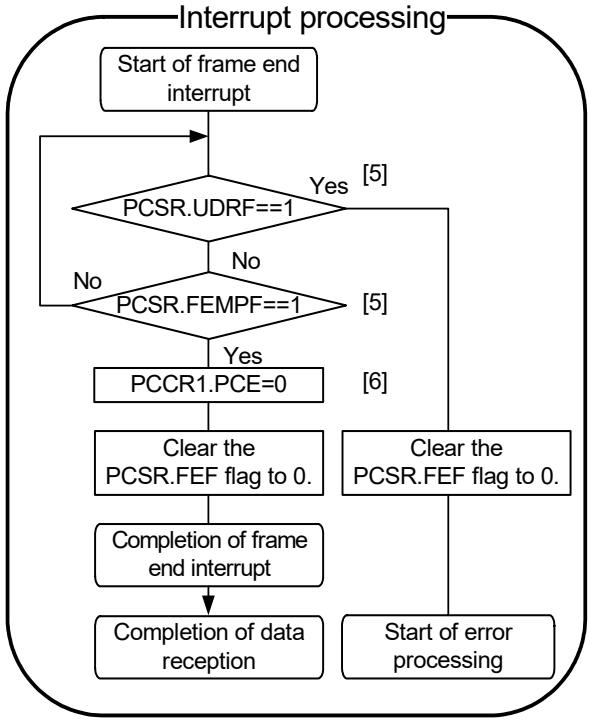
Figure 53.18 Sample Flowchart for Initial Setting of the PDC

53.3.9 Flows of Operations

Figure 53.19 shows an example of the flows of operations when the receive data ready interrupt (to start up the DTC or DMAC) and frame end interrupt are in use. For a description of how to set up the DTC or DMAC, refer to section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCA).







[1] Setting of the DTC or DMAC:
Set the DTC or DMAC according to the size of the captured image.

[2] Enabling of PDC operation:
Set the PCE bit in the PCCR1 register to 1 to enable PDC operation.

[3] Data reception:
Receive the data from the external I/O and wait for generation of a receive data ready interrupt (DTC/DMAC startup) or a frame end interrupt.

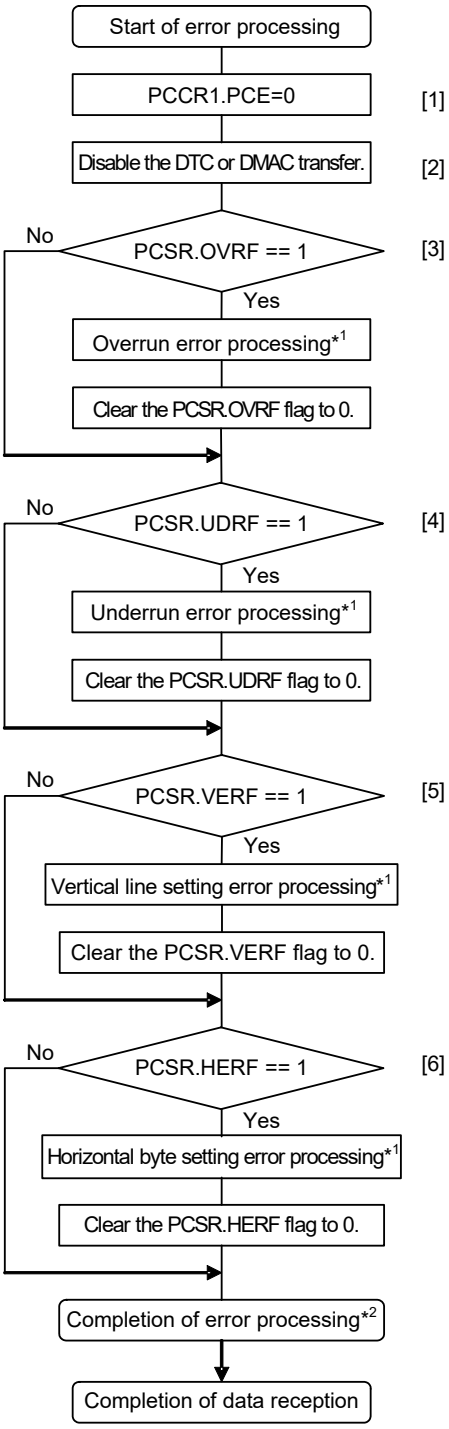
[4] Transfer by the DTC or DMAC:
Transfer the data in the PCDR register to the destination address in response to a receive data ready interrupt (DTC/DMAC startup).
Read the PCDR register in longword access and transfer 32 bytes of data per receive data ready interrupt.

[5] Frame end interrupt:
Read the FEMPF flag in the PCSR register and verify completion of the DTC or DMAC transfer.
If an underrun occurs during DTC or DMAC transfer, setting of the FEMPF flag may not proceed. When an underrun occurs, clear the FEF flag to 0 and branch to the appropriate error processing routine.

[6] Completion of data reception:
After completion of the transfer, clear the PCE bit in the PCCR1 register to 0, and then clear the FEF flag in the PCSR register to 0. This completes handling of the frame end interrupt and data reception.

Figure 53.19 Example of Flows of Operations

Figure 53.20 shows an example of the flow of operations in response to an error interrupt.



[1] Disabling of PDC operation:
Clear the PCE bit in the PCCR1 register to 0 to disable PDC operation.

[2] Disabling of the DTC or DMAC transfer:
Disable the transfer of the DTC or DMAC.

[3] Checking the overrun flag:
Read the PCSR register to check the state of the OVRF flag. If the flag is set, processing to handle the overrun error proceeds and the flag is then cleared.

[4] Checking the underrun flag:
Read the PCSR register to check the state of the UDRF flag. If the flag is set, processing to handle the underrun error proceeds and the flag is then cleared.

[5] Checking the vertical line setting error flag:
Read the PCSR register to check the state of the VERF flag. If the flag is set, processing to handle the vertical line setting error proceeds and the flag is then cleared.

[6] Checking the horizontal byte setting error flag:
Read the PCSR register to check the status of the HERF flag. If the flag is set, processing to handle the horizontal byte setting error proceeds and the flag is then cleared.

Note 1. Processing for each type of error is defined by the user, so the user needs to prepare error processing for each.

Note 2. If the CPU, DTC, or DMAC reads the PCDR register after an error interrupt has occurred in response to an interrupt source other than an underrun, an underrun may occur. If an underrun occurs on completion of error processing, run the error processing again and clear the UDRF flag.
Since an underrun after error processing still requires detection, do not prohibit the underrun interrupt before the error processing.

Figure 53.20 Example of Error Processing Flow

53.3.10 Interrupt Source

The interrupt sources of the PDC are receive data ready, frame end, overrun, underrun, vertical line number setting error, and horizontal byte number setting error. The PDC can start up the DTC or DMAC in response to a receive data ready interrupt request for the transfer of data.

Table 53.5 summarizes the interrupt sources of the PDC. When an interrupt condition listed in Table 53.5 is satisfied, the corresponding interrupt is generated. In the case of receive data ready, the interrupt source flag is cleared by reading of the PCDR register. For the interrupt source flag for frame end, the FEF flag in the PCSR register is cleared. For an overrun, underrun, vertical line number setting error, or a horizontal byte number setting error, the flags must be checked to identify the error source flag because their interrupt vectors are allocated to the same address by the PCERI. Once identified, the corresponding error interrupt source flag (the OVRF, UDRF, VERF, or HERF flag) in the PCSR register should be cleared.

When the DTC or DMAC is to handle data transfer, whichever is selected should be set up first. After enabling the block to handle transfer, set up the PDC. For descriptions of how to set up the DTC and DMAC, refer to section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCA).

In addition, for receive data ready, if the ICU.IRn.IR flag is 1, the interrupt request signal is retained internally but an interrupt request is not output to the ICU, even if the interrupt condition is satisfied. In this case, however, only one interrupt request signal can be retained internally. If the ICU.IRn.IR flag is 0, the retained interrupt request signal is output to the ICU. After completion of output, the request flag is automatically cleared. An interrupt request signal retained internally can also be cleared by setting the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to 0.

When the receive data ready interrupt is disabled from the enabled state, set the ICU.IRn.IR flag to 0 according to the following procedure.

1. Set the IERn.IENj bit to the value that disables interrupt requests.
2. Set the receive data ready interrupt enable bit (the DFIE bit) to “disabled” and read out the register after writing to verify that the bit has the new value.
3. Set the ICU.IRn.IR flag to 0.

Table 53.5 PDC Interrupt Source

Interrupt Source	Abbre -viation	Interrupt Conditions	DTC/DMAC Startup
Receive Data Ready	PCDFI	Receive data ready occurring while the DFIE bit in the PCCR0 register is 1.	Possible
Frame End	PCFEI	Frame end occurring while the FEIE bit in the PCCR0 register is 1.	Impossible
Errors	PCERI	An overrun occurring while the OVIE bit in the PCCR0 register is 1. An underrun occurring while the UDRIE bit in the PCCR0 register is 1. A vertical line number setting error occurring while the VERIE bit in the PCCR0 register is 1. A horizontal byte number setting error occurring while the HERIE bit in the PCCR0 register is 1.	Impossible

53.3.11 Reset State

Resetting of the PDC is divided into two types, PDC reset and other reset. Other resets include RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, voltage-monitoring 4 reset, voltage-monitoring 5 reset, core voltage monitoring reset, temperature monitoring reset, independent watchdog timer rese, watchdog timer reset, bus error reset, common memory error reset, local memory error reset, deep software standby reset, and software reset. Table 53.6 lists the range and states following the two types of reset.

Table 53.6 Reset State

	PDC Reset	Other reset
PCCR0	Retained	Reset
PCCR1	Retained	Reset
PCSR	Reset	Reset
PCMONR	Retained	Reset
PCDR	Retained	Reset
VCR	Retained	Reset
HCR	Retained	Reset

53.4 Usage Notes

53.4.1 Setting of the Module Stop Function

The module stop control register C (MSTPCRC) enables or disables operation of the PDC. Operation of the PDC is stopped when the value is that following a reset. The register becomes accessible following release from the module stop state. For details, refer to section 11, Low Power Consumption.

53.4.2 Notes on the Power Consumption Reduction Function

The following steps are required in advance of using the power consumption reduction function to reduce power consumption by the PDC.

- Set the PCE bit in the PCCR1 register to 0 to disable operations for reception.
- Set the PCKE bit in the PCCR0 register to 0 to disable input through the PIXCLK pin.
- After that, clear the flags in the PCSR register.

If the setting of the PCKOE bit in the PCCR0 register is 1, the following steps are required in advance of using the power consumption reduction function.

- Set the PCKOE bit to 0 to stop output of the PCKO signal.
- Disable input through the PIXCLK pin in the same way as stated above.
- After that, clear the flags in the PCSR register.

On return from the low power consumption state, set the PCKE bit to 1, and after enabling input through the PIXCLK pin, use the PRST bit to initialize the PDC.

53.4.3 Notes on Error Interrupts

When an error interrupt occurs, there is a possibility that the DTC or DMAC might be transmitting parallel data depending on their operation state. Therefore, transmitting executed by the DTC or DMAC should be prohibited after the PDC operation is prohibited at the top of the error interrupt processing routine (PCCR1.PCE = 0).

53.4.4 Notes on Use of the DTC

When the DTC is used with the receive data ready interrupt, set the DISEL bit in the MRB register to 0 and the SZ bit in the MRA register to 10b.
The maximum number of blocks the DTC can transfer in block transfer mode is 65536. If 32 bytes are transferred per block transfer, this represents a total of up to 2097152 bytes. If more data are to be transferred, set up the DTC again during the horizontal blanking period.

53.4.5 Notes on Use of the DMAC

When the DMAC is used with the receive data ready interrupt, set the DISEL bit in the DMCSL register to 0, set the SZ bit in the DMTMD register to 10b, and set the IPRn.IPR[3:0] bit responding to an ICU receive data ready interrupt to level 0 (the interrupt is prohibited).
The maximum number of blocks the DMAC can transfer in block transfer mode is 65536. If 32 bytes are transferred per block transfer, this represents a total of up to 2097152 bytes. If more data are to be transferred, set up the DMAC again during the horizontal blanking period.

53.4.6 Notes on Start of Transfer

When transfer starts while the ICU.IRn.IR flag is 1, an interrupt request signal is internally retained after the start of transfer. This may cause the ICU.IRn.IR flag to behave in an unexpected way.
In such cases, therefore, clear the interrupt request flag by following the procedures below before enabling operations (i.e. before setting the PCE bit in the PCCR1 to 1).

(1) Ensure that transfer is stopped (the PCE bit in the PCCR1 is 0).
(2) Set the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to 0.
(3) Read out the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to ensure that it is 0.
(4) Set the ICU.IRn.IR flag to 0.

54. Boundary Scan

This MCU has boundary scan function.
The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

54.1 Overview

Table 54.1 lists the specifications of boundary scan.
Figure 54.1 shows a block diagram of the boundary scan function.

Table 54.1 Specifications of Boundary Scan

Item	Description
Prerequisite for execution	Run the boundary scan while the signal on the RES# pin is at the low level.
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

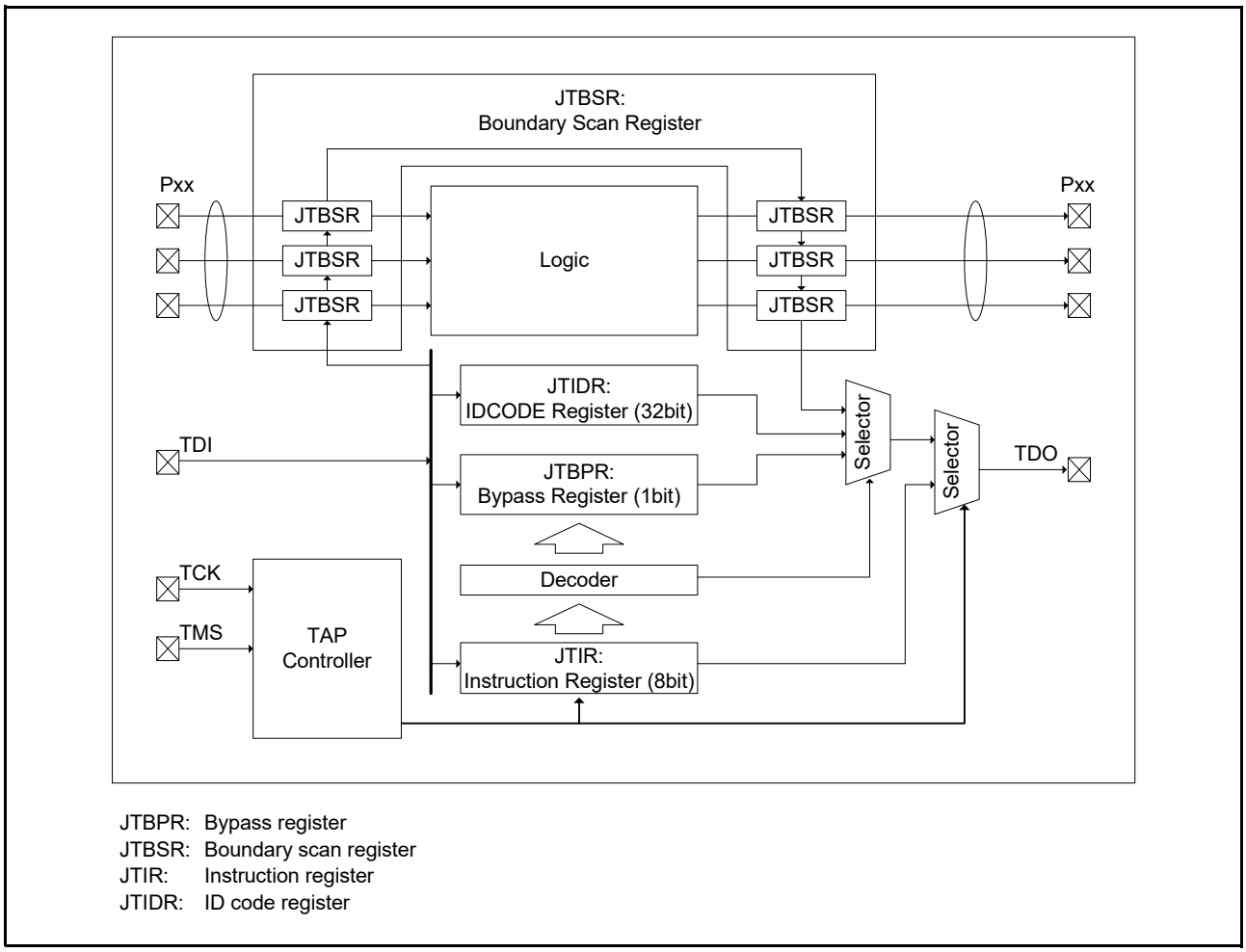


Figure 54.1 Block Diagram of JTAG

Table 54.2 lists the I/O pins used in the boundary scan function.

Table 54.2 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50 percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

54.2 Register Descriptions

Table 54.3 lists the boundary scan registers.

Table 54.3 List of Boundary Scan Registers

Register Name	Symbol	Value after Reset
Instruction register	JTIR	55h
ID code register	JTIDR	0867 C447h
Bypass register	JTBPR	Undefined
Boundary scan register	JTBSR	Undefined

Instructions can be input to the JTIR register via the TDI pin by serial transfer.
The JTBPR register, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
The JTBSR register, which is configured according to Table 54.6 to Table 54.11, is connected between the TDI and TDO pins when test data are being shifted in.
None of the registers is accessible from the CPU.
Table 54.4 lists the availability of serial transfer for the registers.

Table 54.4 Serial Transfer for the Registers

Register Name	Serial Input	Serial Output
Instruction register (JTIR)	Available	Available
ID code register (JTIDR)	Available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

54.2.1 Instruction Register (JTIR)



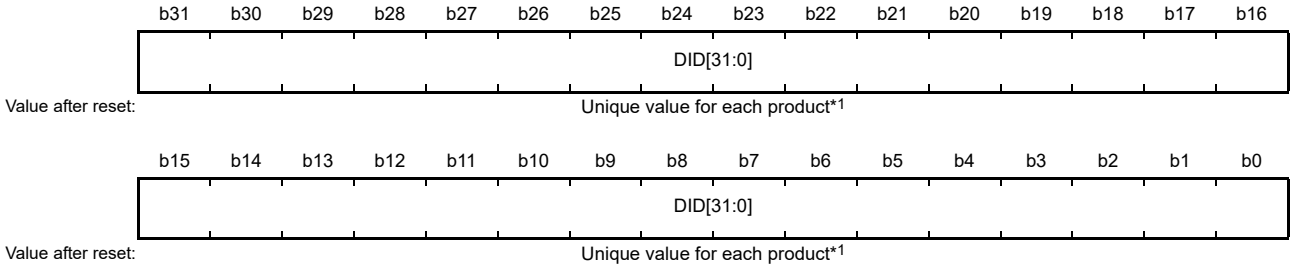
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as listed in Table 54.5.	—

Table 54.5 Command Configuration

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (initial value)
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin.
The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

54.2.2 ID Code Register (JTIDR)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Reserved	JTIDR is a register with the fixed value that indicates the device IDCODE.	—

Note 1. For the ID code for each product, refer to Table 54.3, List of Boundary Scan Registers.

JTIDR data is output from the TDO pin when the IDCODE instruction has been executed.

54.2.3 Bypass Register (JTBPR)

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode.
The JTBPR register cannot be read from or written to by the CPU.

54.2.4 Boundary Scan Register (JTBSR)

The JTBSR register is a shift register to control the external input and output pins of this MCU and is distributed across the pads.
The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply the JTBSR register in boundary-scan testing.
Table 54.6 to Table 54.11 list the correspondence between the JTBSR bits and the pins of this MCU.
The value after reset is undefined.

Table 54.6 Boundary Scan Register 320-Pin LFBGA (1/13)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
W20	P003	Output	567
		Output enable	566
		Input	565
V20	P002	Output	564
		Output enable	563
		Input	562
U20	P001	Output	561
		Output enable	560
		Input	559
T20	P000	Output	558
		Output enable	557
		Input	556
U19	P009	Output	555
		Output enable	554
		Input	553
U18	P008	Output	552
		Output enable	551
		Input	550
R15	P215	Output	549
		Output enable	548
		Input	547
R16	P214	Output	546
		Output enable	545
		Input	544
T18	P213	Output	543
		Output enable	542
		Input	541
T19	P212	Output	540
		Output enable	539
		Input	538
P15	P211	Output	537
		Output enable	536
		Input	535
P16	P210	Output	534
		Output enable	533
		Input	532
N18	P315	Output	531
		Output enable	530
		Input	529
N15	P209	Output	528
		Output enable	527
		Input	526

Table 54.6 Boundary Scan Register 320-Pin LFBGA (2/13)

Pin No.	Pin Name	Input/Output	Bit Name
N16	P208	Output	525
		Output enable	524
		Input	523
M15	P312	Output	522
		Output enable	521
		Input	520
M16	P311	Output	519
		Output enable	518
		Input	517
L15	P310	Output	516
		Output enable	515
		Input	514
L16	P305	Input	513
K15	P207	Output	512
		Output enable	511
		Input	510
K16	P206	Output	509
		Output enable	508
		Input	507
J15	P205	Output	506
		Output enable	505
		Input	504
H15	P204	Output	503
		Output enable	502
		Input	501
H18	PF05	Output	500
		Output enable	499
		Input	498
G16	PF04	Output	497
		Output enable	496
		Input	495
H16	P203	Output	494
		Output enable	493
		Input	492
J16	P202	Output	491
		Output enable	490
		Input	489
G15	PF03	Output	488
		Output enable	487
		Input	486
F15	PF02	Output	485
		Output enable	484
		Input	483

Table 54.6 Boundary Scan Register 320-Pin LFBGA (3/13)

Pin No.	Pin Name	Input/Output	Bit Name
F16	P115	Output	482
		Output enable	481
		Input	480
F18	P114	Output	479
		Output enable	478
		Input	477
E20	P113	Output	476
		Output enable	475
		Input	474
E19	P112	Output	473
		Output enable	472
		Input	471
E18	P111	Output	470
		Output enable	469
		Input	468
D20	P110	Output	467
		Output enable	466
		Input	465
D19	P109	Output	464
		Output enable	463
		Input	462
D18	P108	Output	461
		Output enable	460
		Input	459
C19	P304	Output	458
		Output enable	457
		Input	456
C18	P807	Output	455
		Output enable	454
		Input	453
C17	P806	Output	452
		Output enable	451
		Input	450
C16	P805	Output	449
		Output enable	448
		Input	447
E16	P804	Output	446
		Output enable	445
		Input	444
B18	P201	Output	443
		Output enable	442
		Input	441
A18	P200	Output	440
		Output enable	439
		Input	438

Table 54.6 Boundary Scan Register 320-Pin LFBGA (4/13)

Pin No.	Pin Name	Input/Output	Bit Name
B17	P107	Output	437
		Output enable	436
		Input	435
A17	P106	Output	434
		Output enable	433
		Input	432
C15	P105	Output	431
		Output enable	430
		Input	429
A16	P104	Output	428
		Output enable	427
		Input	426
E15	P507	Output	425
		Output enable	424
		Input	423
C14	P506	Output	422
		Output enable	421
		Input	420
B14	P505	Output	419
		Output enable	418
		Input	417
A14	P504	Output	416
		Output enable	415
		Input	414
A15	P103	Output	413
		Output enable	412
		Input	411
B15	P102	Output	410
		Output enable	409
		Input	408
F12	P101	Output	407
		Output enable	406
		Input	405
E12	P100	Output	404
		Output enable	403
		Input	402
F11	P503	Output	401
		Output enable	400
		Input	399
E11	P502	Output	398
		Output enable	397
		Input	396
F10	P501	Output	395
		Output enable	394
		Input	393

Table 54.6 Boundary Scan Register 320-Pin LFBGA (5/13)

Pin No.	Pin Name	Input/Output	Bit Name
E10	P500	Output	392
		Output enable	391
		Input	390
C9	P803	Output	389
		Output enable	388
		Input	387
B9	P802	Output	386
		Output enable	385
		Input	384
A8	P801	Output	383
		Output enable	382
		Input	381
A9	P800	Output	380
		Output enable	379
		Input	378
A7	PC07	Output	377
		Output enable	376
		Input	375
B8	PC06	Output	374
		Output enable	373
		Input	372
B7	PC05	Output	371
		Output enable	370
		Input	369
A6	PC04	Output	368
		Output enable	367
		Input	366
A5	PC03	Output	365
		Output enable	364
		Input	363
B5	PC02	Output	362
		Output enable	361
		Input	360
A4	PC01	Output	359
		Output enable	358
		Input	357
B4	PC00	Output	356
		Output enable	355
		Input	354
A3	P515	Output	353
		Output enable	352
		Input	351
A2	P514	Output	350
		Output enable	349
		Input	348

Table 54.6 Boundary Scan Register 320-Pin LFBGA (6/13)

Pin No.	Pin Name	Input/Output	Bit Name
B3	P513	Output	347
		Output enable	346
		Input	345
B2	P512	Output	344
		Output enable	343
		Input	342
C7	PB07	Output	341
		Output enable	340
		Input	339
C6	PB06	Output	338
		Output enable	337
		Input	336
C8	PB05	Output	335
		Output enable	334
		Input	333
C5	PB04	Output	332
		Output enable	331
		Input	330
C4	PB03	Output	329
		Output enable	328
		Input	327
E6	PB02	Output	326
		Output enable	325
		Input	324
C3	PB01	Output	323
		Output enable	322
		Input	321
E7	PB00	Output	320
		Output enable	319
		Input	318
F5	PG07	Output	317
		Output enable	316
		Input	315
F6	PG06	Output	314
		Output enable	313
		Input	312
E5	PG05	Output	311
		Output enable	310
		Input	309
F7	PG04	Output	308
		Output enable	307
		Input	306
D3	P511	Output	305
		Output enable	304
		Input	303

Table 54.6 Boundary Scan Register 320-Pin LFBGA (7/13)

Pin No.	Pin Name	Input/Output	Bit Name
E3	P510	Output	302
		Output enable	301
		Input	300
F3	P509	Output	299
		Output enable	298
		Input	297
G3	P508	Output	296
		Output enable	295
		Input	294
B1	PG03	Output	293
		Output enable	292
		Input	291
C1	PG02	Output	290
		Output enable	289
		Input	288
C2	PG01	Output	287
		Output enable	286
		Input	285
D2	PG00	Output	284
		Output enable	283
		Input	282
E2	PA07	Output	281
		Output enable	280
		Input	279
D1	PA06	Output	278
		Output enable	277
		Input	276
E1	PA05	Output	275
		Output enable	274
		Input	273
F2	PA04	Output	272
		Output enable	271
		Input	270
F1	PA03	Output	269
		Output enable	268
		Input	267
G1	PA02	Output	266
		Output enable	265
		Input	264
G2	PA01	Output	263
		Output enable	262
		Input	261
H1	PA00	Output	260
		Output enable	259
		Input	258

Table 54.6 Boundary Scan Register 320-Pin LFBGA (8/13)

Pin No.	Pin Name	Input/Output	Bit Name
H3	PG15	Output	257
		Output enable	256
		Input	255
J2	PG14	Output	254
		Output enable	253
		Input	252
H2	PG13	Output	251
		Output enable	250
		Input	249
J3	PG12	Output	248
		Output enable	247
		Input	246
L3	P707	Output	245
		Output enable	244
		Input	243
L5	P706	Output	242
		Output enable	241
		Input	240
K3	P705	Output	239
		Output enable	238
		Input	237
K5	P704	Output	236
		Output enable	235
		Input	234
J1	PG11	Output	233
		Output enable	232
		Input	231
K2	PG10	Output	230
		Output enable	229
		Input	228
K1	PG09	Output	227
		Output enable	226
		Input	225
L1	PG08	Output	224
		Output enable	223
		Input	222
L6	P703	Output	221
		Output enable	220
		Input	219
K6	P702	Output	218
		Output enable	217
		Input	216
M1	P701	Output	215
		Output enable	214
		Input	213

Table 54.6 Boundary Scan Register 320-Pin LFBGA (9/13)

Pin No.	Pin Name	Input/Output	Bit Name
N1	P700	Output	212
		Output enable	211
		Input	210
M2	P715	Output	209
		Output enable	208
		Input	207
M3	P714	Output	206
		Output enable	205
		Input	204
N3	P713	Output	203
		Output enable	202
		Input	201
N2	P712	Output	200
		Output enable	199
		Input	198
P3	PF09	Output	197
		Output enable	196
		Input	195
R3	PF08	Output	194
		Output enable	193
		Input	192
P2	PE07	Output	191
		Output enable	190
		Input	189
P1	PE06	Output	188
		Output enable	187
		Input	186
R2	PE05	Output	185
		Output enable	184
		Input	183
T2	PE04	Output	182
		Output enable	181
		Input	180
R1	PE03	Output	179
		Output enable	178
		Input	177
U2	PE02	Output	176
		Output enable	175
		Input	174
T1	PE01	Output	173
		Output enable	172
		Input	171
U1	PE00	Output	170
		Output enable	169
		Input	168

Table 54.6 Boundary Scan Register 320-Pin LFBGA (10/13)

Pin No.	Pin Name	Input/Output	Bit Name
T3	PF07	Output	167
		Output enable	166
		Input	165
U3	PF06	Output	164
		Output enable	163
		Input	162
V2	P711	Output	161
		Output enable	160
		Input	159
W2	P710	Output	158
		Output enable	157
		Input	156
V1	P709	Output	155
		Output enable	154
		Input	153
W1	P708	Output	152
		Output enable	151
		Input	150
R5	P607	Output	149
		Output enable	148
		Input	147
R6	P606	Output	146
		Output enable	145
		Input	144
V3	P605	Output	143
		Output enable	142
		Input	141
T5	P604	Output	140
		Output enable	139
		Input	138
V4	P603	Output	137
		Output enable	136
		Input	135
T6	P602	Output	134
		Output enable	133
		Input	132
V6	P601	Output	131
		Output enable	130
		Input	129
V5	P600	Output	128
		Output enable	127
		Input	126
Y2	PD07	Output	125
		Output enable	124
		Input	123

Table 54.6 Boundary Scan Register 320-Pin LFBGA (11/13)

Pin No.	Pin Name	Input/Output	Bit Name
Y3	PD06	Output	122
		Output enable	121
		Input	120
W3	PD05	Output	119
		Output enable	118
		Input	117
Y4	PD04	Output	116
		Output enable	115
		Input	114
W4	PD03	Output	113
		Output enable	112
		Input	111
W6	PD02	Output	110
		Output enable	109
		Input	108
Y5	PD01	Output	107
		Output enable	106
		Input	105
Y6	PD00	Output	104
		Output enable	103
		Input	102
T10	P907	Output	101
		Output enable	100
		Input	99
R10	P906	Output	98
		Output enable	97
		Input	96
V7	P905	Output	95
		Output enable	94
		Input	93
V8	P904	Output	92
		Output enable	91
		Input	90
Y7	P903	Output	89
		Output enable	88
		Input	87
W7	P902	Output	86
		Output enable	85
		Input	84
Y8	P901	Output	83
		Output enable	82
		Input	81
W8	P900	Output	80
		Output enable	79
		Input	78

Table 54.6 Boundary Scan Register 320-Pin LFBGA (12/13)

Pin No.	Pin Name	Input/Output	Bit Name
T12	P407	Output	77
		Output enable	76
		Input	75
T11	P406	Output	74
		Output enable	73
		Input	72
R11	P405	Output	71
		Output enable	70
		Input	69
R12	P404	Output	68
		Output enable	67
		Input	66
W13	P403	Output	65
		Output enable	64
		Input	63
Y13	P402	Output	62
		Output enable	61
		Input	60
W14	P401	Output	59
		Output enable	58
		Input	57
Y14	P400	Output	56
		Output enable	55
		Input	54
V14	P413	Output	53
		Output enable	52
		Input	51
V13	P412	Output	50
		Output enable	49
		Input	48
W15	P411	Output	47
		Output enable	46
		Input	45
Y15	P410	Output	44
		Output enable	43
		Input	42
W16	P409	Output	41
		Output enable	40
		Input	39
Y16	P408	Output	38
		Output enable	37
		Input	36
V16	P415	Output	35
		Output enable	34
		Input	33

Table 54.6 Boundary Scan Register 320-Pin LFBGA (13/13)

Pin No.	Pin Name	Input/Output	Bit Name
V15	P414	Output	32
		Output enable	31
		Input	30
Y17	P015	Output	29
		Output enable	28
		Input	27
V17	P014	Output	26
		Output enable	25
		Input	24
T15	P013	Output	23
		Output enable	22
		Input	21
T16	P012	Output	20
		Output enable	19
		Input	18
V18	P011	Output	17
		Output enable	16
		Input	15
V19	P010	Output	14
		Output enable	13
		Input	12
Y18	P007	Output	11
		Output enable	10
		Input	9
W18	P006	Output	8
		Output enable	7
		Input	6
Y19	P005	Output	5
		Output enable	4
		Input	3
W19	P004	Output	2
		Output enable	1
		Input	0
To TDO			

Table 54.7 Boundary Scan Register 224-Pin LFBGA (1/11)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
P14	P003	Output	567
		Output enable	566
		Input	565
R14	P002	Output	564
		Output enable	563
		Input	562
P15	P001	Output	561
		Output enable	560
		Input	559
N15	P000	Output	558
		Output enable	557
		Input	556
L10	P215	Output	549
		Output enable	548
		Input	547
M12	P214	Output	546
		Output enable	545
		Input	544
N12	P213	Output	543
		Output enable	542
		Input	541
M11	P212	Output	540
		Output enable	539
		Input	538
N14	P211	Output	537
		Output enable	536
		Input	535
N11	P210	Output	534
		Output enable	533
		Input	532
K13	P315	Output	531
		Output enable	530
		Input	529
L11	P209	Output	528
		Output enable	527
		Input	526
N13	P208	Output	525
		Output enable	524
		Input	523
L12	P312	Output	522
		Output enable	521
		Input	520

Table 54.7 Boundary Scan Register 224-Pin LFBGA (2/11)

Pin No.	Pin Name	Input/Output	Bit Name
K12	P311	Output	519
		Output enable	518
		Input	517
K11	P310	Output	516
		Output enable	515
		Input	514
J12	P305	Input	513
J11	P207	Output	512
		Output enable	511
		Input	510
H11	P206	Output	509
		Output enable	508
		Input	507
G11	P205	Output	506
		Output enable	505
		Input	504
G12	P204	Output	503
		Output enable	502
		Input	501
F11	PF05	Output	500
		Output enable	499
		Input	498
E12	PF04	Output	497
		Output enable	496
		Input	495
F13	P203	Output	494
		Output enable	493
		Input	492
F12	P202	Output	491
		Output enable	490
		Input	489
E13	PF03	Output	488
		Output enable	487
		Input	486
E11	PF02	Output	485
		Output enable	484
		Input	483
D12	P115	Output	482
		Output enable	481
		Input	480
C11	P114	Output	479
		Output enable	478
		Input	477

Table 54.7 Boundary Scan Register 224-Pin LFBGA (3/11)

Pin No.	Pin Name	Input/Output	Bit Name
C15	P113	Output	476
		Output enable	475
		Input	474
D13	P112	Output	473
		Output enable	472
		Input	471
C12	P111	Output	470
		Output enable	469
		Input	468
C14	P110	Output	467
		Output enable	466
		Input	465
D11	P109	Output	464
		Output enable	463
		Input	462
C13	P108	Output	461
		Output enable	460
		Input	459
B15	P304	Output	458
		Output enable	457
		Input	456
E10	P807	Output	455
		Output enable	454
		Input	453
D10	P806	Output	452
		Output enable	451
		Input	450
C10	P805	Output	449
		Output enable	448
		Input	447
C9	P804	Output	446
		Output enable	445
		Input	444
B12	P201	Output	443
		Output enable	442
		Input	441
A12	P200	Output	440
		Output enable	439
		Input	438
B11	P107	Output	437
		Output enable	436
		Input	435
A11	P106	Output	434
		Output enable	433
		Input	432

Table 54.7 Boundary Scan Register 224-Pin LFBGA (4/11)

Pin No.	Pin Name	Input/Output	Bit Name
B10	P105	Output	431
		Output enable	430
		Input	429
A10	P104	Output	428
		Output enable	427
		Input	426
E9	P507	Output	425
		Output enable	424
		Input	423
D9	P506	Output	422
		Output enable	421
		Input	420
E8	P505	Output	419
		Output enable	418
		Input	417
D8	P504	Output	416
		Output enable	415
		Input	414
A9	P103	Output	413
		Output enable	412
		Input	411
B9	P102	Output	410
		Output enable	409
		Input	408
D7	P101	Output	407
		Output enable	406
		Input	405
C7	P100	Output	404
		Output enable	403
		Input	402
E7	P503	Output	401
		Output enable	400
		Input	399
D5	P502	Output	398
		Output enable	397
		Input	396
E6	P501	Output	395
		Output enable	394
		Input	393
D6	P500	Output	392
		Output enable	391
		Input	390
B5	P803	Output	389
		Output enable	388
		Input	387

Table 54.7 Boundary Scan Register 224-Pin LFBGA (5/11)

Pin No.	Pin Name	Input/Output	Bit Name
A5	P802	Output	386
		Output enable	385
		Input	384
A4	P801	Output	383
		Output enable	382
		Input	381
C5	P800	Output	380
		Output enable	379
		Input	378
A3	PC07	Output	377
		Output enable	376
		Input	375
B4	PC06	Output	374
		Output enable	373
		Input	372
B3	PC05	Output	371
		Output enable	370
		Input	369
B1	PC04	Output	368
		Output enable	367
		Input	366
A2	PC03	Output	365
		Output enable	364
		Input	363
C2	PC02	Output	362
		Output enable	361
		Input	360
B2	PC01	Output	359
		Output enable	358
		Input	357
C1	PC00	Output	356
		Output enable	355
		Input	354
D4	PB07	Output	341
		Output enable	340
		Input	339
C3	PB06	Output	338
		Output enable	337
		Input	336
C4	PB05	Output	335
		Output enable	334
		Input	333
D3	PB04	Output	332
		Output enable	331
		Input	330

Table 54.7 Boundary Scan Register 224-Pin LFBGA (6/11)

Pin No.	Pin Name	Input/Output	Bit Name
E4	PB03	Output	329
		Output enable	328
		Input	327
E3	PB02	Output	326
		Output enable	325
		Input	324
F3	PB01	Output	323
		Output enable	322
		Input	321
F4	PB00	Output	320
		Output enable	319
		Input	318
G4	PG07	Output	317
		Output enable	316
		Input	315
H3	PG06	Output	314
		Output enable	313
		Input	312
G3	PG05	Output	311
		Output enable	310
		Input	309
H4	PG04	Output	308
		Output enable	307
		Input	306
E2	PG03	Output	293
		Output enable	292
		Input	291
D2	PG02	Output	290
		Output enable	289
		Input	288
E1	PG01	Output	287
		Output enable	286
		Input	285
D1	PG00	Output	284
		Output enable	283
		Input	282
F2	PA07	Output	281
		Output enable	280
		Input	279
G2	PA06	Output	278
		Output enable	277
		Input	276
F1	PA05	Output	275
		Output enable	274
		Input	273

Table 54.7 Boundary Scan Register 224-Pin LFBGA (7/11)

Pin No.	Pin Name	Input/Output	Bit Name
G1	PA04	Output	272
		Output enable	271
		Input	270
H2	PA03	Output	269
		Output enable	268
		Input	267
H1	PA02	Output	266
		Output enable	265
		Input	264
J2	PA01	Output	263
		Output enable	262
		Input	261
J1	PA00	Output	260
		Output enable	259
		Input	258
K1	P707	Output	245
		Output enable	244
		Input	243
K2	P706	Output	242
		Output enable	241
		Input	240
L1	P705	Output	239
		Output enable	238
		Input	237
L2	P704	Output	236
		Output enable	235
		Input	234
J3	P703	Output	221
		Output enable	220
		Input	219
K3	P702	Output	218
		Output enable	217
		Input	216
M2	P701	Output	215
		Output enable	214
		Input	213
M1	P700	Output	212
		Output enable	211
		Input	210
G5	PF09	Output	197
		Output enable	196
		Input	195
H5	PF08	Output	194
		Output enable	193
		Input	192

Table 54.7 Boundary Scan Register 224-Pin LFBGA (8/11)

Pin No.	Pin Name	Input/Output	Bit Name
N1	PE07	Output	191
		Output enable	190
		Input	189
N3	PE06	Output	188
		Output enable	187
		Input	186
P1	PE05	Output	185
		Output enable	184
		Input	183
N2	PE04	Output	182
		Output enable	181
		Input	180
P2	PE03	Output	179
		Output enable	178
		Input	177
P3	PE02	Output	176
		Output enable	175
		Input	174
R2	PE01	Output	173
		Output enable	172
		Input	171
R3	PE00	Output	170
		Output enable	169
		Input	168
J5	PF07	Output	167
		Output enable	166
		Input	165
J4	PF06	Output	164
		Output enable	163
		Input	162
M3	P607	Output	149
		Output enable	148
		Input	147
L4	P606	Output	146
		Output enable	145
		Input	144
K4	P605	Output	143
		Output enable	142
		Input	141
L3	P604	Output	140
		Output enable	139
		Input	138
M4	P603	Output	137
		Output enable	136
		Input	135

Table 54.7 Boundary Scan Register 224-Pin LFBGA (9/11)

Pin No.	Pin Name	Input/Output	Bit Name
N4	P602	Output	134
		Output enable	133
		Input	132
N5	P601	Output	131
		Output enable	130
		Input	129
N6	P600	Output	128
		Output enable	127
		Input	126
P4	PD07	Output	125
		Output enable	124
		Input	123
R4	PD06	Output	122
		Output enable	121
		Input	120
P5	PD05	Output	119
		Output enable	118
		Input	117
R5	PD04	Output	116
		Output enable	115
		Input	114
P6	PD03	Output	113
		Output enable	112
		Input	111
P7	PD02	Output	110
		Output enable	109
		Input	108
R7	PD01	Output	107
		Output enable	106
		Input	105
R6	PD00	Output	104
		Output enable	103
		Input	102
M5	P907	Output	101
		Output enable	100
		Input	99
L7	P906	Output	98
		Output enable	97
		Input	96
M6	P905	Output	95
		Output enable	94
		Input	93
L6	P904	Output	92
		Output enable	91
		Input	90

Table 54.7 Boundary Scan Register 224-Pin LFBGA (10/11)

Pin No.	Pin Name	Input/Output	Bit Name
N7	P903	Output	89
		Output enable	88
		Input	87
M7	P902	Output	86
		Output enable	85
		Input	84
N8	P901	Output	83
		Output enable	82
		Input	81
M8	P900	Output	80
		Output enable	79
		Input	78
L8	P407	Output	77
		Output enable	76
		Input	75
L9	P406	Output	74
		Output enable	73
		Input	72
M9	P405	Output	71
		Output enable	70
		Input	69
N9	P404	Output	68
		Output enable	67
		Input	66
P10	P403	Output	65
		Output enable	64
		Input	63
R10	P402	Output	62
		Output enable	61
		Input	60
P11	P401	Output	59
		Output enable	58
		Input	57
R11	P400	Output	56
		Output enable	55
		Input	54
M10	P415	Output	35
		Output enable	34
		Input	33
N10	P414	Output	32
		Output enable	31
		Input	30
R12	P007	Output	11
		Output enable	10
		Input	9

Table 54.7 Boundary Scan Register 224-Pin LFBGA (11/11)

Pin No.	Pin Name	Input/Output	Bit Name
P12	P006	Output	8
		Output enable	7
		Input	6
R13	P005	Output	5
		Output enable	4
		Input	3
P13	P004	Output	2
		Output enable	1
		Input	0
To TDO			

Table 54.8 Boundary Scan Register 176-Pin LFBGA (1/8)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
N14	P003	Output	567
		Output enable	566
		Input	565
N15	P002	Output	564
		Output enable	563
		Input	562
M14	P001	Output	561
		Output enable	560
		Input	559
M15	P000	Output	558
		Output enable	557
		Input	556
M12	P315	Output	531
		Output enable	530
		Input	529
L12	P312	Output	522
		Output enable	521
		Input	520
L13	P311	Output	519
		Output enable	518
		Input	517
K13	P310	Output	516
		Output enable	515
		Input	514
J13	P305	Input	513
H12	P207	Output	512
		Output enable	511
		Input	510
E13	P206	Output	509
		Output enable	508
		Input	507
C12	P205	Output	506
		Output enable	505
		Input	504
D13	P204	Output	503
		Output enable	502
		Input	501
C15	P203	Output	494
		Output enable	493
		Input	492
C14	P202	Output	491
		Output enable	490
		Input	489

Table 54.8 Boundary Scan Register 176-Pin LFBGA (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
C13	P304	Output	458
		Output enable	457
		Input	456
A13	P805	Output	449
		Output enable	448
		Input	447
B13	P804	Output	446
		Output enable	445
		Input	444
B12	P201	Output	443
		Output enable	442
		Input	441
A12	P200	Output	440
		Output enable	439
		Input	438
B11	P107	Output	437
		Output enable	436
		Input	435
A11	P106	Output	434
		Output enable	433
		Input	432
D11	P105	Output	431
		Output enable	430
		Input	429
C11	P104	Output	428
		Output enable	427
		Input	426
B10	P103	Output	413
		Output enable	412
		Input	411
A10	P102	Output	410
		Output enable	409
		Input	408
D8	P101	Output	407
		Output enable	406
		Input	405
C10	P100	Output	404
		Output enable	403
		Input	402
A6	P503	Output	401
		Output enable	400
		Input	399
B6	P502	Output	398
		Output enable	397
		Input	396

Table 54.8 Boundary Scan Register 176-Pin LFBGA (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
C6	P501	Output	395
		Output enable	394
		Input	393
C5	P500	Output	392
		Output enable	391
		Input	390
B5	P803	Output	389
		Output enable	388
		Input	387
C3	P802	Output	386
		Output enable	385
		Input	384
A5	P801	Output	383
		Output enable	382
		Input	381
D4	P800	Output	380
		Output enable	379
		Input	378
A4	PC07	Output	377
		Output enable	376
		Input	375
B4	PC06	Output	374
		Output enable	373
		Input	372
A3	PC05	Output	371
		Output enable	370
		Input	369
B3	PC04	Output	368
		Output enable	367
		Input	366
A2	PC03	Output	365
		Output enable	364
		Input	363
B2	PC02	Output	362
		Output enable	361
		Input	360
A1	PC01	Output	359
		Output enable	358
		Input	357
B1	PC00	Output	356
		Output enable	355
		Input	354
C2	PB07	Output	341
		Output enable	340
		Input	339

Table 54.8 Boundary Scan Register 176-Pin LFBGA (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
C1	PB06	Output	338
		Output enable	337
		Input	336
D2	PB05	Output	335
		Output enable	334
		Input	333
D1	PB04	Output	332
		Output enable	331
		Input	330
E2	PB03	Output	329
		Output enable	328
		Input	327
E1	PB02	Output	326
		Output enable	325
		Input	324
F2	PB01	Output	323
		Output enable	322
		Input	321
F1	PB00	Output	320
		Output enable	319
		Input	318
F3	PG02	Output	290
		Output enable	289
		Input	288
D3	PG01	Output	287
		Output enable	286
		Input	285
E3	PG00	Output	284
		Output enable	283
		Input	282
G2	PA07	Output	281
		Output enable	280
		Input	279
G1	PA06	Output	278
		Output enable	277
		Input	276
H2	PA05	Output	275
		Output enable	274
		Input	273
H1	PA04	Output	272
		Output enable	271
		Input	270
J2	PA03	Output	269
		Output enable	268
		Input	267

Table 54.8 Boundary Scan Register 176-Pin LFBGA (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
J1	PA02	Output	266
		Output enable	265
		Input	264
K2	PA01	Output	263
		Output enable	262
		Input	261
K1	PA00	Output	260
		Output enable	259
		Input	258
H3	P707	Output	245
		Output enable	244
		Input	243
G3	P706	Output	242
		Output enable	241
		Input	240
J3	P705	Output	239
		Output enable	238
		Input	237
L4	P704	Output	236
		Output enable	235
		Input	234
L3	P703	Output	221
		Output enable	220
		Input	219
K3	P702	Output	218
		Output enable	217
		Input	216
L2	P701	Output	215
		Output enable	214
		Input	213
L1	P700	Output	212
		Output enable	211
		Input	210
M1	PE07	Output	191
		Output enable	190
		Input	189
R2	PE06	Output	188
		Output enable	187
		Input	186
N1	PE05	Output	185
		Output enable	184
		Input	183
N2	PE04	Output	182
		Output enable	181
		Input	180

Table 54.8 Boundary Scan Register 176-Pin LFBGA (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
M2	PE03	Output	179
		Output enable	178
		Input	177
P2	PE02	Output	176
		Output enable	175
		Input	174
P1	PE01	Output	173
		Output enable	172
		Input	171
R1	PE00	Output	170
		Output enable	169
		Input	168
N3	P607	Output	149
		Output enable	148
		Input	147
P3	P606	Output	146
		Output enable	145
		Input	144
R3	P605	Output	143
		Output enable	142
		Input	141
N4	P604	Output	140
		Output enable	139
		Input	138
P4	P603	Output	137
		Output enable	136
		Input	135
R4	P602	Output	134
		Output enable	133
		Input	132
P5	P601	Output	131
		Output enable	130
		Input	129
R5	P600	Output	128
		Output enable	127
		Input	126
N5	PD07	Output	125
		Output enable	124
		Input	123
M4	PD06	Output	122
		Output enable	121
		Input	120
N6	PD05	Output	119
		Output enable	118
		Input	117

Table 54.8 Boundary Scan Register 176-Pin LFBGA (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
M3	PD04	Output	116
		Output enable	115
		Input	114
P6	PD03	Output	113
		Output enable	112
		Input	111
R6	PD02	Output	110
		Output enable	109
		Input	108
R7	PD01	Output	107
		Output enable	106
		Input	105
P7	PD00	Output	104
		Output enable	103
		Input	102
R8	P903	Output	89
		Output enable	88
		Input	87
P8	P902	Output	86
		Output enable	85
		Input	84
R9	P901	Output	83
		Output enable	82
		Input	81
P9	P900	Output	80
		Output enable	79
		Input	78
M10	P407	Output	77
		Output enable	76
		Input	75
N10	P406	Output	74
		Output enable	73
		Input	72
N11	P405	Output	71
		Output enable	70
		Input	69
M11	P404	Output	68
		Output enable	67
		Input	66
R12	P403	Output	65
		Output enable	64
		Input	63
P12	P402	Output	62
		Output enable	61
		Input	60

Table 54.8 Boundary Scan Register 176-Pin LFBGA (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
R13	P401	Output	59
		Output enable	58
		Input	57
P13	P400	Output	56
		Output enable	55
		Input	54
N12	P415	Output	35
		Output enable	34
		Input	33
N13	P414	Output	32
		Output enable	31
		Input	30
R14	P007	Output	11
		Output enable	10
		Input	9
R15	P006	Output	8
		Output enable	7
		Input	6
P14	P005	Output	5
		Output enable	4
		Input	3
P15	P004	Output	2
		Output enable	1
		Input	0
To TDO			

Table 54.9 Boundary Scan Register 176-Pin LFQFP (1/8)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
1	P003	Output	567
		Output enable	566
		Input	565
2	P002	Output	564
		Output enable	563
		Input	562
3	P001	Output	561
		Output enable	560
		Input	559
4	P000	Output	558
		Output enable	557
		Input	556
10	P315	Output	531
		Output enable	530
		Input	529
11	P312	Output	522
		Output enable	521
		Input	520
12	P311	Output	519
		Output enable	518
		Input	517
13	P310	Output	516
		Output enable	515
		Input	514
21	P305	Input	513
33	P207	Output	512
		Output enable	511
		Input	510
34	P206	Output	509
		Output enable	508
		Input	507
35	P205	Output	506
		Output enable	505
		Input	504
36	P204	Output	503
		Output enable	502
		Input	501
40	P203	Output	494
		Output enable	493
		Input	492
41	P202	Output	491
		Output enable	490
		Input	489

Table 54.9 Boundary Scan Register 176-Pin LFQFP (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
42	P304	Output	458
		Output enable	457
		Input	456
47	P805	Output	449
		Output enable	448
		Input	447
48	P804	Output	446
		Output enable	445
		Input	444
49	P201	Output	443
		Output enable	442
		Input	441
50	P200	Output	440
		Output enable	439
		Input	438
51	P107	Output	437
		Output enable	436
		Input	435
52	P106	Output	434
		Output enable	433
		Input	432
53	P105	Output	431
		Output enable	430
		Input	429
54	P104	Output	428
		Output enable	427
		Input	426
58	P103	Output	413
		Output enable	412
		Input	411
59	P102	Output	410
		Output enable	409
		Input	408
60	P101	Output	407
		Output enable	406
		Input	405
61	P100	Output	404
		Output enable	403
		Input	402
70	P503	Output	401
		Output enable	400
		Input	399
71	P502	Output	398
		Output enable	397
		Input	396

Table 54.9 Boundary Scan Register 176-Pin LFQFP (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
72	P501	Output	395
		Output enable	394
		Input	393
73	P500	Output	392
		Output enable	391
		Input	390
74	P803	Output	389
		Output enable	388
		Input	387
75	P802	Output	386
		Output enable	385
		Input	384
79	P801	Output	383
		Output enable	382
		Input	381
80	P800	Output	380
		Output enable	379
		Input	378
81	PC07	Output	377
		Output enable	376
		Input	375
82	PC06	Output	374
		Output enable	373
		Input	372
83	PC05	Output	371
		Output enable	370
		Input	369
84	PC04	Output	368
		Output enable	367
		Input	366
85	PC03	Output	365
		Output enable	364
		Input	363
86	PC02	Output	362
		Output enable	361
		Input	360
87	PC01	Output	359
		Output enable	358
		Input	357
88	PC00	Output	356
		Output enable	355
		Input	354
89	PB07	Output	341
		Output enable	340
		Input	339

Table 54.9 Boundary Scan Register 176-Pin LFQFP (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
90	PB06	Output	338
		Output enable	337
		Input	336
91	PB05	Output	335
		Output enable	334
		Input	333
92	PB04	Output	332
		Output enable	331
		Input	330
96	PB03	Output	329
		Output enable	328
		Input	327
97	PB02	Output	326
		Output enable	325
		Input	324
98	PB01	Output	323
		Output enable	322
		Input	321
99	PB00	Output	320
		Output enable	319
		Input	318
100	PG02	Output	290
		Output enable	289
		Input	288
101	PG01	Output	287
		Output enable	286
		Input	285
102	PG00	Output	284
		Output enable	283
		Input	282
103	PA07	Output	281
		Output enable	280
		Input	279
104	PA06	Output	278
		Output enable	277
		Input	276
105	PA05	Output	275
		Output enable	274
		Input	273
106	PA04	Output	272
		Output enable	271
		Input	270
110	PA03	Output	269
		Output enable	268
		Input	267

Table 54.9 Boundary Scan Register 176-Pin LFQFP (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
111	PA02	Output	266
		Output enable	265
		Input	264
112	PA01	Output	263
		Output enable	262
		Input	261
113	PA00	Output	260
		Output enable	259
		Input	258
114	P707	Output	245
		Output enable	244
		Input	243
115	P706	Output	242
		Output enable	241
		Input	240
116	P705	Output	239
		Output enable	238
		Input	237
117	P704	Output	236
		Output enable	235
		Input	234
118	P703	Output	221
		Output enable	220
		Input	219
119	P702	Output	218
		Output enable	217
		Input	216
120	P701	Output	215
		Output enable	214
		Input	213
121	P700	Output	212
		Output enable	211
		Input	210
125	PE07	Output	191
		Output enable	190
		Input	189
126	PE06	Output	188
		Output enable	187
		Input	186
127	PE05	Output	185
		Output enable	184
		Input	183
128	PE04	Output	182
		Output enable	181
		Input	180

Table 54.9 Boundary Scan Register 176-Pin LFQFP (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
129	PE03	Output	179
		Output enable	178
		Input	177
130	PE02	Output	176
		Output enable	175
		Input	174
131	PE01	Output	173
		Output enable	172
		Input	171
132	PE00	Output	170
		Output enable	169
		Input	168
133	P607	Output	149
		Output enable	148
		Input	147
134	P606	Output	146
		Output enable	145
		Input	144
135	P605	Output	143
		Output enable	142
		Input	141
136	P604	Output	140
		Output enable	139
		Input	138
137	P603	Output	137
		Output enable	136
		Input	135
138	P602	Output	134
		Output enable	133
		Input	132
139	P601	Output	131
		Output enable	130
		Input	129
140	P600	Output	128
		Output enable	127
		Input	126
144	PD07	Output	125
		Output enable	124
		Input	123
145	PD06	Output	122
		Output enable	121
		Input	120
146	PD05	Output	119
		Output enable	118
		Input	117

Table 54.9 Boundary Scan Register 176-Pin LFQFP (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
147	PD04	Output	116
		Output enable	115
		Input	114
148	PD03	Output	113
		Output enable	112
		Input	111
149	PD02	Output	110
		Output enable	109
		Input	108
150	PD01	Output	107
		Output enable	106
		Input	105
151	PD00	Output	104
		Output enable	103
		Input	102
155	P903	Output	89
		Output enable	88
		Input	87
156	P902	Output	86
		Output enable	85
		Input	84
157	P901	Output	83
		Output enable	82
		Input	81
158	P900	Output	80
		Output enable	79
		Input	78
163	P407	Output	77
		Output enable	76
		Input	75
164	P406	Output	74
		Output enable	73
		Input	72
165	P405	Output	71
		Output enable	70
		Input	69
166	P404	Output	68
		Output enable	67
		Input	66
167	P403	Output	65
		Output enable	64
		Input	63
168	P402	Output	62
		Output enable	61
		Input	60

Table 54.9 Boundary Scan Register 176-Pin LFQFP (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
169	P401	Output	59
		Output enable	58
		Input	57
170	P400	Output	56
		Output enable	55
		Input	54
171	P415	Output	35
		Output enable	34
		Input	33
172	P414	Output	32
		Output enable	31
		Input	30
173	P007	Output	11
		Output enable	10
		Input	9
174	P006	Output	8
		Output enable	7
		Input	6
175	P005	Output	5
		Output enable	4
		Input	3
176	P004	Output	2
		Output enable	1
		Input	0
To TDO			

Table 54.10 Boundary Scan Register 144-Pin LFQFP (1/7)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
1	P003	Output	567
		Output enable	566
		Input	565
2	P002	Output	564
		Output enable	563
		Input	562
3	P001	Output	561
		Output enable	560
		Input	559
4	P000	Output	558
		Output enable	557
		Input	556
9	P315	Output	531
		Output enable	530
		Input	529
10	P312	Output	522
		Output enable	521
		Input	520
11	P311	Output	519
		Output enable	518
		Input	517
12	P310	Output	516
		Output enable	515
		Input	514
20	P305	Input	513
25	P207	Output	512
		Output enable	511
		Input	510
26	P206	Output	509
		Output enable	508
		Input	507
27	P205	Output	506
		Output enable	505
		Input	504
28	P204	Output	503
		Output enable	502
		Input	501
32	P203	Output	494
		Output enable	493
		Input	492
33	P202	Output	491
		Output enable	490
		Input	489

Table 54.10 Boundary Scan Register 144-Pin LFQFP (2/7)

Pin No.	Pin Name	Input/Output	Bit Name
34	P304	Output	458
		Output enable	457
		Input	456
39	P201	Output	443
		Output enable	442
		Input	441
40	P200	Output	440
		Output enable	439
		Input	438
41	P107	Output	437
		Output enable	436
		Input	435
42	P106	Output	434
		Output enable	433
		Input	432
43	P105	Output	431
		Output enable	430
		Input	429
44	P104	Output	428
		Output enable	427
		Input	426
48	P103	Output	413
		Output enable	412
		Input	411
49	P102	Output	410
		Output enable	409
		Input	408
50	P101	Output	407
		Output enable	406
		Input	405
51	P100	Output	404
		Output enable	403
		Input	402
58	P803	Output	389
		Output enable	388
		Input	387
59	P802	Output	386
		Output enable	385
		Input	384
63	P801	Output	383
		Output enable	382
		Input	381
64	P800	Output	380
		Output enable	379
		Input	378

Table 54.10 Boundary Scan Register 144-Pin LFQFP (3/7)

Pin No.	Pin Name	Input/Output	Bit Name
65	PC07	Output	377
		Output enable	376
		Input	375
66	PC06	Output	374
		Output enable	373
		Input	372
67	PC05	Output	371
		Output enable	370
		Input	369
68	PC04	Output	368
		Output enable	367
		Input	366
69	PC03	Output	365
		Output enable	364
		Input	363
70	PC02	Output	362
		Output enable	361
		Input	360
71	PC01	Output	359
		Output enable	358
		Input	357
72	PC00	Output	356
		Output enable	355
		Input	354
73	PB07	Output	341
		Output enable	340
		Input	339
74	PB06	Output	338
		Output enable	337
		Input	336
75	PB05	Output	335
		Output enable	334
		Input	333
76	PB04	Output	332
		Output enable	331
		Input	330
80	PB03	Output	329
		Output enable	328
		Input	327
81	PB02	Output	326
		Output enable	325
		Input	324
82	PB01	Output	323
		Output enable	322
		Input	321

Table 54.10 Boundary Scan Register 144-Pin LFQFP (4/7)

Pin No.	Pin Name	Input/Output	Bit Name
83	PB00	Output	320
		Output enable	319
		Input	318
84	PA07	Output	281
		Output enable	280
		Input	279
85	PA06	Output	278
		Output enable	277
		Input	276
86	PA05	Output	275
		Output enable	274
		Input	273
87	PA04	Output	272
		Output enable	271
		Input	270
91	PA03	Output	269
		Output enable	268
		Input	267
92	PA02	Output	266
		Output enable	265
		Input	264
93	PA01	Output	263
		Output enable	262
		Input	261
94	PA00	Output	260
		Output enable	259
		Input	258
95	P702	Output	218
		Output enable	217
		Input	216
96	P701	Output	215
		Output enable	214
		Input	213
97	P700	Output	212
		Output enable	211
		Input	210
101	PE07	Output	191
		Output enable	190
		Input	189
102	PE06	Output	188
		Output enable	187
		Input	186
103	PE05	Output	185
		Output enable	184
		Input	183

Table 54.10 Boundary Scan Register 144-Pin LFQFP (5/7)

Pin No.	Pin Name	Input/Output	Bit Name
104	PE04	Output	182
		Output enable	181
		Input	180
105	PE03	Output	179
		Output enable	178
		Input	177
106	PE02	Output	176
		Output enable	175
		Input	174
107	PE01	Output	173
		Output enable	172
		Input	171
108	PE00	Output	170
		Output enable	169
		Input	168
109	P605	Output	143
		Output enable	142
		Input	141
110	P604	Output	140
		Output enable	139
		Input	138
111	P603	Output	137
		Output enable	136
		Input	135
112	P602	Output	134
		Output enable	133
		Input	132
113	P601	Output	131
		Output enable	130
		Input	129
114	P600	Output	128
		Output enable	127
		Input	126
117	PD07	Output	125
		Output enable	124
		Input	123
118	PD06	Output	122
		Output enable	121
		Input	120
119	PD05	Output	119
		Output enable	118
		Input	117
120	PD04	Output	116
		Output enable	115
		Input	114

Table 54.10 Boundary Scan Register 144-Pin LFQFP (6/7)

Pin No.	Pin Name	Input/Output	Bit Name
121	PD03	Output	113
		Output enable	112
		Input	111
122	PD02	Output	110
		Output enable	109
		Input	108
123	PD01	Output	107
		Output enable	106
		Input	105
124	PD00	Output	104
		Output enable	103
		Input	102
127	P901	Output	83
		Output enable	82
		Input	81
128	P900	Output	80
		Output enable	79
		Input	78
133	P405	Output	71
		Output enable	70
		Input	69
134	P404	Output	68
		Output enable	67
		Input	66
135	P403	Output	65
		Output enable	64
		Input	63
136	P402	Output	62
		Output enable	61
		Input	60
137	P401	Output	59
		Output enable	58
		Input	57
138	P400	Output	56
		Output enable	55
		Input	54
139	P415	Output	35
		Output enable	34
		Input	33
140	P414	Output	32
		Output enable	31
		Input	30
141	P007	Output	11
		Output enable	10
		Input	9

Table 54.10 Boundary Scan Register 144-Pin LFQFP (7/7)

Pin No.	Pin Name	Input/Output	Bit Name
142	P006	Output	8
		Output enable	7
		Input	6
143	P005	Output	5
		Output enable	4
		Input	3
144	P004	Output	2
		Output enable	1
		Input	0
To TDO			

Table 54.11 Boundary Scan Register 100-Pin LFQFP (1/5)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
1	P003	Output	567
		Output enable	566
		Input	565
2	P002	Output	564
		Output enable	563
		Input	562
3	P001	Output	561
		Output enable	560
		Input	559
7	P315	Output	531
		Output enable	530
		Input	529
8	P312	Output	522
		Output enable	521
		Input	520
9	P311	Output	519
		Output enable	518
		Input	517
10	P310	Output	516
		Output enable	515
		Input	514
18	P305	Input	513
23	P304	Output	458
		Output enable	457
		Input	456
28	P201	Output	443
		Output enable	442
		Input	441
29	P200	Output	440
		Output enable	439
		Input	438
30	P107	Output	437
		Output enable	436
		Input	435
31	P106	Output	434
		Output enable	433
		Input	432
32	P105	Output	431
		Output enable	430
		Input	429
33	P104	Output	428
		Output enable	427
		Input	426

Table 54.11 Boundary Scan Register 100-Pin LFQFP (2/5)

Pin No.	Pin Name	Input/Output	Bit Name
37	P103	Output	413
		Output enable	412
		Input	411
38	P102	Output	410
		Output enable	409
		Input	408
39	P101	Output	407
		Output enable	406
		Input	405
44	PC06	Output	374
		Output enable	373
		Input	372
45	PC05	Output	371
		Output enable	370
		Input	369
46	PC04	Output	368
		Output enable	367
		Input	366
47	PC03	Output	365
		Output enable	364
		Input	363
48	PC02	Output	362
		Output enable	361
		Input	360
49	PC01	Output	359
		Output enable	358
		Input	357
50	PC00	Output	356
		Output enable	355
		Input	354
51	PB06	Output	338
		Output enable	337
		Input	336
52	PB05	Output	335
		Output enable	334
		Input	333
53	PB04	Output	332
		Output enable	331
		Input	330
54	PB03	Output	329
		Output enable	328
		Input	327
55	PB02	Output	326
		Output enable	325
		Input	324

Table 54.11 Boundary Scan Register 100-Pin LFQFP (3/5)

Pin No.	Pin Name	Input/Output	Bit Name
56	PB01	Output	323
		Output enable	322
		Input	321
57	PB00	Output	320
		Output enable	319
		Input	318
58	PA07	Output	281
		Output enable	280
		Input	279
59	PA06	Output	278
		Output enable	277
		Input	276
60	PA05	Output	275
		Output enable	274
		Input	273
61	PA04	Output	272
		Output enable	271
		Input	270
65	PA03	Output	269
		Output enable	268
		Input	267
66	PA02	Output	266
		Output enable	265
		Input	264
67	PA01	Output	263
		Output enable	262
		Input	261
68	PA00	Output	260
		Output enable	259
		Input	258
69	PE06	Output	188
		Output enable	187
		Input	186
70	PE05	Output	185
		Output enable	184
		Input	183
71	PE04	Output	182
		Output enable	181
		Input	180
72	PE03	Output	179
		Output enable	178
		Input	177
73	PE02	Output	176
		Output enable	175
		Input	174

Table 54.11 Boundary Scan Register 100-Pin LFQFP (4/5)

Pin No.	Pin Name	Input/Output	Bit Name
74	PE01	Output	173
		Output enable	172
		Input	171
75	PE00	Output	170
		Output enable	169
		Input	168
76	P601	Output	131
		Output enable	130
		Input	129
77	P600	Output	128
		Output enable	127
		Input	126
78	PD07	Output	125
		Output enable	124
		Input	123
79	PD06	Output	122
		Output enable	121
		Input	120
80	PD05	Output	119
		Output enable	118
		Input	117
81	PD04	Output	116
		Output enable	115
		Input	114
82	PD03	Output	113
		Output enable	112
		Input	111
83	PD02	Output	110
		Output enable	109
		Input	108
84	PD01	Output	107
		Output enable	106
		Input	105
85	PD00	Output	104
		Output enable	103
		Input	102
93	P404	Output	68
		Output enable	67
		Input	66
94	P402	Output	62
		Output enable	61
		Input	60
95	P400	Output	56
		Output enable	55
		Input	54

Table 54.11 Boundary Scan Register 100-Pin LFQFP (5/5)

Pin No.	Pin Name	Input/Output	Bit Name
96	P414	Output	32
		Output enable	31
		Input	30
97	P007	Output	11
		Output enable	10
		Input	9
98	P006	Output	8
		Output enable	7
		Input	6
99	P005	Output	5
		Output enable	4
		Input	3
100	P004	Output	2
		Output enable	1
		Input	0
To TDO			

54.3 Operations

While the signal on the RES# pin is asserted, the port pins multiplexed with the JTAG functions are assigned to TCK, TMS, TDI, and TDO by default, and the internal pull-up resistors are enabled for the TCK, TMS, and TDI pins. After the power-on reset signal is deasserted, the boundary scan test can be executed once the setup time has elapsed while the RES# signal is at the low level.

54.3.1 TAP Controller

Figure 54.2 shows the state transition diagram of the TAP controller.

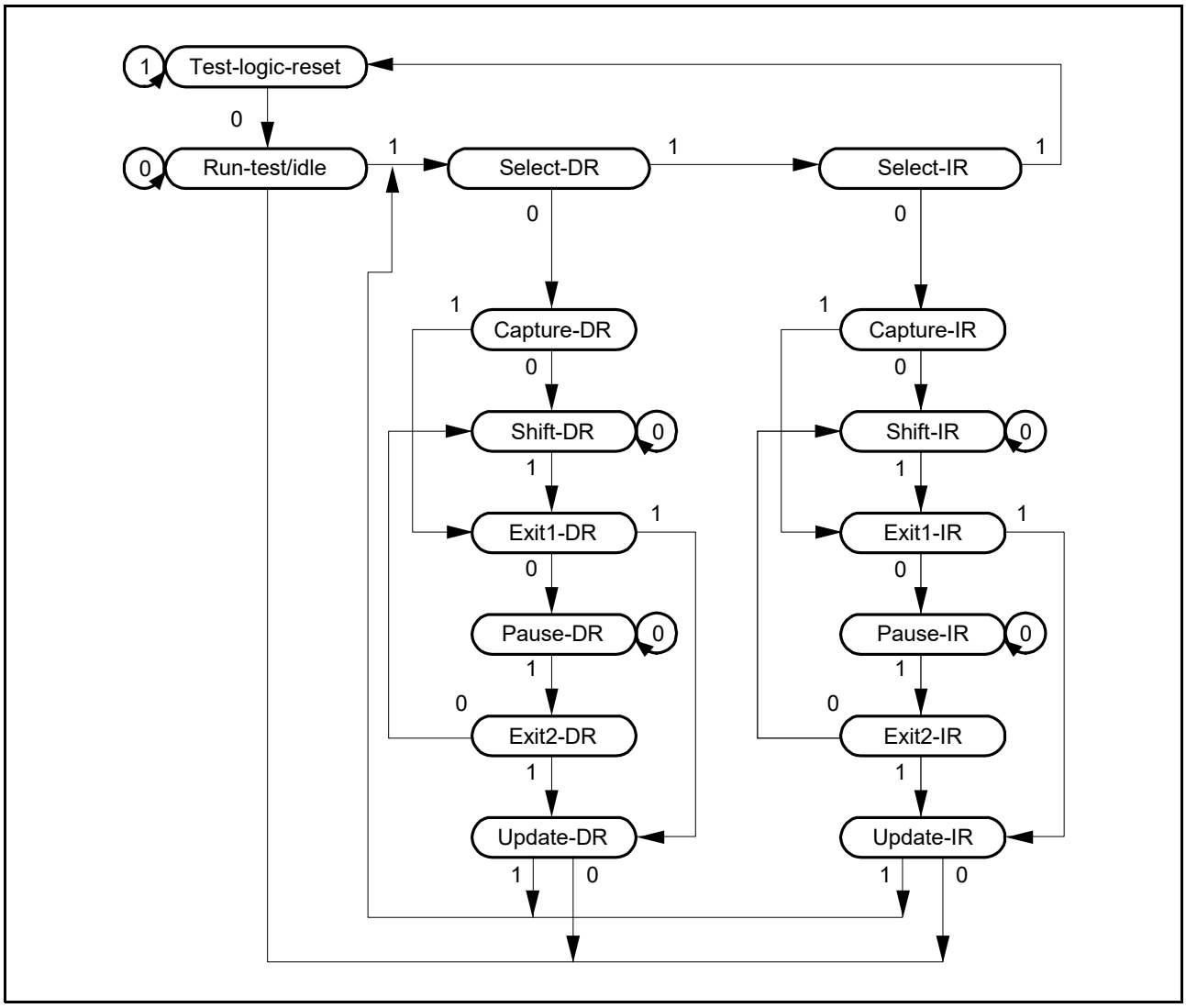


Figure 54.2 State Transition Diagram of TAP Controller

54.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction is an instruction that drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.
The bypass register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is low in the first clock cycle in the shift-DR state; in the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.
In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.
In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. JTIR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state.
BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of boundary scan register is maintained regardless of the state of the TAP controller.
BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

54.4 Usage Notes

(1) Pin serial transfer, data are input or output in LSB order (refer to Figure 54.3).

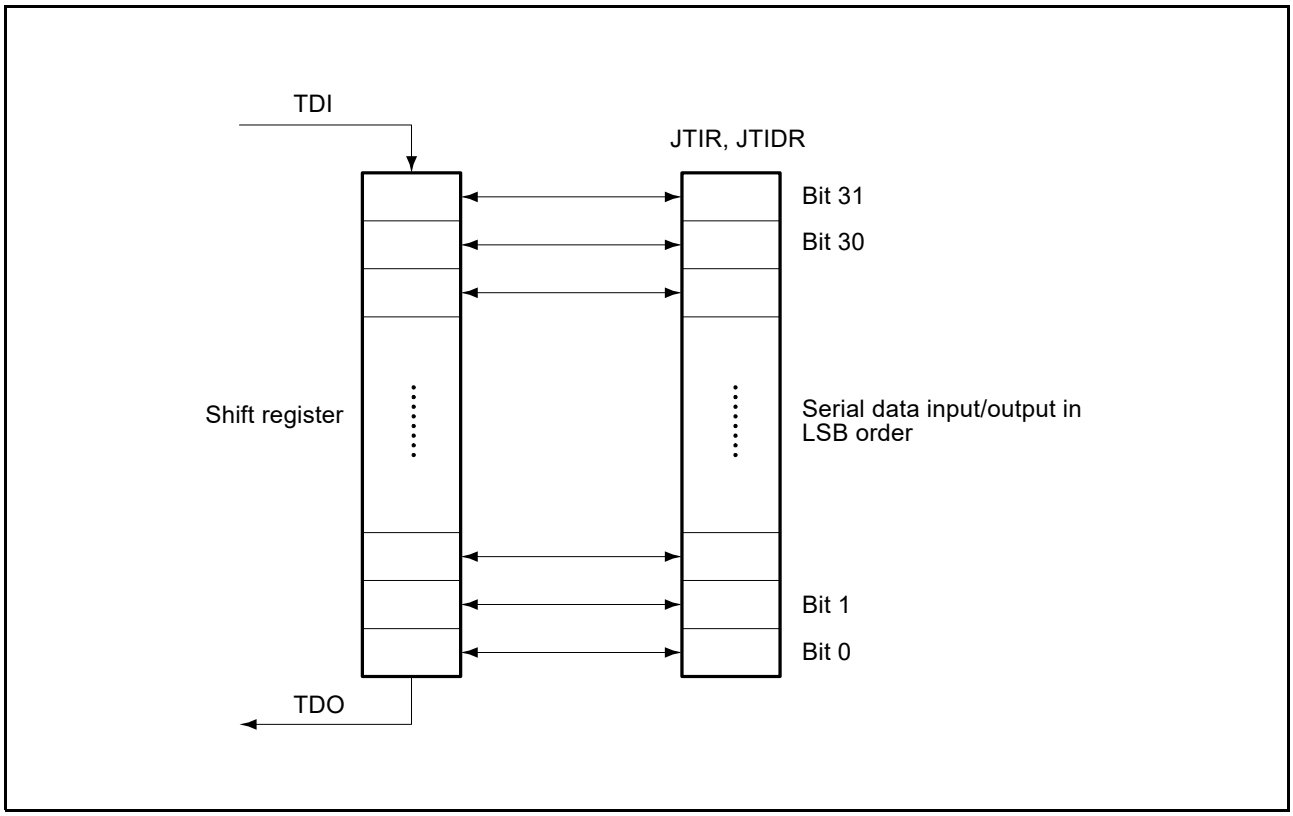


Figure 54.3 Serial Data Input/Output

(2) Power supply pins (VCC, VCC2, VCC_DCDC, VCL, VSS, VSS_DCDC, VLO, VBATT, AVCC0, AVSS0, VCC_USB,VCC_USBHS, AVCC_USBHS, VSS_USB, VSS_USBHS, and AVSS_USBHS) cannot be boundary-scanned.
(3) Analog reference pins (VREFH0, VREFL0) cannot be boundary-scanned.
(4) Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned.
(5) Reset signal (RES#) cannot be boundary-scanned.
(6) USB dedicated pins (USBHS_DP, USBHS_DM, USB0_DP, and USB0_DM) cannot be boundary-scanned.
(7) The boundary-scan pins (TCK, TMS, TDI, and TDO) cannot be boundary-scanned.

55. Arithmetic Unit for Trigonometric Functions (TFUv2)

This MCU has an arithmetic unit for trigonometric functions (TFUv2). TFUv2 has an upward compatible architecture with TFUv1 and supports fixed-point numbers as well as floating-point numbers.

The TFU has hardware processing functionality for sin, cos, atan, and hypot_k calculations that provide sinθ, cosθ, atan(y/x), and $\sqrt{x^2 + y^2}/k$, respectively. For the value of k, see section 55.3.1, Arithmetic Processing.

55.1 Overview

Table 55.1 lists the specifications of the TFU.

Table 55.1 TFU specifications

Item	Description				
Arithmetic Processing	Calculation of sine, cosine, arctangent, and hypotenuse • A sine and cosine can be simultaneously calculated. • An arctangent and hypotenuse can be simultaneously calculated.				
Range and Unit of Values	With the input and output of floating-point numbers	I/O		Range	Unit
	Calculating sine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radians
		Output	sinθ	–1.0 ≤ sinθ ≤ 1.0	—
	Calculating cosine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radians
		Output	cosθ	–1.0 ≤ cosθ ≤ 1.0	—
	Calculating arctangent	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	atan(y/x)	$-\pi \le \text{atan}(y/x) \le \pi$	radians
	Calculating hypotenuse	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	$\sqrt{x^2 + y^2}/k$	$0 \le \sqrt{x^2 + y^2}/k \le \infty$	—
	With the input and output of fixed-point numbers	I/O		Range	Unit
	Calculating sine	Input	Angle θ	–1.0 ≤ θ < 1.0	turns
				–4.0 ≤ θ < 4.0	radians
		Output	sinθ	–1.0 ≤ sinθ ≤ 1.0	—
	Calculating cosine	Input	Angle θ	–1.0 ≤ θ < 1.0	turns
				–4.0 ≤ θ < 4.0	radians
		Output	cosθ	–1.0 ≤ cosθ ≤ 1.0	—
	Calculating arctangent	Input	x and y coordinates	–1.0 ≤ x < 1.0	—
				–1.0 ≤ y < 1.0	
		Output	atan(y/x)	–0.5 ≤ atan(y/x) ≤ 0.5	turns
				$-\pi \le \text{atan}(y/x) \le \pi$	radians
	Calculating hypotenuse	Input	x and y coordinates	–1.0 ≤ x < 1.0	—
				–1.0 ≤ y < 1.0	
		Output	$\sqrt{x^2 + y^2}/k$	$0 \le \sqrt{x^2 + y^2}/k \le \sqrt{2}/k$	—
Data Type for Processing	Single-precision floating-point, fixed-point two's-complement				
Number of cycles for calculation	Sine: 5 Cosine: 5 Arctangent: 14 Hypotenuse: 14				
Protect Zone filter	Security attribute can be set.*2				

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.

Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

55.2 Register Descriptions

55.2.1 Fixed-Point Sincos Input and Output Setting Register (FXSCIOC)

Address(es): TFU.FXSCIOC 8100 1004h, TFU_NS.FXSCIOC 0100 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OF[1:0]		—	—	—	IUF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IUF	Input Unit and Format Setting	0: Unit of turns, Q1.31 format 1: Unit of radians, Q3.29 format	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	OF[1:0]	Output Format Setting	b5 b4 0 0: Q1.31 0 1: Q2.30 1 0: Q3.29 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set this register before starting a calculation. Operation cannot be guaranteed if writing to this register is attempted during a calculation.

IUF Bit (Input Unit and Format Setting)

The IUF bit is used to set the unit and format of the input values in fixed-point sincos operations.

OF[1:0] Bits (Output Format Setting)

The OF[1:0] bits are used to set the format of the output values in fixed-point sincos operations. Exactly expressing the value +1.0 is not possible in Q1.31 format. When the exact value of $\sin\theta$ or $\cos\theta$ should be +1.0 in Q1.31 format, the value read from the FXSCDT0 or FXSCDT1 register is $+1.0 - 2^{-31}$ instead.

55.2.2 Fixed-Point Atanhypot_k Input and Output Setting Register (FXATIOC)

Address(es): TFU.FXATIOC 8100 1005h, TFU_NS.FXATIOC 0100 1005h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OUF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	OUF	Output Unit and Format Setting	0: Unit of turns, Q1.31 format 1: Unit of radians, Q3.29 format	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set this register before starting a calculation. Operation cannot be guaranteed if writing to this register is attempted during a calculation.

OUF Bit (Output Unit and Format Setting)

The OUF bit is used to set the unit and format of the output values of atan calculations in fixed-point atanhypot_k operations. Note that the format of the output value of every hypot_k calculation simultaneous with an atan calculation is Q3.29 regardless of the value of this bit.

55.2.3 Trigonometric Status Register (TRGSTS)

Address(es): TFU.TRGSTS 8100 1008h, TFU_NS.TRGSTS 0100 1008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ERRF	BSYF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BSYF	Calculation in Progress Flag	0: No calculating 1: Calculating	R/W *1
b1	ERRF	Input Error Flag	0: No input error occurred 1: Input error occurred	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to this flag is invalid.

BSYF Flag (Calculation in Progress Flag)

The BSYF flag indicates whether a calculation is in progress. Writing to this flag is ignored.

[Setting condition]

- When calculation started

[Clearing condition]

- When calculation completed

ERRF Flag (Input Error Flag)

The ERRF flag indicates whether an input error occurred during a calculation. Follow the procedure for restoring the error flag described in section 55.4.2, How to Use Interrupts, to write to this register.

[Setting condition]

- When input error occurred*1 or this flag is set to 1

[Clearing condition]

- When the next calculation started or this flag is set to 0

Note 1. For details on input error, see section 55.3.5, Relationship Between Input and Output Values for Atan Operation.

55.2.4 Floating-Point Sincos Data Register 0 (FPSCDT0)

Address(es): TFU.FPSCDT0 8100 1010h, TFU_NS.FPSCDT0 0100 1010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPSCDT0 register is used to read the output value of cosθ in a floating-point sincos operation. For details, see Table 55.2. For how to use this register, see section 55.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a floating-point sincos operation. Writing to any other data register 0, that is, to the FPATDT0, FXSCDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

55.2.5 Floating-Point Sincos Data Register 1 (FPSCDT1)

Address(es): TFU.FPSCDT1 8100 1014h, TFU_NS.FPSCDT1 0100 1014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPSCDT1 register is used to input an angle value θ and to read the output value of sinθ in a floating-point sincos operation. For details, see Table 55.2. Writing to this register starts a floating-point sincos operation. For how to use this register, see section 55.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a floating-point sincos operation. Writing to any other data register 1, that is, to the FPATDT1, FXSCDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 55.2 Input/Output value of FPSCDT0 and FPSCDT1

Register	Input value	Output value
FPSCDT0	Not used	cosθ
FPSCDT1	Angleθ	sinθ

55.2.6 Floating-Point Atanhypot_k Data Register 0 (FPATDT0)

Address(es): TFU.FPATDT0 8100 1018h, TFU_NS.FPATDT0 0100 1018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPATDT0 register is used to input a coordinate value x and to read the output value of $\sqrt{x^2 + y^2}/k$ in a floating-point atanhypot_k operation. For details, see Table 55.3. For how to use this register, see section 55.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a floating-point atanhypot_k operation. Writing to any other data register 0, that is, to the FPSCDT0, FXSCDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

55.2.7 Floating-Point Atanhypot_k Data Register 1 (FPATDT1)

Address(es): TFU.FPATDT1 8100 101Ch, TFU_NS.FPATDT1 0100 101Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPATDT1 register is used to input a coordinate value y and to read the output value of atan(y/x) in a floating-point atanhypot_k operation. For details, see Table 55.3. Writing to this register starts a floating-point atanhypot_k operation. For how to use this register, see section 55.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a floating-point atanhypot_k operation. Writing to any other data register 1, that is, to the FPSCDT1, FXSCDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 55.3 Input/Output value of FPATDT0 and FPATDT1

Register	Input value	Output value
FPATDT0	Input coordinates x	$\sqrt{x^2 + y^2}/k$
FPATDT1	Input coordinates y	atan(y/x)

55.2.8 Fixed-Point Sincos Data Register 0 (FXSCDT0)

Address(es): TFU.FXSCDT0 8100 1020h, TFU_NS.FXSCDT0 0100 1020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXSCDT0 register is used to read the output value of cosθ in a fixed-point sincos operation. For details, see Table 55.4. For how to use this register, see section 55.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a fixed-point sincos operation. Writing to any other data register 0, that is, to the FPSCDT0, FPATDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

55.2.9 Fixed-Point Sincos Data Register 1 (FXSCDT1)

Address(es): TFU.FXSCDT1 8100 1024h, TFU_NS.FXSCDT1 0100 1024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXSCDT1 register is used to input an angle value θ and to read the output value of sinθ in a fixed-point sincos operation. For details, see Table 55.4. Writing to this register starts a fixed-point sincos operation. For how to use this register, see section 55.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a fixed-point sincos operation. Writing to any other data register 1, that is, to the FPSCDT1, FPATDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 55.4 Input/Output value of FXSCDT0 and FXSCDT1

Register	Input value	Unit and format of the input values	Output value	Output value formats
FXSCDT0	Not used	—	cosθ	Q1.31, Q2.30, or Q3.29 in accordance with the value of FXSCIOC.OF[1:0] bits
FXSCDT1	Angleθ	FXSCIOC.IUF = 0: turns, Q1.31 FXSCIOC.IUF = 1: radians, Q3.29	sinθ	

Exactly expressing the value +1.0 is not possible when the output value format is Q1.31. When the exact value of sinθ or cosθ should be +1.0 in Q1.31 format, the value read from the FXSCDT0 or FXSCDT1 register is $+1.0 - 2^{-31}$ instead.

55.2.10 Fixed-Point Atanhypot_k Data Register 0 (FXATDT0)

Address(es): TFU.FXATDT0 8100 1028h, TFU_NS.FXATDT0 0100 1028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXATDT0 register is used to input a coordinate value x and to read the output value of $\sqrt{x^2+y^2}/k$ in a fixed-point atanhypot_k operation. For details, see Table 55.5. For how to use this register, see section 55.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a fixed-point atanhypot_k operation. Writing to any other data register 0, that is, to the FPSCDT0, FPATDT0, or FXSCDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

55.2.11 Fixed-Point Atanhypot_k Data Register 1 (FXATDT1)

Address(es): TFU.FXATDT1 8100 102Ch, TFU_NS.FXATDT1 0100 102Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXATDT1 register is used to input a coordinate value y and to read the output value of atan(y/x) in a fixed-point atanhypot_k operation. For details, see Table 55.5. Writing to this register starts a fixed-point atanhypot_k operation. For how to use this register, see section 55.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a fixed-point atanhypot_k operation. Writing to any other data register 1, that is, to the FPSCDT1, FPATDT1, or FXSCDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 55.5 Input/Output value of FXATDT0 and FXATDT1

Register	Input value	Input value formats	Output value	Unit and format of the output values
FXATDT0	Input coordinates x	Q1.31	$\sqrt{x^2 + y^2}/k$	—, Q3.29
FXATDT1	Input coordinates y	Q1.31	atan(y/x)	FXATIOC.OUF = 0: turns, Q1.31 FXATIOC.OUF = 1: radians, Q3.29

55.2.12 Data Saving and Restoring Register 0 (DTSR0)

Address(es): TFU.DTSR0 8100 1030h, TFU_NS.DTSR0 0100 1030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTSR0 register is used to save and restore the data in data registers 0 that are FPSCDT0, FPATDT0, FXSCDT0, or FXATDT0 registers. The value read from this register is either that most recently written to any of data registers 0 or the latest result of arithmetic processing in any of data registers 0. Writing to this register makes the values of all data registers 0 the same as that written to this register.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For how to use this register, see section 55.4.2, How to Use Interrupts.

55.2.13 Data Saving and Restoring Register 1 (DTSR1)

Address(es): TFU.DTSR1 8100 1034h, TFU_NS.DTSR1 0100 1034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTSR1 register is used to save and restore the data in data registers 1 that are FPSCDT1, FPATDT1, FXSCDT1, or FXATDT1 registers. The value read from this register is either that most recently written to any of data registers 1 or the latest result of arithmetic processing in any of data registers 1. Writing to this register makes the values of all data registers 1 the same as that written to this register.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For how to use this register, see section 55.4.2, How to Use Interrupts.

55.2.14 TFU Security Attribution Register (TFUSAR)

Address(es): CPSCU.TFUSAR 8701 8140h, CPSCU_NS.TFUSAR 0701 8140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TFUSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TFUSA	TFU Security Attribution	0: Secure 1: Non-secure	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the PRCR_S.PRC4 bit.

This register is used to configure the security attribute of the TFU.

55.3 Operation

55.3.1 Arithmetic Processing

The trigonometric function unit has two arithmetic operations, the sincos operation and the atanhypot_k operation. Here, “sincos operation” means simultaneous sin and cos calculations and “atanhypot_k operation” means simultaneous atan and hypot_k calculations. For details, see Table 55.6.

Table 55.6 Arithmetic Processing

Operation	Input value	Output value
sincos	Angle value θ	cosθ and sinθ
atanhypot_k	Coordinates x and y	atan(y/x) and $\sqrt{x^2+y^2}/k$

Note: The value of scaling factor *k* is:

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1+2^{-2i}}} \fallingdotseq 0.607252935008881256169 4$$

Multiplying the output value of the TFU $\sqrt{x^2+y^2}/k$ by k provides $\sqrt{x^2+y^2}$.

55.3.2 Input and Output Value Formats

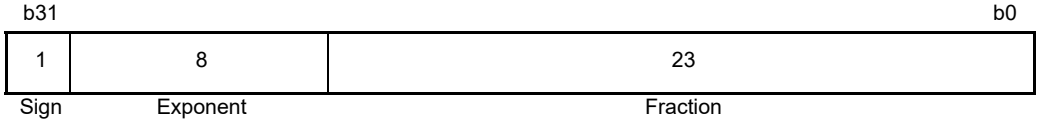
The TFU supports single-precision floating-point numbers and fixed-point numbers as its input and output values. For details on single-precision floating-point numbers and fixed-point numbers, see section 55.3.2.1, Single-Precision Floating-Point Numbers, and section 55.3.2.2, Fixed-Point Numbers, respectively. Note that the precision of fixed-point calculations is lower than 32 bits, although the input and output values are fixed-point numbers in 32-bit format.

55.3.2.1 Single-Precision Floating-Point Numbers

Floating-Point Number

The TFU supports the format for single-precision floating-point numbers specified in the IEEE754 standard.

- Single-Precision Floating-Point Number



Sign: sign field (1-bit)
Exponent: exponent field (8-bit)
Fraction: significand field (23-bit)

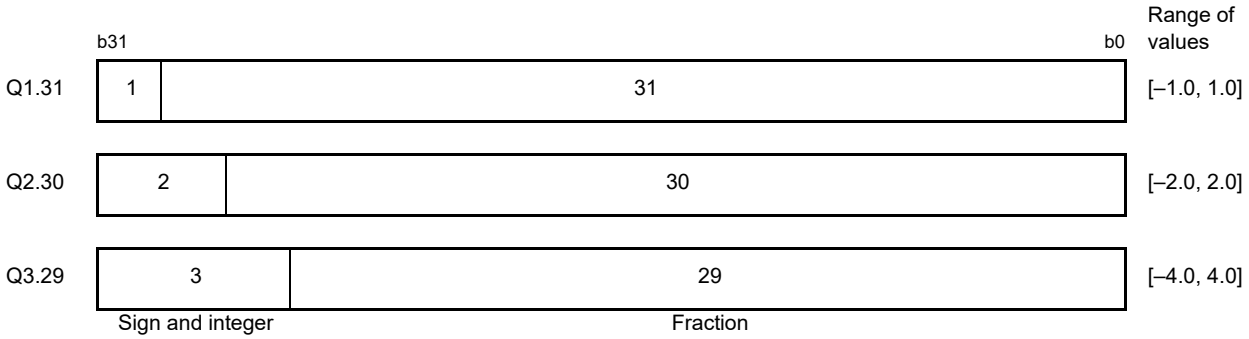
Numerical = $(-1)^S \times (1 + F \times 2^{-23}) \times 2^{(E-127)}$ (normalized value)
Numerical = $(-1)^S \times (0 + F \times 2^{-23}) \times 2^{-126}$ (denormalized value)

The TFU supports the numerical formats of single-precision floating-point numbers listed below.

S	E	F	Numerical
Any value	0 < E < 255	Any value	$(-1)^S \times 1.F \times 2^{(E-127)}$ (Normal Numbers)
Any value	E = 0	F > 0	$(-1)^S \times 0.F \times 2^{-126)}$ (Subnormal Numbers)
S = 0	E = 0	F = 0	$(-1)^0 \times 0.0$ (Positive zero: +0)
S = 1	E = 0	F = 0	$(-1)^{-1} \times 0.0$ (Negative zero: –0)
S = 0	E = 255	F = 0	Positive infinity: +∞
S = 1	E = 255	F = 0	Negative infinity: –∞
Any value	E = 255	400000h > F > 0	SNaN (Signaling Not a Number)
Any value	E = 255	F ≥ 400000h	QNaN (Quiet Not a Number)

55.3.2.2 Fixed-Point Numbers

The TFU supports the two’s-complement formats shown below.



55.3.3 Units of Angles

The TFU supports radians and turns as the units for angles. 0 to 2π radians and 0 to 1 turns respectively represent 0- to 360-degree rotations.

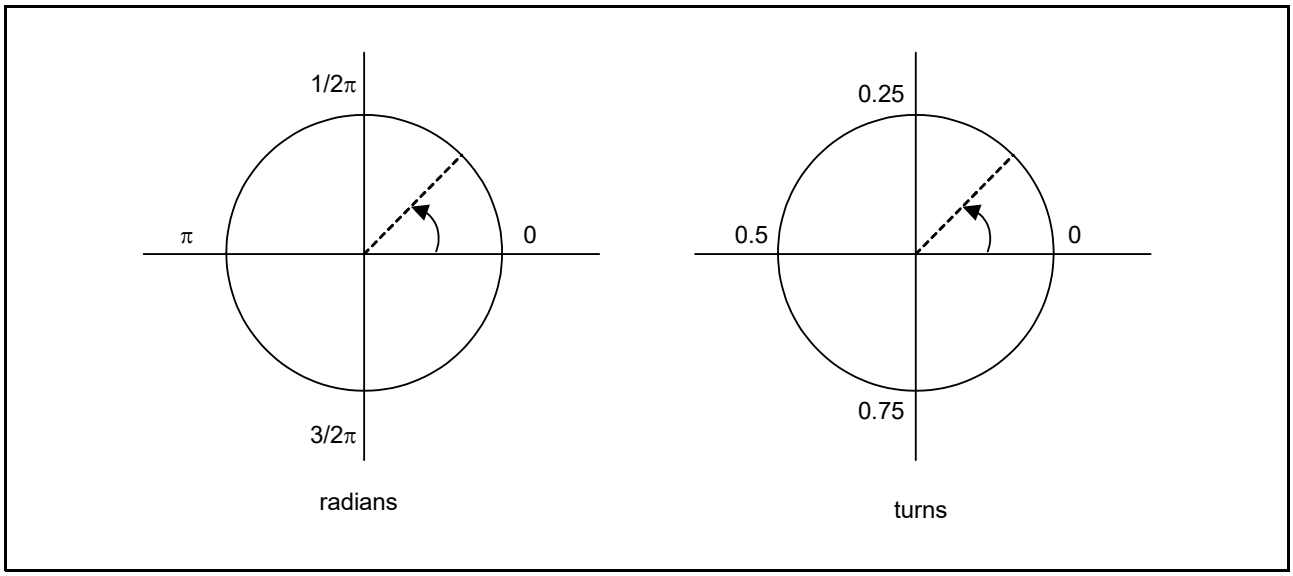


Figure 55.1 Units of Angles

55.3.4 Relationship Between Input and Output Values for Sincos Operation

The output value of a floating-point sin or cos calculation is specified as listed in Table 55.7 when the input value is ±0, ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number).

Table 55.7 Relationship Between Special Input Value and Its Output Value (for Sincos Operation)

Input (*θ*)	Output (cos)	Output (sin)
–∞	QNaN	QNaN
–0	+1	–0
+0	+1	+0
+∞	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN

Note: The output value of QNaN is FFC0 0000h.

55.3.5 Relationship Between Input and Output Values for Atan Operation

The output value of a floating-point atan calculation is specified as listed in Table 55.8 when either of the input values x or y is ±0, ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number). An input error is detected when both of the input values x and y are ±0.

Table 55.8 Relationship Between Special Input Value and Its Output Value (for Floating-Point Atan Operation)

y \ x	−∞	Negative value	−0	+0	Positive value	+∞	SNaN/QNaN
−∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
Negative value	QNaN	—	−π/2	−π/2	—	QNaN	QNaN
−0	QNaN	−π	QNaN*1	QNaN*1	−0	QNaN	QNaN
+0	QNaN	+π	QNaN*1	QNaN*1	+0	QNaN	QNaN
Positive value	QNaN	—	+π/2	+π/2	—	QNaN	QNaN
+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

An input error is detected when both of the input values x and y of a fixed-point atan calculation are 0, even for the values other than 0 in 32-bit fixed-point format that are handled as 0 by the TFU. The output value of an atan calculation is 0 when an input error has occurred.

55.3.6 Relationship Between Input and Output Values for hypot_k Operation

The output value of a floating-point hypot_k calculation is specified as listed in Table 55.9 in either of the following cases: when either of the input values x or y is ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number), or when both of the input values x and y are ±0. An input error is detected when both of the input values x and y are ±0. The reason is that such values cause errors in atan calculations although hypot_k calculations are possible.

Table 55.9 Relationship Between Special Input Value and Its Output Value (for hypot_k Operation)

y \ x	−∞	Negative value	−0	+0	Positive value	+∞	SNaN/QNaN
−∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
Negative value	QNaN	—	—	—	—	QNaN	QNaN
−0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
+0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
Positive value	QNaN	—	—	—	—	QNaN	QNaN
+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

An input error is detected when both of the input values x and y of a fixed-point hypot_k calculation are 0, even for the values other than 0 in 32-bit fixed-point format that are handled as 0 by the TFU. The reason is that such values cause errors in atan calculations although hypot_k calculations are possible. The output value of a hypot_k calculation is 0 when an input error has occurred.

55.4 Usage Procedures

55.4.1 Procedures for Using the TFU

There are two procedures for reading the result of an operation: reading the data register after the operation is completed (Procedure 1) and reading the data register without waiting for the operation to be completed (Procedure 2). Table 55.10 shows the advantages and disadvantages of each procedure. Figure 55.2 shows the procedures for sincos operations and Figure 55.3 shows the procedures for atanhypot_k operations. Read SCDTn (n = 0, 1) in Figure 55.2 as FPSCDTn and FXSCDTn in the case of floating-point sincos operations and fixed-point sincos operations, respectively. Read ATDTn in Figure 55.3 as FPATDTn and FXATDTn in the case of floating-point atanhypot_k operations and fixed-point atanhypot_k operations, respectively.

Table 55.10 Advantages and Disadvantages of Procedures for Sincos and Atanhypot_k Operations

Method	Advantages	Disadvantages
Procedure 1: reading the data register after the operation is completed	The bus is not occupied.	Detecting completion of a calculation by checking the TRGSTS.BSYF flag is required.
Procedure 2: reading the data register without waiting for the operation to be completed	Detecting completion of a calculation by checking the TRGSTS.BSYF flag is not required, resulting in fewer cycles being required for execution.	The bus is occupied until completion of the calculation.

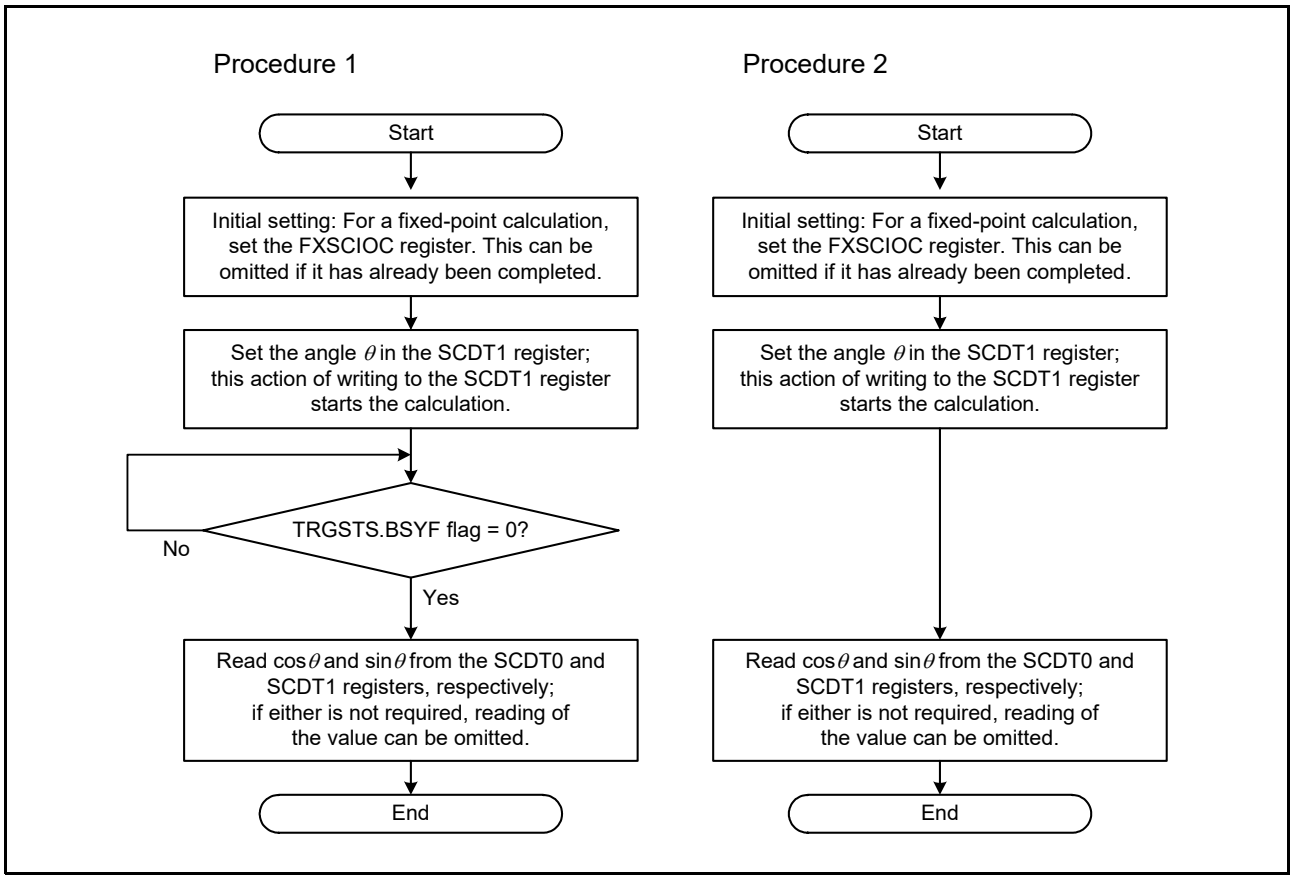


Figure 55.2 Procedure for Using TFU (Sincos Operation)

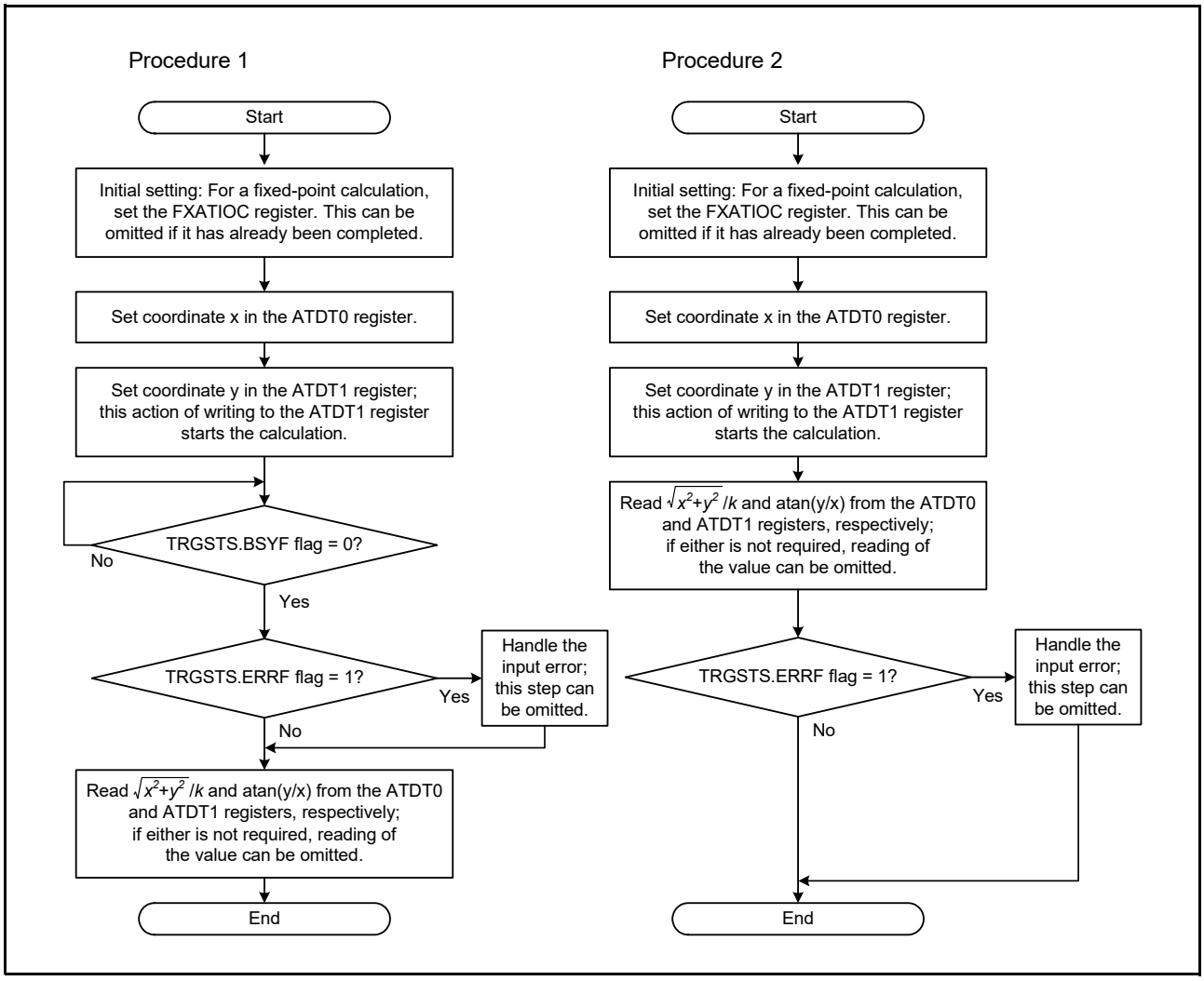


Figure 55.3 Procedure for Using TFU (Atanhypot_k Operation)

55.4.2 How to Use Interrupts

If an interrupt is accepted in the following order, the values stored in the data registers and the error flag prior to the acceptance of the interrupt may become incorrect in terms of the software after execution is returned from the interrupt handler.

1. The interrupt is accepted between writing to the data registers and reading of the data registers and the error flag during any of the procedures described in section 55.4.1, Procedures for Using the TFU.
2. The TFU is used to execute any calculation in the handler.

Accordingly, avoid such incorrect operation when, for example, using the TFU with multiple interrupts and their handlers by taking either of the following measures.

(1) Disabling interrupts that otherwise may disrupt processing between writing to the data registers and reading of the data registers and the error flag

(2) Saving and restoring data registers and the error flag in interrupt handlers that may disrupt processing*1, *2

Note 1. Regardless of the type of operation executed in the interrupt handler or the type of operation executed before accepting an interrupt, use the DTSR0, DTSR1, and TRGSTS registers to save and restore data registers and error flags. Figure 55.4 shows the save/restore procedure in an interrupt handler. By following this procedure, after returning from the interrupt handler, the processing that was being performed before accepting the interrupt can be continued without malfunctioning.

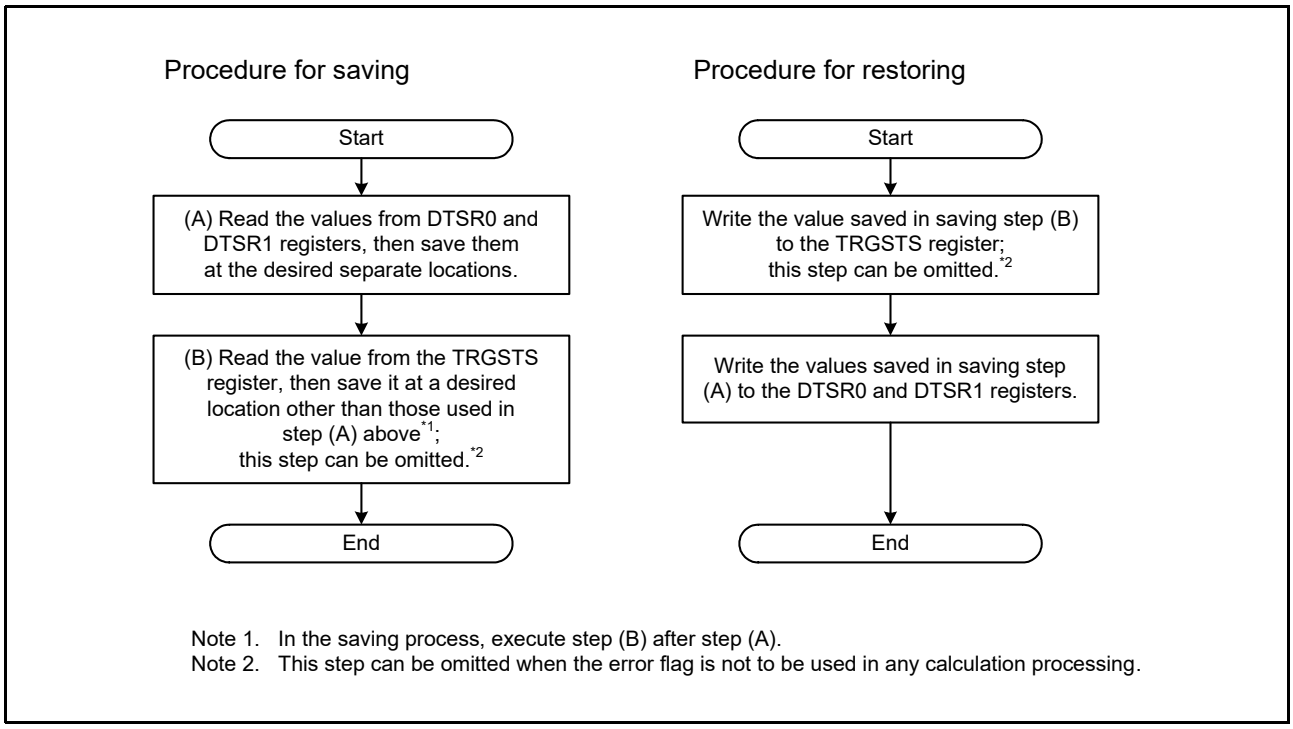


Figure 55.4 Procedures for Saving and Restoring in Interrupt Handlers

Note 2. As well as the data registers and the error flag, interrupt handlers that may change the values of the setting registers (FXSCIOC and FXATIOC) also require saving and restoring of the setting register values. Accordingly, save and restore the values of the setting registers as required. Saving of the setting registers must be executed after step (A) of the saving process in Figure 55.4.

56. Security Features

56.1 Features

- Protect Zone security
  - CPU core
    - Security technology based on a three-world structure
  - Bus access and address space
    - Divides slave resources into secure and non-secure.
  - Implements the Security Attribution Unit (SAU).
    - Divides the address space into secure and non-secure.
    - Manages security attributes and detects security violations for CPU bus access.
    - Implements the Master Security Attribution Unit (MSAU) (SAU for bus masters other than the CPU).
  - Interrupts
    - The destination can be configured to be secure or non-secure for each interrupt vector.
  - Code MRAM
    - Up to two regions (secure/non-secure) in linear mode
    - Up to four regions (two secure regions and two non-secure regions) in dual mode
  - Data MRAM
    - Up to two regions (secure/non-secure)
  - SRAM
    - Up to two regions (secure/non-secure) for each SRAM
  - Backup Registers
    - Up to two regions (secure/non-secure)
  - Peripherals
    - Security attributes can be set individually for each unit/channel.
- Device lifecycle management
- Three debug levels
  - AL2: Non-secure and secure debug functions are enabled and accessible from the debugger.
  - AL1: Only the non-secure debug function is enabled, and the debugger can access only defined non-secure debug accessible regions.
  - AL0: No debug functions are available.
- Secure key injection
- Secure factory programming
  - Supports programming of on-chip MRAM in ciphertext format.
- Secure boot

  Supports up to four expirable Roots of Trust (RoTs) with OTP. This is the SHA-256 hash digest of the RoT public key.
  - Verification of the integrity and authenticity of the image to be programmed
  - Before executing the image, the immutable (OTP) first-stage bootloader (FSBL) and user boot loader verify the integrity and authenticity of the executable image.
- Cryptographic accelerator
  - See section 57, Renesas Secure IP (RSIP-E50D).

- Secure pin multiplexing
  - All I/O ports can be individually configured as secure or non-secure.
  - The peripheral pin function is valid when the security attributes of the peripheral function and I/O port match. See section 22, I/O Ports.
- Decryption on the fly
  - The circuits for decryption on the fly (DOTFs) decrypt encrypted contents of external memory in real time. See section 49, Decryption On The Fly (DOTF).

Note: The external address spaces do not have Protect Zone filters. For this reason, even if these spaces are specified as secure with the settings by the SAU, they are accessible by non-secure masters other than the CPU.

56.2 Detection of Tampering

This device has I/O ports that are capable of detecting external tampering.
For tamper events

- Time stamp processing
  - The RTC can be used to obtain time stamps, record events, and generate interrupts. See section 30, Realtime Clock (RTCBb).
- Hardware unique key
  - Zeroization (setting 0) is applied to the hardware unique key (HUK) following the detection of a tamper event. See section 65, MRAM.
- Backup registers
  - Zeroization (setting 0) is applied to the backup registers following the detection of a tamper event. See section 12, Battery Backup Function (VBATTBa).

56.3 Protect Zone Feature

56.3.1 Protect Zone Technology

The RXv3e CPU core has three security states called worlds. These are called root world, secure world, and non-secure world, respectively. Programs executed in the secure world and the non-secure world, as well as the resources they can access, can be separated. A program cannot be moved directly between the secure world and non-secure world; it can be moved only via the root world. Some resources of the CPU core, such as control registers, are separated by the CPU core's hardware, but other resources are separated by a program in the root world.
The world after a reset is the root world.

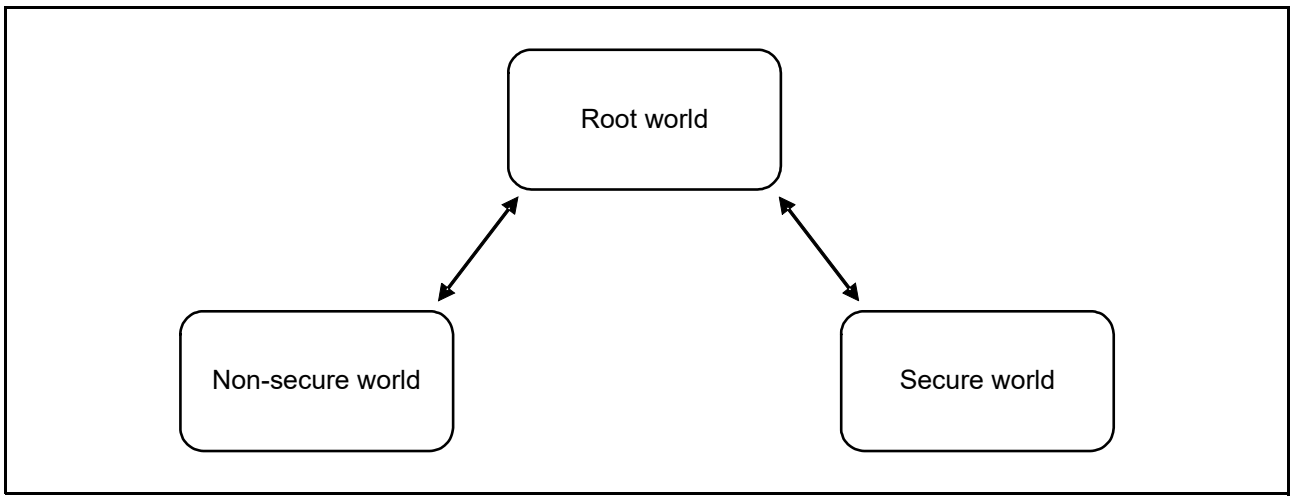


Figure 56.1 Three Worlds of the Protect Zone

56.3.2 Flat-Project and Non-Flat-Project

Running programs is also possible without taking the other worlds into account. Such software projects are called Flat-Projects*1. A Flat-Project is only used in the root world and programs can be executed without taking the other worlds into account.
In contrast with Flat-Projects, all software projects that use the Protect Zone's worlds are called a Non-Flat-Project. A Non-Flat Project consists of multiple software projects. These are software projects for the root and secure worlds and software projects for the non-secure world.
Table 56.1 lists the functions that require operation switching depending on whether software is developed as a Flat-Project or a Non-Flat-Projec. These operations can be switched using the CPSCU.PRTZCR register and the SAU.AAS register. The operation for a Flat-Project is selected after release from the reset state. To use a Non-Flat-Project, appropriately set the registers so that the operation for a Non-Flat-Project proceeds.

Note 1. This term is used in the descriptions of development tools provided by Renesas.

Table 56.1 Register Settings for Operation for a Flat-Project and Operation for a Non-Flat-Project

Item	Register Setting	Description
Operation for Flat-Project	CPSCU.PRTZCR.PRTZC = 0	The destination world for all interrupts and non-maskable interrupts is the root world.
	SAU.AAS.NS = 0	Instruction fetch from the non-secure address space is enabled in the root world and secure world.
Operation for Non-Flat-Project	CPSCU.PRTZCR.PRTZC = 1	The destination world for interrupts can be set to either the secure world or non-secure world for each vector. The destination world for non-maskable interrupts can also be set to either the secure world or non-secure world.
	SAU.AAS.NS = 1	Instruction fetch from the non-secure address space is disabled in the root world and secure world and generates an exception.

56.3.3 Relationship Between Interrupts and Worlds

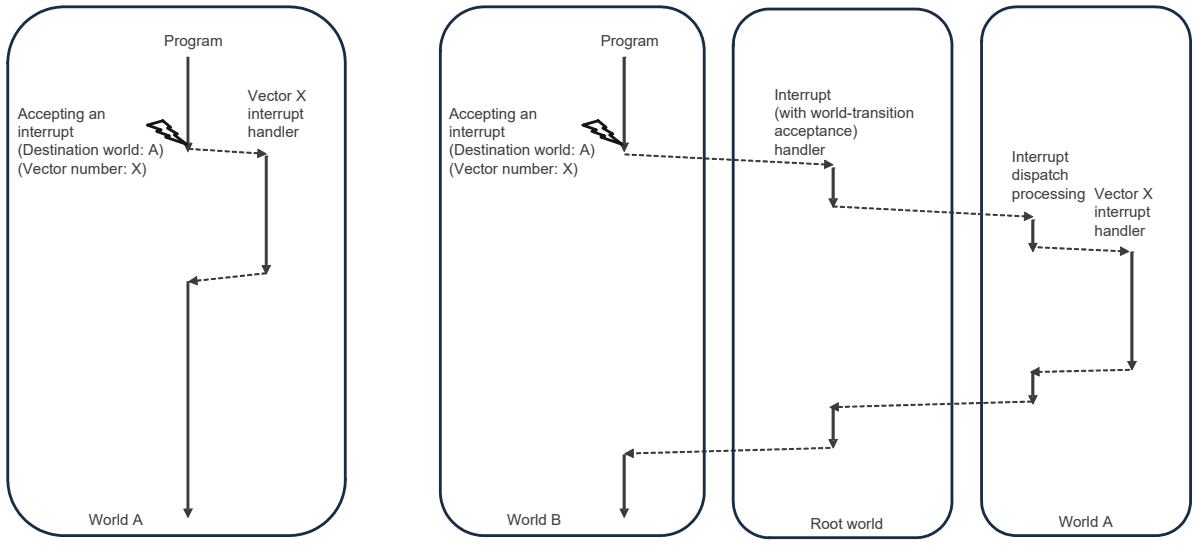
An interrupt notified from the ICU to the CPU contains information about the world that is the destination of the interrupt. Interrupt acceptance is determined based on the CPU status (world where the program is being executed, PSW.IPL, etc.) and interrupt information from the ICU (interrupt priority level and interrupt destination world). For details on acceptance conditions, see section 2, CPU.

An interrupt vector table exists independently for each world.

When an interrupt is accepted, the following occurs.

If the world where the program is being executed is the same as the interrupt destination world, the CPU references the vector table within that world and moves on to the interrupt handler processing (Figure 56.2 (a)).

If the world where the program is being executed is different from the interrupt destination world, the CPU moves on to the interrupt (with world-transition acceptance) handler of the root world. The handler in the root world calls the interrupt dispatch processing of the destination world. Interrupts can be handled by referencing the vector table within the interrupt dispatch processing and calling the interrupt handler (Figure 56.2 (b)). Note that the handler in the root world and the interrupt dispatch processing software must be prepared in advance. The hardware performs only the minimum necessary processing. The handler in the root world should perform operations such as saving resources that need to be hidden or protected between worlds.



(a) When the world where the program is being executed and the interrupt destination world are the same

(b) When the world where the program is being executed and the interrupt destination world are different

Figure 56.2 Operation When Accepting an Interrupt

56.3.3.1 When Set for Operation for Non-Flat-Project

For each vector, the secure or non-secure world can be selected as the interrupt destination world. The secure or non-secure world can be selected as the destination world for non-maskable interrupts. The root world cannot be selected as the destination.
The interrupt priority level for interrupts destined for the secure world can be set from 0 to 31. The interrupt priority level for interrupts destined for the non-secure world can be set from 0 to 15.

56.3.3.2 When Set for Operation for Flat-Project

The destination world for all interrupts and non-maskable interrupts is the root world.
The interrupt priority level for interrupts can be set from 0 to 31.

56.3.4 Protect Zone Initial State

In the initial state after resetting, operation starts in the root world.
The reset vector and the initial value of the EXTB register are changed to match the program execution start area.

Table 56.2 Reset Vector and the Initial Value of the EXTB Register

Program Execution Start Area	Reset Vector	EXTB Initial Value
User area	FFFF FFFCh	FFFF FF80h
User extra MRAM area	86D8 007Ch	86D8 0000h

All spaces in the root world become accessible for instruction fetching and data access.

56.4 Bus Access and Address Space

The bus master (CPU, DMAC, etc.) has security attributes as information for issuing bus transactions, and is classified into “secure bus master” and “non-secure bus master” depending on the transaction to be issued. The security attributes of the CPU are determined by the world that executes the bus transaction. Security attributes for bus masters other than the CPU are configurable. For resources with selectable security attributes, the default setting after a reset is “secure”.
The address space accessible to the bus master is broadly divided into “secure access region” and “non-secure access region.” Bit 31 of the address determines which region is accessible. The SAU (Security Attribution Unit) associated with the CPU or the SAU (MSAU) of a bus master other than the CPU determines whether access is permitted based on the security attributes of the transaction issued by the bus master and the address of the access destination, and then determines the security attributes of the transaction to be issued to the bus slave. Except for some regions such as external bus regions, a bus slave is mapped to both the secure access region and non-secure access region.
Bus-accessible resources (slave resources) such as memory and registers are classified as either “secure” or “non-secure”. As a general rule, secure resources can only be accessed by secure transactions, and non-secure resources can only be accessed by non-secure transactions. The PZF (Protect Zone Filter) associated with each slave module transmits only access-enabled transactions to the slave. Note that there are some exceptional resources that can be accessed by both secure and non-secure transactions, but the PZF also determines whether such resources can be accessed.
The security attributes (assignment to secure/non-secure) of slave resources are either fixed in advance as hardware specifications or selectable through settings. In addition, some read-only slave resources are not subject to access control by security attributes. For resources with selectable security attributes, the default setting after a reset is “secure”. For details, see the descriptions of the peripheral security attribution registers (PSARx, x = B to F) in this section and the security attribution registers in other sections.

Figure 56.3 shows how transactions are processed from the master to the slave according to security attributes. There are two types of processing methods in the SAU for the secure bus master. See Table 56.3 for the security attributes and types of each bus master.
Also, for some register resources, read access is not subject to access control, while write access is subject to security access control, meaning that read and write access are handled differently. In addition, whether or not a Protect Zone access error is generated when an access violation is detected by the PZF depends on the register. For details, see the description for each register.

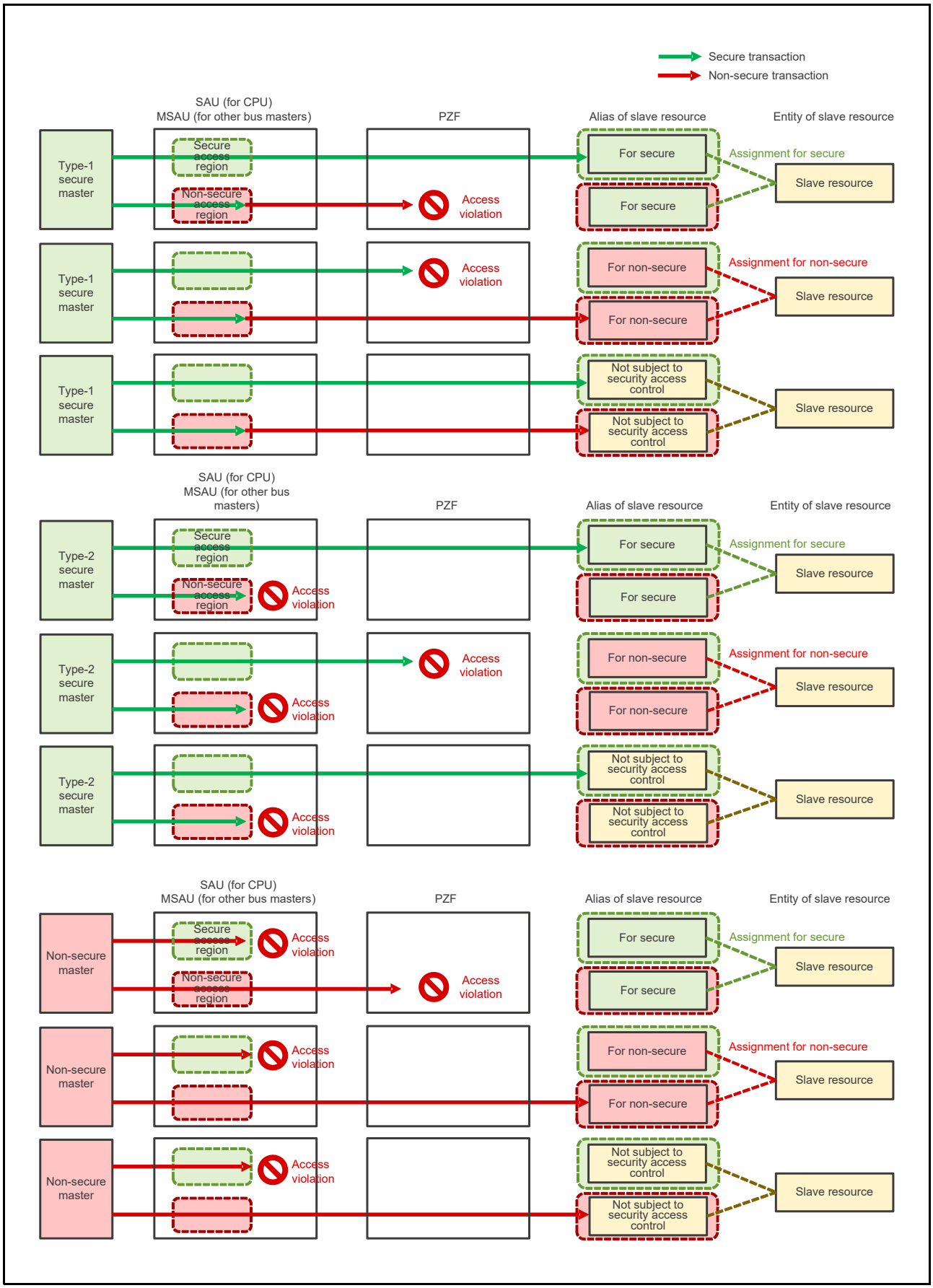


Figure 56.3 Processing of Transactions from Each Master

Table 56.3 Security Attributes of Each Bus Master

Bus Master	Security Attribute
CPU (root world) operand access	Secure*1
CPU (root world) instruction fetch	Secure*2
CPU (secure world) operand access	Secure*1
CPU (secure world) instruction fetch	Secure*2
CPU (non-secure world) operand access	Non-secure
CPU (non-secure world) instruction fetch	Non-secure
DMAC	Select secure*1/non-secure for each channel.
DTC	Select secure*1/non-secure.
EXDMAC	Select secure*1/non-secure for each channel.
XSPI0/XSPI1	Select secure*1/non-secure for each channel.

Note 1. Type-1 secure
Note 2. Type-1 secure when operation for a Flat-Project is selected. Type-2 secure when operation for a Non-Flat-Project is selected.

56.5 Security Attribution Unit (SAU)

This MCU has a built-in address-space Security Attribution Unit (SAU) that manages security attributes for CPU bus access.
The SAU manages security attributes by generating secure exceptions based on the CPU access type and the security attributes of the access target. All attribute management is effective regardless of the CPU's secure/non-secure state or user/supervisor mode.
Table 56.4 shows the specifications for the Security Attribution Unit.

Table 56.4 Specifications for the Security Attribution Unit (SAU)

Specification	Description
Address-space security attribute	The address space has the following fixed security attributes: 0000 0000h to 7FFF FFFFh: Non-secure attribute 8000 0000h to FFFF FFFFh: Secure attribute
Secure violation detection function	Detects violations by comparing CPU access types and address space security attributes. (secure exception generation) In the case of operand access, stores the address that caused the violation in the secure exception address register (SEA). It also stores the cause in the secure exception status register (SESTS).
Transaction issuance function	If no violations are detected, issue a transaction with attributes corresponding to the address-space security attributes.

56.5.1 Types of CPU Access

The CPU has security attributes of secure (root world, secure world) and non-secure (non-secure world) corresponding to the world where the program is currently being executed. In order to perform instruction access and data access in each state, there are four types of CPU access: "instruction access (secure)", "data access (secure)", "instruction access (non-secure)", and "data access (non-secure)".

56.5.2 Address-Space Security Attribute

Each address has two security attributes: Secure space and non-secure space. The security attributes for addresses 0000 0000h to 7FFF FFFFh are fixed as non-secure space, and addresses 8000 0000h to FFFF FFFFh are fixed as secure space.

56.5.3 Transaction Types Issued

For CPU access, a secure transaction or a non-secure transaction is issued in accordance with the address-space security attribute. In addition, cache attributes are added to each issued transaction based on the MPU and background cache settings.

56.5.4 Access Across Different Security Attribute Spaces

If access occurs across different security attribute spaces, a secure violation is detected.

56.5.5 Functions

56.5.5.1 Secure Memory Protection Function

The secure memory protection function monitors whether the program in the secure or non-secure state is only accessing what is permitted in each state. If an access violation (secure memory protection error) is detected, the SAU notifies the CPU of this information, and the CPU starts secure exception handling.
The secure memory protection function is always enabled after a reset.
If a violation is detected, the corresponding transaction will not occur and a secure exception will occur. If no violations are detected, transactions are issued according to the security attributes of the address space being accessed.

Table 56.5 Secure Memory Protection and Issued Transactions

Types of CPU Access	Security Attribute of the Address Space to Be Accessed	Violation Detection (Exception Occurs)	Issued Transaction
Instruction access (secure)	Secure	Not detected	Secure
	Non-secure	Detected (secure exception)*1	Not detected
Instruction access (non-secure)	Secure	Detected (secure exception)	Not detected
	Non-secure	Not detected	Non-secure
Data access (secure)	Secure	Not detected	Secure
	Non-secure	Not detected	Non-secure
Data access (non-secure)	Secure	Detected (secure exception)	Not detected
	Non-secure	Not detected	Non-secure

Note 1. This is the behavior when operation for a Non-Flat-Project is selected (AAS.NS = 1). When operation for a Flat-Project is selected (AAS.NS = 0), violation detection = none and issued transaction = non-secure in the same way as data access (secure).

56.5.5.2 Protection of Security Attribution Management Registers

The AAS, SESTS, SECLR, and SEA registers cannot be accessed by any means other than CPU operand access in secure state and Supervisor mode (instruction fetch, DMA). When the user-mode CPU accesses an operand, a data memory protection error occurs regardless of whether memory protection is enabled. If the CPU attempts to access data in non-secure state and supervisor mode, a secure exception occurs.

56.5.5.3 Access Determination Flow for the Secure Memory Protection Function

Figure 56.4 shows the instruction access determination flow when operation for a Non-Flat-Project is selected (AAS.NS = 1). When operation for a Flat-Project is selected (AAS.NS = 0), secure access to the non-secure space is possible in the same way as the data access determination flow shown in Figure 56.5.

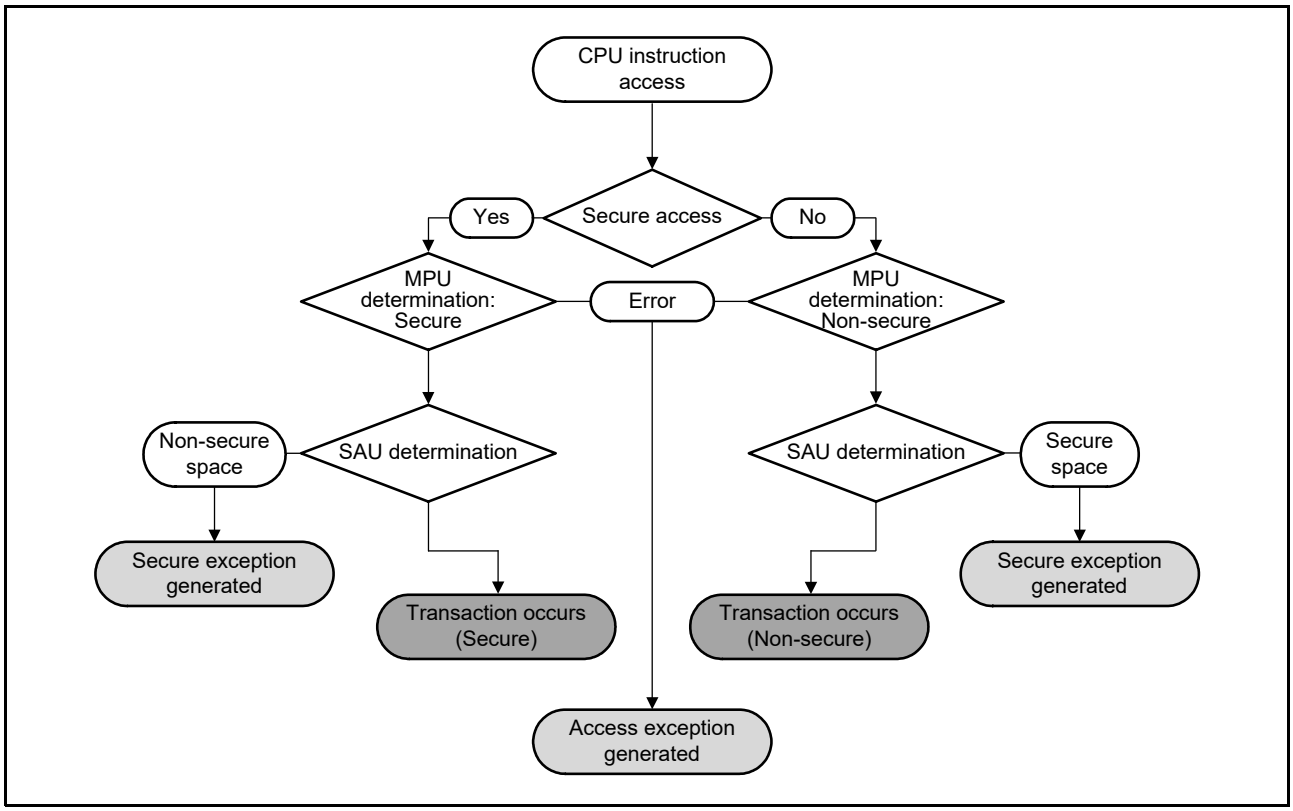


Figure 56.4 Access Determination Flow During Instruction Access (AAS.NS = 1)

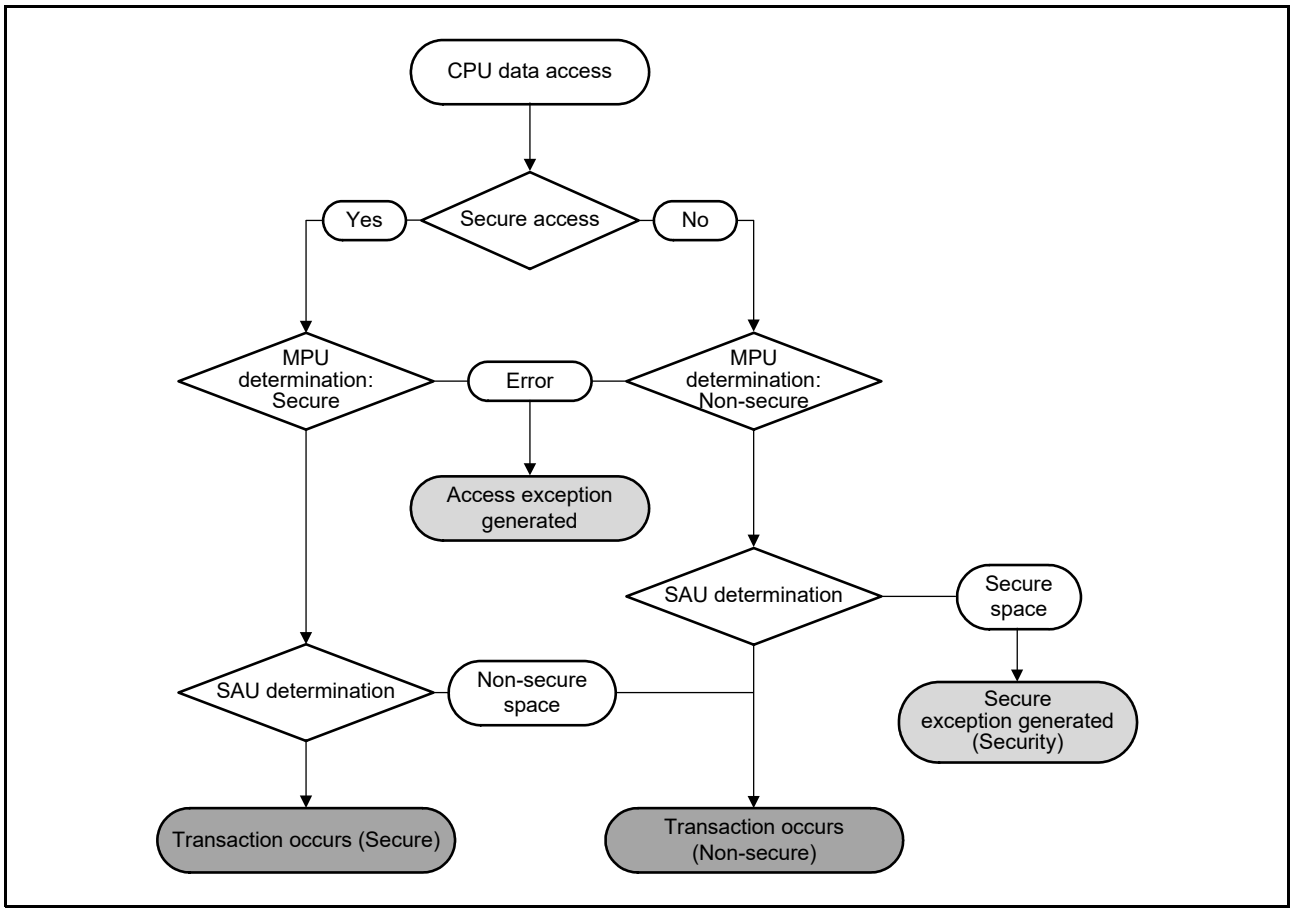


Figure 56.5 Access Determination Flow During Data Access

56.5.6 Instructions for Use

56.5.6.1 Handling Secure Exceptions

When a secure memory protection error is detected, the CPU starts secure exception handling. In addition, the cause of the secure exception is stored in the secure exception status register (SESTS), and in the case of a secure memory protection error caused by operand access, the operand access address that caused the exception is stored in the secure exception address register (SEA). For details on CPU operations for secure exception handling, see section 14, Exception Handling.

56.6 Memory Security Attributes

Memory is divided into secure and non-secure regions. The memory security attributes of code MRAM and data MRAM are stored into nonvolatile memory by a boot command when the device lifecycle is in the OEM state and the authentication level is AL2. These memory security attributes are applied before application execution. They cannot be updated by the application but are readable using dedicated registers. The memory security attributes of SRAM are set by a dedicated security attribution register writable only by secure access.

The code MRAM can be divided into up to two regions in linear mode and four regions in dual mode. The partitioning is the same between bank0 and bank1 in the dual mode. The data MRAM can be divided into up to two regions. The SRAM can be divided into up to two regions.

Figure 56.6 shows the memory mapping. Table 56.6 shows the sizes of the memory regions and Table 56.7 shows access permission of the memory regions.

It is prohibited to place secure regions in a block swappable area in linear mode because the secure application would be placed in the non-secure region after block swapping.

The contents in secure regions must be the same in both bank0 and bank1 in the dual mode. If they are not the same, the contents of the secure regions or non-secure regions may not match after the firmware update in the field. See section 56.13, Field Updating in Dual Mode.

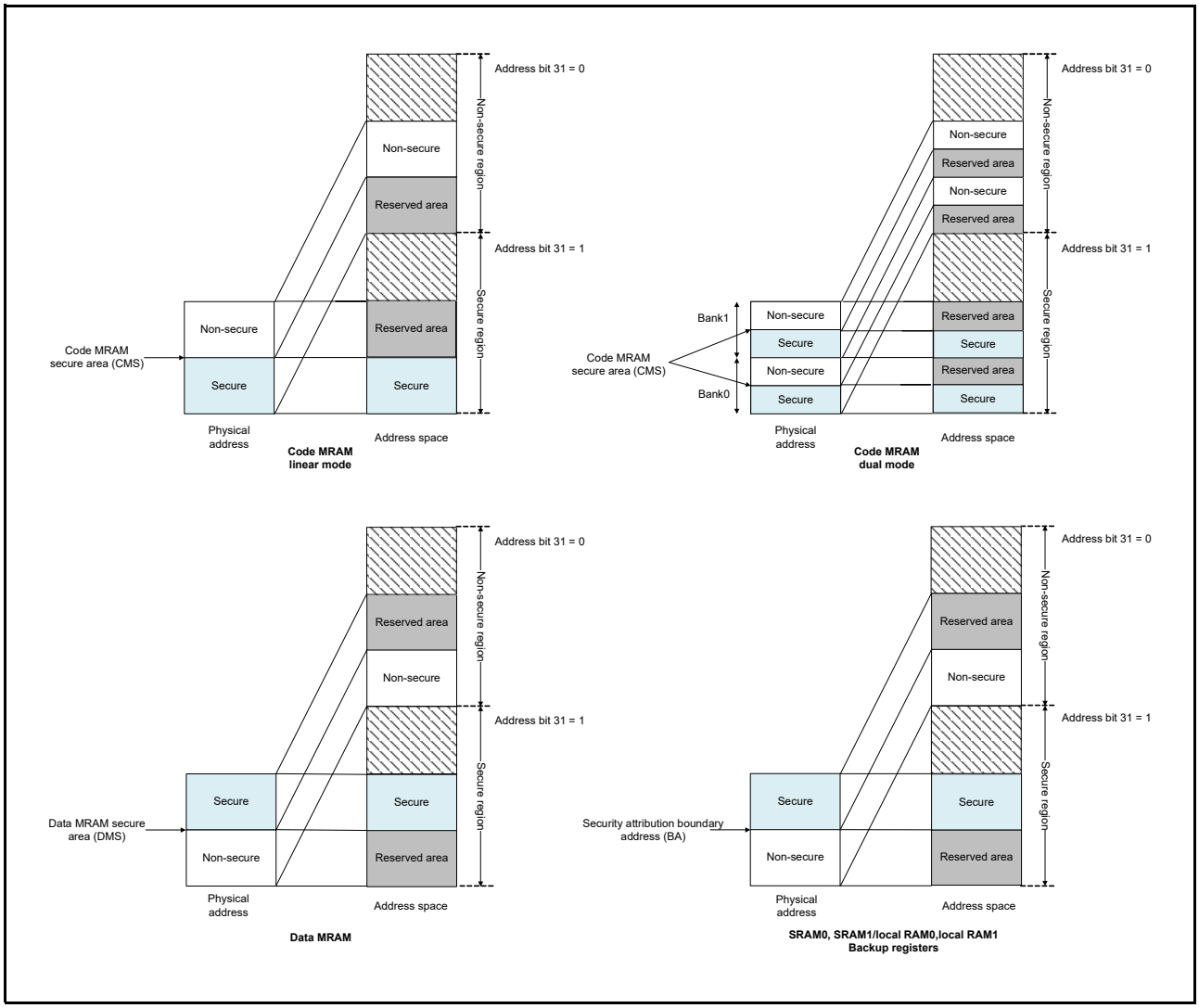


Figure 56.6 Memory Mapping

Table 56.6 Sizes of Memory Regions

Memory Region		Start Address	Size
Linear mode	Code MRAM secure	FFF0 0000h + ((20h – CMS) × 32 KB)	CMS × 32 KB
	Code MRAM non-secure	7FF0 0000h	((20h – CMS) × 32 KB)
Dual mode	Code MRAM bank0 secure	FFF8 0000h + ((10h – CMS) × 32 KB)	CMS × 32 KB
	Code MRAM bank0 non-secure	7FF8 0000h	((10h – CMS) × 32 KB)
	Code MRAM bank1 secure	FFB8 0000h + ((10h – CMS) × 32 KB)	CMS × 32 KB
	Code MRAM bank1 non-secure	7FB8 0000h	((10h – CMS) × 32 KB)
Data MRAM secure		8690 0000h	DMS × 2 KB
Data MRAM non-secure		0690 0000h + (DMS × 2 KB)	(1Bh – DMS) × 2 KB
SRAM0 secure		8200 0000h	BA*1
SRAM0 non-secure		0200 0000h + BA*1	SRAM0 size – BA*1
SRAM1 secure		8210 0000h	BA*1
SRAM1 non-secure		0210 0000h + BA*1	SRAM1 size – BA*1
Local RAM0 secure		8000 0000h + BA*2: Access address from CPU 8280 0000h + BA*2: Access address from bus masters other than CPU	BA*2
Local RAM0 non-secure		0000 0000h + BA*2: Access address from CPU 0280 0000h + BA*2: Access address from bus masters other than CPU	Local RAM0 size – BA*2
Local RAM1 secure		8002 0000h + BA*2: Access address from CPU 8282 0000h + BA*2: Access address from bus masters other than CPU	BA*2
Local RAM1 non-secure		0002 0000h + BA*2: Access address from CPU 0282 0000h + BA*2: Access address from bus masters other than CPU	Local RAM1 size – BA*2
Backup registers secure		8701 9D00h	BA*3
Backup registers non-secure		8701 0000h + BA*3	Backup register size – BA*3

Note 1. Boundary address specified by CPSCU.SRAMSABARn (n = 0, 1)
Note 2. Boundary address specified by CPSCU.LRMSABARn (n = 0, 1)
Note 3. Boundary address specified by SYSC.BKRSABR

Table 56.7 Access Permission of Memory Regions

Memory Region	Secure Transaction	Non-secure Transaction
Each memory region configured as secure	Permitted	Write ignored/Read ignored Protect Zone access error is generated.
Each memory region configured as non-secure	Write ignored/Read ignored Protect Zone access error is generated.	Permitted

56.7 Security Attribution of Peripherals

Each peripheral can be set to secure or non-secure by configuring the security attribution register. Peripherals are classified into two types based on the registers to be used for security attribution. Type-1 peripherals have one security attribute. Access to all registers is controlled by one security attribute. Type-1 peripheral security attributes are set in the PSARx (x = B to F) register by the secure application. Type-2 peripherals have a security attribute for each register or for a bit. Among the type-2 peripherals, the TFU only has one security attribute for the module. Access to each register or bit field is controlled according to these security attributes. Type-2 peripheral security attributes are set in the security attribution register in each module by the secure application. For the security attribution register, see sections in the user's manual for each peripheral. Table 56.8 shows the classification of peripheral types.

Table 56.8 Classification of Peripheral Types

Type	Peripheral and CPU Acceleration Function
1	RSCI, RSPIA, XSPI, DOTF, USBHS, USB, RIICHS, RI3C, RSIP-E50D, ESWM, CANFDG, CANFD, PDMIF, DOC, CRC, CAC, TEMPS, ADC16H, DAC12R, POEG, HRPWM, GPTW, RTC, IWDT, WDT, PDC, TMR, CMTW, CMT, PIO, PCIF, SSIE, SDHI, ETHPHYCLK
2	System control (Resets, PVD, Clock Generation Circuit, Low Power Consumption Function, Battery Backup Function, Register Write Protection Function), CPU acceleration function (IFU and caches), MRAM Controller, SRAM Controller, Local RAM Controller, DMAC, DTC, EXDMAC, ICU, MPU, BUS, Security Setting, ELC, TFU, and I/O Ports

Table 56.9 shows the access permission of type-1 peripherals. The access permissions of type-2 peripherals vary depending on the peripheral. See the register description for each peripheral.

Table 56.9 Access Permissions of Type-1 Peripherals

	Secure Access	Non-secure Access
Peripheral configured as secure	Permitted	Write ignored/Read ignored Protect Zone access error is generated.
Peripheral configured as non-secure	Write ignored/Read ignored Protect Zone access error is generated.	Permitted

56.8 Protect Zone Access Error

Table 56.10 shows the operation when a Protect Zone access error occurs. The operation varies depending on the master. This operation does not apply to bufferable write access. For more information on the operations when a bufferable write access error is detected, see section 16.8.2, Operations When a Bus Error Occurs.

Table 56.10 Operations When a Protect Zone Access Error Occurs

CPU	DMAC/DTC/EXDMAC	ESWM
• An instruction secure exception is generated for an error in instruction fetch.*1 • An interrupt or reset*2 can be issued.	• Transfer is stopped. • An interrupt or reset*2 can be issued.	• An interrupt or reset*2 can be issued.

Note 1. A secure exception is not generated for an error in operand access.
Note 2. The operation after error detection is selected by the BUSOAD.SLERROAD bit.

56.9 Register Descriptions

56.9.1 Secure Address-Space Attribute Setting Register (AAS)

Address(es): SAU.AAS 8100 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NS	Non-secure Address Space Setting	0: Instruction access (secure) to 0000 0000h to 7FFF FFFFh is enabled. 1: Instruction access (secure) to 0000 0000h to 7FFF FFFFh is disabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

NS Bit (Non-secure Address Space Setting)

This bit disables subsequent instruction access (secure) to the non-secure address space (0000 0000h to 7FFF FFFFh) after the CPU is shifted from a secure state (operation for a Flat-Project is selected) to a mixed secure/non-secure state (operation for a Non-Flat Project is selected). When this bit is set to 1, instruction access (secure) to 0000 0000h to 7FFF FFFFh cannot be executed, and a secure exception occurs.

It is initialized to 0 when reset, and can be written to 1, but cannot be cleared afterwards.

56.9.2 Secure Exception Status Register (SESTS)

Address(es): SAU.SESTS 8100 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSER	ISER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ISER	Instruction Access Secure Exception Occurrence Flag	0: No secure exceptions occurred by instruction access. 1: A secure exception occurred by instruction access.	R
b1	DSER	Data Access Secure Exception Occurrence Flag	0: No secure exceptions occurred by data access. 1: A secure exception occurred by data access.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

ISER Flag (Instruction Access Secure Exception Occurrence Flag)

This flag indicates that a secure exception has occurred due to instruction execution.

The ISER flag is set to 0 only when the exception status clear bit (CLR) of the secure exception status clear register (SECLR) is set to 1.

DSER Flag (Data Access Secure Exception Occurrence Flag)

This flag indicates that a secure exception has occurred due to operand access.

The DSER flag is set to 0 only when the exception status clear bit (CLR) of the secure exception status clear register (SECLR) is set to 1.

56.9.3 Secure Exception Status Clear Register (SECLR)

Address(es): SAU.SECLR 8100 0408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR	Exception Status Clear	[When read] 0: Fixed value for reading [When written] 0: No operation 1: Sets the SESTS.DSER and SESTS.ISER flags to 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

CLR Bit (Exception Status Clear)

This bit sets the data access secure exception flag (DSER) and instruction access secure exception flag (ISER) in the secure exception status register (SESTS) to 0.

56.9.4 Secure Exception Address Register (SEA)

Address(es): SAU.SEA 8100 040Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SEA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SEA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SEA[31:0]	Secure Exception Address	Address of the access that caused a secure exception	R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

SEA[31:0] Bits (Secure Exception Address)

These bits hold the address of the access that caused a secure exception due to operand access. They are not updated when a secure exception occurs due to instruction access.

56.9.5 Peripheral Security Attribution Register B (PSARB)

Address(es): PSCU.PSARB 8780 4004h,
PSCU_NS.PSARB 0780 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSARB31	PSARB30	PSARB29	PSARB28	PSARB27	PSARB26	PSARB25	PSARB24	PSARB23	PSARB22	PSARB21	PSARB20	PSARB19	PSARB18	PSARB17	PSARB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PSARB12	PSARB11	—	—	PSARB8	PSARB7	—	—	PSARB4	—	—	—	PSARB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSARB0	Serial Communications Interface 12 Security Attribution	Target module: RSCI12 and the MSTPCRB.MSTPB0 bit 0: Secure 1: Non-secure	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PSARB4	I3C Bus Interface 0 Security Attribution	Target module: RI3C0 and the MSTPCRB.MSTPB4 bit 0: Secure 1: Non-secure	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	PSARB7	High-Speed I^2C-bus Interface 2 Security Attribution	Target module: RIICHS2 and the MSTPCRB.MSTPB7 bit 0: Secure 1: Non-secure	R/W
b8	PSARB8	High-Speed I^2C-bus Interface 1 Security Attribution	Target module: RIICHS1 and the MSTPCRB.MSTPB8 bit 0: Secure 1: Non-secure	R/W
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	PSARB11	USB 2.0 FS Host/Function Module 0 Security Attribution	Target module: USB0 and the MSTPCRB.MSTPB11 bit 0: Secure 1: Non-secure	R/W
b12	PSARB12	USB 2.0 High-Speed Host/Function Module Security Attribution	Target module: USBHS and the MSTPCRB.MSTPB12 bit 0: Secure 1: Non-secure	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PSARB16	Expanded Serial Peripheral Interface 0 Security Attribution	Target module: XSPI0 (plus DOTF0), the bus master MPU for XSPI0, and the MSTPCRB.MSTPB16 bit 0: Secure 1: Non-secure	R/W
b17	PSARB17	Expanded Serial Peripheral Interface 1 Security Attribution	Target module: XSPI1 (plus DOTF1), the bus master MPU for XSPI1, and the MSTPCRB.MSTPB17 bit 0: Secure 1: Non-secure	R/W
b18	PSARB18	Serial Peripheral Interface 1 Security Attribution	Target module: RSPIA1 and the MSTPCRB.MSTPB18 bit 0: Secure 1: Non-secure	R/W
b19	PSARB19	Serial Peripheral Interface 0 Security Attribution	Target module: RSPIA0 and the MSTPCRB.MSTPB19 bit 0: Secure 1: Non-secure	R/W
b20	PSARB20	Serial Communications Interface 11 Security Attribution	Target module: RSCI11 and the MSTPCRB.MSTPB20 bit 0: Secure 1: Non-secure	R/W
b21	PSARB21	Serial Communications Interface 10 Security Attribution	Target module: RSCI10 and the MSTPCRB.MSTPB21 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Bit Name	Description	R/W
b22	PSARB22	Serial Communications Interface 9 Security Attribution	Target module: RSCI9 and the MSTPCRB.MSTPB22 bit 0: Secure 1: Non-secure	R/W
b23	PSARB23	Serial Communications Interface 8 Security Attribution	Target module: RSCI8 and the MSTPCRB.MSTPB23 bit 0: Secure 1: Non-secure	R/W
b24	PSARB24	Serial Communications Interface 7 Security Attribution	Target module: RSCI7 and the MSTPCRB.MSTPB24 bit 0: Secure 1: Non-secure	R/W
b25	PSARB25	Serial Communications Interface 6 Security Attribution	Target module: RSCI6 and the MSTPCRB.MSTPB25 bit 0: Secure 1: Non-secure	R/W
b26	PSARB26	Serial Communications Interface 5 Security Attribution	Target module: RSCI5 and the MSTPCRB.MSTPB26 bit 0: Secure 1: Non-secure	R/W
b27	PSARB27	Serial Communications Interface 4 Security Attribution	Target module: RSCI4 and the MSTPCRB.MSTPB27 bit 0: Secure 1: Non-secure	R/W
b28	PSARB28	Serial Communications Interface 3 Security Attribution	Target module: RSCI3 and the MSTPCRB.MSTPB28 bit 0: Secure 1: Non-secure	R/W
b29	PSARB29	Serial Communications Interface 2 Security Attribution	Target module: RSCI2 and the MSTPCRB.MSTPB29 bit 0: Secure 1: Non-secure	R/W
b30	PSARB30	Serial Communications Interface 1 Security Attribution	Target module: RSCI1 and the MSTPCRB.MSTPB30 bit 0: Secure 1: Non-secure	R/W
b31	PSARB31	Serial Communications Interface 0 Security Attribution	Target module: RSCI0 and the MSTPCRB.MSTPB31 bit 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PSARB register is used to set the security attributes of each module and the corresponding module stop control register's bits.

56.9.6 Peripheral Security Attribution Register C (PSARC)

Address(es): PSCU.PSARC 8780 4008h,
PSCU_NS.PSARC 0780 4008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSARC31	PSARC30	—	PSARC28	PSARC27	—	PSARC25	PSARC24	PSARC23	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSARC13	PSARC12	—	—	—	PSARC8	PSARC7	—	—	—	—	PSARC2	PSARC1	PSARC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSARC0	Clock Frequency Accuracy Measurement Circuit Security Attribution	Target module: CAC0/1/2 and the MSTPCRC.MSTPC0 bit 0: Secure 1: Non-secure	R/W
b1	PSARC1	CRC Calculator Security Attribution	Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Secure 1: Non-secure	R/W
b2	PSARC2	Parallel Data Capture Unit Security Attribution	Target module: PDC and the MSTPCRC.MSTPC2 bit 0: Secure 1: Non-secure	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	PSARC7	Serial Sound Interface 1 Security Attribution	Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Secure 1: Non-secure	R/W
b8	PSARC8	Serial Sound Interface 0 Security Attribution	Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Secure 1: Non-secure	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	PSARC12	SD/MMC Host Interface Security Attribution	Target module: SDHI and the MSTPCRC.MSTPC12 bit 0: Secure 1: Non-secure	R/W
b13	PSARC13	Data Operation Circuit Security Attribution	Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Secure 1: Non-secure	R/W
b22 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	PSARC23	Serial Peripheral Interface 2 Security Attribution	Target module: RSPIA2 and the MSTPCRC.MSTPC23 bit 0: Secure 1: Non-secure	R/W
b24	PSARC24	Pulse-Density Modulation Interface Security Attribution	Target module: PDMIF and the MSTPCRC.MSTPC24 bit 0: Secure 1: Non-secure	R/W
b25	PSARC25	CANFDG Module Security Attribution	Target module: CANFDG and the MSTPCRC.MSTPC25 bit 0: Secure 1: Non-secure	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	PSARC27	CANFD Module Security Attribution	Target module: Controller area network with CANFD and the MSTPCRC.MSTPC27 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	PSARC28	Ether-PHY Clock Security Attribution	Target module: ETHPHYCLK and the MSTPCRC.MSTPC28 bit 0: Secure 1: Non-secure	R/W
b29	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30	PSARC30	Ethernet Module Security Attribution	Target module: ESWM and the MSTPCRC.MSTPC30 bit 0: Secure 1: Non-secure	R/W
b31	PSARC31	Renesas Secure IP Security Attribution	Target module: RSIP-E50D and the MSTPCRC.MSTPC31 bit 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PSARC register is used to set the security attributes of each module and the corresponding module stop control register's bits.

56.9.7 Peripheral Security Attribution Register D (PSARD)

Address(es): PSCU.PSARD 8780 400Ch,
PSCU_NS.PSARD 0780 400Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	PSARD22	PSARD21	PSARD20	PSARD19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSARD14	PSARD13	PSARD12	PSARD11	—	—	—	—	PSARD6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSARD6	High Resolution PWM Waveform Generation Circuit Security Attribution	Target module: HRPWM and the MSTPCRD.MSTPD6 bit 0: Secure 1: Non-secure	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	PSARD11	Port Output Enable for GPTW Group D Security Attribution	Target module: POEG Group D and the MSTPCRD.MSTPD11 bit 0: Secure 1: Non-secure	R/W
b12	PSARD12	Port Output Enable for GPTW Group C Security Attribution	Target module: POEG Group C and the MSTPCRD.MSTPD12 bit 0: Secure 1: Non-secure	R/W
b13	PSARD13	Port Output Enable for GPTW Group B Security Attribution	Target module: POEG Group B and the MSTPCRD.MSTPD13 bit 0: Secure 1: Non-secure	R/W
b14	PSARD14	Port Output Enable for GPTW Group A Security Attribution	Target module: POEG Group A and the MSTPCRD.MSTPD14 bit 0: Secure 1: Non-secure	R/W
b18 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19	PSARD19	12-Bit D/A Converter 1 Security Attribution	Target module: DAC1 and the MSTPCRD.MSTPD19 bit 0: Secure 1: Non-secure	R/W
b20	PSARD20	12-Bit D/A Converter 0 Security Attribution	Target module: DAC0 and the MSTPCRD.MSTPD20 bit 0: Secure 1: Non-secure	R/W
b21	PSARD21	16-Bit A/D Converter Security Attribution	Target module: ADC16H and the MSTPCRD.MSTPD21 bit 0: Secure 1: Non-secure	R/W
b22	PSARD22	Temperature Sensor Security Attribution	Target module: TEMPS and the MSTPCRD.MSTPD22 bit 0: Secure 1: Non-secure	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PSARD register is used to set the security attributes of each module and the corresponding module stop control register's bits.

56.9.8 Peripheral Security Attribution Register E (PSARE)

Address(es): PSCU.PSARE 8780 4010h,
PSCU_NS.PSARE 0780 4010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSARE31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSARE15	—	—	—	—	—	—	—	—	—	—	—	PSARE3	PSARE2	PSARE1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	PSARE1	Watchdog Timer Security Attribution	Target module: WDT 0: Secure 1: Non-secure	R/W
b2	PSARE2	Independent Watchdog Timer Security Attribution	Target module: IWDT 0: Secure 1: Non-secure	R/W
b3	PSARE3	Real Time Clock Security Attribution	Target module: RTC 0: Secure 1: Non-secure	R/W
b14 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PSARE15	General PWM Timer Common Security Attribution	Target module: GPTW GTCLKCR register 0: Secure 1: Non-secure	R/W
b30 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	PSARE31	General PWM Timer Channel 15 to 0 Security Attribution	Target module: GPTW15 to GPTW0 and the MSTPCRE.MSTPE31 bit 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PSARE register is used to set the security attributes of each module and the corresponding module stop control register's bits.

56.9.9 Peripheral Security Attribution Register F (PSARF)

Address(es): PSCU.PSARF 8780 4040h,
PSCU_NS.PSARF 0780 4040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSARF31	PSARF30	—	—	—	—	—	—	—	—	—	—	—	—	—	PSARF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSARF15	PSARF14	PSARF13	PSARF12	—	—	—	—	PSARF7	PSARF6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSARF6	8-Bit Timer 3, 2 (Unit 1) Attribution	Target module: TMR3, TMR2 and the MSTPCRF.MSTPF6 bit 0: Secure 1: Non-secure	R/W
b7	PSARF7	8-Bit Timer 1, 0 (Unit 0) Attribution	Target module: TMR1, TMR0 and the MSTPCRF.MSTPF7 bit 0: Secure 1: Non-secure	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	PSARF12	Compare Match Timer W (Unit 1) Security Attribution	Target module: CMTW1 and the MSTPCRF.MSTPF12 bit 0: Secure 1: Non-secure	R/W
b13	PSARF13	Compare Match Timer W (Unit 0) Security Attribution	Target module: CMTW0 and the MSTPCRF.MSTPF13 bit 0: Secure 1: Non-secure	R/W
b14	PSARF14	Compare Match Timer 3, 2 (Unit 1) Security Attribution	Target module: CMT3, CMT2 and the MSTPCRF.MSTPF14 bit 0: Secure 1: Non-secure	R/W
b15	PSARF15	Compare Match Timer 1, 0 (Unit 0) Security Attribution	Target module: CMT1, CMT0 and the MSTPCRF.MSTPF15 bit 0: Secure 1: Non-secure	R/W
b16	PSARF16	Software Configurable 32-Bit Programmable I/O Port Security Attribution	Target module: PIO and the MSTPCRF.MSTPF16 bit 0: Secure 1: Non-secure	R/W
b29 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	PSARF30	Phase Counting Interface 1 Security Attribution	Target module: PCIF1 and the MSTPCRF.MSTPF30 bit 0: Secure 1: Non-secure	R/W
b31	PSARF31	Phase Counting Interface 0 Security Attribution	Target module: PCIF0 and the MSTPCRF.MSTPF31 bit 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The PSARF register is used to set the security attributes of each module and the corresponding module stop control register's bits.

56.9.10 Module Stop Security Attribution Register (MSSAR)

Address(es): PSCU.MSSAR 8780 4014h,
PSCU_NS.MSSAR 0780 4014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSSAR31	—	MSSAR29	—	—	—	—	—	—	MSSAR22	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSSAR1	MSSAR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSSAR0	SRAM0 Clock Stop Security Attribution	Target module: MSTPCRA.MSTPA0 bit 0: Secure 1: Non-secure	R/W
b1	MSSAR1	SRAM1 Clock Stop Security Attribution	Target module: MSTPCRA.MSTPA1 bit 0: Secure 1: Non-secure	R/W
b21 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b22	MSSAR22	DMAC/DTC Clock Stop Security Attribution	Target module: MSTPCRA.MSTPA22 bit 0: Secure 1: Non-secure	R/W
b28 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	MSSAR29	EXDMAC Clock Stop Security Attribution	Target module: MSTPCRA.MSTPA29 bit 0: Secure 1: Non-secure	R/W
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31	MSSAR31	ELC Clock Stop Security Attribution	Target module: MSTPCRC.MSTPC14 bit 0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.

The MSSAR register is used to set the security attributes of the corresponding module stop control register's bits.

56.9.11 Code MRAM Security Attribution Monitor Register (CMSAMON)

Address(es): PSCU.CMSAMON 8780 4030h,
PSCU_NS.CMSAMON 0780 4030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CMS[8:1]							
Value after reset:	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMS[0:0]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note 1. The value is 1FFh in a blank product. It is set to the value written by your application.

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0.	R
b23 to b15	CMS[8:0]	Code MRAM Secure Area	Indicate the area of the secure region for code MRAM.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

56.9.12 Data MRAM Security Attribution Monitor Register (DMSAMON)

Address(es): PSCU.DMSAMON 8780 4034h,
PSCU_NS.DMSAMON 0780 4034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMS[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset:	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0	0

Note 1. The value is 1Fh in a blank product. It is set to the value written by your application.

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	—	Reserved	These bits are read as 0.	R
b15 to b11	DMS[4:0]	Data MRAM Secure Area	Indicate the area of the secure region for data MRAM.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

56.9.13 Device Lifecycle Management State Monitor Register (DLMMON)

Address(es): PSCU.DLMMON 8780 4038h,
PSCU_NS.DLMMON 0780 4038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	DLMMON[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1

Note 1. The value is 0100b in a blank product. The value of these bits depends on the DLM status.

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLMMON[3:0]	Device Lifecycle Management State Monitor	Indicate the DLM status value. b3 b0 0 1 0 0: OEM 0 1 1 0: LCK_BOOT 0 1 1 1: RMA_REQ 1 0 0 0: RMA_ACK 1 0 0 1: RMA_RET Other than above: Reserved	R
b31 to b4	—	Reserved	These bits are read as 0.	R

56.9.14 Master Security Attribution Operation-After-Detection Register (MSAOAD)

Address(es): BUS.MSAOAD 8701 E010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OAD	Operation after access violation detection	0: Interrupt 1: Reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key code	These bits enable or disable writing to the OAD bit.	W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

OAD Bit (Operation after access violation detection)

The OAD bit specifies operation when the access violation is detected. When OAD = 0, the error response is returned to the bus master. When BUSIRQFN<Master Name>.EN = 1, an interrupt request is generated. For details, see section 16.3.35, Bus Error Interrupt Enable Register (BUSINTENCPU0IF, BUSINTENCPU0OA, BUSINTENDMAC0, BUSINTENEXDMAC, BUSINTENEDMAC, BUSINTENXSPI0, BUSINTENXSPI1). Also, bus errors can be used as non-masukable interrupts. For details, see the specifications for the ICU.NMIER register.
When OAD = 1, a reset request is generated.

KEY[7:0] Bits (Key code)

The KEY[7:0] bits are used to enable or disable writing to the OAD bit. When writing to the OAD bit, use 16-bit access to write A5h to the KEY[7:0] bits at the same time. If this value is not written to the KEY[7:0] bits, the OAD bit is not updated.

56.9.15 Master Security Attribution Protect Register (MSAPT)

Address(es): BUS.MSAPT 8701 E014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PROTECT	Register Write Protection	0: Disables write protection. 1: Enables write protection. Reading is possible.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writing to the PROTECT bit.	W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

PROTECT Bit (Register Write Protection)

The PPROTECT bit enables or disables writes to the associated register to be protected. The MSAPT.PROTECT bit controls the MSAOAD register.

KEY[7:0] Bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, use 16-bit access to write A5 to KEY[7:0] bits at the same time. If this value is not written to the KEY[7:0] bits, the PROTECT bit is not updated.

56.10 Secure Key Injection

To inject a user key into the MCU, perform the following steps.
Renesas provides the Security Key Management Tool, available on the Renesas website, to assist with key injection preparation.

1. Create a 256-bit installation key.
   This key is called the user factory programming key (UFPK) and is used to wrap a user key.
2. Get the wrapped version (W-UFPK) of the UFPK through the Renesas key wrapping service.
3. Wrap the user key using the UFPK.
4. Send the W-UFPK and the wrapped user key to the MCU using a boot firmware interface.
   First, the user key is unwrapped. Then, it is wrapped with the MCU's unique hardware key and stored in non-volatile memory. DLM keys are stored in MRAM that is not memory-mapped. An application key is stored at the address specified with the key injection command.

Figure 56.7 shows an example of key injection and Table 56.11 shows the keys that can be injected.

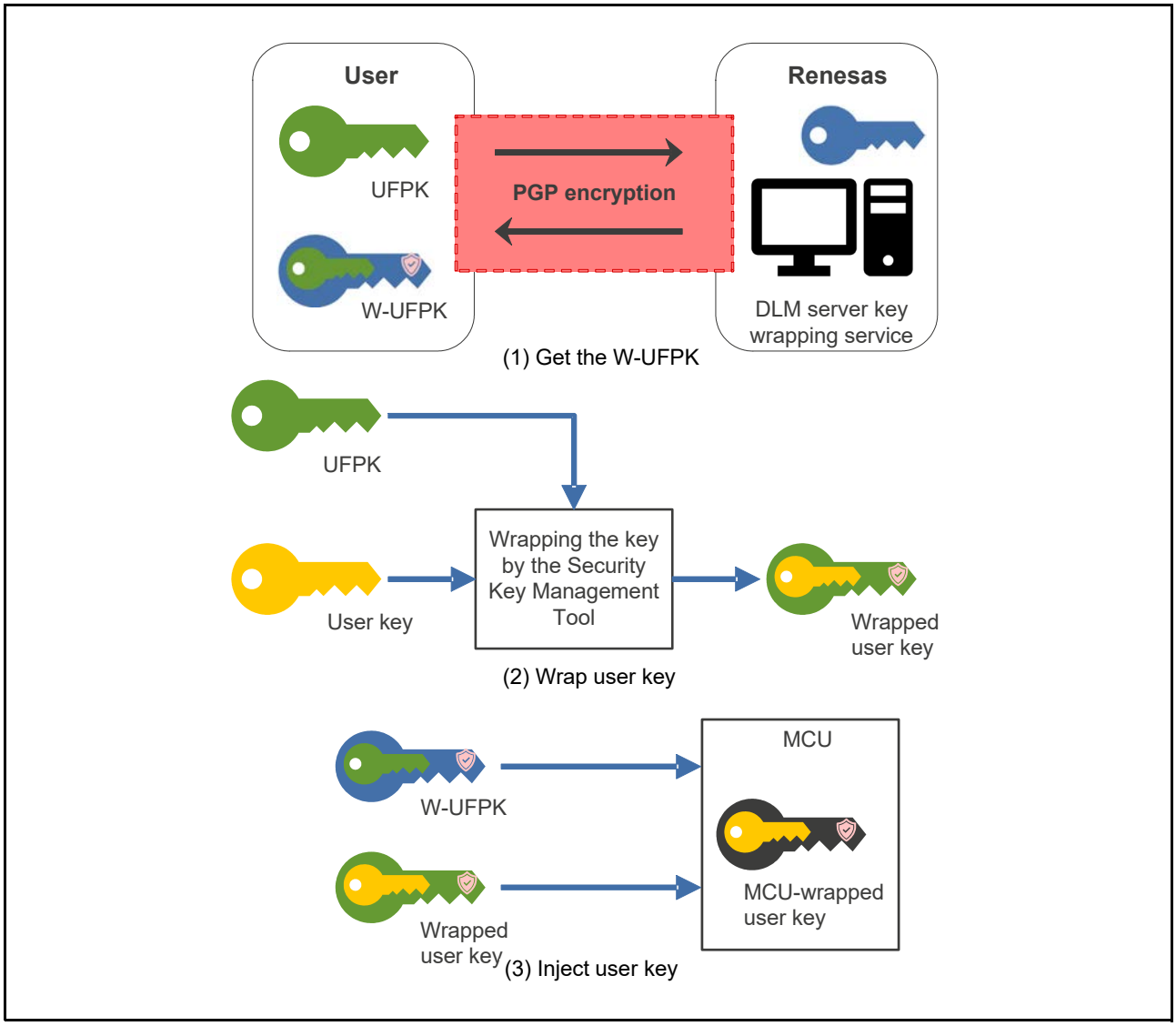


Figure 56.7 Key Injection

Table 56.11 Keys That Can Be Injected

DLM Transition	RMA_KEY (up to four)
AL Transition	AL2_KEY and AL1_KEY (up to four of each)
AES	AES-128, AES-192, AES-256, AES-XTS-128, AES-XTS-256
ChaCha20-Poly-1305	ChaCha20-Poly-1305
RSA	RSA-1024, RSA-2048, RSA-3072, RSA-4096 (public and private keys) RSA-2048 Public key for TLS
ECC	secp192r1, secp224r1, secp256r1, secp384r1, secp521r1 (public and private keys) Brainpool P256r1, Brainpool P384r1, Brainpool P512r1, Koblitz secp256k1 (public and private keys)
HMAC	HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, HMAC-SHA512/224, HMAC-SHA512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512
EdDSA	Ed25519 (public and private keys)
Secure Boot	OEM_ROOT_PK (Up to 4)
Others	Key-update key

56.11 Secure Factory Programming

Secure factory programming is to program a firmware image in ciphertext format to prevent asset leakage during production programming. This MCU supports secure factory programming in addition to supporting secure injection of the DLM keys, debug authentication keys, application keys, and secure boot keys shown in Table 56.11. This enables programming in an environment that is not secure. This function is implemented in the boot program. Figure 56.8 shows an example of secure factory programming of an encrypted image. The user should encrypt the firmware image with AES128-CCM by using the firmware image encryption key (called the image encryption key). The image encryption key should be wrapped with the UFPK. When the user sends the W-UFPK, the wrapped image encryption key, and encrypted image to the MCU through the serial programming interface, the MCU decrypts and programs the image.

To support secure factory programming in an environment that is not secure, the DLM state, protection level, and authentication keys can all be set via a single boot command. Points to note about this boot command include:

- Execution of the boot command that supports secure factory programming is only possible when the MCU is in the OEM state.
- This command will change the protection level of the MCU. The initial PL of the device is PL2. Set the final PL to PL0.
- If the DLM state will remain in the OEM state, AL2_KEY must be injected. AL1_KEY can also be injected optionally.
- The MCU can be transitioned to the LCK_BOOT state. In this case, AL keys cannot be injected.
- AL keys must be wrapped with the same UFPK as for the image encryption key.
- This command will erase all code MRAM areas and data MRAM areas except option-setting memory before programming an encrypted image. Therefore, if the permanent block protection function has been applied to any area, this command cannot be executed.
- This command will not be executed if the current register settings or the write values to the registers related to startup area selection and startup bank selection are other than the following:
  - SAS2.BTFLG = 1b
  - BANKSEL_NS.BANKSWP[2:0] = 111b
  - BANKSEL_S.BANKSWP[2:0] = 111b

- All option-setting memory values must be included in the encrypted image, including settings that are not used with default values. However, if the following areas are write-protected, the write data in these areas should not be included in the image. If included, this command will terminate with an error.
  - Configuration setting area protected by the POFSPSSn registers
  - Configuration setting area protected by the POFSPSNS0 registers

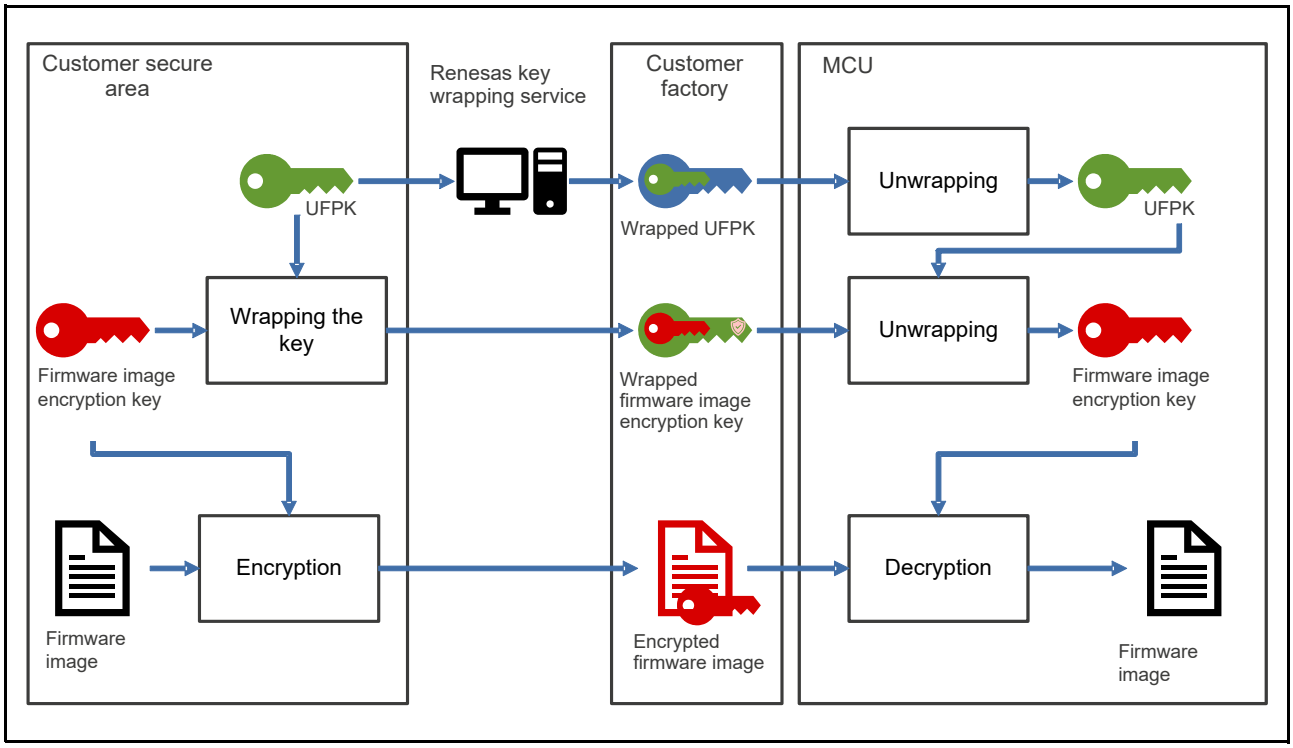


Figure 56.8 Secure Factory Programming of Encrypted Image

56.12 Secure Boot

Secure boot verifies the integrity and authenticity of an OEM boot loader (called OEM_BL) starting at the beginning of the user program.
The OEM_BL is verified when it is programmed and before it is excecuted.
It is verified when it is programmed using the boot firmware. Two certificates are used for validation. One is the key certificate, which is used to certify the OEM_BL verification key (called the OEM_BL_PK, the public key of the OEM_BL verification key pair). The key certificate is signed by the secret key of OEM_BL verification root key pair (called OEM_ROOT_SK). The public key of OEM_BL verification root key pair (called OEM_ROOT_PK) is registered with the MCU as the root of trust key using secure key injection as described in section 56.10, Secure Key Injection.
The OEM_ROOT_SK and OEM_ROOT_PK are a 256-bit ECC (secp256r1 curve) key pair. The SHA2-256 hash of the OEM_ROOT_PK is stored during the secure key injection process.
The other certificate for validation during programming is the code certificate, which is used to certify the OEM_BL. The code certificate is signed by the secret key of OEM_BL verification key pair (called OEM_BL_SK). After successful validation of the OEM_BL, the HMAC value of OEM_BL and the code certificate (called OEM_BL_digest) is generated and programmed into MRAM. HMAC SHA2-256 is used as the HMAC value. A key derived from the HUK is used in HMAC generation, so the HMAC value is unique in each MCU.
In single-chip mode, the first-stage boot loader (FSBL) stored in the area that cannot be rewritten is executed after a reset when the FSBL is enabled in the FSBLCTRL0 register. When HMAC authentication is selected in FSBLCTRL1 register, the FSBL generates the HMAC value of the OEM_BL and the code certificate, and compares it to the expected HMAC value. When the HMAC values match, the FSBL jumps to the OEM_BL. When CRC validation is selected in the FSBLCTRL1 register, the FSBL calculates the CRC of the OEM_BL and compares it to the expected CRC value, which is located in the code certificate. If the CRC values match, the FSBL jumps to the OEM_BL.
The code certificate must be programmed into MRAM, and its location in MRAM must be programmed into the SACC0 register or SACC1 register. The SACC0 register specifies the start address of the code certificate when BANKSWP[2:0] = 111b in dual mode or when BTFLG = 1 in linear mode. The SACC1 register specifies the start address of the code certificate when BANKSWP[2:0] = 000b in dual mode or when BTFLG = 0 in linear mode. The SACC0n and SACC1n registers are in the OTP area and cannot be erased. Therefore, each set consists of four registers allocated to four locations for use in updating the start address of the code certificate. The FSBL reads the values of the given registers in order, starting with the one with the highest number n, and uses the first register that includes non-1 bits as the start address of the code certificate. Accordingly, write to the registers of each set in order, starting with the register having the lowest number n.
When measurement reporting is enabled in the FSBLCTRL1 register, the FSBL stores the measurement report at the SRAM address specified by the SAMR register.
FSBL execution can be skipped after a software reset or deep software standby reset, as configured by the FSBLCTRL0 register. If the generated HMAC or CRC value is different from the expected value, a high level is output to the port set by the FSBLCTRL2 register and the MCU transitions to CPU sleep mode.

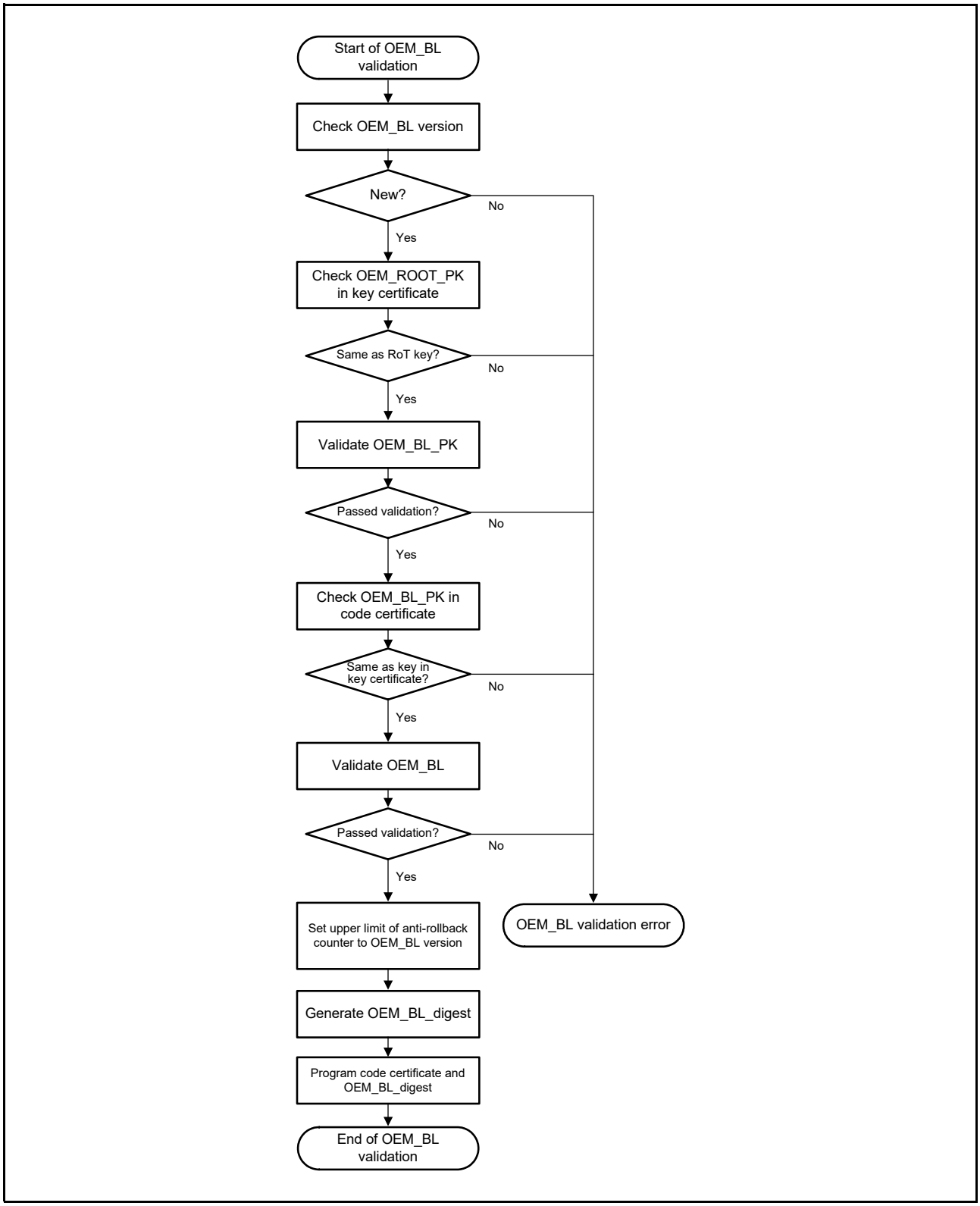


Figure 56.9 OEM_BL Validation Flow in Serial Programming Mode

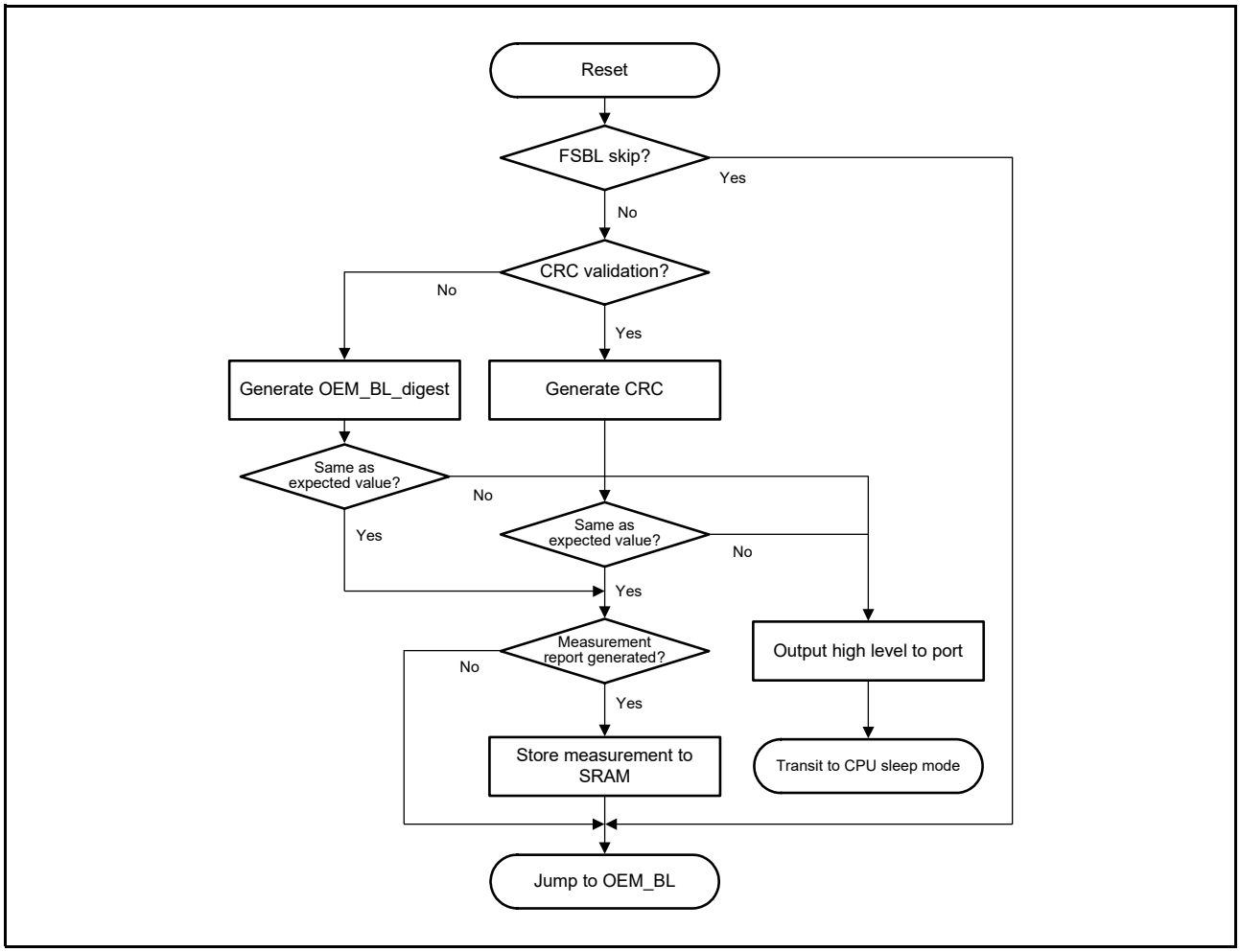


Figure 56.10 FSBL Flow in Single-Chip Mode

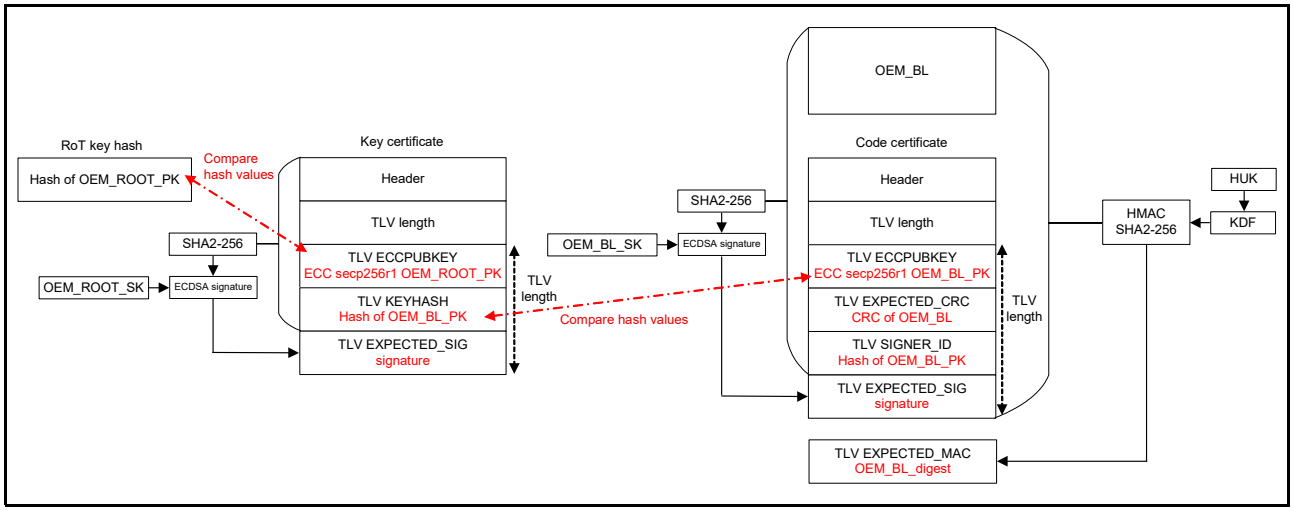


Figure 56.11 Validation Structure and Certificate Format

Table 56.12 Details of Key Certificate

Field		Size (Bytes)	Description
Header	Magic	4	Set to 6B657963h
	Manifest version	4	Set to 00010000h
	Flag	4	Reserved for future use
	Reserved	20	Reserved for future use
TLV length		4	Length of TLV field (in bytes)
TLV ECCPUBEY	Type & length	4	Set to 00088010h
	Value	64	ECC secp256r1 OEM_ROOT_PK
TLV KEYHASH	Type & length	4	Set to 10144008h
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & length	4	Set to 20088410h
	Value	64	Signature

Table 56.13 Detail of code certificate

Field		Size (Bytes)	Description
Header	Magic	4	Set to 636F6463h
	Manifest version	4	Set to 00010000h
	Flag	4	Set to 00000000h
	Load Addr	4	Set to the start address of the OEM_BL
	Dest Addr	4	Set to the start address of the OEM_BL
	Image size	4	Size in bytes of the OEM_BL. Specify a multiple of 16. The minimum size is 64 bytes.
	Image version	4	Version number of OEM_BL. It can be specified in the range of 1 to 64.
	Build number	4	Set to 00000000h
TLV length		4	Byte length of TLV field
TLV ECCPUBEY	Type & length	4	Set to 01088010h
	Value	64	ECC secp256r1 OEM_BL_PK
TLV EXPECTED_CRC	Type & length	4	Set to 40000001h
	Value	4	CRC32 of OEM_BL
TLV SIGNER_ID	Type & length	4	Set to 10144008h
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & length	4	Set to 25088410h
	Value	64	OEM_BL signed by signature OEM_BL_SK in [code certificate] When generating the signature, enter the code certificate and OEM_BL in the hash function in this order.

Table 56.14 Details of OEM_BL_digest

Field		Size (Bytes)	Description
TLV EXPECTED_MAC	Type & length	4	Fixed value 30184008h
	Value	32	Unique OEM_BL_digest in each MCU

Note: OEM_BL_digest should be programmed in the area contiguous with the code certificate.

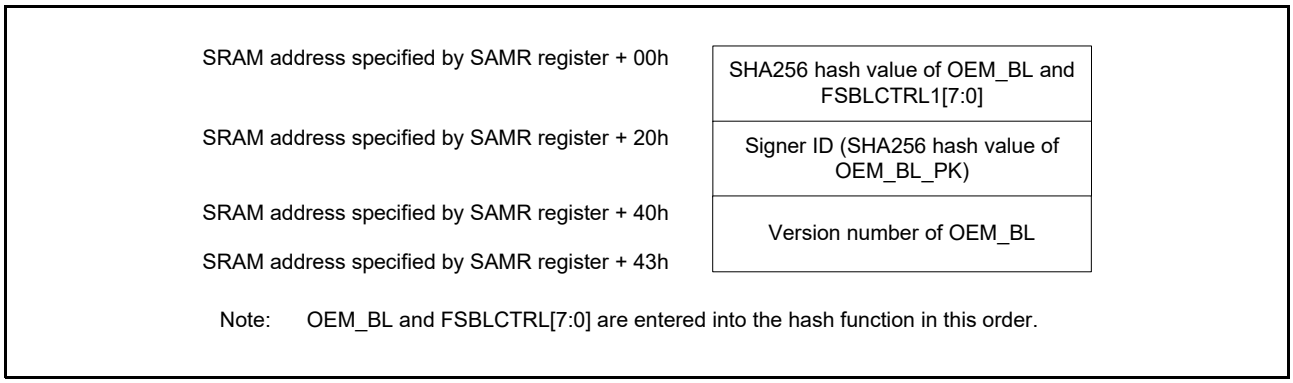
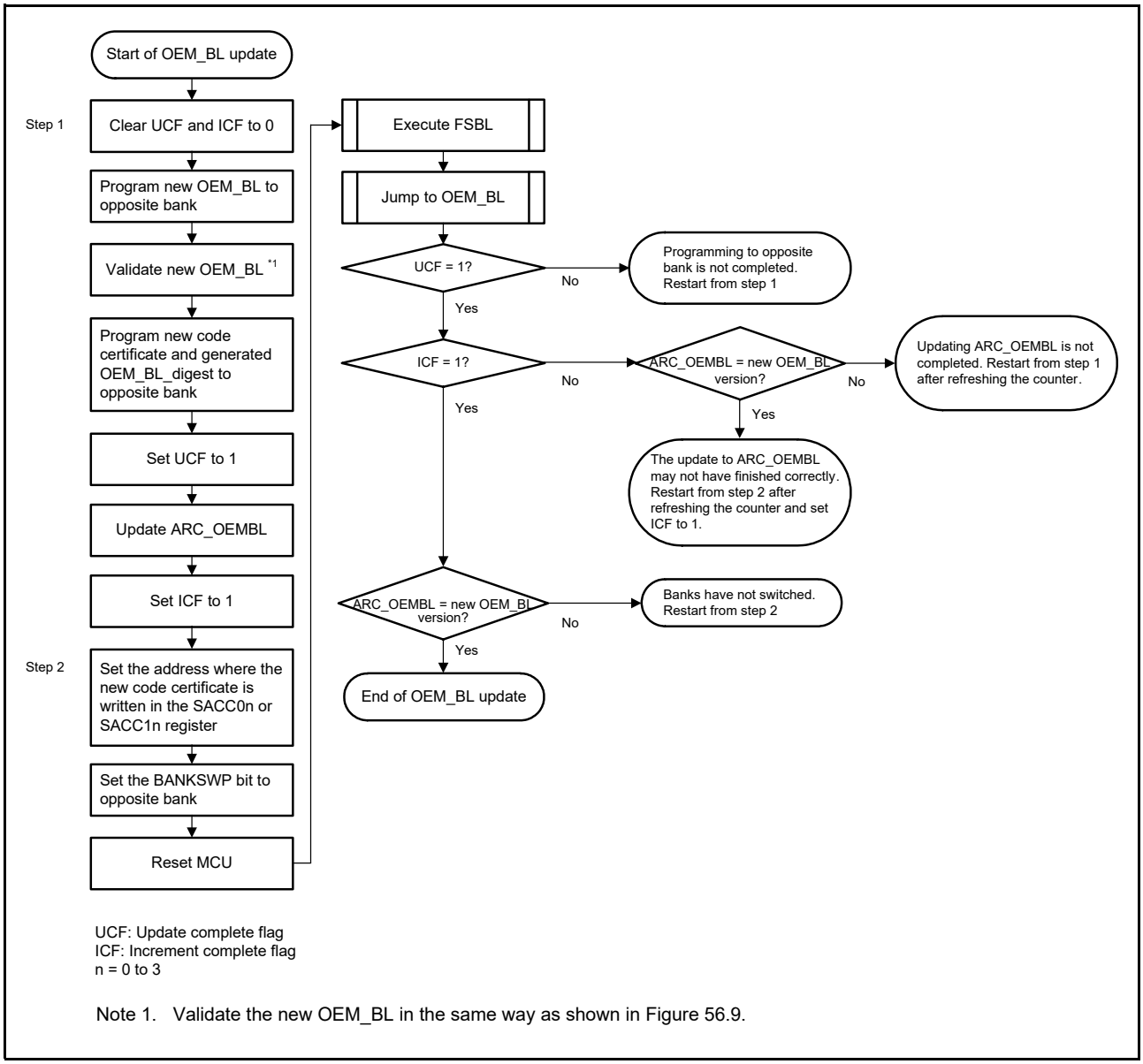


Figure 56.12 Format and Location of the Measurement Report

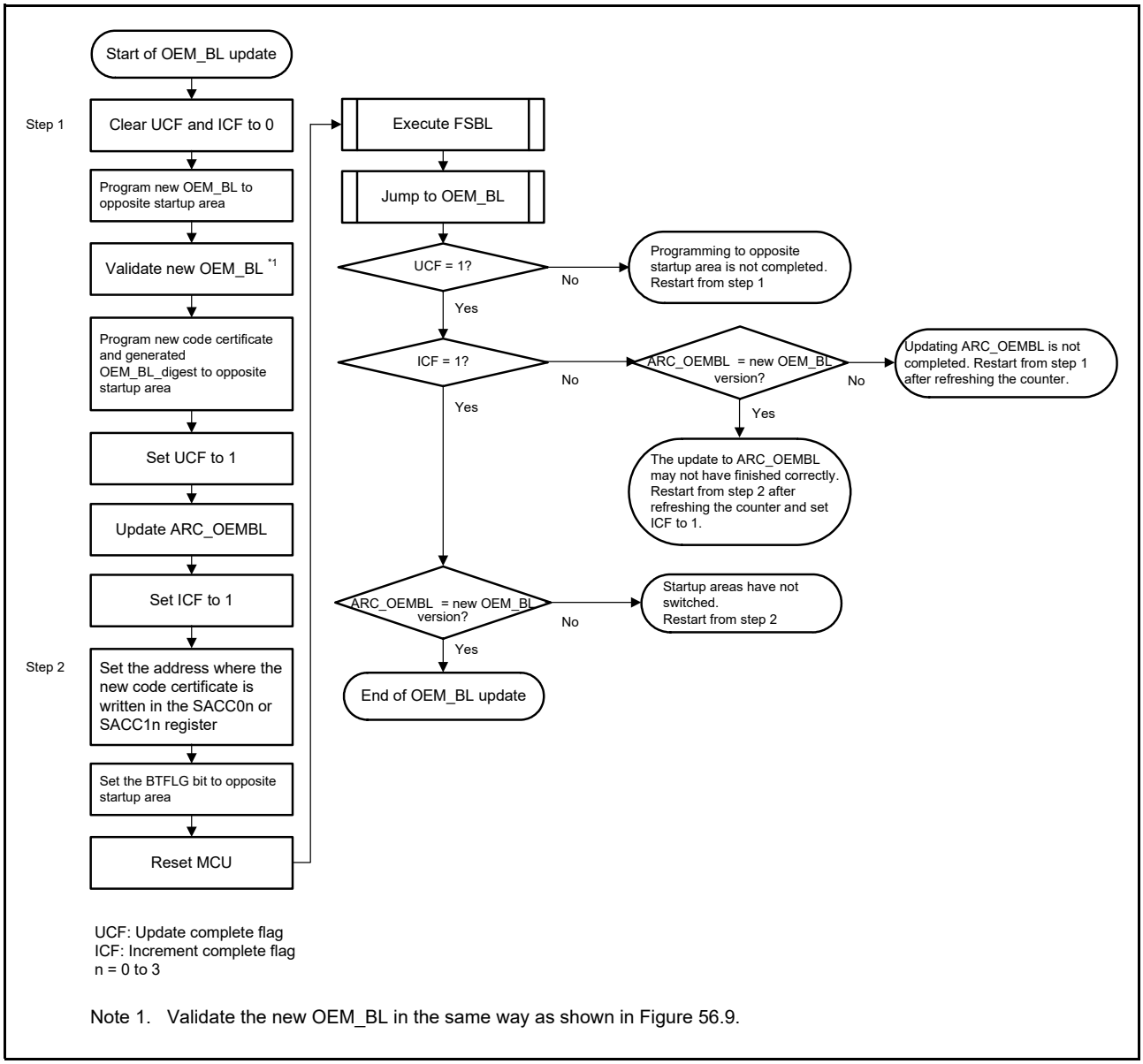
OEM_BL can be updated by application code. In this case, the application code must validate the updated bootloader, generate the OEM_BL_digest, and store it immediately after the new code certificate, as per the flow described by Figure 56.12. Figure 56.13 and Figure 56.14 show the OEM_BL update flow in dual mode and linear mode. When using linear mode, if the total size of OEM_BL, code certificate, and OEM_BL_digest is larger than the startup area size of 8 KB, this update flow cannot be used. It is recommended to use dual mode.

The implementation of the OEM_BL is user defined and application specific. However, it is recommended to implement status flags such as Update Complete Flag (UCF) and Increment Complete Flag (ICF) in code MRAM as a countermeasure against update errors due to unexpected power failure. If an update error occurs, recovery from the error is possible as shown in the following figures. If an update error occurs, the current OEM_BL is executed after FSBL execution. Note that the update error determination program is required for both the current OEM_BL and the new OEM_BL.



Note 1. Validate the new OEM_BL in the same way as shown in Figure 56.9.

Figure 56.13 OEM_BL Update Flow in Dual Mode



Note 1. Validate the new OEM_BL in the same way as shown in Figure 56.9.

Figure 56.14 OEM_BL Update Flow in Linear Mode

56.13 Field Updating in Dual Mode

Be careful when performing field firmware update to keep the contents of secure (S) or non-secure (NS) regions after bank swapping in the dual mode. Figure 56.15 and Figure 56.16 show the field update execution flow for secure and non-secure regions in dual mode.

Figure 56.15 and Figure 56.16 show the information update flow for the secure area and non-secure regions in dual bank mode in the field. Initially, bank0 contains Version 1 of the firmware, and bank1 is blank. As shown in Figure 56.15, to update the secure firmware (S firmware) to Version 2, first copy the Version 1 non-secure firmware (NS firmware) in bank0 to bank1, update the S firmware to Version 2, and then execute the start-up bank switch.

As shown in Figure 56.16, to update the NS firmware to Version 2, first copy the Version 1 S firmware from bank0 to bank1. Note that this process must be done by a secure service. After that, non-secure users should update the NS firmware to Version 2 and execute the start-up bank switch.

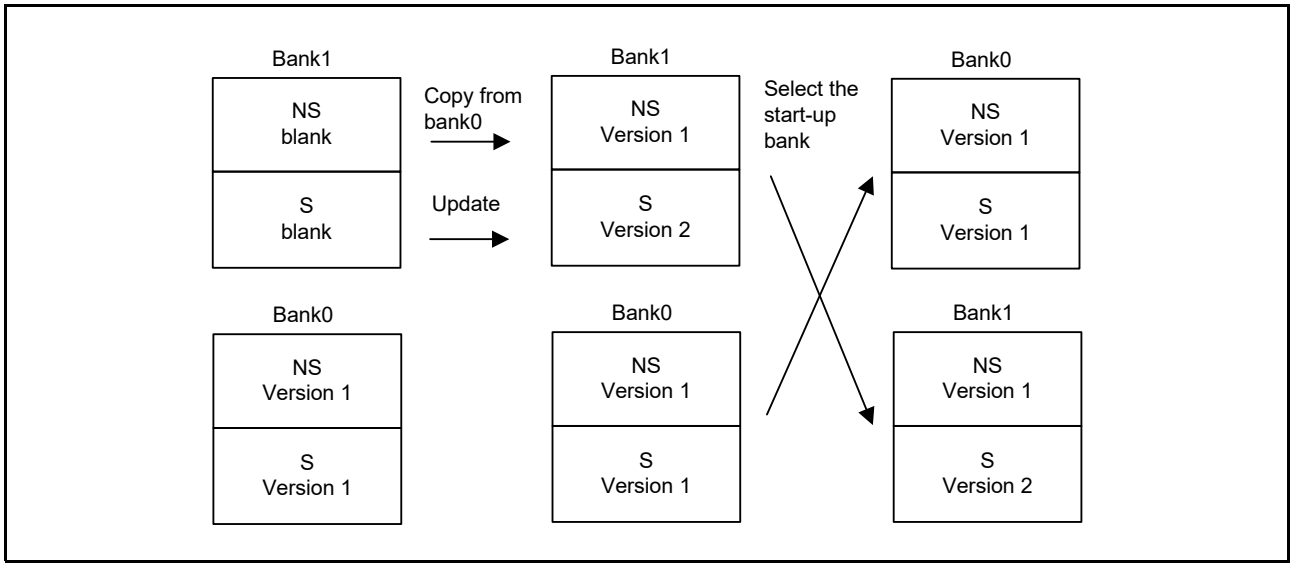


Figure 56.15 Field Updating Flow of Secure Region

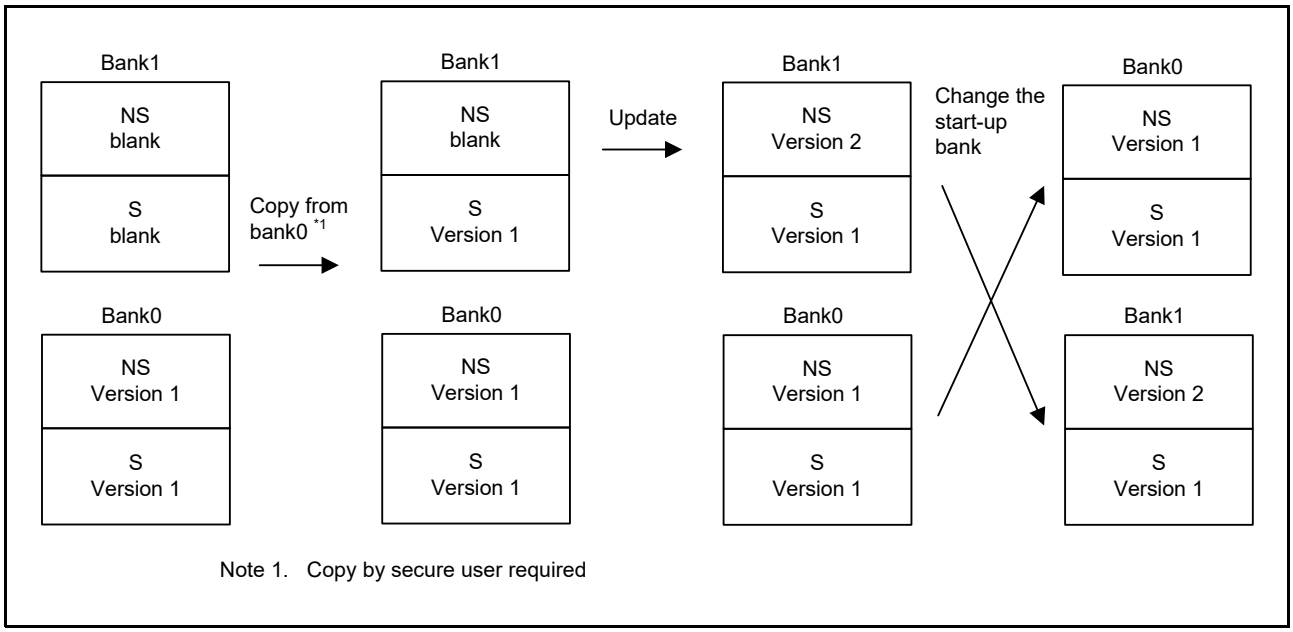


Figure 56.16 Field Updating Flow of Non-Secure Region

56.14 Device Lifecycle Management

Device Lifecycle Management (DLM) identifies the current development/production/deployment phase of the device and controls the capabilities of the debug function, the serial programming interface, and Renesas test mode. Figure 56.17 shows the available device lifecycle states and Table 56.15 shows the lifecycle state definitions and capabilities in each state.

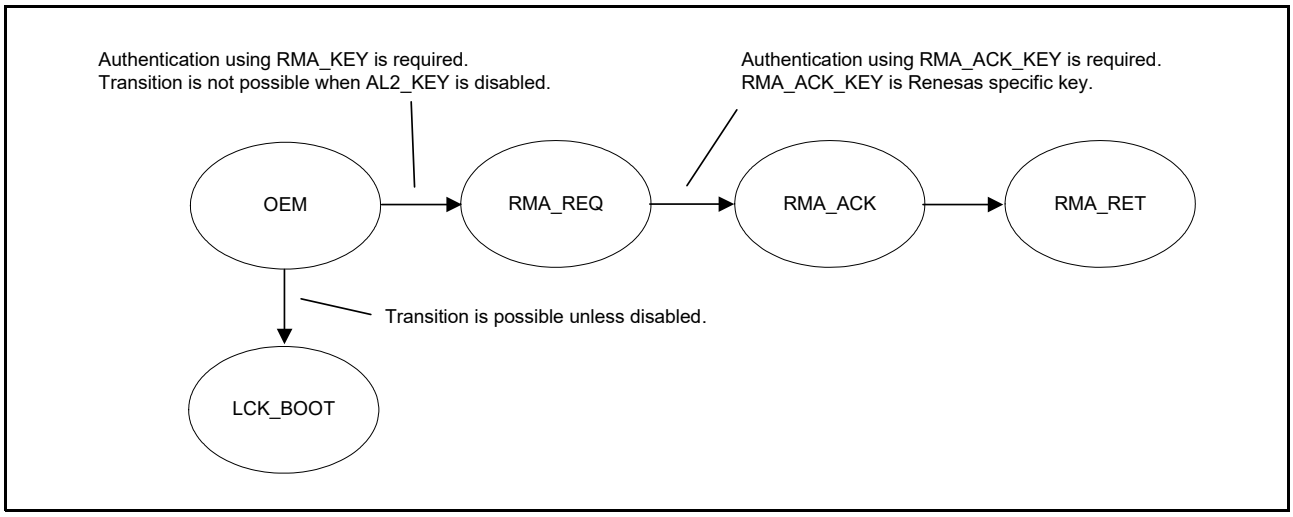


Figure 56.17 Device Lifecycle States

Table 56.15 Lifecycle State Definitions and Functions in Each State

Lifecycle	Definition	Protection Level	Debug Function	Serial Programming	Renesas Test Mode
OEM	Original Equipment Manufacturer The device is owned by the customer.	PL2, PL1 or PL0	Depends on the authentication level.		Not available
LCK_BOOT	LoCKed BOOT interface (Boot interface locked) The debug interface and the serial programming interface are permanently disabled.	PL0	Not available	Not available	Not available
RMA_REQ	Return Material Authorization REQuest (RMA request) RMA request. The customer must send the device to Renesas in this state.	PL0	Not available	Available Access to code MRAM area and data MRAM area not possible	Not available
RMA_ACK	Return Material Authorization ACKnowledged (RMA acknowledged) Failure analysis in Renesas	PL2	Non-secure and secure debug functions are enabled.	Available Access to code MRAM area and data MRAM area not possible	Available
RMA_RET	Return Material Authorization ACKnowledged (RMA return) This device has been returned to the customer. The device does not boot.	PL0	Not available	Not available	Not available

56.14.1 Changing Lifecycle States

Use the boot commands to change the device lifecycle state. The lifecycle state cannot be updated by an application, but the current lifecycle state can be read through the dedicated registers.
As shown in Figure 56.18, each transition is one-way and the state cannot be regressed.
Transition from OEM to RMA_REQ requires key authentication using the RMA_KEY which must have been previously injected by the user. The key length of the RMA_KEY is 128 bits. Inject the RMA_KEY as shown in section 56.10, Secure Key Injection.
The key authentication uses challenge and response authentication or authentication using the MCU's unique ID. The response (challenge and response authentication) or the authentication code (using the MCU's unique ID) can be calculated as follows:
Response = AES-128 CMAC (RMA_KEY, 128-bit challenge)
Authentication code = AES-128 CMAC (RMA_KEY, 128-bit MCU unique ID)
The contents of the code MRAM and data MRAM except permanently locked blocks or registers are erased when transitioning to RMA_REQ. The contents of the permanently locked blocks or registers can be read by Renesas at failure analysis. MRAM blocks can be permanently locked by setting the PBPS_NS and PBPS_S registers or PBPSD_NS and PBPSD_S registers to permanently disable rewriting of the code and data MRAM blocks. The SAS and SAS2 registers can be permanently locked by using the FSPRS and FSPR bits, which permanently disable programming and erasure of the registers. Transition to RMA_REQ is not possible if the AL2_KEY is disabled. The MCU does not respond after changing the device lifecycle state to RMA_REQ. To continue to use boot commands, re-transition back to boot mode is required after a reset.
Transition from OEM to LCK_BOOT is possible unless it has been explicitly disabled. Use the parameter setting command at authentication level AL2 or AL1 to prohibit the transition to LCK_BOOT. The transition to LCK_BOOT is a permanently disabled and cannot be reversed. The debug interface and serial programming interface are permanently disabled in LCK_BOOT.

56.14.2 Protection and Authentication Levels

The protection level (PL) and the authentication level (AL) determine the availability of the debug function and the serial programming interface. PL and AL are fixed except in the OEM state. In OEM mode, three levels can be set for PL and AL. AL indicates a temporary authentication status and is initialized to PL after an MCU power-on reset. Both PL and AL can be changed only by the boot firmware and cannot be changed by the application. Figure 56.18 shows the available protection level and authentication level states and transitions. Table 56.16 shows the availability of the debug function and the serial programming interface in each level.

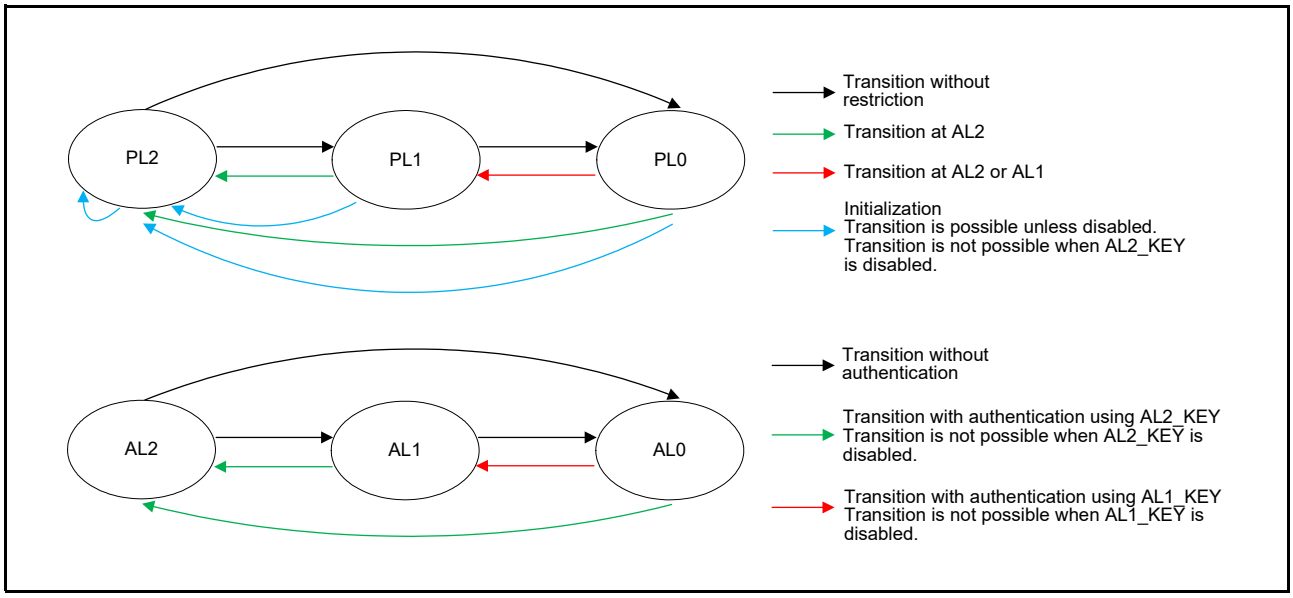


Figure 56.18 PL and AL States and Transitions

Table 56.16 Availability of the Debug Function and the Serial Programming Interface in Each Authentication Level

AL	Debug Function	Serial Programming Interface
AL2	Non-secure and secure debug functions are enabled and accessible from the debugger.	All functions are available
AL1	Only the non-secure debug function is enabled, and the debugger can access only defined non-secure debug accessible regions.	Available, but programming, erasing, and reading of the secure region of code MRAM and data MRAM are not possible.
AL0	No debug functions are available.	Available, but access to code MRAM and data MRAM is not possible.

Use the boot commands to change the PL and AL.

Changing to a lower PL can be done with no restriction. Changing to a higher PL requires that the MCU be at that AL or higher. For example, when the current AL is AL1, changing to PL1 is possible but changing to PL2 is impossible. The PL can be reset with an Initialize command as long as the command itself is not invalid. The PL is back to PL2 and the contents on MRAM are erased. If the OTP area is not empty, the Initialize command will not be executed. The Initialize command can be disabled permanently in all AL states by the parameter setting command to prevent users from erasing the contents of MRAM.

The Initialize command is disabled also when AL2_KEY is disabled. The MCU does not respond after executing the Initialize command. To continue to use the boot commands, re-transition back to boot mode is required after a reset.

Changing to a lower AL can be done without authentication. Changing to a higher AL requires key authentication using AL2_KEY or AL1_KEY, as appropriate. These keys are 128-bit keys. Injection of AL2_KEY or AL1_KEY is performed as shown in section 56.10, Secure Key Injection. AL2_KEY can be injected at authentication level AL2. AL1_KEY can be injected at authentication level AL2 or AL1. The key authentication uses challenge and response authentication. The response can be calculated as follows:

Response = AES-128 CMAC (KEY, 128-bit challenge)

AL2_KEY can be disabled permanently at authentication level AL2 with the parameter setting command. AL1_KEY can be disabled permanently at authentication level AL2 or AL1 with the parameter setting command.

56.14.3 Serial Programming

Whether a serial programmer can be connected and the range of MRAM that can be accessed depends on the device lifecycle state and AL as shown in Table 56.15 and Table 56.16. In addition, the acceptable serial programming commands differ depending on the device lifecycle state and AL.

56.14.4 Device Lifecycle States and PL Change Examples

This section describes typical device lifecycle states and PL change examples.

- Secure developers
  - Set the memory security attributes of the code MRAM and data MRAM using the boot commands.
  - Program and debug the secure application.
  - If the AL2_KEY and RMA_KEY are required, inject them using the boot commands.
  - Inject application AES, ChaCha20-Poly1305, RSA, ECC, HMAC, EdDSA keys listed in Table 56.11 (if required).
  - MCU preparations for non-secure developers
    - If the non-secure developer will not be permitted to use the Initialize command, disable it using the boot commands.
    - If the AL2_KEY will not be used and the non-secure developer will not be permitted to use the Initialize function, disable the AL2_KEY using the boot commands.
    - Change the PL from PL2 to PL1 using the boot commands.
- Non-secure developers
  - Program and debug non-secure applications.
  - If the AL1_KEY is required, inject it using the boot commands.
  - Inject application AES, RSA, ECC, HMAC keys listed in Table 56.11 (if required).
  - MCU preparations for end-product deployment
    - Disable the Initialize command using the boot commands (if required).
    - If the AL1_KEY will not be used, disable it using the boot commands.
    - Change the PL state from PL1 to PL0 using the boot commands.

56.14.5 Failure Analysis

If the user requests failure analysis by Renesas, it is necessary to send the device after changing the device lifecycle state to RMA_REQ. If the device lifecycle state is not RMA_REQ, Renesas cannot perform the failure analysis. After failure analysis, Renesas changes the lifecycle to RMA_RET and returns the device to the user.

Note: RMA_KEY is required to change the device lifecycle state to RMA_REQ or the MCU's Unique ID must be used as part of the authentication code. For details, see section 56.14.1, Changing Lifecycle States.

56.15 Usage Notes

56.15.1 Notes on Settings of the Security Attribution Bits

Program execution begins in the root world after release from a reset. Before shifting to another world, set the security attribute for each resource of the memory and peripheral modules. Specify the security attribute for a resource before using the resource and do not change the setting. Determine the security attribute for a memory address before accessing the address.

56.15.2 Security Attribution Bit Write Timing

When writing a security attribution bit, confirm that the write is complete by reading the security attribution register until it matches the value written. The protection is not effective until the write to the security attribution register is complete.

56.15.3 Executing the WAIT Instruction in Non-Flat-Project Operation

When CPSCU.PRTZCR.PRTZC is set to 1 (Non-Flat-Project operation), do not execute the WAIT instruction in the root world. In this state, the root world does not accept interrupts destined for the secure or non-secure world, so the MCU cannot be released from the low-power state by these interrupts.

57. Renesas Secure IP (RSIP-E50D)

This MCU incorporates a Renesas Secure IP (RSIP-E50D) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the RSIP library, the RSIP can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).

Key information to be used in encrypting and decrypting data is only stored within the RSIP, and any external access can be shut out to obtain a system with strong security.

57.1 Overview

Table 57.1 summarizes the specifications of the RSIP. Figure 57.1 shows a block diagram of the RSIP.

Table 57.1 Specifications of RSIP (1/2)

Item	Description
Access control	Access management circuit • In case of irregular access to the RSIP due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the RSIP.
Symmetric-key cryptography	AES: Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128, 192, or 256 bits • Block sizes: 128 bits • Block cipher mode of operation ECB, CBC, CTR: Compliant with NIST SP 800-38A CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with IEEE 1619-2007 • Message Authentication Code Algorithm CMAC: Compliant with NIST SP 800-38B GMAC: Compliant with NIST SP 800-38D ChaCha20-Poly1305: Compliant with RFC7539 • Key sizes: 256 bits • Block sizes: 512 bits • ChaCha20 algorithm ChaCha20-Poly1305
Public-key cryptography	RSA • Key sizes: 1024, 2048, 3072, or 4096 bits • Block sizes: Up to 4224 bits • Signing, Signature verification, public-key encryption, private-key decryption ECC • Key sizes: Up to 521 bits • Block sizes: Up to 576 bits • Signing, Signature verification, Key generation
Message digest function	HASH • Secure hash algorithm SHA-1, SH-224, SHA-256, SHA-512/224, SHA-512/256, SHA-384, SHA-512: Compliant with FIPS PUB 180-4 SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256: Compliant with FIPS PUB 202 • Message authentication algorithm HMAC: Compliant with FIPS PUB 198 • Key sizes: Up to 512-bit • Block sizes: 512 bits (SHA-1, SHA-224, SHA-256) 1024 bits (SHA-512/224, SHA-512/256, SHA-384, SHA-512) 1152 bits (SHA3-224) 1088 bits (SHA3-256) 832 bits (SHA3-384) 576 bits (SHA3-512) 1344 bits (SHAKE128) 1088 bits (SHAKE256)
Generation of random numbers	128-bit true random number generator

Table 57.1 Specifications of RSIP (2/2)

Item	Description
Hardware unique key (HUK)	• 256-bit read-only key unique to each individual MCU • HUK is wrapped with unique ID and stored in an area isolated from CPU. • HUK can only be read from the access management circuit through the dedicated bus. HUK is not accessible from CPU.
Unique ID	• 128-bit read-only ID code to identify individual MCUs. • Unique ID is accessible from the access management circuit through the dedicated bus. • Unique ID is used to unwrap the HUK.
Key management	• User key can be securely stored in MRAM or RAM by wrapping with HUK. Raw key is not stored. • Keys wrapped with HUK are valid only within the corresponding RSIP. Even if the wrapped key is leaked, it cannot be unwrapped by RSIP on other MCUs.
Supervisor mode	• The supervisor mode signal is connected to the access management circuit and is used to limit control of the RSIP module to supervisor mode only.
OEM boot loader version	• The RSIP outputs the OEM boot loader version (OEM BL Ver.). • The MRAM sequencer supports Anti-Rollback Counter Protection using the OEM boot loader version, which is protected by the RSIP.
On-the-fly decryption	The RSIP outputs key data for on-the-fly decryption (DOTF) through a dedicated bus.
Preventing tampering	Countermeasures against side channel attacks, including SPA/DPA and timing attacks, are available.
Low power consumption	Setting of the module stop state is possible.
Protect Zone filter	Security attribution can be set.

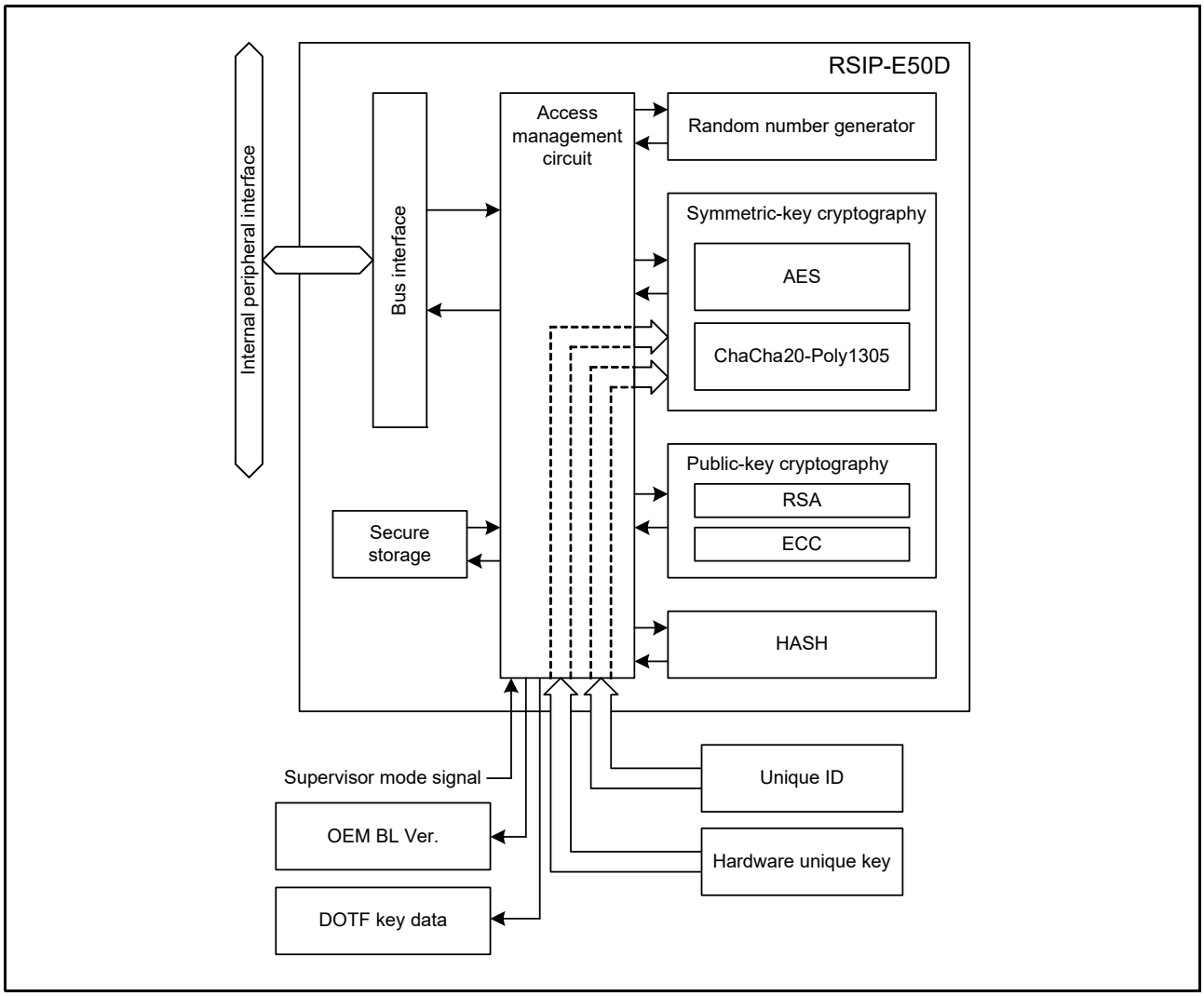


Figure 57.1 RSIP Block Diagram

57.2 Operation

57.2.1 Encryption Engine

Figure 57.2 shows processes of the encryption engine integrated in the RSIP.
The encryption engine, using the wrapped user key, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is user key or intermediate data ever exposed outside of the RSIP.

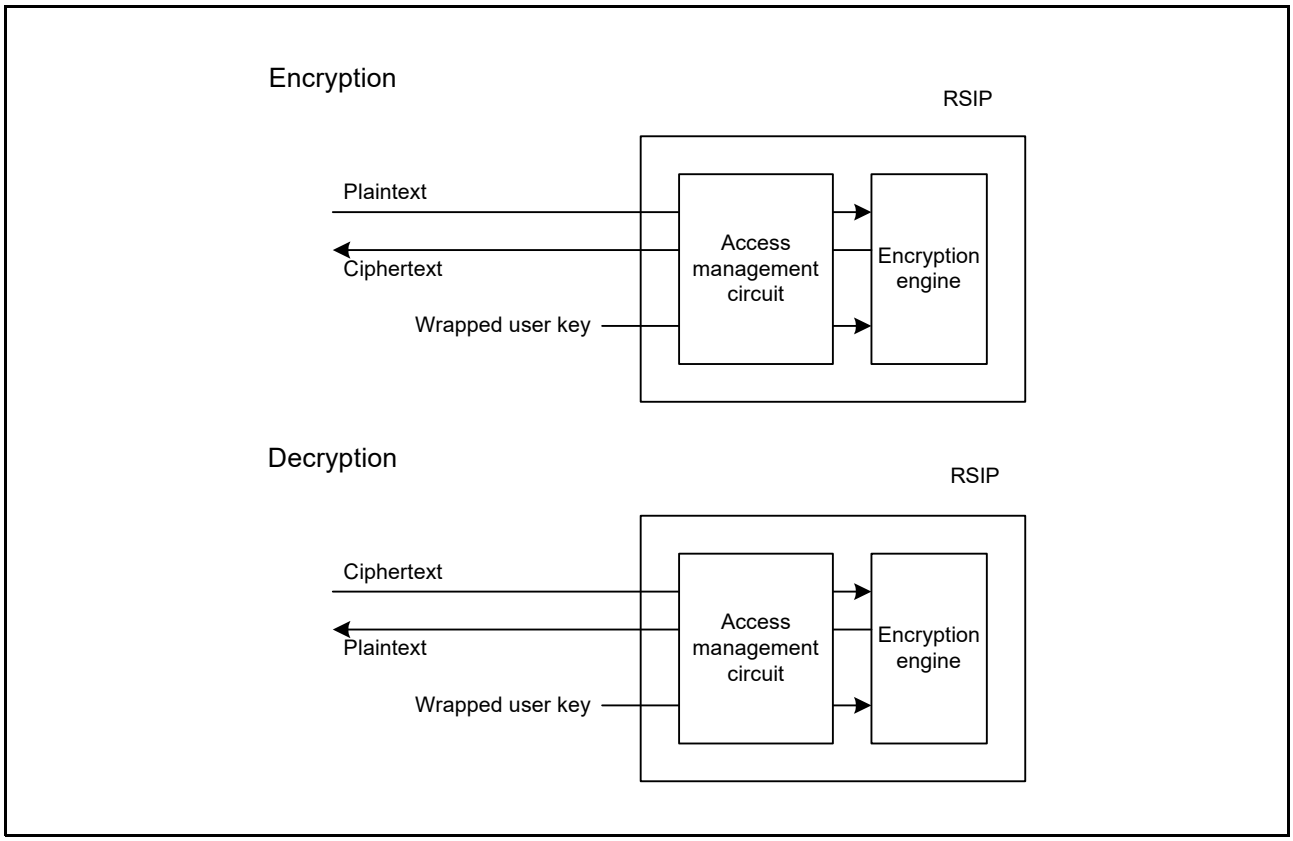


Figure 57.2 Encryption and Decryption processes by Encryption Engine

57.2.2 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the wrapped user key into the RSIP, and recover the user key.
(2) Input the data to encrypt or decrypt into the RSIP. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 57.3 shows the timing diagram.

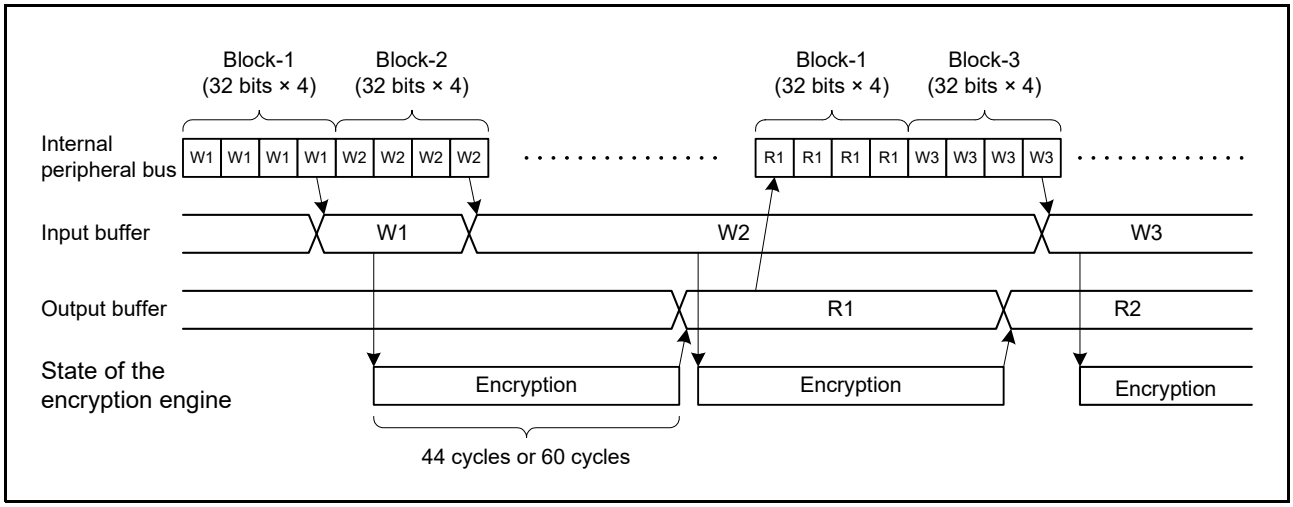


Figure 57.3 Encryption and Decryption Timing Diagram

57.3 Usage Notes

57.3.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only when the encryption engine is stopped.

57.3.2 Setting the Module Stop Function

The module stop control register C (MSTPCRC) enables or disables operation of the RSIP. After a reset, the RSIP is stopped. After exiting the module stop state, the RSIP can be accessed. Refer to section 11, Low Power Consumption for details.

57.3.3 Protection against SPA/DPA

Protection against SPA/DPA can be enabled in the encryption/decryption process.

57.3.4 RSIP Library

Use of the RSIP requires the RSIP library provided by Renesas Electronics.

58. Pulse-Density Modulation Interface (PDM-IF)

58.1 Overview

This MCU has a built-in, one-channel pulse-density modulation interface (PDM-IF). It outputs pulse-density modulation signals that can be connected to an external microphone.
The PDM-IF filters a 1-bit digital data stream with its pulse density modulated at a high sampling rate and converts it into 20-bit or 16-bit digital data at a lower sampling rate.
In this section, PCLK refers to PCLKB.
Table 58.1 lists the specifications for the PDM-IF. Figure 58.1 shows a block diagram of the PDM-IF.

Table 58.1 PDM-IF Specifications

Item	Description
Number of channels	1 channel
Clock	Core clock: PDMIFCLK Bus clock: PCLK
Functions	• Filters a 1-bit digital input data stream (PDMDAT0) and converts it into 20-bit or 16-bit digital data. The bit order is little-endian. • Supports audio active detection. • Contains programmable filters such as a 4th-order Sinc filter, high-pass filter (for DC bias suppression), compensation filter (for Sinc passband distortion), and half-band decimation filter (for aliasing distortion). – The decimation ratio is flexibly programmable. – A Sinc filter can be selected from 1st, 2nd, 3rd, and 4th order. • DMA transfer via the APB is possible. • Internal buffer – The buffer size is 32-stage. – Audio data can be stored in low power consumption mode. • Error detection function available for debugging
Interrupt sources	Three sources • Data receive interrupt • Audio detection interrupt • Error detection interrupt
Lower power consumption function	• Supports low power consumption configuration (low clock speed) of microphones. Can switch to high-speed clock after audio detection. • Bus clock can be stopped in low power consumption mode*1.
Protect Zone filter	Security attribution can be set*2.

Note 1. When performing audio detection in software standby mode, set the SYSC.MOCOSCR.MOCOSOKP bit to 1 to enable the core clock to operate even in software standby mode.
Note 2. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The PDM-IF consists of the following blocks:

- Clock control: Generates PDMCLK0 for the microphone and internal timing signals for the Sinc filter.
- Data input control: Receives data input from the microphone to the PDMDAT0 pin.
- Sinc filter: Converts a 1-bit digital data stream input from the PDMDAT0 pin into 34-bit signed data, and then clips the 34-bit signed data into 20-bit signed data.
- High-pass filter: Filter for suppressing DC bias
- Compensation filter: Filter for Sinc passband distortion
- Low-pass (half-band decimation) filter: Filter for aliasing distortion
- Moving average filter: Simple filter for reducing noise before audio detection
- Audio detection: Generates a wake-up interrupt and returns from low power consumption mode.
- Data buffer: Stores filtered 20-bit or 16-bit data.

- Short-circuit detection: Detects a short-circuit error indicating a continuous input of 0 or 1 from the PDMDAT0 pin.
- Overvoltage detection: Detects overvoltage errors.
- Error control: Generates error flags from other blocks.

The 1-bit digital data stream input from the PDMDAT0 pin is converted to 34-bit signed data by a Sinc filter and then clipped to 20-bit signed data. The 20-bit signed data is used for audio detection and data storage in the buffer.
The audio detection block receives the 20-bit signed data through a high-pass filter block and a moving average filter block. This block compares the received data with the set threshold values and generates a wake-up interrupt when the data exceeds the upper threshold or falls below the lower threshold.
The data buffer block receives 20-bit signed data through a high-pass filter, compensation filter, and low-pass filter. The block stores 20-bit signed data or clipped 16-bit signed data. Then, when an APB read access occurs, the block sends 20-bit signed data or 16-bit signed data to the APB-IF.

Table 58.2 PDM-IF I/O Pins

Pin Name	I/O	Function
PDMCLK0	Output	Clock output pin
PDMDAT0	Input	Data input pins

Figure 58.1 shows a block diagram of the PDM-IF.

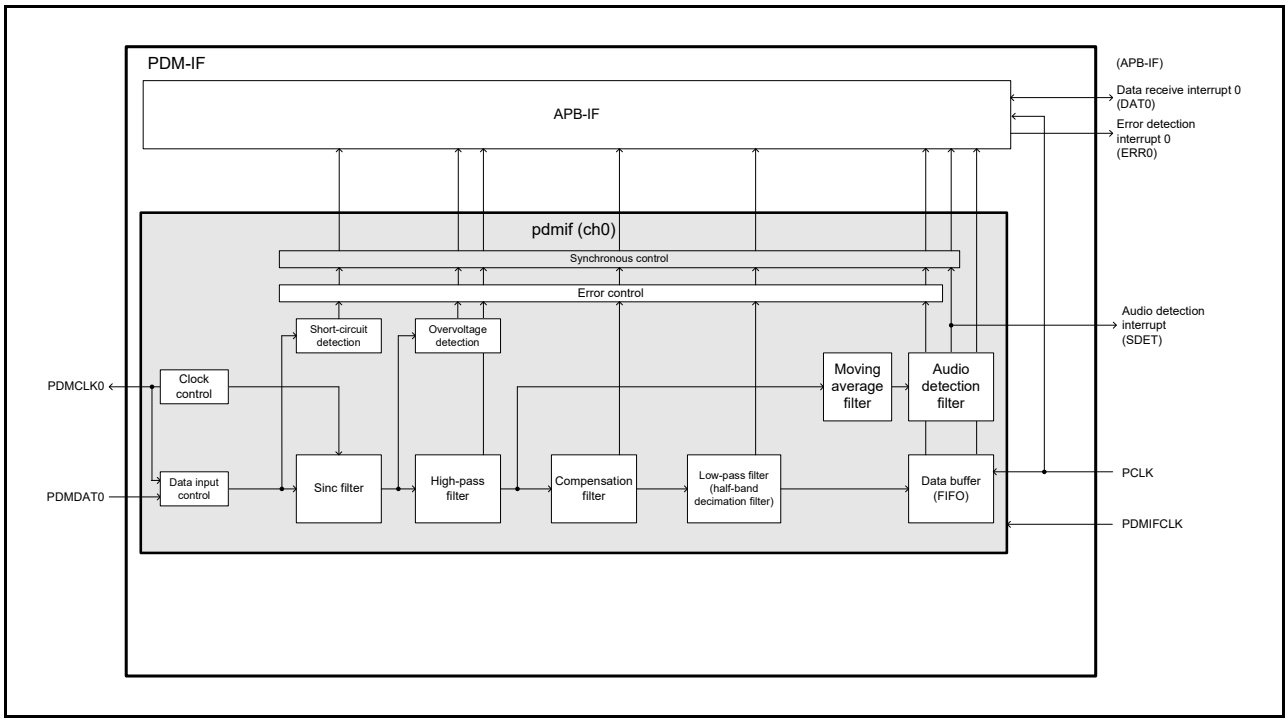


Figure 58.1 PDMIF Block Diagram

58.2 Register Descriptions

58.2.1 Register Map

Table 58.3 Register Map (1/4)

Base Address Symbol*1	Offset Address	Register Symbol	Register Name	R/W	Initial Value	Writable Period*2				
						(1) Start Flow	(2) Stop Flow	(3) Normal Processing Flow	(4) Low Power Consumption Mode Transition Flow	(5) Normal Mode Transition Flow
PDMIF	000h	PDCSTRTR	Channel Software Start Trigger Register	W	00000000h	✓	—	—	—	—
PDMIF	004h	PDCSTPTR	Channel Software Stop Trigger Register	W	00000000h	—	✓	—	—	—
PDMIF	008h	PDCCHGTR	Channel Software Change Trigger Register	W	00000000h	—	—	—	✓	✓
PDMIF	00Ch	PDCICR	Channel Interrupt Control Register	R/W	00000000h	✓	✓	—	✓	✓
PDMIF	010h	PDCSR	Channel Status Register	R	00000000h	—	—	—	—	—
PDMIF	014h	PDCSCR	Channel Status Clear Register	W	00000000h	✓	✓	—	✓	✓
PDMIF	020h	PDCSDCR	Channel Audio Detection Control Register	R/W	00000000h	✓	✓	—	✓	✓
PDMIF	024h	PDCDRCR	Channel Data Read Control Register	R/W	00000000h	—	—	✓	—	—
PDMIF	028h	PDCDCR	Channel Data Clear Register	W	00000000h	—	—	✓	—	—
PDMIF	080h	PDVR	Version Register	R	00000010h	—	—	—	—	—
PDMIF	100h	PDSTRTRCH0	Software Start Trigger Register Channel 0	W	00000000h	✓	—	—	—	—
PDMIF	104h	PDSTPTRCH0	Software Stop Trigger Register Channel 0	W	00000000h	—	✓	—	—	—
PDMIF	108h	PDCHGTRCH0	Software Change Trigger Register Channel 0	W	00000000h	—	—	—	✓	✓
PDMIF	10Ch	PDICRCH0	Interrupt Control Register Channel 0	R/W	00000000h	✓	✓	—	✓	✓
PDMIF	110h	PDSDCRCH0	Status Detection Control Register Channel 0	R/W	00000000h	✓	✓	—	—	—
PDMIF	114h	PDSRCH0	Status Register Channel 0	R	00000000h	—	—	—	—	—
PDMIF	118h	PDSCRCH0	Status Clear Register Channel 0	W	00000000h	✓	✓	—	✓	✓
PDMIF	120h	PDMDSRCH0	Mode Setting Register Channel 0	R/W	00000000h	✓	—	—	—	—
PDMIF	124h	PDSFCRCH0	Sinc Filter Control Register Channel 0	R/W	057C0000h	✓	—	—	✓	✓
PDMIF	128h	PDHFCS0RCH0	High-Pass Filter Coefficient s (0) Register Channel 0	R/W	00003F61h	✓	—	—	—	—
PDMIF	12Ch	PDHFCK1RCH0	High-Pass Filter Coefficient k (1) Register Channel 0	R/W	00003EC1h	✓	—	—	—	—
PDMIF	130h	PDHFCH0RCH0	High-Pass Filter Coefficient h (0) Register Channel 0	R/W	00004000h	✓	—	—	—	—
PDMIF	134h	PDHFCH1RCH0	High-Pass Filter Coefficient h (1) Register Channel 0	R/W	0000C000h	✓	—	—	—	—

Table 58.3 Register Map (2/4)

Base Address Symbol*1	Offset Address	Register Symbol	Register Name	R/W	Initial Value	Writable Period*2				
						(1) Start Flow	(2) Stop Flow	(3) Normal Processing Flow	(4) Low Power Consumption Mode Transition Flow	(5) Normal Mode Transition Flow
PDMIF	138h	PDCFCH00RCH0	Compensation Filter Coefficient h (0) Register Channel 0	R/W	00001FE8h	✓	—	—	—	—
PDMIF	13Ch	PDCFCH01RCH0	Compensation Filter Coefficient h (1) Register Channel 0	R/W	00000039h	✓	—	—	—	—
PDMIF	140h	PDCFCH02RCH0	Compensation Filter Coefficient h (2) Register Channel 0	R/W	0000003Ch	✓	—	—	—	—
PDMIF	144h	PDCFCH03RCH0	Compensation Filter Coefficient h (3) Register Channel 0	R/W	00001E56h	✓	—	—	—	—
PDMIF	148h	PDCFCH04RCH0	Compensation Filter Coefficient h (4) Register Channel 0	R/W	000001DCh	✓	—	—	—	—
PDMIF	14Ch	PDCFCH05RCH0	Compensation Filter Coefficient h (5) Register Channel 0	R/W	000006E1h	✓	—	—	—	—
PDMIF	150h	PDCFCH06RCH0	Compensation Filter Coefficient h (6) Register Channel 0	R/W	000001DCh	✓	—	—	—	—
PDMIF	154h	PDCFCH07RCH0	Compensation Filter Coefficient h (7) Register Channel 0	R/W	00001E56h	✓	—	—	—	—
PDMIF	158h	PDCFCH08RCH0	Compensation Filter Coefficient h (8) Register Channel 0	R/W	0000003Ch	✓	—	—	—	—
PDMIF	15Ch	PDCFCH09RCH0	Compensation Filter Coefficient h (9) Register Channel 0	R/W	00000039h	✓	—	—	—	—
PDMIF	160h	PDCFCH10RCH0	Compensation Filter Coefficient h (10) Register Channel 0	R/W	00001FE8h	✓	—	—	—	—
PDMIF	164h	PDLFCH010RCH0	Low-Pass Filter Coefficient h0 (10) Register Channel 0	R/W	00000400h	✓	—	—	—	—
PDMIF	168h	PDLFCH100RCH0	Low-Pass Filter Coefficient h1 (0) Register Channel 0	R/W	00001FF8h	✓	—	—	—	—
PDMIF	16Ch	PDLFCH101RCH0	Low-Pass Filter Coefficient h1 (1) Register Channel 0	R/W	0000000Ah	✓	—	—	—	—
PDMIF	170h	PDLFCH102RCH0	Low-Pass Filter Coefficient h1 (2) Register Channel 0	R/W	00001FF0h	✓	—	—	—	—
PDMIF	174h	PDLFCH103RCH0	Low-Pass Filter Coefficient h1 (3) Register Channel 0	R/W	00000018h	✓	—	—	—	—
PDMIF	178h	PDLFCH104RCH0	Low-Pass Filter Coefficient h1 (4) Register Channel 0	R/W	00001FDCh	✓	—	—	—	—
PDMIF	17Ch	PDLFCH105RCH0	Low-Pass Filter Coefficient h1 (5) Register Channel 0	R/W	00000034h	✓	—	—	—	—
PDMIF	180h	PDLFCH106RCH0	Low-Pass Filter Coefficient h1 (6) Register Channel 0	R/W	00001FB3h	✓	—	—	—	—

Table 58.3 Register Map (3/4)

Base Address Symbol*1	Offset Address	Register Symbol	Register Name	R/W	Initial Value	Writable Period*2				
						(1) Start Flow	(2) Stop Flow	(3) Normal Processing Flow	(4) Low Power Consumption Mode Transition Flow	(5) Normal Mode Transition Flow
PDMIF	184h	PDLFCH107RCH0	Low-Pass Filter Coefficient h1 (7) Register Channel 0	R/W	00000076h	✓	—	—	—	—
PDMIF	188h	PDLFCH108RCH0	Low-Pass Filter Coefficient h1 (8) Register Channel 0	R/W	00001F2Eh	✓	—	—	—	—
PDMIF	18Ch	PDLFCH109RCH0	Low-Pass Filter Coefficient h1 (9) Register Channel 0	R/W	00000289h	✓	—	—	—	—
PDMIF	190h	PDLFCH110RCH0	Low-Pass Filter Coefficient h1 (10) Register Channel 0	R/W	00000289h	✓	—	—	—	—
PDMIF	194h	PDLFCH111RCH0	Low-Pass Filter Coefficient h1 (11) Register Channel 0	R/W	00001F2Eh	✓	—	—	—	—
PDMIF	198h	PDLFCH112RCH0	Low-Pass Filter Coefficient h1 (12) Register Channel 0	R/W	00000076h	✓	—	—	—	—
PDMIF	19Ch	PDLFCH113RCH0	Low-Pass Filter Coefficient h1 (13) Register Channel 0	R/W	00001FB3h	✓	—	—	—	—
PDMIF	1A0h	PDLFCH114RCH0	Low-Pass Filter Coefficient h1 (14) Register Channel 0	R/W	00000034h	✓	—	—	—	—
PDMIF	1A4h	PDLFCH115RCH0	Low-Pass Filter Coefficient h1 (15) Register Channel 0	R/W	00001FDCh	✓	—	—	—	—
PDMIF	1A8h	PDLFCH116RCH0	Low-Pass Filter Coefficient h1 (16) Register Channel 0	R/W	00000018h	✓	—	—	—	—
PDMIF	1ACh	PDLFCH117RCH0	Low-Pass Filter Coefficient h1 (17) Register Channel 0	R/W	00001FF0h	✓	—	—	—	—
PDMIF	1B0h	PDLFCH118RCH0	Low-Pass Filter Coefficient h1 (18) Register Channel 0	R/W	0000000Ah	✓	—	—	—	—
PDMIF	1B4h	PDLFCH119RCH0	Low-Pass Filter Coefficient h1 (19) Register Channel 0	R/W	00001FF8h	✓	—	—	—	—
PDMIF	1B8h	PDSDLTRCH0	Audio Detection Lower Threshold Register Channel 0	R/W	00000000h	✓	—	—	✓	—
PDMIF	1BCh	PDSDUTRCH0	Audio Detection Upper Threshold Register Channel 0	R/W	00000000h	✓	—	—	✓	—
PDMIF	1C0h	PDDBCRCH0	Data Buffer Control Register Channel 0	R/W	00000000h	✓	—	—	—	—
PDMIF	1C4h	PDSCTSRCH0	Short-Circuit Threshold Setting Register Channel 0	R/W	00000000h	✓	—	—	—	—
PDMIF	1C8h	PDOVLTRCH0	Overvoltage Lower Threshold Register Channel 0	R/W	00000000h	✓	—	—	—	—
PDMIF	1CCh	PDOVUTRCH0	Overvoltage Upper Threshold Register Channel 0	R/W	00000000h	✓	—	—	—	—

Table 58.3 Register Map (4/4)

Base Address Symbol*1	Offset Address	Register Symbol	Register Name	R/W	Initial Value	Writable Period*2				
						(1) Start Flow	(2) Stop Flow	(3) Normal Processing Flow	(4) Low Power Consumption Mode Transition Flow	(5) Normal Mode Transition Flow
PDMIF	1E0h	PDDRCRCH0	Data Read Control Register Channel 0	R/W	00000000h	—	—	✓	—	—
PDMIF	1E4h	PDDCRCH0	Data Clear Register Channel 0	W	00000000h	—	—	✓	—	—
PDMIF	1E8h	PDDRRCH0	Data Read Register Channel 0	R	00000000h	—	—	—	—	—
PDMIF	1ECh	PDDSRCH0	Data Status Register Channel 0	R	00000000h	—	—	—	—	—

Note: The 10 registers with offset addresses 000h to 080h are common registers. Common registers, excluding the Version Register, are registers used for synchronous control in multi-channel products and have mirroring registers for each channel. For more information on the mirroring registers, see section 58.2.67, Mirroring Registers.

Note 1. The base address of the PDMIF is 8785 6000h.

Note 2. For details on writing to each register, see section 58.4, Setting Flows.

58.2.2 Channel Software Start Trigger Register (PDCSTRTR)

Address(es): PDMIF.PDCSTRTR 8785 6000h, PDMIF_NS.PDCSTRTR 0785 6000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STRTRG0	Channel 0 Start Trigger	0: No operation 1: Starts channel 0.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is used to set the start trigger.

58.2.3 Channel Software Stop Trigger Register (PDCSTPTR)

Address(es): PDMIF.PDCSTPTR 8785 6004h, PDMIF_NS.PDCSTPTR 0785 6004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STPTRG0	Channel 0 Stop Trigger	0: No operation 1: Stops channel 0.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is used to set the stop trigger.

58.2.4 Channel Software Change Trigger Register (PDCCHGTR)

Address(es): PDMIF.PDCCHGTR 8785 6008h, PDMIF_NS.PDCCHGTR 0785 6008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHGTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHGTRG0	Channel 0 Change Trigger	0: No operation 1: Changes the clock (PDMCLK0) settings.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is used to set the change trigger for the clock (PDMCLK0) settings.

58.2.5 Channel Interrupt Control Register (PDCICR)

Address(es): PDMIF.PDCICR 8785 600Ch, PDMIF_NS.PDCICR 0785 600Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	IEDE0	—	—	—	—	—	—	—	IDRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ISDE0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ISDE0	Channel 0 Audio Detection Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IDRE0	Channel 0 Data Receive Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	IEDE0	Channel 0 Error Detection Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the interrupt outputs.

58.2.6 Channel Status Register (PDCSR)

Address(es): PDMIF.PDCSR 8785 6010h, PDMIF_NS.PDCSR 0785 6010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	EDF0	—	—	—	—	—	—	—	DRF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDF0	—	—	—	—	—	—	—	STATE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STATE0	Channel 0 Status Flag	0: Stopped 1: Operating	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	SDF0	Channel 0 Audio Detection Flag	0: No audio detected above the threshold 1: Audio detected above the threshold	R
b15 to b9	—	Reserved	These bits are read as 0.	R
b16	DRF0	Channel 0 Data Receive Flag	0: The amount of data stored in the buffer is less than the threshold. 1: The amount of data stored in the buffer exceeds the threshold.	R
b23 to b17	—	Reserved	These bits are read as 0.	R
b24	EDF0	Channel 0 Error Detection Flag	0: No error detected 1: Error detected	R
b31 to b25	—	Reserved	These bits are read as 0.	R

This status register indicates the channel status.

STATE0 Flag (Channel 0 Status Flag)

This flag indicates the operating status of channel 0. For details, see section 58.3.2, Channel Start and Channel Stop.

[Setting condition]

- When 1 is written to the PDCSTRTR.STRTRG0 bit

[Clearing condition]

- When 1 is written to the PDCSTPTR.STPTRG0 bit

SDF0 Flag (Channel 0 Audio Detection Flag)

This flag indicates the audio detection status of channel 0. For details, see section 58.3.5, Audio Detection.

[Setting conditions]

- When the moving average filter result is greater than or equal to the value of the PDSDUTR.SDETU[19:0] bits with PDCSDCR.SDE0 = 1
- When the moving average filter result is less than or equal to the value of the PDSDLTR.SDETL[19:0] bits with PDCSDCR.SDE0 = 1

[Clearing condition]

- When 1 is written to the PDCSCR.SDFC0 bit

DRF0 Flag (Channel 0 Data Receive Flag)

This flag indicates the data receive status of channel 0. For details, see section 58.3.6, Data Buffer.

[Setting condition]

- When the value of the PDDSRCH0.DATNUM[7:0] bits is greater than or equal to the threshold set by the PDDBCRCH0.DATRITHR[2:0] bits with PDCDRCR.DATRE0 = 1

[Clearing condition]

- When PDCDRCR.DATRE0 = 0 or the value of the PDDSRCH0.DATNUM[7:0] bits is less than the threshold

EDF0 Flag (Channel 0 Error Detection Flag)

This flag indicates the error detection status of channel 0.

[Setting condition]

- When an error has occurred For error types, see section 58.2.17, Status Register Channel 0 (PDSRCH0).

[Clearing condition]

- When all error flags are cleared by writing to the PDSCRCH0 register

58.2.7 Channel Status Clear Register (PDCSCR)

Address(es): PDMIF.PDCSCR 8785 6014h, PDMIF_NS.PDCSCR 0785 6014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDFC0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	The write value should be 0.	W
b8	SDFC0	Channel 0 Audio Detection Flag Clear	0: No operation 1: Clears the PDCSR.SDF0 flag.	W
b31 to b9	—	Reserved	The write value should be 0.	W

This register is used to clear the audio detection flag of channel 0.

58.2.8 Channel Audio Detection Control Register (PDCSDCR)

Address(es): PDMIF.PDCSDCR 8785 6020h, PDMIF_NS.PDCSDCR 0785 6020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SDE0	Channel 0 Audio Detection Enable	0: Disables audio detection. 1: Enables audio detection.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control audio detection.

58.2.9 Channel Data Read Control Register (PDCDRCR)

Address(es): PDMIF.PDCDRCR 8785 6024h, PDMIF_NS.PDCDRCR 0785 6024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DATRE0	Channel 0 Data Read Enable	0: Disables reading from the data buffer. 1: Enables reading from the data buffer.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the reading of data.

58.2.10 Channel Data Clear Register (PDCDCR)

Address(es): PDMIF.PDCDCR 8785 6028h, PDMIF_NS.PDCDCR 0785 6028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DATC0	Channel 0 Data Clear	0: No operation 1: Clears data.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is used to clear the data of channel 0.

58.2.11 Version Register (PDVR)

Address(es): PDMIF.PDVR 8785 6080h, PDMIF_NS.PDVR 0785 6080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	VER[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	VER[11:0]	Version	Indicates the version of the PDM-IF.	R
b31 to b12	—	Reserved	These bits are read as 0.	R

This register indicates the IP version of the PDM-IF.

58.2.12 Software Start Trigger Register Channel 0 (PDSTRTRCH0)

Address(es): PDMIF.PDSTRTRCH0 8785 6100h, PDMIF_NS.PDSTRTRCH0 0785 6100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STRTRG	Start Trigger	0: No operation 1: Starts the channel.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is a mirroring register of the PDCSTRTR register that is used to set the start trigger.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.13 Software Stop Trigger Register Channel 0 (PDSTPTRCH0)

Address(es): PDMIF.PDSTPTRCH0 8785 6104h, PDMIF_NS.PDSTPTRCH0 0785 6104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STPTRG	Stop Trigger	0: No operation 1: Stops the channel.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is a mirroring register of the PDCSTPTR register that is used to set the stop trigger.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.14 Software Change Trigger Register Channel 0 (PDCHGTRCH0)

Address(es): PDMIF.PDCHGTRCH0 8785 6108h, PDMIF_NS.PDCHGTRCH0 0785 6108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHGTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CHGTRG	Change Trigger	0: No operation 1: Changes the clock (PDMCLK0) settings.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is a mirroring register of the PDCCHGTR register that is used to change the clock settings.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.15 Interrupt Control Register Channel 0 (PDICRCH0)

Address(es): PDMIF.PDICRCH0 8785 610Ch, PDMIF_NS.PDICRCH0 0785 610Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IEDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	IDRE	ISDE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ISDE	Audio Detection Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b2	IDRE	Data Receive Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IEDE	Error Detection Interrupt Enable	0: Disables interrupt output. 1: Enables interrupt output.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a mirroring register of the PDCICR register that is used to control the interrupt outputs.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.16 Status Detection Control Register Channel 0 (PDSDCRCH0)

Address(es): PDMIF.PDSDCRCH0 8785 6110h, PDMIF_NS.PDSDCRCH0 0785 6110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	BFOWDE	—	—	—	—	—	—	—	—	OVUDE	OVLDE	SCDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SDE	Audio Detection Enable	0: Disables audio detection. 1: Enables audio detection.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SCDE	Short-Circuit Detection Enable	0: Disables short-circuit detection on the PDMDAT0 pin. 1: Enables short-circuit detection on the PDMDAT0 pin.	R/W
b17	OVLDE	Overvoltage Less-Than-Lower-Limit Detection Enable	0: Disables detection of less-than-lower-limit data. 1: Enables detection of less-than-lower-limit data.	R/W
b18	OVUDE	Overvoltage Upper-Limit-Exceeded Detection Enable	0: Disables detection of data exceeding the upper limit. 1: Enables detection of data exceeding the upper limit.	R/W
b26 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27	BFOWDE	Buffer Overwrite Detection Enable	0: Disables buffer overwrite detection. 1: Enables buffer overwrite detection.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a mirroring register of the PDCSDCR register that is used to control the audio detection, and it is also used to control the error detection functions.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.17 Status Register Channel 0 (PDSRCH0)

Address(es): PDMIF.PDSRCH0 8785 6114h, PDMIF_NS.PDSRCH0 0785 6114h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	BFOWDF	—	—	—	—	—	—	—	—	OVUDF	OVLDF	SCDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRF	SDF	STATE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STATE	Status Flag	0: Stopped 1: Operating	R
b1	SDF	Audio Detection Flag	0: No audio detected above the threshold 1: Audio detected above the threshold	R
b2	DRF	Data Receive Flag	0: The amount of data stored in the buffer is less than the threshold. 1: The amount of data stored in the buffer exceeds the threshold.	R
b15 to b3	—	Reserved	These bits are read as 0.	R
b16	SCDF	Short-Circuit Detection Flag	0: No short circuit detected at the PDMDAT0 pin 1: Short circuit detected at PDMDAT0 pin	R
b17	OVLDF	Overvoltage Less-Than-Lower-Limit Detection Flag	0: Data is not less than the lower limit. 1: Data is less than the lower limit.	R
b18	OVUDF	Overvoltage Upper-Limit-Exceeded Detection Flag	0: Data is not exceeding the upper limit. 1: Data is exceeding the upper limit.	R
b26 to b19	—	Reserved	These bits are read as 0.	R
b27	BFOWDF	Buffer Overwrite Detection Flag	0: No buffer overwrites have occurred. 1: Buffer overwrite has occurred.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

This register is a mirroring register of the PDCSR register that indicates the channel status.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

STATE Flag (Status Flag)

This flag indicates the operating status. For details, see section 58.3.2, Channel Start and Channel Stop.
[Setting condition]

- When 1 is written to the PDSTRTRCH0.STRTRG bit

[Clearing condition]

- When 1 is written to the PDSTPTRCH0.STPTRG bit

SDF Flag (Audio Detection Flag)

This flag indicates the audio detection status. For details, see section 58.3.5, Audio Detection.
[Setting conditions]

- When the moving average filter result is greater than or equal to the value of the PDSDUTRCH0.SDETU[19:0] bits with PDSDCRCH0.SDE = 1
- When the moving average filter result is less than or equal to the value of the PDSDLTRCH0.SDETL[19:0] bits with PDSDCRCH0.SDE = 1

[Clearing condition]

- When 1 is written to the PDSCRCH0.SDFC bit

DRF Flag (Data Receive Flag)

This flag indicates the data receive status. For details, see section 58.3.6, Data Buffer.
[Setting condition]

- When the value of the PDDSRCH0.DATNUM[7:0] bits is greater than or equal to the threshold set by the PDDBCRCH0.DATRITHR[2:0] bits with PDDRCRCH0.DATRE = 1

[Clearing condition]

- When PDDRCRCH0.DATRE = 0 or the value of the PDDSRCH0.DATNUM[7:0] bits is less than the threshold set by the PDDBCRCH0.DATRITHR[2:0] bits

SCDF Flag (Short-Circuit Detection Flag)

This flag indicates that a short circuit has been detected at the PDMDAT0 pin. For details, see section 58.3.7, Short-Circuit Detection.
[Setting conditions]

- When the number of times that 1 is input consecutively to the PDMDAT0 pin exceeds the number set by PDSCTSRCH0.SCDH[12:0] bits with PDSDCRCH0.SCDE = 1
- When the number of times that 0 is input consecutively to the PDMDAT0 pin exceeds the number set by PDSCTSRCH0.SCDL[12:0] bits with PDSDCRCH0.SCDE = 1

[Clearing condition]

- When 1 is written to the PDSCRCH0.SCDFC bit

OVLDF Flag (Overvoltage Less-Than-Lower-Limit Detection Flag)

This flag indicates that the clipped Sinc filter result is below the lower limit for overvoltage detection. For details, see section 58.3.8, Overvoltage Detection.
[Setting condition]

- When the clipped Sinc filter result is smaller than the lower limit value (PDOVLTRCH0.OVDL[19:0]) with PDSDCRCH0.OVLDE = 1

[Clearing condition]

- When 1 is written to the PDSCRCH0.OVLDFC bit

OVUDF Flag (Overvoltage Upper-Limit-Exceeded Detection Flag)

This flag indicates that the clipped Sinc filter result exceeded the upper limit for overvoltage detection. For details, see section 58.3.8, Overvoltage Detection.
[Setting condition]

- When the clipped Sinc filter result is greater than the upper limit value (PDOVUTRCH0.OVDU[19:0]) with PDSDCRCH0.OVUDE = 1

[Clearing condition]

- When 1 is written to the PDSCRCH0.OVUDFC bit

BFOWDF Flag (Buffer Overwrite Detection Flag)

This flag indicates whether a buffer overwrite has occurred. For details, see section 58.3.6, Data Buffer.
[Setting condition]

- When data is written without reading data from the buffer when the buffer is full while PDSDCRCH0.BFOWDE = 1

[Clearing condition]

- When 1 is written to the PDSCRCH0.BFOWDFC bit

58.2.18 Status Clear Register Channel 0 (PDSCRCH0)

Address(es): PDMIF.PDSCRCH0 8785 6118h, PDMIF_NS.PDSCRCH0 0785 6118h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	BFOWDFC	—	—	—	—	—	—	—	—	OVUDFC	OVLDFC	SCDFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDFC	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	The write value should be 0.	W
b1	SDFC	Audio Detection Flag Clear	0: No operation 1: Clears the PDSRCH0.SDF flag.	W
b15 to b2	—	Reserved	The write value should be 0.	W
b16	SCDFC	Short-Circuit Detection Flag Clear	0: No operation 1: Clears the PDSRCH0.SCDF flag.	W
b17	OVLDFC	Overvoltage Less-Than-Lower-Limit Detection Clear	0: No operation 1: Clears the PDSRCH0.OVLDF flag.	W
b18	OVUDFC	Overvoltage Upper-Limit-Exceeded Detection Flag Clear	0: No operation 1: Clears the PDSRCH0.OVUDF flag.	W
b26 to b19	—	Reserved	The write value should be 0.	W
b27	BFOWDFC	Buffer Overwrite Detection Flag Clear	0: No operation 1: Clears the PDSRCH0.BFOWDF flag.	W
b31 to b28	—	Reserved	The write value should be 0.	W

This register is used to clear each error detection flag of the PDSRCH0 register.
Note that the SDFC bit is a mirroring register of the PDCSCR.SDFC0 bit.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.19 Mode Setting Register Channel 0 (PDMDSRCH0)

Address(es): PDMIF.PDMDSRCH0 8785 6120h, PDMIF_NS.PDMDSRCH0 0785 6120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DBIS[3:0]				—	—	SDMAMD[1:0]		—	—	—	—	—	—	LFIS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CFIS[1:0]		—	—	HFIS[1:0]		—	SFMD[2:0]			—	—	—	INPSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INPSEL	Input Data Select	0: Rising-edge data 1: Falling-edge data	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	SFMD[2:0]	Sinc Filter Mode Setting	b6 b4 0 0 1: 1st order 0 1 0: 2nd order 0 1 1: 3rd order Other than above: 4th order (default)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9 to b8	HFIS[1:0]	High-Pass Filter Input Shift Setting	b9 b8 0 0: No shift 0 1: Right-shifted by 1 bit 1 0: Right-shifted by 2 bits 1 1: Right-shifted by 3 bits	R/W
b11 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b12	CFIS[1:0]	Compensation Filter Input Shift Setting	b13 b12 0 0: No shift 0 1: Right-shifted by 1 bit 1 0: Right-shifted by 2 bits 1 1: Right-shifted by 3 bits	R/W
b15 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17 to b16	LFIS[1:0]	Low-Pass (Half-Band Decimation) Filter Input Shift Setting	b17 b16 0 0: No shift 0 1: Right-shifted by 1 bit 1 0: Right-shifted by 2 bits 1 1: Right-shifted by 3 bits	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b24	SDMAMD[1:0]	Moving Average Mode Select for Audio Detection Data	b25 b26 0 0: 1st order (filter skipped) (default) 0 1: 2nd order 1 0: 4th order 1 1: Setting prohibited	R/W
b27 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DBIS[3:0]	Data Buffer Input Shift Setting	S: Signed bit b31 b28 0 0 0 0: 20-bit mode, {1{S},[18:0]} 0 0 0 1: 20-bit mode, {2{S},[18:1]} 0 0 1 0: 20-bit mode, {3{S},[18:2]} 0 0 1 1: 20-bit mode, {4{S},[18:3]} 1 0 0 0: 16-bit mode, {S,D[18:4]} 1 0 0 1: 16-bit mode, {S,D[17:3]} 1 0 1 0: 16-bit mode, {S,D[16:2]} 1 0 1 1: 16-bit mode, {S,D[15:1]} 1 1 0 0: 16-bit mode, {S,D[14:0]} Other than above: Settings prohibited	R/W

This register is used to set the operating mode of the PDM-IF.

58.2.20 Sinc Filter Control Register Channel 0 (PDSFCRCH0)

Address(es): PDMIF.PDSFCRCH0 8785 6124h, PDMIF_NS.PDSFCRCH0 0785 6124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SINCRNG[4:0]					SINCDEC[7:0]							
Value after reset:	0	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CKDIV[3:0]	PDMCLK0 Core Clock Division Ratio Setting	b3 b0 0 0 0 0: Divided by 2 (default) 0 0 0 1: Divided by 4 0 0 1 0: Divided by 6 : 1 1 0 1: Divided by 28 1 1 1 0: Divided by 30 1 1 1 1: Divided by 32	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	SINCDEC[7:0]	Sinc Filter Decimation Ratio Setting	M: Decimation ratio Settings less than M = 4 are prohibited. When such values are set, Sinc filter operation is not guaranteed. Set so that D×M > 12. The operation of other filters is not guaranteed if this condition is not satisfied. (D: PDMCLK0 division ratio) Decimation ratio M is set as follows: M = SINCDEC[7:0] + 1 00h: M = 1 (Setting prohibited) 01h: M = 2 (Setting prohibited) 02h: M = 3 (Setting prohibited) 03h: M = 4 : 7Ch: M = 125 (Default) : FFh: M = 256	R/W

Bit	Symbol	Bit Name	Description	R/W
b28 to b24	SINCRNG[4:0]	Sinc Filter Output Effective Range Setting	When an overflow occurs, 7FFFFh is output. When an underflow occurs, 80000h is output. S: Signed bit 00h: {S, [32:14]} 01h: {S, [31:13]} 02h: {S, [30:12]} 03h: {S, [29:11]} 04h: {S, [28:10]} 05h: {S, [27:9]} (Default) 06h: {S, [26:8]} 07h: {S, [25:7]} 08h: {S, [24:6]} 09h: {S, [23:5]} 0Ah: {S, [22:4]} 0Bh: {S, [21:3]} 0Ch: {S, [20:2]} 0Dh: {S, [19:1]} 0Eh: {S, [18:0]} 0Fh: {S, [17:0], 0} 10h: {S, [16:0], 00} 11h: {S, [15:0], 000} 12h: {S, [14:0], 0000} 13h: {S, [13:0], 00000} 14h: {S, [12:0], 000000} 15h: {S, [11:0], 0000000} 16h: {S, [10:0], 00000000} 17h: {S, [9:0], 000000000} 18h: {S, [8:0], 0000000000} 19h: {S, [7:0], 00000000000} 1Ah: {S, [6:0], 000000000000} 1Bh: {S, [5:0], 0000000000000} 1Ch: {S, [4:0], 00000000000000} 1Dh: {S, [3:0], 000000000000000} 1Eh: {S, [2:0], 0000000000000000} 1Fh: {S, [1:0], 00000000000000000} (Setting prohibited)	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the Sinc filter.

58.2.21 High-Pass Filter Coefficient s (0) Register Channel 0 (PDHFCS0RCH0)

Address(es): PDMIF.PDHFCS0RCH0 8785 6128h, PDMIF_NS.PDHFCS0RCH0 0785 6128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HFS[15:0]															
Value after reset:	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	HFS[15:0]	High-Pass Filter Coefficient s (0)	16-bit signed data (fractional bits [13:0])	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a high-pass filter coefficient.

58.2.22 High-Pass Filter Coefficient k (1) Register Channel 0 (PDHFCK1RCH0)

Address(es): PDMIF.PDHFCK1RCH0 8785 612Ch, PDMIF_NS.PDHFCK1RCH0 0785 612Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HFK[15:0]															
Value after reset:	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	HFK[15:0]	High-Pass Filter Coefficient k (1)	16-bit signed data (fractional bits [13:0])	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a high-pass filter coefficient.

58.2.23 High-Pass Filter Coefficient h (0) Register Channel 0 (PDHFCH0RCH0)

Address(es): PDMIF.PDHFCH0RCH0 8785 6130h, PDMIF_NS.PDHFCH0RCH0 0785 6130h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HFH[15:0]															
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	HFH[15:0]	High-Pass Filter Coefficient h (0)	16-bit signed data (fractional bits [13:0])	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a high-pass filter coefficient.

58.2.24 High-Pass Filter Coefficient h (1) Register Channel 0 (PDHFCH1RCH0)

Address(es): PDMIF.PDHFCH1RCH0 8785 6134h, PDMIF_NS.PDHFCH1RCH0 0785 6134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HFH[15:0]															
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	HFH[15:0]	High-Pass Filter Coefficient h (1)	16-bit signed data (fractional bits [13:0])	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a high-pass filter coefficient.

58.2.25 Compensation Filter Coefficient h (0) Register Channel 0 (PDCFCH00RCH0)

Address(es): PDMIF.PDCFCH00RCH0 8785 6138h, PDMIF_NS.PDCFCH00RCH0 0785 6138h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (0)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.26 Compensation Filter Coefficient h (1) Register Channel 0 (PDCFCH01RCH0)

Address(es): PDMIF.PDCFCH01RCH0 8785 613Ch, PDMIF_NS.PDCFCH01RCH0 0785 613Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (1)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.27 Compensation Filter Coefficient h (2) Register Channel 0 (PDCFCH02RCH0)

Address(es): PDMIF.PDCFCH02RCH0 8785 6140h, PDMIF_NS.PDCFCH02RCH0 0785 6140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (2)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.28 Compensation Filter Coefficient h (3) Register Channel 0 (PDCFCH03RCH0)

Address(es): PDMIF.PDCFCH03RCH0 8785 6144h, PDMIF_NS.PDCFCH03RCH0 0785 6144h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	1	1	1	1	0	0	1	0	1	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (3)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.29 Compensation Filter Coefficient h (4) Register Channel 0 (PDCFCH04RCH0)

Address(es): PDMIF.PDCFCH04RCH0 8785 6148h, PDMIF_NS.PDCFCH04RCH0 0785 6148h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (4)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.30 Compensation Filter Coefficient h (5) Register Channel 0 (PDCFCH05RCH0)

Address(es): PDMIF.PDCFCH05RCH0 8785 614Ch, PDMIF_NS.PDCFCH05RCH0 0785 614Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (5)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.31 Compensation Filter Coefficient h (6) Register Channel 0 (PDCFCH06RCH0)

Address(es): PDMIF.PDCFCH06RCH0 8785 6150h, PDMIF_NS.PDCFCH06RCH0 0785 6150h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (6)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.32 Compensation Filter Coefficient h (7) Register Channel 0 (PDCFCH07RCH0)

Address(es): PDMIF.PDCFCH07RCH0 8785 6154h, PDMIF_NS.PDCFCH07RCH0 0785 6154h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	1	1	1	1	0	0	1	0	1	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (7)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.33 Compensation Filter Coefficient h (8) Register Channel 0 (PDCFCH08RCH0)

Address(es): PDMIF.PDCFCH08RCH0 8785 6158h, PDMIF_NS.PDCFCH08RCH0 0785 6158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (8)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.34 Compensation Filter Coefficient h (9) Register Channel 0 (PDCFCH09RCH0)

Address(es): PDMIF.PDCFCH09RCH0 8785 615Ch, PDMIF_NS.PDCFCH09RCH0 0785 615Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (9)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.35 Compensation Filter Coefficient h (10) Register Channel 0 (PDCFCH10RCH0)

Address(es): PDMIF.PDCFCH10RCH0 8785 6160h, PDMIF_NS.PDCFCH10RCH0 0785 6160h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	CFH[12:0]	Compensation Filter Coefficient h (10)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set a compensation filter coefficient.

58.2.36 Low-Pass Filter Coefficient h0 (10) Register Channel 0 (PDLFCH010RCH0)

Address(es): PDMIF.PDLFCH010RCH0 8785 6164h, PDMIF_NS.PDLFCH010RCH0 0785 6164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH0[12:0]												
Value after reset*1:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH0[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h0 (10)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.37 Low-Pass Filter Coefficient h1 (0) Register Channel 0 (PDLFCH100RCH0)

Address(es): PDMIF.PDLFCH100RCH0 8785 6168h, PDMIF_NS.PDLFCH100RCH0 0785 6168h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (0)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.38 Low-Pass Filter Coefficient h1 (1) Register Channel 0 (PDLFCH101RCH0)

Address(es): PDMIF.PDLFCH101RCH0 8785 616Ch, PDMIF_NS.PDLFCH101RCH0 0785 616Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (1)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.39 Low-Pass Filter Coefficient h1 (2) Register Channel 0 (PDLFCH102RCH0)

Address(es): PDMIF.PDLFCH102RCH0 8785 6170h, PDMIF_NS.PDLFCH102RCH0 0785 6170h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (2)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.40 Low-Pass Filter Coefficient h1 (3) Register Channel 0 (PDLFCH103RCH0)

Address(es): PDMIF.PDLFCH103RCH0 8785 6174h, PDMIF_NS.PDLFCH103RCH0 0785 6174h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (3)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.41 Low-Pass Filter Coefficient h1 (4) Register Channel 0 (PDLFCH104RCH0)

Address(es): PDMIF.PDLFCH104RCH0 8785 6178h, PDMIF_NS.PDLFCH104RCH0 0785 6178h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (4)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.42 Low-Pass Filter Coefficient h1 (5) Register Channel 0 (PDLFCH105RCH0)

Address(es): PDMIF.PDLFCH105RCH0 8785 617Ch, PDMIF_NS.PDLFCH105RCH0 0785 617Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (5)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.43 Low-Pass Filter Coefficient h1 (6) Register Channel 0 (PDLFCH106RCH0)

Address(es): PDMIF.PDLFCH106RCH0 8785 6180h, PDMIF_NS.PDLFCH106RCH0 0785 6180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset:*1	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (6)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.44 Low-Pass Filter Coefficient h1 (7) Register Channel 0 (PDLFCH107RCH0)

Address(es): PDMIF.PDLFCH107RCH0 8785 6184h, PDMIF_NS.PDLFCH107RCH0 0785 6184h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (7)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.45 Low-Pass Filter Coefficient h1 (8) Register Channel 0 (PDLFCH108RCH0)

Address(es): PDMIF.PDLFCH108RCH0 8785 6188h, PDMIF_NS.PDLFCH108RCH0 0785 6188h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (8)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.46 Low-Pass Filter Coefficient h1 (9) Register Channel 0 (PDLFCH109RCH0)

Address(es): PDMIF.PDLFCH109RCH0 8785 618Ch, PDMIF_NS.PDLFCH109RCH0 0785 618Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (9)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.47 Low-Pass Filter Coefficient h1 (10) Register Channel 0 (PDLFCH110RCH0)

Address(es): PDMIF.PDLFCH110RCH0 8785 6190h, PDMIF_NS.PDLFCH110RCH0 0785 6190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (10)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.48 Low-Pass Filter Coefficient h1 (11) Register Channel 0 (PDLFCH111RCH0)

Address(es): PDMIF.PDLFCH111RCH0 8785 6194h, PDMIF_NS.PDLFCH111RCH0 0785 6194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (11)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.49 Low-Pass Filter Coefficient h1 (12) Register Channel 0 (PDLFCH112RCH0)

Address(es): PDMIF.PDLFCH112RCH0 8785 6198h, PDMIF_NS.PDLFCH112RCH0 0785 6198h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (12)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.50 Low-Pass Filter Coefficient h1 (13) Register Channel 0 (PDLFCH113RCH0)

Address(es): PDMIF.PDLFCH113RCH0 8785 619Ch, PDMIF_NS.PDLFCH113RCH0 0785 619Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (13)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.51 Low-Pass Filter Coefficient h1 (14) Register Channel 0 (PDLFCH114RCH0)

Address(es): PDMIF.PDLFCH114RCH0 8785 61A0h, PDMIF_NS.PDLFCH114RCH0 0785 61A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (14)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.52 Low-Pass Filter Coefficient h1 (15) Register Channel 0 (PDLFCH115RCH0)

Address(es): PDMIF.PDLFCH115RCH0 8785 61A4h, PDMIF_NS.PDLFCH115RCH0 0785 61A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (15)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.53 Low-Pass Filter Coefficient h1 (16) Register Channel 0 (PDLFCH116RCH0)

Address(es): PDMIF.PDLFCH116RCH0 8785 61A8h, PDMIF_NS.PDLFCH116RCH0 0785 61A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (16)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.54 Low-Pass Filter Coefficient h1 (17) Register Channel 0 (PDLFCH117RCH0)

Address(es): PDMIF.PDLFCH117RCH0 8785 61ACh, PDMIF_NS.PDLFCH117RCH0 0785 61ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (17)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.55 Low-Pass Filter Coefficient h1 (18) Register Channel 0 (PDLFCH118RCH0)

Address(es): PDMIF.PDLFCH118RCH0 8785 61B0h, PDMIF_NS.PDLFCH118RCH0 0785 61B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (18)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.56 Low-Pass Filter Coefficient h1 (19) Register Channel 0 (PDLFCH119RCH0)

Address(es): PDMIF.PDLFCH119RCH0 8785 61B4h, PDMIF_NS.PDLFCH119RCH0 0785 61B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	LFH1[12:0]	Low-Pass (Half-Band Decimation) Filter Coefficient h1 (19)	13-bit signed data (fractional bits [10:0])	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Low-pass filter cut-off frequency = 7 kHz

This register is used to set a low-pass filter coefficient.

58.2.57 Audio Detection Lower Threshold Register Channel 0 (PDSDLTRCH0)

Address(es): PDMIF.PDSDLTRCH0 8785 61B8h, PDMIF_NS.PDSDLTRCH0 0785 61B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	SDETL[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SDETL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	SDETL[19:0]	Audio Detection Lower Limit Setting	Set the lower limit for audio detection (20-bit signed data).	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the lower limit for audio detection.

58.2.58 Audio Detection Upper Threshold Register Channel 0 (PDSDUTRCH0)

Address(es): PDMIF.PDSDUTRCH0 8785 61BCh, PDMIF_NS.PDSDUTRCH0 0785 61BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	SDETU[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SDETU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	SDETU[19:0]	Audio Detection Upper Limit Setting	Set the upper limit for audio detection (20-bit signed data).	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the upper limit for audio detection.

58.2.59 Data Buffer Control Register Channel 0 (PDDBCRCH0)

Address(es): PDMIF.PDDBCRCH0 8785 61C0h, PDMIF_NS.PDDBCRCH0 0785 61C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRITHR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DATRITHR[2:0]	Data Receive Interrupt Threshold Setting	b2 b0 0 0 0: Outputs an interrupt when one or more pieces of data are received. 0 0 1: Outputs an interrupt when two or more pieces of data are received. 0 1 0: Outputs an interrupt when four or more pieces of data are received. 0 1 1: Outputs an interrupt when eight or more pieces of data are received. Other than above: Outputs an interrupt when 16 or more pieces of data are received.	W
b31 to b3	—	Reserved	The write value should be 0.	W

This register is used to set the threshold for data reception interrupt output.

58.2.60 Short-Circuit Threshold Setting Register Channel 0 (PDSCTSRCH0)

Address(es): PDMIF.PDSCTSRCH0 8785 61C4h, PDMIF_NS.PDSCTSRCH0 0785 61C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SCDH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCDL[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	SCDL[12:0]	Short-Circuit Detection Low Consecutive Detection Count Setting	Set the upper limit for the number of times 0 is input consecutively to the PDMDAT0 pin.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b16	SCDH[12:0]	Short-Circuit Detection High Consecutive Detection Count Setting	Set the upper limit for the number of times 1 is input consecutively to the PDMDAT0 pin.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the thresholds for the number of consecutive lows/highs for short-circuit detection.

58.2.61 Overvoltage Lower Threshold Register Channel 0 (PDOVLTRCH0)

Address(es): PDMIF.PDOVLTRCH0 8785 61C8h, PDMIF_NS.PDOVLTRCH0 0785 61C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	OVDL[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVDL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	OVDL[19:0]	Overvoltage Detection Lower Limit Setting	Set the lower limit of the effective voltage for overvoltage detection (20-bit signed data).	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the lower limit of the effective voltage for overvoltage detection.

58.2.62 Overvoltage Upper Threshold Register Channel 0 (PDOVUTRCH0)

Address(es): PDMIF.PDOVUTRCH0 8785 61CCh, PDMIF_NS.PDOVUTRCH0 0785 61CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	OVDU[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVDU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	OVDU[19:0]	Overvoltage Detection Upper Limit Setting	Set the upper limit of the effective voltage for overvoltage detection (20-bit signed data).	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the upper limit of the effective voltage for overvoltage detection.

58.2.63 Data Read Control Register Channel 0 (PDDRCRCH0)

Address(es): PDMIF.PDDRCRCH0 8785 61E0h, PDMIF_NS.PDDRCRCH0 0785 61E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DATRE	Data Read Enable	0: Disables reading from the data buffer. 1: Enables reading from the data buffer.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a mirroring register of the PDCDRCR register that is used to control data read.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.64 Data Clear Register Channel 0 (PDDCRCH0)

Address(es): PDMIF.PDDCRCH0 8785 61E4h, PDMIF_NS.PDDCRCH0 0785 61E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DATC	Data Clear	0: No operation 1: Clears data.	W
b31 to b1	—	Reserved	The write value should be 0.	W

This register is a mirroring register of the PDCDCR register that is used to clear data.
For more information on the mirroring register, see section 58.2.67, Mirroring Registers.

58.2.65 Data Read Register Channel 0 (PDDRRCH0)

Address(es): PDMIF.PDDRRCH0 8785 61E8h, PDMIF_NS.PDDRRCH0 0785 61E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	DAT[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DAT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	DAT[19:0]	Data	20-bit or 16-bit signed audio data is read. In 20-bit mode (PDMDSRCH0.DBIS[3:0] = 0xxxb), the sign bit is [19]. In 16-bit mode (PDMDSRCH0.DBIS[3:0] = 1xxxb), the sign bits are [19:15].	R
b31 to b20	—	Reserved	These bits are read as 0.	R

This register is used to read audio data.

58.2.66 Data Status Register Channel 0 (PDDSRCH0)

Address(es): PDMIF.PDDSRCH0 8785 61ECh, PDMIF_NS.PDDSRCH0 0785 61ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DATNUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATNUM[7:0]	Number of Data Stored in Buffer	When PDDRCRCH0.DATRE = 0, 0 is read. When PDDRCRCH0.DATRE = 1, the number of data stored in the buffer is read. The maximum value is 31.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

This register is used to read the number of data stored in the buffer.

58.2.67 Mirroring Registers

Common registers, excluding the Version Register, have mirroring registers for each channel.
The PDCSR.EDF0 flag is set by ORing all error flags in the mirroring registers of each channel.
Common registers are used for synchronous control of multiple-channel products. For single-channel products, use either common registers or channel-specific mirroring registers.
Table 58.4 shows the common registers and corresponding mirroring registers.

Table 58.4 Mirroring Registers

Common Register		Mirroring Register of Each Channel	
Register Name	**Bit Name**	**Register Name**	**Bit Name**
PDCSTRTR	STRTRG0	PDSTRTRCH0	STRTRG
PDCSTPTR	STPTRG0	PDSTPTRCH0	STPTRG
PDCCHGTR	CHGTRG0	PDCHGTRCH0	CHGTRG
PDCICR	ISDE0	PDICRCH0	ISDE
	IDRE0		IDRE
	IEDE0		IEDE
PDCSR	STATE0	PDSRCH0	STATE
	SDF0		SDF
	DRF0		DRF
PDCSCR	SDFC0	PDSCRCH0	SDFC
PDCSDCR	SDE0	PDSDCRCH0	SDE
PDCDRCR	DATRE0	PDDRCRCH0	DATRE
PDCDCR	DATC0	PDDCRCH0	DARC

58.3 Operation

58.3.1 Clock Specifications

The core clock is divided and output to PDMCLK0.
Table 58.5 shows the clock frequencies and clock modes guaranteed by the PDM-IF. Note that PDMIFCLK and PCLK are asynchronous.

Table 58.5 Guaranteed Clock Frequencies and Clock Modes

PDMIFCLK Frequency (MHz)	PCLK Frequency (MHz)	PDMCLK0 Frequency (MHz)	Supported Core Clock Division Ratio	PDSFCRCH0.CKDIV[3:0] Setting
8	1 to 75*1	4	2	0000
		2	4	0001
		1.33	6	0010
		1	8	0011
		0.8	10	0100
		0.67	12	0101
		0.57	14	0110
		0.5	16	0111
		0.44	18	1000
		0.4	20	1001
		0.36	22	1010
		0.33	24	1011
		0.31	26	1100
		0.29	28	1101
		0.27	30	1110
		0.25	32	1111

Note 1. Although the PDM-IF can operate at frequencies below 1 MHz, the guaranteed frequency range is 1 to 75 MHz.

58.3.1.1 Output Sampling Frequency

PDM-IF output sampling frequency is shown by the following formula:
$F_{out} = F_{clk} / (M \times 2)$

- F_{out}: PDM-IF output sampling frequency
- F_{clk}: PDMCLK0 frequency
- M: Sinc filter decimation ratio (PDSFCRCH0.SINCDEC[7:0] + 1)

58.3.2 Channel Start and Channel Stop

The channel start and channel stop are controlled by the registers shown in Table 58.6. When synchronizing multiple channels, use common registers to simultaneously set the bits of the target channels to 1.
Set channel start according to section 58.4.1, Start Flow and channel stop according to section 58.4.2, Stop Flow.

Table 58.6 Channel Start Trigger and Channel Stop Trigger

	Register and Bit Name	
Channel Control	**Common Register**	**Channel Register**
Channel start	PDCSTRTR.STRTRG0	PDSTRTRCH0.STRTRG
Channel stop	PDCSTPTR.STPTRG0	PDSTPTRCH0.STPTRG

58.3.3 Changing the PDMCLK0 Frequency

The external microphone switches the power mode according to the input clock frequency. The PDM-IF enables transition of the microphone's power mode by changing the operating frequency of the PDMCLK0 clock output to the microphone during operation. The operating frequency of PDMCLK0 can be changed with the PDSFCRCH0.CKDIV[3:0] bits. When the frequency is changed, the decimation ratio of the Sinc filter must also be changed. Set the PDSFCRCH0.SINCDEC[7:0] bits so that the sampling frequency after decimation is 32 kHz*1. In addition, changing the decimation ratio changes the output range of the Sinc filter. Also change the PDSFCRCH0.SINCRNG[4:0] bits according to the output range. For details, see section Table 58.7, Clipping the Sinc Filter Output.
After changing these register settings, the PDF-IF must wait for the settling time of the microphone and PDM-IF channels. For details, see sections section 58.4.4, Low Power Consumption Mode Transition Flow and section 58.4.5, Normal Mode Transition Flow.

Note 1. If the sampling frequency differs from 32 kHz, the filter characteristics after the Sinc filter may change.

58.3.4 Filters

58.3.4.1 Data Clipping and Shifting Among Filters

Audio sample data is clipped or shifted among filters.

Figure 58.2 shows how data is clipped and shifted among filters.

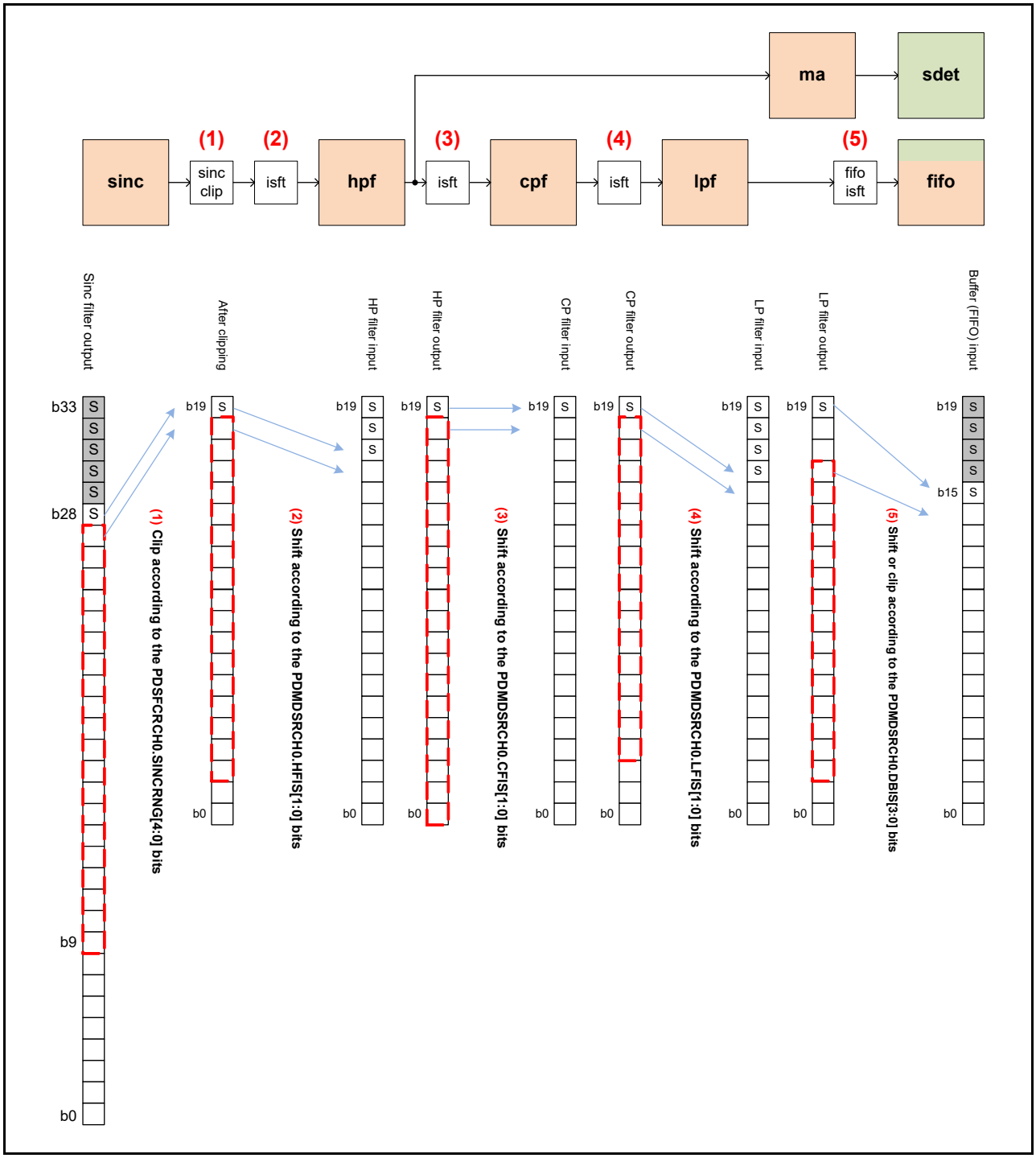


Figure 58.2 Data Clipping and Shifting Among Filters

(1) Clipping the Sinc filter output

Table 58.7 shows the available settings for the Sinc filter and data clipping. These settings are controlled by the PDSFCRCH0.SINCRNG[4:0] bits.

Table 58.7 Clipping the Sinc Filter Output (1/2)

PDMIFCLK (MHz)	Sinc Filter Order	PDMDSRCH0.SFMD[2:0]	PDMCLK Division Ratio	PDSFCRCH0.CKDIV[3:0]	Sinc Filter Decimation Ratio	PDSFCRCH0.SINCDEC[7:0]	Sinc Filter Output Effective Bits	Sinc Filter Output Clipping (20 Bits)	PDSFCRCH0.SINCRNG[4:0]	Resolution	PDMCLK0 Frequency (MHz)	Sinc Filter Output Sampling Frequency (kHz)	PDM-IF Output Sampling Frequency (kHz)
8	4	000	2	0000	125	7Ch	b28 to b0	b28 to b9	00101	20	4.0	32	16
			4	0001	62	3Dh	b24 to b0	b24 to b5	01001	20	2.0	32.26	16.13
			6	0010	42	29h	b22 to b0	b22 to b3	01011	20	1.33	31.75	15.87
			8	0011	31	1Eh	b20 to b0	b20 to b1	01101	20	1.0	32.26	16.13
			10	0100	25	18h	b19 to b0	b19 to b0	01110	20	0.8	32	16
			12	0101	21	14h	b18 to b0	b18 to b0, 0	01111	19	0.67	31.75	15.87
			14	0110	18	11h	b17 to b0	b17 to b0, 00	10000	18	0.57	31.75	15.87
			16	0111	16	0Fh	b17 to b0	b17 to b0, 00	10000	18	0.50	31.25	15.63
			18	1000	14	0Dh	b16 to b0	b16 to b0, 000	10001	17	0.44	31.75	15.87
			20	1001	12	0Bh	b15 to b0	b15 to b0, 0000	10010	16	0.40	33.33	16.67
			22	1010	11	0Ah	b14 to b0	b14 to b0, 00000	10011	15	0.36	33.06	16.53
			24	1011	10	09h	b14 to b0	b14 to b0, 00000	10011	15	0.33	33.33	16.67
			26	1100	9	08h	b13 to b0	b13 to b0, 000000	10100	14	0.31	34.19	17.09
			28	1101	9	08h	b13 to b0	b13 to b0, 000000	10100	14	0.29	31.75	15.87
			30	1110	8	07h	b13 to b0	b13 to b0, 000000	10100	14	0.27	33.33	16.67
			32	1111	8	07h	b13 to b0	b13 to b0, 000000	10100	14	0.25	31.25	15.63
8	3	011	2	0000	125	7Ch	b21 to b0	b21 to b2	01100	20	4.0	32	16
			4	0001	62	3Dh	b18 to b0	b18 to b0, 0	01111	19	2.0	32.26	16.13
			6	0010	42	29h	b17 to b0	b17 to b0, 00	10000	18	1.33	31.75	15.87
			8	0011	31	1Eh	b15 to b0	b15 to b0, 0000	10010	16	1.0	32.26	16.13
			10	0100	25	18h	b14 to b0	b14 to b0, 00000	10011	15	0.8	32	16
			12	0101	21	14h	b14 to b0	b14 to b0, 00000	10011	15	0.67	31.75	15.87
			14	0110	18	11h	b13 to b0	b13 to b0, 000000	10100	14	0.57	31.75	15.87
			16	0111	16	0Fh	b13 to b0	b13 to b0, 000000	10100	14	0.50	31.25	15.63
			18	1000	14	0Dh	b12 to b0	b12 to b0, 0000000	10101	13	0.44	31.75	15.87
			20	1001	12	0Bh	b11 to b0	b11 to b0, 00000000	10110	12	0.40	33.33	16.67
			22	1010	11	0Ah	b11 to b0	b11 to b0, 00000000	10110	12	0.36	33.06	16.53
			24	1011	10	09h	b10 to b0	b10 to b0, 000000000	10111	11	0.33	33.33	16.67
			26	1100	9	08h	b10 to b0	b10 to b0, 000000000	10111	11	0.31	34.19	17.09
			28	1101	9	08h	b10 to b0	b10 to b0, 000000000	10111	11	0.29	31.75	15.87
			30	1110	8	07h	b10 to b0	b10 to b0, 000000000	10111	11	0.27	33.33	16.67
			32	1111	8	07h	b10 to b0	b10 to b0, 000000000	10111	11	0.25	31.25	15.63

Table 58.7 Clipping the Sinc Filter Output (2/2)

PDMIFCLK (MHz)	Sinc Filter Order	PDMDSRCH0.SFMD[2:0]	PDMCLK Division Ratio	PDSFCRCH0.CKDIV[3:0]	Sinc Filter Decimation Ratio	PDSFCRCH0.SINCDEC[7:0]	Sinc Filter Output Effective Bits	Sinc Filter Output Clipping (20 Bits)	PDSFCRCH0.SINCRNG[4:0]	Resolution	PDMCLK0 Frequency (MHz)	Sinc Filter Output Sampling Frequency (kHz)	PDM-IF Output Sampling Frequency (kHz)
8	2	010	2	0000	125	7Ch	b14 to b0	b14 to b0, 00000	10011	15	4.0	32	16
			4	0001	62	3Dh	b12 to b0	b12 to b0, 0000000	10101	13	2.0	32.26	16.13
			6	0010	42	29h	b11 to b0	b11 to b0, 00000000	10110	12	1.33	31.75	15.87
			8	0011	31	1Eh	b10 to b0	b10 to b0, 000000000	10111	11	1.0	32.26	16.13
			10	0100	25	18h	b10 to b0	b10 to b0, 000000000	10111	11	0.8	32	16
			12	0101	21	14h	b9 to b0	b9 to b0, 0000000000	11000	10	0.67	31.75	15.87
			14	0110	18	11h	b9 to b0	b9 to b0, 0000000000	11000	10	0.57	31.75	15.87
			16	0111	16	0Fh	b9 to b0	b9 to b0, 0000000000	11000	10	0.50	31.25	15.63
			18	1000	14	0Dh	b8 to b0	b8 to b0, 00000000000	11001	9	0.44	31.75	15.87
			20	1001	12	0Bh	b8 to b0	b8 to b0, 00000000000	11001	9	0.40	33.33	16.67
			22	1010	11	0Ah	b7 to b0	b7 to b0, 000000000000	11010	8	0.36	33.06	16.53
			24	1011	10	09h	b7 to b0	b7 to b0, 000000000000	11010	8	0.33	33.33	16.67
			26	1100	9	08h	b7 to b0	b7 to b0, 000000000000	11010	8	0.31	34.19	17.09
			28	1101	9	08h	b7 to b0	b7 to b0, 000000000000	11010	8	0.29	31.75	15.87
			30	1110	8	07h	b7 to b0	b7 to b0, 000000000000	11010	8	0.27	33.33	16.67
			32	1111	8	07h	b7 to b0	b7 to b0, 000000000000	11010	8	0.25	31.25	15.63
8	1	001	2	0000	125	7Ch	b7 to b0	b7 to b0, 000000000000	11010	8	4.0	32	16
			4	0001	62	3Dh	b6 to b0	b6 to b0, 0000000000000	11011	7	2.0	32.26	16.13
			6	0010	42	29h	b6 to b0	b6 to b0, 0000000000000	11011	7	1.33	31.75	15.87
			8	0011	31	1Eh	b5 to b0	b5 to b0, 00000000000000	11100	6	1.0	32.26	16.13
			10	0100	25	18h	b5 to b0	b5 to b0, 00000000000000	11100	6	0.8	32	16
			12	0101	21	14h	b5 to b0	b5 to b0, 00000000000000	11100	6	0.67	31.75	15.87
			14	0110	18	11h	b5 to b0	b5 to b0, 00000000000000	11100	6	0.57	31.75	15.87
			16	0111	16	0Fh	b5 to b0	b5 to b0, 00000000000000	11100	6	0.50	31.25	15.63
			18	1000	14	0Dh	b4 to b0	b4 to b0, 000000000000000	11101	5	0.44	31.75	15.87
			20	1001	12	0Bh	b4 to b0	b4 to b0, 000000000000000	11101	5	0.40	33.33	16.67
			22	1010	11	0Ah	b4 to b0	b4 to b0, 000000000000000	11101	5	0.36	33.06	16.53
			24	1011	10	09h	b4 to b0	b4 to b0, 000000000000000	11101	5	0.33	33.33	16.67
			26	1100	9	08h	b4 to b0	b4 to b0, 000000000000000	11101	5	0.31	34.19	17.09
			28	1101	9	08h	b4 to b0	b4 to b0, 000000000000000	11101	5	0.29	31.75	15.87
			30	1110	8	07h	b4 to b0	b4 to b0, 000000000000000	11101	5	0.27	33.33	16.67
			32	1111	8	07h	b4 to b0	b4 to b0, 000000000000000	11101	5	0.25	31.25	15.63

When the PDMDSRCH0.SFMD[2:0], PDSFCRCH0.CKDIV[3:0], PDSFCRCH0.SINCDEC[7:0], and PDSFCRCH0.SINCRNG[4:0] bits are set in the combinations shown in Table 58.7, no overflows or underflows will occur. However, if other combinations are set, an overflow or underflow may occur. When an overflow occurs, 7FFFFh is output. When an underflow occurs, 80000h is output. To check whether an overflow or underflow has occurred, use the overvoltage detection function.

(2) Shifting the high-pass filter input

The clipped output of the Sinc filter is shifted to the right and fed into a high-pass filter.
The high-pass filter input shift is controlled by the PDMDSRCH0.HFIS[1:0] bits.

Table 58.8 Shifting the High-Pass Filter Input

PDMDSRCH0.HFIS[1:0]	Function	Shifted Data*1
00	No shift	S, b18 to b0
01	Right-shifted by 1 bit	S, S, b18 to b1
10	Right-shifted by 2 bits	S, S, S, b18 to b2
11	Right-shifted by 3 bits	S, S, S, S, b18 to b3

Note 1. S: Signed bit (b19)

(3) Shifting the compensation filter input

The output of the high-pass filter is shifted to the right and input to a compensation filter.
The compensation filter input shift is controlled by the PDMDSRCH0.CFIS[1:0] bits.

Table 58.9 Shifting the Compensation Filter Input

PDMDSRCH0.CFIS[1:0]	Function	Shifted Data*1
00	No shift	S, b18 to b0
01	Right-shifted by 1 bit	S, S, b18 to b1
10	Right-shifted by 2 bits	S, S, S, b18 to b2
11	Right-shifted by 3 bits	S, S, S, S, b18 to b3

Note 1. S: Signed bit (b19)

(4) Shifting the low-pass filter input

The output of the compensation filter is shifted to the right and input to a low-pass (half-band decimation) filter.
The low-pass (half-band decimation) filter input shift is controlled by the PDMDSRCH0.LFIS[1:0] bits.

Table 58.10 Shifting the Low-Pass Filter Input

PDMDSRCH0.LFIS[1:0]	Function	Shifted Data*1
00	No shift	S, b18 to b0
01	Right-shifted by 1 bit	S, S, b18 to b1
10	Right-shifted by 2 bits	S, S, S, b18 to b2
11	Right-shifted by 3 bits	S, S, S, S, b18 to b3

Note 1. S: Signed bit (b19)

(5) Shifting or clipping buffer input

The compensation filter output is shifted to the right or clipped by 16 bits. After that, the shifted or clipped data is input to a buffer.
The buffer input shift is controlled by the PDMDSRCH0.DBIS[3:0] bits.

Table 58.11 Shifting or Clipping Buffer Input

PDMDSRCH0.DBIS[3:0]	Function	Effective Data Width	Shifted or Clipped Data*1
0000	No shift	20 bits	S, b18 to b0
0001	Right-shifted by 1 bit	20 bits	S, S, b18 to b1
0010	Right-shifted by 2 bits	20 bits	S, S, S, b18 to b2
0011	Right-shifted by 3 bits	20 bits	S, S, S, S, b18 to b3
1000	Signed bit and bits 18 to 4 are clipped.	16 bits	S, b18 to b4
1001	Signed bit and bits 17 to 3 are clipped.*2	16 bits	S, b17 to b3
1010	Signed bit and bits 16 to 2 are clipped.*2	16 bits	S, b16 to b2
1011	Signed bit and bits 15 to 1 are clipped.*2	16 bits	S, b15 to b1
1100	Signed bit and bits 14 to 0 are clipped.*2	16 bits	S, b14 to b0

Note 1. S: Signed bit (bit 19)
Note 2. When an overflow occurs, 7FFFh is input to the buffer, When an underflow occurs, 8000h is input to the buffer.

58.3.4.2 Sinc Filter

The operating clock for the Sinc filter is PDMIFCLK. The Sinc filter operates using 1-bit stream data latched at the rising or falling edge of PDMCLK0. Figure 58.3 shows the timing and configuration of data input control.

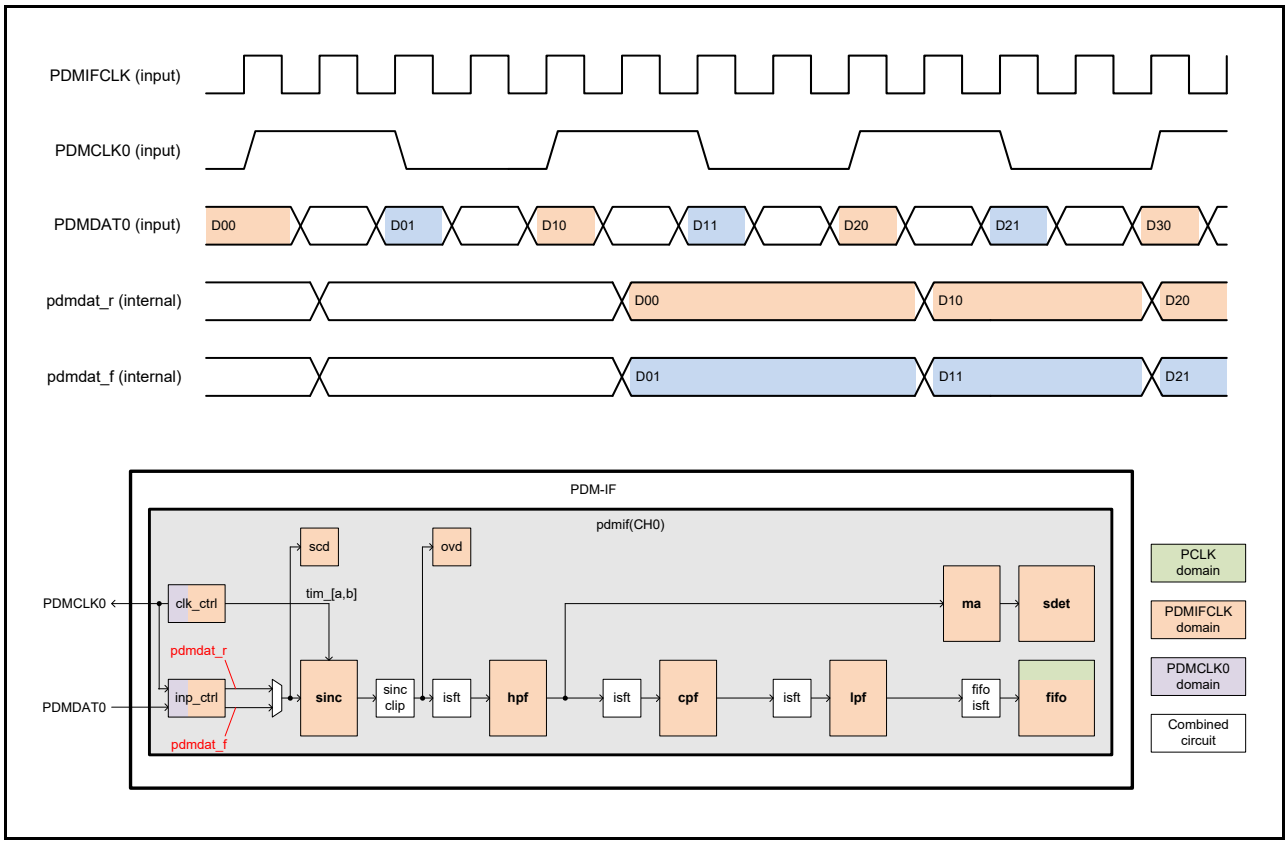


Figure 58.3 Data Input Control

Figure 58.4 shows a block diagram of the Sinc filter.

The differentiation stage operates using a decimation clock (frequency is 1/M of PDMCLK0). M is the decimation ratio and is set by the PDSFCRCH0.SINCDEC[7:0] bits. The filter results are output to the register for each decimation clock.

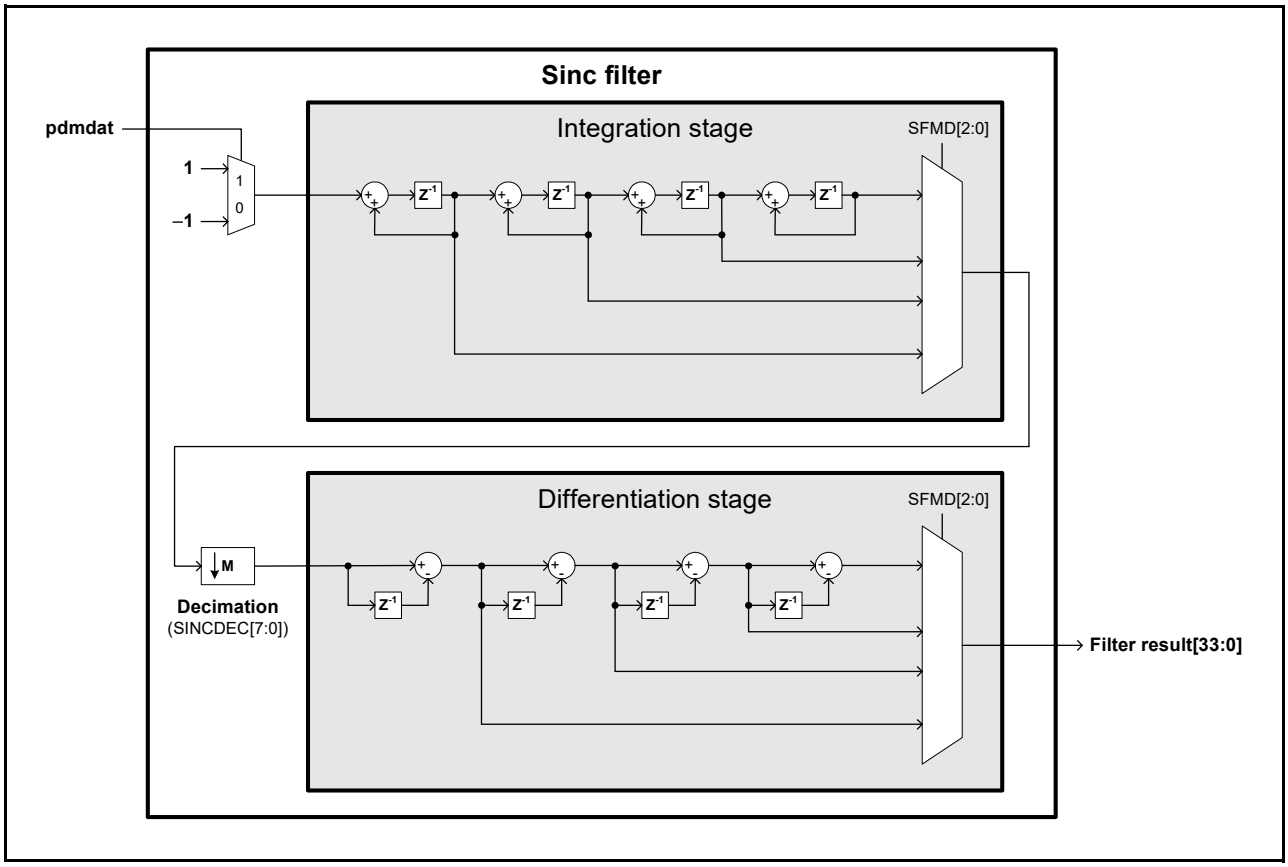


Figure 58.4 Sinc Filter Block Diagram

58.3.4.3 High-Pass Filter

The operating clock for the high-pass filter is PDMIFCLK. The high-pass filter operates using the results of the Sinc filter.

Figure 58.5 shows a block diagram of the high-pass filter, and Table 58.12 lists the specifications for the high-pass filter.

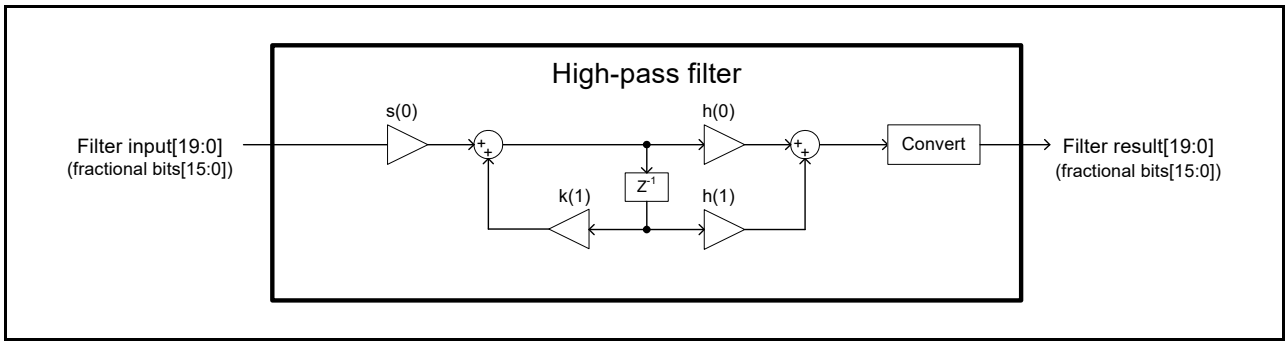


Figure 58.5 High-Pass Filter Block Diagram

Table 58.12 High-Pass Filter Specifications

Function		Specification
Input	Bit width	Signed 20 bits (16 fractional bits)
Output	Bit width	Signed 20 bits (16 fractional bits)
Coefficient	Bit width	Signed 16 bits (14 fractional bits) For details, see the following sections: • section 58.2.21, High-Pass Filter Coefficient s (0) Register Channel 0 (PDHFCS0RCH0) • section 58.2.22, High-Pass Filter Coefficient k (1) Register Channel 0 (PDHFCK1RCH0) • section 58.2.23, High-Pass Filter Coefficient h (0) Register Channel 0 (PDHFCH0RCH0)
Delay Z	Bit width	Signed 32 bits (18 fractional bits)
Multiplier	Bit width of input A	Signed 32 bits (18 fractional bits)
	Bit width of input B	Signed 16 bits (14 fractional bits)
	Output bit width	Signed 47 bits (32 fractional bits)
Adder	Bit width of input A	Signed 32 bits (18 fractional bits)
	Bit width of input B	Signed 32 bits (18 fractional bits)
	Output bit width	Signed 33 bits (18 fractional bits)
Output conversion	Input bit width	Signed 32 bits (18 fractional bits)
	Output bit width	Signed 20 bits (16 fractional bits)
Excessive fractional bit processing		Excessive fractional bits are truncated at the function unit input or final output.

58.3.4.4 Compensation Filter

The operating clock for the compensation filter is PDMIFCLK. The compensation filter operates using the results of the high-pass filter.

Figure 58.6 shows a block diagram of the compensation filter, and Table 58.13 lists the specifications for the compensation filter.

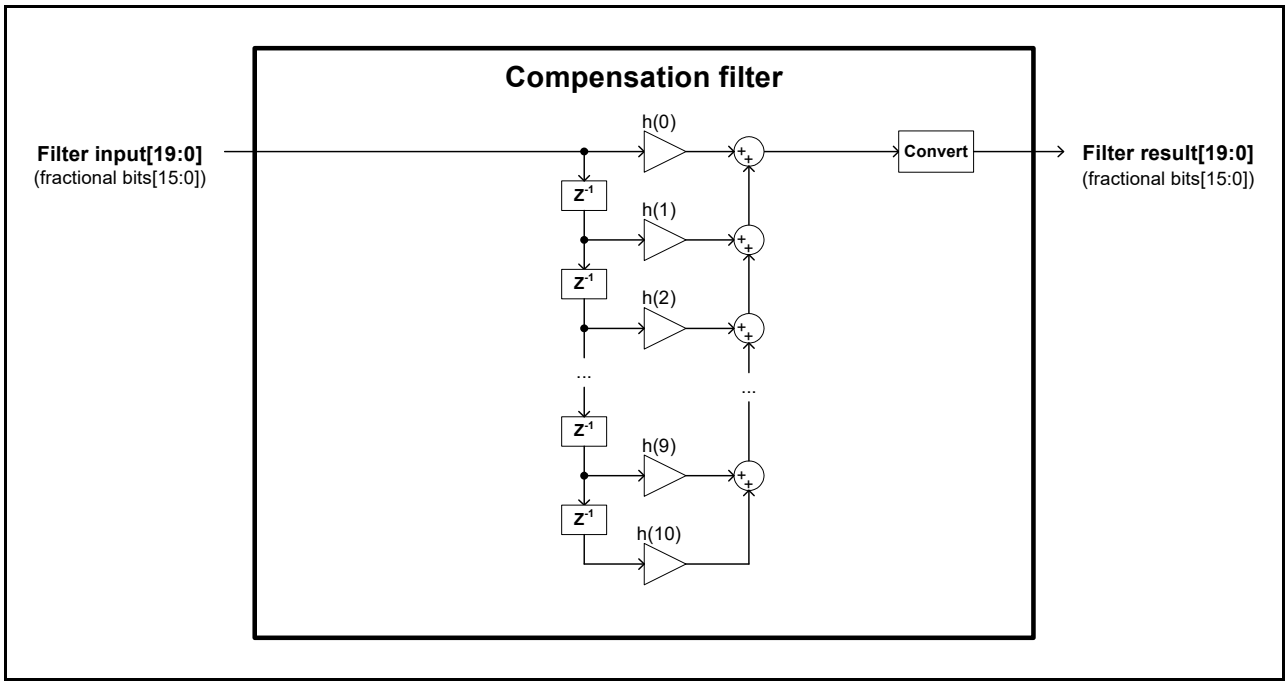


Figure 58.6 Compensation Filter Block Diagram

Table 58.13 Compensation Filter Specifications

Function		Specification
Input	Bit width	Signed 20 bits (16 fractional bits)
Output	Bit width	Signed 20 bits (16 fractional bits)
Coefficient	Bit width	Signed 13 bits (11 fractional bits) For details, see the following sections: • section 58.2.25, Compensation Filter Coefficient h (0) Register Channel 0 (PDCFCH00RCH0)
Delay Z	Bit width	Signed 20 bits (16 fractional bits)
Multiplier	Bit width of input A	Signed 20 bits (16 fractional bits)
	Bit width of input B	Signed 13 bits (11 fractional bits)
	Output bit width	Signed 32 bits (27 fractional bits)
Adder	Bit width of input A	Signed 22 bits (18 fractional bits)
	Bit width of input B	Signed 22 bits (18 fractional bits)
	Output bit width	Signed 23 bits (18 fractional bits)
Output conversion	Input bit width	Signed 22 bits (18 fractional bits)
	Output bit width	Signed 20 bits (16 fractional bits)
Excessive fractional bit processing		Excessive fractional bits are truncated at the function unit input or final output.

58.3.4.5 Low-Pass (Half-Band Decimation) Filter

The operating clock for the low-pass (half-band decimation) filter is PDMIFCLK. The low-pass filter operates using the results of the compensation filter. The output data is decimated to half of the input data. As a result, the output data frequency is reduced to half the input frequency, and the output data frequency becomes equal to the sampling rate of the data stored in the FIFO.

Figure 58.7 shows a block diagram of the low-pass filter, and Table 58.14 lists the specifications for the low-pass filter.

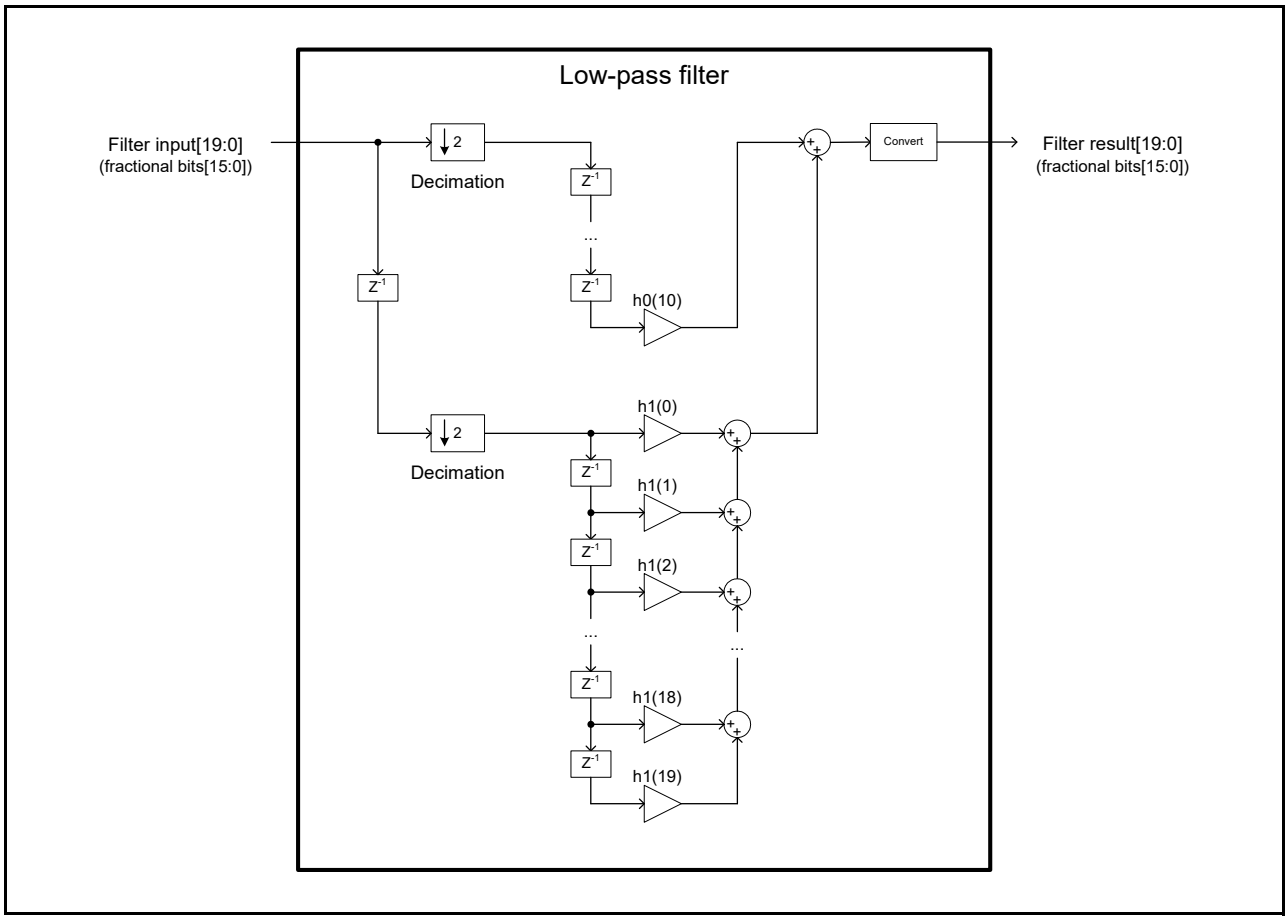


Figure 58.7 Low-Pass Filter Block Diagram

Table 58.14 Low-Pass Filter Specifications

Function		Specification
Input	Bit width	Signed 20 bits (16 fractional bits)
Output	Bit width	Signed 20 bits (16 fractional bits)
Coefficient	Bit width	Signed 13 bits (11 fractional bits) For details, see the following sections: • section 58.2.36, Low-Pass Filter Coefficient h0 (10) Register Channel 0 (PDLFCH010RCH0) • section 58.2.37, Low-Pass Filter Coefficient h1 (0) Register Channel 0 (PDLFCH100RCH0)
Delay Z	Bit width	Signed 20 bits (16 fractional bits)
Multiplier	Bit width of input A	Signed 20 bits (16 fractional bits)
	Bit width of input B	Signed 13 bits (11 fractional bits)
	Output bit width	Signed 32 bits (27 fractional bits)
Adder	Bit width of input A	Signed 22 bits (18 fractional bits)
	Bit width of input B	Signed 22 bits (18 fractional bits)
	Output bit width	Signed 23 bits (18 fractional bits)
Output conversion	Input bit width	Signed 22 bits (18 fractional bits)
	Output bit width	Signed 20 bits (16 fractional bits)
Excessive fractional bit processing		Excessive fractional bits are truncated at the function unit input or final output.

58.3.4.6 Moving Average Filter

The operating clock for the moving average filter is PDMIFCLK. The moving average filter operates using the results of the high-pass filter.

The order of the moving average filter can be changed by setting the PDMDSRCH0.SDMAMD[1:0] bits.

Figure 58.8 shows a block diagram of the moving average filter, and Table 58.15 lists the specifications for the moving average filter.

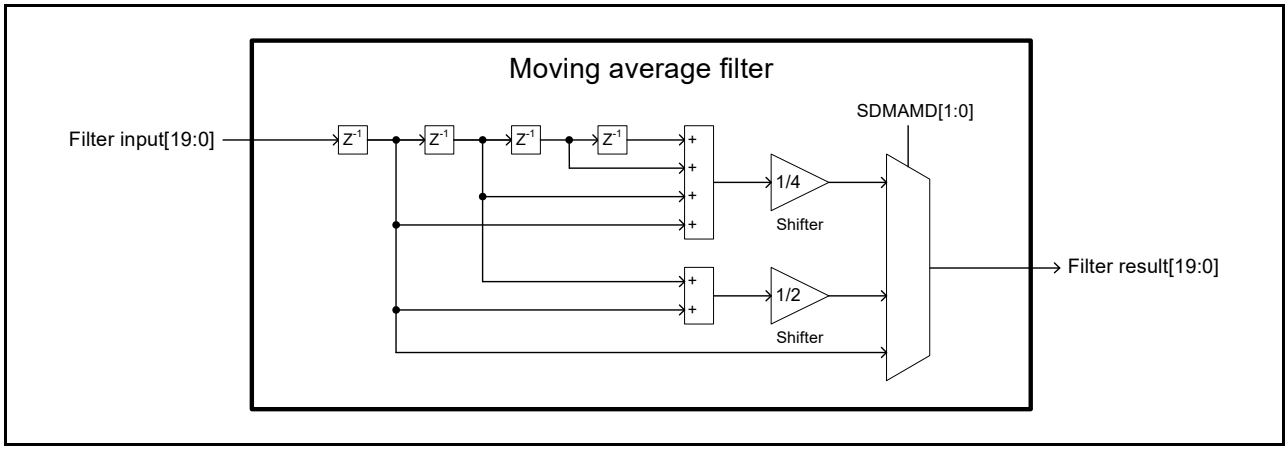


Figure 58.8 Moving Average Filter Block Diagram

Table 58.15 Moving Average Filter Specifications

Function		Specification
Input	Bit width	Signed 20 bits (16 fractional bits)
Output	Bit width	Signed 20 bits (16 fractional bits)
Delay Z	Bit width	Signed 20 bits (16 fractional bits)
Excessive fractional bit processing		Excessive fractional bits are truncated by a shifter.

58.3.4.7 Settling Times of Channel Start and Setting Change

The channel start settling time is the time from when the start trigger PDCSTRTR.STRTRG0 (or PDSTRTRCH0.STRTRG) bit is set until the filter results are output stably. The settling time for channel setting changes is the time from when the change trigger PDCCHGTR.CHGTRG0 (or PDCHGTRCH0.CHGTRG) bit is set until the filter results are output stably.

Figure 58.9 shows the measurement points for settling time, and Table 58.16 shows the settling time for channel start/setting changes.

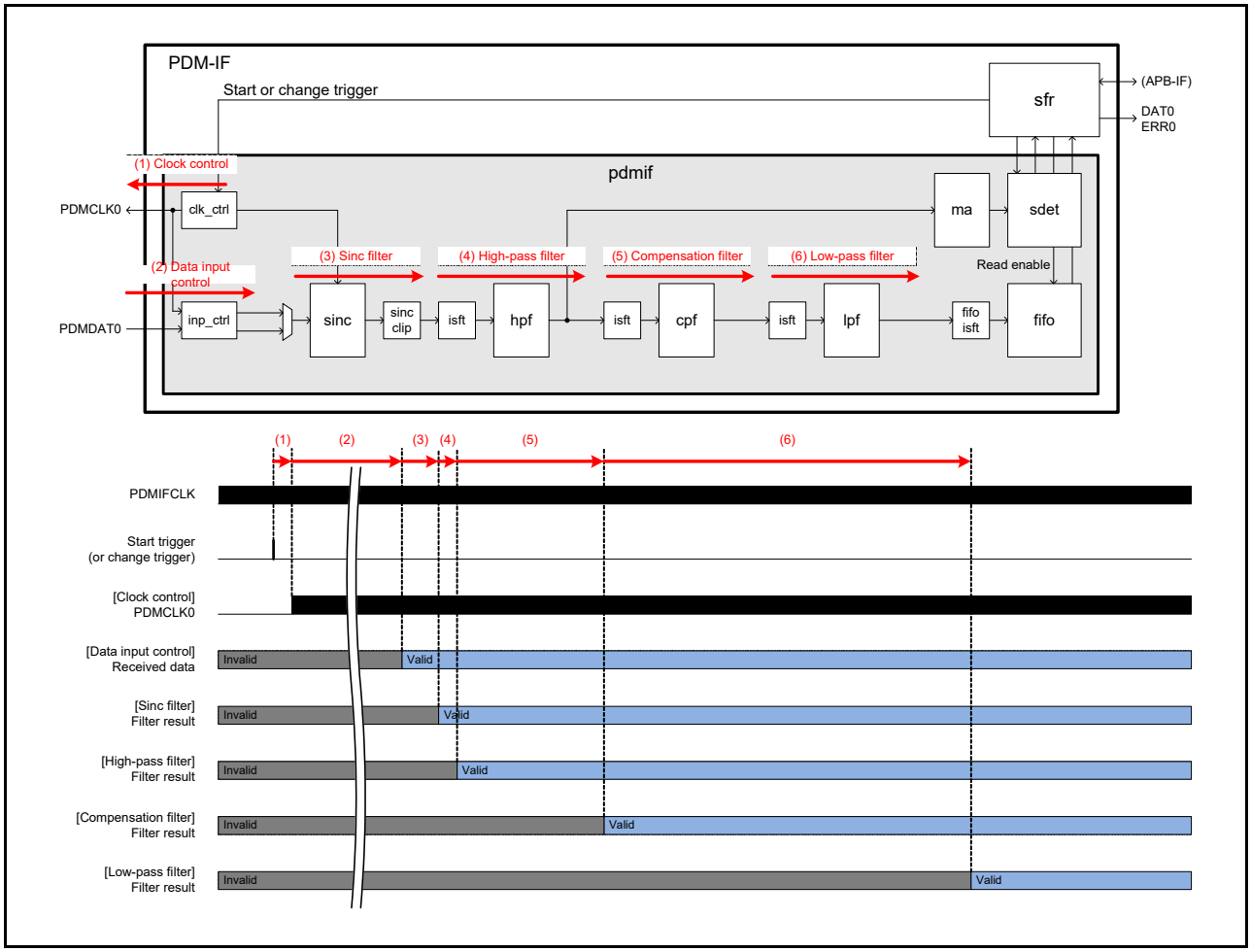


Figure 58.9 Channel Start Settling Time

Table 58.16 Channel Start and Setting Change Settling Times

No.	Function	Settling Time	
		Channel Start	Channel Setting Change
(1)	Clock control	(4 + (D / 2)) × Cp	(3 + Do + (D / 2)) × Cp
(2)	Data input control	(D /2) × Cp + (Microphone stabilization time*1)	
(3)	Sinc filter	(K × D × M) × Cp	
(4)	High-pass filter	(D × M + 5) × Cp	
(5)	Compensation filter	(10× D × M + 12) × Cp	
(6)	Low-pass filter	(38× D × M + 22) × Cp	

Note 1. Wake-up time from low power consumption mode, switching time, etc. For details, see the Data Sheet for the microphone.
Cp: PDMIFCLK period
D: PDMCLK0 division ratio
Do: PDMCLK0 division ratio (previous setting)
K: Sinc filter order
M: Decimation ratio of Sinc filter

58.3.5 Audio Detection

The operating clock for the audio detection function is PDMIFCLK. When the PDSDCRCH0.SDE bit is set to 1, audio detection is determined using the result of the moving average filter. When the result exceeds the upper limit (PDSDUTRCH0.SDETU[19:0]) or falls below the lower limit (PDSDLTRCH0.SDETL[19:0]), the audio detection flag (PDSRCH0.SDF) is set to 1. The audio detection flag is cleared to 0 by writing 1 to the PDSCRCH0.SDFC bit.
Figure 58.10 shows a block diagram of the audio detection function.

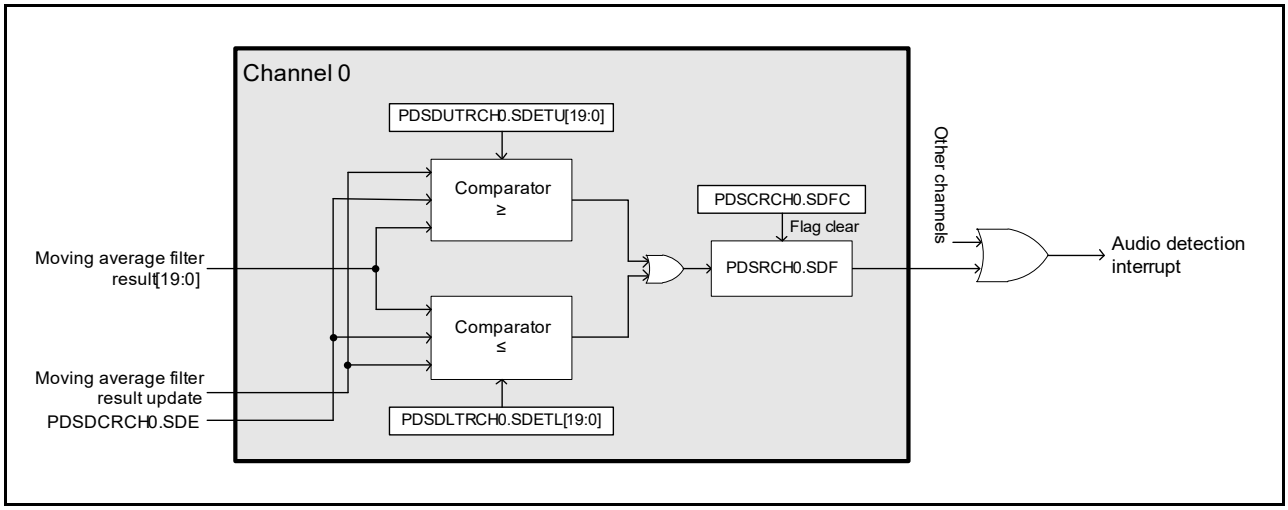


Figure 58.10 Audio Detection Function Block Diagram

58.3.6 Data Buffer

The data buffer writes audio data to the FIFO in synchronization with PDMIFCLK and reads audio data from the FIFO in synchronization with PCLK. The data buffer stores audio data in the FIFO until it is read out.

While the PDDSRCH0.DATRE bit is 0, the PDDSRCH0.DATNUM[7:0] bits indicate 0, and the PDM-IF does not allow reading of audio data. When the DATRE bit is changed to 1, the DATNUM[7:0] bits indicate the value of “buffer size - 2”, and the PDM-IF allows reading of audio data. While the DATRE bit is 1, the data buffer increments the DATNUM[7:0] bits when data is written and decrements the DATNUM[7:0] bits when data is read.

If the DATNUM[7:0] bits are set to a value greater than or equal to the threshold set by the PDDBCRCH0.DATRITHR[2:0] bits, the PDSRCH0.DRF flag is set to 1. When the value of the DATNUM[7:0] bits falls below the threshold due to data readout, the DRF flag is cleared to 0. When the value of the DATNUM[7:0] bits is set to “buffer size - 2” and data is written to the data buffer without being read, the oldest data is overwritten with the new data, and the buffer overwrite detection flag (PDSRCH0.BFOWDF) is set to 1. After overwriting, the value of the DATNUM[7:0] bits is reset to the value of “buffer size - 2”. However, the data buffer will be overwritten until the first data is read after the DATRE bit changes from 0 to 1, but the BFOWDF flag will not be set to 1.

Figure 58.11 shows a block diagram of the data buffer function.

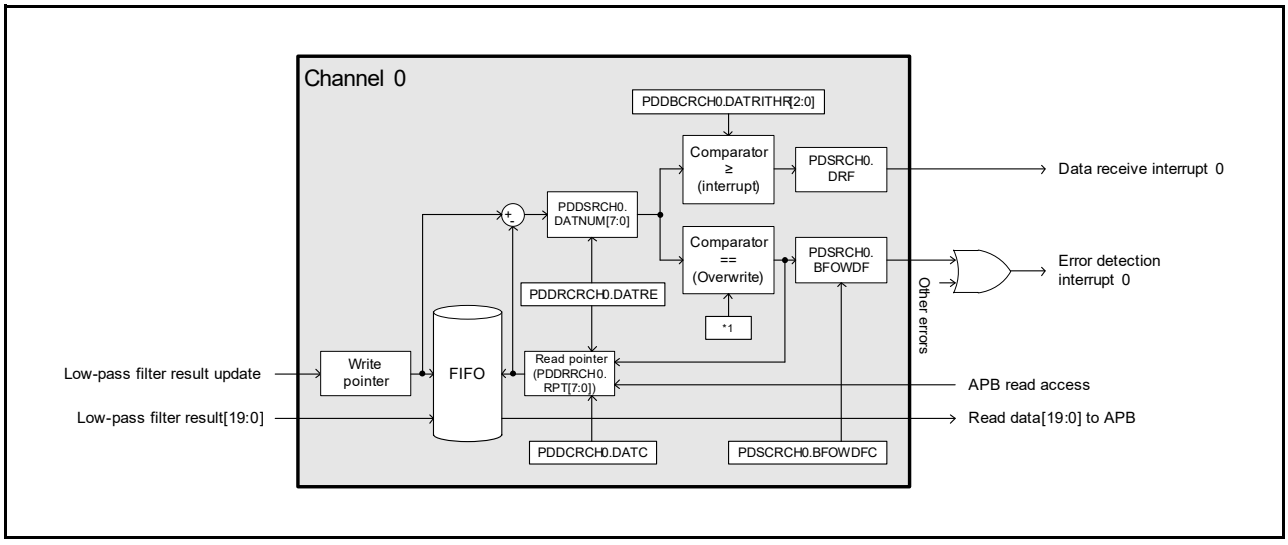


Figure 58.11 Data Buffer Function Block Diagram

58.3.7 Short-Circuit Detection

Short-circuit detection is enabled when the short-circuit detection enable bit (PDSDCRCH0.SCDE) is set to 1.
A dedicated 13-bit counter counts the consecutive 0s or 1s from pdmdat output by the data input control block of channel 0.
When the number of consecutive 0s exceeds the value set in the PDSCTSRCH0.SCDL[12:0] bits, or the number of consecutive 1s exceeds the value set in the PDSCTSRCH0.SCDH[12:0] bits, a short-circuit detection interrupt request is generated.
Figure 58.12 shows a block diagram of the short-circuit detection function.

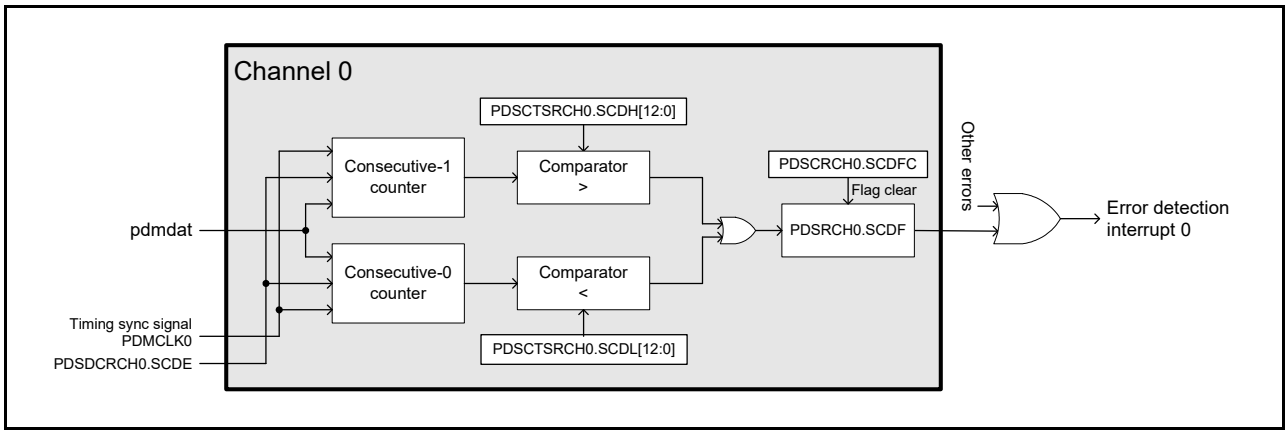


Figure 58.12 Short-Circuit Detection Function Block Diagram

58.3.8 Overvoltage Detection

When the overvoltage upper-limit-exceeded detection enable bit (PDSDCRCH0.OVUDE) is set to 1, overvoltage upper-limit-exceeded detection operates. In addition, when the overvoltage less-than-lower-limit detection enable bit (PDSDCRCH0.OVLDE) is set to 1, overvoltage less-than-lower-limit detection operates.
The audio data is clipped to 20 bits from the 34-bit data output from the Sinc filter by setting the PDSFCRCH0.SINCRNG[4:0] bits. For details, see section Table 58.7, Clipping the Sinc Filter Output.
When the clipped audio data exceeds the setting value of the PDOVUTRCH0.OVDU[19:0] bits, the overvoltage upper-limit-exceeded detection flag (PDSRCH0.OVUDF) is set to 1. When the clipped audio data falls below the value of the PDOVLTRCH0.OVDL[19:0] bits, the overvoltage less-than-lower-limit detection flag (PDSRCH0.OVLDF) is set to 1.
Figure 58.13 shows a block diagram of the overvoltage detection function.

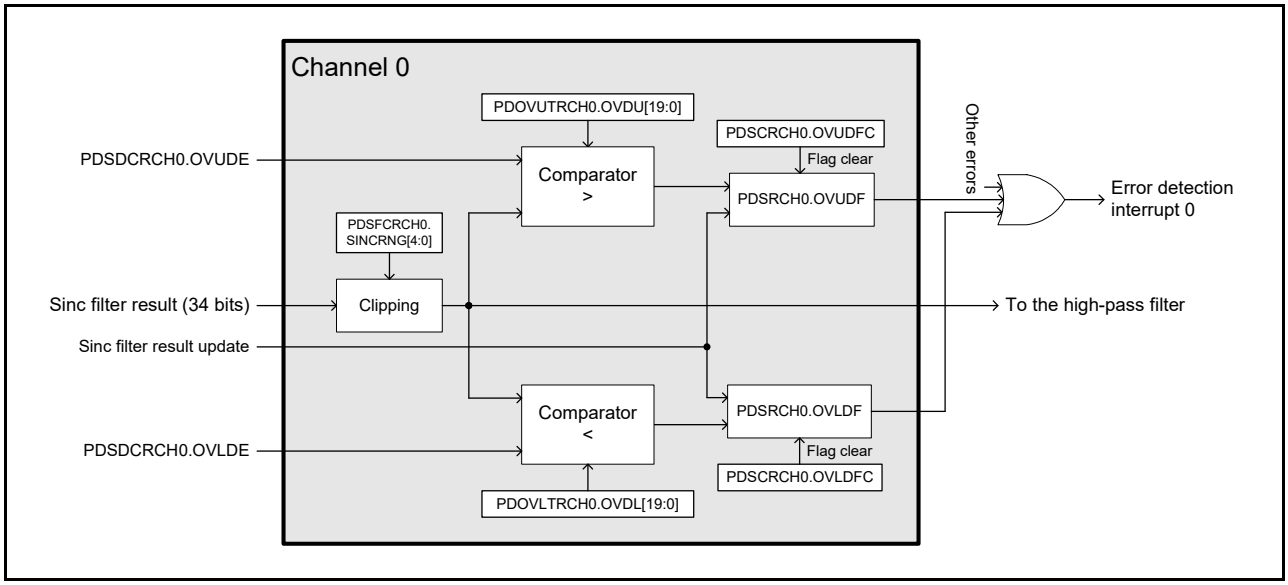


Figure 58.13 Overvoltage Detection Function Block Diagram

58.3.9 Errors

Table 58.17 lists errors with the PDM-IF.

Table 58.17 PDM-IF Error List

No	Register and Bit Symbol	Bit Name	Comment	Function (Reference)
1	PDSRCH0.SCDF	Short-circuit detection flag	—	section 58.3.7, Short-Circuit Detection
2	PDSRCH0.OVLDF	Overvoltage less-than-lower-limit detection flag	—	section 58.3.8, Overvoltage Detection
3	PDSRCH0.OVUDF	Overvoltage upper-limit-exceeded detection flag	—	section 58.3.8, Overvoltage Detection
4	PDSRCH0.BFOWDF	Buffer overwrite detection flag	For debugging	section 58.3.6, Data Buffer

The buffer overwrite detection error is for debugging purposes. Before operating the PDM-IF, the registers must be set so that these errors do not occur in the software. If any of these errors occur, stop the PDM-IF according to section 58.4.2, Stop Flow and change the settings.

58.3.10 Interrupt Functions

Table 58.18 shows the interrupt functions of the PDM-IF.

Table 58.18 List of PDM-IF Interrupts

No.	Signal Name	Interrupt Name	Number of Signals	Pulse or Level	Related Registers and Bit Names		Function (Reference)
					Common Register	Channel Register	
1	DAT0	Data receive interrupt 0	1	Pulse	PDCSR.DRF0	PDSRCH0.DRF	section 58.3.6, Data Buffer
2	SDET	Audio detection interrupt	1	Level	PDCSR.SDF0	PDSRCH0.SDF	section 58.3.5, Audio Detection
3	ERR0	Error detection interrupt 0	1	Level	PDCSR.EDF0	PDSRCH0.SCDF	section 58.3.7, Short-Circuit Detection
						PDSRCH0.OVLDF	section 58.3.8, Overvoltage Detection
						PDSRCH0.OVUDF	section 58.3.8, Overvoltage Detection
						PDSRCH0.BFOWDF	section 58.3.6, Data Buffer

58.4 Setting Flows

Figure 58.14 shows an example of PDM-IF operation, and Figure 58.15 shows the main flow of PDM-IF settings with transition to low power consumption mode.

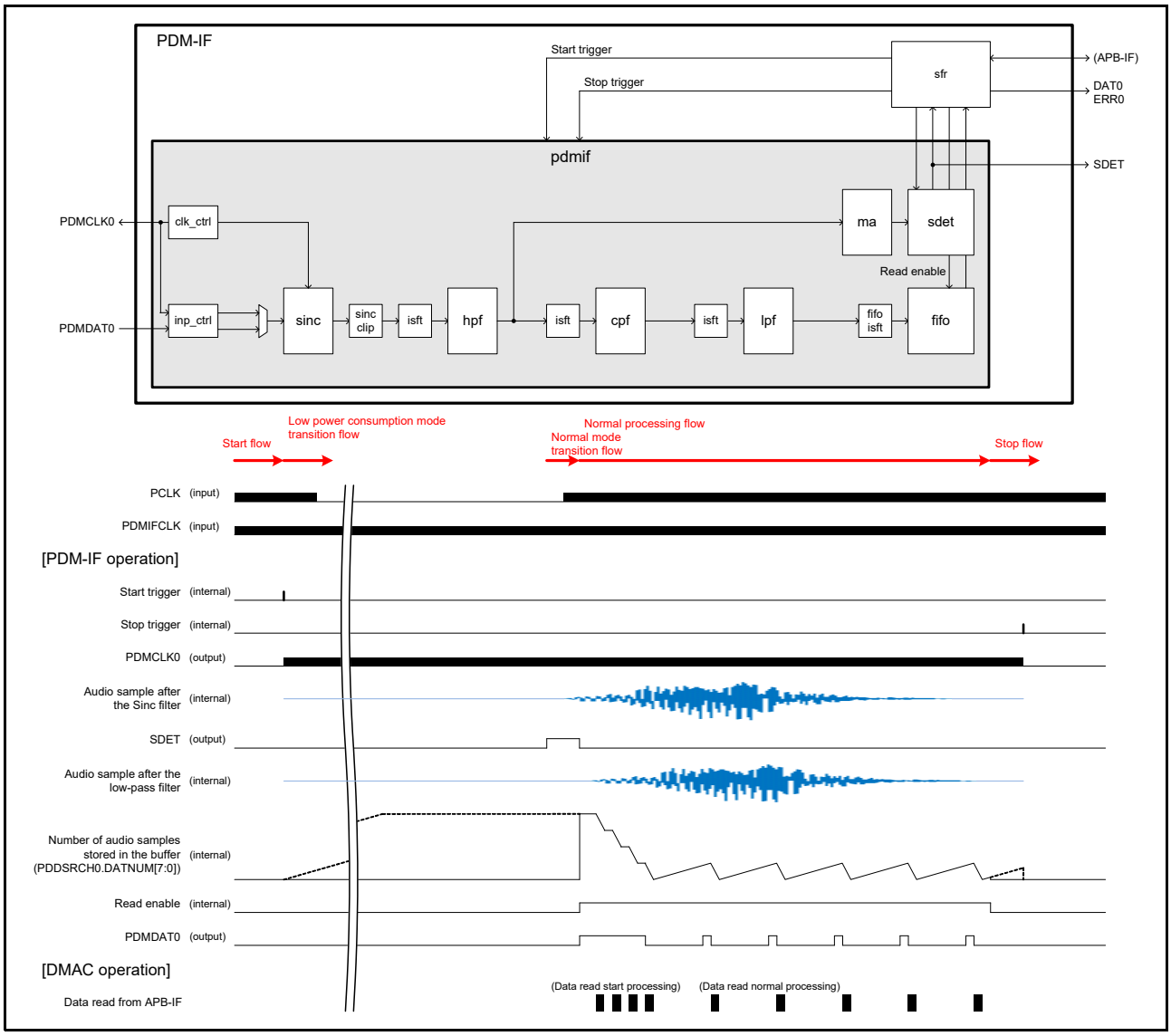


Figure 58.14 Example of PDM-IF Operation

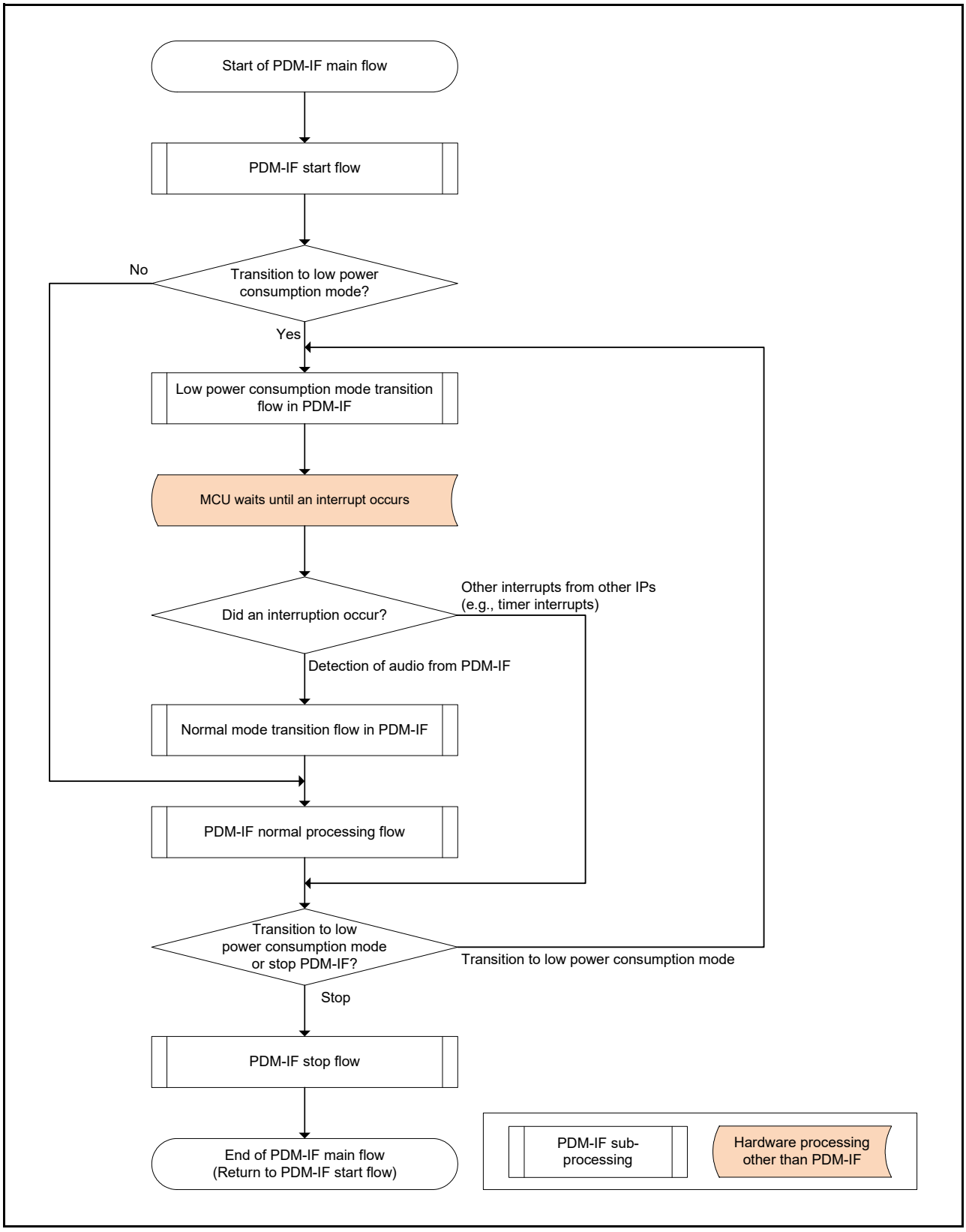


Figure 58.15 Main Flow of PDM-IF Settings with Transition to Low Power Consumption Mode

Figure 58.16 shows the main flow of PDM-IF settings without transition to low power consumption mode.

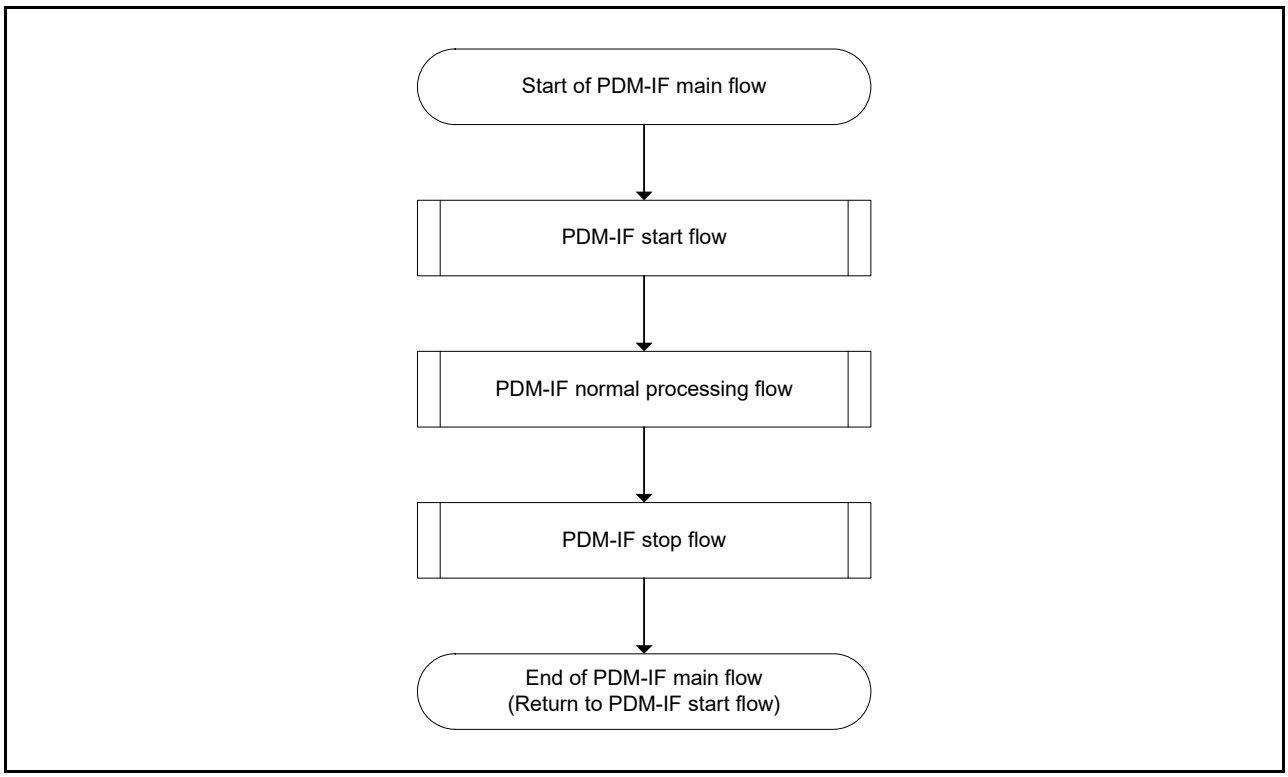
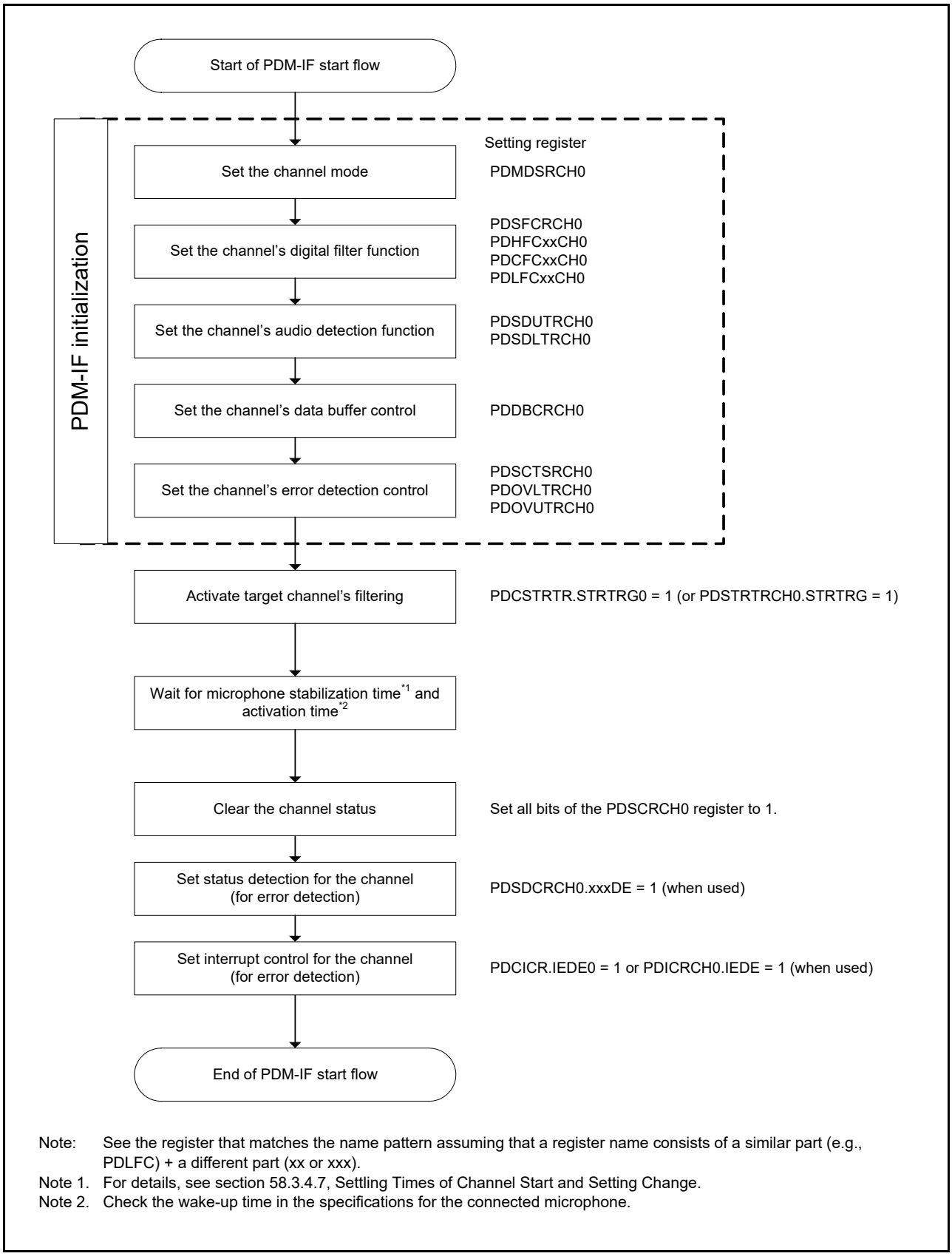


Figure 58.16 Main Flow of PDM-IF Settings without Transition to Low Power Consumption Mode

58.4.1 Start Flow

Figure 58.17 shows an example of a start procedure for the PDM-IF.

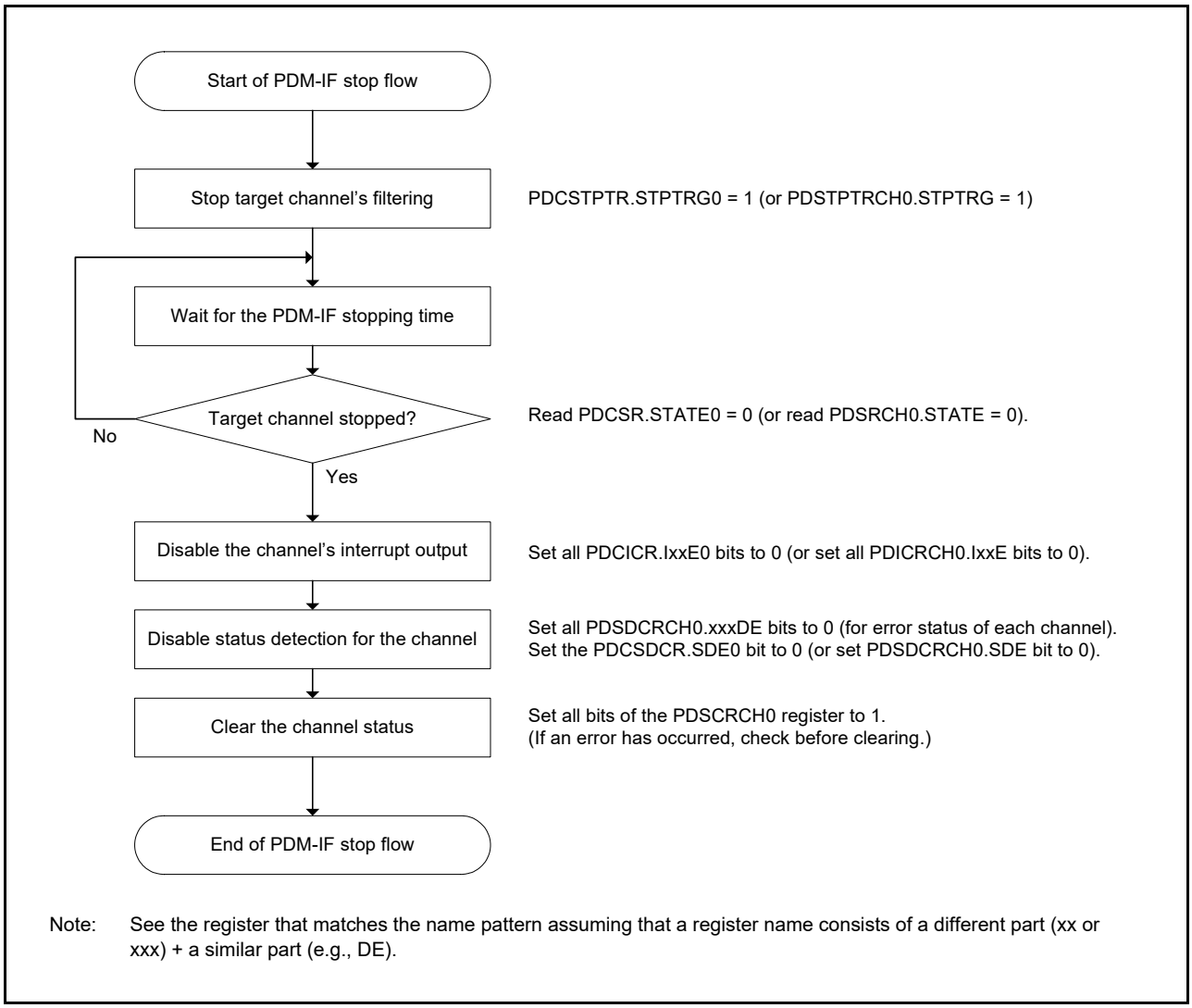


Note: See the register that matches the name pattern assuming that a register name consists of a similar part (e.g., PDLFC) + a different part (xx or xxx).
Note 1. For details, see section 58.3.4.7, Settling Times of Channel Start and Setting Change.
Note 2. Check the wake-up time in the specifications for the connected microphone.

Figure 58.17 PDM-IF Start Flow

58.4.2 Stop Flow

Figure 58.18 shows an example of a stop procedure for the PDM-IF.

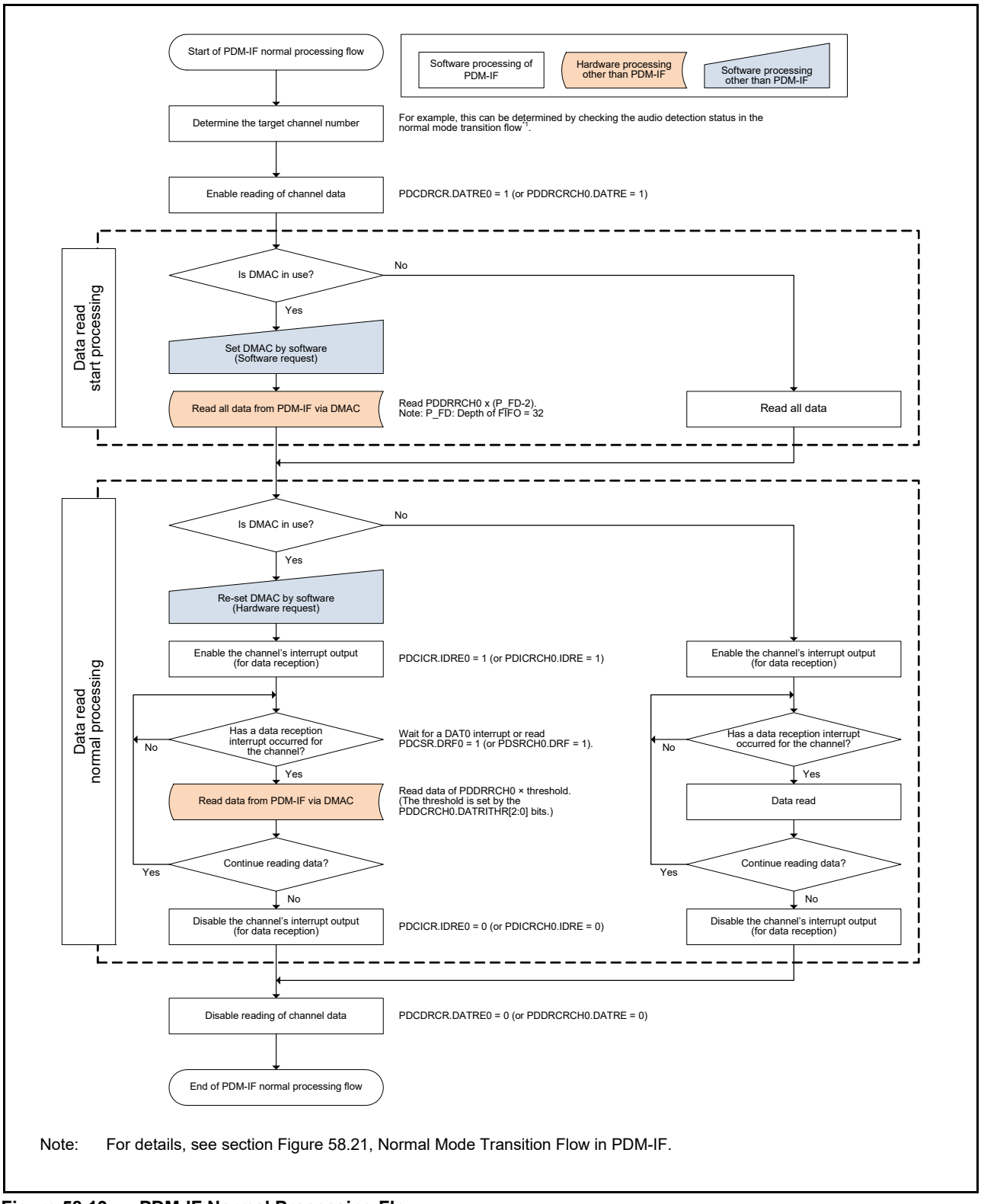


Note: See the register that matches the name pattern assuming that a register name consists of a different part (xx or xxx) + a similar part (e.g., DE).

Figure 58.18 PDM-IF Stop Flow

58.4.3 Normal Processing Flow

Figure 58.19 shows an example of a normal processing procedure for the PDM-IF.

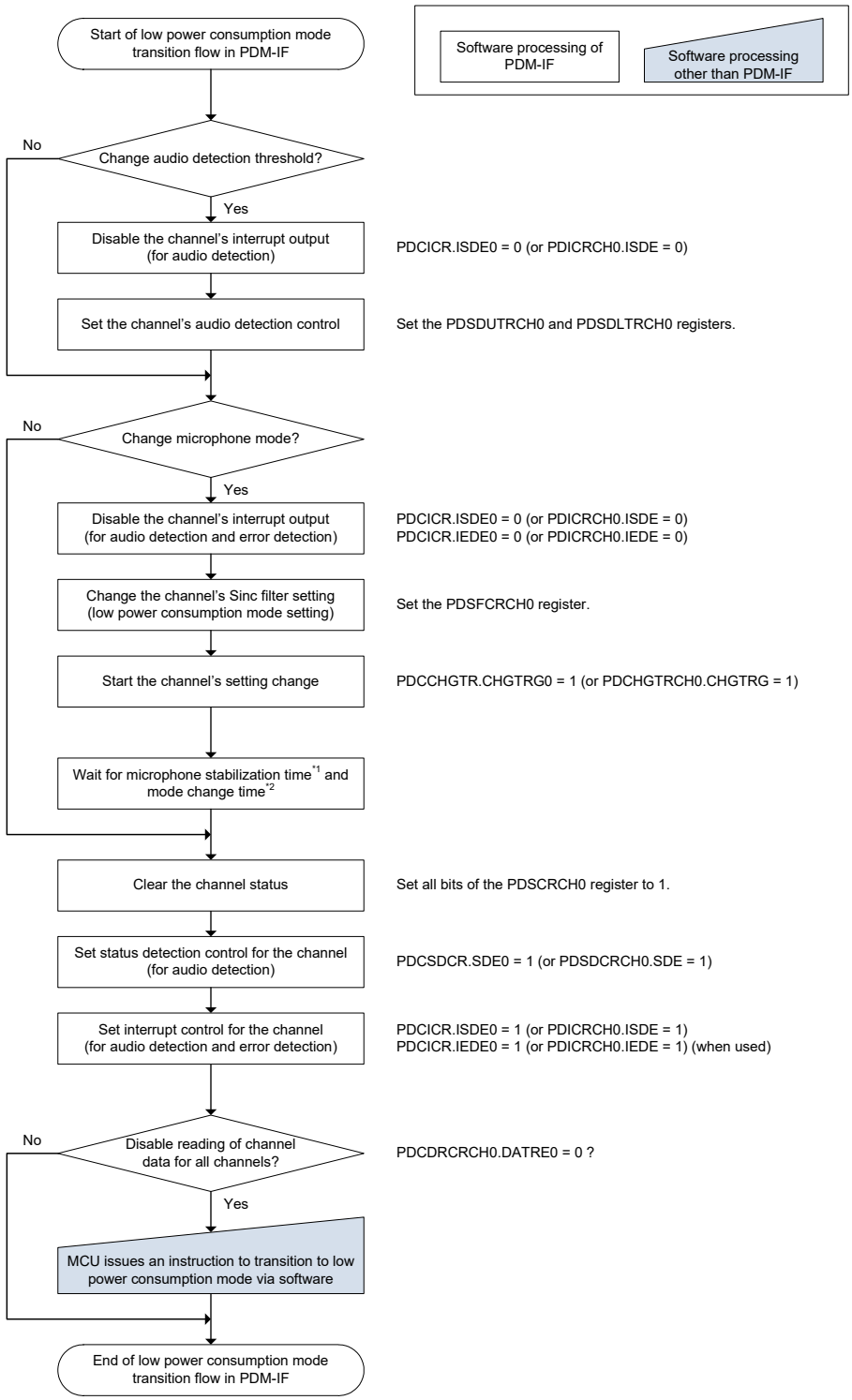


Note: For details, see section Figure 58.21, Normal Mode Transition Flow in PDM-IF.

Figure 58.19 PDM-IF Normal Processing Flow

58.4.4 Low Power Consumption Mode Transition Flow

Figure 58.20 shows an example of a transition procedure to low power consumption mode in the PDM-IF.

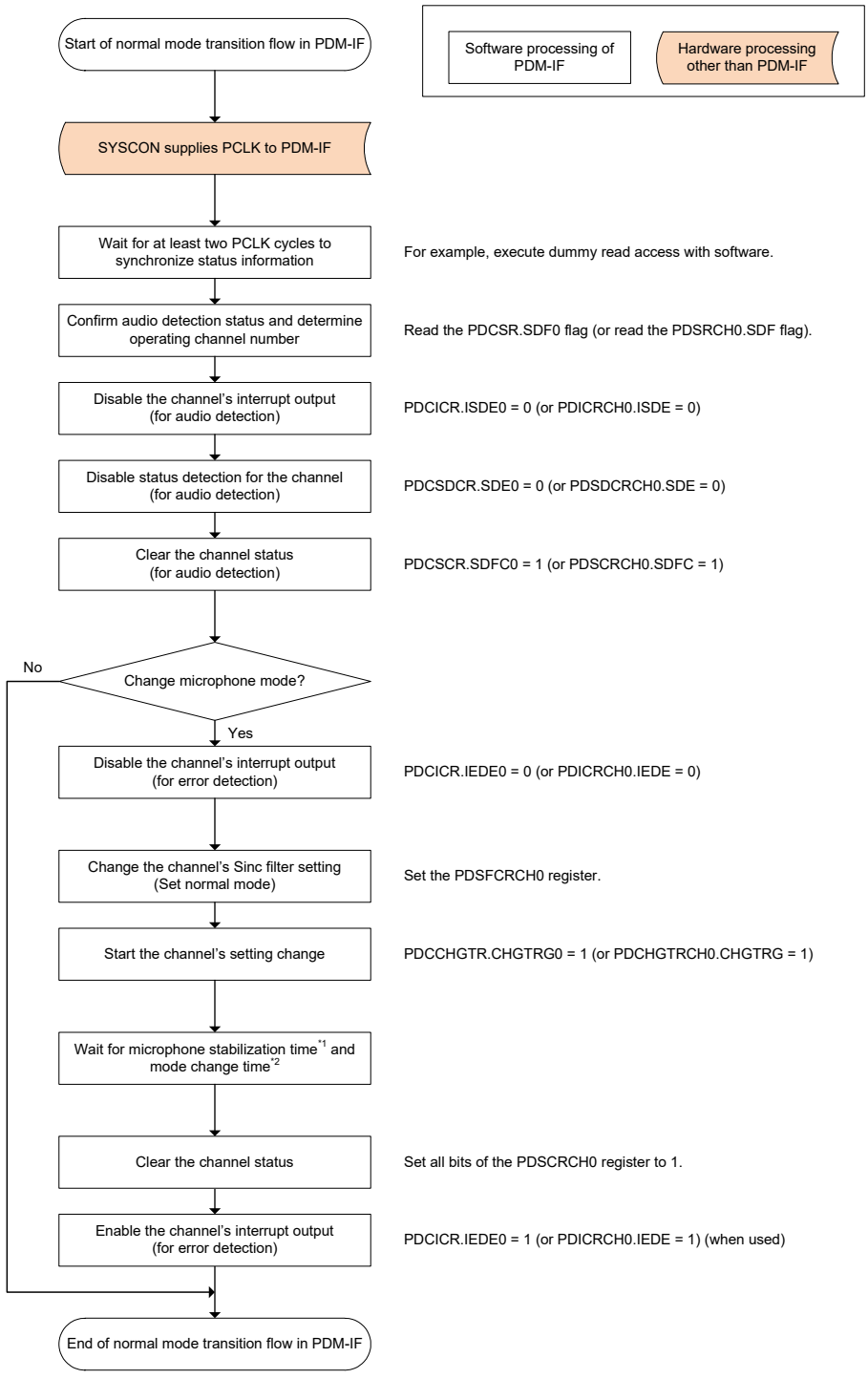


Note 1. For details, see section 58.3.4.7, Settling Times of Channel Start and Setting Change.
Note 2. For the PDM microphone mode frequency and mode change time, check the specifications for the connected microphone.

Figure 58.20 Low Power Consumption Mode Transition Flow in PDM-IF

58.4.5 Normal Mode Transition Flow

Figure 58.21 shows an example of a transition procedure to normal mode in the PDM-IF.



Note 1. For details, see section 58.3.4.7, Settling Times of Channel Start and Setting Change.
Note 2. For the PDM microphone mode frequency and mode change time, check the specifications for the connected microphone.

Figure 58.21 Normal Mode Transition Flow in PDM-IF

58.5 Usage Notes

58.5.1 Setting the Data Receive Interrupt Threshold

When data read by DMA is complete, if the number of data items stored in the buffer that have not yet been read exceeds the data reception interrupt threshold (hereinafter referred to as the threshold) set by PDDBCRCH0.DATRITHR[2:0] bits, the data reception interrupt (DAT0) that should normally be issued is not issued, and the next data read cannot be requested by the interrupt controller. This occurs when the time required to read the data exceeds [sampling rate at which data is stored in the data buffer × threshold value]. Set the threshold appropriately to avoid loss of interrupt requests. If this cannot be avoided even by setting the threshold value, check the number of data stored in the buffer when data read is complete using the PDDSRCH0.DATNUM[7:0] bits, and then start DMA by software to read the confirmed number of data.

58.5.2 Default Filter Frequency Characteristics

Figure 58.22 and Table 58.19 show the default filter frequency characteristics for reference.

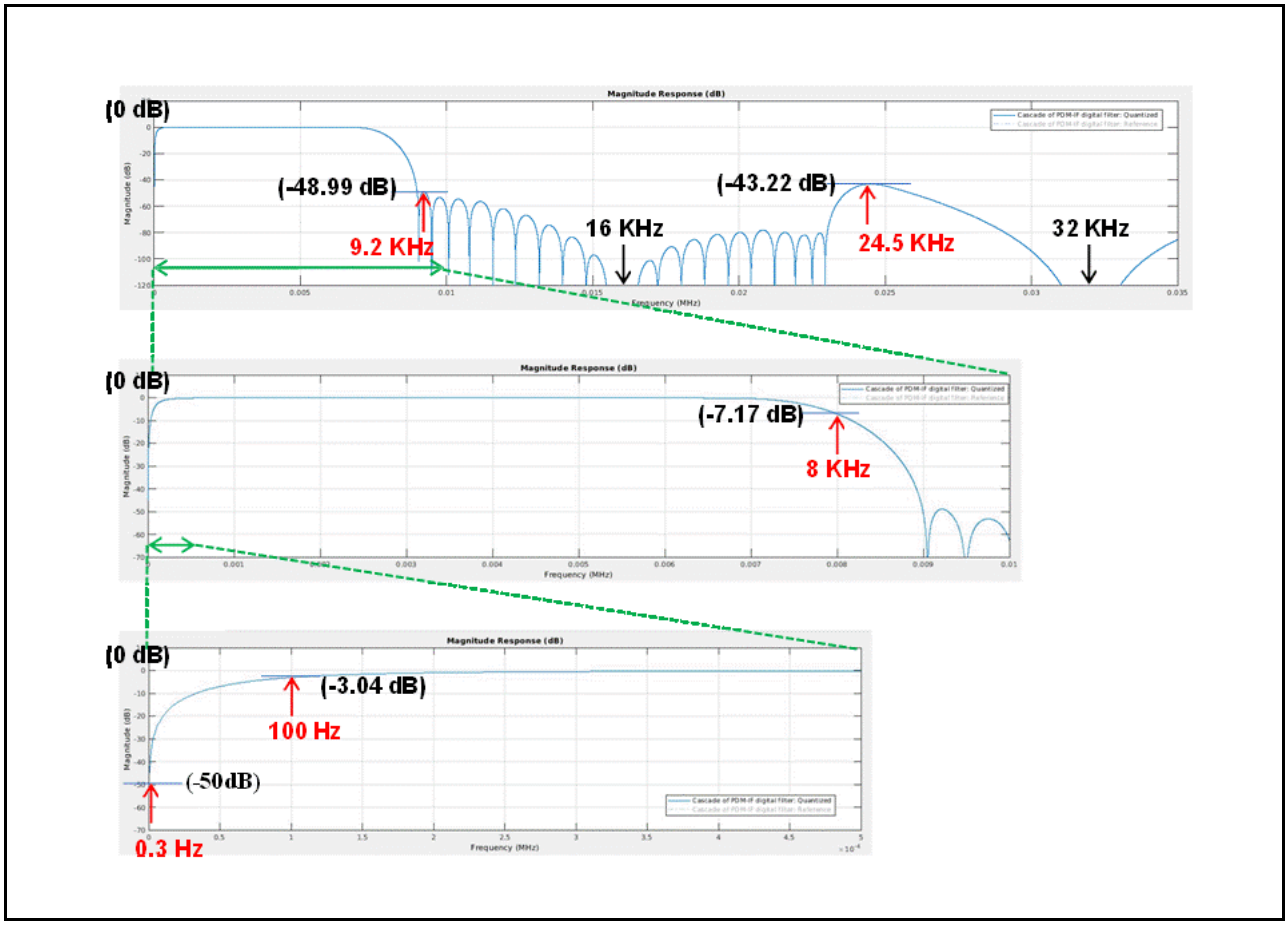


Figure 58.22 Default Filter Frequency Characteristics

Table 58.19 Default Filter Frequency Characteristics

Frequency (Hz)		Amplitude (dB)
LPF cut-off frequency	24.5 kHz	–43.22
	…	…
	9.2 kHz	–48.99
	8 kHz	–7.17
Passband	7 kHz	–0.39
	6 kHz	–0.07
	…	…
	150 Hz	–1.63
	100 Hz	–3.04
HPF cut-off frequency	50 Hz	–7.02
	0.3 Hz	–50.00

59. Phase-Counting Interface (PCIF)

This MCU has two phase-counting interfaces (PCIF0 and PCIF1). In this section, “PCLK” is used to refer to PCLKB.

59.1 Overview

The PCIF outputs A-phase, B-phase, and Z-phase signals according to phase information. There are two PCIF modes: phase counting mode and incremental mode.
In phase counting mode, the edge count and carrier period event are input according to the register settings from the CPU, and then the PCIF calculates the change timing of each phase and generates ABZ-phase outputs.
In incremental mode, A-, B-, and Z-phase signals are directly input from an incremental encoder, and then the PCIF generates ABZ-phase outputs by reducing the resolution of the input to the desired value.
Table 59.1 lists the PCIF specifications, and Figure 59.1 shows a block diagram of the PCIF.

Table 59.1 PCIF Specifications

Item	Description
Core clock (CCLK)	Clock for receiving event trigger input Selectable from two types of clocks: PCLK and GPTCLK
Frequency-divided clock (CCLK_DIV)	Clock with which ABZ-phase output is synchronized. Selectable from division rates of 1, 2, 3, 4, 5 … 32.
Reset	Hardware reset or software reset
Phase counting mode	ABZ-phase signals are output for the number of edges set in the carrier period. • Number of edges per rotation: 16 to 65,536 (multiple of 4) • Carrier period: 10 µs to CCLK_DIV period × 65,535 • Carrier period control function supporting two ways of stopping outputs in this mode, two modes for generating the carrier period start timing (carrier period event) • Automatic edge count clearing function • Function for changing the synchronization phase and output width of Z-phase • Four types of errors can be detected.
Incremental mode	ABZ-phase signals are output by changing the resolution of ABZ-phase input • Supporting five types of I/O protocols • Resolution changing function • Five ways of clearing the internal counters • Five ways of generating Z-phase signals • Four Z-phase synchronization modes
Output controller	ABZ-phase output inversion function
Event link function (input)	• Carrier period event input (ELC_IN) • Crest/trough event trigger input (PEK_IN or VLY_IN) initializes the internal counters to clear the AB-phase outputs. • Crest/trough event trigger input (PEK_IN or VLY_IN) generates Z-phase output.
Protect Zone filter	Security attribute can be set for each channel.*1

Note 1. When each channel is set to the secure attribute, registers in the channel are accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When each channel is set to the non-secure attribute, registers in the channel are accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

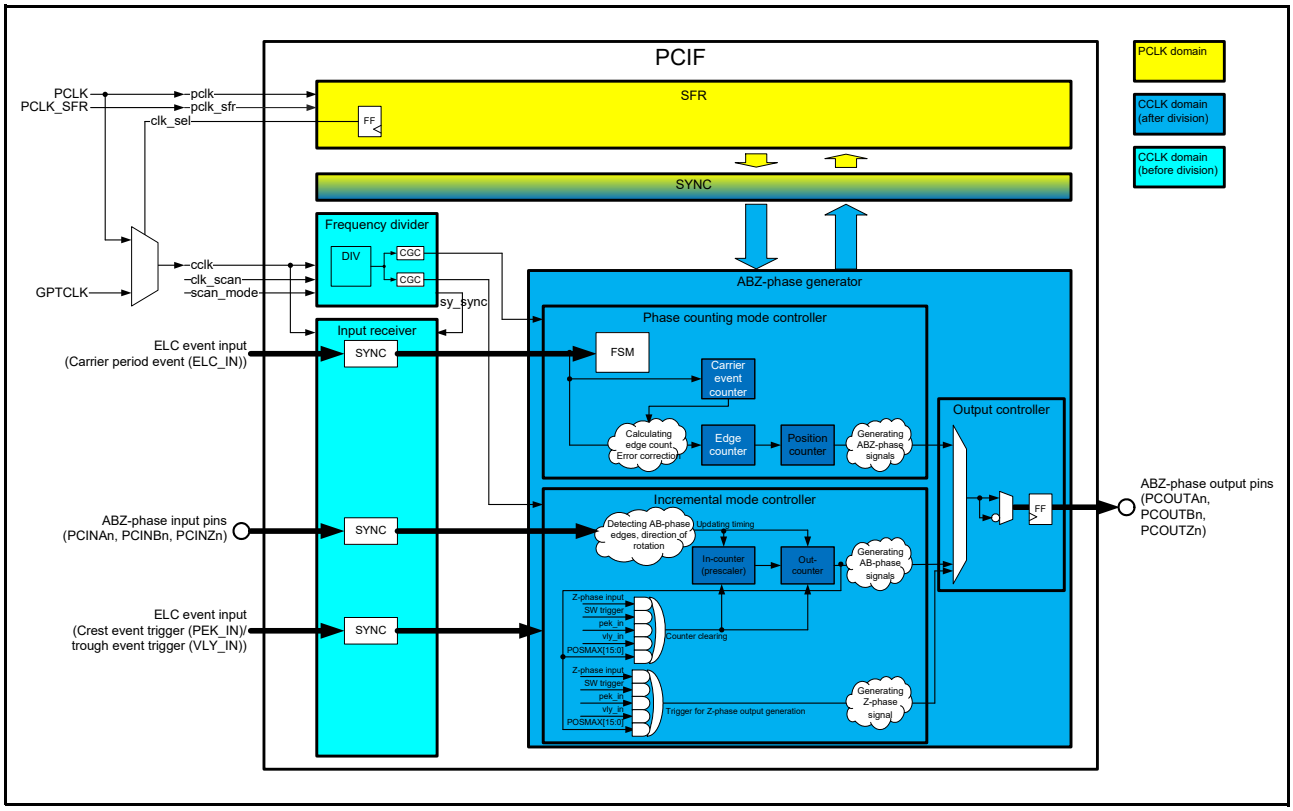


Figure 59.1 PCIF Block Diagram (n = 0, 1)

Table 59.2 shows the clock frequency range used for the PCIF and the usage conditions, and Figure 59.2 shows the clock configuration diagram of the PCIF.

Table 59.2 List of PCIF Clocks

Clock Name	Minimum Frequency (MHz)	Maximum Frequency (MHz)
PCLK	—	75
CCLK*1	—	200
CCLK_DIV*2	10	20

Note 1. If the asynchronous clock (GPTCLK) is to be selected as the CCLK clock source, make sure that this satisfies the condition for frequency GPTCLK ≤ 8 × PCLK.
Note 2. Make sure that this satisfies the condition for frequency CCLK_DIV ≤ 1/2 × PCLK.

[Clock frequency setting examples]

- For the maximum frequency: PCLK = 75 MHz, CCLK = 200 MHz, CCLK_DIV = 20 MHz
- For the minimum frequency: PCLK = 20 MHz, CCLK = 10 MHz, CCLK_DIV = 10 MHz
- For an unbalance frequency: PCLK = 20 MHz, CCLK = 160 MHz*1, CCLK_DIV = 10 MHz

Note 1. Setting of CCLK > 160 MHz is prohibited.

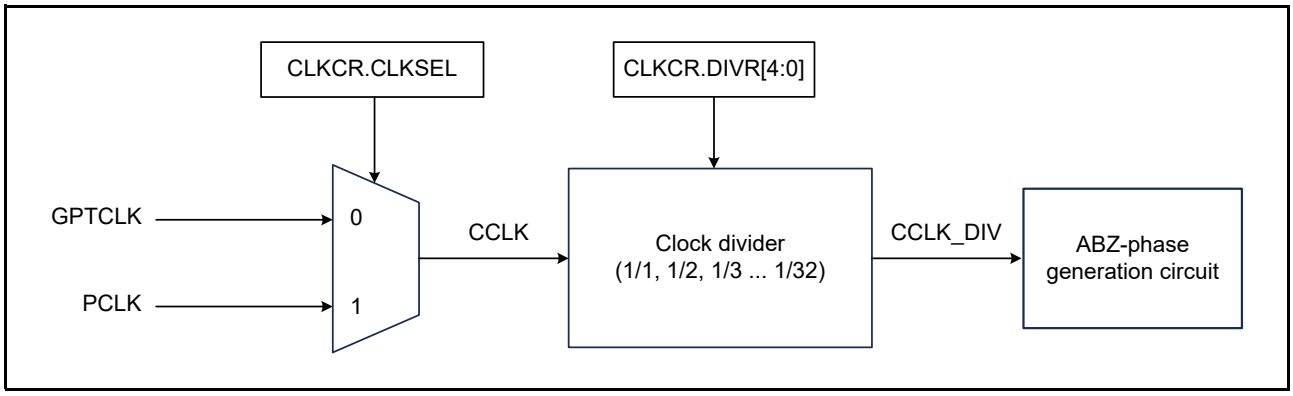


Figure 59.2 PCIF Clock Configuration Diagram

In phase counting mode, the CPU obtains position information from the encoder for each period of the current loop and position control loop notified by the timer module (carrier period event), and then calculates the difference from the previous position information. After the CPU initializes the PCIF, the PCIF generates ABZ-phase outputs by periodically updating the calculated value.
Carrier period events can be input to the PCIF via the ELC (carrier period input). It is also possible to generate a carrier period event within the PCIF.

In incremental mode, the CPU only initializes the PCIF. The PCIF generates ABZ-phase outputs from AB(Z)-phase signals directly input from the incremental encoder after startup or by using the timer module's crest/trough interrupts that are input through the ELC as events.
Table 59.3 lists the I/O pins used in the PCIF, and Figure 59.3 shows a system configuration example using the PCIF (n = 0, 1).

Table 59.3 I/O Pins of PCIF (n = 0, 1)

Interface	Pin Name	I/O	Function
PCIFn	PCINAn	Input	A-phase input pin of PCIFn
	PCINBn	Input	B-phase input pin of PCIFn
	PCINZn	Input	Z-phase input pin of PCIFn
	PCOUTAn	Output	A-phase output pin of PCIFn
	PCOUTBn	Output	B-phase output pin of PCIFn
	PCOUTZn	Output	Z-phase output pin of PCIFn

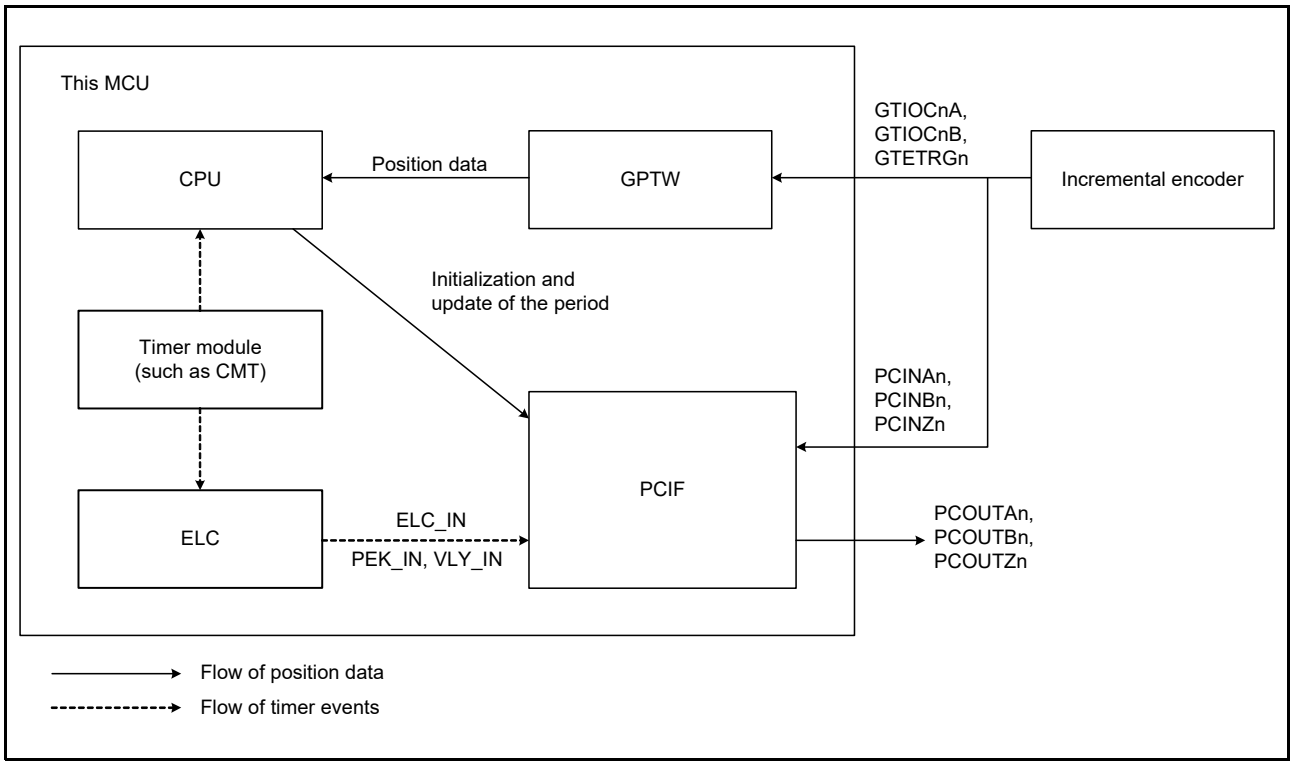


Figure 59.3 Example of System Configuration Using PCIF (n = 0, 1)

59.2 Register Descriptions

59.2.1 PCIF Clock Enable Register (CLKER)

Address(es): PCIF0.CLKER 8782 7000h, PCIF1.CLKER 8782 7080h,
PCIF0_NS.CLKER 0782 7000h, PCIF1_NS.CLKER 0782 7080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLKEN	Frequency-Divided Clock Enable	0: Frequency-divided clock operation disabled 1: Frequency-divided clock operation enabled	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables operation of the frequency-divided clock (CCLK_DIV).

CLKEN Bit (Frequency-Divided Clock Enable)

This bit enables operation of the frequency-divided clock (CCLK_DIV).

After a reset is released, make the setting of this bit according to the procedure in Figure 59.44 Overall Operating Flow for PCIF.

59.2.2 PCIF Clock Status Register (CLKSR)

Address(es): PCIF0.CLKSR 8782 7004h, PCIF1.CLKSR 8782 7084h,
PCIF0_NS.CLKSR 0782 7004h, PCIF1_NS.CLKSR 0782 7084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLKS	Frequency-Divided Clock Status Flag	0: Frequency-divided clock stopped 1: Frequency-divided clock operating	R
b31 to b1	—	Reserved	These bits are read as 0.	R

This register indicates the state of the frequency-divided clock (CCLK_DIV).

CLKS Flag (Frequency-Divided Clock Status Flag)

This status flag indicates whether the frequency-divided clock (CCLK_DIV) is operating or stopped.

59.2.3 PCIF Clock Control Register (CLKCR)

Address(es): PCIF0.CLKCR 8782 7008h, PCIF1.CLKCR 8782 7088h,
PCIF0_NS.CLKCR 0782 7008h, PCIF1_NS.CLKCR 0782 7088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DIVR[4:0]					—	—	—	—	—	—	—	CLKSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLKSEL	CCLK Clock Source Select	0: Asynchronous clock (GPTCLK) 1: Synchronous clock (PCLK)	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	DIVR[4:0]	Frequency-Divided Clock Ratio Select	b12 b8 0 0 0 0 0: Divided by 1 0 0 0 0 1: Divided by 2 (duty cycle 1:1) 0 0 0 1 0: Divided by 3 (duty cycle 2:1) 0 0 0 1 1: Divided by 4 (duty cycle 2:2) 0 0 1 0 0: Divided by 5 (duty cycle 3:2) 0 0 1 0 1: Divided by 6 (duty cycle 3:3) : : 1 1 1 1 1: Divided by 32 (duty cycle 16:16)	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects the CCLK clock source and the division ratio of the frequency-divided clock (CCLK_DIV).

CLKSEL Bit (CCLK Clock Source Select)

Selects the CCLK clock source.

If the asynchronous clock (GPTCLK) is to be selected as the CCLK clock source, make sure that this satisfies the condition for frequency GPTCLK ≤ 8 × PCLK.

Rewriting is prohibited when the frequency-divided clock is in operation (CLKSR.CLKS flag = 1).

DIVR[4:0] Bits (Frequency-Divided Clock Ratio Select)

Select the division ratio of the frequency-divided clock (CCLK_DIV).

Make sure that this satisfies the condition for frequency CCLK_DIV ≤ 1/2 × PCLK.

The setting can be changed only when CCLK_DIV is stopped (CLKER.CLKEN bit = 0).

Rewriting is prohibited when the frequency-divided clock is in operation (CLKSR.CLKS flag = 1).

59.2.4 PCIF Control Register (CTLR)

Address(es): PCIF0.CTLR 8782 700Ch, PCIF1.CTLR 8782 708Ch,
PCIF0_NS.CTLR 0782 700Ch, PCIF1_NS.CTLR 0782 708Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ZPOL	BPOL	APOL	—	—	—	—	—	—	—	PCIFMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCIFMD	PCIF Mode Select	0: Phase counting mode 1: Incremental mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	APOL	A-Phase Output Active Sense Select	0: Positive 1: Negative	R/W
b9	BPOL	B-Phase Output Active Sense Select	0: Positive 1: Negative	R/W
b10	ZPOL	Z-Phase Output Active Sense Select	0: Positive 1: Negative	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects the mode and the active senses of ABZ-phase signals.

PCIFMD Bit (PCIF Mode Select)

Selects the mode of the PCIF.
Rewriting is prohibited when the frequency-divided clock is in operation (CLKSR.CLKS flag = 1).

APOL Bit (A-Phase Output Active Sense Select)

Selects the active sense of A-phase output.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).
For details on this function, refer to section 59.3.3.1, ABZ-Phase Output Inversion Function.

BPOL Bit (B-Phase Output Active Sense Select)

Selects the active sense of B-phase output.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).
For details on this function, refer to section 59.3.3.1, ABZ-Phase Output Inversion Function.

ZPOL Bit (Z-Phase Output Active Sense Select)

Selects the active sense of Z-phase output.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).
For details on this function, refer to section 59.3.3.1, ABZ-Phase Output Inversion Function.

59.2.5 PCIF Maximum Position Setting Register (POSMAXR)

Address(es): PCIF0.POSMAXR 8782 7010h, PCIF1.POSMAXR 8782 7090h,
PCIF0_NS.POSMAXR 0782 7010h, PCIF1_NS.POSMAXR 0782 7090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POSMAX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	POSMAX[15:0]	Maximum Position Setting	In phase counting mode: The set value is used as the maximum value of the position counter (PHCPCMR.PHCPOSCNTM[15:0] bits). In incremental mode: The set value is used as the maximum value of the output counter (INCPCMR.INCOCNTM[15:0] bits) when the internal counters are cleared (INCZGCTLR.INCZAGE bit = 1). A value that meets “4 × N – 1 (N is an integer equal to or greater than 4)” can be set in these bits.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register sets the maximum position of the output counter.
Rewriting this register is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.6 PCIF Start/Stop Trigger Register (TRGR)

Address(es): PCIF0.TRGR 8782 7014h, PCIF1.TRGR 8782 7094h,
PCIF0_NS.TRGR 0782 7014h, PCIF1_NS.TRGR 0782 7094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PHCSTPT	—	—	—	—	—	—	—	PCIFSTPT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCIFSTRT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCIFSTRT	PCIF Start Trigger	0: Does not operate. 1: Starts outputs by the PCIF.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PCIFSTPT	PCIF Stop Trigger	0: Does not operate. 1: Immediately stops outputs by the PCIF.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	PHCSTPT	Phase Counting Mode Stop Trigger	0: Does not operate. 1: Stops outputs by the PCIF after generating ABZ-phase signals set in the PHCECR.PHCEDGCNT[15:0] bits.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of outputs by the PCIF and stop of outputs exclusively for use in phase counting mode.

PCIFSTRT Bit (PCIF Start Trigger)

This bit is a PCIF start trigger bit for use in phase counting mode and incremental mode. This bit is read as 0. Setting this bit to 1 at the same time as other bits is prohibited.

PCIFSTPT Bit (PCIF Stop Trigger)

This bit is a PCIF stop trigger bit for use in phase counting mode and incremental mode. In phase counting mode, setting this bit to 1 immediately stops outputs by the PCIF.
This bit is read as 0. Setting this bit to 1 at the same time as other bits is prohibited.
For details on how to stop outputs in phase counting mode, refer to (1) Immediate Stop Function in section 59.3.1.2, Stop Control. For details on how to stop outputs in incremental mode, refer to section 59.3.2.1, Startup/Stop Control.

PHCSTPT Bit (Phase Counting Mode Stop Trigger)

This bit is a PCIF stop trigger bit for use in phase counting mode. Setting this bit to 1 stops outputs by the PCIF in synchronization with the carrier period.
Carrier period events will not be accepted until the PCIFSTRT bit is set to 1.
This bit is read as 0. Setting this bit to 1 at the same time as other bits is prohibited.
For details on how to stop outputs by using this bit, refer to (2) Carrier Period-Synchronized Stop Function in section 59.3.1.2, Stop Control.

59.2.7 PCIF Status Register (SR)

Address(es): PCIF0.SR 8782 7018h, PCIF1.SR 8782 7098h,
PCIF0_NS.SR 0782 7018h, PCIF1_NS.SR 0782 7098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCIFPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCIFPS	PCIF Operation Status Flag	0: PCIF is stopped. 1: PCIF is in operation.	R
b31 to b1	—	Reserved	These bits are read as 0.	R

This register is a status register that indicates the PCIF operating state.

59.2.8 PCIF Phase Counting Mode Control Register (PHCCTLR)

Address(es): PCIF0.PHCCTLR 8782 7020h, PCIF1.PHCCTLR 8782 70A0h, PCIF0_NS.PHCCTLR 0782 7020h, PCIF1_NS.PHCCTLR 0782 70A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PHCECACMD	—	—	—	—	—	—	—	PHCCEDAMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PHCZOS	PHCZW[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PHCZW[2:0]	Z-Phase Pulse Width Select in Phase Counting Mode	b2 b0 0 0 0: Z-phase output is prohibited. 0 0 1: Pulse output when the PHCPCMR.PHCPOSCNTM[15:0] bits are set to 0 0 1 0: Pulse output when the PHCPCMR.PHCPOSCNTM[15:0] bits are set to –1 or 0 (sync to A-phase) or set to 0 or 1 (sync to B-phase) 0 1 1: Pulse output when the PHCPCMR.PHCPOSCNTM[15:0] bits are set to –1, 0, or 1 1 0 0: Pulse output when the PHCPCMR.PHCPOSCNTM[15:0] bits are set to –1, 0, 1, or 2 (sync to A-phase) or set to –2, –1, 0, or 1 (sync to B-phase) Settings other than the above are prohibited.	R/W
b3	PHCZOS	Z-Phase Output Synchronization Phase Select in Phase Counting Mode	0: A-phase 1: B-phase	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PHCCEDAMD	Carrier Period Event Difference Adjustment Mode Select in Phase Counting Mode	0: Event-based mode 1: Register-based mode	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	PHCECACMD	Automatic Edge Count Clear Mode Select in Phase Counting Mode	0: Automatic clearing is not performed. 1: Automatic clearing is performed.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects the operating mode in phase counting mode.

PHCZW[2:0] Bits (Z-Phase Pulse Width Select in Phase Counting Mode)

These bits set the pulse width of Z-phase output in phase counting mode. Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on this function, refer to section 59.3.1.5, Generating A-, B-, and Z-Phase Signals.

PHCZOS Bit (Z-Phase Output Synchronization Phase Select in Phase Counting Mode)

This bit selects the phase with which the Z-phase output is to be synchronized in phase counting mode. Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on this function, refer to section 59.3.1.5, Generating A-, B-, and Z-Phase Signals.

PHCCEDAMD Bit (Carrier Period Event Difference Adjustment Mode Select in Phase Counting Mode)

This bit selects the processing to be performed if the intervals between carrier period events differ from the carrier period. In event-based mode, A-phase and B-phase are generated when a carrier period event is input. In register-based mode, A-phase and B-phase are generated for each carrier period set in the PHCPERIODR.PHCPERIOD[15:0] bits. In both modes, the AB-phase change intervals are generated according to the setting of the PHCPERIOD[15:0] bits. Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on the event-based mode, refer to section 59.3.1.3, Carrier Period Control, (1) Event-Based Mode. For details on the register-based mode, refer to section 59.3.1.3, Carrier Period Control, (2) Register-Based Mode.

PHCECACMD Bit (Automatic Edge Count Clear Mode Select in Phase Counting Mode)

This bit selects the automatic clear mode for the PHCECR.PHCEDGCNT[15:0] bits. Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on the automatic clearing function, refer to section 59.3.1.4, Position Counter Controll, (2) Automatic Edge Count Clearing Function.

59.2.9 Carrier Period Event Difference Threshold Select Register for PCIF Phase Counting Mode (PHCCEDTHR)

Address(es): PCIF0.PHCCEDTHR 8782 7024h, PCIF1.PHCCEDTHR 8782 70A4h, PCIF0_NS.PHCCEDTHR 0782 7024h, PCIF1_NS.PHCCEDTHR 0782 70A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHCCEDTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PHCCEDTH[15:0]	Carrier Period Event Difference Threshold in Phase Counting Mode	Set the tolerance of difference between the interval at which the carrier period event is generated and the carrier period setting.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register sets the carrier period event difference threshold in phase counting mode.

PHCCEDTH[15:0] Bits (Carrier Period Event Difference Threshold in Phase Counting Mode)

These bits set the tolerance of difference between the interval at which the carrier period event is generated and the carrier period setting (PHCPERIODR.PHCPERIOD[15:0] bits). Setting a value no less than the PHCPERIOD[15:0] bit value is prohibited.

If either of the following conditions is satisfied, an error occurs (PHCSR.PHCCEES flag = 1).

Carrier period event generation interval < PHCPERIOD[15:0] bits – PHCCEDTH[15:0] bits, or

Carrier period event generation interval > PHCPERIOD[15:0] bits + PHCCEDTH[15:0] bits

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.10 PCIF Phase Counting Mode Carrier Period Setting Register (PHCPERIODR)

Address(es): PCIF0.PHCPERIODR 8782 7028h, PCIF1.PHCPERIODR 8782 70A8h,
PCIF0_NS.PHCPERIODR 0782 7028h, PCIF1_NS.PHCPERIODR 0782 70A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHCPERIOD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PHCPERIOD[15:0]	Phase Counting Mode Carrier Period Setting	Set the carrier period.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register sets the carrier period in phase counting mode.

PHCPERIOD[15:0] Bits (Phase Counting Mode Carrier Period Setting)

These bits set the carrier period in cycles of the frequency-divided clock (CCLK_DIV) in accordance with the average period of event input from the ELC or an expected interval at which the carrier period event is generated. Specify the setting so that the carrier period is 10 µs or longer (maximum value is CCLK_DIV period × 65,535). If the average period of event input from the ELC is not divisible by the frequency-divided clock (CCLK_DIV) cycle, set the nearest value of the quotient.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.11 PCIF Phase Counting Mode Edge Count Setting Register (PHCECR)

Address(es): PCIF0.PHCECR 8782 702Ch, PCIF1.PHCECR 8782 70ACh,
PCIF0_NS.PHCECR 0782 702Ch, PCIF1_NS.PHCECR 0782 70ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHCEDGCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PHCEDGCNT[15:0]	Edge Count Setting in Phase Counting Mode	Set the number of output edges for the next carrier period.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register sets the number of output edges for the next carrier period in phase counting mode.

PHCEDGCNT[15:0] Bits (Edge Count Setting in Phase Counting Mode)

Set the number of output edges for the next carrier period.
For reverse rotation, set a negative value by using the two's-complement format. The range of settable values is from – PHCPERIOD[15:0] bits to PHCPERIOD[15:0] bits.

If the automatic clear mode is disabled (PHCCTLR.PHCECACMD bit = 0), the value is retained even when a carrier period event is received. If the automatic clear mode is enabled (PHCCTLR.PHCECACMD bit = 1), the value is set to 0000h when a carrier period event is received.
Writing to these bits causes the status flag corresponding to the edge count setting register (PHCSR.PHCECUS) to be set to 1.

59.2.12 PCIF Phase Counting Mode Position Count Monitor Register (PHCPCMR)

Address(es): PCIF0.PHCPCMR 8782 7030h, PCIF1.PHCPCMR 8782 70B0h,
PCIF0_NS.PHCPCMR 0782 7030h, PCIF1_NS.PHCPCMR 0782 70B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHCPOSCNTM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PHCPOSCNTM[15:0]	Position Count Monitor in Phase Counting Mode	Position count monitor in phase counting mode	R
b31 to b16	—	Reserved	These bits are read as 0.	R

This register monitors the position count in phase counting mode.

PHCPOSCNTM[15:0] Bits (Position Count Monitor in Phase Counting Mode)

These bits indicate the current position information in phase counting mode.
The available range is from 0 to the POSMAXR.POSMAX[15:0] bit setting. The value of these bits is not guaranteed in incremental mode.

59.2.13 PCIF Phase Counting Mode Position Counter Write Register (PHCPCWDR)

Address(es): PCIF0.PHCPCWDR 8782 7034h, PCIF1.PHCPCWDR 8782 70B4h,
PCIF0_NS.PHCPCWDR 0782 7034h, PCIF1_NS.PHCPCWDR 0782 70B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHCPOSCNTWD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PHCPOSCNTWD[15:0]	Position Counter Write Data in Phase Counting Mode	Set the current position information.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is the position counter write register in phase counting mode.

PHCPOSCNTWD[15:0] Bits (Position Counter Write Data in Phase Counting Mode)

These bits are used to write data to the position counter in phase counting mode.
The available range is from 0 to the POSMAXR.POSMAX[15:0] bit setting.
The value written to this register is reflected to the PHCPCMR.PHCPOSCNTM[15:0] bits. Writing to these bits during operation causes the status flag corresponding to the position counter write register (PHCSR.PHCPCUS) to be set to 1.
The timing at which the value written to these bits is reflected to the PHCPOSCNTM[15:0] bits varies depending on the PCIF state. If the PCIF is stopped (until the phase counting mode startup processing flow*1), the value written to these bits is immediately reflected to the PHCPOSCNTM[15:0] bits. If the PCIF is operating (during main processing flow*2), the value written to these bits is reflected to the PHCPOSCNTM[15:0] bits when the next carrier period event is generated. For internal fetch timing, refer to Figure 59.16 PHCOENS Flag Waveform.

Note 1. Even during main processing, if the carrier period event generation intervals are extremely long, as shown in Figure 59.20, the input delay might exceed the sum of the PHCPERIOD[15:0] and PHCCEDTH[15:0] bits. In this case, the value written to these bits is immediately reflected to the PHCPOSCNTM[15:0] bits in the same way as when the PCIF is stopped.
Note 2. Once the flow of stop processing has started, writing to these bits after the previous carrier period event is ignored. If necessary, repeat the writing.

59.2.14 PCIF Phase Counting Mode Status Register (PHCSR)

Address(es): PCIF0.PHCSR 8782 7038h, PCIF1.PHCSR 8782 70B8h,
PCIF0_NS.PHCSR 0782 7038h, PCIF1_NS.PHCSR 0782 70B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	PHCOECUES	PHCIECOES	PHCIECES	PHCCEES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PHCECUS	PHCPCUS	—	—	—	—	—	—	—	PHCOENS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PHCOENS	Output Edge Count Status Flag in Phase Counting Mode	0: The number of edges specified in the PHCECR.PHCEDGCNT[15:0] bits has been output. 1: The number of edges specified in the PHCECR.PHCEDGCNT[15:0] bits is being output.	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	PHCPCUS	Position Count Available Status Flag in Phase Counting Mode	0: The current PHCPCWDR.PHCPOSCNTWD[15:0] value has been written to the PHCPCMR.PHCPOSCNTM[15:0] bits (including the state in which writing is not required after a reset and immediate write when PCIF is not operating). 1: Data was written to the PHCPCWDR.PHCPOSCNTWD[15:0] bits during PCIF operation, but was not written to the PHCPCMR.PHCPOSCNTM[15:0] bits.	R
b9	PHCECUS	Edge Count Available Status Flag in Phase Counting Mode	0: The current PHCECR.PHCEDGCNT[15:0] bits have been or are being used for the ABZ-phase output processing. 1: The current PHCECR.PHCEDGCNT[15:0] bits have not been used for the ABZ-phase output processing.	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b16	PHCCEES	Carrier Period Event Error Status Flag in Phase Counting Mode	0: The carrier period event is normal. 1: The carrier period event has an error. This bit is set to 1 when either of the following conditions is met: Carrier period event generation interval < PHCPERIODR.PHCPERIOD[15:0] bits – PHCCEDTHR.PHCCEDTH[15:0] bits, or Carrier period event generation interval > PHCPERIODR.PHCPERIOD[15:0] bits + PHCCEDTHR.PHCCEDTH[15:0] bits	R
b17	PHCIECES	Internal Edge Count Error Status Flag in Phase Counting Mode	0: The number of remaining edges is normal. 1: Error in the number of remaining edges (number of remaining edges > PHCPERIODR.PHCPERIOD[15:0] bits)	R
b18	PHCIECOES	Internal Edge Count Overflow Error Status Flag in Phase Counting Mode	0: The number of remaining edges is normal. 1: Error in the number of remaining edges (overflow)	R
b19	PHCOECUES	Old Edge Count Usage Error Status Flag in Phase Counting Mode	0: Normal PHCECR.PHCEDGCNT[15:0] bits 1: PHCECR.PHCEDGCNT[15:0] bit error (when processing was started with the PHCEDGCNT[15:0] bits not updated)	R
b31 to b20	—	Reserved	These bits are read as 0.	R

This register is a status register in phase counting mode.

For details on each status flag, refer to section 59.3.1.6, States.

59.2.15 PCIF Phase Counting Mode Status Clear Register (PHCSCR)

Address(es): PCIF0.PHCSCR 8782 703Ch, PCIF1.PHCSCR 8782 70BCh, PCIF0_NS.PHCSCR 0782 703Ch, PCIF1_NS.PHCSCR 0782 70BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	PHCOECUESC	PHCIECOESC	PHCIECESC	PHCCEESC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PHCCEESC	Carrier Period Event Error Status Clear in Phase Counting Mode	Writing 1 to this bit clears the PHCSR.PHCCEES flag. This bit is read as 0.	R/W
b17	PHCIECESC	Internal Edge Count Error Status Clear in Phase Counting Mode	Writing 1 to this bit clears the PHCSR.PHCIECES flag. This bit is read as 0.	R/W
b18	PHCIECOESC	Internal Edge Count Overflow Error Status Clear in Phase Counting Mode	Writing 1 to this bit clears the PHCSR.PHCIECOES flag. This bit is read as 0.	R/W
b19	PHCOECUESC	Old Edge Count Usage Error Status Clear in Phase Counting Mode	Writing 1 to this bit clears the PHCSR.PHCOECUES flag. This bit is read as 0.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a status clear register in phase counting mode.

59.2.16 PCIF Phase Counting Mode Status Enable Register (PHCSER)

Address(es): PCIF0.PHCSER 8782 7040h, PCIF1.PHCSER 8782 70C0h,
PCIF0_NS.PHCSER 0782 7040h, PCIF1_NS.PHCSER 0782 70C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	PHCOECUESE	PHCIECOESE	PHCIECESE	PHCCEESE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PHCCEESE	Carrier Period Event Error Status Enable in Phase Counting Mode	0: The PHCSR.PHCCEES flag is disabled (fixed to 0). 1: The PHCSR.PHCCEES flag is enabled.	R/W
b17	PHCIECESE	Internal Edge Count Error Status Enable in Phase Counting Mode	0: The PHCSR.PHCIECES flag is disabled (fixed to 0). 1: The PHCSR.PHCIECES flag is enabled.	R/W
b18	PHCIECOESE	Internal Edge Count Overflow Error Status Enable in Phase Counting Mode	0: The PHCSR.PHCIECOES flag is disabled (fixed to 0). 1: The PHCSR.PHCIECOES flag is enabled.	R/W
b19	PHCOECUESE	Old Edge Count Usage Error Status Enable in Phase Counting Mode	0: The PHCSR.PHCOECUES flag is disabled (fixed to 0). 1: The PHCSR.PHCOECUES flag is enabled.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the status flags in phase counting mode.
Rewriting this register is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.17 PCIF Incremental Mode Control Register (INCCTLR)

Address(es): PCIF0.INCCTLR 8782 7050h, PCIF1.INCCTLR 8782 70D0h,
PCIF0_NS.INCCTLR 0782 7050h, PCIF1_NS.INCCTLR 0782 70D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	INCRSL[5:0]						INCZOMIN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	INCZOS[1:0]		—	INCOMD	INCIMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	INCIMD[1:0]	ABZ-Phase Input Mode Select in Incremental Mode	b1 b0 0 0: Mode 0 (AB-phase pulses) 0 1: Mode 1 (A-phase pulse) (B-phase rotation direction) 1 0: Mode 2 (pulse switching) 1 1: Setting prohibited	R/W
b2	INCOMD	ABZ-Phase Output Mode Select in Incremental Mode	0: Same as the input mode 1: Mode 0 (AB-phase pulses)	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	INCZOS[1:0]	Z-Phase Output Synchronous Phase Select in Incremental Mode	b5 b4 0 0: Immediate output 0 1: Synchronized with A-phase (output protocol mode 0 only) 1 0: Synchronized with B-phase (output protocol mode 0 only) 1 1: Synchronized with pulse (output protocol modes 1 and 2 only)	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	INCZOMIN[7:0]	Z-Phase Output Minimum Cycle Setting in Incremental Mode	00h: 1 cycle 01h: 2 cycles 02h: 3 cycles : : FFh: 256 cycles	R/W
b29 to b24	INCRSL[5:0]	Incremental Mode Resolution Ratio Select	00h: Same size 01h: ×1/2 02h: ×1/3 03h: ×1/4 : : 3Fh: ×1/64	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a control register in incremental mode.

INCIMD[1:0] Bits (ABZ-Phase Input Mode Select in Incremental Mode)

These bits are used to select the AB-phase input protocol mode.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on input protocol modes, refer to section 59.3.2.2, Generating A- and B-Phase Signals, (1) I/O Protocol Change Function.

INCOMD Bit (ABZ-Phase Output Mode Select in Incremental Mode)

This bit selects the AB-phase output protocol mode.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

For details on output protocol modes, refer to section 59.3.2.2, Generating A- and B-Phase Signals, (1) I/O Protocol Change Function.

INCZOS[1:0] Bits (Z-Phase Output Synchronous Phase Select in Incremental Mode)

These bits are used to select the phase to be synchronized with Z-phase output.
Setting 11b is prohibited in output protocol mode 0. Setting 01b or 10b is prohibited in output protocol modes 1 and 2.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).
For details on this function, refer to section 59.3.2.3, Generating Z-Phase Signals, (2) Selecting the Z-Phase Output Synchronization Method. Table 59.4 lists the output protocol modes that can be used with the INCZOS[1:0] bits.

Table 59.4 Output Protocol Modes Available with the INCZOS[1:0] Bits

INCIMD[1:0] Bits	INCOMD Bit	Output Protocol Mode	INCZOS[1:0] Bit Settings			
			00b	01b	10b	11b
00b	×	Mode 0	Available	Available	Available	Not available
01b	0	Mode 1	Available	Not available	Not available	Available
01b	1	Mode 0	Available	Available	Available	Not available
10b	0	Mode 2	Available	Not available	Not available	Available
10b	1	Mode 0	Available	Available	Available	Not available

×: Don’t care

INCZOMIN[7:0] Bits (Z-Phase Output Minimum Cycle Setting in Incremental Mode)

These bits are used to set the minimum number of cycles for Z-phase output.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).
For details on this function, refer to section 59.3.2.3, Generating Z-Phase Signals, (3) Setting the Minimum Pulse Width of Z-Phase Output.

INCRSL[5:0] Bits (Incremental Mode Resolution Ratio Select)

These bits are used to select the resolution ratio of the AB(Z)-phase input and AB(Z)-phase output.
For details on this function, refer to section 59.3.2.2, Generating A- and B-Phase Signals, (2) Internal Counter Operation, (b) Resolution Changing Function.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.18 PCIF Incremental Mode Z-Phase Generation Control Register (INCZGCTLR)

Address(es): PCIF0.INCZGCTLR 8782 7054h, PCIF1.INCZGCTLR 8782 70D4h, PCIF0_NS.INCZGCTLR 0782 7054h, PCIF1_NS.INCZGCTLR 0782 70D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	INCSWZGE	INCVIZGE	INCPIZGE	INCZIZGE	INCZAGE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INCZAGE	Incremental Mode Z-Phase Automatic Generation Enable	0: Z-phase generation with the INCPCMR.INCOCNTM[15:0] bits is disabled. 1: Z-phase generation with the INCPCMR.INCOCNTM[15:0] bits is enabled.	R/W
b1	INCZIZGE	Incremental Mode Z-Phase Generation with Z-Phase Input Enable	0: Z-phase generation by using Z-phase input is disabled. 1: Z-phase generation by using Z-phase input is enabled.	R/W
b2	INCPIZGE	Incremental Mode Z-Phase Generation with Crest Event Trigger Input Enable	0: Z-phase generation with crest event trigger input is disabled. 1: Z-phase generation with crest event trigger input is enabled.	R/W
b3	INCVIZGE	Incremental Mode Z-Phase Generation with Trough Event Trigger Input Enable	0: Z-phase generation with trough event trigger input is disabled. 1: Z-phase generation with trough event trigger input is enabled.	R/W
b4	INCSWZGE	Incremental Mode Z-Phase Generation with Software Trigger Enable	0: Z-phase generation with software trigger is disabled. 1: Z-phase generation with software trigger is enabled.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is a control register for Z-phase generation in incremental mode.

INCZAGE Bit (Incremental Mode Z-Phase Automatic Generation Enable)

This is the enable bit for Z-phase automatic generation with the internal counter INCOCNTM[15:0] bits.

Setting 1 is prohibited when the INCZIZGE bit = 1.

When the INCOCNTM[15:0] bit setting is equal to the POSMAX[15:0] bit setting, the INCOCNTM[15:0] bits are automatically set to 0000h.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCZIZGE Bit (Incremental Mode Z-Phase Generation with Z-Phase Input Enable)

This is the enable bit for Z-phase generation by using Z-phase input.

Setting 1 is prohibited when the INCZAGE bit = 1.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCPIZGE Bit (Incremental Mode Z-Phase Generation with Crest Event Trigger Input Enable)

This is the enable bit for Z-phase generation with crest event trigger input.

Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCVIZGE Bit (Incremental Mode Z-Phase Generation with Trough Event Trigger Input Enable)

This is the enable bit for Z-phase generation with trough event trigger input.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCSWZGE Bit (Incremental Mode Z-Phase Generation with Software Trigger Enable)

This is the enable bit for Z-phase generation with a software trigger.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.19 PCIF Incremental Mode Position Counter Clear Enable Register (INCPCCER)

Address(es): PCIF0.INCPCCER 8782 7058h, PCIF1.INCPCCER 8782 70D8h, PCIF0_NS.INCPCCER 0782 7058h, PCIF1_NS.INCPCCER 0782 70D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	INCSW CLRE	INCVIC LRE	INCPIC LRE	INCZIC LRE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	INCZICLRE	Incremental Mode Z-Phase Input Position Counter Clear Enable	0: The counter is not cleared. 1: The counter is cleared.	R/W
b2	INCPICLRE	Incremental Mode Crest Event Trigger Input Position Counter Clear Enable	0: The counter is not cleared. 1: The counter is cleared.	R/W
b3	INCVICLRE	Incremental Mode Trough Event Trigger Input Position Counter Clear Enable	0: The counter is not cleared. 1: The counter is cleared.	R/W
b4	INCSWCLRE	Incremental Mode Software Trigger Position Counter Clear Enable	0: The counter is not cleared. 1: The counter is cleared.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register clears the position counter (INCPCMR.INCOCNTM[15:0] bits) in incremental mode.

INCZICLRE Bit (Incremental Mode Z-Phase Input Position Counter Clear Enable)

This bit clears the INCOCNTM[15:0] bits with Z-phase input. Setting 1 is prohibited when the INCZGCTLR.INCZAGE bit = 1. Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCPICLRE Bit (Incremental Mode Crest Event Trigger Input Position Counter Clear Enable)

This bit clears the INCOCNTM[15:0] bits with crest event trigger input.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCVICLRE Bit (Incremental Mode Trough Event Trigger Input Position Counter Clear Enable)

This bit clears the INCOCNTM[15:0] bits with trough event trigger input.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

INCSWCLRE Bit (Incremental Mode Software Trigger Position Counter Clear Enable)

This bit clears the INCOCNTM[15:0] bits with a software trigger.
Rewriting is prohibited when the PCIF is in operation (SR.PCIFPS flag = 1).

59.2.20 PCIF Incremental Mode Software Trigger Register (INCSWTRGR)

Address(es): PCIF0.INCSWTRGR 8782 705Ch, PCIF1.INCSWTRGR 8782 70DCh, PCIF0_NS.INCSWTRGR 0782 705Ch, PCIF1_NS.INCSWTRGR 0782 70DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INCSWT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INCSWT	Incremental Mode Software Trigger	0: Does not operate. 1: Generates a software trigger (the read value is always 0).	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register generates software triggers in incremental mode.

59.2.21 PCIF Incremental Mode Position Counter Monitor Register (INCPCMR)

Address(es): PCIF0.INCPCMR 8782 7060h, PCIF1.INCPCMR 8782 70E0h,
PCIF0_NS.INCPCMR 0782 7060h, PCIF1_NS.INCPCMR 0782 70E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	INCROTM	—	INCICNTM[5:0]						—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INCOCNTM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	INCOCNTM[15:0]	Incremental Mode Output Position Counter Monitor	Monitor bits of the output counter	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b29 to b24	INCICNTM[5:0]	Incremental Mode Input Position Counter Monitor	Monitor bits of the input counter	R
b30	—	Reserved	This bit is read as 0.	R
b31	INCROTM	Incremental Mode Rotation Direction Monitor	0: Forward direction (clockwise (CW)) 1: Reverse direction (counterclockwise (CCW))	R

This register monitors the I/O counters and rotation direction in incremental mode.

INCOCNTM[15:0] Bits (Incremental Mode Output Position Counter Monitor)

These are monitor bits of the output counter.

A- and B-phase signals are output based on the 2 lower-order bits of this register. The counter is incremented when the INCICNTM[5:0] setting changes from the maximum value (INCCTLR.INCRSL[5:0] bits) to 0, and is decremented when the INCICNTM[5:0] setting changes from 0 to the maximum value (INCRSL[5:0] bits).

Operation of these bits is not guaranteed in phase counting mode.

INCICNTM[5:0] Bits (Incremental Mode Input Position Counter Monitor)

These are monitor bits of the input counter.

The INCOCNTM[15:0] bit value is incremented and decremented based on this register and the resolution is changed.

Position information for the AB(Z)-phase input can be obtained with the following formula.

Position information for AB(Z)-phase input = INCOCNTM[15:0] bits × (INCRSL[5:0] bits + 1) + INCICNTM[5:0] bits

INCROTM Bit (Incremental Mode Rotation Direction Monitor)

This monitor bit indicates the current rotation direction.

Operation of this bit is not guaranteed in phase counting mode.

59.2.22 PCIF Software Reset Register (SWRSTR)

Address(es): PCIF0.SWRSTR 8782 7070h, PCIF1.SWRSTR 8782 70F0h, PCIF0_NS.SWRSTR 0782 7070h, PCIF1_NS.SWRSTR 0782 70F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWRST	Software Reset	0: Releases the registers from the software reset state. 1: Applies a software reset to the registers.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register performs a software reset.

SWRST Bit (Software Reset)

This bit applies a software reset to registers that operate with the frequency-divided clock (CCLK_DIV). Table 59.5 lists the registers that can be cleared by a software reset.

Table 59.5 Registers That Can be Cleared by Software Reset

Register Name	Bit Name	Clearing by Software Reset	Value following a Software Reset
CLKSR	CLKS	Not applicable	The previous value is retained.
SR	PCIFPS	Applicable	0
PHCECR	PHCEDGCNT[15:0]	Applicable	0000h
PHCPCMR	PHCPOSCNTM[15:0]	Applicable	0000h
PHCPCWDR	PHCPOSCNTWD[15:0]	Applicable	0000h
PHCSR	PHCOENS	Applicable	0
	PHCPCUS	Applicable	0
	PHCECUS	Applicable	0
	PHCCEES	Not applicable	The previous value is retained.
	PHCIECES	Not applicable	The previous value is retained.
	PHCIECOES	Not applicable	The previous value is retained.
	PHCOECUES	Not applicable	The previous value is retained.
INCPCMR	INCOCNTM[15:0]	Applicable	0000h
	INCICNTM[5:0]	Applicable	00h
	INCROTM	Applicable	0
Other than above	—	Not applicable	The previous value is retained.

59.3 Operation

59.3.1 Phase Counting Mode

When the CTLR.PCIFMD bit is set to 0, the PCIF operates in phase counting mode.
In phase counting mode, ABZ-phase signals are output according to the register settings from the CPU that obtained the motor position information.
In this mode, A-, B-, and Z-phase signals are generated by using two types of information: a carrier period event and the edge count setting register (PHCECR.PHCEDGCNT[15:0] bits).
A carrier period event is a signal that is asserted at the unit time interval for measuring the number of edges. The PHCEDGCNT[15:0] bits indicate the number of edges in the AB-phase inputs generated by rotating the motor during the interval between the carrier period events.
If a carrier period event is generated after setting the PHCEDGCNT[15:0] bits, counting is performed at regular intervals until the value of the position counter (PHCPCMR.PHCPOSCNTM[15:0] bits) reaches the value of (*2) in the following expression after PHCPERIODR.PHCPERIOD[15:0] × CCLK_DIV cycles.
Figure 59.4 shows details of the values of (*1) to (*3) in the following expression. For details, refer to section 59.3.1.4, Position Counter Control.

PHCPOSCNTM[15:0] bits (*2) = PHCPOSCNTM[15:0] bits (*1) + PHCEDGCNT[15:0] bits (*3)

Figure 59.4 shows the basic operation waveforms in phase counting mode.

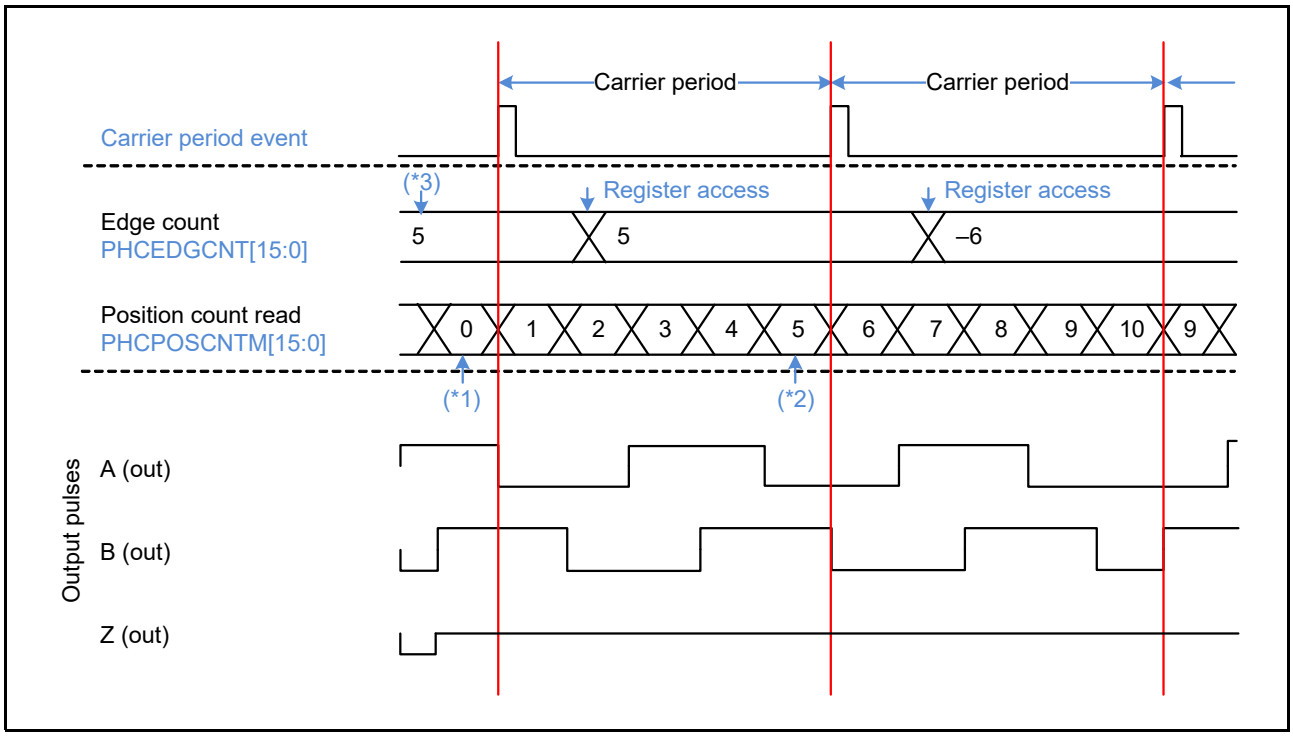


Figure 59.4 Basic Operation Waveforms in Phase Counting Mode

59.3.1.1 Startup Control

The phase counting mode is started by using the TRGR.PCIFSTRT bit. When the CTLR.PCIFMD bit = 0, setting the PCIFSTRT bit to 1 starts the phase counting mode and sets the SR.PCIFPS flag to 1. When the phase counting mode is started, A-, B-, and Z -phase signals are output in accordance with the position counter (PHCPCMR.PHCPOSCNTM[15:0] bits). When a carrier period event is generated, the position counter operates and then the ABZ-phase outputs change according to the counter value.
For details on the startup processing flow, refer to section 59.3.4.3, Flows of Phase Counting Mode.

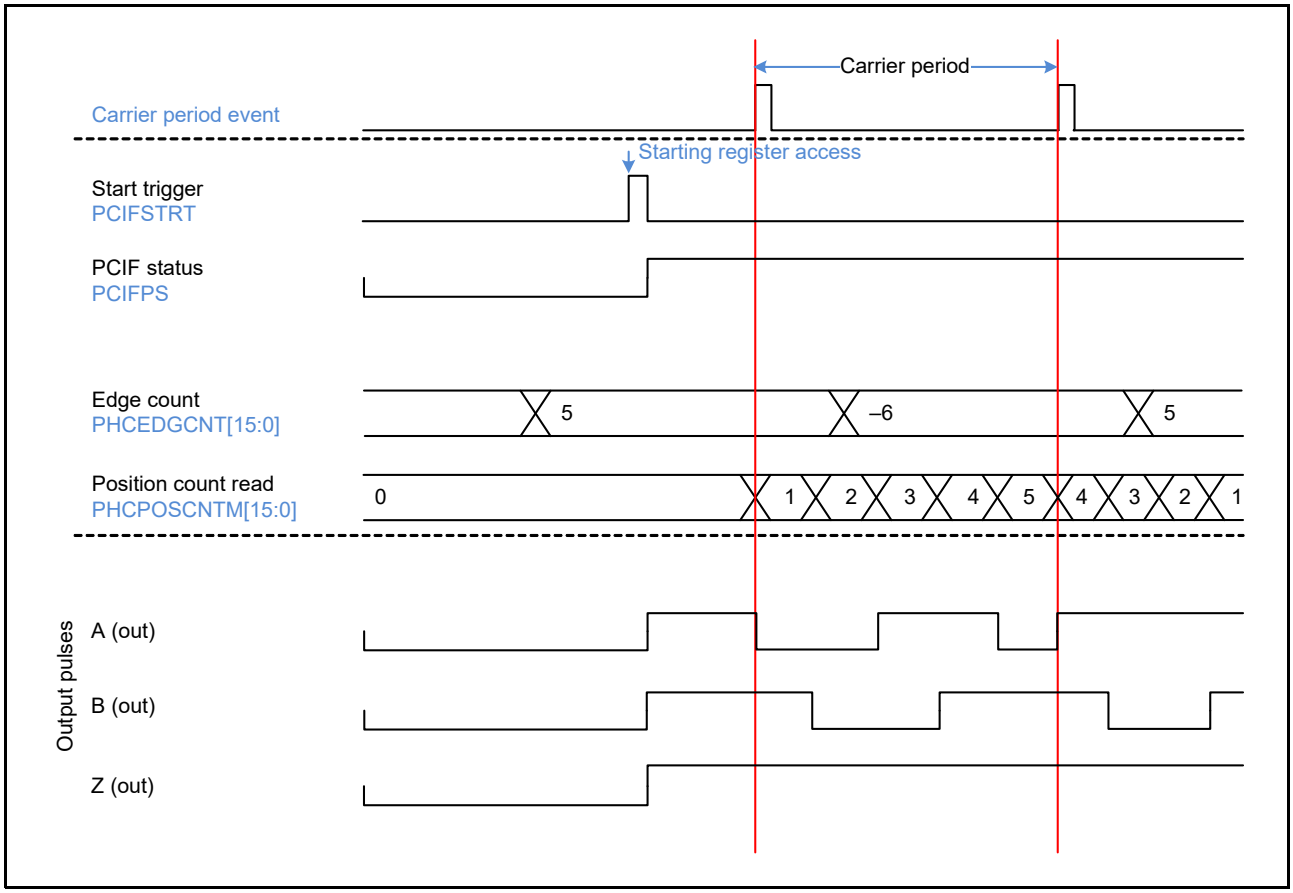


Figure 59.5 Startup Waveforms in Phase Counting Mode

59.3.1.2 Stop Control

The phase counting mode can be stopped in two ways. One way uses the immediate stop function with the TRGR.PCIFSTPT bit. The other way uses the carrier period-synchronized stop function with the TRGR.PHCSTPT bit. The following explains stop control. For details on the stop processing flow, refer to section 59.3.4.3, Flows of Phase Counting Mode.

(1) Immediate Stop Function

If the PCIFSTPT bit is set to 0 in phase counting mode, the SR.PCIFPS flag is set to 0 and ABZ-phase outputs stop irrespective of the current processing.

Figure 59.6 shows the operation waveforms at the time of immediate stop.

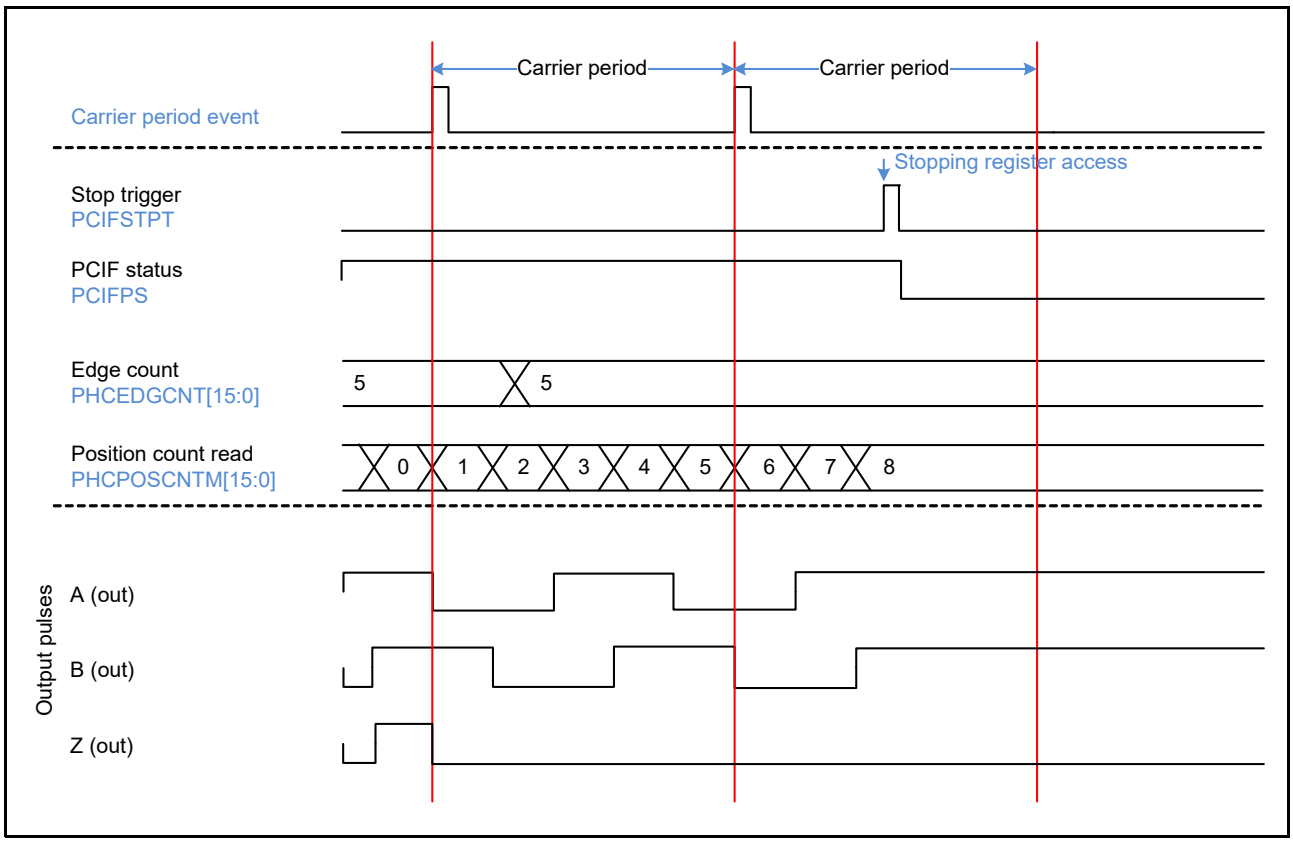


Figure 59.6 Operation Waveforms at Immediate Stop of Phase Counting Mode

(2) Carrier Period-Synchronized Stop Function

If the TRGR.PHCSTPT bit is set to 1 in phase counting mode, the SR.PCIFPS flag is set to 0 and ABZ-phase outputs stop after the current edge count operation ends. For details on how to calculate the total number of edges, refer to section 59.3.1.3, Carrier Period Control, (1) Event-Based Mode.

Figure 59.7 shows the operation waveforms of the carrier period-synchronized stop function.

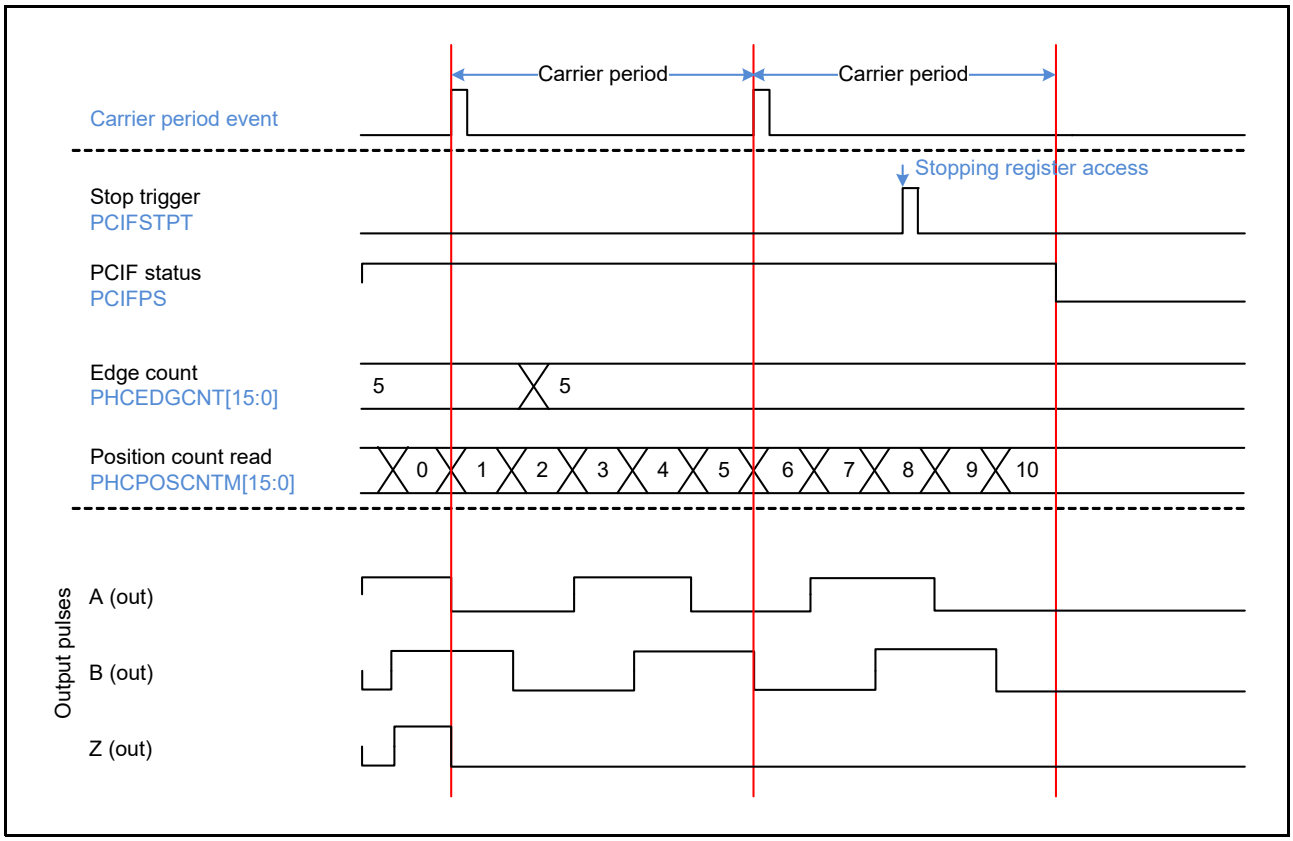


Figure 59.7 Operation Waveforms of the Carrier Period-Synchronized Stop Function in Phase Counting Mode

59.3.1.3 Carrier Period Control

The carrier period is determined by the interval at which a carrier period event is generated.
There are two methods of controlling the carrier periods: event-based mode that uses input pins as carrier period events, and register-based mode that uses register settings. These modes can be switched by using the PHCCTLR.PHCCEDAMD bit.

(1) Event-Based Mode

Setting the PHCCEDAMD bit to 0 sets event-based mode.
In event-based mode, the edge count set in the PHCECR.PHCEDGCNT[15:0] bits is processed within the carrier period set in the PHCPERIODR.PHCPERIOD[15:0] bits from the time of notification of a carrier period input by the ELC.
Figure 59.8 shows the operation waveforms in event-based mode.

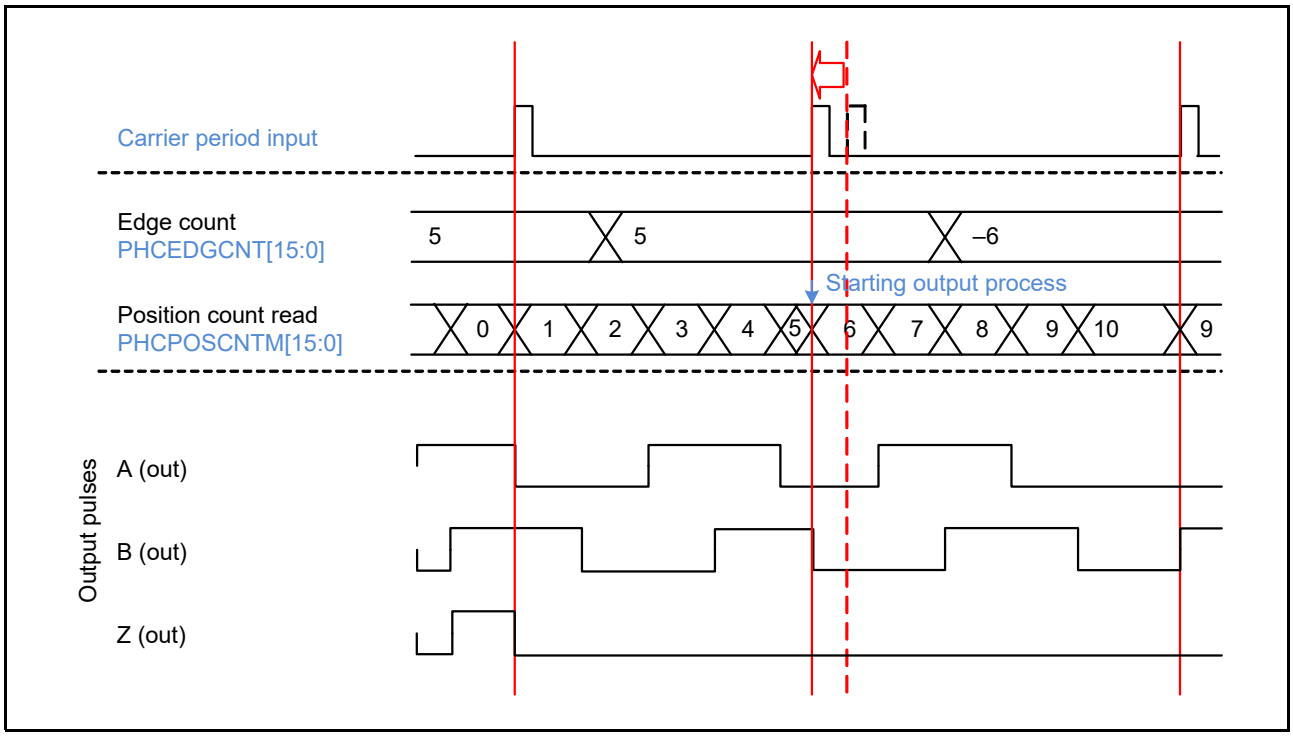


Figure 59.8 Operation Waveforms in Event-Based Mode

If the next carrier period event is received before completion of edge count output set in the PHCECR.PHCEDGCNT[15:0] bits, the current processing is suspended. Then, the next processing is performed with the total edge count calculated from the following expression.

Total edge count = current PHCEDGCNT[15:0] bits + number of unprocessed edges when processing was suspended

Each parameter has a sign indicating the rotation direction. There might be cases in which the rotation direction differs before and after suspension. In such cases, ABZ-phase signals are output for the difference obtained by the above expression. For example, if the number of unprocessed edges when processing was suspended is +2 and the current PHCEDGCNT[15:0] bits value is –5, the total edge count is –3. As a result, 3 edges in reverse direction will be generated.
Figure 59.9 shows the waveforms when the next carrier period event is received before completion of edge count output.

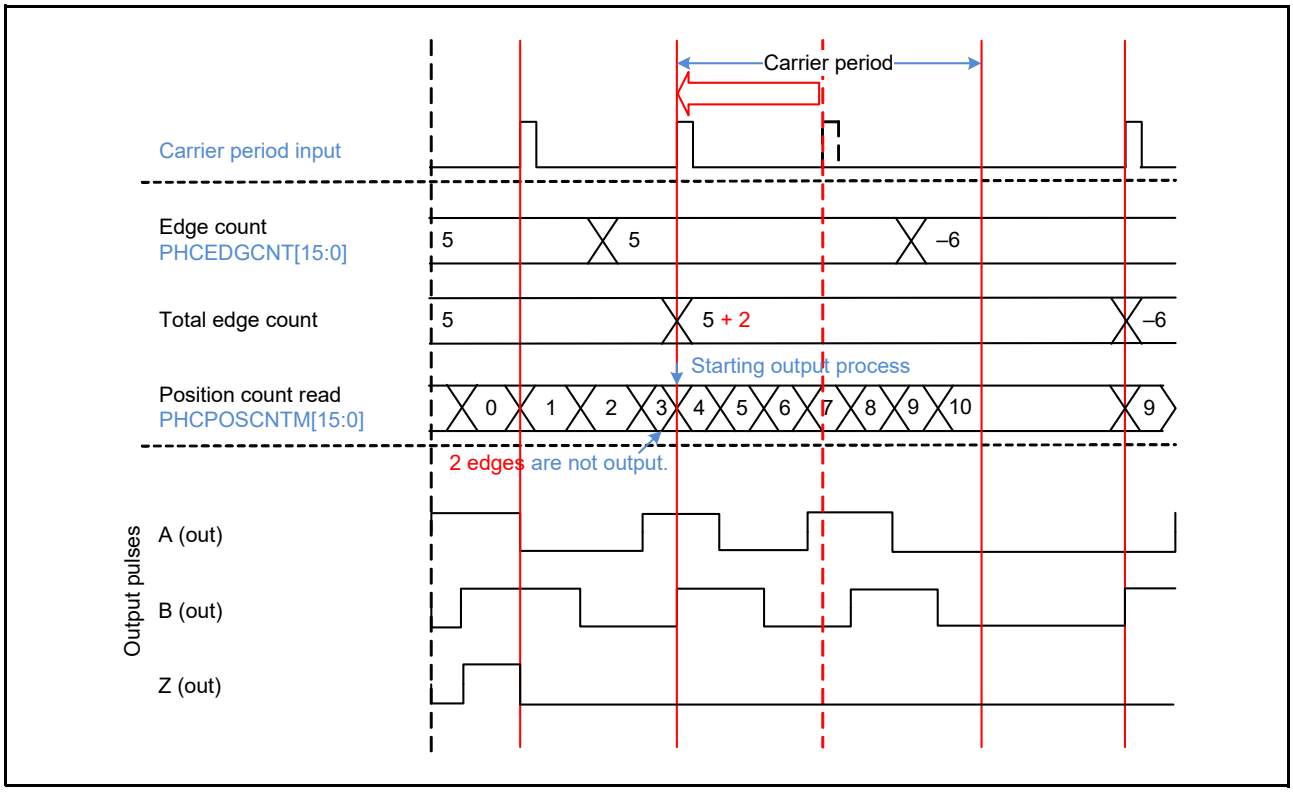


Figure 59.9 Event-Based Mode Waveforms 1 (When the Next Carrier Period Event is Received before Completion of Edge Count Output)

If a carrier period input is delayed, processing stops until a notification of the next carrier period input. Figure 59.10 shows the waveforms in event-based mode when a carrier period input is delayed.

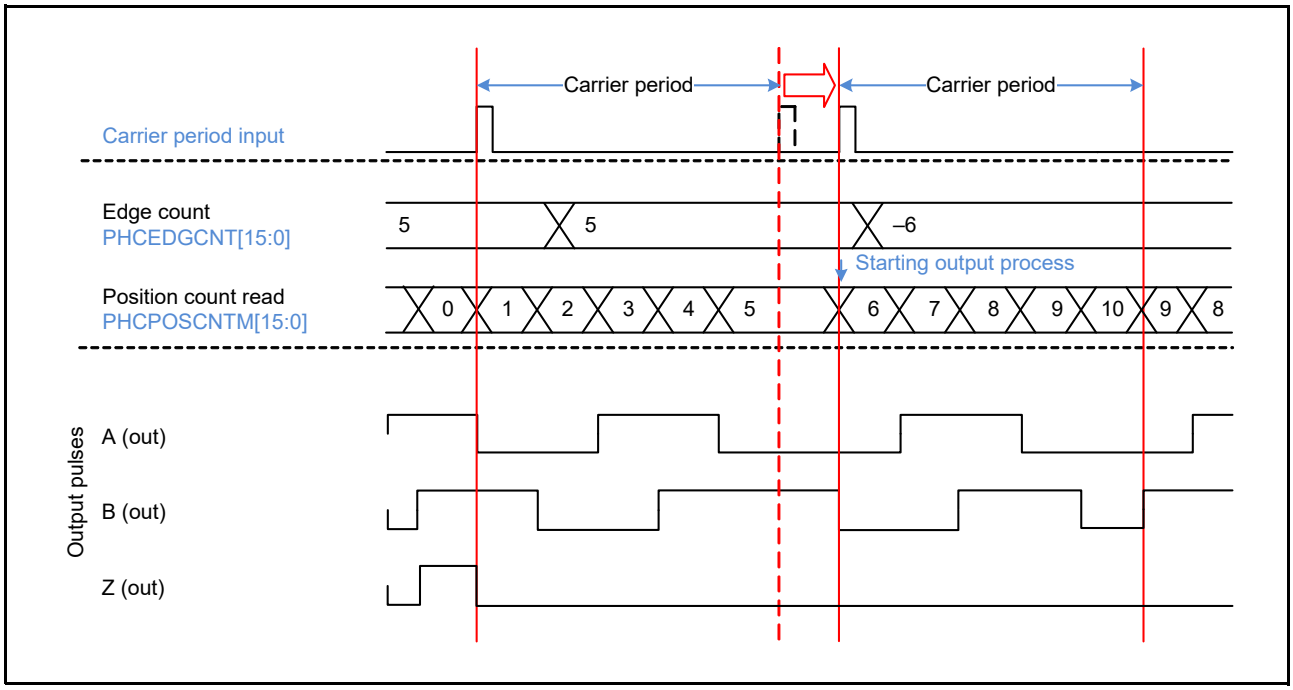


Figure 59.10 Event-Based Mode Waveforms 2 (When a Carrier Period Input is Delayed)

(2) Register-Based Mode

Setting the PHCCTLR.PHCCEDAMD bit to 1 sets the register-based mode.
In register-based mode, carrier period input is not used. Instead, the edge count set in the PHCECR.PHCEDGCNT[15:0] bits is processed at the carrier period intervals set in the PHCPERIODR.PHCPERIOD[15:0] bits. Measurement of the carrier period starts when the PCIF operation is started by setting the TRGR.PCIFSTRT bit to 1 in booting of the PCIF.
Figure 59.11 shows the waveforms in register-based mode.

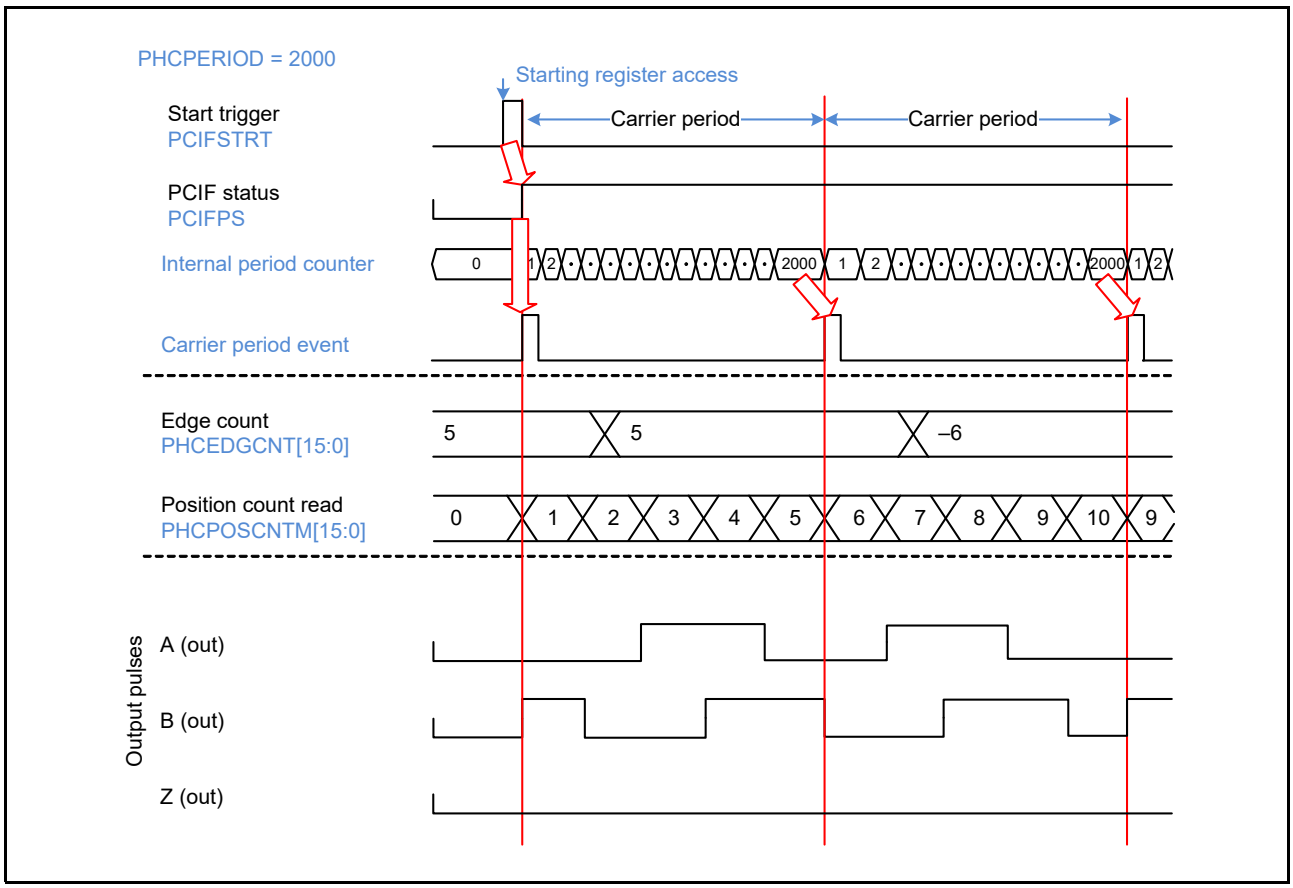


Figure 59.11 Register-Based Mode Waveforms

59.3.1.4 Position Counter Control

(1) Edge Processing for Carrier Periods

The following describes the position counter (PHCPCMR.PHCPOSCNTM[15:0] bits) and how to output the edge count set in the PHCECR.PHCEDGCNT[15:0] bits.

For each carrier period, the total edge count (X) calculated from the PHCEDGCNT[15:0] bits (as described in section 59.3.1.3, Carrier Period Control) is processed. When a carrier period event is generated, the position counter (PHCPOSCNTM[15:0] bits) starts counting. The counting is performed at regular intervals until the position counter (PHCPOSCNTM[15:0] bits) reaches “PHCPOSCNTM[15:0] setting + X” by the carrier period set in the PHCPERIODR.PHCPERIOD[15:0] bits. The position count change intervals (edge intervals) change at (PHCPERIOD[15:0] bits/X) cycles. The fractional parts of the change intervals are distributed to each edge interval to adjust the position counter. The adjusted position counter is used to output the corresponding AB-phase signals.

Figure 59.12 shows the total edge count X and the A-phase and B-phase output change timing for forward rotation (X > 0). Figure 59.13 shows the total edge count X and the A-phase and B-phase output change timing for reverse rotation (X < 0).

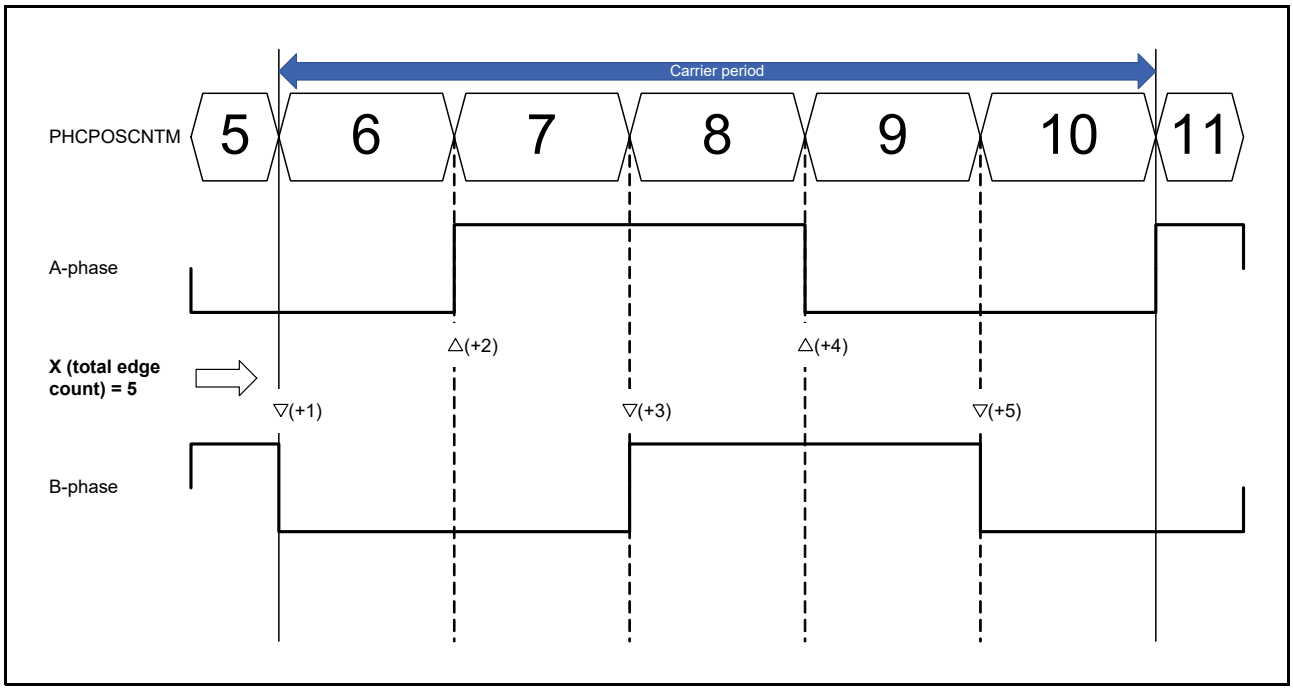


Figure 59.12 AB-Phase Outputs after Setting the PHCEDGCNT[15:0] Bits (Forward Rotation)

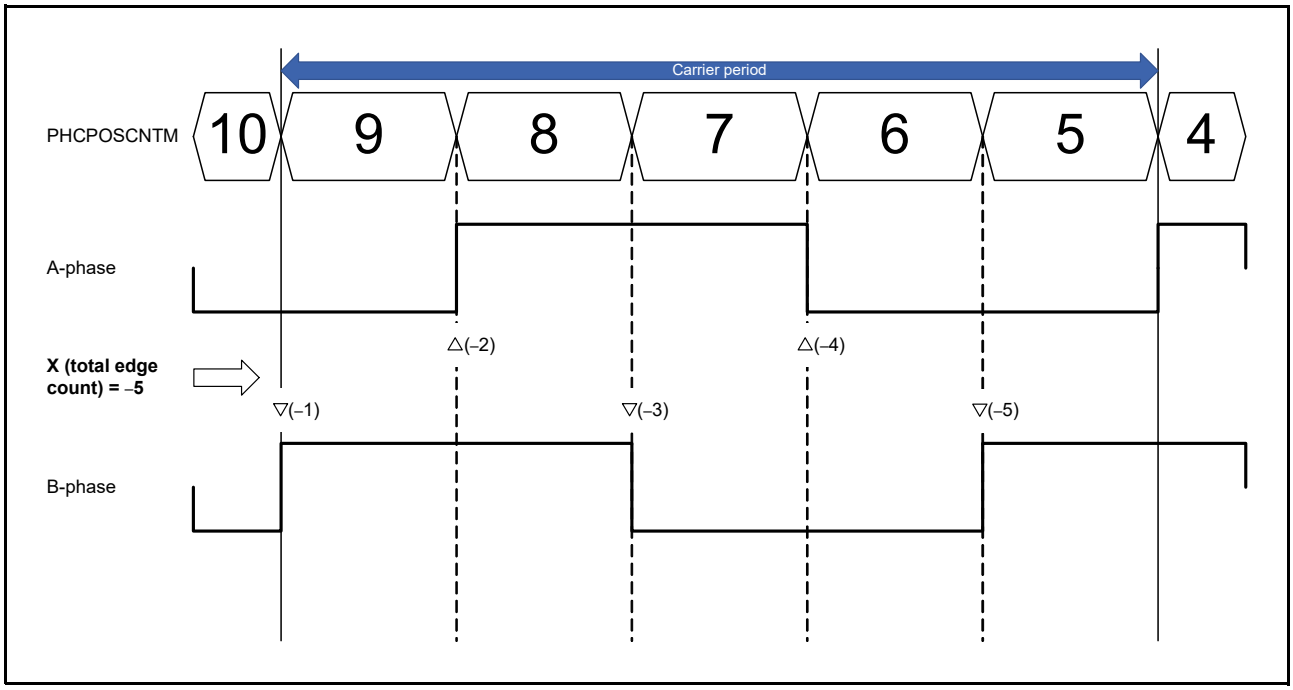


Figure 59.13 AB-Phase Outputs after Setting the PHCEDGCNT[15:0] Bits (Reverse Rotation)

(2) Automatic Edge Count Clearing Function

The automatic edge count clearing function automatically clears the PHCECR.PHCEDGCNT[15:0] bits when a carrier period event is generated. When this function is enabled, if the next carrier period event is generated without updating the PHCEDGCNT[15:0] bits, ABZ-phase outputs stop because processing is performed with the value after clearing (PHCEDGCNT[15:0] bits = 0000h). This function is enabled by setting the PHCCTLR.PHCECACMD bit to 1. Figure 59.14 shows the waveforms of the automatic edge count clearing function.

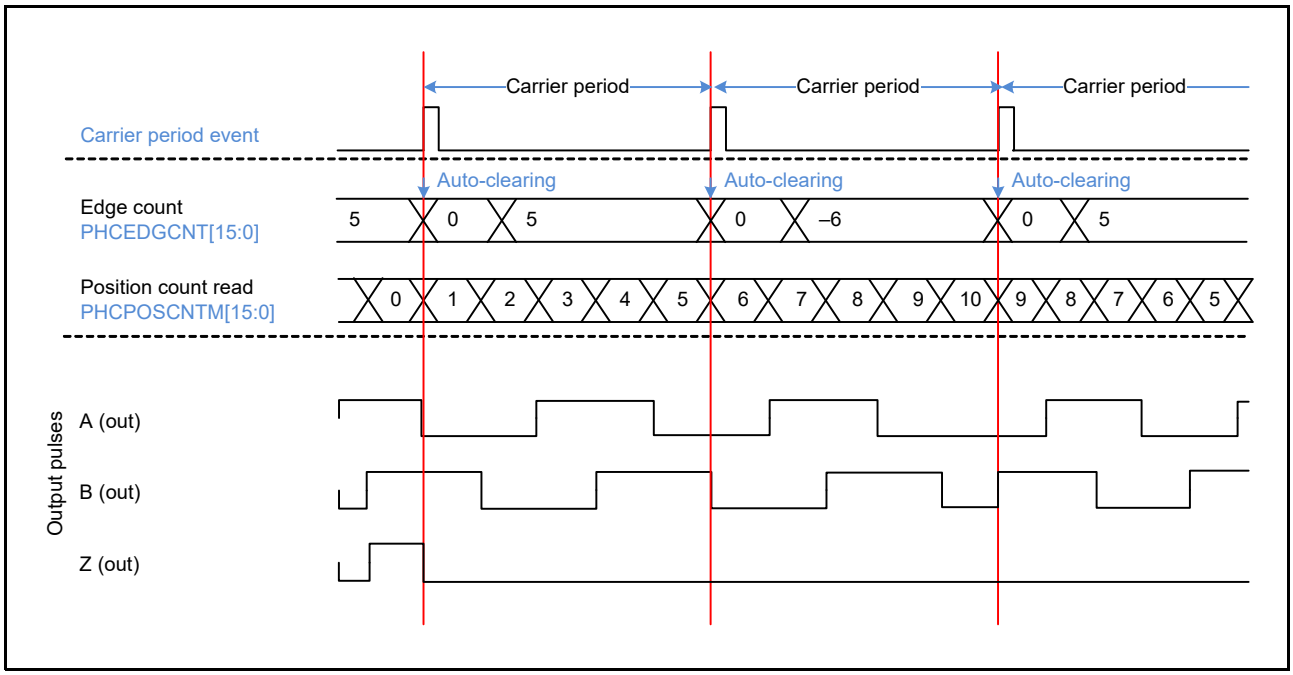


Figure 59.14 Waveforms of the Automatic Edge Count Clearing Function

59.3.1.5 Generating A-, B-, and Z-Phase Signals

A-phase and B-phase signals are uniquely output according to the register settings and the two lower-order bits of the position counter (PHCPCMR.PHCPOSCNTM[1:0] bits). For negative phase output, set the CTLR.APOL and BPOL bits to 1.

Table 59.6 shows the A- and B-phase output patterns.

Table 59.6 A- and B-Phase Output Patterns in Phase Counting Mode

PHCPOSCNTM[1:0] Bits	00b	01b	10b	11b
A-phase output (APOL bit = 0: Positive phase)	1	0	0	1
A-phase output (APOL bit = 1: Negative phase)	0	1	1	0
B-phase output (BPOL bit = 0: Positive phase)	1	1	0	0
B-phase output (BPOL bit = 1: Negative phase)	0	0	1	1

Figure 59.15 shows the correspondence between the PHCPOSCNTM[15:0] bit value and the ABZ-phase output waveforms.

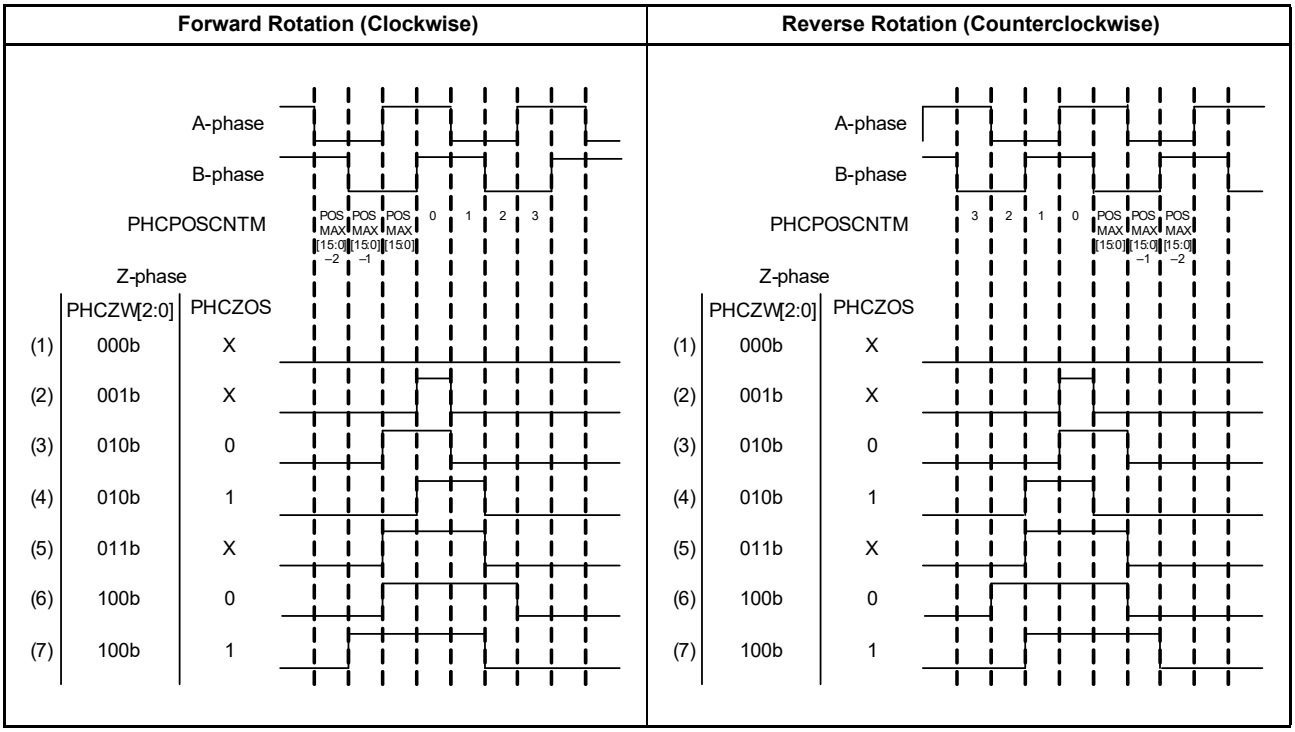


Figure 59.15 ABZ-Phase Output Waveforms in Phase Counting Mode

59.3.1.6 States

(1) Normal Operation

The phase counting mode has three types of status flags.

- Output Edge Count Status Flag in Phase Counting Mode (PHCOENS)
  This status flag indicates whether any edges have not been output yet.

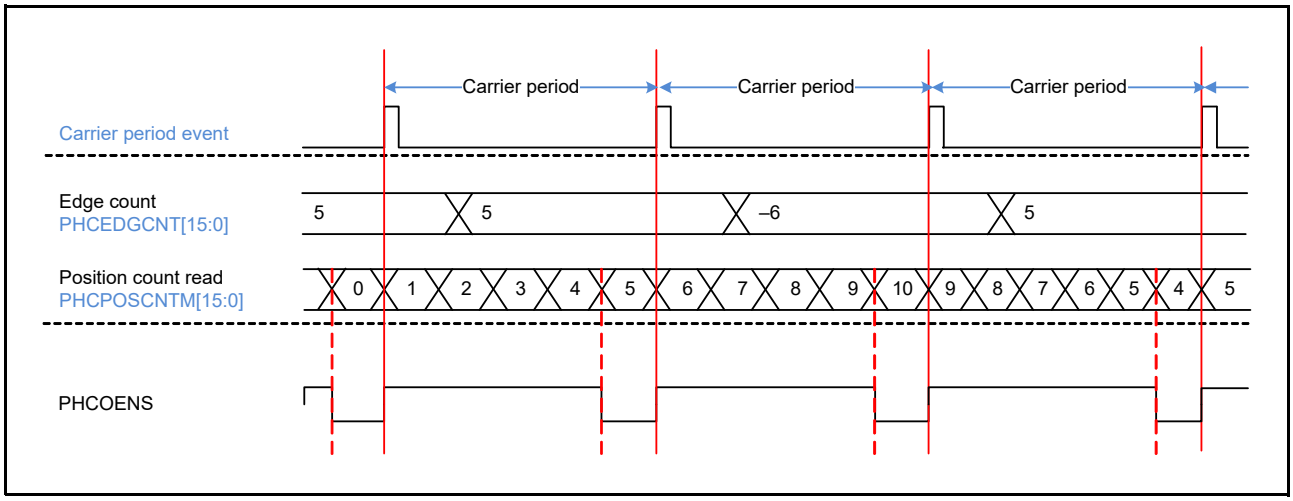


Figure 59.16 PHCOENS Flag Waveform

- Position Count Available Status Flag in Phase Counting Mode (PHCPCUS)
  The PHCSR.PHCPCUS flag indicates that the values written to the PHCPCWDR.PHCPOSCNTWD[15:0] bits have not been reflected to the PHCPCMR.PHCPOSCNTM[15:0] bits.
  When the PCIF is not operating (other than the main processing flow), this flag always indicates 0 because the value written to the PHCPOSCNTWD[15:0] bits is immediately reflected to the PHCPOSCNTM[15:0] bits. When the PCIF is operating (in the main processing flow), this flag is set by writing to the PHCPOSCNTWD[15:0] bits, and is cleared when a carrier period event is generated.

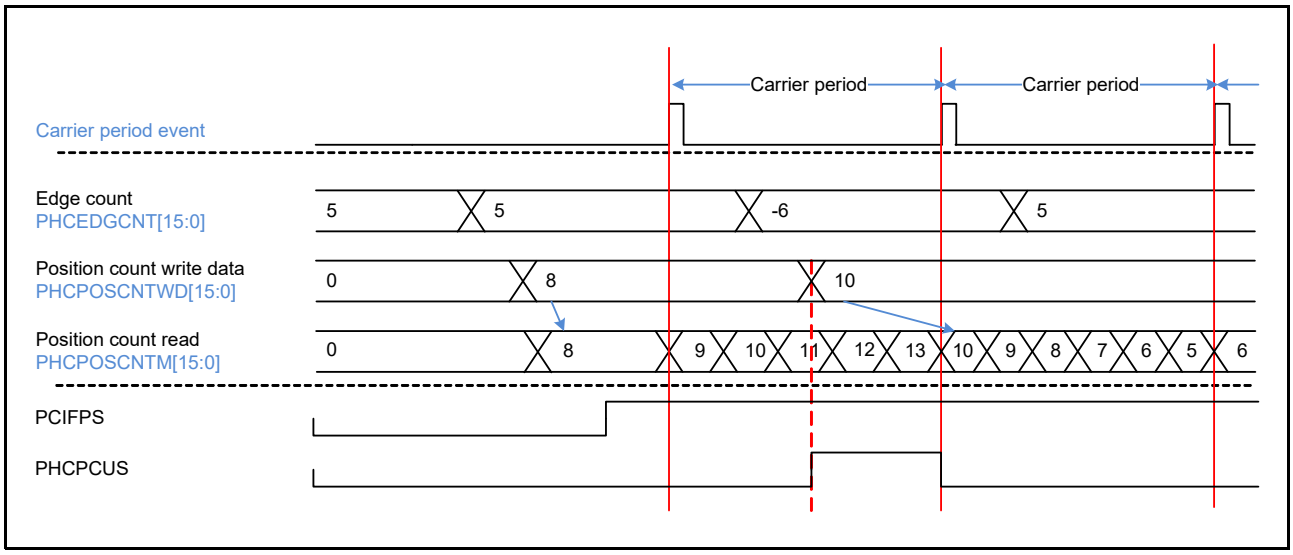


Figure 59.17 PHCPCUS Flag Waveform

Note: If the PCIF is stopped (PCIFPS flag = 0) when the PHCPCUS flag is 1, the PHCPCUS flag is set to 0.

- Edge Count Available Status Flag in Phase Counting Mode (PHCECUS)
  The PHCSR.PHCECUS flag indicates the period over which the value written to the PHCECR.PHCEDGCNT[15:0] bits is held pending until it is used for ABZ-phase output.
  This flag is set by writing to the PHCEDGCNT[15:0] bits, and is cleared when a carrier period event is generated.

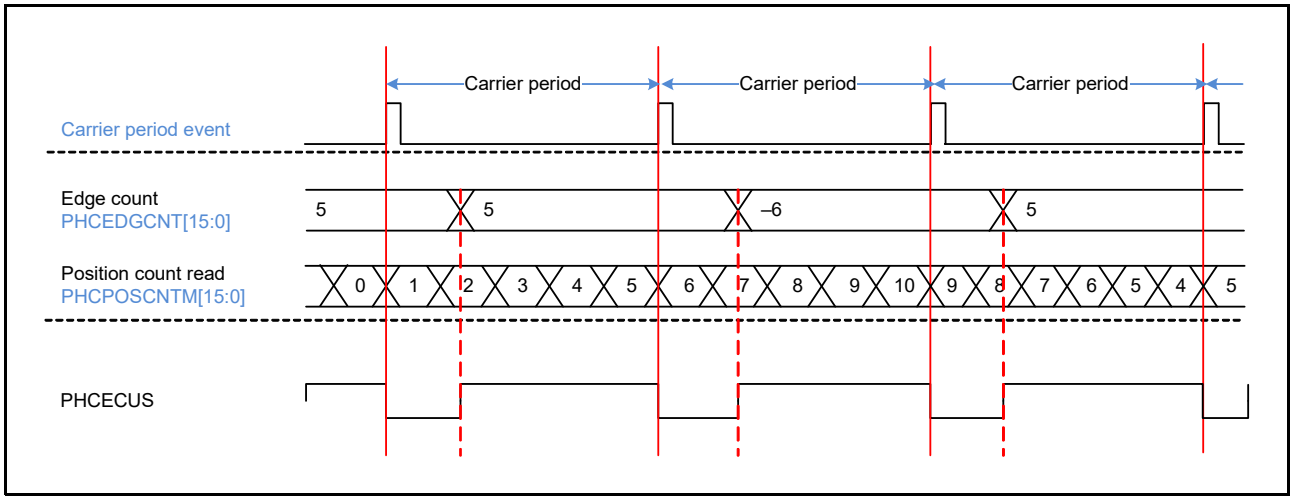


Figure 59.18 PHCECUS Flag Waveform

Note: If the PCIF is stopped (PCIFPS flag = 0) when the PHCECUS flag is 1, the PHCECUS flag is retained as 1.

(2) Error Operations

The phase counting mode has four types of error status flags.

Table 59.7 shows the error determination timings of each of the error status flags and the processing to be performed if an error occurs.

If an error indicated in this section has occurred, operation is not guaranteed. Apply a software reset to clear the error states, and then confirm the register settings.

Table 59.7 Processing When an Error Occurs

Error Status Flag	Error Determination Timing	Processing When Error Occurs
PHCCEES	A carrier period event is generated at an interval shorter than the number of cycles obtained by "PHCPERIOD[15:0] setting – PHCCEDTH[15:0] setting" since the last carrier period event.	The carrier period event is not accepted and processing resumes from the next carrier period event.
	The number of cycles obtained by "PHCPERIOD[15:0] setting + PHCCEDTH[15:0] setting" has elapsed before a carrier period event is input.	Processing resumes from the delayed carrier period event.
PHCIECES	A carrier period event is generated.	Unprocessed edges are taken over to the next carrier period event.
PHCIECOES	A carrier period event is generated.	Edges for the overflow are lost and processing is performed with the maximum value.
PHCOECUES	A carrier period event is generated.	—

Note: Setting the error status flags is delayed relative to the error determination timings.

- Carrier Period Event Error Status Flag in Phase Counting Mode (PHCCEES)
  The PHCSR.PHCCEES flag indicates that the interval at which carrier period events are generated is extremely long or short. This flag is set when the absolute value of the difference between the carrier period event interval and the carrier period (PHCPERIODR.PHCPERIOD[15:0] bits) is greater than the number of cycles set in the PHCCEDTHR.PHCCEDTH[15:0] bits.
  If this error occurs when the interval at which carrier period events are generated is extremely short, the carrier period event that caused the error is not accepted.

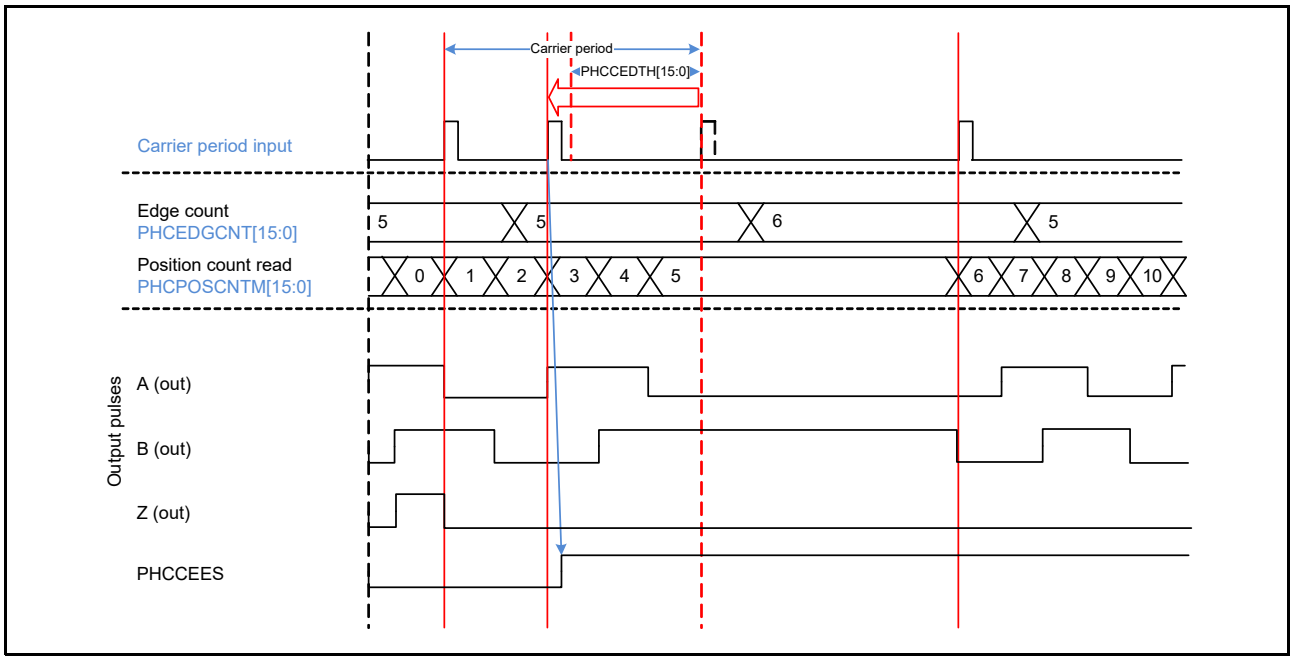


Figure 59.19 Waveforms for Occurrence of a PHCCEES Error Caused by Early Carrier Period Input

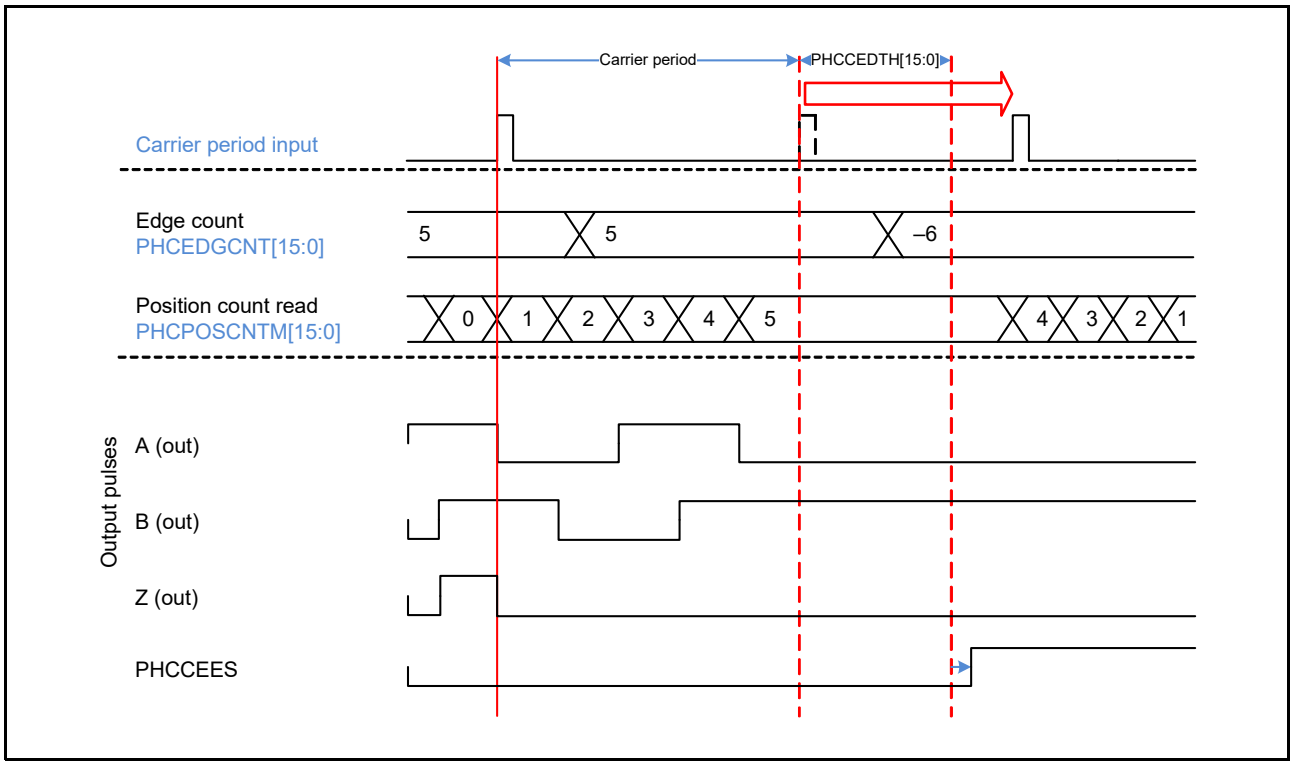


Figure 59.20 Waveforms for Occurrence of a PHCCEES Error Caused by Delayed Carrier Period Input

- Internal Edge Count Error Status Flag in Phase Counting Mode (PHCIECES)
  The PHCSR.PHCIECES flag indicates that pulses for the number of output edges set in the PHCECR.PHCEDGCNT[15:0] bits cannot be output within the carrier period.
  This flag is set when the carrier period event is generated at a short interval and the total edge count exceeds the carrier period (PHCPERIODR.PHCPERIOD[15:0] bits).
  If this error occurs and pulses for the current total edge count cannot be processed within the carrier period, the unprocessed edges are taken over to the next carrier period.

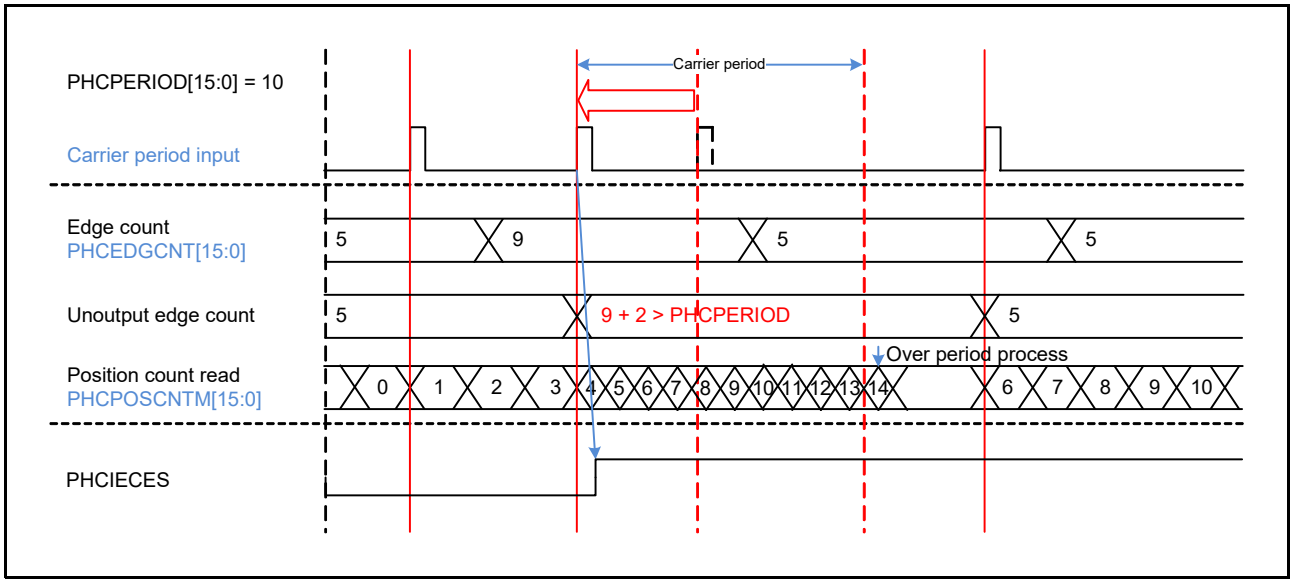


Figure 59.21 Waveforms for Occurrence of a PHCIECES Error

- Internal Edge Count Overflow Error Status Flag in Phase Counting Mode (PHCIECOES)
  The PHCSR.PHCIECOES flag indicates that output edges were lost due to an overflow or underflow of the total edge count to be output within the carrier period.
  This flag is set when the total edge count exceeding 32,767 resulted in an overflow. This flag is also set when the total edge count less than –32,768 resulted in an underflow.
  If this error is caused by an overflow, the total edge count is processed as 32,767. If this error is caused by an underflow, the total edge count is processed as –32,768.

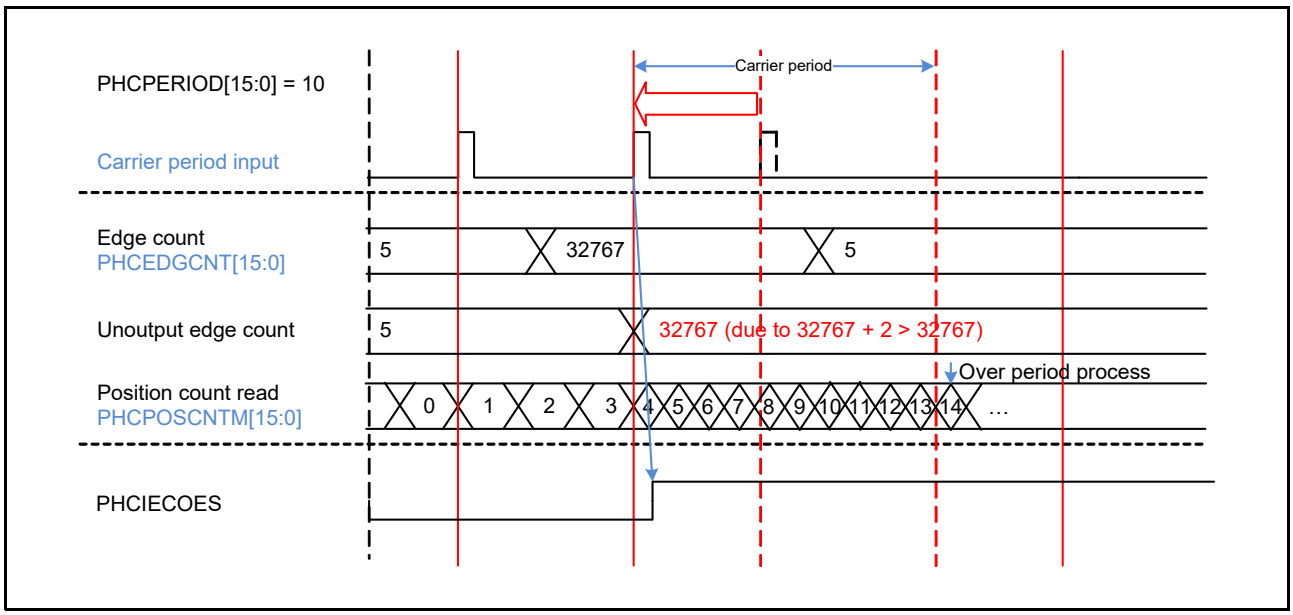


Figure 59.22 Waveforms for Occurrence of a PHCIECOES Error

- Old Edge Count Usage Error Status Flag in Phase Counting Mode (PHCOECUES)
  The PHCSR.PHCOECUES flag is set when, after a carrier period event is generated, the next carrier period event is generated without updating the PHCECR.PHCEDGCNT[15:0] bits.
  If this error occurs, the current value of the PHCEDGCNT[15:0] bits is used for the processing after occurrence of the error.
  If the automatic clearing function (PHCCTLR.PHCECACMD bit = 1) is used, the PHCEDGCNT[15:0] bits = 0000h is assumed for the processing and the ABZ-phase outputs stop.

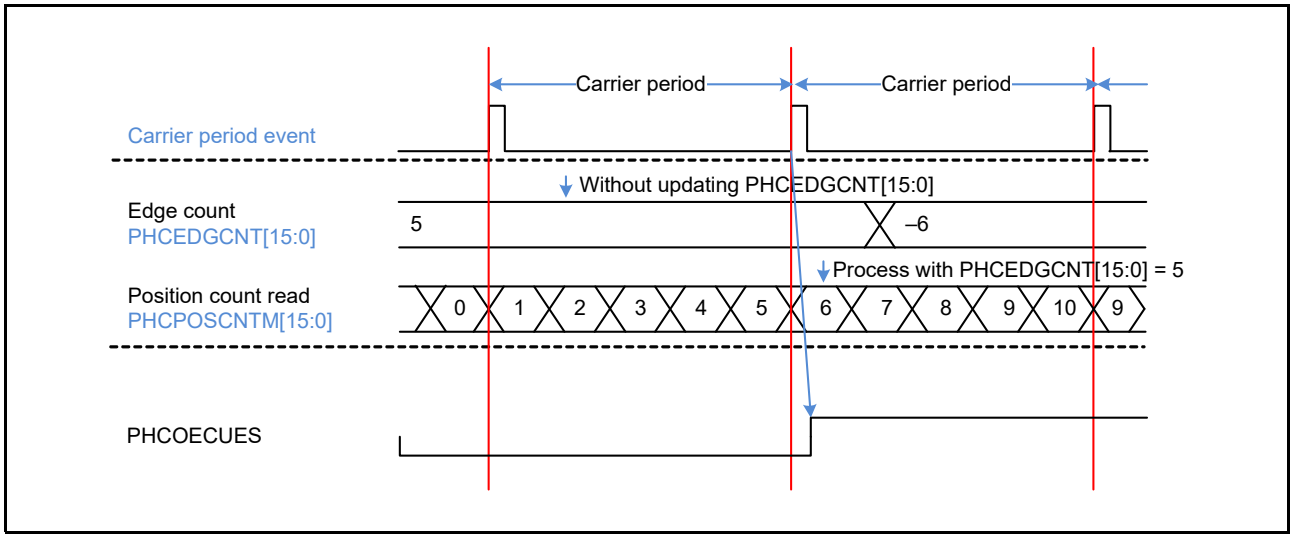


Figure 59.23 Waveforms for Occurrence of a PHCOECUES Error

59.3.2 Incremental Mode

When the CTLR.PCIFMD bit is set to 1, the PCIF operates in incremental mode.
In incremental mode, ABZ-phase output values are generated based on the ABZ-phase signals directly input from the motor without using the CPU. This mode also provides the resolution changing function and three types of I/O protocols to support various motors and host controllers.
Figure 59.24 shows the basic operation waveforms in incremental mode.

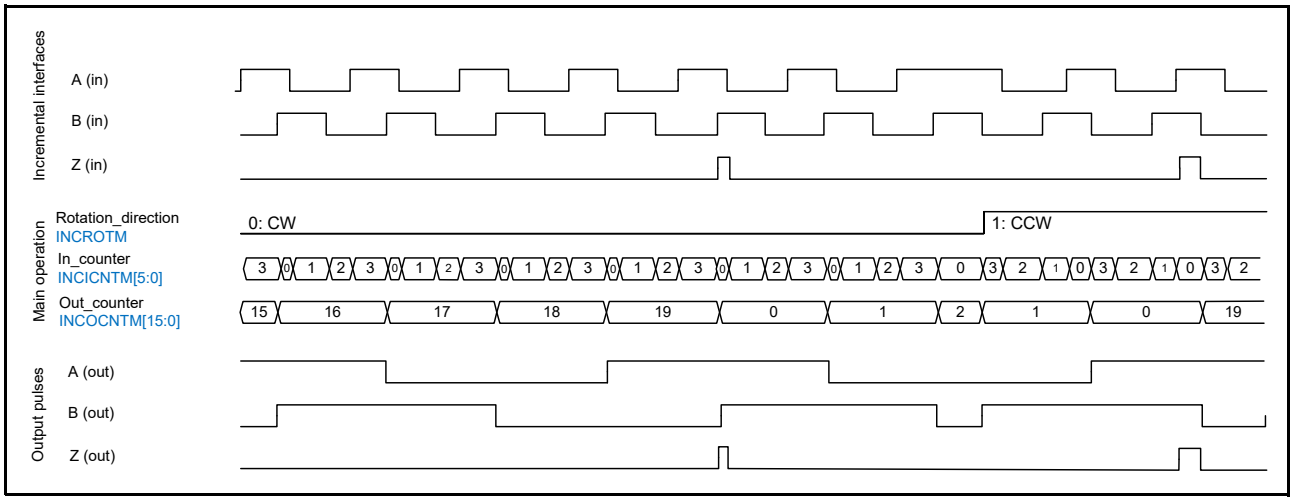


Figure 59.24 Basic Operation Waveforms in Incremental Mode

59.3.2.1 Startup/Stop Control

The incremental mode is started by using the TRGR.PCIFSTRT bit. When the CTLR.PCIFMD bit = 1, setting the PCIFSTRT bit to 1 starts the incremental mode and sets the SR.PCIFPS flag to 1. When the incremental mode is started, ABZ-phase input reception starts and ABZ-phase outputs begin in accordance with the output counter (INCPCMR.INCOCNTM[15:0] bits).

If AB(Z)-phase input is toggled, the output counter changes, and the corresponding A-, B-, and Z-phase signals are output.

For details on the start and stop operation flows, refer to section 59.3.4.4, Flows of Incremental Mode. Figure 59.25 and Figure 59.26 show the waveforms when starting and stopping the incremental mode, respectively.

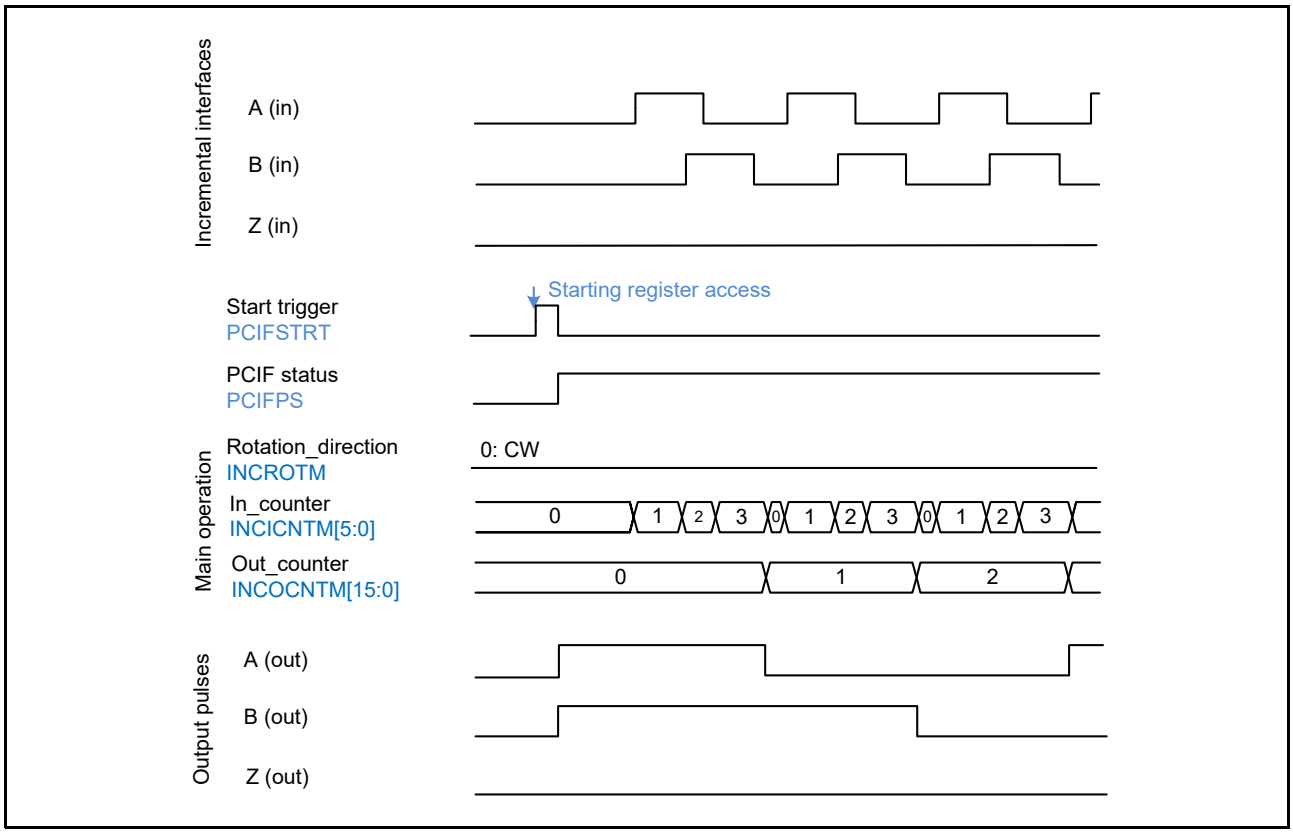


Figure 59.25 Waveforms when Starting Incremental Mode

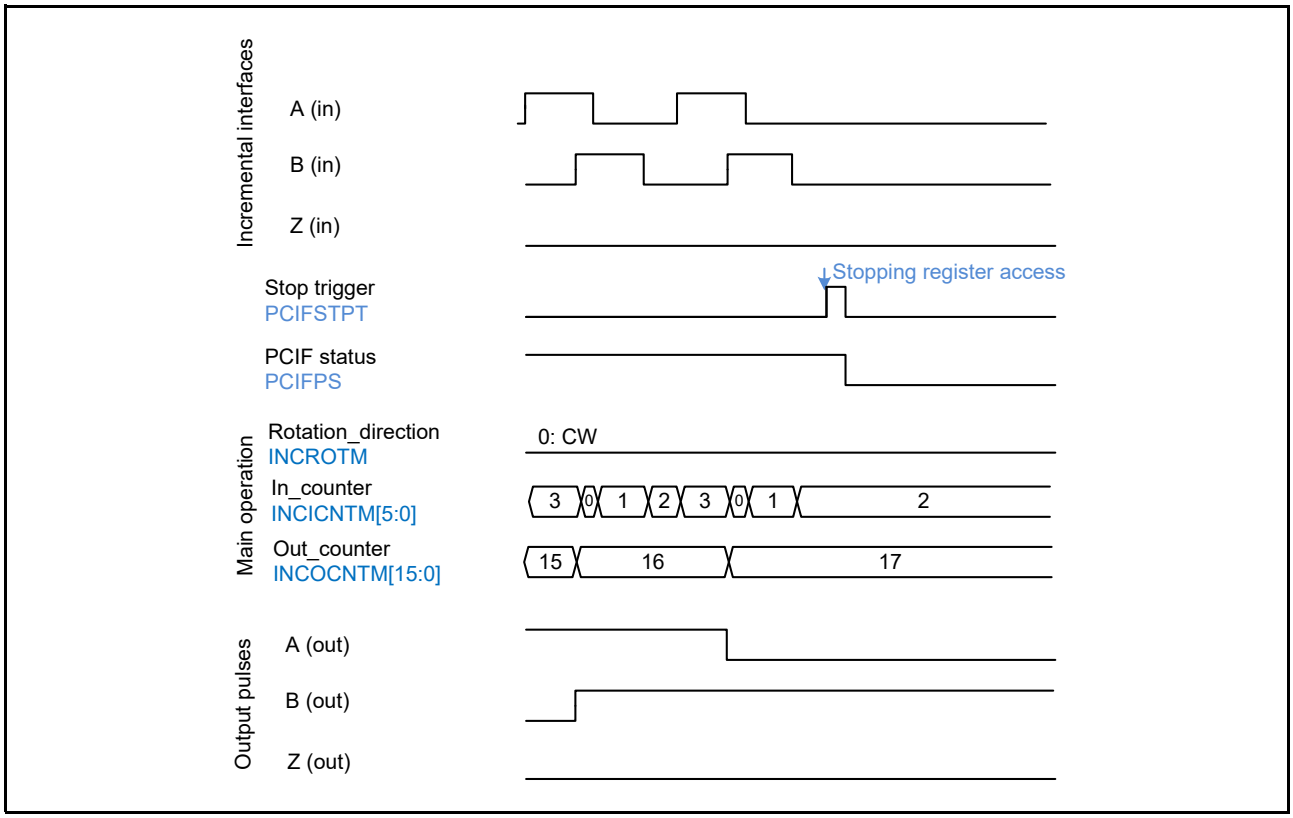


Figure 59.26 Waveforms when Stopping Incremental Mode

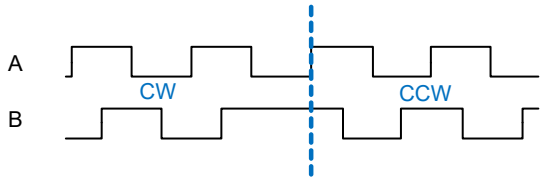
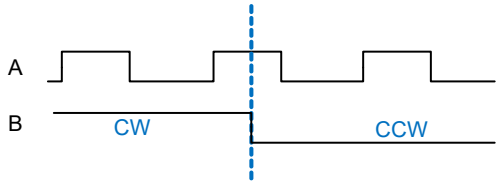
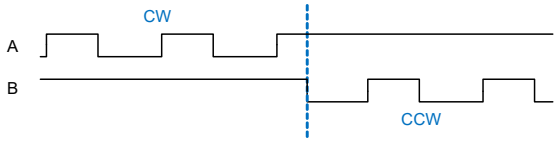
59.3.2.2 Generating A- and B-Phase Signals

This section describes generating AB-phase outputs.
First, the number of edges in AB-phase inputs is counted by the input counter (INCPCMR.INCICNTM[5:0] bits). Then, based on the input counter, the output counter (INCPCMR.INCOCNTM[15:0] bits) operates for each edge count appropriate for the resolution ratio. Finally, A- and B-phase signals corresponding to the output counter are output.

(1) I/O Protocol Change Function

In incremental mode, three types of AB-phase input protocols can be accepted. Furthermore, A- or B-phase signals converted to a different protocol from the input protocol can be output. The INCCTLR.INCIMD[1:0] bits are used to control input protocols, and the INCOMD bit is used to control output protocols.
Table 59.8 shows the five available combinations of I/O protocols.
When the incremental mode is started, the rotation direction of each protocol is determined by toggling input signals, and the output changes according to the protocol mode.

Table 59.8 Correspondence between Register Settings and I/O Protocols

INCIMD[1:0] Bits	INCOMD Bit	Input Protocol Mode	Output Protocol Mode	Waveforms
00b	x	Mode 0: AB-phase pulses 	Mode 0: AB-phase pulses	Figure 59.24
01b	0	Mode 1: A-phase pulse (B-phase rotation direction) 	Mode 1: A-phase pulse (B-phase rotation direction)	Figure 59.27
	1		Mode 0: AB-phase pulses	Figure 59.28
10b	0	Mode 2: Pulse switching 	Mode 2: Pulse switching	Figure 59.29
	1		Mode 0: AB-phase pulses	Figure 59.30

x: Don't care

In mode 0, both A-phase and B-phase are toggled and the rotation direction is determined by the phase relation. After edge counting ends, the combination of A- and B-phases is in one-to-one correspondence with the two lower-order bits of the output counter.
In mode 1, toggling A-phase changes the edge, and the B-phase value determines the rotation direction. A-phase corresponds to the one lower-order bit of the output counter.
In mode 2, the edge changes in the forward rotation direction when A-phase is toggled and in the reverse rotation direction when B-phase is toggled. The toggled phase corresponds to the one lower-order bit of the output counter.

The basic operation of each function is indicated with the INCCTLR.INCIMD[1:0] bits = 00b (mode 0: AB-phase pulses). Each function of the incremental mode supports all I/O protocols. Figure 59.24 shows the operation example

when the INCIMD[1:0] bits = 00b. Figure 59.27 to Figure 59.30 show operation examples in other cases.

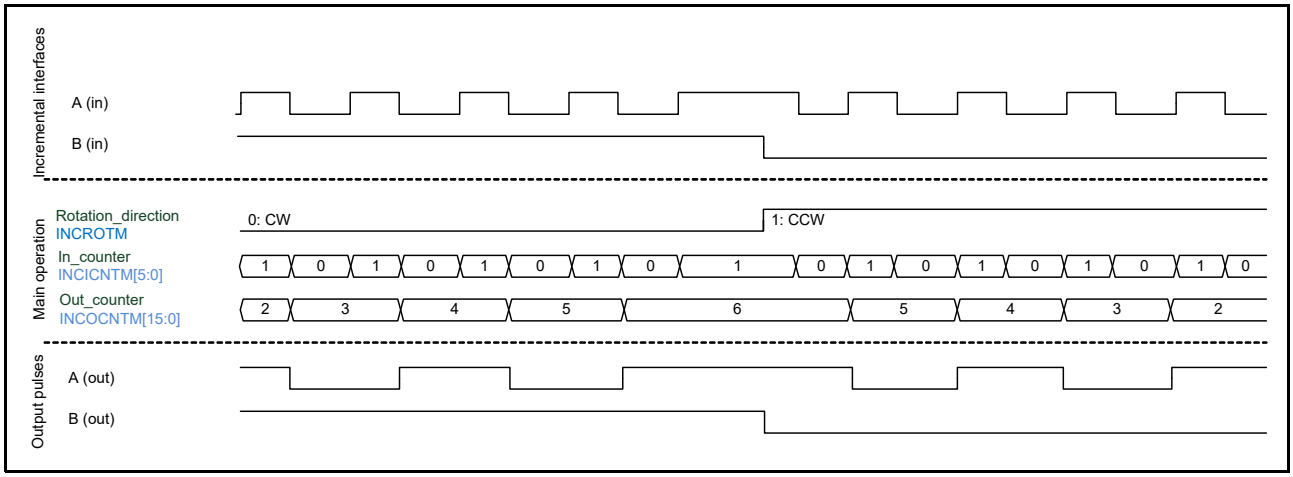


Figure 59.27 Incremental Mode Operation Waveforms 1 (INCIMD[1:0] Bits = 01b, INCOMD Bit = 0)

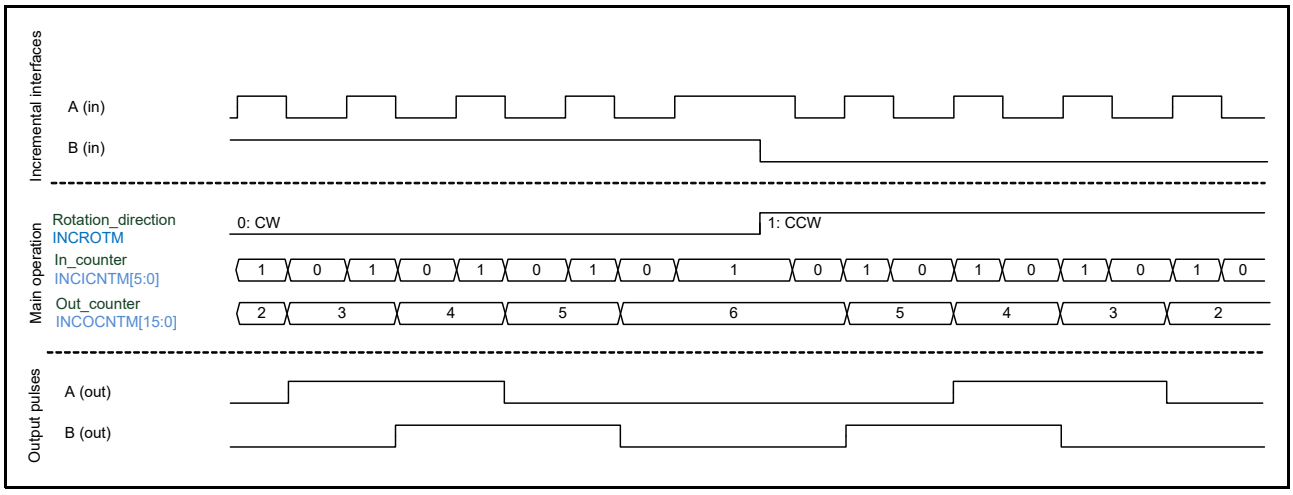


Figure 59.28 Incremental Mode Operation Waveforms 2 (INCIMD[1:0] Bits = 01b, INCOMD Bit = 1)

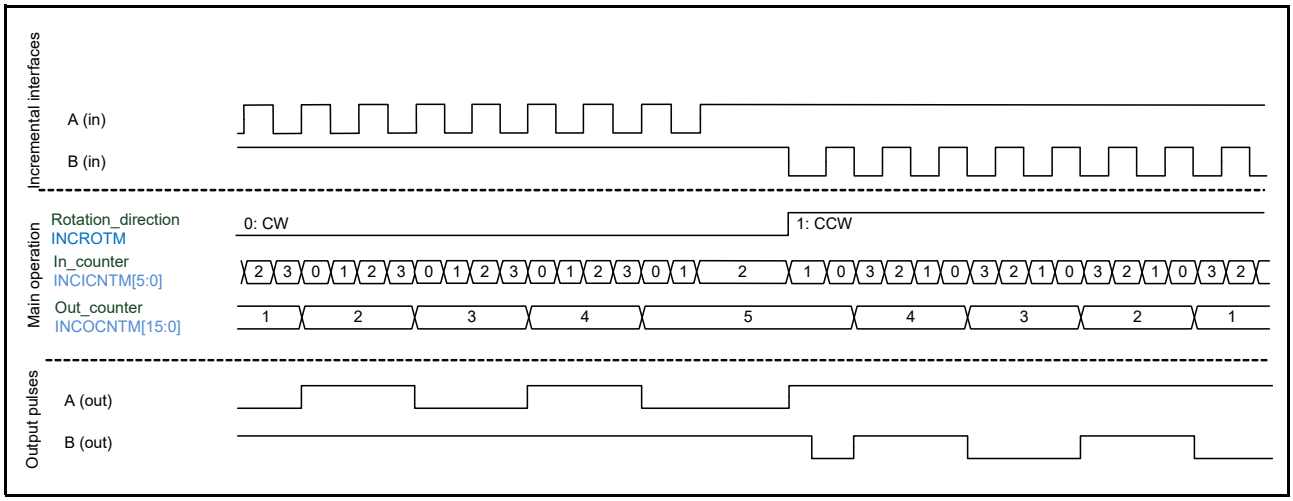


Figure 59.29 Incremental Mode Operation Waveforms 3 (INCIMD[1:0] Bits = 10b, INCOMD Bit = 0)

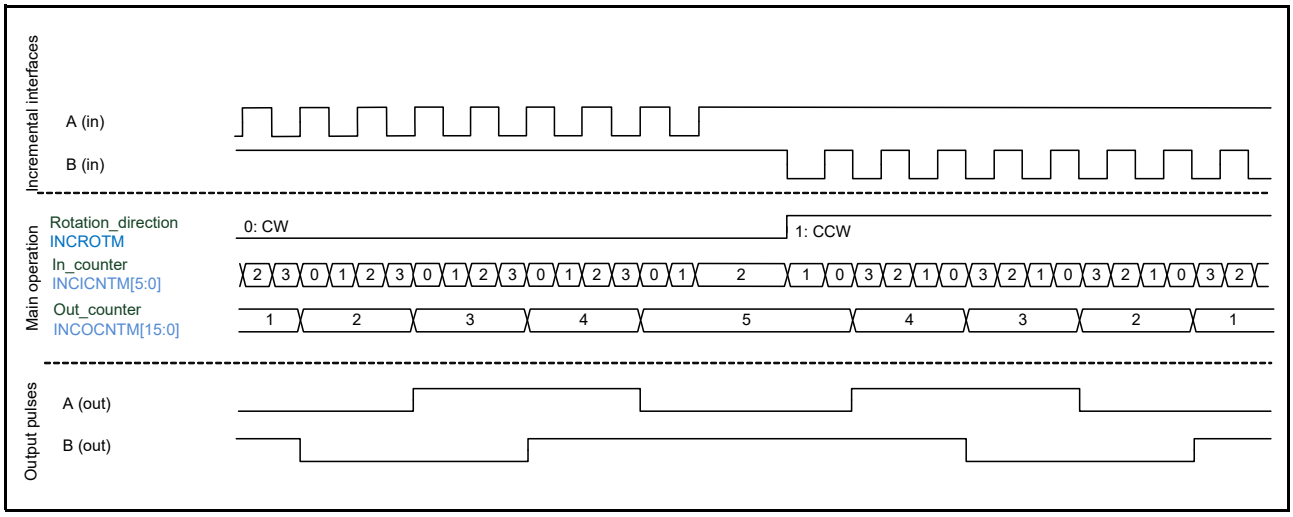


Figure 59.30 Incremental Mode Operation Waveforms 4 (INCIMD[1:0] Bits = 10b, INCOMD Bit = 1)

(2) Internal Counter Operation

In incremental mode, two internal counters, input counter (INCICNT[5:0]) and output counter (INCOCNT[15:0]), are used to generate AB-phase outputs through the following steps 1 to 3.

1. Conversion from AB-phase input to the input counter (INCICNT[5:0]) (AB-phase reception)
2. Conversion from the input counter to the output counter (resolution change)
3. AB-phase output from the output counter (INCOCNT[15:0]) (AB-phase generation)

The following describes details of each processing.

(a) AB-Phase Reception

In AB-phase reception, edges in the AB-phase inputs are detected and the number of edges in the AB-phase inputs is counted by the input counter (INCICNT[5:0]).
Table 59.9, Table 59.10, and Table 59.11 show the methods of counting by the input counter when AB-phase input is in mode 0, mode 1, and mode 2, respectively.

Operation is not guaranteed if "Unexpected protocol" in Table 59.9 to Table 59.11 is executed.

Table 59.9 Method of Counting by the Input Counter When AB-Phase Input is in Mode 0

A-Phase Input	B-Phase Input	Input Counter Operation	Rotation Direction (INCPCMR.INCROTM Bit)	Remarks
0	Falling	Up-counting	0 (forward rotation)	—
1	Rising			—
Rising	0			—
Falling	1			—
0	Rising	Down-counting	1 (reverse rotation)	—
1	Falling			—
Falling	0			—
Rising	1			—
0	0	Retention	Retention	—
0	1			—
1	0			—
1	1			—
Rising	Rising			Unexpected protocol
Rising	Falling			Unexpected protocol
Falling	Rising			Unexpected protocol
Falling	Falling			Unexpected protocol

Table 59.10 Method of Counting by the Input Counter When AB-Phase Input is in Mode 1

A-Phase Input	B-Phase Input	Input Counter Operation	Rotation Direction (INCPCMR.INCROTM Bit)	Remarks
Rising	1	Up-counting	0 (forward rotation)	—
Falling	1			—
Rising	0	Down-counting	1 (reverse rotation)	—
Falling	0			—
0	1	Retention	0 (forward rotation)	—
0	Rising			—
1	1			—
1	Rising			—
0	0		1 (reverse rotation)	—
0	Falling			—
1	0			—
1	Falling			—
Rising	Rising		0 (forward rotation)	—
Falling	Rising			—
Rising	Falling		1 (reverse rotation)	—
Falling	Falling			—

Table 59.11 Method of Counting by the Input Counter When AB-Phase Input is in Mode 2

A-Phase Input	B-Phase Input	Input Counter Operation	Rotation Direction (INCPCMR.INCROTM Bit)	Remarks
Rising	1	Up-counting	0 (forward rotation)	—
Falling	1			—
1	Rising	Down-counting	1 (reverse rotation)	—
1	Falling			—
0	0	Retention	Retention	Unexpected protocol except for the initial state
0	1			—
1	0			—
1	1			—
0	Rising			Unexpected protocol
0	Falling			Unexpected protocol
Rising	0			Unexpected protocol
Falling	0			Unexpected protocol
Rising	Rising			Unexpected protocol
Rising	Falling			Unexpected protocol
Falling	Rising			Unexpected protocol
Falling	Falling			Unexpected protocol

(b) Resolution Changing Function

The resolution changing function outputs AB-phase signals by reducing the resolution of AB-phase input to allow host controllers to receive data. The resolution can be changed in the range from 1/1 (same size) to 1/64.
The resolution changing function is implemented with the input counter (INCICNT[5:0]) and output counter (INCOCNT[15:0]).
Figure 59.31 shows the operation waveforms when changing the resolution to 1/4.

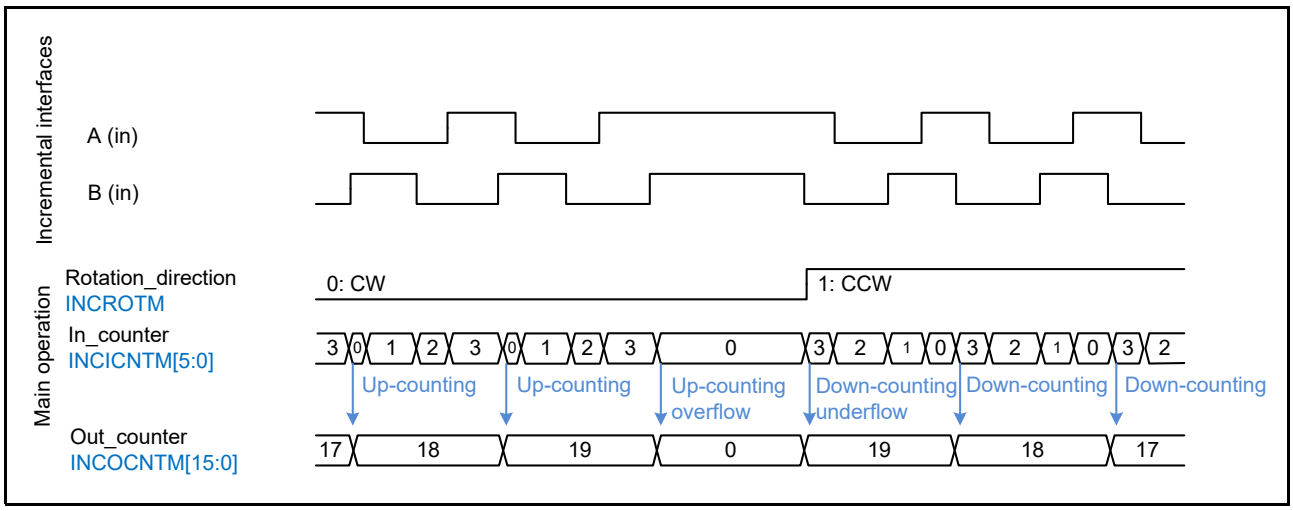


Figure 59.31 Waveforms of Resolution Changing Function (INCCTLR.INCRSL[5:0] Bits = 03h, POSMAXR.POSMAX[15:0] Bits = 0013h)

The input counter (INCICNT[5:0]) has a prescaler function. If the input counter overflows, the output counter performs up-counting. If the input counter underflows, the output counter performs down-counting. The maximum value of the input counter is the value set in the INCCTLR.INCRSL[5:0] bits. If the output counter (INCOCNT[15:0]) overflows, it is cleared to zero. If the output counter underflows, the maximum value is set.
The maximum value of the output counter is shown in Table 59.12. Figure 59.32 shows waveforms when the output counter exceeds the value of the POSMAXR.POSMAX[15:0] bits, causing the INCOCNTM[15:0] bits to be cleared.

Table 59.12 Maximum Value of the Output Counter (INCOCNT[15:0])

INCZGCTLR.INCZAGE Bit	Maximum Value of the Output Counter (INCOCNT[15:0])
0	FFFFh
1	POSMAX[15:0] setting

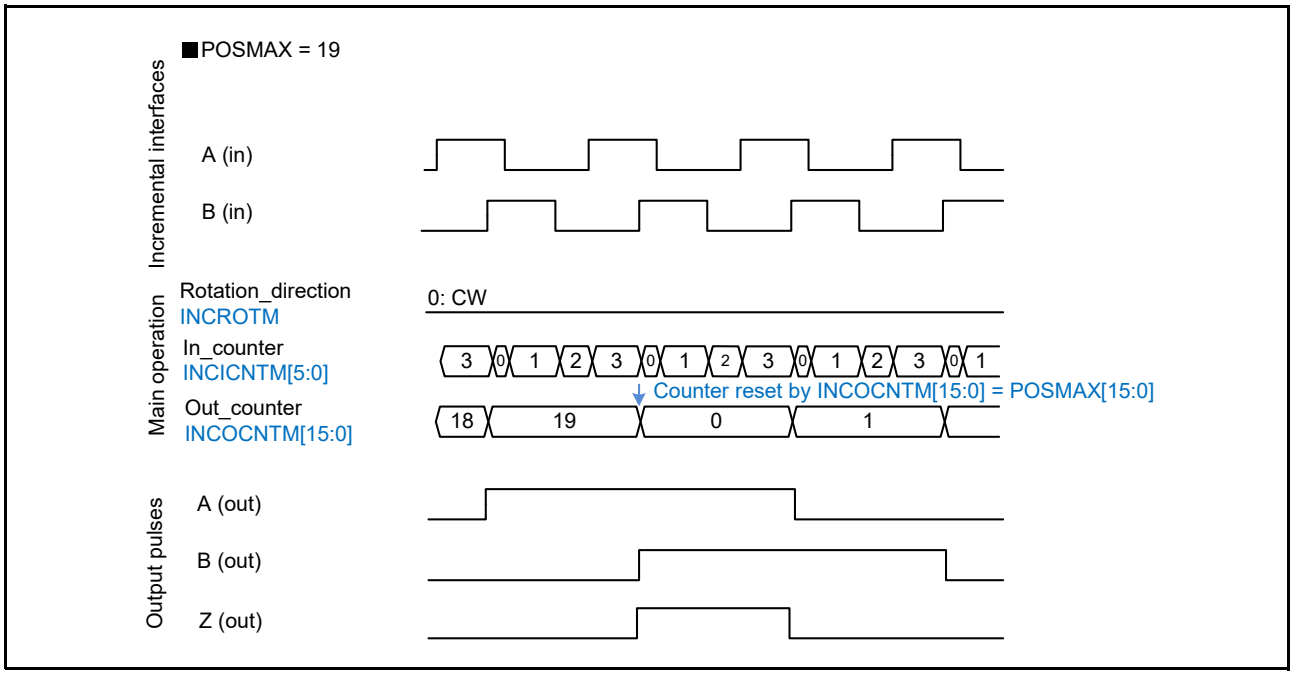


Figure 59.32 Waveforms When the Output Counter (INCOCNT[15:0]) is Cleared by the POSMAX[15:0] Setting

(c) AB-Phase Generation

The following describes how AB-phase outputs are generated based on the position information indicated by the output counter (INCOCNT[15:0]).

The AB-phase output generation method when the INCCTLR.INCIMD[1:0] bits = 00b or the INCCTLR.INCOMD bit = 1 (mode 0 output) is shown in Table 59.13. The AB-phase output generation method when the INCIMD[1:0] bits = 01b and the INCOMD bit = 0 (mode 1 output) is shown in Table 59.14. The AB-phase output generation method when the INCIMD[1:0] bits = 10b and the INCOMD bit = 0 (mode 2 output) is shown in Table 59.15.

Table 59.13 Truth Table for Generating AB-Phase Outputs by Using the Output Counter (Mode 0)

INCOCNT[1:0]	00b	01b	10b	11b
A-phase output	1	0	0	1
B-phase output	1	1	0	0

Table 59.14 Truth Table for Generating AB-Phase Outputs by Using the Output Counter (Mode 1)

INCOCNT[0]	0	1	0	1
INCROTM Bit	0 (Forward Rotation)	0 (Forward Rotation)	1 (Reverse Rotation)	1 (Reverse Rotation)
A-phase output	1	0	1	0
B-phase output	1	1	0	0

Table 59.15 Truth Table for Generating AB-Phase Outputs by Using the Output Counter (Mode 2)

INCOCNT[0]	0	1	0	1
INCROTM Bit	0 (Forward Rotation)	0 (Forward Rotation)	1 (Reverse Rotation)	1 (Reverse Rotation)
A-phase output	1	0*1	1	1
B-phase output	1	1	1	0*1

Note 1. If the value of the INCROTM bit changes, the AB-phase outputs are always placed at the high level for one cycle of CCLK_DIV irrespective of the above setting. Example waveforms are shown in Figure 59.33, Example of Changing the Rotation Direction in Mode 2.

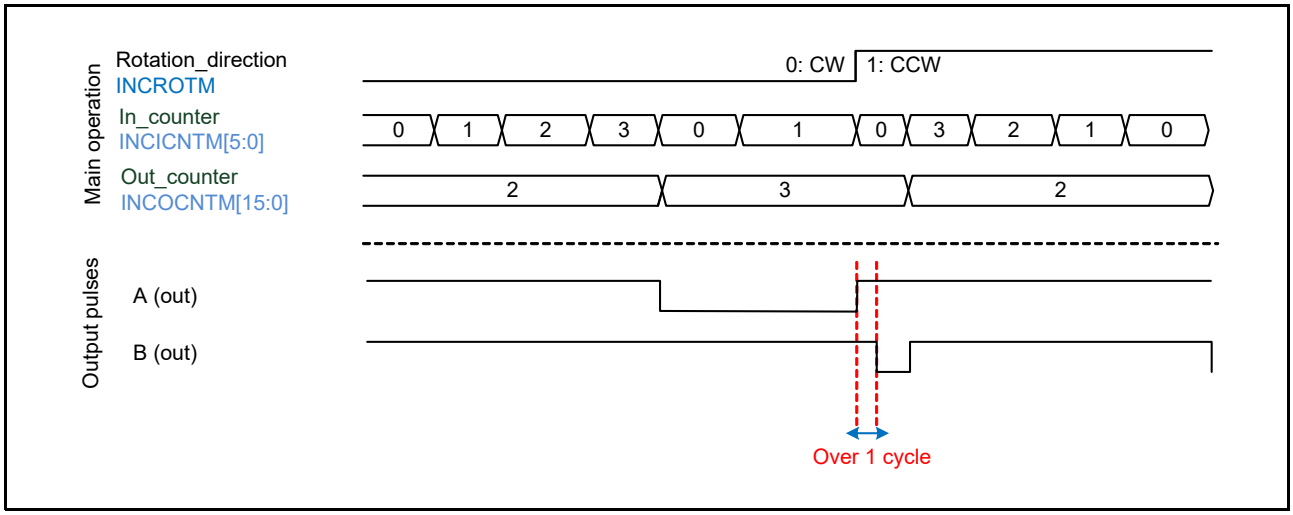


Figure 59.33 Example of Changing the Rotation Direction in Mode 2

(3) Internal Counter Clearing Function

As shown in (2), AB-phase output corresponds to the output counter (INCPCMR.INCOCNTM[1:0] bits). In addition to the clearing function for the INCOCNTM[15:0] bits, the PCIF has the following four types of internal counter clearing functions that allow AB-phase outputs to be cleared on the system side. When any of the conditions for clearing is met, the internal counters (INCOCNT[15:0] and INCICNT[5:0]) are initialized.

- Clearing by a Z-phase input*1
- Clearing by a crest event trigger (PEK_IN)
- Clearing by a trough event trigger (VLY_IN)
- Clearing by a software trigger

You can control any combination of these functions by setting the INCPCCER register.*2

Note 1. When the internal counters are cleared by a Z-phase input, note that the counter value might be incremented by 1 due to the deviation in the timing of input between the AB-phase and Z-phase signals.
Note 2. When the internal counters are cleared by setting the POSMAXR.POSMAX[15:0] bits (INCZGCTLR.INCZAGE bit = 1), clearing the counters by a Z-phase input is prohibited.

For details on the waveforms for internal counter clearing by a Z-phase input, refer to Figure 59.24. Figure 59.34 shows the waveforms for internal counter clearing by a software trigger.

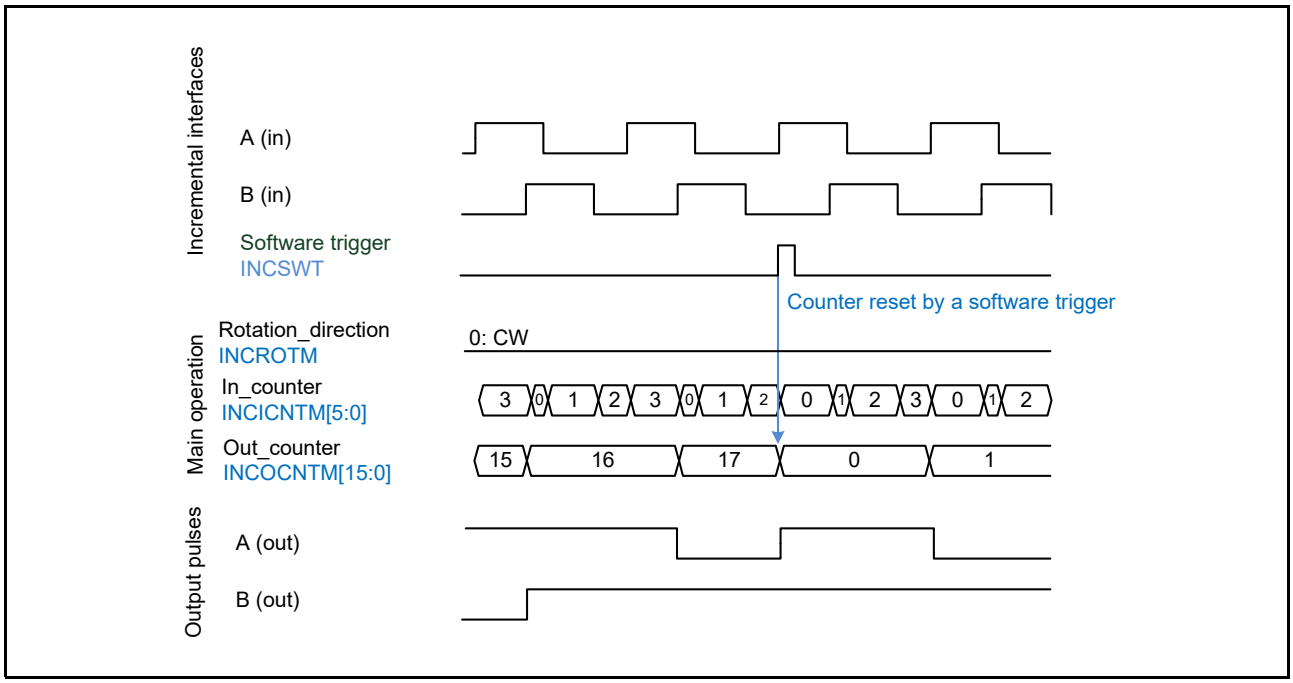


Figure 59.34 Waveforms for Internal Counter Clearing by a Software Trigger

59.3.2.3 Generating Z-Phase Signals

Z-phase output is a signal indicating that the motor rotates one round. Z-phase output is generated by using the output counter, a Z-phase input, crest/trough event trigger, or software trigger. In addition, you can select whether to output Z-phase signals synchronized with AB-phase output or output immediately.

(1) Selecting the Z-Phase Output Trigger

Z-phase signals can be output by using one of the following five ways:

- Generation with the output counter ...(a)
- Generation by a Z-phase input ...(b)
- Generation by a crest event trigger ...(c)
- Generation by a trough event trigger ...(c)
- Generation by a software trigger ...(c)

You can control any combination of these methods by setting the INCZGCTLR register.

(a) Generation with the Output Counter

Z-phase output can be generated by using the output counter (INCPCMR.INCOCNTM[15:0] bits) for AB-phase operations.
If the INCZGCTLR.INCZAGE bit is set to 1, Z-phase output is asserted when the INCOCNTM[15:0] bits = 0000h. The maximum value of the counter (INCOCNTM[15:0] bits) is set in the POSMAXR.POSMAX[15:0] bits. If an overflow or underflow occurs, the count wraps around within the range from 0 to the POSMAX[15:0] bit setting.
Figure 59.35 shows example waveforms for Z-phase output generation with the output counter.

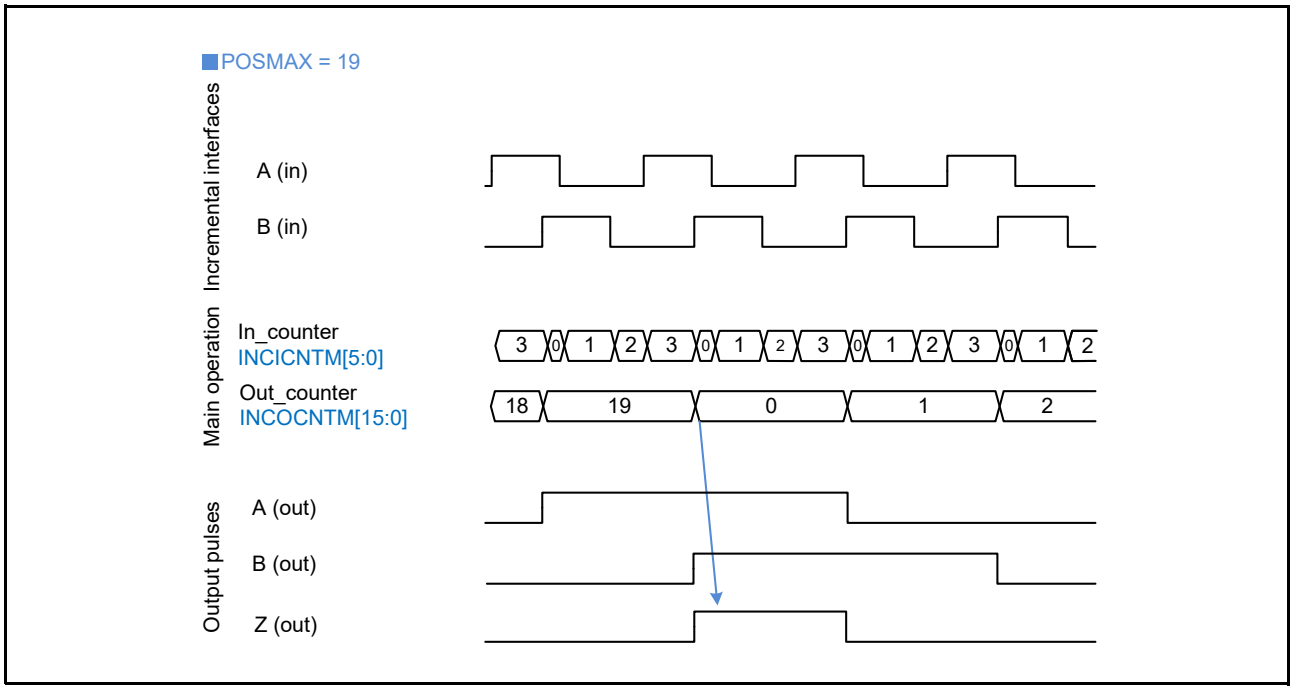


Figure 59.35 Example of Z-Phase Output Generation with the Output Counter (INCOCNTM[15:0] Bits) (INCZGCTLR.INCZAGE Bit = 1, INCCTLR.INCZOS[1:0] Bits = 00b)

(b) Generation by a Z-Phase Input

Z-phase output can be generated in synchronization with Z-phase input. If the INCZGCTLR.INCZIZGE bit is set to 1, when the Z-phase input is asserted, Z-phase output is also asserted.
Refer to Figure 59.24 for an example of Z-phase output generation in response to a Z-phase input when the INCZGCTLR.INCZIZGE bit = 1 and the INCCTLR.INCZOS[1:0] bits = 00b.

(c) Generation by Crest/Trough Event Triggers and Software Triggers

Besides the previously mentioned generation methods using the output counter and a Z-phase input, Z-phase output can also be generated by using triggers. When the INCZGCTLR.INCPIZGE bit, INCVIZGE bit, and INCSWZGE bit are set to 1, Z-phase output is asserted when a crest event trigger (PEK_IN), a trough event trigger (VLY_IN), or a software trigger is executed, respectively.
Figure 59.36 shows an example of Z-phase output generation by a software trigger (INCZGCTLR.INCSWZGE bit = 1, INCCTLR.INCZOS[1:0] bits = 00b).

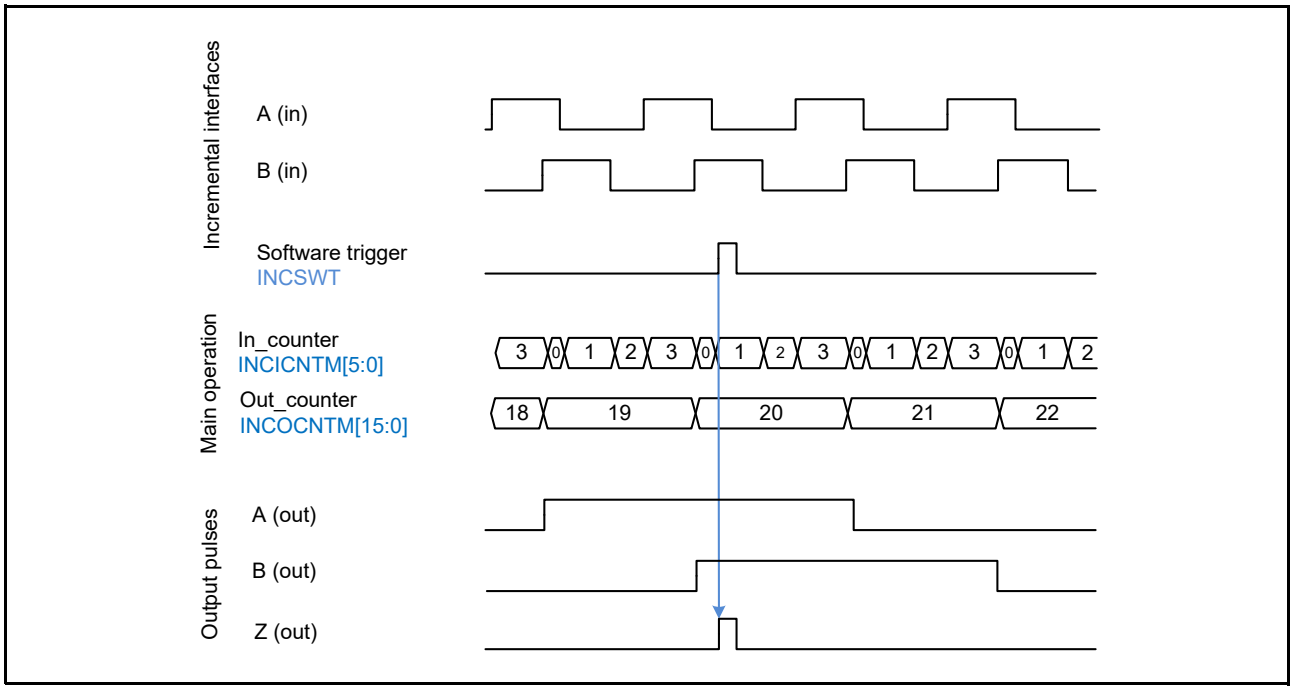


Figure 59.36 Example of Z-Phase Output Generation by a Software Trigger (INCZGCTLR.INCSWZGE Bit = 1, INCCTLR.INCZOS[1:0] Bits = 00b)

(2) Selecting the Z-Phase Output Synchronization Method

For the PCIF, you can select whether to generate Z-phase output when any of the Z-phase output conditions is met as described in (1) Selecting the Z-Phase Output Trigger, or to synchronize the Z-phase output generation to AB-phase output.

Table 59.16 shows the types of synchronization methods. Figure 59.37 shows the Z-phase output timing in mode 0, Figure 59.38 shows the Z-phase output timing in mode 1, and Figure 59.39 shows the Z-phase output timing in mode 2.

Table 59.16 Types of Z-Phase Output Synchronization Methods

Z-Phase Sync Condition	INCCTLR. INCZOS[1:0] Bits	Assertion Timing	De-assertion Timing	Remarks
Immediate output	00b	When a Z-phase generation condition is met.	When no Z-phase generation conditions are met.	The Z-phase generation conditions are as follows. • When the INCZGCTLR.INCZAGE bit = 1, INCPCMR.INCOCNTM[15:0] bits = 0000h • When the INCZGCTLR.INCZIZGE bit = 1, Z-phase input = 1 • When the INCZGCTLR.INCPIZGE bit = 1, PEK_IN = 1 • When the INCZGCTLR.INCVIZGE bit = 1, VLY_IN = 1 • When the INCZGCTLR.INCSWZGE bit = 1, 1 is written to the INCSWTRGR.INCSWT bit.
Synchronization with the A-phase	01b	Rising edge of A-phase before the next inversion of the active sense when a Z-phase generation condition is met	Falling edge of A-phase before the inversion of the active sense	Available only when the output protocol is mode 0
Synchronization with the B-phase	10b	Rising edge of B-phase before the next inversion of the active sense when a Z-phase generation condition is met	Falling edge of B-phase before the inversion of the active sense	Available only when the output protocol is mode 0
Synchronization with the pulse	11b	Rising edge of the pulse output signal*1 before the next inversion of the active sense when a Z-phase generation condition is met	Falling edge of the pulse output signal*1 before the inversion of the active sense	Available only when the output protocol is mode 1 or 2

Note 1. A-phase in mode 1, and A-phase or B-phase in mode 2.

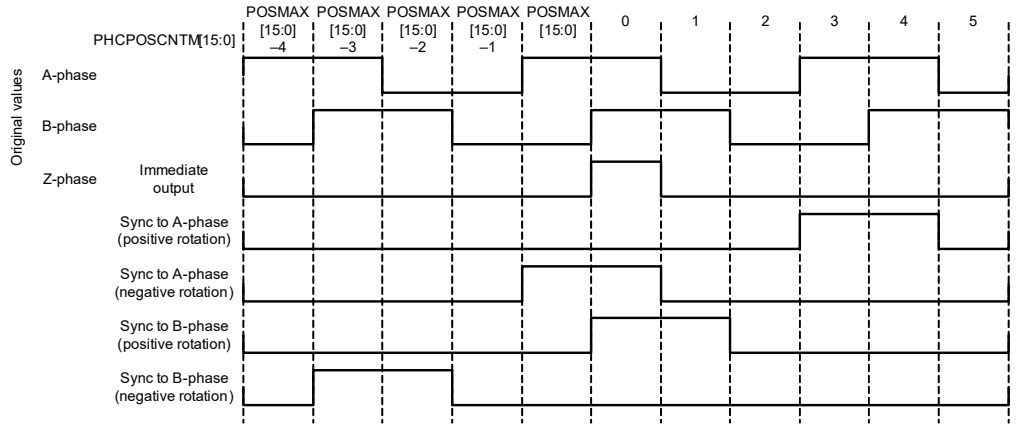
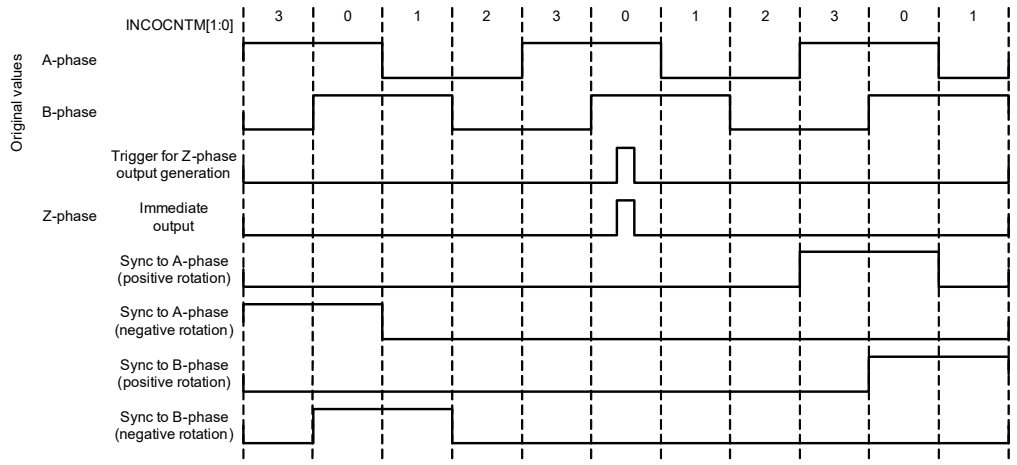
Z-phase generation with the output counter (INCZAGE = 1)	
Other than above	

Figure 59.37 Synchronized Z-Phase Output Timing (Mode 0)

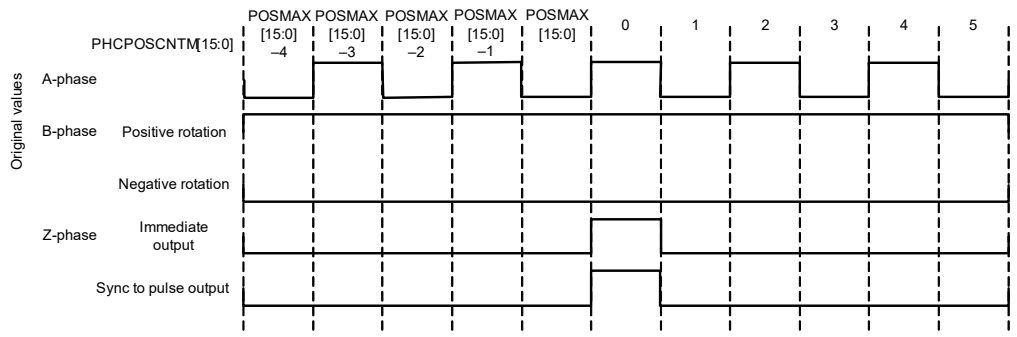
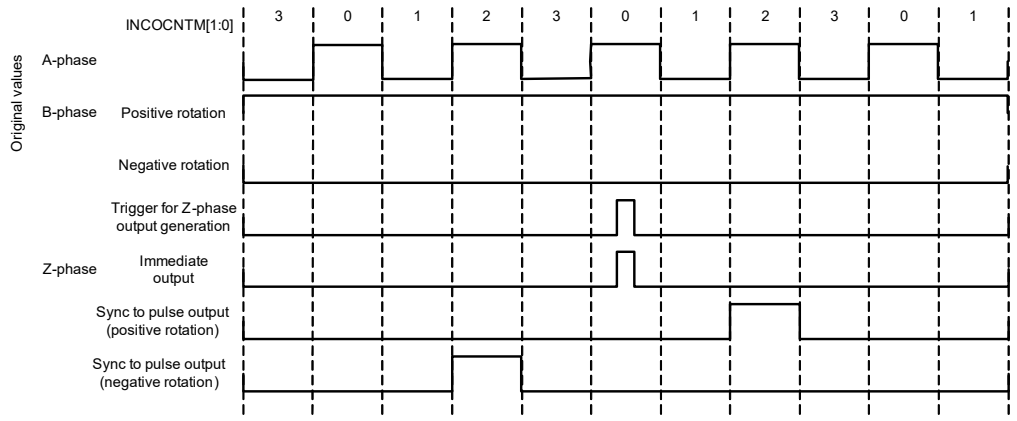
Z-phase generation with the output counter (INCZAGE = 1)	
Other than above	

Figure 59.38 Synchronized Z-Phase Output Timing (Mode 1)

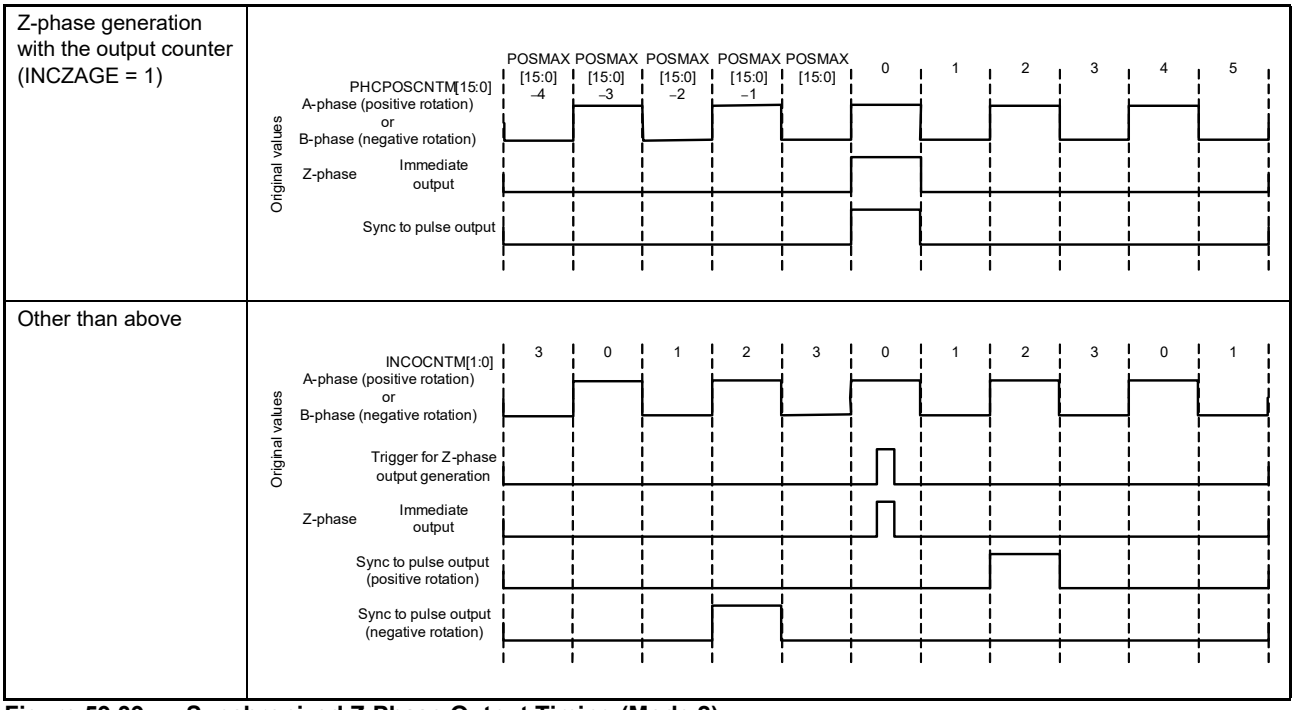


Figure 59.39 Synchronized Z-Phase Output Timing (Mode 2)

In synchronization with pulse output in mode 2, if a Z-phase output trigger is generated when the rotation direction changes, the Z-phase output might be synchronized with different phases at rising and falling edges. Figure 59.40 shows an example.

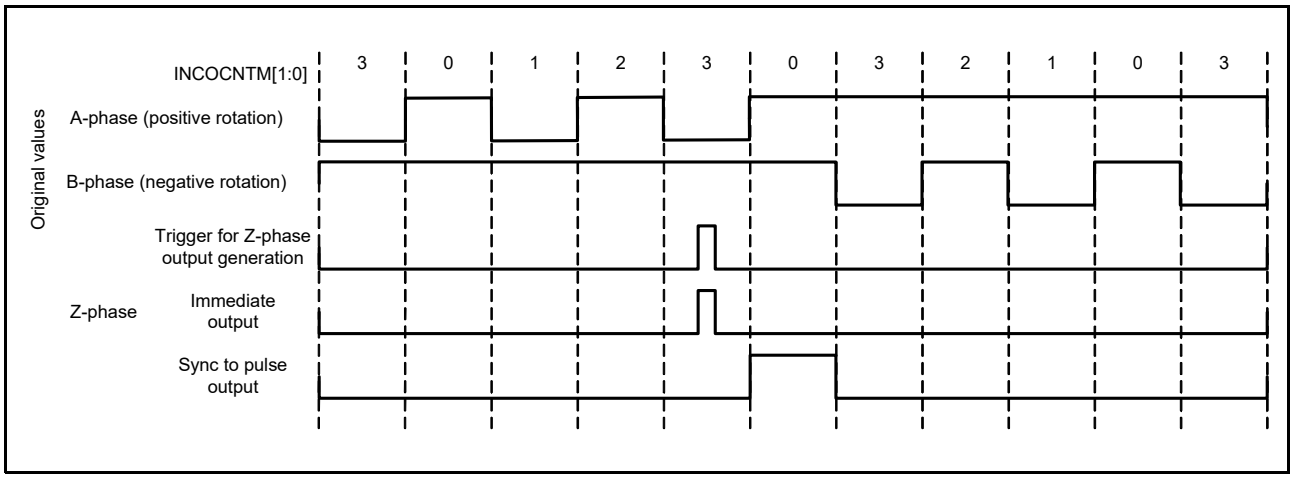


Figure 59.40 Synchronized Z-Phase Output Timing When the Rotation Direction Changes (Mode 2)

(3) Setting the Minimum Pulse Width of Z-Phase Output

The minimum assertion period of Z-phase output is one cycle of CCLK_DIV. The assertion period can be extended by up to 256 cycles of CCLK_DIV by setting the INCCTLR.INCZOMIN[7:0] bits. While the Z-phase output assertion period is being extended, any other Z-phase output conditions are ignored even if they are met. A new Z-phase output condition can be accepted after the extension is complete.

Figure 59.41 shows the Z-phase generation waveforms when setting the minimum pulse width by a software trigger.

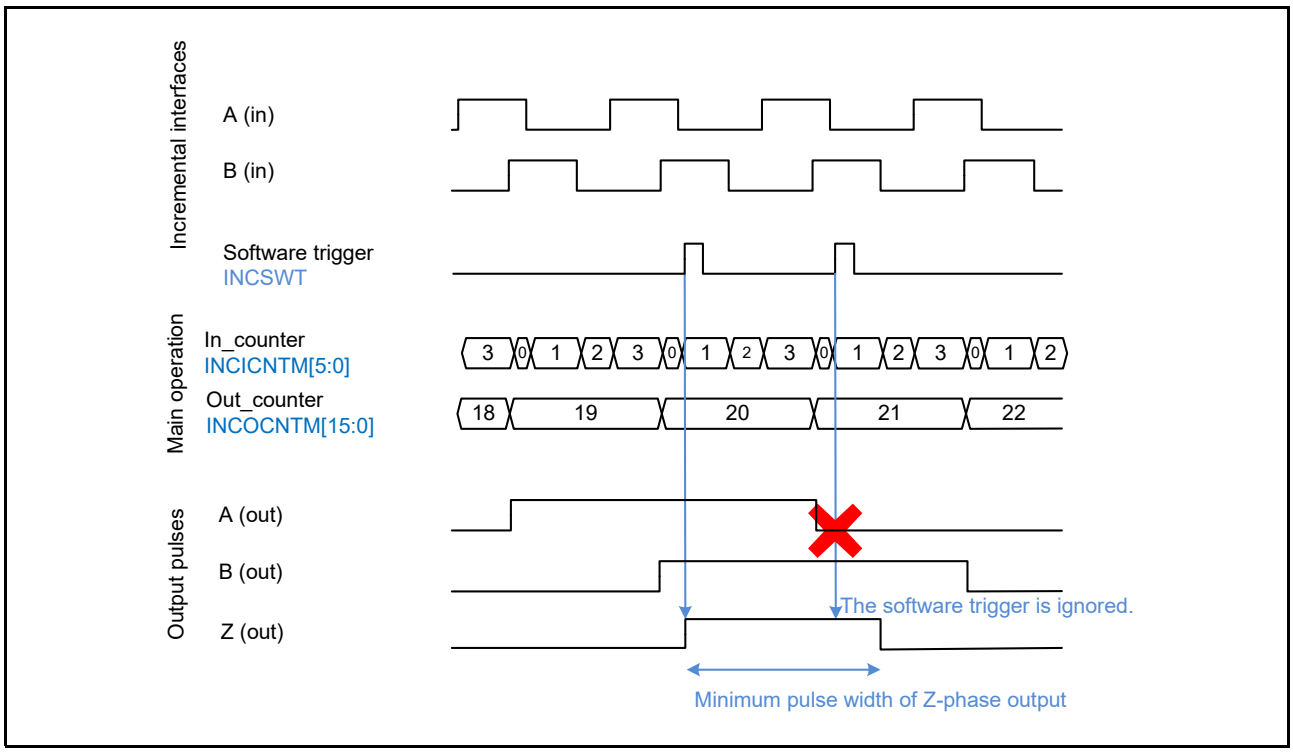


Figure 59.41 Z-Phase Generation Waveforms when Setting the Minimum Pulse Width

However, if the extended Z-phase output assertion period overlaps with a Z-phase output condition, the Z-phase signal will be output according to the longer period.

Figure 59.42 shows the waveforms for the extended Z-phase output.

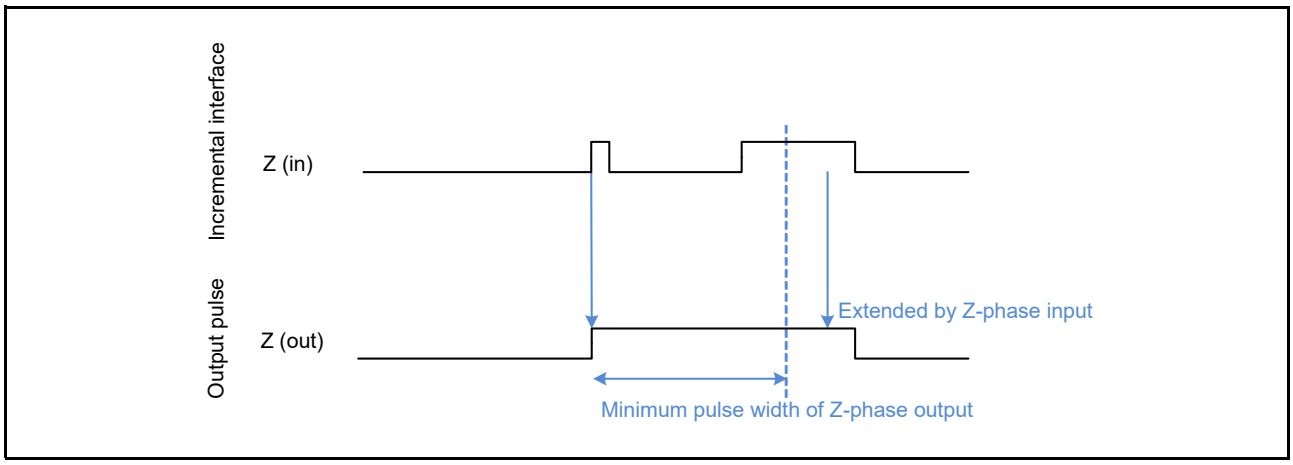


Figure 59.42 Waveforms for Extended Z-Phase Output

59.3.3 Output Controller

The following describes the function that inverts and outputs A-, B-, and Z-phase signals generated in phase counting mode or incremental mode.

59.3.3.1 ABZ-Phase Output Inversion Function

Table 59.17 shows the output values of ABZ-phase output signals. The ABZ-phase outputs can be inverted by setting the CTLR.APOL, BPOL, and ZPOL bits, respectively.
Figure 59.43 shows example waveforms when A-phase output is inverted and Z-phase output is synchronized with A-phase.

Table 59.17 Output Values of ABZ-Phase Output Signals

SWRSTR.SWRST Bit	SR.PCIFPS Flag	CTLR.PCIFMD Bit	CTLR.XPOL Bit*1	Operation	X-Phase
1	—	—	—	Software reset is in progress.	0
0	0	—	—	Stopped	The previous value is retained.
0	1	0	0	Phase counting mode	Generated value*2
0	1	0	1	Phase counting mode	Inverted value*3
0	1	1	0	Incremental mode	Generated value*4
0	1	1	1	Incremental mode	Inverted value*3

Note 1. XPOL indicates each of the APOL, BPOL, and ZPOL bits.
Note 2. ABZ-phase signals generated in phase counting mode (Refer to section 59.3.1.5.)
Note 3. ABZ-phase signals generated by inverting the values generated in the corresponding mode
Note 4. ABZ-phase signals generated in incremental mode (Refer to section 59.3.2.2 and section 59.3.2.3.)

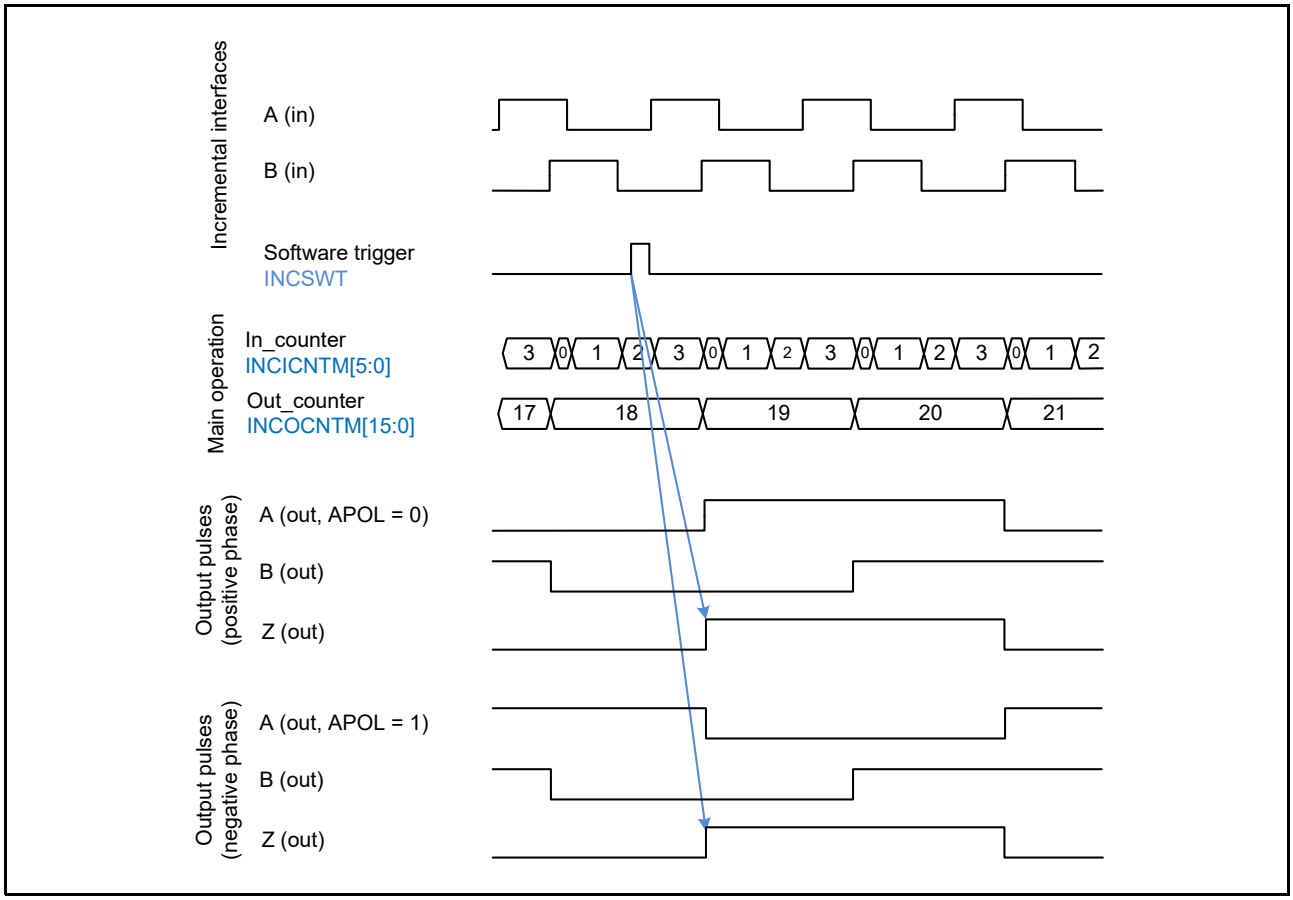


Figure 59.43 Example Waveforms for Z-Phase Output Synchronized with A-Phase (Positive/Negative Phase)

59.3.4 Operating Flows

This section describes the PCIF operating flows.

59.3.4.1 Overall Operation

Figure 59.44 shows the overall operating flow for the PCIF.

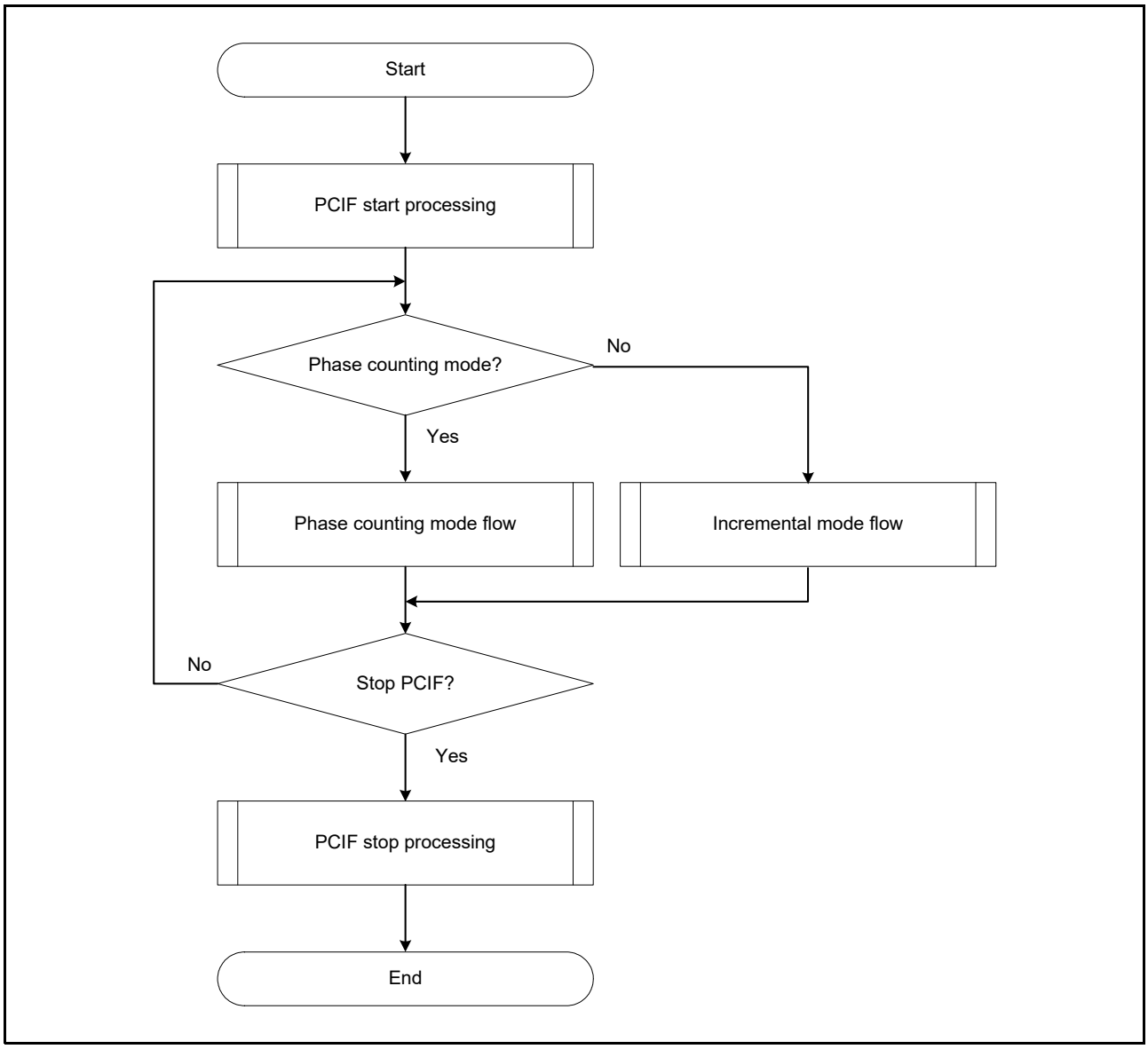


Figure 59.44 Overall Operating Flow for PCIF

59.3.4.2 PCIF Startup Processing

Figure 59.45 shows the startup processing flow.

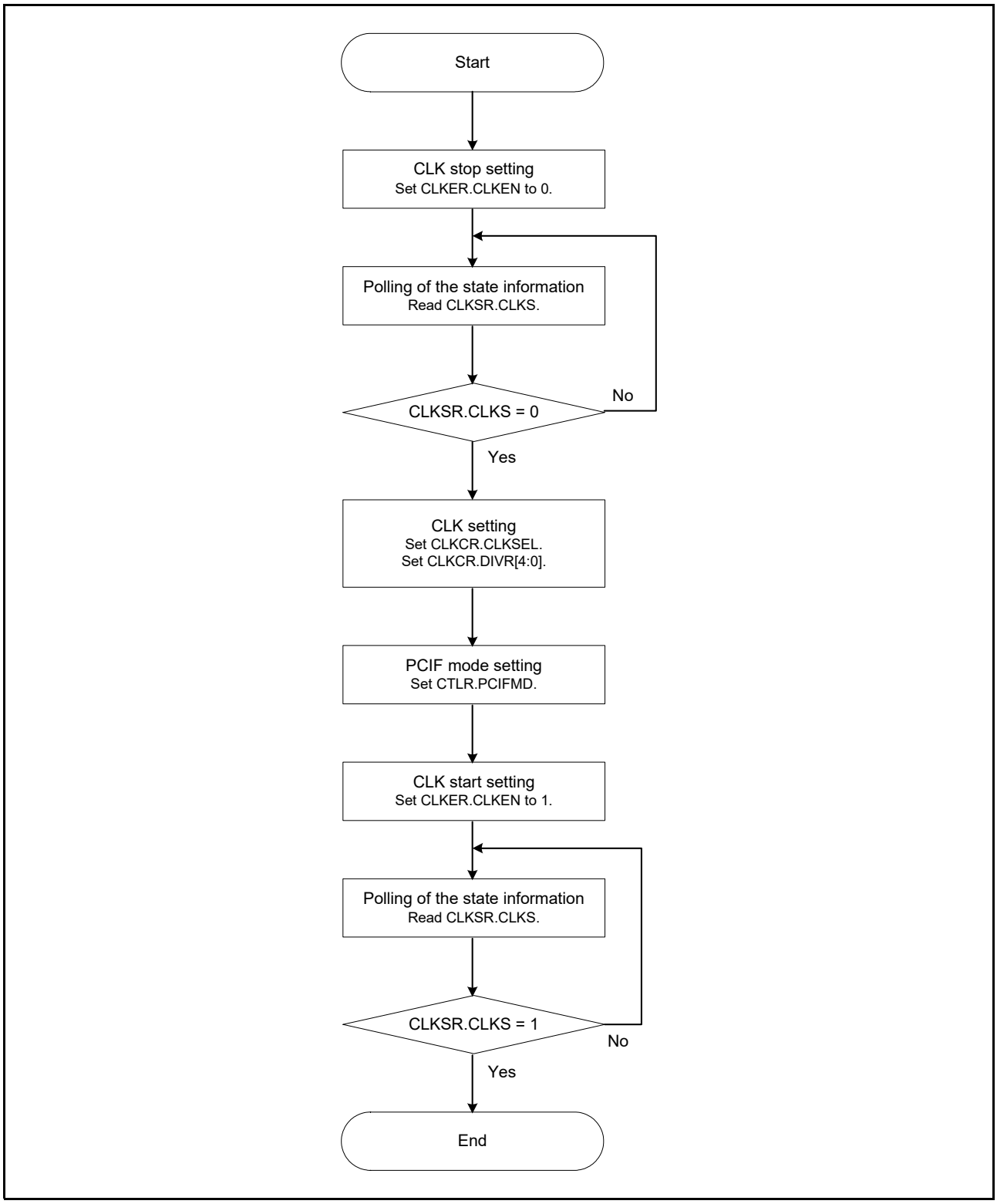


Figure 59.45 PCIF Startup Processing Flow

59.3.4.3 Flows of Phase Counting Mode

Figure 59.46 shows the processing flow of the phase counting mode.

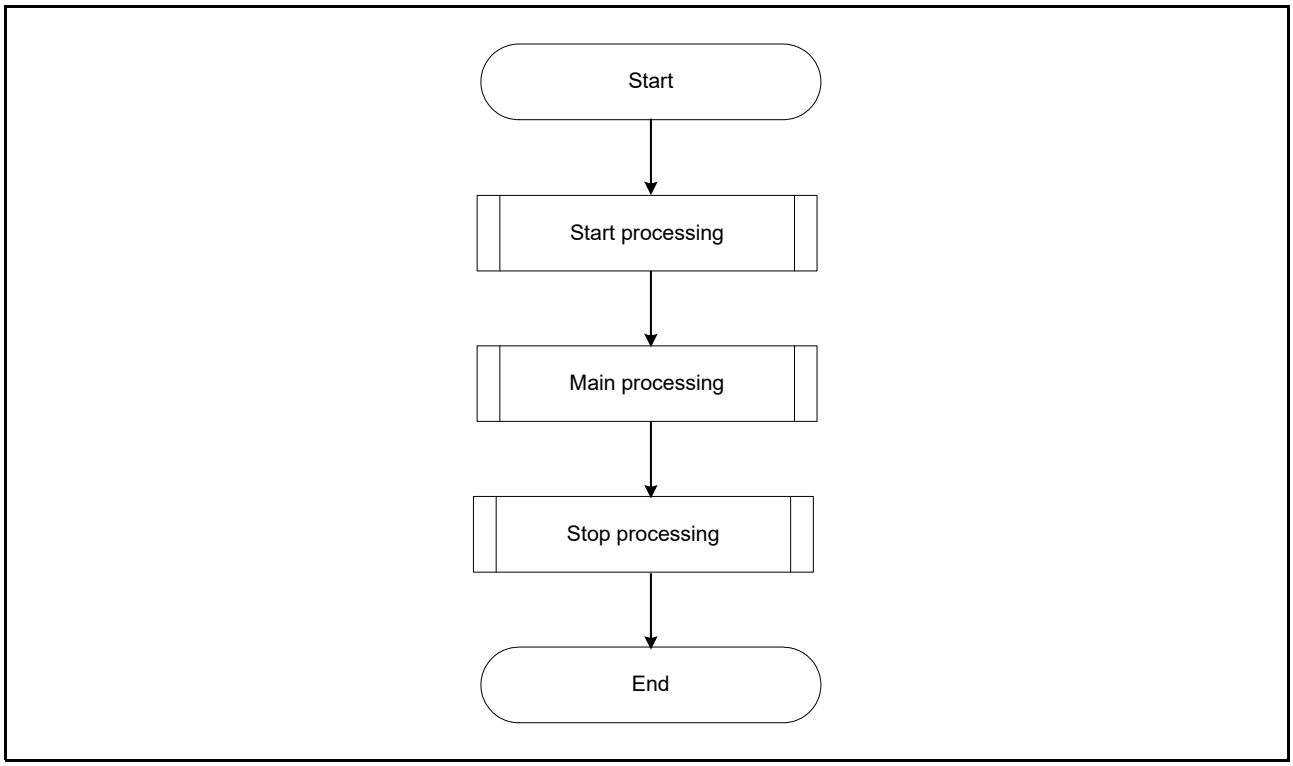


Figure 59.46 Processing Flow of Phase Counting Mode

(1) Startup Processing

Figure 59.47 shows the startup processing flow of the phase counting mode. Operation of carrier period input is not guaranteed during the startup processing flow. Start carrier period generation by the ELC after the startup processing flow ends.

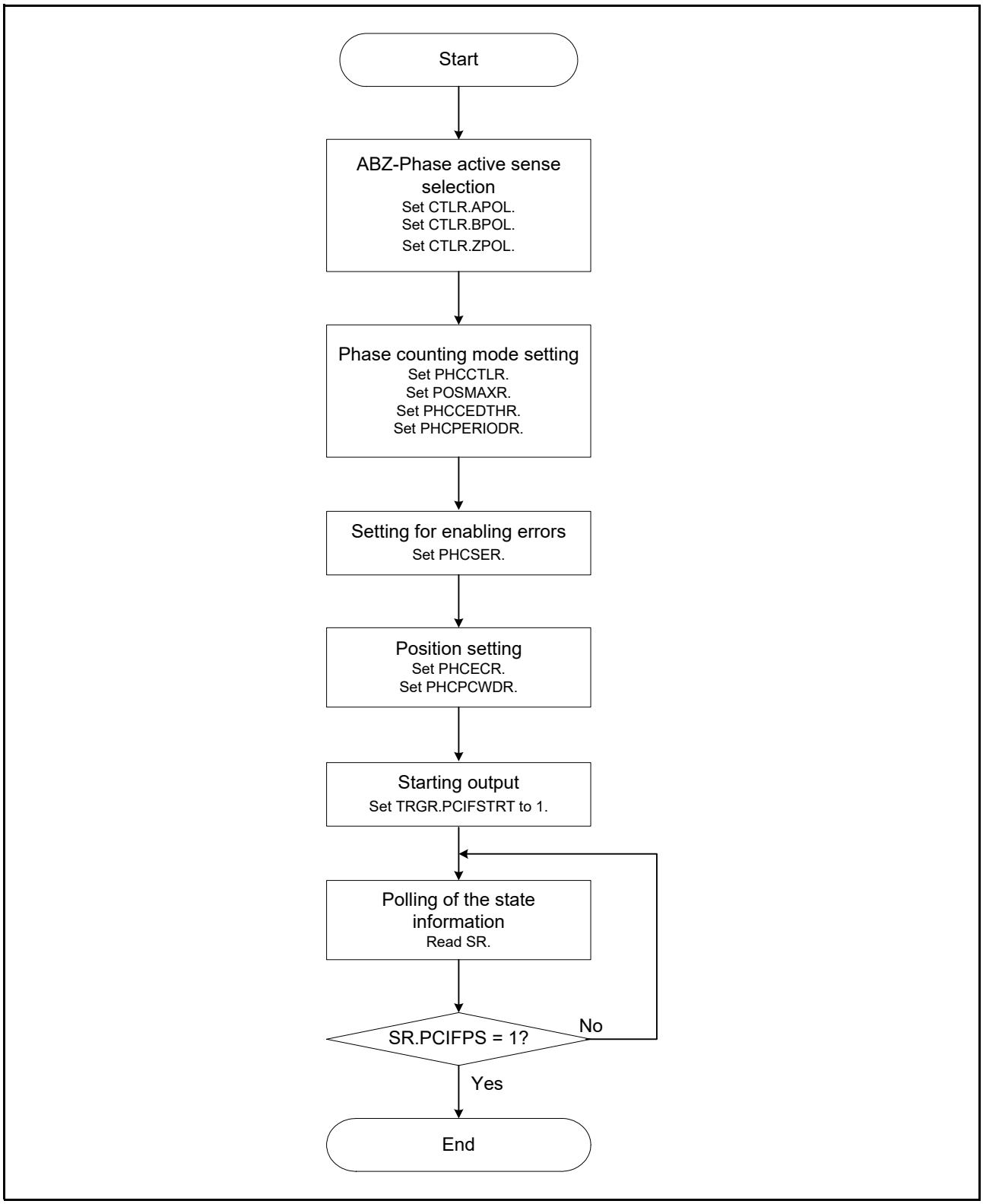


Figure 59.47 Startup Processing Flow of Phase Counting Mode

(2) Main Processing

Figure 59.48 shows the main processing flow of the phase counting mode.

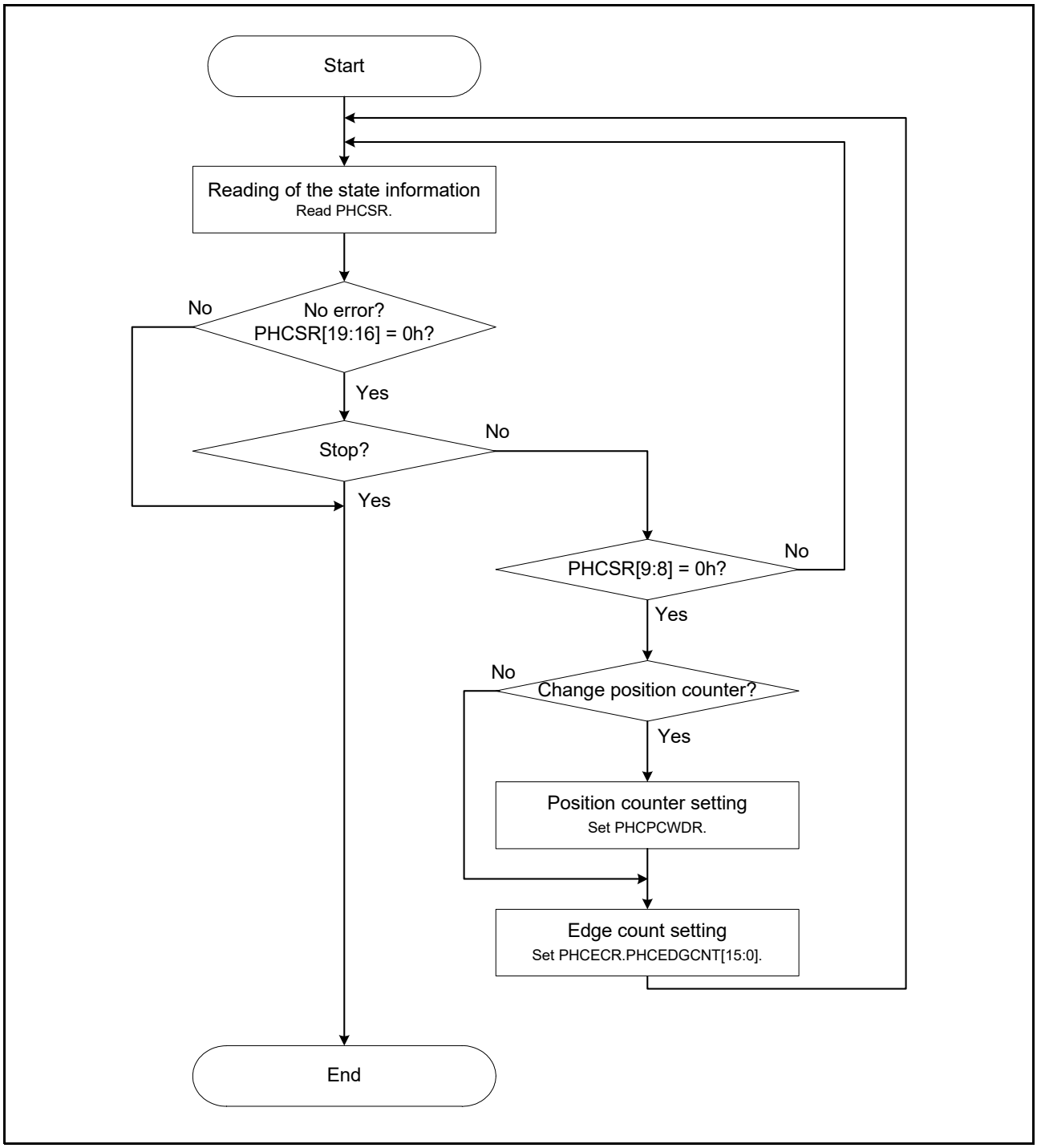


Figure 59.48 Main Processing Flow of Phase Counting Mode

(3) Stop Processing

Figure 59.49 shows the stop processing flow of the phase counting mode.
To immediately stop the phase counting mode, apply a software reset to clear the internal information such as the total edge count.

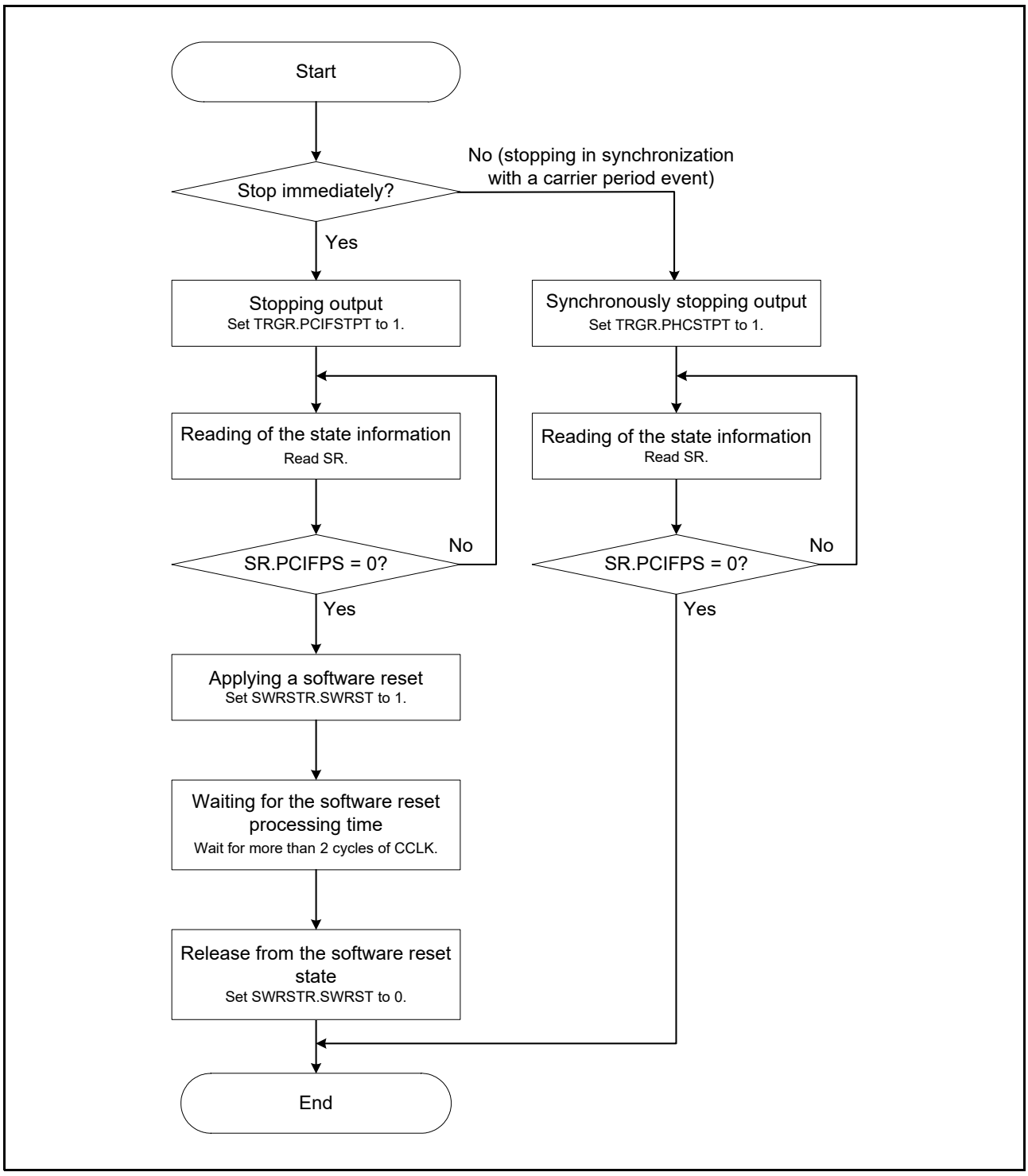


Figure 59.49 Stop Processing Flow of Phase Counting Mode

59.3.4.4 Flows of Incremental Mode

Figure 59.50 shows the processing flow of incremental mode.

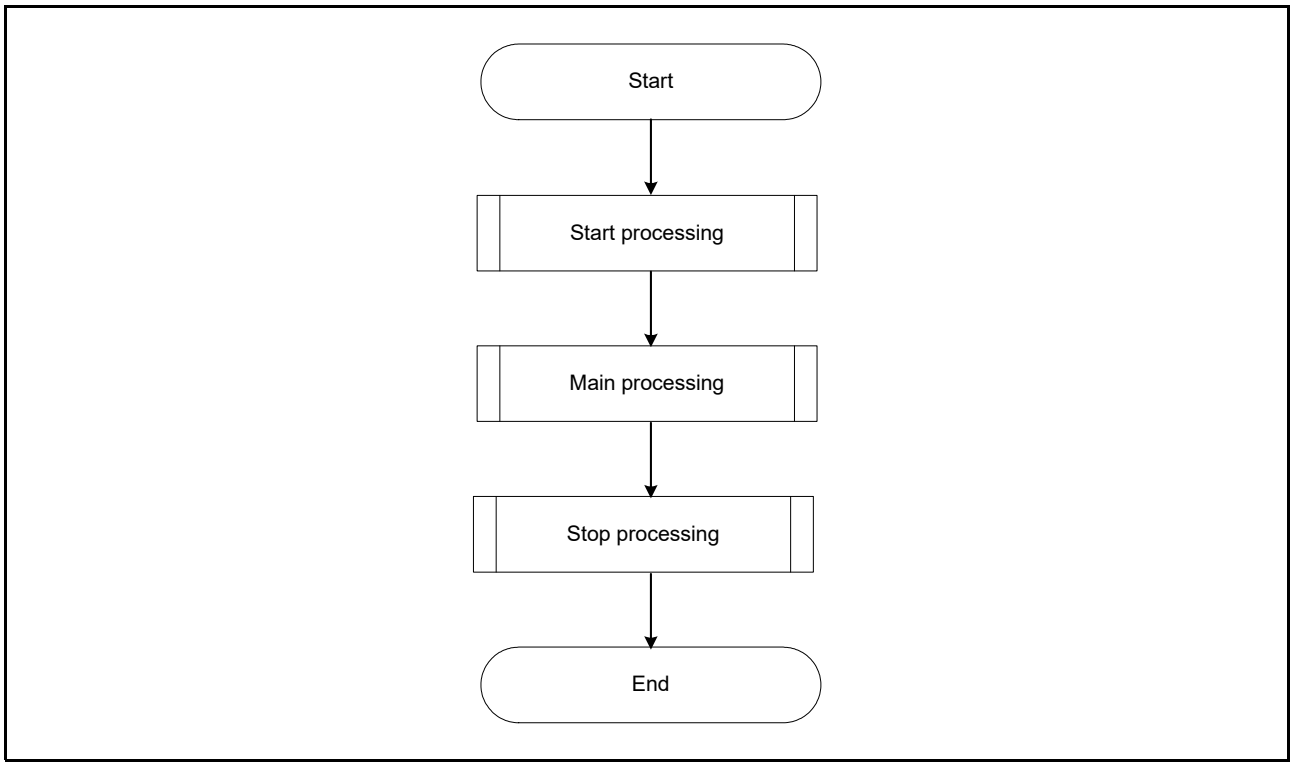


Figure 59.50 Processing Flow of Incremental Mode

(1) Startup Processing

Figure 59.51 shows the startup processing flow of the incremental mode. To restart the PCIF after changing register settings, initialize the internal information by using a software reset (SWRSTR.SWRST bit = 1) immediately before the startup processing flow.

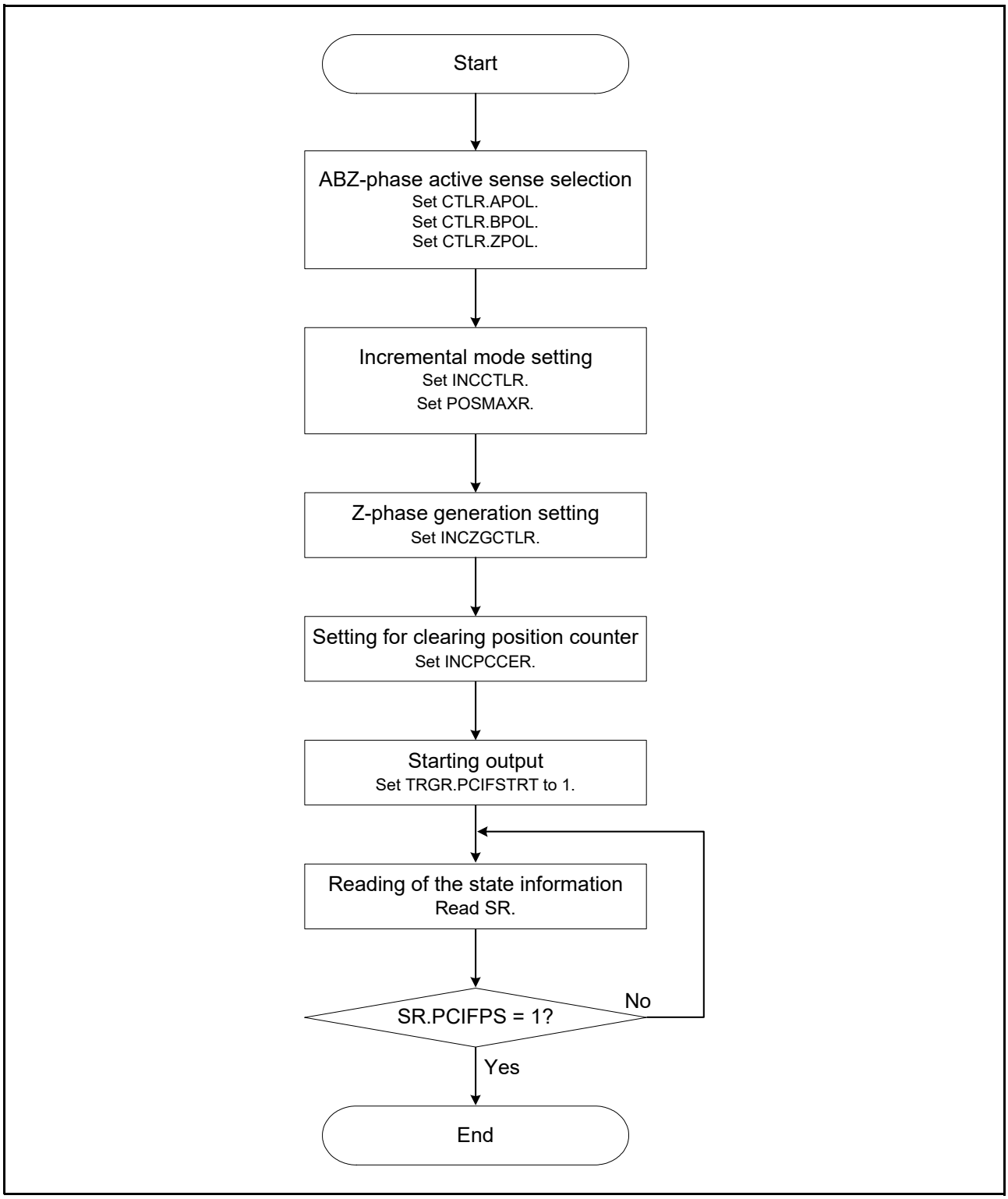


Figure 59.51 Startup Processing Flow of Incremental Mode

(2) Main Processing

Software operation is not required during main processing. ABZ-phase outputs are automatically generated from the external input of the PCIF. Based on AB-phase inputs, AB-phase signals are output according to the resolution ratio, and the Z-phase signal is output by using a Z-phase input, crest event trigger, trough event trigger, or software trigger input, or when the output counter's maximum value (POSMAXR.POSMAX[15:0] bits) is reached.
For details, refer to section 59.3.2, Incremental Mode.

(3) Stop Processing

Figure 59.52 shows the stop processing flow of the incremental mode.

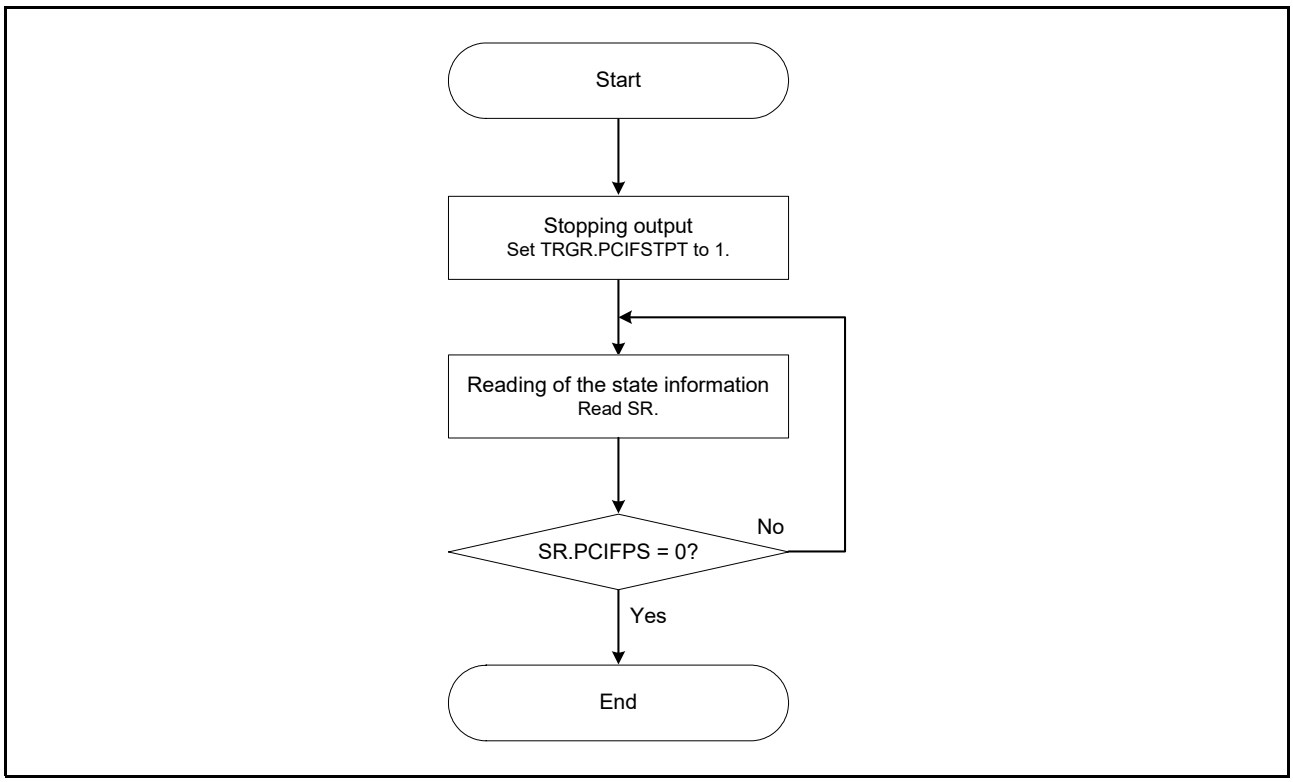


Figure 59.52 Stop Processing Flow of Incremental Mode

59.3.4.5 PCIF Stop Processing

Figure 59.53 shows the stop processing flow.

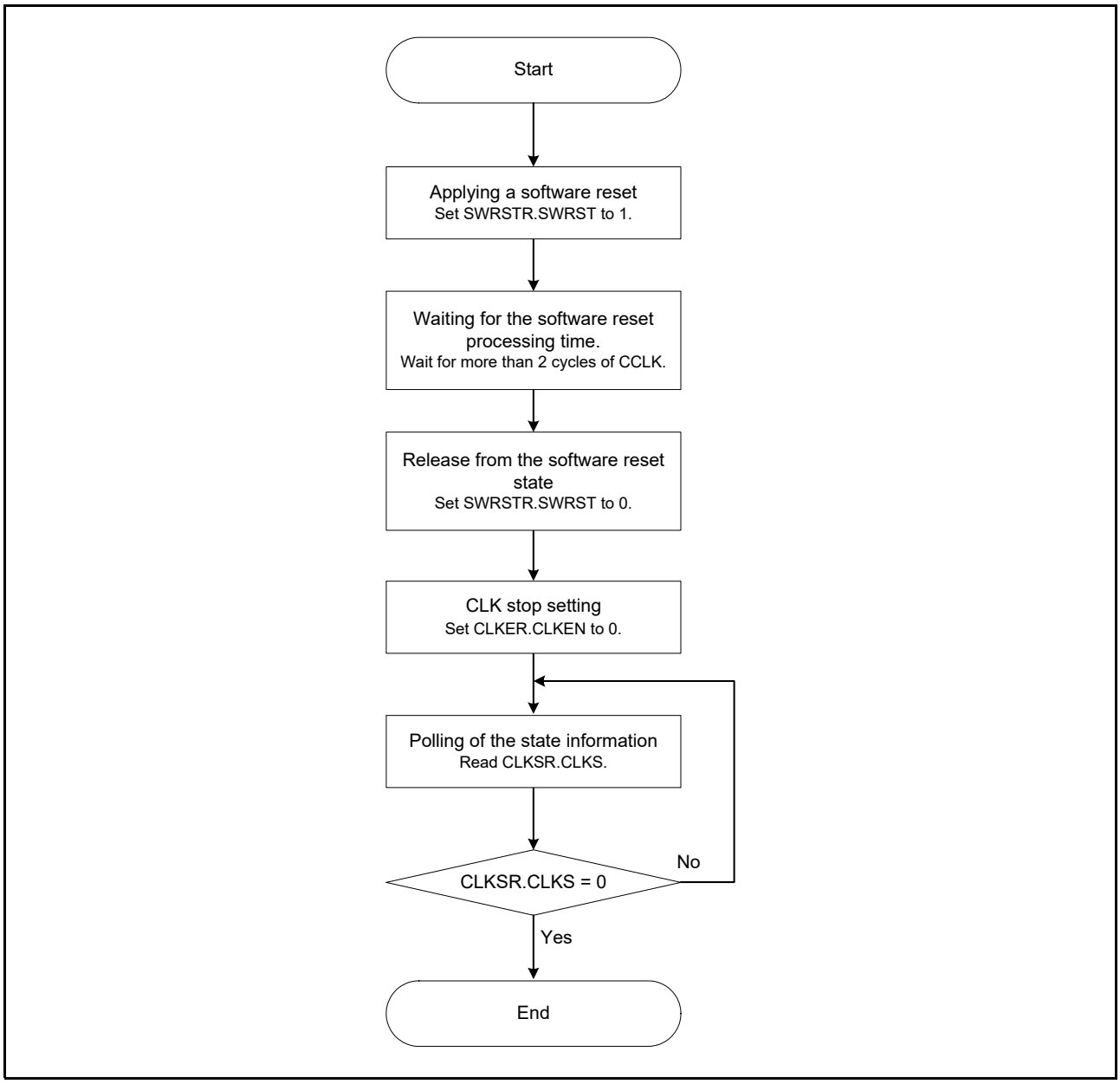


Figure 59.53 PCIF Stop Processing Flow

59.4 Event Link Function

59.4.1 Carrier Period Event Input (ELC_IN)

You can control the carrier period by setting the event-based mode (PHCCTLR.PHCCEDAMD bit = 0) and using the carrier period event (ELC_IN) output from the ELC.
For details, refer to section 59.3.1.3, Carrier Period Control, (1) Event-Based Mode.

59.4.2 Clearing the Internal Counters by Crest/Trough Event Trigger Input (PEK_IN, VLY_IN)

Input of a crest event trigger (PEK_IN) or trough event trigger (VLY_IN) from the ELC can initialize the internal counters (INCOCNTM[15:0] and INCICNTM[5:0] bits) to clear the AB-phase outputs.
For details, refer to section 59.3.2.2, Generating A- and B-Phase Signals, (3) Internal Counter Clearing Function.

59.4.3 Generating Z-Phase Output by Crest/Trough Event Trigger Input (PEK_IN, VLY_IN)

Input of a crest event trigger (PEK_IN) or trough event trigger (VLY_IN) from the ELC can assert the Z-phase output.
For details, refer to section 59.3.2.3, Generating Z-Phase Signals, (1) Selecting the Z-Phase Output Trigger.

59.5 Usage Notes

59.5.1 Restrictions on Input Pins

- Minimum toggle intervals for ABZ-phase inputs

Table 59.18 shows the minimum toggle intervals for ABZ-phase inputs. The toggle intervals in the table mean the intervals for toggling of A-phase or B-phase inputs for use in switching between counting up and counting down in Table 59.9 to Table 59.11. For mode 2 (when the rotation direction changes), the minimum toggle interval means the period indicated by a blue double-headed arrow in Figure 59.54.

Table 59.18 Minimum Toggle Intervals for ABZ-Phase Inputs

Input Protocol Mode and Condition	Minimum Required Toggle Interval
Mode 0	1 CCLK cycle + 1 CCLK_DIV cycle
Mode 1	1 CCLK cycle + 1 CCLK_DIV cycle
Mode 2 (when the rotation direction is constant)	1 CCLK cycle + 1 CCLK_DIV cycle
Mode 2 (when the rotation direction changes)	1 CCLK cycle + 2 CCLK_DIV cycles

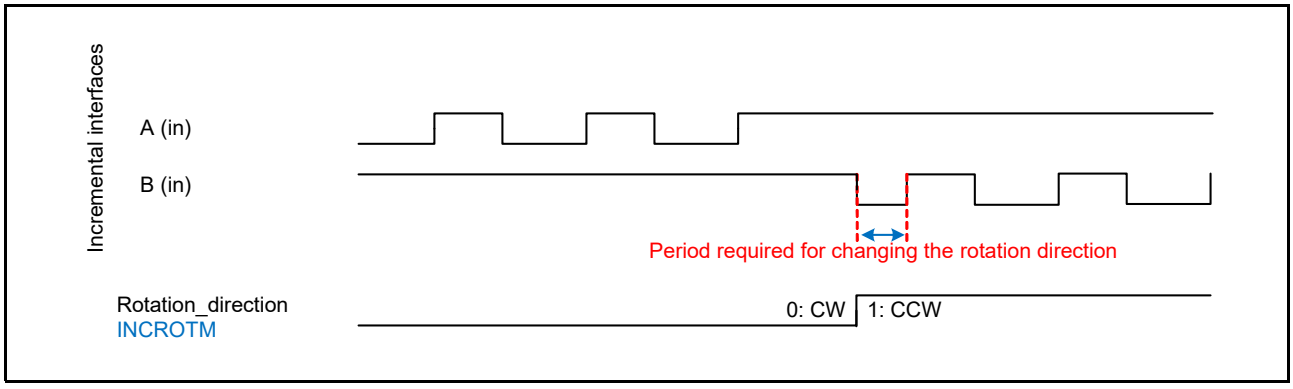


Figure 59.54 Waveforms When the Rotation Direction Changes in Mode 2

For modes 0 and 1 or mode 2 (when the rotation direction is constant), the number of cycles for synchronization (1 CCLK cycle) and the number of cycles that the CCLK_DIV domain circuit can receive (1 CCLK_DIV cycle) are required.

For mode 2 (when the rotation direction changes), in addition to the previously mentioned number of cycles, 1 CCLK_DIV cycle is required for the high-to-low period (first pulse) immediately after the rotation direction changes. This input restriction is provided to guarantee 1 CCLK_DIV cycle in the high-to-high period for output and 1 CCLK_DIV cycle in the high-to-low period of the first pulse. For details on this specification, refer to section 59.3.2.2, Generating A- and B-Phase Signals, (2) Internal Counter Operation, (c) AB-Phase Generation.
In any mode and condition, if the sufficient toggle intervals are not ensured, an edge might be lost when the rotation direction changes.

59.5.2 Restriction on Register Settings

- PHCPERIODR.PHCPERIOD[15:0] bit setting

Make sure to set the PHCPERIOD[15:0] bits to the mean value of the carrier period event intervals. If the mean value of the carrier period event intervals differs from the value set in the PHCPERIOD[15:0] bits, edges cannot be processed completely and will be lost during long-term operation.

59.5.3 Restrictions on Operation

- CTLR.PCIFMD bit setting timing

To switch the PCIF mode by changing the value of the PCIFMD bit, disable CCLK_DIV operation before writing to the bit in accord with the flow of startup processing in Figure 59.45.

- ELC startup timing

Operation of carrier period input is not guaranteed during the startup processing flow. Start carrier period generation by carrier period event input from the ELC after the startup processing flow ends.

- Action to be taken if an error occurs in phase counting mode

After an error specified in the PHCSR register occurs, operation is not guaranteed. If an error occurs, apply a software reset according to the stop processing flow in section 59.3.4.3, Flows of Phase Counting Mode, (3) Stop Processing. After the stop, use the PHCSCR register to clear the error state, and then review the register settings.

- Crest/trough event trigger input timing

If a crest event trigger (PEK_IN) or trough event trigger (VLY_IN) is input during the startup processing flow, operation is not guaranteed. Input the trigger after the startup processing flow ends.

59.5.4 Transition to Software Standby Mode

For phase counting mode, stop the PCIF according to Figure 59.49 Stop Processing Flow of Phase Counting Mode. After phase-counting by the PCIF has been stopped, stop the core clock according to Figure 59.53 PCIF Stop Processing Flow, and then initiate a transition to software standby mode. For incremental mode, stop the PCIF according to Figure 59.52 Stop Processing Flow of Incremental Mode. After incremental mode operation of the PCIF has been stopped, stop the core clock according to Figure 59.53 PCIF Stop Processing Flow, and then initiate a transition to software standby mode.

59.5.5 Return from Software Standby Mode

For phase counting mode, after a return from software standby mode, start the core clock according to Figure 59.45 PCIF Startup Processing Flow, and then start the PCIF operating in phase counting mode according to Figure 59.47 Startup Processing Flow of Phase Counting Mode. For incremental mode, after a return from software standby mode, start the core clock according to Figure 59.45 PCIF Startup Processing Flow, and then start the PCIF operating in incremental mode according to Figure 59.51 Startup Processing Flow of Incremental Mode.

60. 16-Bit A/D Converter (ADC16H)

60.1 Overview

This MCU contains three units of noise-shaping SAR-type A/D converters (ADC) that is the hybrid architecture with features of SAR-type and delta-sigma modulation-type. The A/D converter unit 0 (ADC0) can select up to 8 channels of analog inputs. The A/D converter unit 1 (ADC1) can select up to 28 channels of analog inputs. The A/D converter unit 2 (ADC2) can select up to 28 channels of analog inputs. The 28 channels of analog input shared by unit 1 and unit 2 use the same port. The temperature sensor, internal reference voltage, D/A converters, and VBATT 1/6 voltage monitor output can be A/D-converted by A/D converter unit 0, unit 1 or unit 2. A/D conversion data can be selected from 16-bit, 14-bit, 12-bit, and 10-bit length data formats.
The ADC has the following features.

- Resolution: Up to 16-bit
- Fast conversion: Up to 6.25 Msps (0.16 µs per channel) (when A/D conversion clock ADCLK = 50 MHz)
- Input channels: Up to 36 analog input channels
- Support single-ended input or differential inputs
- Self-calibration Function
- Built-in channel-dedicated sample-and-hold circuit (S&H)

Table 60.1 shows the ADC specifications, Table 60.2 shows the correspondence between ADC functions and ADC operation mode, and Table 60.3 shows the ADC I/O pins.
Figure 60.1 shows the block diagram of ADC. Figure 60.2 shows the analog channel structure for A/D converter unit 0. Figure 60.3 shows the analog channel structure for A/D converter unit 1. Figure 60.4 shows the analog channel structure for A/D converter unit 2.
Table 60.4 shows the ADC channel configuration of the analog inputs. Table 60.5 shows the ADC channel configuration of the extended analog function.

Table 60.1 ADC Specifications (1/4)

Item	Description
Number of units	Three units (unit 0, unit 1, and unit 2)
Input channels	• Support single-ended input or differential inputs • Up to 36 analog input channels • A/D converter unit 0: Up to 8 analog input channels at single-ended input, 8 analog input channels (4 pairs) of which support differential inputs. • A/D converter unit 1: Up to 28 analog input channels at single-ended input, 8 analog input channels (4 pairs) of which support differential inputs. • A/D converter unit 2: Up to 28 analog input channels at single-ended input, 8 analog input channels (4 pairs) of which support differential inputs. • 28 analog input channels are shared by A/D converter unit 1 and unit 2
Extended analog function	Self-diagnosis, temperature sensor, internal reference voltage, D/A converter (DA0 and DA1), VBATT 1/6 voltage monitor output
A/D conversion method	• Successive approximation register method in SAR mode • Noise-shaping successive approximation register method in oversampling mode and hybrid mode
Resolution of A/D converter	• Up to 12-bits in SAR mode • Up to 16-bits in oversampling mode or hybrid mode
Conversion time	0.16 µs per channel (when A/D conversion clock ADCLK = 50 MHz).
A/D conversion clock	The A/D conversion clock (ADCLK) can be set by selecting the clock source and the division ratio as follows: • The clock source: the peripheral module clock PCLKA*1, ADC clock ADCCLK*1, the GPT clock GPTCLK*1. • The division ratio: 1/2/3/4/5/6/7/8 A/D conversion clock (ADCLK) can operate between 25 MHz at a minimum and 60 MHz at a maximum.
A/D conversion data	• A/D conversion results are stored in data register or FIFO. • A/D conversion results are available in 16-, 14-, 12-, and 10-bit data formats.

Table 60.1 ADC Specifications (2/4)

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Operating modes</td><td>
• SAR mode
  • A/D converter samples the signal source once, and convert by successive approximation register method
  • Fast A/D conversion
  • Support up to 8 channels per one scan group
  • Support only single-ended input (excluding self-diagnosis function)
• Oversampling mode
  • A/D converter oversamples the signal source, and converts analog-to-digital by noise shaping successive approximation register method
  • High-accuracy A/D conversion
  • Support up to 8 channels per one scan group
  • Support single-ended input and differential input
• Hybrid mode
  • A/D converter oversamples the signal source, and converts analog-to-digital by noise shaping successive approximation register method
  • High-accuracy A/D conversion
  • High data rate in continuous scanning operation
  • Support up to 4 channels per one scan group
  • Support single-ended input and differential input
</td></tr>
<tr><td>Scan mode</td><td>
• Single scan mode
  • Assign any selected analog input or analog channel of the extended analog function to any scan group, and convert the selected analog input only once per scan group for A/D conversion.
  • By selecting the scan start conditions for each scan group individually, A/D conversion for each scan group can be started at different times.
• Continuous scan mode
  • Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
• Background continuous scan mode
  • Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
  • The A/D conversion is performed in the background and, if the A/D conversion start trigger is entered, the A/D conversion data at that point is acquired.
• Same channel continuous scan mode
  • Repeats A/D conversion of one analog channel assigned to any scan group
  • When an A/D conversion trigger is input during scanning, A/D conversion of the next analog channel begins
  • A/D conversion channel can be changed whenever an external factor occurs
• Fixed channel continuous scan mode
  • Analog channels designated as fixed channels are A/D converted in hybrid mode, and analog channels designated as non-fixed channels are A/D converted in SAR mode
  • The non-fixed channels to be A/D converted are switched every time the scan operation is completed, and A/D conversion of fixed and non-fixed channels is repeated
  • A larger number of channels can be converted than in normal hybrid mode
  • The scan interval can be adjusted by adding channels that are not converted (free channels) after scanning non-fixed channels
</td></tr>
<tr><td>Combination of operation mode and scan mode</td><td>
<table>
<tr><th rowspan="2">Scan mode</th><th colspan="3">Operation mode of A/D converter</th></tr>
<tr><th>SAR mode</th><th>Oversampling mode</th><th>Hybrid mode</th></tr>
<tr><td>Single scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Continuous scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Background continuous scan mode</td><td>—</td><td>—</td><td>✓</td></tr>
<tr><td>Same channel continuous scan mode</td><td>—</td><td>✓</td><td>—</td></tr>
<tr><td>Fixed Channel Continuous Scan Mode</td><td>—</td><td>—</td><td>✓Fixed*1</td></tr>
</table>
✓: available, —: not available
Note 1. Non-fixed channels are in SAR mode
</td></tr>
<tr><td>Conditions for A/D conversion start</td><td>
• Software trigger (for simultaneous activation of scan group: max. 12 triggers)
• Software trigger (for individual scan group activation: up to 12 triggers)
• Trigger from event link controller: 9 triggers
• Triggers from GPT: 32 triggers
• External trigger input: 3 triggers (ADTRGn input (n = 0 to 2))
</td></tr>
</table>

Table 60.1 ADC Specifications (3/4)

Item	Description
Functions	• Virtual channel function (50 virtual channels) • Scan group function (up to 12 scan groups) • Variable sampling time (select from 16 tables per virtual channel) • Channel-dedicated sample-and-hold circuit (S&H) (3 SH units for A/D converter unit 0, constant sampling can be set) • Self-diagnosis function for A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection assist function (discharge function/precharge function) • Selectable the data format from 16- / 14- / 12- / 10-bit. • Selectable the signed-data or the unsigned-data format • Limiter clip function (Up to 8 tables) • Compare match function (Up to 8 tables) • Self-calibration function • User's gain adjustment function • User's offset adjustment function • Built-in FIFO (16 stages per each scan group) • FIFO clear function (when scanning starts/resumes) • Built-in digital filter (2 type) (support only oversampling mode and hybrid mode) • A/D conversion data decimation function (only oversampling mode (same channel continuous scan mode) is supported) • Multiple A/D converters unit-to-unit synchronous operation function • GPT can be used for inter-unit sync signals • Accuracy priority function (High accuracy mode)
Interrupt sources	• A/D scan end interrupt • Generates the interrupt requests and the ELC events at the end of A/D scan operation for the scan group i (ADIi (i = 0 to 11)). The interrupt requests are independent for each scan group. • FIFO data read request interrupt • Generates the interrupt requests when the number of vacant stages of FIFO for the scan group i becomes less than or equal to the specified value (FIFOREQi (i = 0 to 11)). The interrupt requests are independent for each scan group. • FIFO data overflow interrupt • Generates the interrupt request when the overflow occurs in any of the FIFO for the scan groups 0 to 11 (FIFOOVF). • Limiter clip interrupt • Generates the interrupt request when the limiter clip using the limiter tables 0 to 7 is occurred for A/D conversion results (LIMCLPI). • Compare match interrupt • Generates the interrupt requests when the compare match using the compare match tables j are occurred for A/D conversion results (CMPIj (j = 0 to 7)). The interrupt requests are independent for each compare match table. • Composite compare match interrupt • Generates the interrupt requests and the ELC events when the compare match of the composite condition using compare match tables 0 to 7 occurs (CCMPMm (m = 0 to 7)). • Composite compare unmatch interrupt • Generates the interrupt requests and the ELC events when the compare unmatch of the composite condition using compare match tables 0 to 7 occurs (CCMPUMm (m = 0 to 7)). • A/D converter error interrupt • Generates the interrupt requests when the operational error is detected in A/D converter unit j (ERRj (j = 0 to 2)). • A/D conversion overflow interrupt • Generates the interrupt request when the A/D conversion result overflow is occurred (RESOVFj (j = 0 to 2)) The interrupt requests are independent for A/D converter unit j. • A/D converter calibration end interrupt • Generates the interrupt requests at the end of calibration operation of A/D converter unit j (CALENDj (j = 0 to 2)).
ELC interface	• Trigger Input • Scan can be started by the trigger from the ELC. • Event Generation • An event is generated at the end of each scan operation for the scan group 0 to 11. • An event is generated when composite compare match or composite compare unmatch is occurred.
Reference voltage	VREFH0 is the analog reference voltage. VREFL0 is the analog reference ground.
Module-stop function	Module-stop state can be set to reduce power consumption.*2 A/D converter units can be set to sleep mode independently

Table 60.1 ADC Specifications (4/4)

Item	Description
Protect Zone filter	Security attribute can be set.*3

Note 1. For details, see section 9, Clock Generation Circuit.
Note 2. For details, see section 11, Low Power Consumption
Note 3. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Table 60.2 Association between ADC Functions and ADC Operation Mode

Function			SAR Mode		Oversampling Mode			Hybrid Mode			
			Single Scan Mode	Continuous Scan Mode	Single Scan Mode	Continuous Scan Mode	Same Channel Continuous Scan Mode	Single Scan Mode	Continuous Scan Mode	Background Continuous Scan Mode	Fixed Channel Continuous Scan Mode
Analog input	Single-ended input		✓	✓	✓	✓	✓	✓	✓	✓	✓
	Differential input		✓	✓	✓	✓	✓	✓	✓	✓	✓
	Channel-dedicated sample-and-hold circuit		✓	—	—	—	—	✓	✓	✓	✓*1
Diagnosis function/ extended analog function	Disconnection detection assist function	Channel-dedicated sample-and-hold circuit disabled	✓	✓	—	—	—	✓	✓	—	—
		Channel-dedicated sample-and-hold circuit enabled	✓	—	—	—	—	—	—	—	—
	Self-diagnosis	Channel-dedicated sample-and-hold circuit disabled	✓	✓	✓	—	—	✓	—	—	—
		Channel-dedicated sample-and-hold circuit enabled	✓	—	—	—	—	—	—	—	—
	Internal reference voltage		✓	✓	✓	✓	✓	✓	✓	✓	✓
	VBATT 1/6 voltage monitor output		✓	✓	✓	✓	✓	✓	✓	✓	✓
	Temperature sensor		✓	✓	✓	✓	✓	✓	✓	✓	✓
	D/A converter (DA0, DA1)		✓	✓	✓	✓	✓	✓	✓	✓	✓
Scanning operation	Group priority operation		✓*2	✓*2	—	—	—	—	—	—	—
	Multiple A/D converter synchronous operation		✓	✓	✓	✓	✓	✓	✓	✓	✓
Digital calculation	Digital filter function		—	—	✓*3	✓*3	✓*3	✓*3	✓*3	✓*3	✓*3
	User's gain/offset adjustment function		✓	✓	✓	✓	✓	✓	✓	✓	✓
	Addition/average function		✓	✓	✓	✓	✓	✓	✓	✓	—
	Limiter clip function		✓	✓	✓	✓	✓	✓	✓	✓	✓
	Compare match function		✓	✓	✓	✓	✓	✓	✓	✓	✓
FIFO function			✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. Fixed channels can be used when the number of fixed channels is set to 3 and all fixed channels are set to channel-dedicated sample-and-hold circuits. Non-fixed channels (SAR mode) cannot use the channel-dedicated sample-and-hold circuits.
Note 2. When group scan is enabled, only the highest priority group can use the channel-dedicated sample-and-hold circuits.
Note 3. Use of the digital filter function is required when in oversampling mode and hybrid mode.

Table 60.3 Input/Output Pins of ADC

Item	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block ground pin
VREFH0	Input	Analog reference power supply pin
VREFL0	Input	Analog reference ground pin
AN000 to AN035	Input	Analog input pins
SHIN0P to SHIN2P	Input	Analog input pins for channel-dedicated sample-and-hold circuit • For single-ended input: Connect to signal source • For differential input: Functions as the non-inverting inputs (+) terminal for differential input
SHIN0N to SHIN2N	Input	Analog input pins for channel-dedicated sample-and-hold circuit • For single-ended input: Connect to signal source • For differential input: Functions as the inverting inputs (–) terminal for differential input
ADTRG0, ADTRG1, ADTRG2	Input	External trigger input pin for starting A/D conversion, active-low.
ADST0, ADST1, ADST2	Output	A/D conversion period monitor output
ADCNV0, ADCNV1, ADCNV2	Output	A/D comparison period monitor output
ADSYNC	Output	A/D converter unit inter-unit synchronous monitor output

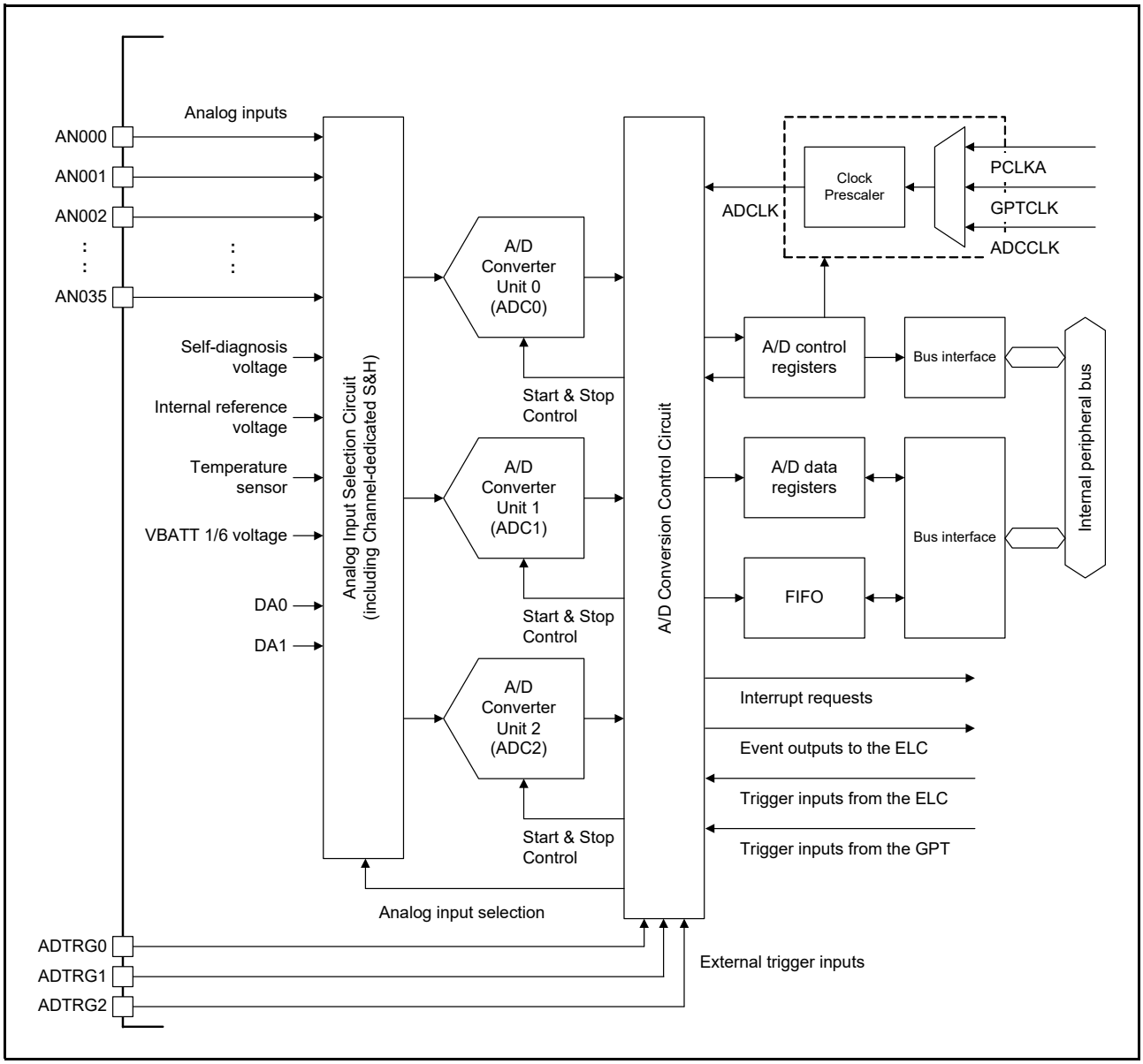


Figure 60.1 Block Diagram of ADC

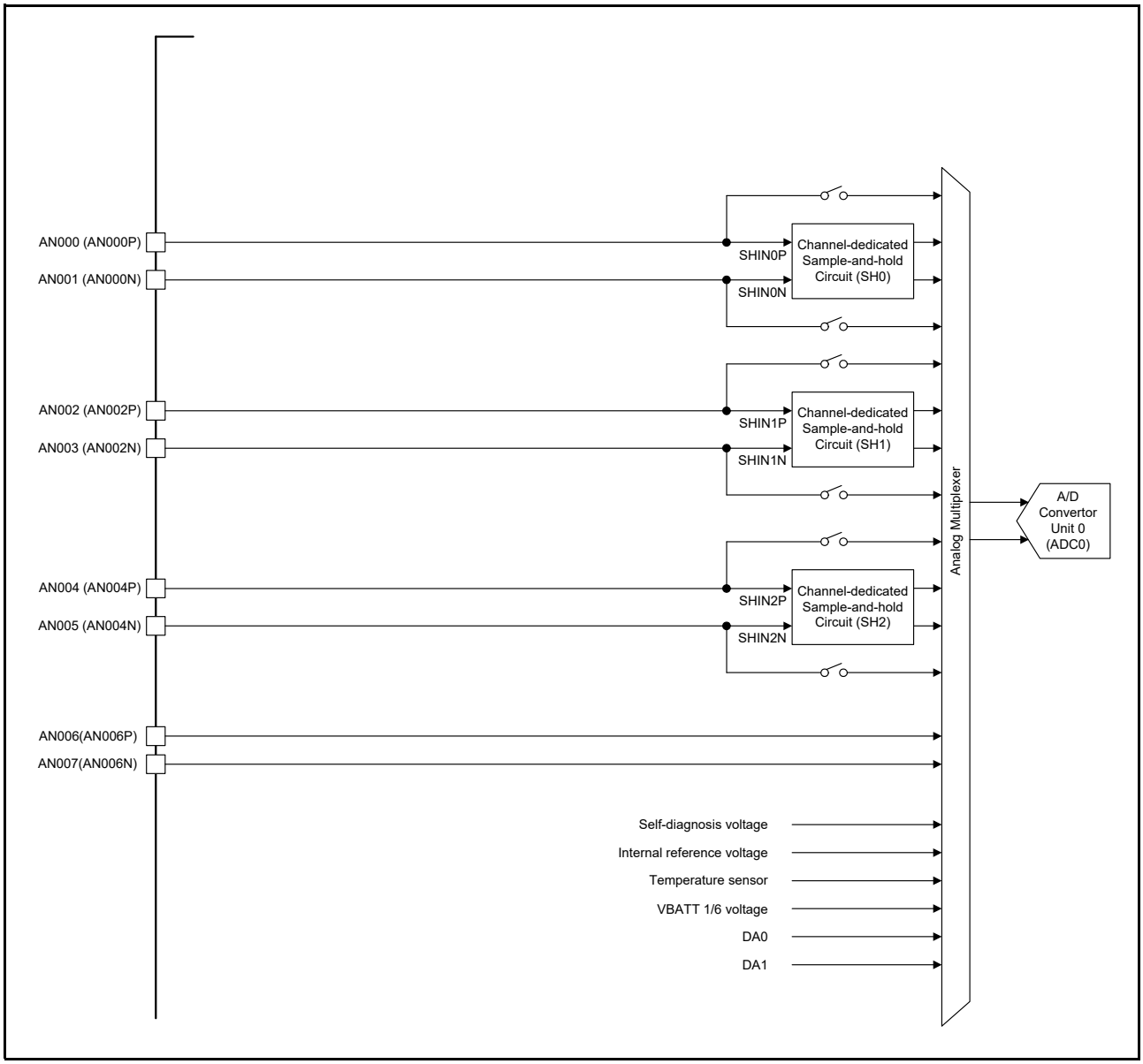


Figure 60.2 Analog Channel Structure for A/D Converter Unit 0

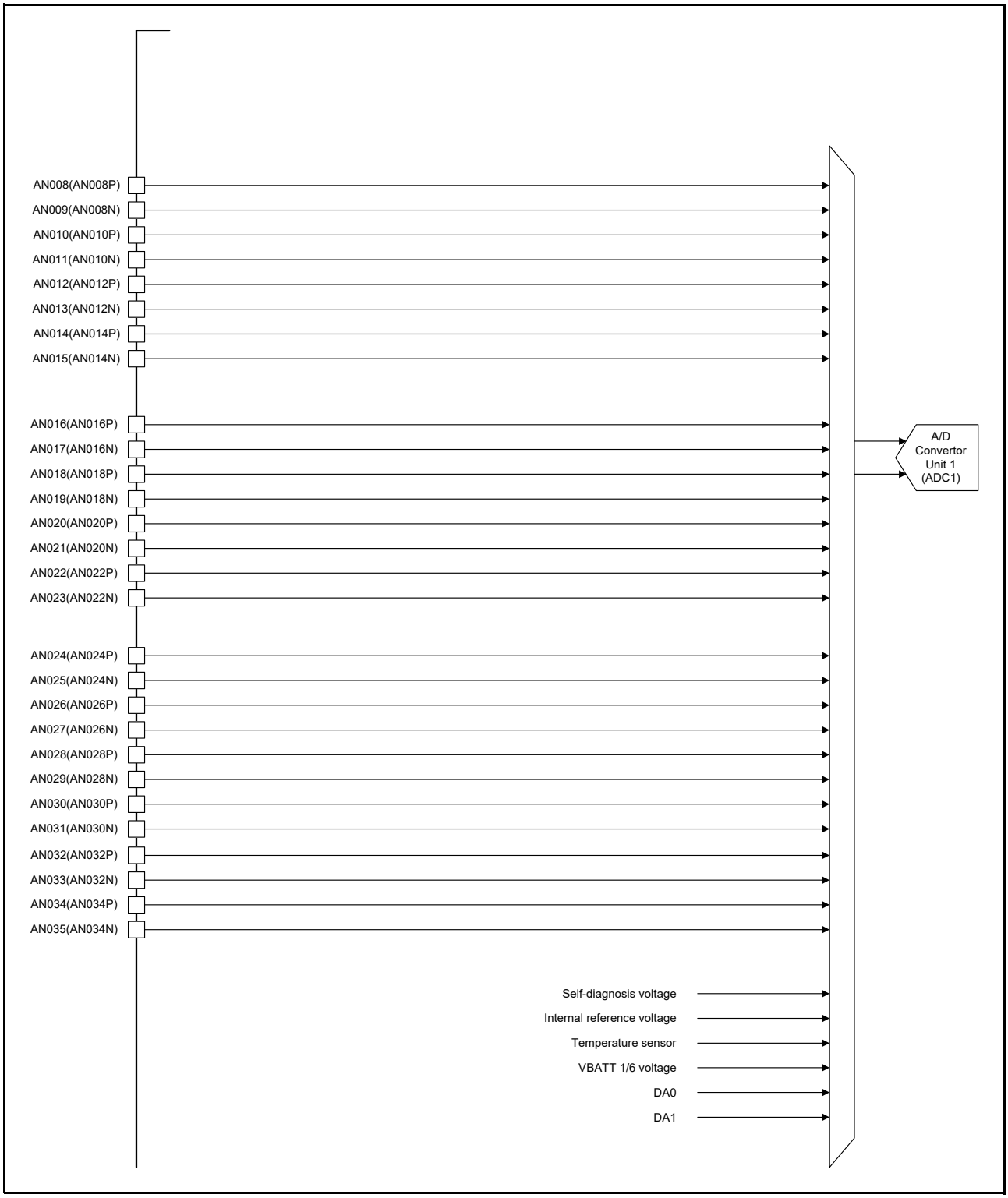


Figure 60.3 Analog Channel Structure for A/D Converter Unit 1

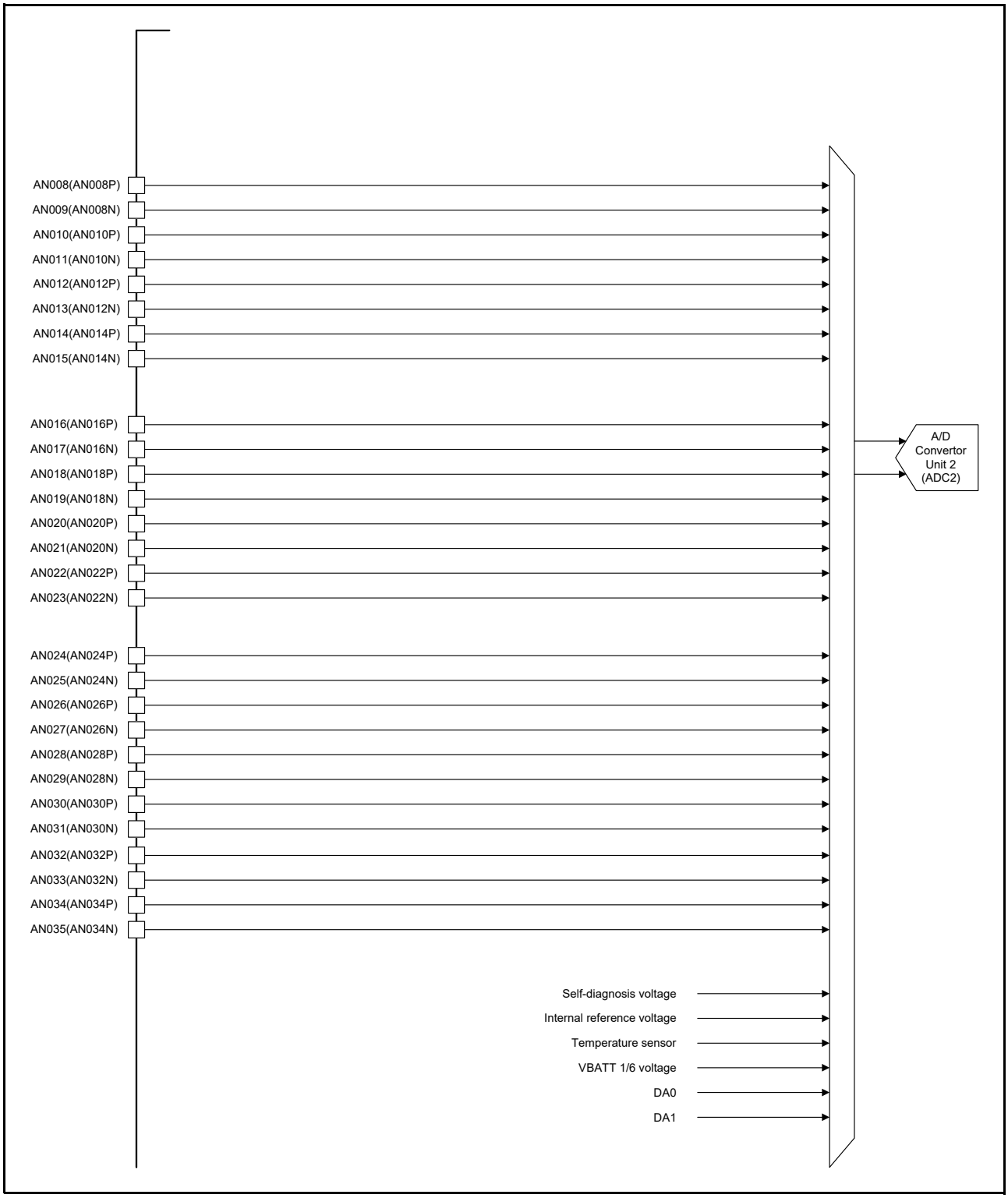


Figure 60.4 Analog Channel Structure for A/D Converter Unit 2

Table 60.4 ADC Channel Configuration of the Analog Inputs

Analog Channel Number	Source of Analog Signal		A/D Converter Unit 0		A/D Converter Unit 1		A/D Converter Unit 2	
	Analog Input	Channel-Dedicated Sample-and-Hold Circuit (S&H)*2	Single-Ended Input	Differential Input *3	Single-Ended Input	Differential Input *3	Single-Ended Input	Differential Input *3
0	AN000 (AN000P)	SHIN0P	✓	✓	—	—	—	—
1	AN001 (AN000N)	SHIN0N	✓		—		—	
2	AN002 (AN002P)	SHIN1P	✓	✓	—	—	—	—
3	AN003 (AN002N)	SHIN1N	✓		—		—	
4	AN004 (AN004P)	SHIN2P	✓	✓	—	—	—	—
5	AN005 (AN004N)	SHIN2N	✓		—		—	
6	AN006 (AN006P)	—	✓	✓	—	—	—	—
7	AN007 (AN006N)	—	✓		—		—	
8	AN008 (AN008P) *1	—	—	—	✓	✓	✓	✓
9	AN009 (AN008N) *1	—	—		✓		✓	
10	AN010 (AN010P) *1	—	—	—	✓	✓	✓	✓
11	AN011 (AN010N) *1	—	—		✓		✓	
12	AN012 (AN012P) *1	—	—	—	✓	✓	✓	✓
13	AN013 (AN012N) *1	—	—		✓		✓	
14	AN014 (AN014P) *1	—	—	—	✓	✓	✓	✓
15	AN015 (AN014N) *1	—	—		✓		✓	
16	AN016 (AN016P) *1	—	—	—	✓	—	✓	—
17	AN017 (AN016N) *1	—	—		✓		✓	
18	AN018 (AN018P) *1	—	—	—	✓	—	✓	—
19	AN019 (AN018N) *1	—	—		✓		✓	
20	AN020 (AN020P) *1	—	—	—	✓	—	✓	—
21	AN021 (AN020N) *1	—	—		✓		✓	
22	AN022 (AN022P) *1	—	—	—	✓	—	✓	—
23	AN023 (AN022N) *1	—	—		✓		✓	
24	AN024 (AN024P) *1	—	—	—	✓	—	✓	—
25	AN025 (AN024N) *1	—	—		✓		✓	
26	AN026 (AN026P) *1	—	—	—	✓	—	✓	—
27	AN027 (AN026N) *1	—	—		✓		✓	
28	AN028 (AN028P) *1	—	—	—	✓	—	✓	—
29	AN029 (AN028N) *1	—	—		✓		✓	
30	AN030 (AN030P) *1	—	—	—	✓	—	✓	—
31	AN031 (AN030N) *1	—	—		✓		✓	
32	AN032 (AN032P) *1	—	—	—	✓	—	✓	—
33	AN033 (AN032N) *1	—	—		✓		✓	
34	AN034 (AN034P) *1	—	—	—	✓	—	✓	—
35	AN035 (AN034N) *1	—	—		✓		✓	

Note 1. Do not perform A/D conversions of the same signal source from both A/D converter unit 1 and unit 2.
Note 2. When the channel-dedicated sample-and-hold circuit is used with a single-ended input, the SHINxP (non-inverting input (+)) and SHINxN (inverting input (–)) channels should be assigned to different scan groups (x = 0 to 2), or assigned to the same scan group with one of them set as the bias path.
Note 3. When performing A/D conversion with differential-input, set the even-numbered channel as the A/D conversion target. For differential input, the even-numbered channels function as non-inverting inputs (+), and the odd-numbered channels function as inverting inputs (–).

Table 60.5 ADC Channel Configuration of the Extended Analog Function

Analog Channel Number	Source of Analog Signal	A/D Converter Unit 0		A/D Converter Unit 1		A/D Converter Unit 2	
	Analog Input	Single-Ended Input*2	Differential Input*3	Single-Ended Input*2	Differential Input*3	Single-Ended Input*2	Differential Input*3
96	Self-diagnosis Unit 0	—	✓	—	—	—	—
97	Self-diagnosis Unit 1	—	—	—	✓	—	—
98	Self-diagnosis Unit 2	—	—	—	—	—	✓
100	Temperature sensor*1	✓	—	✓	—	✓	—
101	Internal reference voltage*1	✓	—	✓	—	✓	—
102	VBATT 1/6*1	✓	—	✓	—	✓	—
104	D/A converter channel 0*1	✓	—	✓	—	✓	—
105	D/A converter channel 1*1	✓	—	✓	—	✓	—
112	Self-diagnosis S&H 0	—	✓	—	—	—	—
113	Self-diagnosis S&H 1	—	✓	—	—	—	—
114	Self-diagnosis S&H 2	—	✓	—	—	—	—

Note 1. Do not perform A/D conversions of the same signal source from multiple A/D converter units.
Note 2. The extend analog function (except for the self-diagnosis function) is only supported with the single-ended input. Do not set to the differential input.
Note 3. The self-diagnosis function is only supported with differential input.

60.2 Register Descriptions

60.2.1 A/D Conversion Clock Enable Register (ADCLKENR)

Address(es): ADC_B.ADCLKENR 8743 8000h, ADC_B_NS.ADCLKENR 0743 8000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLKEN	ADCLK Operating Enable	0: Stop ADCLK 1: Supply ADCLK	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCLKENR controls the supply of the A/D conversion clock (ADCLK).

60.2.2 A/D Conversion Clock Status Register (ADCLKSR)

Address(es): ADC_B.ADCLKSR 8743 8004h, ADC_B_NS.ADCLKSR 0743 8004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYLOCKSR	CLKSR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLKSR	ADCLK Status	0: ADCLK is stopped 1: ADCLK is in supply	R
b1	SYLOCKSR	Sync Status	0: Not synchronized with an external signal (synchronization signal from GPT). 1: Synchronized with an external signal (synchronization signal from GPT).	R
b31 to b2	—	Reserved	These bits are read as 0.	R

The ADCLKSR register indicates the supply status of the A/D conversion clock (ADCLK).

CLKSR Bit (ADCLK Status)

The CLKSR bit indicates the supply status of the A/D converter operating clock (ADCLK).

SYLOCKSR Bit (Sync Status)

The SYLOCKSR bit indicates the synchronization status with an external signal (synchronization signal from the GPT). This bit is valid when the ADSYCR.ADSYEXTEN bit is 1 and one of the ADSYCR.ADSYDISm (m = 0 to 2) bits is 0. In all other cases, this bit is invalid and is always set to 0.

This bit is 1 when the ADC is synchronized to an external signal (sync signal from GPT). This bit is 0 when the clock is stopped (CLKSR = 0).

To detect the synchronization status with an external signal (synchronous signal from GPT), it takes a maximum of three ADCLK cycles after the clock status bit (CLKSR) becomes 1.

60.2.3 A/D Conversion Clock Control Register (ADCLKCR)

Address(es): ADC_B.ADCLKCR 8743 8008h, ADC_B_NS.ADCLKCR 0743 8008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	DIVR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CLKSEL[1:0]	ADCLK Clock Source Select	b1 b0 0 0: ADC clock (ADCCLK) 0 1: GPT clock (GPTCLK) 1 0: Peripheral module clock A (PCLKA) 1 1: Setting prohibited	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	DIVR[2:0]	Clock Division Ratio Select	b18 b16 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/3 0 1 1: 1/4 1 0 0: 1/5 1 0 1: 1/6 1 1 0: 1/7 1 1 1: 1/8	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCLKCR register selects the frequency of A/D conversion clock (ADCLK). The frequency of ADCLK should be set within the range specified in the section 67, Electrical Characteristics.

CLKSEL[1:0] Bits (ADCLK Clock Source Select)

CLKSEL[1:0] bits select the clock source of A/D conversion clock (ADCLK).

DIVR[2:0] Bits (Clock Division Ratio Select)

DIVR[2:0] bits select the division ratio for the clock source selected by CLKSEL[1:0] bits. The clock divided by the clock source selected by CLKSEL[1:0] bits at the ratio set by DIVR[2:0] bits becomes the A/D conversion operating clock (ADCLK).

60.2.4 A/D Converter Synchronous Operation Control Register (ADSYCR)

Address(es): ADC_B.ADSYCR 8743 800Ch, ADC_B_NS.ADSYCR 0743 800Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ADSYEXTEN	—	—	—	—	—	ADSYDIS2	ADSYDIS1	ADSYDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADSYCYC [7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ADSYCYC [7:0]	Synchronous Operation Period Cycle	These bits should be set in the range from 5 to 255. The settings of 0 to 4 are prohibited.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ADSYDIS0	ADC0 Synchronous Operation Select	0: Enable ADC0 synchronous operation 1: Disable ADC0 synchronous operation	R/W
b17	ADSYDIS1	ADC1 Synchronous Operation Select	0: Enable ADC1 synchronous operation 1: Disable ADC1 synchronous operation	R/W
b18	ADSYDIS2	ADC2 Synchronous Operation Select	0: Enable ADC2 synchronous operation 1: Disable ADC2 synchronous operation	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ADSYEXTEN	External Signal (GPT Sync Signal) Synchronization Enable	0: Do not synchronize the synchronization timing between the external signal (GPT synchronization signal) and the ADC unit. 1: Synchronize the synchronization timing between the external signal (GPT synchronization signal) and the ADC unit.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSYCR register controls the synchronous operation function. For more information, refer to section 60.3.20, Synchronous Operation.

ADSYCYC [7:0] Bits (Synchronous Operation Period Cycle)

These bits are selected the synchronous operation period cycles.
These bits set in the range from 5 to 255 regardless of whether the synchronous operation function is used or not. The settings of 0 to 4 are prohibited.

ADSYDIS0 Bit (ADC0 Synchronous Operation Select)

ADSYDIS0 bit selects enable/disable the synchronous operation of ADC0. If the synchronous operation is enabled, the ADC0 operates synchronously with other A/D converter that has synchronous operation enabled.

ADSYDIS1 Bit (ADC1 Synchronous Operation Select)

ADSYDIS1 bit selects enables/disables to the synchronous operation of ADC1. If the synchronous operation is enabled, the ADC1 operates synchronously with other A/D converter that has synchronous operation enabled.

ADSYDIS2 Bit (ADC2 Synchronous Operation Select)

ADSYDIS2 bit selects enables/disables to the synchronous operation of ADC2. If the synchronous operation is enabled, the ADC2 operates synchronously with other A/D converter that has synchronous operation enabled.

ADSYEXTEN Bit (External Signal (GPT Sync Signal) Synchronization Enable)

The ADSYEXTEN bit enables synchronization with an external signal (synchronization signal from GPT) when performing inter-unit synchronous operation. This bit is only valid for the ADC of a unit that has the ADSYDISx (x = 0 to 2) bit set to 0 when an external signal (synchronous signal from GPT) is being input.

Before setting this bit to 1, set the GPT registers and input an external signal (synchronous signal from the GPT) to the ADC. For the GPTW settings, refer to 24 General Purpose PWM Timer (GPTW).

When this bit is set to 1, A/D conversion is performed in synchronization with an external signal (synchronous signal from GPT).

This allows the synchronous operation wait time to be constant when A/D conversion is started from the GPT.

60.2.5 A/D Converter Unit Sleep Control Register 0 (ADUSLPCR0)

Address(es): ADC_B.ADUSLPCR0 8743 8010h, ADC_B_NS.ADUSLPCR0 0743 8010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADUSLP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADUSLP0	ADC0 Sleep Setting	0: ADC0 is set in normal operation mode 1: ADC0 is set in sleep mode	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADUSLPCR0 register is used to control the A/D converter unit 0 Sleep mode.

60.2.6 A/D Converter Unit Sleep Control Register 1 (ADUSLPCR1)

Address(es): ADC_B.ADUSLPCR1 8743 8014h, ADC_B_NS.ADUSLPCR1 0743 8014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADUSLP1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADUSLP1	ADC1 Sleep Setting	0: ADC1 is set in normal operation mode 1: ADC1 is set in sleep mode	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADUSLPCR1 register is used to control the A/D converter unit 1 Sleep mode.

60.2.7 A/D Converter Unit Sleep Control Register 2 (ADUSLPCR2)

Address(es): ADC_B.ADUSLPCR2 8743 8018h, ADC_B_NS.ADUSLPCR2 0743 8018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADUSLP2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADUSLP2	ADC2 Sleep Setting	0: ADC2 is set in normal operation mode 1: ADC2 is set in sleep mode	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADUSLPCR2 register is used to control the A/D converter unit 2 Sleep mode.

60.2.8 A/D Conversion Error Interrupt Enable Register (ADERINTCR)

Address(es): ADC_B.ADERINTCR 8743 8020h, ADC_B_NS.ADERINTCR 0743 8020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEIE2	ADEIE1	ADEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADEIE0	ADC0 A/D Conversion Error Interrupt Enable	0: Disable A/D conversion error interrupt for ADC0 1: Enable A/D conversion error interrupt for ADC0	R/W
b1	ADEIE1	ADC1 A/D Conversion Error Interrupt Enable	0: Disable A/D conversion error interrupt for ADC1 1: Enable A/D conversion error interrupt for ADC1	R/W
b2	ADEIE2	ADC2 A/D Conversion Error Interrupt Enable	0: Disable A/D conversion error interrupt for ADC2 1: Enable A/D conversion error interrupt for ADC2	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADERINTCR register controls the enable/disable to the A/D conversion error interrupt.

60.2.9 A/D Conversion Overflow Interrupt Enable Register (ADOVFINTCR)

Address(es): ADC_B.ADOVFINTCR 8743 8024h, ADC_B_NS.ADOVFINTCR 0743 8024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFIE2	ADOVFIE1	ADOVFIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADOVFIE0	ADC0 A/D Conversion Overflow Interrupt Enable	0: Disable A/D conversion overflow interrupt for ADC0 1: Enable A/D conversion overflow interrupt for ADC0	R/W
b1	ADOVFIE1	ADC1 A/D Conversion Overflow Interrupt Enable	0: Disable A/D conversion overflow interrupt for ADC1 1: Enable A/D conversion overflow interrupt for ADC1	R/W
b2	ADOVFIE2	ADC2 A/D Conversion Overflow Interrupt Enable	0: Disable A/D conversion overflow interrupt for ADC2 1: Enable A/D conversion overflow interrupt for ADC2	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADOVFINTCR register controls the enable/disable to the A/D conversion overflow interrupt.

60.2.10 Calibration interrupt Enable Register (ADCALINTCR)

Address(es): ADC_B.ADCALINTCR 8743 8028h, ADC_B_NS.ADCALINTCR 0743 8028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDIE2	CALENDIE1	CALENDIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	CALENDIE0	ADC0 Calibration End Interrupt Enable	0: Disable Calibration End Interrupt for ADC0 1: Enable Calibration End Interrupt for ADC0	R/W
b17	CALENDIE1	ADC1 Calibration End Interrupt Enable	0: Disable Calibration End Interrupt for ADC1 1: Enable Calibration End Interrupt for ADC1	R/W
b18	CALENDIE2	ADC2 Calibration End Interrupt Enable	0: Disable Calibration End Interrupt for ADC2 1: Enable Calibration End Interrupt for ADC2	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCALINTCR register controls the enable/disable to the Calibration End Interrupt.

60.2.11 A/D Converter Mode Selection Register (ADMDR)

Address(es): ADC_B.ADMDR 8743 8040h, ADC_B_NS.ADMDR 0743 8040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	ADMD2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADMD1[3:0]				—	—	—	—	ADMD0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADMD0[3:0]	ADC0 Mode Selection	b3 b0 0 0 0 0: SAR mode – Single scan mode 0 0 0 1: SAR mode – Continuous scan mode 0 1 0 0: Oversampling mode – Single scan mode 0 1 0 1: Oversampling mode – Continuous scan mode 0 1 1 0: Oversampling mode – One-channel continuous scan mode 1 0 0 0: Hybrid mode – Single scan mode 1 0 0 1: Hybrid mode – Continuous scan mode 1 0 1 0: Hybrid mode – Background continuous scan mode 1 0 1 1: Hybrid mode – Fixed channel continuous scan mode Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	ADMD1[3:0]	ADC1 Mode Selection	b11 b8 0 0 0 0: SAR mode – Single scan mode 0 0 0 1: SAR mode – Continuous scan mode 0 1 0 0: Oversampling mode – Single scan mode 0 1 0 1: Oversampling mode – Continuous scan mode 0 1 1 0: Oversampling mode – One-channel continuous scan mode 1 0 0 0: Hybrid mode – Single scan mode 1 0 0 1: Hybrid mode – Continuous scan mode 1 0 1 0: Hybrid mode – Background continuous scan mode 1 0 1 1: Hybrid mode – Fixed channel continuous scan mode Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	ADMD2[3:0]	ADC2 Mode Selection	b19 b16 0 0 0 0: SAR mode – Single scan mode 0 0 0 1: SAR mode – Continuous scan mode 0 1 0 0: Oversampling mode – Single scan mode 0 1 0 1: Oversampling mode – Continuous scan mode 0 1 1 0: Oversampling mode – One-channel continuous scan mode 1 0 0 0: Hybrid mode – Single scan mode 1 0 0 1: Hybrid mode – Continuous scan mode 1 0 1 0: Hybrid mode – Background continuous scan mode 1 0 1 1: Hybrid mode – Fixed channel continuous scan mode Settings other than above are prohibited.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADMDR register selects the operation mode and the scan mode for each A/D converter.

60.2.12 A/D Group scan Priority Control Register (ADGSPCR)

Address(es): ADC_B.ADGSPCR 8743 8044h, ADC_B_NS.ADGSPCR 0743 8044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	GRP2	LGRRS2	RSCN2	PGS2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	GRP1	LGRRS1	RSCN1	PGS1	—	—	—	—	GRP0	LGRRS0	RSCN0	PGS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS0	ADC0 Group Priority Control Setting	0: ADC0 operation is without group priority control 1: ADC0 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
b1	RSCN0	ADC0 Group Priority Control Setting 2	0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
b2	LGRRS0	ADC0 Group Priority Control Setting 3	0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
b3	GRP0	ADC0 Group Priority Control Setting 4	0: Set the following case: • When PGS0 is set to 0 • When PGS0 is set to 1 and ADC0 is SAR mode – Single scan mode. 1: Set when PGS0 is set to 1 and ADC0 is SAR mode – Continuous scan mode.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PGS1	ADC1 Group Priority Control Setting	0: ADC1 operation is without group priority control 1: ADC1 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
b9	RSCN1	ADC1 Group Priority Control Setting 2	0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W
b10	LGRRS1	ADC1 Group Priority Control Setting 3	0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W
b11	GRP1	ADC1 Group Priority Control Setting 4	0: Set the following case: • When PGS1 is set to 0 • When PGS1 is set to 1 and ADC1 is SAR mode – Single scan mode. 1: Set when PGS1 is set to 1 and ADC1 is SAR mode – Continuous scan mode.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PGS2	ADC2 Group Priority Control Setting	0: ADC2 operation is without group priority control 1: ADC2 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
b17	RSCN2	ADC2 Group Priority Control Setting 2	0: Set when PGS2 is set to 0. 1: Set when PGS2 is set to 1.	R/W
b18	LGRRS2	ADC2 Group Priority Control Setting 3	0: Set when PGS2 is set to 0. 1: Set when PGS2 is set to 1.	R/W
b19	GRP2	ADC2 Group Priority Control Setting 4	0: Set the following case: • When PGS2 is set to 0 • When PGS2 is set to 1 and ADC2 is SAR mode – Single scan mode. 1: Set when PGS2 is set to 1 and ADC2 is SAR mode – Continuous scan mode.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADGSPCR register controls the group priority operation to each A/D converter. For details on setting this register, see section 60.3.19, Group Priority Operation.

60.2.13 Scan Group Enable Register (ADSGER)

Address(es): ADC_B.ADSGER 8743 8048h, ADC_B_NS.ADSGER 0743 8048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SGRE11	SGRE10	SGRE9	SGRE8	SGRE7	SGRE6	SGRE5	SGRE4	SGRE3	SGRE2	SGRE1	SGRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	SGREn	Scan Group n Enable	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the scan group n 1: Enable the scan group n	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADSGER register selects the enable/disable to each scan group.

SGREn bit (Scan Group n Enable) (n = 0 to 11)

The SGREn bit selects the enable/disable to the scan group n. When SGREn is set to 1, A/D conversion with the scan group n is available. When SGREn is set to 0, A/D conversion with the scan group n is not available, even if the trigger corresponding to the scan group n is input.

60.2.14 Scan Group Control Register 0 (ADSGCR0)

Address(es): ADC_B.ADSGCR0 8743 804Ch, ADC_B_NS.ADSGCR0 0743 804Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SGADS3[1:0]		—	—	—	—	—	—	SGADS2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SGADS1[1:0]		—	—	—	—	—	—	SGADS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SGADS0[1:0]	Scan Group 0 A/D Converter Selection	b1 b0 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SGADS1[1:0]	Scan Group 1 A/D Converter Selection	b9 b8 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SGADS2[1:0]	Scan Group 2 A/D Converter Selection	b17 b16 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	SGADS3[1:0]	Scan Group 3 A/D Converter Selection	b25 b24 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR0 register selects which A/D converter is used for each scan group.

60.2.15 Scan Group Control Register 1 (ADSGCR1)

Address(es): ADC_B.ADSGCR1 8743 8050h, ADC_B_NS.ADSGCR1 0743 8050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SGADS7[1:0]		—	—	—	—	—	—	SGADS6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SGADS5[1:0]		—	—	—	—	—	—	SGADS4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SGADS4[1:0]	Scan Group 4 A/D Converter Selection	b1 b0 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SGADS5[1:0]	Scan Group 5 A/D Converter Selection	b9 b8 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SGADS6[1:0]	Scan Group 6 A/D Converter Selection	b17 b16 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	SGADS7[1:0]	Scan Group 7 A/D Converter Selection	b25 b24 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR1 register selects which A/D converter is used for each scan group.

60.2.16 Scan Group Control Register 2 (ADSGCR2)

Address(es): ADC_B.ADSGCR2 8743 8054h, ADC_B_NS.ADSGCR2 0743 8054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SGADS11[1:0]		—	—	—	—	—	—	SGADS10[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SGADS9[1:0]		—	—	—	—	—	—	SGADS8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SGADS8[1:0]	Scan Group 8 A/D Converter Selection	b1 b0 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SGADS9[1:0]	Scan Group 9 A/D Converter Selection	b9 b8 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SGADS10[1:0]	Scan Group 10 A/D Converter Selection	b17 b16 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	SGADS11[1:0]	Scan Group 11 A/D Converter Selection	b25 b24 0 0: Select ADC0 0 1: Select ADC1 1 0: Select ADC2 1 1: Setting prohibited	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR2 register selects which A/D converter is used for each scan group.

60.2.17 Scan End Interrupt Enable Register (ADINTCR)

Address(es): ADC_B.ADINTCR 8743 805Ch, ADC_B_NS.ADINTCR 0743 805Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADIE11	ADIE10	ADIE9	ADIE8	ADIE7	ADIE6	ADIE5	ADIE4	ADIE3	ADIE2	ADIE1	ADIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	ADIE0 to ADIE11	Scan Group n Scan End Interrupt Enable	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n scan end interrupt 1: Enable scan group n scan end interrupt	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADINTCR register enables/disables the scan end interrupt.

60.2.18 External Trigger Enable Register n (ADTRGEXTn) (n = 0 to 11)

Address(es): ADC_B.ADTRGEXT0 8743 80C0h, ADC_B.ADTRGEXT1 8743 80D0h, ADC_B.ADTRGEXT2 8743 80E0h, ADC_B.ADTRGEXT3 8743 80F0h, ADC_B.ADTRGEXT4 8743 8100h, ADC_B.ADTRGEXT5 8743 8110h, ADC_B.ADTRGEXT6 8743 8120h, ADC_B.ADTRGEXT7 8743 8130h, ADC_B.ADTRGEXT8 8743 8140h, ADC_B.ADTRGEXT9 8743 8150h, ADC_B.ADTRGEXT10 8743 8160h, ADC_B.ADTRGEXT11 8743 8170h, ADC_B_NS.ADTRGEXT0 0743 80C0h, ADC_B_NS.ADTRGEXT1 0743 80D0h, ADC_B_NS.ADTRGEXT2 0743 80E0h, ADC_B_NS.ADTRGEXT3 0743 80F0h, ADC_B_NS.ADTRGEXT4 0743 8100h, ADC_B_NS.ADTRGEXT5 0743 8110h, ADC_B_NS.ADTRGEXT6 0743 8120h, ADC_B_NS.ADTRGEXT7 0743 8130h, ADC_B_NS.ADTRGEXT8 0743 8140h, ADC_B_NS.ADTRGEXT9 0743 8150h, ADC_B_NS.ADTRGEXT10 0743 8160h, ADC_B_NS.ADTRGEXT11 0743 8170h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGEXT2	TRGEXT1	TRGEXT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRGEXT0	External Trigger Input 0 (ADTRG0) Enable	0: Disable ADTRG0 1: Enable ADTRG0	R/W
b1	TRGEXT1	External Trigger Input 1 (ADTRG1) Enable	0: Disable ADTRG1 1: Enable ADTRG1	R/W
b2	TRGEXT2	External Trigger Input 2 (ADTRG2) Enable	0: Disable ADTRG2 1: Enable ADTRG2	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGEXTn register enables/disables the External Trigger input 0, 1, and 2 (ADTRG0, ADTRG1, and ADTRG2) as the starting conditions for A/D conversion of scan group n.

60.2.19 ELC Trigger Enable Register n (ADTRGELCn) (n = 0 to 11)

Address(es): ADC_B.ADTRGELC0 8743 80C4h, ADC_B.ADTRGELC1 8743 80D4h, ADC_B.ADTRGELC2 8743 80E4h, ADC_B.ADTRGELC3 8743 80F4h, ADC_B.ADTRGELC4 8743 8104h, ADC_B.ADTRGELC5 8743 8114h, ADC_B.ADTRGELC6 8743 8124h, ADC_B.ADTRGELC7 8743 8134h, ADC_B.ADTRGELC8 8743 8144h, ADC_B.ADTRGELC9 8743 8154h, ADC_B.ADTRGELC10 8743 8164h, ADC_B.ADTRGELC11 8743 8174h, ADC_B_NS.ADTRGELC0 0743 80C4h, ADC_B_NS.ADTRGELC1 0743 80D4h, ADC_B_NS.ADTRGELC2 0743 80E4h, ADC_B_NS.ADTRGELC3 0743 80F4h, ADC_B_NS.ADTRGELC4 0743 8104h, ADC_B_NS.ADTRGELC5 0743 8114h, ADC_B_NS.ADTRGELC6 0743 8124h, ADC_B_NS.ADTRGELC7 0743 8134h, ADC_B_NS.ADTRGELC8 0743 8144h, ADC_B_NS.ADTRGELC9 0743 8154h, ADC_B_NS.ADTRGELC10 0743 8164h, ADC_B_NS.ADTRGELC11 0743 8174h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRGELC8	TRGELC7	TRGELC6	TRGELC5	TRGELC4	TRGELC3	TRGELC2	TRGELC1	TRGELC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	TRGELC0 to TRGELC8	ELC Trigger m Enable	The suffix number of each bit symbol corresponds to the ELC trigger number m. 0: Disable ELC Trigger m 1: Enable ELC Trigger m	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = 0 to 8

The ADTRGELCn register enables/disables the ELC Trigger m as the starting conditions for A/D conversion of scan group n.

60.2.20 GPT Trigger Enable Register n (ADTRGGPTn) (n = 0 to 11)

Address(es): ADC_B.ADTRGGPT0 8743 80C8h, ADC_B.ADTRGGPT1 8743 80D8h, ADC_B.ADTRGGPT2 8743 80E8h, ADC_B.ADTRGGPT3 8743 80F8h, ADC_B.ADTRGGPT4 8743 8108h, ADC_B.ADTRGGPT5 8743 8118h, ADC_B.ADTRGGPT6 8743 8128h, ADC_B.ADTRGGPT7 8743 8138h, ADC_B.ADTRGGPT8 8743 8148h, ADC_B.ADTRGGPT9 8743 8158h, ADC_B.ADTRGGPT10 8743 8168h, ADC_B.ADTRGGPT11 8743 8178h, ADC_B_NS.ADTRGGPT0 0743 80C8h, ADC_B_NS.ADTRGGPT1 0743 80D8h, ADC_B_NS.ADTRGGPT2 0743 80E8h, ADC_B_NS.ADTRGGPT3 0743 80F8h, ADC_B_NS.ADTRGGPT4 0743 8108h, ADC_B_NS.ADTRGGPT5 0743 8118h, ADC_B_NS.ADTRGGPT6 0743 8128h, ADC_B_NS.ADTRGGPT7 0743 8138h, ADC_B_NS.ADTRGGPT8 0743 8148h, ADC_B_NS.ADTRGGPT9 0743 8158h, ADC_B_NS.ADTRGGPT10 0743 8168h, ADC_B_NS.ADTRGGPT11 0743 8178h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TRGGPTB15	TRGGPTB14	TRGGPTB13	TRGGPTB12	TRGGPTB11	TRGGPTB10	TRGGPTB9	TRGGPTB8	TRGGPTB7	TRGGPTB6	TRGGPTB5	TRGGPTB4	TRGGPTB3	TRGGPTB2	TRGGPTB1	TRGGPTB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRGGPTA15	TRGGPTA14	TRGGPTA13	TRGGPTA12	TRGGPTA11	TRGGPTA10	TRGGPTA9	TRGGPTA8	TRGGPTA7	TRGGPTA6	TRGGPTA5	TRGGPTA4	TRGGPTA3	TRGGPTA2	TRGGPTA1	TRGGPTA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRGGPTA0 to TRGGPTA15	GPT Channel m A/D Conversion Starting Request A Enable	The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request A from GPT channel m 1: Enable the A/D conversion stating request A from GPT channel m	R/W
b31 to b16	TRGGPTB0 to TRGGPTB15	GPT Channel m A/D Conversion Starting Request B Enable	The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request B from GPT channel m 1: Enable the A/D conversion stating request B from GPT channel m	R/W

m = 0 to 15

The ADTRGGPTn register enables/disables the A/D conversion starting request A/B from GPT channel m as the starting conditions for A/D conversion of scan group n.

60.2.21 A/D Conversion Start Trigger Enable Register (ADTRGENR)

Address(es): ADC_B.ADTRGENR 8743 8C08h, ADC_B_NS.ADTRGENR 0743 8C08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STTRGEN11	STTRGEN10	STTRGEN9	STTRGEN8	STTRGEN7	STTRGEN6	STTRGEN5	STTRGEN4	STTRGEN3	STTRGEN2	STTRGEN1	STTRGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	STTRGEN0 to STTRGEN11	Scan Group n A/D Conversion Start Trigger Enable	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the A/D conversion start trigger 1: Enable the A/D conversion start trigger	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADTRGENR register enables/disables the trigger from the peripheral modules for starting the A/D conversion of Scan group n. The triggers for each scan group are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers.

STTRGENn Bit (Scan Group n A/D Conversion Start Trigger Enable)

The STTRGENn bit selects whether the trigger inputs from the peripheral module, which are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers, is used for the A/D conversion start condition.
When the STTRGENn bit is set to 1, the A/D conversion of scan group n can be started by trigger inputs from the peripheral module.
When the STTRGENn bit is set to 0, the A/D conversion of scan group n cannot be started by trigger inputs from the peripheral module. In this setting, the A/D conversion of scan group n can be only started by writing to the ADSYSTR or ADSTRn registers.

60.2.22 A/D Conversion Start Trigger Delay Register 0 (ADTRGDLR0)

Address(es): ADC_B.ADTRGDLR0 8743 81C0h, ADC_B_NS.ADTRGDLR0 0743 81C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY0[7:0]	Scan Group 0	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY1[7:0]	Scan Group 1	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR0 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.23 A/D Conversion Start Trigger Delay Register 1 (ADTRGDLR1)

Address(es): ADC_B.ADTRGDLR1 8743 81C4h, ADC_B_NS.ADTRGDLR1 0743 81C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY3[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY2[7:0]	Scan Group 2	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY3[7:0]	Scan Group 3	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR1 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.24 A/D Conversion Start Trigger Delay Register 2 (ADTRGDLR2)

Address(es): ADC_B.ADTRGDLR2 8743 81C8h, ADC_B_NS.ADTRGDLR2 0743 81C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY5[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY4[7:0]	Scan Group 4	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY5[7:0]	Scan Group 5	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR2 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.25 A/D Conversion Start Trigger Delay Register 3 (ADTRGDLR3)

Address(es): ADC_B.ADTRGDLR3 8743 81CCh, ADC_B_NS.ADTRGDLR3 0743 81CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY7[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY6[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY6[7:0]	Scan Group 6	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY7[7:0]	Scan Group 7	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR3 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.26 A/D Conversion Start Trigger Delay Register 4 (ADTRGDLR4)

Address(es): ADC_B.ADTRGDLR4 8743 81D0h, ADC_B_NS.ADTRGDLR4 0743 81D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY9[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY8[7:0]	Scan Group 8	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY9[7:0]	Scan Group 9	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR4 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.27 A/D Conversion Start Trigger Delay Register 5 (ADTRGDLR5)

Address(es): ADC_B.ADTRGDLR5 8743 81D4h, ADC_B_NS.ADTRGDLR5 0743 81D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	TRGDLY11[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRGDLY10[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TRGDLY10[7:0]	Scan Group 10	Trigger Input Delay Configuration	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TRGDLY11[7:0]	Scan Group 11	Trigger Input Delay Configuration	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR5 register controls the input delay added to the external trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

60.2.28 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 0 (ADSWNR0)

Address(es): ADC_B.ADSWNR0 8743 8060h, ADC_B_NS.ADSWNR0 0743 8060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SWNOFIX0[2:0]			—	—	—	—	—	—	SWFIX0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BCSSTSL0[3:0]				—	—	—	—	—	SWNUM0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SWNUM0[2:0]	ADC0 Fixed Channel Mode Scan Cycle Setting (Number of Conversions)	b2 b0 0 0 0: 1-time 0 0 1: 2-time 0 1 0: 3-time 0 1 1: 4-time : : 1 1 1: 8-time	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	BCSSTSL0[3:0]	Sampling State Setting Table Selection for Vacant Channels	b11 b8 0 0 0 0: Sampling state table 0 (Select the ADSSTR0.SST0[9:0]) 0 0 0 1: Sampling state table 1 (Select the ADSSTR0.SST1[9:0]) 0 0 1 0: Sampling state table 2 (Select the ADSSTR1.SST2[9:0]) 0 0 1 1: Sampling state table 3 (Select the ADSSTR1.SST3[9:0]) : : 1 1 1 1: Sampling state table 15 (Select the ADSSTR7.SST15[9:0])	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SWFIX0[1:0]	ADC0 Fixed Channel Count Setting	b17 b16 0 0: 1 Fixed channel 0 1: 2 Fixed channels 1 0: 3 Fixed channels 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	SWNOFIX0[2:0]	ADC0 Non-Fixed Channel Count Setting	b26 b24 0 0 0: 1 channel Others: Setting prohibited	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSWNR0 register sets the A/D conversion period for the fixed channel scan of A/D converter Unit 0.
The ADSWNR0 register is valid only when ADC0 is set to hybrid mode (fixed channel continuous scan mode).

SWNUM0[2:0] Bits (ADC0 Fixed Channel Mode Scan Cycle Setting (Number of Conversions))

The SWNUM0[2:0] bits specify the A/D conversion cycle for fixed channel scan of A/D converter Unit 0.
The A/D conversion cycle is the value set by SWNUM0[2:0] + 1.

BCSSTSL0[3:0] Bits (Sampling State Setting Table Selection for Vacant Channels)

The BCSSTSL0[3:0] bits set the number of sampling states on blank channels that do not get the A/D conversion data.

Although analog input sampling is not performed on unused channels, a sampling time setting table is selected to set the number of channel states in the same way as for virtual channels. The sampling state value of the selected table is used.

SWFIX0[1:0] Bits (ADC0 Fixed Channel Count Setting)

The SWFIX0[1:0] bits set the number of fixed channels that are converted channels sweeping through each scan.

SWNOFIX0[2:0] Bits (ADC0 Non-Fixed Channel Count Setting)

The SWNOFIX0[2:0] bits set the number of non-fixed channels that are converted while switching channels in each scan.

60.2.29 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 1 (ADSWNR1)

Address(es): ADC_B.ADSWNR1 8743 8064h, ADC_B_NS.ADSWNR1 0743 8064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SWNOFIX1[2:0]			—	—	—	—	—	—	SWFIX1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BCSSTSL1[3:0]				—	—	—	—	—	SWNUM1[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SWNUM1[2:0]	ADC1 Fixed Channel Mode Scan Cycle Setting (Number of Conversions)	b2 b0 0 0 0: 1-time 0 0 1: 2-time 0 1 0: 3-time 0 1 1: 4-time : : 1 1 1: 8-time	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	BCSSTSL1[3:0]	Sampling State Setting Table Selection for Vacant Channels	b11 b8 0 0 0 0: Sampling state table 0 (Select the ADSSTR0.SST0[9:0]) 0 0 0 1: Sampling state table 1 (Select the ADSSTR0.SST1[9:0]) 0 0 1 0: Sampling state table 2 (Select the ADSSTR1.SST2[9:0]) 0 0 1 1: Sampling state table 3 (Select the ADSSTR1.SST3[9:0]) : : 1 1 1 1: Sampling state table 15 (Select the ADSSTR7.SST15[9:0])	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SWFIX1[1:0]	ADC1 Fixed Channel Count Setting	b17 b16 0 0: 1 Fixed channel 0 1: 2 Fixed channels 1 0: 3 Fixed channels 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	SWNOFIX1[2:0]	ADC1 Non-Fixed Channel Count Setting	b26 b24 0 0 0: 1 channel Others: Setting prohibited	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSWNR1 register sets the A/D conversion period for the fixed channel scan of A/D converter Unit 1.
The ADSWNR1 register is valid only when ADC1 is set to hybrid mode (fixed channel continuous scan mode).

SWNUM1[2:0] Bits (ADC1 Fixed Channel Mode Scan Cycle Setting (Number of Conversions))

The SWNUM1[2:0] bits specify the A/D conversion cycle for fixed channel scan of A/D converter Unit 1.
The A/D conversion cycle is the value set by SWNUM1[2:0] + 1.

BCSSTSL1[3:0] Bits (Sampling State Setting Table Selection for Vacant Channels)

The BCSSTSL1[3:0] bits set the number of sampling states on blank channels that do not get the A/D conversion data. Although analog input sampling is not performed on unused channels, a sampling time setting table is selected to set the number of channel states in the same way as for virtual channels. The sampling state value of the selected table is used.

SWFIX1[1:0] Bits (ADC1 Fixed Channel Count Setting)

The SWFIX1[1:0] bits set the number of fixed channels that are converted channels sweeping through each scan.

SWNOFIX1[2:0] Bits (ADC1 Non-Fixed Channel Count Setting)

The SWNOFIX1[2:0] bits set the number of non-fixed channels that are converted while switching channels in each scan.

60.2.30 Hybrid Mode (Fixed Channel Continuous Scan Mode) Conversion Count per Scan Register 2 (ADSWNR2)

Address(es): ADC_B.ADSWNR2 8743 8068h, ADC_B_NS.ADSWNR2 0743 8068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SWNOFIX2[2:0]			—	—	—	—	—	—	SWFIX2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BCSSTSL2[3:0]				—	—	—	—	—	SWNUM2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SWNUM2[2:0]	ADC2 Fixed Channel Mode Scan Cycle Setting (Number of Conversions)	b2 b0 0 0 0: 1-time 0 0 1: 2-time 0 1 0: 3-time 0 1 1: 4-time : : 1 1 1: 8-time	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	BCSSTSL2[3:0]	Sampling State Setting Table Selection for Vacant Channels	b11 b8 0 0 0 0: Sampling state table 0 (Select the ADSSTR0.SST0[9:0]) 0 0 0 1: Sampling state table 1 (Select the ADSSTR0.SST1[9:0]) 0 0 1 0: Sampling state table 2 (Select the ADSSTR1.SST2[9:0]) 0 0 1 1: Sampling state table 3 (Select the ADSSTR1.SST3[9:0]) : : 1 1 1 1: Sampling state table 15 (Select the ADSSTR7.SST15[9:0])	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SWFIX2[1:0]	ADC2 Fixed Channel Count Setting	b17 b16 0 0: 1 Fixed channel 0 1: 2 Fixed channels 1 0: 3 Fixed channels 1 1: Setting prohibited	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	SWNOFIX2[2:0]	ADC2 Non-Fixed Channel Count Setting	b26 b24 0 0 0: 1 channel Others: Setting prohibited	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSWNR2 register sets the A/D conversion period for the fixed channel scan of A/D converter Unit 2.
The ADSWNR2 register is valid only when ADC2 is set to hybrid mode (fixed channel continuous scan mode).

SWNUM2[2:0] Bits (ADC2 Fixed Channel Mode Scan Cycle Setting (Number of Conversions))

The SWNUM2[2:0] bits specify the A/D conversion cycle for fixed channel scan of A/D converter Unit 2.
The A/D conversion cycle is the value set by SWNUM2[2:0] + 1.

BCSSTSL2[3:0] Bits (Sampling State Setting Table Selection for Vacant Channels)

The BCSSTSL2[3:0] bits set the number of sampling states on blank channels that do not get the A/D conversion data.

Although analog input sampling is not performed on unused channels, a sampling time setting table is selected to set the number of channel states in the same way as for virtual channels. The sampling state value of the selected table is used.

SWFIX2[1:0] Bits (ADC2 Fixed Channel Count Setting)

The SWFIX2[1:0] bits set the number of fixed channels that are converted channels sweeping through each scan.

SWNOFIX2[2:0] Bits (ADC2 Non-Fixed Channel Count Setting)

The SWNOFIX2[2:0] bits set the number of non-fixed channels that are converted while switching channels in each scan.

60.2.31 Oversampling Mode (Same Channel Continuous Scan Mode) Decimation Control Register (ADDECCR)

Address(es): ADC_B.ADDECCR 8743 8080h, ADC_B_NS.ADDECCR 0743 8080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	DCIM2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DCIM1[4:0]					—	—	—	DCIM0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DCIM0[4:0]	ADC0 Output Decimation Constant	b4 b0 0 0 0 0 0: No decimation 0 0 0 1 1: Output data once for every 4 conversions (thinned 3 times) 0 0 1 1 1: Output data once for every 8 conversions (thinned 7 times) Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	DCIM1[4:0]	ADC1 Output Decimation Constant	b12 b8 0 0 0 0 0: No decimation 0 0 0 1 1: Output data once for every 4 conversions (thinned 3 times) 0 0 1 1 1: Output data once for every 8 conversions (thinned 7 times) Settings other than above are prohibited.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	DCIM2[4:0]	ADC2 Output Decimation Constant	b20 b16 0 0 0 0 0: No decimation 0 0 0 1 1: Output data once for every 4 conversions (thinned 3 times) 0 0 1 1 1: Output data once for every 8 conversions (thinned 7 times) Settings other than above are prohibited.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDECCR register controls the data output decimation of each A/D converter. The ADDECCR register is only valid when the A/D converter unit is set to oversampling mode (same channel continuous scan mode).

60.2.32 A/D Converter Accuracy Mode Register (ADACMDR)

Address(es): ADC_B.ADACMDR 8743 8084h, ADC_B_NS.ADACMDR 0743 8084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADHACMD2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADHACMD1	—	—	—	—	—	—	—	ADHACMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADHACMD0	ADC0 High Accuracy Mode	0: High accuracy mode of A/D converter is disabled 1: High accuracy mode of A/D converter is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ADHACMD1	ADC1 High Accuracy Mode	0: High accuracy mode of A/D converter is disabled 1: High accuracy mode of A/D converter is enabled	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ADHACMD2	ADC2 High Accuracy Mode	0: High accuracy mode of A/D converter is disabled 1: High accuracy mode of A/D converter is enabled	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADACMDR register controls the conversion precision mode for each A/D converter.

ADHACMDm Bit (ADCm High Accuracy Mode) (m = 0 to 2)

The ADHACMDm bit sets the accuracy of ADCm.
Setting the ADHACMDm bit to 1 enables A/D conversion in high precision mode.
Enabling high precision mode allows you to obtain A/D conversion results with higher accuracy than when high precision mode is disabled.
For details about accuracy, see section 67, Electrical Characteristics.

Enabling high precision mode adds setting restrictions to some registers.
For details, see section 60.10.17, Restrictions on the High Accuracy Mode.

60.2.33 A/D Conversion Channel Configuration Register n (ADCHCRn) (n = 0 to 49)

Address(es): ADC_B.ADCHCR0 8743 8600h, ADC_B.ADCHCR1 8743 8610h, ADC_B.ADCHCR2 8743 8620h, ADC_B.ADCHCR3 8743 8630h, ADC_B.ADCHCR4 8743 8640h, ADC_B.ADCHCR5 8743 8650h, ADC_B.ADCHCR6 8743 8660h, ADC_B.ADCHCR7 8743 8670h, ADC_B.ADCHCR8 8743 8680h, ADC_B.ADCHCR9 8743 8690h, ADC_B.ADCHCR10 8743 86A0h, ADC_B.ADCHCR11 8743 86B0h, ADC_B.ADCHCR12 8743 86C0h, ADC_B.ADCHCR13 8743 86D0h, ADC_B.ADCHCR14 8743 86E0h, ADC_B.ADCHCR15 8743 86F0h, ADC_B.ADCHCR16 8743 8700h, ADC_B.ADCHCR17 8743 8710h, ADC_B.ADCHCR18 8743 8720h, ADC_B.ADCHCR19 8743 8730h, ADC_B.ADCHCR20 8743 8740h, ADC_B.ADCHCR21 8743 8750h, ADC_B.ADCHCR22 8743 8760h, ADC_B.ADCHCR23 8743 8770h, ADC_B.ADCHCR24 8743 8780h, ADC_B.ADCHCR25 8743 8790h, ADC_B.ADCHCR26 8743 87A0h, ADC_B.ADCHCR27 8743 87B0h, ADC_B.ADCHCR28 8743 87C0h, ADC_B.ADCHCR29 8743 87D0h, ADC_B.ADCHCR30 8743 87E0h, ADC_B.ADCHCR31 8743 87F0h, ADC_B.ADCHCR32 8743 8800h, ADC_B.ADCHCR33 8743 8810h, ADC_B.ADCHCR34 8743 8820h, ADC_B.ADCHCR35 8743 8830h, ADC_B.ADCHCR36 8743 8840h, ADC_B.ADCHCR37 8743 8850h, ADC_B.ADCHCR38 8743 8860h, ADC_B.ADCHCR39 8743 8870h, ADC_B.ADCHCR40 8743 8880h, ADC_B.ADCHCR41 8743 8890h, ADC_B.ADCHCR42 8743 88A0h, ADC_B.ADCHCR43 8743 88B0h, ADC_B.ADCHCR44 8743 88C0h, ADC_B.ADCHCR45 8743 88D0h, ADC_B.ADCHCR46 8743 88E0h, ADC_B.ADCHCR47 8743 88F0h, ADC_B.ADCHCR48 8743 8900h, ADC_B.ADCHCR49 8743 8910h,
ADC_B_NS.ADCHCR0 0743 8600h, ADC_B_NS.ADCHCR1 0743 8610h, ADC_B_NS.ADCHCR2 0743 8620h, ADC_B_NS.ADCHCR3 0743 8630h, ADC_B_NS.ADCHCR4 0743 8640h, ADC_B_NS.ADCHCR5 0743 8650h, ADC_B_NS.ADCHCR6 0743 8660h, ADC_B_NS.ADCHCR7 0743 8670h, ADC_B_NS.ADCHCR8 0743 8680h, ADC_B_NS.ADCHCR9 0743 8690h, ADC_B_NS.ADCHCR10 0743 86A0h, ADC_B_NS.ADCHCR11 0743 86B0h, ADC_B_NS.ADCHCR12 0743 86C0h, ADC_B_NS.ADCHCR13 0743 86D0h, ADC_B_NS.ADCHCR14 0743 86E0h, ADC_B_NS.ADCHCR15 0743 86F0h, ADC_B_NS.ADCHCR16 0743 8700h, ADC_B_NS.ADCHCR17 0743 8710h, ADC_B_NS.ADCHCR18 0743 8720h, ADC_B_NS.ADCHCR19 0743 8730h, ADC_B_NS.ADCHCR20 0743 8740h, ADC_B_NS.ADCHCR21 0743 8750h, ADC_B_NS.ADCHCR22 0743 8760h, ADC_B_NS.ADCHCR23 0743 8770h, ADC_B_NS.ADCHCR24 0743 8780h, ADC_B_NS.ADCHCR25 0743 8790h, ADC_B_NS.ADCHCR26 0743 87A0h, ADC_B_NS.ADCHCR27 0743 87B0h, ADC_B_NS.ADCHCR28 0743 87C0h, ADC_B_NS.ADCHCR29 0743 87D0h, ADC_B_NS.ADCHCR30 0743 87E0h, ADC_B_NS.ADCHCR31 0743 87F0h, ADC_B_NS.ADCHCR32 0743 8800h, ADC_B_NS.ADCHCR33 0743 8810h, ADC_B_NS.ADCHCR34 0743 8820h, ADC_B_NS.ADCHCR35 0743 8830h, ADC_B_NS.ADCHCR36 0743 8840h, ADC_B_NS.ADCHCR37 0743 8850h, ADC_B_NS.ADCHCR38 0743 8860h, ADC_B_NS.ADCHCR39 0743 8870h, ADC_B_NS.ADCHCR40 0743 8880h, ADC_B_NS.ADCHCR41 0743 8890h, ADC_B_NS.ADCHCR42 0743 88A0h, ADC_B_NS.ADCHCR43 0743 88B0h, ADC_B_NS.ADCHCR44 0743 88C0h, ADC_B_NS.ADCHCR45 0743 88D0h, ADC_B_NS.ADCHCR46 0743 88E0h, ADC_B_NS.ADCHCR47 0743 88F0h, ADC_B_NS.ADCHCR48 0743 8900h, ADC_B_NS.ADCHCR49 0743 8910h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	SSTSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AINMD	CNVCS[6:0]							—	—	—	SGSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	SGSEL[4:0]	Scan Group Selection	b4 b0 0 0 0 0 0: Not select any scan group 0 0 0 0 1: Select the scan group 0 0 0 0 1 0: Select the scan group 1 0 0 0 1 1: Select the scan group 2 0 0 1 0 0: Select the scan group 3 : : 0 1 1 0 0: Select the scan group 11 Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b14 to b8	CNVCS[6:0]	A/D Conversion Channel Selection	b14 b8 0 0 0 0 0 0 0: AN000 0 0 0 0 0 0 1: AN001 : : 0 1 0 0 0 1 1: AN035 1 1 0 0 0 0 0: Self-diagnosis (ADC0)*1 1 1 0 0 0 0 1: Self-diagnosis (ADC1)*1 1 1 0 0 0 1 0: Self-diagnosis (ADC2)*1 1 1 0 0 1 0 0: Temperature sensor 1 1 0 0 1 0 1: Internal reference voltage 1 1 0 0 1 1 0: VBATT 1/6 voltage 1 1 0 1 0 0 0: D/A converter channel 0 1 1 0 1 0 0 1: D/A converter channel 1 1 1 1 0 0 0 0: Self-diagnosis (Channel-dedicated sample-and-hold circuit unit 0)*1 1 1 1 0 0 0 1: Self-diagnosis (Channel-dedicated sample-and-hold circuit unit 1)*1 1 1 1 0 0 1 0: Self-diagnosis (Channel-dedicated sample-and-hold circuit unit 2)*1 Settings other than above are prohibited.	R/W
b15	AINMD*2	Analog Input Mode Selection*1	0: Single-ended input 1: Differential input	R/W
b19 to b16	SSTSEL[3:0]	Sampling State Table Selection	b19 b16 0 0 0 0: Select the sampling state table 0 0 0 0 1: Select the sampling state table 1 0 0 1 0: Select the sampling state table 2 0 0 1 1: Select the sampling state table 3 : : 1 1 1 1: Select the sampling state table 15	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Select differential input when self-diagnosis is selected.
Note 2. When using a channel-dedicated sample-and-hold circuit, make sure that the setting of channel-dedicated sample-and-hold circuit unit n for the channel selected by CNVCS[6:0] matches the setting of ADSHCR0.SHMDn.

The ADCHCRn register selects the Virtual channel configuration for A/D conversion.

SGSEL[4:0] Bits (Scan Group Selection)

The SGSEL[4:0] bits select which scan group to assign the selected channel in CNVCS[6:0]. If SGSEL[4:0] is set to 00h, the channel selected by CNVCS[6:0] is not available for A/D conversion. If SGSEL[4:0] is set to other than 00h, the channel selected by CNVCS[6:0] is available for A/D conversion in the scan operation of the scan group corresponding to the setting value.

Up to 8 A/D conversion channels can be assigned per scan group in the SAR mode or oversampling mode; Up to 4 A/D conversion channels can be assigned per scan group in the hybrid mode. Do not assign more than specified number of A/D conversion channels per scan group.

In fixed channel continuous scan mode, the scan group setting of the virtual channel assigned to the fixed channel (hybrid mode) must be set to no setting (00h).

CNVCS[6:0] Bits (A/D Conversion Channel Selection)

The CNVCS[6:0] bits select the analog signal source for the A/D conversion. The CNVCS[6:0] bits should be set for the channel number corresponding to the analog input or the extended analog function. For A/D conversion of the analog signal source in differential input mode, the CNVCS[6:0] bits should be set the even-numbered channel; When the odd-numbered channel is set in differential input mode, A/D conversion is not guaranteed.

To A/D convert the channel selected by the CNVCS[6:0] bits, it should be assigned to one of the scan groups by the SGSEL[4:0] bits.

AINMD Bit (Analog Input Mode Selection)

The AINMD bit selects the input method for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. When the AINMD bit is set to 0, A/D conversion is performed in single-ended input. When the AINMD bit is set to 1, A/D conversion is performed in differential input.
If the self-diagnosis channel is selected with the CNVCS[6:0] bits, a differential input should be selected.

SSTSEL[3:0] Bits (Sampling State Table Selection)

The SSTSEL[3:0] bits select the sampling state table for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. The number of sampling states is the value set in the selected sampling state table.

60.2.34 A/D Conversion Data Operation Control A Register n (ADDOPCRAn) (n = 0 to 49)

Address(es): ADC_B.ADDOPCRA0 8743 8604h, ADC_B.ADDOPCRA1 8743 8614h, ADC_B.ADDOPCRA2 8743 8624h, ADC_B.ADDOPCRA3 8743 8634h, ADC_B.ADDOPCRA4 8743 8644h, ADC_B.ADDOPCRA5 8743 8654h, ADC_B.ADDOPCRA6 8743 8664h, ADC_B.ADDOPCRA7 8743 8674h, ADC_B.ADDOPCRA8 8743 8684h, ADC_B.ADDOPCRA9 8743 8694h, ADC_B.ADDOPCRA10 8743 86A4h, ADC_B.ADDOPCRA11 8743 86B4h, ADC_B.ADDOPCRA12 8743 86C4h, ADC_B.ADDOPCRA13 8743 86D4h, ADC_B.ADDOPCRA14 8743 86E4h, ADC_B.ADDOPCRA15 8743 86F4h, ADC_B.ADDOPCRA16 8743 8704h, ADC_B.ADDOPCRA17 8743 8714h, ADC_B.ADDOPCRA18 8743 8724h, ADC_B.ADDOPCRA19 8743 8734h, ADC_B.ADDOPCRA20 8743 8744h, ADC_B.ADDOPCRA21 8743 8754h, ADC_B.ADDOPCRA22 8743 8764h, ADC_B.ADDOPCRA23 8743 8774h, ADC_B.ADDOPCRA24 8743 8784h, ADC_B.ADDOPCRA25 8743 8794h, ADC_B.ADDOPCRA26 8743 87A4h, ADC_B.ADDOPCRA27 8743 87B4h, ADC_B.ADDOPCRA28 8743 87C4h, ADC_B.ADDOPCRA29 8743 87D4h, ADC_B.ADDOPCRA30 8743 87E4h, ADC_B.ADDOPCRA31 8743 87F4h, ADC_B.ADDOPCRA32 8743 8804h, ADC_B.ADDOPCRA33 8743 8814h, ADC_B.ADDOPCRA34 8743 8824h, ADC_B.ADDOPCRA35 8743 8834h, ADC_B.ADDOPCRA36 8743 8844h, ADC_B.ADDOPCRA37 8743 8854h, ADC_B.ADDOPCRA38 8743 8864h, ADC_B.ADDOPCRA39 8743 8874h, ADC_B.ADDOPCRA40 8743 8884h, ADC_B.ADDOPCRA41 8743 8894h, ADC_B.ADDOPCRA42 8743 88A4h, ADC_B.ADDOPCRA43 8743 88B4h, ADC_B.ADDOPCRA44 8743 88C4h, ADC_B.ADDOPCRA45 8743 88D4h, ADC_B.ADDOPCRA46 8743 88E4h, ADC_B.ADDOPCRA47 8743 88F4h, ADC_B.ADDOPCRA48 8743 8904h, ADC_B.ADDOPCRA49 8743 8914h,
ADC_B_NS.ADDOPCRA0 0743 8604h, ADC_B_NS.ADDOPCRA1 0743 8614h, ADC_B_NS.ADDOPCRA2 0743 8624h, ADC_B_NS.ADDOPCRA3 0743 8634h, ADC_B_NS.ADDOPCRA4 0743 8644h, ADC_B_NS.ADDOPCRA5 0743 8654h, ADC_B_NS.ADDOPCRA6 0743 8664h, ADC_B_NS.ADDOPCRA7 0743 8674h, ADC_B_NS.ADDOPCRA8 0743 8684h, ADC_B_NS.ADDOPCRA9 0743 8694h, ADC_B_NS.ADDOPCRA10 0743 86A4h, ADC_B_NS.ADDOPCRA11 0743 86B4h, ADC_B_NS.ADDOPCRA12 0743 86C4h, ADC_B_NS.ADDOPCRA13 0743 86D4h, ADC_B_NS.ADDOPCRA14 0743 86E4h, ADC_B_NS.ADDOPCRA15 0743 86F4h, ADC_B_NS.ADDOPCRA16 0743 8704h, ADC_B_NS.ADDOPCRA17 0743 8714h, ADC_B_NS.ADDOPCRA18 0743 8724h, ADC_B_NS.ADDOPCRA19 0743 8734h, ADC_B_NS.ADDOPCRA20 0743 8744h, ADC_B_NS.ADDOPCRA21 0743 8754h, ADC_B_NS.ADDOPCRA22 0743 8764h, ADC_B_NS.ADDOPCRA23 0743 8774h, ADC_B_NS.ADDOPCRA24 0743 8784h, ADC_B_NS.ADDOPCRA25 0743 8794h, ADC_B_NS.ADDOPCRA26 0743 87A4h, ADC_B_NS.ADDOPCRA27 0743 87B4h, ADC_B_NS.ADDOPCRA28 0743 87C4h, ADC_B_NS.ADDOPCRA29 0743 87D4h, ADC_B_NS.ADDOPCRA30 0743 87E4h, ADC_B_NS.ADDOPCRA31 0743 87F4h, ADC_B_NS.ADDOPCRA32 0743 8804h, ADC_B_NS.ADDOPCRA33 0743 8814h, ADC_B_NS.ADDOPCRA34 0743 8824h, ADC_B_NS.ADDOPCRA35 0743 8834h, ADC_B_NS.ADDOPCRA36 0743 8844h, ADC_B_NS.ADDOPCRA37 0743 8854h, ADC_B_NS.ADDOPCRA38 0743 8864h, ADC_B_NS.ADDOPCRA39 0743 8874h, ADC_B_NS.ADDOPCRA40 0743 8884h, ADC_B_NS.ADDOPCRA41 0743 8894h, ADC_B_NS.ADDOPCRA42 0743 88A4h, ADC_B_NS.ADDOPCRA43 0743 88B4h, ADC_B_NS.ADDOPCRA44 0743 88C4h, ADC_B_NS.ADDOPCRA45 0743 88D4h, ADC_B_NS.ADDOPCRA46 0743 88E4h, ADC_B_NS.ADDOPCRA47 0743 88F4h, ADC_B_NS.ADDOPCRA48 0743 8904h, ADC_B_NS.ADDOPCRA49 0743 8914h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	OFSETSEL[3:0]				—	—	—	—	GAINSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DFSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFSEL[2:0]	Digital Filter Selection*1, *2	b2 b0 0 0 0: Not use the digital filter 0 0 1: Use the 1st digital filter 0 1 0: Use the 2nd digital filter 0 1 1: Use the 3rd digital filter 1 0 0: Use the 4th digital filter Settings other than above are prohibited.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	GAINSEL[3:0]	User Gain Table Selection	b19 b16 0 0 0 0: Not use the user gain table 0 0 0 1: Use the user gain table 0 0 0 1 0: Use the user gain table 1 0 0 1 1: Use the user gain table 2 : : 1 0 0 0: Use the user gain table 7 Settings other than above are prohibited.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	OFSETSEL[3:0]	User Offset Table Selection	b27 b24 0 0 0 0: Not use the user offset table 0 0 0 1: Use the user offset table 0 0 0 1 0: Use the user offset table 1 0 0 1 1: Use the user offset table 2 : : 1 0 0 0: Use the user offset table 7 Settings other than above are prohibited.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Whenever A/D conversion is performed in oversampling mode or hybrid mode, the digital filter must be used.
Note 2. Whenever A/D conversion is performed in SAR mode, the digital filter must not be used.

The ADDOPCRAn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

DFSEL[2:0] Bits (Digital Filter Selection)

The DFSEL[2:0] bits select the digital filter for inputting A/D converted data of virtual channel n. When 000b is set in the DFSEL[2:0] bits, the digital filter is not used. When 001b, 010b, 011b, or 100b is set in the DFSEL[2:0] bits, the digital filter corresponding to the setting of the DFSEL[2:0] bits is selected. The characteristics of the digital filters are specified by ADDFSRm (m = 0 to 2) register.
For SAR mode, select 000b for the DFSEL[2:0] bits. For oversampling mode or hybrid mode, set the DFSEL[2:0] bits to 001b, 010b, 011b, or 100b.

GAINSEL[3:0] Bits (User Gain Table Selection)

The GAINSEL[3:0] bits select the user gain table for adjusting the A/D conversion data of virtual channel n.
When the GAINSEL[3:0] bits are set to 0h, no gain adjustment is performed. When the GAINSEL[3:0] bits are set to any of 1h to 8h, the gain adjustment is performed according to the user gain table selected by the GAINSEL[3:0] bits.

OFSETSEL[3:0] Bits (User Offset Table Selection)

The OFSETSEL[3:0] bits select the user offset table for adjusting the A/D conversion data of virtual channel n. When the OFSETSEL[3:0] bits are set to 0h, no offset adjustment is performed. When the OFSETSEL[3:0] bits are set to any of 1h to 8h, the offset adjustment is performed according to the user offset table selected by the OFSETSEL[3:0] bits.

60.2.35 A/D Conversion Data Operation Control B Register n (ADDOPCRBn) (n = 0 to 49)

Address(es): ADC_B.ADDOPCRB0 8743 8608h, ADC_B.ADDOPCRB1 8743 8618h, ADC_B.ADDOPCRB2 8743 8628h, ADC_B.ADDOPCRB3 8743 8638h, ADC_B.ADDOPCRB4 8743 8648h, ADC_B.ADDOPCRB5 8743 8658h, ADC_B.ADDOPCRB6 8743 8668h, ADC_B.ADDOPCRB7 8743 8678h, ADC_B.ADDOPCRB8 8743 8688h, ADC_B.ADDOPCRB9 8743 8698h, ADC_B.ADDOPCRB10 8743 86A8h, ADC_B.ADDOPCRB11 8743 86B8h, ADC_B.ADDOPCRB12 8743 86C8h, ADC_B.ADDOPCRB13 8743 86D8h, ADC_B.ADDOPCRB14 8743 86E8h, ADC_B.ADDOPCRB15 8743 86F8h, ADC_B.ADDOPCRB16 8743 8708h, ADC_B.ADDOPCRB17 8743 8718h, ADC_B.ADDOPCRB18 8743 8728h, ADC_B.ADDOPCRB19 8743 8738h, ADC_B.ADDOPCRB20 8743 8748h, ADC_B.ADDOPCRB21 8743 8758h, ADC_B.ADDOPCRB22 8743 8768h, ADC_B.ADDOPCRB23 8743 8778h, ADC_B.ADDOPCRB24 8743 8788h, ADC_B.ADDOPCRB25 8743 8798h, ADC_B.ADDOPCRB26 8743 87A8h, ADC_B.ADDOPCRB27 8743 87B8h, ADC_B.ADDOPCRB28 8743 87C8h, ADC_B.ADDOPCRB29 8743 87D8h, ADC_B.ADDOPCRB30 8743 87E8h, ADC_B.ADDOPCRB31 8743 87F8h, ADC_B.ADDOPCRB32 8743 8808h, ADC_B.ADDOPCRB33 8743 8818h, ADC_B.ADDOPCRB34 8743 8828h, ADC_B.ADDOPCRB35 8743 8838h, ADC_B.ADDOPCRB36 8743 8848h, ADC_B.ADDOPCRB37 8743 8858h, ADC_B.ADDOPCRB38 8743 8868h, ADC_B.ADDOPCRB39 8743 8878h, ADC_B.ADDOPCRB40 8743 8888h, ADC_B.ADDOPCRB41 8743 8898h, ADC_B.ADDOPCRB42 8743 88A8h, ADC_B.ADDOPCRB43 8743 88B8h, ADC_B.ADDOPCRB44 8743 88C8h, ADC_B.ADDOPCRB45 8743 88D8h, ADC_B.ADDOPCRB46 8743 88E8h, ADC_B.ADDOPCRB47 8743 88F8h, ADC_B.ADDOPCRB48 8743 8908h, ADC_B.ADDOPCRB49 8743 8918h,
ADC_B_NS.ADDOPCRB0 0743 8608h, ADC_B_NS.ADDOPCRB1 0743 8618h, ADC_B_NS.ADDOPCRB2 0743 8628h, ADC_B_NS.ADDOPCRB3 0743 8638h, ADC_B_NS.ADDOPCRB4 0743 8648h, ADC_B_NS.ADDOPCRB5 0743 8658h, ADC_B_NS.ADDOPCRB6 0743 8668h, ADC_B_NS.ADDOPCRB7 0743 8678h, ADC_B_NS.ADDOPCRB8 0743 8688h, ADC_B_NS.ADDOPCRB9 0743 8698h, ADC_B_NS.ADDOPCRB10 0743 86A8h, ADC_B_NS.ADDOPCRB11 0743 86B8h, ADC_B_NS.ADDOPCRB12 0743 86C8h, ADC_B_NS.ADDOPCRB13 0743 86D8h, ADC_B_NS.ADDOPCRB14 0743 86E8h, ADC_B_NS.ADDOPCRB15 0743 86F8h, ADC_B_NS.ADDOPCRB16 0743 8708h, ADC_B_NS.ADDOPCRB17 0743 8718h, ADC_B_NS.ADDOPCRB18 0743 8728h, ADC_B_NS.ADDOPCRB19 0743 8738h, ADC_B_NS.ADDOPCRB20 0743 8748h, ADC_B_NS.ADDOPCRB21 0743 8758h, ADC_B_NS.ADDOPCRB22 0743 8768h, ADC_B_NS.ADDOPCRB23 0743 8778h, ADC_B_NS.ADDOPCRB24 0743 8788h, ADC_B_NS.ADDOPCRB25 0743 8798h, ADC_B_NS.ADDOPCRB26 0743 87A8h, ADC_B_NS.ADDOPCRB27 0743 87B8h, ADC_B_NS.ADDOPCRB28 0743 87C8h, ADC_B_NS.ADDOPCRB29 0743 87D8h, ADC_B_NS.ADDOPCRB30 0743 87E8h, ADC_B_NS.ADDOPCRB31 0743 87F8h, ADC_B_NS.ADDOPCRB32 0743 8808h, ADC_B_NS.ADDOPCRB33 0743 8818h, ADC_B_NS.ADDOPCRB34 0743 8828h, ADC_B_NS.ADDOPCRB35 0743 8838h, ADC_B_NS.ADDOPCRB36 0743 8848h, ADC_B_NS.ADDOPCRB37 0743 8858h, ADC_B_NS.ADDOPCRB38 0743 8868h, ADC_B_NS.ADDOPCRB39 0743 8878h, ADC_B_NS.ADDOPCRB40 0743 8888h, ADC_B_NS.ADDOPCRB41 0743 8898h, ADC_B_NS.ADDOPCRB42 0743 88A8h, ADC_B_NS.ADDOPCRB43 0743 88B8h, ADC_B_NS.ADDOPCRB44 0743 88C8h, ADC_B_NS.ADDOPCRB45 0743 88D8h, ADC_B_NS.ADDOPCRB46 0743 88E8h, ADC_B_NS.ADDOPCRB47 0743 88F8h, ADC_B_NS.ADDOPCRB48 0743 8908h, ADC_B_NS.ADDOPCRB49 0743 8918h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CMPTBLE7	CMPTBLE6	CMPTBLE5	CMPTBLE4	CMPTBLE3	CMPTBLE2	CMPTBLE1	CMPTBLE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADC[3:0]				—	—	—	—	—	—	AVEMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	AVEMD[1:0]	Addition/Averaging Mode Selection	b1 b0 0 0: Not use Addition/Averaging mode 0 1: Addition mode 1 0: Averaging mode 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	ADC[3:0]	Addition/Averaging Times Selection	b11 b8 0 0 0 0: 1-time conversion (no addition, same as normal conversion) 0 0 0 1: 2-time conversion (1 addition) 0 0 1 1: 4-time conversion (3 additions) 0 1 0 0: 8-time conversion (7 additions) 0 1 0 1: 16-time conversion (15 additions) 0 1 1 0: 32-time conversion (31 additions) 0 1 1 1: 64-time conversion (63 additions) 1 0 0 0: 128-time conversion (127 additions) 1 0 0 1: 256-time conversion (255 additions) 1 0 1 0: 512-time conversion (511 additions) 1 0 1 1: 1024-time conversion (1023 additions) Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b23 to b16	CMPTBLE0 to CMPTBLE7	Compare Match Enable	The suffix number of each bit symbol corresponds to the compare match table number m. 0: Disable the compare match with the compare match table m 1: Enable the compare match with the compare match table m	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = 0 to 7

The ADDOPCRBn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

AVEMD[1:0] Bits (Addition/Averaging Mode Selection)

The AVEMD[1:0] bits select the addition mode or the averaging mode for the A/D conversion of virtual channel n. When the AVEMD[1:0] bits are set to 00b, the addition/averaging operation is not performed. When the AVEMD[1:0] bits are set to 01b or 10b, the addition/averaging operation is performed according to the setting value of the ADC[3:0] bits.

ADC[3:0] Bits (Addition/Averaging Times Selection)

The ADC[3:0] bits select the number of times of addition/averaging for the A/D conversion of virtual channel n. If the addition mode or the averaging mode is selected in the AVEMD[1:0] bits, the A/D conversion is performed a number of times according to the setting value of the ADC[3:0] bits.

CMPTBLEm Bits (Compare Match Enable) (m = 0 to 7)

The CMPTBLEm bits enable/disable the compare match with the compare match table m for the A/D conversion data of virtual channel n.

60.2.36 A/D Conversion Data Operation Control C Register n (ADDOPCRCn) (n = 0 to 49)

Address(es): ADC_B.ADDOPCRC0 8743 860Ch, ADC_B.ADDOPCRC1 8743 861Ch, ADC_B.ADDOPCRC2 8743 862Ch, ADC_B.ADDOPCRC3 8743 863Ch, ADC_B.ADDOPCRC4 8743 864Ch, ADC_B.ADDOPCRC5 8743 865Ch, ADC_B.ADDOPCRC6 8743 866Ch, ADC_B.ADDOPCRC7 8743 867Ch, ADC_B.ADDOPCRC8 8743 868Ch, ADC_B.ADDOPCRC9 8743 869Ch, ADC_B.ADDOPCRC10 8743 86ACh, ADC_B.ADDOPCRC11 8743 86BCh, ADC_B.ADDOPCRC12 8743 86CCh, ADC_B.ADDOPCRC13 8743 86DCh, ADC_B.ADDOPCRC14 8743 86ECh, ADC_B.ADDOPCRC15 8743 86FCh, ADC_B.ADDOPCRC16 8743 870Ch, ADC_B.ADDOPCRC17 8743 871Ch, ADC_B.ADDOPCRC18 8743 872Ch, ADC_B.ADDOPCRC19 8743 873Ch, ADC_B.ADDOPCRC20 8743 874Ch, ADC_B.ADDOPCRC21 8743 875Ch, ADC_B.ADDOPCRC22 8743 876Ch, ADC_B.ADDOPCRC23 8743 877Ch, ADC_B.ADDOPCRC24 8743 878Ch, ADC_B.ADDOPCRC25 8743 879Ch, ADC_B.ADDOPCRC26 8743 87ACh, ADC_B.ADDOPCRC27 8743 87BCh, ADC_B.ADDOPCRC28 8743 87CCh, ADC_B.ADDOPCRC29 8743 87DCh, ADC_B.ADDOPCRC30 8743 87ECh, ADC_B.ADDOPCRC31 8743 87FCh, ADC_B.ADDOPCRC32 8743 880Ch, ADC_B.ADDOPCRC33 8743 881Ch, ADC_B.ADDOPCRC34 8743 882Ch, ADC_B.ADDOPCRC35 8743 883Ch, ADC_B.ADDOPCRC36 8743 884Ch, ADC_B.ADDOPCRC37 8743 885Ch, ADC_B.ADDOPCRC38 8743 886Ch, ADC_B.ADDOPCRC39 8743 887Ch, ADC_B.ADDOPCRC40 8743 888Ch, ADC_B.ADDOPCRC41 8743 889Ch, ADC_B.ADDOPCRC42 8743 88ACh, ADC_B.ADDOPCRC43 8743 88BCh, ADC_B.ADDOPCRC44 8743 88CCh, ADC_B.ADDOPCRC45 8743 88DCh, ADC_B.ADDOPCRC46 8743 88ECh, ADC_B.ADDOPCRC47 8743 88FCh, ADC_B.ADDOPCRC48 8743 890Ch, ADC_B.ADDOPCRC49 8743 891Ch,
ADC_B_NS.ADDOPCRC0 0743 860Ch, ADC_B_NS.ADDOPCRC1 0743 861Ch, ADC_B_NS.ADDOPCRC2 0743 862Ch, ADC_B_NS.ADDOPCRC3 0743 863Ch, ADC_B_NS.ADDOPCRC4 0743 864Ch, ADC_B_NS.ADDOPCRC5 0743 865Ch, ADC_B_NS.ADDOPCRC6 0743 866Ch, ADC_B_NS.ADDOPCRC7 0743 867Ch, ADC_B_NS.ADDOPCRC8 0743 868Ch, ADC_B_NS.ADDOPCRC9 0743 869Ch, ADC_B_NS.ADDOPCRC10 0743 86ACh, ADC_B_NS.ADDOPCRC11 0743 86BCh, ADC_B_NS.ADDOPCRC12 0743 86CCh, ADC_B_NS.ADDOPCRC13 0743 86DCh, ADC_B_NS.ADDOPCRC14 0743 86ECh, ADC_B_NS.ADDOPCRC15 0743 86FCh, ADC_B_NS.ADDOPCRC16 0743 870Ch, ADC_B_NS.ADDOPCRC17 0743 871Ch, ADC_B_NS.ADDOPCRC18 0743 872Ch, ADC_B_NS.ADDOPCRC19 0743 873Ch, ADC_B_NS.ADDOPCRC20 0743 874Ch, ADC_B_NS.ADDOPCRC21 0743 875Ch, ADC_B_NS.ADDOPCRC22 0743 876Ch, ADC_B_NS.ADDOPCRC23 0743 877Ch, ADC_B_NS.ADDOPCRC24 0743 878Ch, ADC_B_NS.ADDOPCRC25 0743 879Ch, ADC_B_NS.ADDOPCRC26 0743 87ACh, ADC_B_NS.ADDOPCRC27 0743 87BCh, ADC_B_NS.ADDOPCRC28 0743 87CCh, ADC_B_NS.ADDOPCRC29 0743 87DCh, ADC_B_NS.ADDOPCRC30 0743 87ECh, ADC_B_NS.ADDOPCRC31 0743 87FCh, ADC_B_NS.ADDOPCRC32 0743 880Ch, ADC_B_NS.ADDOPCRC33 0743 881Ch, ADC_B_NS.ADDOPCRC34 0743 882Ch, ADC_B_NS.ADDOPCRC35 0743 883Ch, ADC_B_NS.ADDOPCRC36 0743 884Ch, ADC_B_NS.ADDOPCRC37 0743 885Ch, ADC_B_NS.ADDOPCRC38 0743 886Ch, ADC_B_NS.ADDOPCRC39 0743 887Ch, ADC_B_NS.ADDOPCRC40 0743 888Ch, ADC_B_NS.ADDOPCRC41 0743 889Ch, ADC_B_NS.ADDOPCRC42 0743 88ACh, ADC_B_NS.ADDOPCRC43 0743 88BCh, ADC_B_NS.ADDOPCRC44 0743 88CCh, ADC_B_NS.ADDOPCRC45 0743 88DCh, ADC_B_NS.ADDOPCRC46 0743 88ECh, ADC_B_NS.ADDOPCRC47 0743 88FCh, ADC_B_NS.ADDOPCRC48 0743 890Ch, ADC_B_NS.ADDOPCRC49 0743 891Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SIGNSEL	—	—	ADPRC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	LIMTBLS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LIMTBLS[3:0]	Limiter Clip Table Selection	b3 b0 0 0 0 0: Not use the limiter clip table 0 0 0 1: Use the limiter clip table 0 0 0 1 0: Use the limiter clip table 1 0 0 1 1: Use the limiter clip table 2 : : 1 0 0 0: Use the limiter clip table 7 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	ADPRC[1:0]	A/D Conversion Data Format Selection	b17 b16 0 0: Store the A/D conversion result as 16-bit data format 0 1: Store the A/D conversion result as 14-bit data format 1 0: Store the A/D conversion result as 12-bit data format 1 1: Store the A/D conversion result as 10-bit data format	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	SIGNSEL	A/D Conversion Data Signed/ Un-Signed Selection	0: Signed data format 1: Un-signed data format	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDOPCRCn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

LIMTBLS[3:0] Bits (Limiter Clip Table Selection)

The LIMTBLS[3:0] bits select the limiter clip table for clipping the A/D conversion data of virtual channel n. When the LIMTBLS[3:0] bits are set to 0h, the A/D conversion data is not clipped. When the LIMTBLS[3:0] bits are set to any of 1h to 8h, the A/D conversion data is clipped according to the limiter clip table selected by the LIMTBLS[3:0] bits.

ADPRC[1:0] Bits (A/D Conversion Data Format Selection)

The ADPRC[1:0] bits select the data length of A/D conversion data. The A/D conversion data of virtual channel n is stored to the A/D data register or the FIFO with selected data length.

SIGNSEL Bit (A/D Conversion Data Signed/Un-Signed Selection)

The SIGNSEL bit selects the Signed/Unsigned data format of the A/D conversion data. The A/D conversion data of the virtual channel n is stored to the A/D data register or the FIFO with selected data format.
Select the signed data format (SIGNSEL = 0) whenever the A/D conversion of the self-diagnosis channel is performed. Select the unsigned data format (SIGNSEL = 1) whenever the A/D conversion in SAR mode, unless to convert the self-diagnosis channel, is performed. The signed data format or the unsigned data format is available for the A/D conversion in oversampling mode or hybrid mode unless converting to the self-diagnosis channel.

60.2.37 Scan Group Diagnosis Function Control Register n (ADSGDCRn) (n = 0 to 11)

Address(es): ADC_B.ADSGDCR0 8743 8200h, ADC_B.ADSGDCR1 8743 8204h, ADC_B.ADSGDCR2 8743 8208h, ADC_B.ADSGDCR3 8743 820Ch, ADC_B.ADSGDCR4 8743 8210h, ADC_B.ADSGDCR5 8743 8214h, ADC_B.ADSGDCR6 8743 8218h, ADC_B.ADSGDCR7 8743 821Ch, ADC_B.ADSGDCR8 8743 8220h, ADC_B.ADSGDCR9 8743 8224h, ADC_B.ADSGDCR10 8743 8228h, ADC_B.ADSGDCR11 8743 822Ch, ADC_B_NS.ADSGDCR0 0743 8200h, ADC_B_NS.ADSGDCR1 0743 8204h, ADC_B_NS.ADSGDCR2 0743 8208h, ADC_B_NS.ADSGDCR3 0743 820Ch, ADC_B_NS.ADSGDCR4 0743 8210h, ADC_B_NS.ADSGDCR5 0743 8214h, ADC_B_NS.ADSGDCR6 0743 8218h, ADC_B_NS.ADSGDCR7 0743 821Ch, ADC_B_NS.ADSGDCR8 0743 8220h, ADC_B_NS.ADSGDCR9 0743 8224h, ADC_B_NS.ADSGDCR10 0743 8228h, ADC_B_NS.ADSGDCR11 0743 822Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADNDIS[7:0]								—	—	ADDISN	ADDISP	—	—	—	ADDISEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DIAGVAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DIAGVAL[2:0]	Self-Diagnosis Mode Selection	b2 b0 0 0 0: Set when any self-diagnosis channel is not included. Setting prohibited when any self-diagnosis channels are included. 1 0 0: Self-diagnosis mode 1 1 0 1: Self-diagnosis mode 2 1 1 0: Self-diagnosis mode 3 Settings other than above are prohibited.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ADDISEN	Disconnection Detection Assist Enable	0: Disable the disconnection detection assist function 1: Enable the disconnection detection assist function	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	ADDISP	Disconnection Detection Assist Mode Selection (for the Even-Numbered Analog Channel)	0: Discharge 1: Precharge	R/W
b21	ADDISN	Disconnection Detection Assist Mode Selection (for the Odd-Numbered Analog Channel)	0: Discharge 1: Precharge	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	ADNDIS[7:0]	Disconnection Detection Assist Period	b31 b24 0 0 0 0 0 0 0 0: Setting prohibited when the disconnection detection assist function is enabled 0 0 0 0 0 0 0 1: Setting prohibited 0 0 0 0 0 0 1 0: Setting prohibited 0 0 0 0 0 0 1 1 to 1 1 1 1 1 1 1 1: The number of states for the discharge or precharge period – 1	R/W

The ADSGDCRn register controls the diagnosis function at the scanning operation of scan group n.

DIAGVAL[2:0] Bits (Self-Diagnosis Mode Selection)

DIAGVAL[2:0] bits select the self-diagnosis mode for A/D converter.

The DIAGVAL[2:0] bits must be set to 000b when the scan group n does not include any virtual channels that selected the self-diagnosis channel. The DIAGVAL[2:0] bits must be set to 100b, 101b or 110b when the scan group n is including the virtual channel that selected the self-diagnosis channel.

ADDISEN Bit (Disconnection Detection Assist Enable)

The ADDISEN bit enables/disables the disconnection detection assist function. When the ADDISEN bit is set to 0, the disconnection detection assist operation is not performed. When the ADDISEN bit is set to 1, the disconnection detection assist operation according to the setting of the ADDISP bit and the ADDISN bit is performed at the scanning operation of scan group n.

ADDISP Bit (Disconnection Detection Assist Mode Selection (for the Even-Numbered Analog Channel))

The ADDISP bit select the discharge/precharge operation for the disconnection detection. The ADDISP bit selects the discharge/precharge operation for the even-numbered analog channel.

ADDISN Bit (Disconnection Detection Assist Mode Selection (for the Odd-Numbered Analog Channel)

The ADDISN bit select the discharge/precharge operation for the disconnection detection. The ADDISN bit selects the discharge/precharge operation for the odd-numbered analog channel.

ADNDIS[7:0] Bits (Disconnection Detection Assist Period)

The ADNDIS[7:0] bits select the discharge/precharge period. When the disconnection detection assist function is disabled (ADDISEN = 0), the ADNDIS[7:0] bits should be set to 00h. When the disconnection detection assist function is enabled (ADDISEN = 1), the ADNDIS[7:0] bits should be set the value within 03h to FFh as the number of state for the discharge/ precharge period. The number of states actually valid in the operation is the value of the register setting minus one.

60.2.38 Sampling State Table Register 0 (ADSSTR0)

Address(es): ADC_B.ADSSTR0 8743 8240h, ADC_B_NS.ADSSTR0 0743 8240h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST1[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST0[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST0[9:0]	Sampling State Table 0	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST1[9:0]	Sampling State Table 1	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR0 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.39 Sampling State Table Register 1 (ADSSTR1)

Address(es): ADC_B.ADSSTR1 8743 8244h, ADC_B_NS.ADSSTR1 0743 8244h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST3[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST2[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST2[9:0]	Sampling State Table 2	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST3[9:0]	Sampling State Table 3	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR1 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.40 Sampling State Table Register 2 (ADSSTR2)

Address(es): ADC_B.ADSSTR2 8743 8248h, ADC_B_NS.ADSSTR2 0743 8248h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST5[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST4[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST4[9:0]	Sampling State Table 4	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST5[9:0]	Sampling State Table 5	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR2 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.41 Sampling State Table Register 3 (ADSSTR3)

Address(es): ADC_B.ADSSTR3 8743 824Ch, ADC_B_NS.ADSSTR3 0743 824Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST7[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST6[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST6[9:0]	Sampling State Table 6	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST7[9:0]	Sampling State Table 7	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR3 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.42 Sampling State Table Register 4 (ADSSTR4)

Address(es): ADC_B.ADSSTR4 8743 8250h, ADC_B_NS.ADSSTR4 0743 8250h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST9[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST8[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST8[9:0]	Sampling State Table 8	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST9[9:0]	Sampling State Table 9	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR4 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.43 Sampling State Table Register 5 (ADSSTR5)

Address(es): ADC_B.ADSSTR5 8743 8254h, ADC_B_NS.ADSSTR5 0743 8254h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST11[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST10[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST10[9:0]	Sampling State Table 10	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST11[9:0]	Sampling State Table 11	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR5 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.44 Sampling State Table Register 6 (ADSSTR6)

Address(es): ADC_B.ADSSTR6 8743 8258h, ADC_B_NS.ADSSTR6 0743 8258h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST13[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST12[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST12[9:0]	Sampling State Table 12	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST13[9:0]	Sampling State Table 13	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR6 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.45 Sampling State Table Register 7 (ADSSTR7)

Address(es): ADC_B.ADSSTR7 8743 825Ch, ADC_B_NS.ADSSTR7 0743 825Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SST15[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SST14[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SST14[9:0]	Sampling State Table 14	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	SST15[9:0]	Sampling State Table 15	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR7 register selects the sampling time at the A/D conversion that is using each sampling state table.

60.2.46 A/D Conversion State Register (ADCNVSTR)

Address(es): ADC_B.ADCNVSTR 8743 8260h, ADC_B_NS.ADCNVSTR 0743 8260h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CST2[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST1[5:0]						—	—	CST0[5:0]					
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CST0[5:0]	A/D Converter Unit 0 (ADC0) A/D Successive Approximation Time Configuration	These bits set the successive approximation time in the range from 3 to 63 states.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	CST1[5:0]	A/D Converter Unit 1 (ADC1) A/D Successive Approximation Time Configuration	These bits set the successive approximation time in the range from 3 to 63 states.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21 to b16	CST2[5:0]	A/D Converter Unit 2 (ADC2) A/D Successive Approximation Time Configuration	These bits set the successive approximation time in the range from 3 to 63 states.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCNVSTR register specifies the A/D successive approximation time, which required by the A/D converter Unit 0, Unit 1 or Unit 2 to convert the sampled analog input to the digital value, in the number of cycles based on A/D conversion clock (ADCLK). The A/D successive approximation time does not include the sampling time of the analog input.

The A/D successive approximation time must be set to meet the value specified in section 67, Electrical Characteristics. If the successive approximation time is set to a value over the range specified for the electrical characteristics, the A/D conversion result is not guaranteed. In addition, the A/D successive approximation time for each A/D converter should be set to be the same. (Set the same value to CST0[5:0] bits, CST1[5:0] bits, and CST2[5:0] bits).

60.2.47 Channel-Dedicated Sample-and-Hold Circuit Control Register 0 (ADSHCR0)

Address(es): ADC_B.ADSHCR0 8743 8280h, ADC_B_NS.ADSHCR0 0743 8280h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SHMD2	SHMD1	SHMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SHEN2	SHEN1	SHEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHEN0*1	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Bypass Select	0: Bypass the circuit unit 0 1: Use the circuit unit 0	R/W
b1	SHEN1*1	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Bypass Select	0: Bypass the circuit unit 1 1: Use the circuit unit 1	R/W
b2	SHEN2*1	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Bypass Select	0: Bypass the circuit unit 2 1: Use the circuit unit 2	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SHMD0*2	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Input Mode Select	0: Single-ended Input mode 1: Differential Input mode	R/W
b17	SHMD1*2	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Input Mode Select	0: Single-ended Input mode 1: Differential Input mode	R/W
b18	SHMD2*2	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Input Mode Select	0: Single-ended Input mode 1: Differential Input mode	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using a channel-dedicated sample-and-hold circuit, also set ADSHSBPCR (for single-ended input mode) or ADSHDBPCR (for differential input mode).
Note 2. When using a channel-dedicated sample-and-hold circuit, make sure that its polarity matches the setting of the ADCHCRn.AINMD bit of virtual channel n (n = 0 to 49) to which the channel-dedicated sample-and-hold circuit is assigned.

The ADSHCR0 register controls the channel-dedicated sample-and-hold circuits (unit 0, unit 1 and unit 2) connected to the A/D converter unit 0 (ADC0).

SHENn Bit (Channel-dedicated Sample-and-hold Circuit Bypass Select) (n = 0 to 2)

The SHENn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuits unit n.
When the SHENn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is disabled. If A/D conversion of analog input channel which is connected to the channel-dedicated sample-and-hold circuit unit n is performed, the channel-dedicated sample-and-hold circuit is not used and is bypassed.
When the SHENn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is enabled. In this case, A/D conversion of analog input channel to which the channel-dedicated sample-and-hold circuit unit n is performed with the channel-dedicated sample-and-hold circuit used.

SHMDn Bit (Channel-dedicated Sample-and-hold Circuit Input Mode Select) (n = 0 to 2)

The SHMDn bit selects the input mode of the channel-dedicated sample-and-hold circuits unit n.
When the SHMDn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is in single-ended input mode.
When the SHMDn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is in differential input mode.

60.2.48 Channel-Dedicated Sample-and-Hold Circuit Diagnosis Control Register 0 (ADSHDCR0)

Address(es): ADC_B.ADSHDCR0 8743 8284h, ADC_B_NS.ADSHDCR0 0743 8284h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SH0DIS	—	—	—	SH0DISEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SH0DISEN	Disconnection Detection Assist Enable	0: Disable the disconnection detection assist function 1: Enable the disconnection detection assist function	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SH0DIS	Disconnection Detection Assist Mode Selection	0: Discharge 1: Precharge	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHDCR0 register controls settings of the diagnosis function for the channel-dedicated sample-and-hold circuits (unit 0, unit 1 and unit 2) connected to the A/D converter unit 0 (ADC0).

SH0DISEN Bit (Disconnection Detection Assist Enable)

The SH0DISEN bit enables or disables the open-wire detection assist function of the channel-dedicated sample-and-hold circuit. When the SH0DISEN bit is set to 0, the open-wire detection assist operation is not performed. When the SH0DISEN bit is set to 1, the open-wire detection assist operation according to the setting of the SH0DIS bit is performed during the scan operation of scan group n.

SH0DIS Bit (Disconnection Detection Assist Mode Selection)

The SH0DIS bit selects whether to discharge or precharge when a disconnection is detected in the channel-specific sample-and-hold circuit.

60.2.49 Channel-Dedicated Sample-and-Hold Circuit State Register 0 (ADSHSTR0)

Address(es): ADC_B.ADSHSTR0 8743 8288h, ADC_B_NS.ADSHSTR0 0743 8288h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SHHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHSST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SHSST[7:0]	Sampling Time Setting	These bits set the sampling time in the range from 4 to 255 states	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	SHHST[2:0]	Hold Mode Switching Time Setting	These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHSTR0 register specifies the sampling time and the hold mode switching time for the channel-dedicated sample-and-hold circuits (unit 0, unit 1, and unit 2) connected to the A/D converter unit 0 (ADC0). The value set in this register should be set to meet the value specified in section 67, Electrical Characteristics.

60.2.50 Channel-Dedicated Sample-and-Hold Circuit Single-Ended Input Mode Bypass Control Register (ADSHSBPCR)

Address(es): ADC_B.ADSHSBPCR 8743 8310h, ADC_B_NS.ADSHSBPCR 0743 8310h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SHSBN2	SHSBN1	SHSBN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SHSBP2	SHSBP1	SHSBP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHSBP0	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Single-Ended Input Bypass Selection (for the Even-Numbered Analog Channel)	0: Use the circuit unit 0 1: Bypass the circuit unit 0	R/W
b1	SHSBP1	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Single-Ended Input Bypass Selection (for the Even-Numbered Analog Channel)	0: Use the circuit unit 1 1: Bypass the circuit unit 1	R/W
b2	SHSBP2	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Single-Ended Input Bypass Selection (for the Even-Numbered Analog Channel)	0: Use the circuit unit 2 1: Bypass the circuit unit 2	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SHSBN0	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Single-Ended Input Bypass Selection (for the Odd-Numbered Analog Channel)	0: Use the circuit unit 0 1: Bypass the circuit unit 0	R/W
b17	SHSBN1	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Single-Ended Input Bypass Selection (for the Odd-Numbered Analog Channel)	0: Use the circuit unit 1 1: Bypass the circuit unit 1	R/W
b18	SHSBN2	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Single-Ended Input Bypass Selection (for the Odd-Numbered Analog Channel)	0: Use the circuit unit 2 1: Bypass the circuit unit 2	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Write access is permitted only if ADSHCR0.SHENn for the target sample-and-hold circuit is set to 0.
Note: If you set both the even-numbered and odd-numbered channels of the same unit to bypass using this register, do not set the two corresponding virtual channels to the same scan group.

The ADSHSBPCR register controls the bypass operation for even-numbered or odd-numbered analog channels individually when using the channel-dedicated sample-and-hold circuits (unit 0, unit 1, and unit 2) in single-ended input mode.

SHSBPn Bit (Channel-Dedicated Sample-and-Hold Circuit Unit n Single-Ended Input Bypass Selection (for the Even-Numbered Analog Channel)) (n = 0 to 2)

The SHSBPn bit control the analog input for A/D conversion when the channel-dedicated sample-and-hold circuit is enabled.

When the bit is set to 0, the analog output held by the channel-dedicated sample-and-hold circuit is A/D-converted.

When the bit is set to 1, the analog input of the channel-dedicated sample-and-hold circuit is A/D-converted.

SHSBNn Bit (Channel-Dedicated Sample-and-Hold Circuit Unit n Single-Ended Input Bypass Selection (for the Odd-Numbered Analog Channel)) (n = 0 to 2)

The SHSBNn bit control the analog input for A/D conversion when the channel-dedicated sample-and-hold circuit is enabled.

When the bit is set to 0, the analog output held by the channel-dedicated sample-and-hold circuit is A/D-converted.

When the bit is set to 1, the analog input of the channel-dedicated sample-and-hold circuit is A/D-converted.

60.2.51 Channel-Dedicated Sample-and-Hold Circuit Differential Input Mode Bypass Control Register (ADSHDBPCR)

Address(es): ADC_B.ADSHDBPCR 8743 8314h, ADC_B_NS.ADSHDBPCR 0743 8314h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SHDBP 11	SHDBP 10	SHDBP 9	SHDBP 8	SHDBP 7	SHDBP 6	SHDBP 5	SHDBP 4	SHDBP 3	SHDBP 2	SHDBP 1	SHDBP 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHDBP0	Scan Group 0 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b1	SHDBP1	Scan Group 1 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b2	SHDBP2	Scan Group 2 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b3	SHDBP3	Scan Group 3 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b4	SHDBP4	Scan Group 4 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b5	SHDBP5	Scan Group 5 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b6	SHDBP6	Scan Group 6 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b7	SHDBP7	Scan Group 7 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b8	SHDBP8	Scan Group 8 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b9	SHDBP9	Scan Group 9 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b10	SHDBP10	Scan Group 10 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b11	SHDBP11	Scan Group 11 Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection	0: Use the channel-dedicated sample-and-hold circuit 1: Bypass the channel-dedicated sample-and-hold circuit	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHDBPCR register controls the bypass operation of the channel-dedicated sample-and-hold circuits in differential input mode of scan group n.

SHDBPn Bit (Scan Group n Channel Dedicated Sample-and-Hold Circuit Differential Input Bypass Selection) (n = 0 to 11)

The SHDBPn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuit of scan group n.
When the SHDBPn bit is set to 0, the output of the channel-dedicated sample-and-hold circuit is A/D converted.
When the SHDBPn bit is set to 1, the analog channel input can be A/D converted without going through the channel-dedicated sample-and-hold circuit.
If the A/D disconnection detection assist function is required when using the channel-dedicated sample-and-hold circuit in differential input mode, recalibration due to mode switching is not required if this bit is set to bypass.
For details, see section 60.3.17.2 (5) Advanced operation (in differential input mode) with the disconnection detection assist function in use in SAR mode.

60.2.52 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Control Register 0 (ADSHSDCR0)

Address(es): ADC_B.ADSHSDCR0 8743 8318h, ADC_B_NS.ADSHSDCR0 0743 8318h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SHDIAG7[2:0]			—	SHDIAG6[2:0]			—	SHDIAG5[2:0]			—	SHDIAG4[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	SHDIAG3[2:0]			—	SHDIAG2[2:0]			—	SHDIAG1[2:0]			—	SHDIAG0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SHDIAG0[2:0]	Scan Group 0 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b2 b0 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SHDIAG1[2:0]	Scan Group 1 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b6 b4 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	SHDIAG2[2:0]	Scan Group 2 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b10 b8 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	SHDIAG3[2:0]	Scan Group 3 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b14 b12 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W

Bit	Symbol	Bit Name	Description	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	SHDIAG4[2:0]	Scan Group 4 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b18 b16 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	SHDIAG5[2:0]	Scan Group 5 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b22 b20 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	SHDIAG6[2:0]	Scan Group 6 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b26 b24 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	SHDIAG7[2:0]	Scan Group 7 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b30 b28 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The ADSHSDCR0 register controls the self-diagnostic function of the channel-dedicated sample-and-hold circuit.

60.2.53 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Control Register 1 (ADSHSDCR1)

Address(es): ADC_B.ADSHSDCR1 8743 831Ch, ADC_B_NS.ADSHSDCR1 0743 831Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	SHDIAG11[2:0]			—	SHDIAG10[2:0]			—	SHDIAG9[2:0]			—	SHDIAG8[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SHDIAG8[2:0]	Scan Group 8 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b2 b1 b0 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SHDIAG9[2:0]	Scan Group 9 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b6 b5 b4 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	SHDIAG10[2:0]	Scan Group 10 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b10 b9 b8 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b14 to b12	SHDIAG11[2:0]	Scan Group 11 Channel-Dedicated Sample-and-Hold Circuit Self-Diagnosis Voltage Select	b14 b12 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 0 0 1: Self-diagnosis mode 4 0 1 0: Self-diagnosis mode 5 0 1 1: Self-diagnosis mode 6 1 0 0: Self-diagnosis mode 7 1 0 1: Self-diagnosis mode 8 1 1 0: Self-diagnosis mode 9 Settings other than above are prohibited.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHSDCR1 register controls the self-diagnostic function of the channel-dedicated sample-and-hold circuit.

60.2.54 Channel-Dedicated Sample-and-Hold Circuit Constant Sampling Control Register (ADSHCSCR)

Address(es): ADC_B.ADSHCSCR 8743 8C04h, ADC_B_NS.ADSHCSCR 0743 8C04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SHCSST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHCSST0	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Constant Sampling Enabled	0: Disable constant sampling 1: Enable constant sampling	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set after self-calibration has been performed.
Note: Do not set this register if the channel-dedicated sample-and-hold circuit is not used.

The ADSHCSCR register controls the constant sampling function of channel-dedicated sample-and-hold circuit.
When constant sampling is enabled, sampling of the channel-dedicated sample-and-hold circuit begins.
For details about the constant sampling function, see section 60.3.17.2 (4) SAR mode (with constant sampling enabled).

60.2.55 Internal Reference Voltage Monitor Enable Register (ADREFCR)

Address(es): ADC_B.ADREFCR 8743 8320h, ADC_B_NS.ADREFCR 0743 8320h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VDE	Internal Reference Voltage A/D Conversion Select	0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADREFCR register controls the function of the internal reference voltage circuit. When the A/D conversion of the internal reference voltage is performed, the VDE bit must be set to 1.

60.2.56 A/D Converter Digital Filter Selection Register n (ADDFSRn) (n = 0 to 2)

Address(es): ADC_B.ADDFSR0 8743 8340h, ADC_B.ADDFSR1 8743 8344h, ADC_B.ADDFSR2 8743 8348h
ADC_B_NS.ADDFSR0 0743 8340h, ADC_B_NS.ADDFSR1 0743 8344h, ADC_B_NS.ADDFSR2 0743 8348h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	DFSEL3[1:0]		—	—	—	—	—	—	DFSEL2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DFSEL1[1:0]		—	—	—	—	—	—	DFSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DFSEL0[1:0]	A/D Converter Unit n the 1st Digital Filter Characteristic Selection	b1 b0 0 1: Sinc filter Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DFSEL1[1:0]	A/D Converter Unit n the 2nd Digital Filter Characteristic Selection	b9 b8 0 1: Sinc filter Settings other than above are prohibited.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	DFSEL2[1:0]	A/D Converter Unit n the 3rd Digital Filter Characteristic Selection	b17 b16 0 1: Sinc filter Settings other than above are prohibited.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	DFSEL3[1:0]	A/D Converter Unit n the 4th Digital Filter Characteristic Selection	b25 b24 0 1: Sinc filter Settings other than above are prohibited.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDFSRn register selects the characteristics of the digital filter for the A/D converter unit n. The selection of the digital filter to be used at A/D conversion is specified by the ADDOPCRAm register (m = 0 to 49). If the digital filter function is used with prohibited values set to the ADDFSRn.DFSELx[1:0] (x = 0 to 3) bits, the A/D conversion characteristics and A/D conversion results are not guaranteed.

60.2.57 A/D Converter Self-Calibration State Register (ADCALSTCR)

Address(es): ADC_B.ADCALSTCR 8743 8264h, ADC_B_NS.ADCALSTCR 0743 8264h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CALADCST[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CALADSST[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CALADSST[9:0]	A/D Converter Self-Calibration Sampling Time Configuration	These bits set the sampling time in the range from 2 to 1023 states.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21 to b16	CALADCST[5:0]	A/D Converter Self-Calibration Successive Approximation Time Configuration	These bits set the successive approximation time in the range from 3 to 63 states.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCALSTCR register specifies the sampling and the A/D successive approximation times at the self-calibration operation for all A/D converter (ADC0, ADC1, ADC2). The sampling time and successive approximation time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 67, Electrical Characteristics.

60.2.58 Channel-Dedicated Sample-and-Hold Circuit Self-Calibration State Register (ADCALSHCR)

Address(es): ADC_B.ADCALSHCR 8743 82B0h, ADC_B_NS.ADCALSHCR 0743 82B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CALSHHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CALSHSST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CALSHSST[7:0]	Channel-Dedicated Sample-and-Hold Circuit Self-Calibration Sampling Time Configuration	These bits set the sampling time in the range from 4 to 255 states.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	CALSHHST[2:0]	Channel-Dedicated Sample-and-Hold Circuit Self-Calibration Hold Mode Switching Time Configuration	These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCALSHCR register specifies the sampling time and the hold mode switching time at the self-calibration operation for all channel-dedicated sample-and-hold circuits (SH0 to SH2). The sampling time and the hold mode switching time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 67, Electrical Characteristics.

60.2.59 User Offset Table Register n (ADUOFTRn) (n = 0 to 7)

Address(es): ADC_B.ADUOFTR0 8743 8360h, ADC_B.ADUOFTR1 8743 8364h, ADC_B.ADUOFTR2 8743 8368h, ADC_B.ADUOFTR3 8743 836Ch, ADC_B.ADUOFTR4 8743 8370h, ADC_B.ADUOFTR5 8743 8374h, ADC_B.ADUOFTR6 8743 8378h, ADC_B.ADUOFTR7 8743 837Ch, ADC_B_NS.ADUOFTR0 0743 8360h, ADC_B_NS.ADUOFTR1 0743 8364h, ADC_B_NS.ADUOFTR2 0743 8368h, ADC_B_NS.ADUOFTR3 0743 836Ch, ADC_B_NS.ADUOFTR4 0743 8370h, ADC_B_NS.ADUOFTR5 0743 8374h, ADC_B_NS.ADUOFTR6 0743 8378h, ADC_B_NS.ADUOFTR7 0743 837Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	UOFSET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	UOFSET[15:0]	User Offset Table n	7FFFh: +32767 7FFEh: +32766 : 0002h: +2 0001h: +1 0000h: 0 (without user offset) FFFFh: –1 FFFEh: –2 : 8001h: –32767 8000h: –32768	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADUOFTRn register specifies the offset value in the user offset table n.

60.2.60 User Gain Table Register n (ADUGTRn) (n = 0 to 7)

Address(es): ADC_B.ADUGTR0 8743 8380h, ADC_B.ADUGTR1 8743 8384h, ADC_B.ADUGTR2 8743 8388h, ADC_B.ADUGTR3 8743 838Ch, ADC_B.ADUGTR4 8743 8390h, ADC_B.ADUGTR5 8743 8394h, ADC_B.ADUGTR6 8743 8398h, ADC_B.ADUGTR7 8743 839Ch, ADC_B_NS.ADUGTR0 0743 8380h, ADC_B_NS.ADUGTR1 0743 8384h, ADC_B_NS.ADUGTR2 0743 8388h, ADC_B_NS.ADUGTR3 0743 838Ch, ADC_B_NS.ADUGTR4 0743 8390h, ADC_B_NS.ADUGTR5 0743 8394h, ADC_B_NS.ADUGTR6 0743 8398h, ADC_B_NS.ADUGTR7 0743 839Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	UGAIN[15:0]															
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	UGAIN[15:0]	User Gain Table n	These bits set the gain value to be multiplied by the A/D conversion result. UGAIN[15:14]: Integer part of gain UGAIN[13:0]: Fractional part of gain	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADUGTRn register specifies the gain value in the user gain table n.

60.2.61 Limiter Clip Interrupt Enable Register (ADLIMINTCR)

Address(es): ADC_B.ADLIMINTCR 8743 83A0h, ADC_B_NS.ADLIMINTCR 0743 83A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LIMIE11	LIMIE10	LIMIE9	LIMIE8	LIMIE7	LIMIE6	LIMIE5	LIMIE4	LIMIE3	LIMIE2	LIMIE1	LIMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	LIMIE0 to LIMIE11	Limiter Clip Interrupt n Enable	The suffix number of each bit symbol corresponds to the limiter clip interrupt number n. 0: Disable the limiter clip interrupt n 1: Enable the limiter clip interrupt n	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADLIMINTCR register enables/disables the limiter clip interrupt n.

60.2.62 Limiter Clip Table Register n (ADLIMTRn) (n = 0 to 7)

Address(es): ADC_B.ADLIMTR0 8743 83A4h, ADC_B.ADLIMTR1 8743 83A8h, ADC_B.ADLIMTR2 8743 83ACh, ADC_B.ADLIMTR3 8743 83B0h, ADC_B.ADLIMTR4 8743 83B4h, ADC_B.ADLIMTR5 8743 83B8h, ADC_B.ADLIMTR6 8743 83BCh, ADC_B.ADLIMTR7 8743 83C0h, ADC_B_NS.ADLIMTR0 0743 83A4h, ADC_B_NS.ADLIMTR1 0743 83A8h, ADC_B_NS.ADLIMTR2 0743 83ACh, ADC_B_NS.ADLIMTR3 0743 83B0h, ADC_B_NS.ADLIMTR4 0743 83B4h, ADC_B_NS.ADLIMTR5 0743 83B8h, ADC_B_NS.ADLIMTR6 0743 83BCh, ADC_B_NS.ADLIMTR7 0743 83C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LIMU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LIML[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LIML[15:0]	Limiter Clip Table n: Lower-side Limit Value		R/W
b31 to b16	LIMU[15:0]	Limiter Clip Table n: Upper-side Limit Value		R/W

The ADLIMTRn register specifies the lower- and upper-side limit value of the limiter clip table n.

60.2.63 Limiter Clip Scan Group Status Register (ADLIMGRSR)

Address(es): ADC_B.ADLIMGRSR 8743 8D28h, ADC_B_NS.ADLIMGRSR 0743 8D28h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LIMGRF11	LIMGRF10	LIMGRF9	LIMGRF8	LIMGRF7	LIMGRF6	LIMGRF5	LIMGRF4	LIMGRF3	LIMGRF2	LIMGRF1	LIMGRF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	LIMGRF0 to LIMGRF11	Scan Group n Limiter Clip Flag	The suffix number of each bit symbol corresponds to the scan group number n. 0: Limiter clip for scan group n is not detected 1: Limiter clip for scan group n is detected	R
b31 to b12	—	Reserved	These bits are read as 0.	R

n = 0 to 11

The ADLIMGRSR register indicates whether the limiter clip occurred in the scanning operation for scan group n. Each flag can be cleared in the ADLIMGRSCR.

60.2.64 Limiter Clip Channel Status Register 0 (ADLIMCHSR0)

Address(es): ADC_B.ADLIMCHSR0 8743 8D2Ch, ADC_B_NS.ADLIMCHSR0 0743 8D2Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LIMCHF31	LIMCHF30	LIMCHF29	LIMCHF28	LIMCHF27	LIMCHF26	LIMCHF25	LIMCHF24	LIMCHF23	LIMCHF22	LIMCHF21	LIMCHF20	LIMCHF19	LIMCHF18	LIMCHF17	LIMCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LIMCHF15	LIMCHF14	LIMCHF13	LIMCHF12	LIMCHF11	LIMCHF10	LIMCHF9	LIMCHF8	LIMCHF7	LIMCHF6	LIMCHF5	LIMCHF4	LIMCHF3	LIMCHF2	LIMCHF1	LIMCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	LIMCHF0 to LIMCHF31	Analog Channel n: Limiter Clip Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Limiter clip is not detected 1: Limiter clip is detected	R

n = 0 to 31

The ADLIMCHSR0 register indicates whether the limiter clip occurred when the A/D conversion for the analog channel n is performed. Each flag can be cleared in ADLIMCHSCR0.

60.2.65 Limiter Clip Channel Status Register 1 (ADLIMCHSR1)

Address(es): ADC_B.ADLIMCHSR1 8743 8D30h, ADC_B_NS.ADLIMCHSR1 0743 8D30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	LIMCHF35	LIMCHF34	LIMCHF33	LIMCHF32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LIMCHF32 to LIMCHF35	Analog Channel n: Limiter Clip Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Limiter clip is not detected 1: Limiter clip is detected	R
b31 to b4	—	Reserved	These bits are read as 0.	R

n = 32 to 35

The ADLIMCHSR1 register indicates whether the limiter clip occurred when the A/D conversion for the analog channel n is performed. Each flag can be cleared in ADLIMCHSCR1.

60.2.66 Extended Analog Limiter Clip Status Register (ADLIMEXSR)

Address(es): ADC_B.ADLIMEXSR 8743 8D38h, ADC_B_NS.ADLIMEXSR 0743 8D38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	LIMEXF18	LIMEXF17	LIMEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LIMEXF9	LIMEXF8	—	LIMEXF6	LIMEXF5	LIMEXF4	—	LIMEXF2	LIMEXF1	LIMEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LIMEXF0	ADC0 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b1	LIMEXF1	ADC1 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b2	LIMEXF2	ADC2 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	LIMEXF4	Temperature Sensor Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b5	LIMEXF5	Internal Reference Voltage Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b6	LIMEXF6	VBATT1/6 Voltage Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b7	—	Reserved	This bit is read as 0.	R
b8	LIMEXF8	D/A Converter 0 Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b9	LIMEXF9	D/A Converter 1 Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b16	LIMEXF16	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b17	LIMEXF17	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b18	LIMEXF18	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Self-Diagnosis Channel: Limiter Clip Flag	0: Limiter clip is not detected 1: Limiter clip is detected	R
b31 to b19	—	Reserved	These bits are read as 0.	R

The ADLIMEXSR register indicates whether the limiter clip occurred when the A/D conversion for the extended analog function (which channel no. is 96 to 98, 100 to 102, 104, 105, 112 to 114) is performed. Each flag can be cleared in the ADLIMEXSCR.

60.2.67 Limiter Clip Scan Group Status Clear Register (ADLIMGRSCR)

Address(es): ADC_B.ADLIMGRSCR 8743 8D3Ch, ADC_B_NS.ADLIMGRSCR 0743 8D3Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LIMGRC11	LIMGRC10	LIMGRC9	LIMGRC8	LIMGRC7	LIMGRC6	LIMGRC5	LIMGRC4	LIMGRC3	LIMGRC2	LIMGRC1	LIMGRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	LIMGRC0 to LIMGRC11	Scan Group n Limiter Clip Flag Clear	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADLIMGRSR.LIMGRFn is cleared	W
b31 to b12	—	Reserved	The write value should be 0.	W

n = 0 to 11

The ADLIMGRSCR register clears the limiter clip flag for scan group n (ADLIMGRSR.LIMGRFn).

60.2.68 Limiter Clip Channel Status Clear Register 0 (ADLIMCHSCR0)

Address(es): ADC_B.ADLIMCHSCR0 8743 8D40h, ADC_B_NS.ADLIMCHSCR0 0743 8D40h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LIMCHC31	LIMCHC30	LIMCHC29	LIMCHC28	LIMCHC27	LIMCHC26	LIMCHC25	LIMCHC24	LIMCHC23	LIMCHC22	LIMCHC21	LIMCHC20	LIMCHC19	LIMCHC18	LIMCHC17	LIMCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LIMCHC15	LIMCHC14	LIMCHC13	LIMCHC12	LIMCHC11	LIMCHC10	LIMCHC9	LIMCHC8	LIMCHC7	LIMCHC6	LIMCHC5	LIMCHC4	LIMCHC3	LIMCHC2	LIMCHC1	LIMCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	LIMCHC0 to LIMCHC31	Analog Channel n Limiter Clip Flag Clear bit	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADLIMCHSR0.LIMCHFn is cleared	W

n = 0 to 31

The ADLIMCHSCR0 register clears the limiter clip flag for the analog channel n (ADLIMCHSR0.LIMCHFn).

60.2.69 Limiter Clip Channel Status Clear Register 1 (ADLIMCHSCR1)

Address(es): ADC_B.ADLIMCHSCR1 8743 8D44h, ADC_B_NS.ADLIMCHSCR1 0743 8D44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	LIMCHC35	LIMCHC34	LIMCHC33	LIMCHC32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LIMCHC32 to LIMCHC35	Analog Channel n Limiter Clip Flag Clear bit	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADLIMCHSR1.LIMCHFn is cleared	W
b31 to b4	—	Reserved	The write value should be 0.	W

n = 32 to 35

The ADLIMCHSCR1 register clears the limiter clip flag for the analog channel n (ADLIMCHSR1.LIMCHFn).

60.2.70 Extended Analog Limiter Clip Status Clear Register (ADLIMEXSCR)

Address(es): ADC_B.ADLIMEXSCR 8743 8D4Ch, ADC_B_NS.ADLIMEXSCR 0743 8D4Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	LIMEXC18	LIMEXC17	LIMEXC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LIMEXC9	LIMEXC8	—	LIMEXC6	LIMEXC5	LIMEXC4	—	LIMEXC2	LIMEXC1	LIMEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LIMEXC0	ADC0 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF0 is cleared	W
b1	LIMEXC1	ADC1 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF1 is cleared	W
b2	LIMEXC2	ADC2 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF2 is cleared	W
b3	—	Reserved	The write value should be 0.	W
b4	LIMEXC4	Temperature Sensor Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF4 is cleared	W
b5	LIMEXC5	Internal Reference Voltage Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF5 is cleared	W
b6	LIMEXC6	VBATT1/6 Voltage Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF6 is cleared	W
b7	—	Reserved	The write value should be 0.	W
b8	LIMEXC8	D/A Converter 0 Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF8 is cleared	W
b9	LIMEXC9	D/A Converter 1 Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF9 is cleared	W
b15 to b10	—	Reserved	The write value should be 0.	W
b16	LIMEXC16	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF16 is cleared	W
b17	LIMEXC17	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF17 is cleared	W
b18	LIMEXC18	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Self-Diagnosis Channel: Limiter Clip Flag Clear	0: No effect 1: ADLIMEXSR.LIMEXF18 is cleared	W
b31 to b19	—	Reserved	The write value should be 0.	W

The ADLIMEXSCR register clears the limiter clip flag for the extended analog function.

60.2.71 Compare Match Enable Register (ADCMPENR)

Address(es): ADC_B.ADCMPENR 8743 8400h, ADC_B_NS.ADCMPENR 0743 8400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPEN7	CMPEN6	CMPEN5	CMPEN4	CMPEN3	CMPEN2	CMPEN1	CMPEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CMPEN0 to CMPEN7	Compare Match n Enable	The suffix number of each bit symbol corresponds to the compare match number n. 0: Disable the compare match n 1: Enable the compare match n	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The ADCMPENR register enables/disables the compare match n.

60.2.72 Compare Match Interrupt Enable Register (ADCMPINTCR)

Address(es): ADC_B.ADCMPINTCR 8743 8404h, ADC_B_NS.ADCMPINTCR 0743 8404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPIE7	CMPIE6	CMPIE5	CMPIE4	CMPIE3	CMPIE2	CMPIE1	CMPIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CMPIE0 to CMPIE7	Compare Match Interrupt n Enable	The suffix number of each bit symbol corresponds to the compare match interrupt number n. 0: Disable the compare match interrupt n 1: Enable the compare match interrupt n	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The ADCMPINTCR register enables/disables the compare match interrupt n.

60.2.73 Composite Compare Match Configuration Register n (ADCCMPCRn) (n = 0 to 7)

Address(es): ADC_B.ADCCMPCR0 8743 8408h, ADC_B.ADCCMPCR1 8743 840Ch, ADC_B.ADCCMPCR2 8743 8410h, ADC_B.ADCCMPCR3 8743 8414h, ADC_B.ADCCMPCR4 8743 8418h, ADC_B.ADCCMPCR5 8743 841Ch, ADC_B.ADCCMPCR6 8743 8420h, ADC_B.ADCCMPCR7 8743 8424h, ADC_B_NS.ADCCMPCR0 0743 8408h, ADC_B_NS.ADCCMPCR1 0743 840Ch, ADC_B_NS.ADCCMPCR2 0743 8410h, ADC_B_NS.ADCCMPCR3 0743 8414h, ADC_B_NS.ADCCMPCR4 0743 8418h, ADC_B_NS.ADCCMPCR5 0743 841Ch, ADC_B_NS.ADCCMPCR6 0743 8420h, ADC_B_NS.ADCCMPCR7 0743 8424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CCMPTBL7	CCMPTBL6	CCMPTBL5	CCMPTBL4	CCMPTBL3	CCMPTBL2	CCMPTBL1	CCMPTBL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCMPCND[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CCMPCND[1:0]	Composite Compare Match Condition Selection	b1 b0 0 0: Logical disjunction (OR) conditions Settings other than above are prohibited.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	CCMPTBL0 to CCMPTBL7	Composite Compare Match Condition Table Selection	0: Not use the compare match table m 1: Use the compare match table m	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = 0 to 7

The ADCCMPCRn register configures the conditions for the composite compare match interrupt n.

CCMPCND[1:0] Bits (Composite Compare Match Condition Selection)

The CCMPCND[1:0] bits select the generating condition for the composite compare match interrupt n.

CCMPTBLm Bits (Composite Compare Match Condition Table Selection) (m = 0 to 7)

The CCMPTBLm bits select whether to use or not to use the compare match table m as the generating conditions of the composite compare match interrupt.

60.2.74 Compare Match Mode Selection Register 0 (ADCMPMDR0)

Address(es): ADC_B.ADCMPMDR0 8743 8448h, ADC_B_NS.ADCMPMDR0 0743 8448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	CMPMD3[1:0]		—	—	—	—	—	—	CMPMD2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPMD1[1:0]		—	—	—	—	—	—	CMPMD0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMPMD0[1:0]	Compare Match 0: Match Mode Selection	b1 b0 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	CMPMD1[1:0]	Compare Match 1: Match Mode Selection	b9 b8 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CMPMD2[1:0]	Compare Match 2: Match Mode Selection	b17 b16 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	CMPMD3[1:0]	Compare Match 3: Match Mode Selection	b25 b24 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR0 register selects the generating condition of each compare match event.

60.2.75 Compare Match Mode Selection Register 1 (ADCMPMDR1)

Address(es): ADC_B.ADCMPMDR1 8743 844Ch, ADC_B_NS.ADCMPMDR1 0743 844Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	CMPMD7[1:0]		—	—	—	—	—	—	CMPMD6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPMD5[1:0]		—	—	—	—	—	—	CMPMD4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMPMD4[1:0]	Compare Match 4: Match Mode Selection	b1 b0 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	CMPMD5[1:0]	Compare Match 5: Match Mode Selection	b9 b8 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CMPMD6[1:0]	Compare Match 6: Match Mode Selection	b17 b16 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	CMPMD7[1:0]	Compare Match 7: Match Mode Selection	b25 b24 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR1 register selects the generating condition of each compare match event.

60.2.76 Compare Match Table Register n (ADCMPTBRn) (n = 0 to 7)

Address(es): ADC_B.ADCMPTBR0 8743 8458h, ADC_B.ADCMPTBR1 8743 845Ch, ADC_B.ADCMPTBR2 8743 8460h, ADC_B.ADCMPTBR3 8743 8464h, ADC_B.ADCMPTBR4 8743 8468h, ADC_B.ADCMPTBR5 8743 846Ch, ADC_B.ADCMPTBR6 8743 8470h, ADC_B.ADCMPTBR7 8743 8474h, ADC_B_NS.ADCMPTBR0 0743 8458h, ADC_B_NS.ADCMPTBR1 0743 845Ch, ADC_B_NS.ADCMPTBR2 0743 8460h, ADC_B_NS.ADCMPTBR3 0743 8464h, ADC_B_NS.ADCMPTBR4 0743 8468h, ADC_B_NS.ADCMPTBR5 0743 846Ch, ADC_B_NS.ADCMPTBR6 0743 8470h, ADC_B_NS.ADCMPTBR7 0743 8474h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMPTBH[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPTBL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CMPTBL[15:0]	Compare Match Table n: Low-Side Level	Set CMPTBH > CMPTBL	R/W
b31 to b16	CMPTBH[15:0]	Compare Match Table n: High-Side Level	Set CMPTBH > CMPTBL	R/W

The ADCMPTBRn register specifies the low-side level and the high-side level for the compare match table n.

60.2.77 Compare Match Table Status Register (ADCMPTBSR)

Address(es): ADC_B.ADCMPTBSR 8743 8D00h, ADC_B_NS.ADCMPTBSR 0743 8D00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPTBF7	CMPTBF6	CMPTBF5	CMPTBF4	CMPTBF3	CMPTBF2	CMPTBF1	CMPTBF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CMPTBF0 to CMPTBF7	Compare Match Table n Match Flag	The suffix number of each bit symbol corresponds to the compare match table number n. 0: Match event with compare match table n is not detected 1: Match event with compare match table n is detected	R
b31 to b8	—	Reserved	These bits are read as 0.	R

n = 0 to 7

The ADCMPTBSR register indicates whether the match event with the compare match table n occurred during the A/D conversion. Each flag can be cleared in the ADCMPTBSCR.

60.2.78 Compare Match Table Status Clear Register (ADCMPTBSCR)

Address(es): ADC_B.ADCMPTBSCR 8743 8D04h, ADC_B_NS.ADCMPTBSCR 0743 8D04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPTBC7	CMPTBC6	CMPTBC5	CMPTBC4	CMPTBC3	CMPTBC2	CMPTBC1	CMPTBC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CMPTBC0 to CMPTBC7	Compare Match Table n: Match Flag Clear	The suffix number of each bit symbol corresponds to the compare match table number n. 0: No effect 1: ADCMPTBSR.CMPTBFn is cleared	W
b31 to b8	—	Reserved	The write value should be 0.	W

n = 0 to 7

The ADCMPTBSCR register clears the match flag for the compare match n (ADCMPTBSR.CMPTBFn).

60.2.79 Compare Match Channel Status Register 0 (ADCMPCHSR0)

Address(es): ADC_B.ADCMPCHSR0 8743 8D08h, ADC_B_NS.ADCMPCHSR0 0743 8D08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMPCHF31	CMPCHF30	CMPCHF29	CMPCHF28	CMPCHF27	CMPCHF26	CMPCHF25	CMPCHF24	CMPCHF23	CMPCHF22	CMPCHF21	CMPCHF20	CMPCHF19	CMPCHF18	CMPCHF17	CMPCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPCHF15	CMPCHF14	CMPCHF13	CMPCHF12	CMPCHF11	CMPCHF10	CMPCHF9	CMPCHF8	CMPCHF7	CMPCHF6	CMPCHF5	CMPCHF4	CMPCHF3	CMPCHF2	CMPCHF1	CMPCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	CMPCHF0 to CMPCHF31	Analog Channel n: Compare Match Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Compare match is not detected 1: Compare match is detected	R

n = 0 to 31

The ADCMPCHSR0 register indicates whether the compare match event for the analog channel n is detected. Each flag can be cleared in ADCMPCHSCR0.

60.2.80 Compare Match Channel Status Register 1 (ADCMPCHSR1)

Address(es): ADC_B.ADCMPCHSR1 8743 8D0Ch, ADC_B_NS.ADCMPCHSR1 0743 8D0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPCHF35	CMPCHF34	CMPCHF33	CMPCHF32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPCHF32 to CMPCHF35	Analog Channel n: Compare Match Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Compare match is not detected 1: Compare match is detected	R
b31 to b4	—	Reserved	These bits are read as 0.	R

n = 32 to 35

The ADCMPCHSR1 register indicates whether the compare match event for the analog channel n is detected. Each flag can be cleared in ADCMPCHSCR1.

60.2.81 Extended Analog Compare Match Status Register (ADCMPEXSR)

Address(es): ADC_B.ADCMPEXSR 8743 8D14h, ADC_B_NS.ADCMPEXSR 0743 8D14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPEXF18	CMPEXF17	CMPEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPEXF9	CMPEXF8	—	CMPEXF6	CMPEXF5	CMPEXF4	—	CMPEXF2	CMPEXF1	CMPEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPEXF0	ADC0 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b1	CMPEXF1	ADC1 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b2	CMPEXF2	ADC2 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	CMPEXF4	Temperature Sensor Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b5	CMPEXF5	Internal Reference Voltage Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b6	CMPEXF6	VBATT 1/6 Voltage Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b7	—	Reserved	This bit is read as 0.	R
b8	CMPEXF8	D/A Converter 0 Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b9	CMPEXF9	D/A Converter 1 Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b16	CMPEXF16	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b17	CMPEXF17	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b18	CMPEXF18	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Self-Diagnosis Channel: Compare Match Flag	0: Compare match is not detected 1: Compare match is detected	R
b31 to b19	—	Reserved	These bits are read as 0.	R

The ADCMPEXSR register indicates whether the compare match event for the extended analog function channel is detected. Each flag can be cleared in ADCMPEXSCR.

60.2.82 Compare Match Channel Status Clear Register 0 (ADCMPCHSCR0)

Address(es): ADC_B.ADCMPCHSCR0 8743 8D18h, ADC_B_NS.ADCMPCHSCR0 0743 8D18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMPCHC31	CMPCHC30	CMPCHC29	CMPCHC28	CMPCHC27	CMPCHC26	CMPCHC25	CMPCHC24	CMPCHC23	CMPCHC22	CMPCHC21	CMPCHC20	CMPCHC19	CMPCHC18	CMPCHC17	CMPCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPCHC15	CMPCHC14	CMPCHC13	CMPCHC12	CMPCHC11	CMPCHC10	CMPCHC9	CMPCHC8	CMPCHC7	CMPCHC6	CMPCHC5	CMPCHC4	CMPCHC3	CMPCHC2	CMPCHC1	CMPCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	CMPCHC0 to CMPCHC31	Analog Channel n: Compare Match Flag Clear	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADCMPCHSR0.CMPCHFn is cleared	W

n = 0 to 31

The ADCMPCHSCR0 register clears the compare match flag for the analog channel n (ADCMPCHSR0.CMPCHFn).

60.2.83 Compare Match Channel Status Clear Register 1 (ADCMPCHSCR1)

Address(es): ADC_B.ADCMPCHSCR1 8743 8D1Ch, ADC_B_NS.ADCMPCHSCR1 0743 8D1Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPCHC35	CMPCHC34	CMPCHC33	CMPCHC32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPCHC32 to CMPCHC35	Analog Channel n: Compare Match Flag Clear	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADCMPCHSR1.CMPCHFn is cleared	W
b31 to b4	—	Reserved	The write value should be 0.	W

n = 32 to 35

The ADCMPCHSCR1 register clears the compare match flag for the analog channel n (ADCMPCHSR1.CMPCHFn).

60.2.84 Extended Analog Compare Match Status Clear Register (ADCMPEXSCR)

Address(es): ADC_B.ADCMPEXSCR 8743 8D24h, ADC_B_NS.ADCMPEXSCR 0743 8D24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPEXC18	CMPEXC17	CMPEXC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPEXC9	CMPEXC8	—	CMPEXC6	CMPEXC5	CMPEXC4	—	CMPEXC2	CMPEXC1	CMPEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPEXC0	ADC0 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF0 is cleared	W
b1	CMPEXC1	ADC1 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF1 is cleared	W
b2	CMPEXC2	ADC2 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF2 is cleared	W
b3	—	Reserved	The write value should be 0.	W
b4	CMPEXC4	Temperature Sensor Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF4 is cleared	W
b5	CMPEXC5	Internal Reference Voltage Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF5 is cleared	W
b6	CMPEXC6	VBATT 1/6 Voltage Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF6 is cleared	W
b7	—	Reserved	The write value should be 0.	W
b8	CMPEXC8	D/A Converter 0 Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF8 is cleared	W
b9	CMPEXC9	D/A Converter 1 Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF9 is cleared	W
b15 to b10	—	Reserved	The write value should be 0.	W
b16	CMPEXC16	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF16 is cleared	W
b17	CMPEXC17	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF17 is cleared	W
b18	CMPEXC18	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Self-Diagnosis Channel: Compare Match Flag Clear	0: No effect 1: ADCMPEXSR.CMPEXF18 is cleared	W
b31 to b19	—	Reserved	The write value should be 0.	W

The ADCMPEXSCR register clears the compare match flag for the extended analog function channel (ADCMPEXSR.CMPEXFn).

60.2.85 A/D Converter Self-Calibration Start Register (ADCALSTR)

Address(es): ADC_B.ADCALSTR 8743 8C00h, ADC_B_NS.ADCALSTR 0743 8C00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCALST2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ADCALST1[1:0]		—	—	—	—	—	ADCALST0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADCALST0[2:0]*1	A/D Converter Unit 0 (ADC0) Self-Calibration Start Control	[Function of each bit] b0: Start bit of the internal circuit calibration b1: Start bit of the gain and offset calibration b2: Start bit of the channel-dedicated sample-and-hold circuit calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
b7 to b3	—	Reserved	The write value should be 0.	W
b9 to b8	ADCALST1[1:0]*1	A/D Converter Unit 1 (ADC1) Self-Calibration Start Control	[Function of each bit] b8: Start bit of the internal circuit calibration b9: Start bit of the gain and offset calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
b15 to b10	—	Reserved	The write value should be 0.	W
b17 to b16	ADCALST2[1:0]*1	A/D Converter Unit 2 (ADC2) Self-Calibration Start Control	[Function of each bit] b16: Start bit of the internal circuit calibration b17: Start bit of the gain and offset calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
b31 to b18	—	Reserved	The write value should be 0.	W

Note 1. To perform a calibration operation, write 1 to each of the calibration start bits simultaneously.

The ADCALSTR register controls the start of the self-calibration for each A/D converter.

ADCALST0[2:0] Bits (A/D Converter Unit 0 (ADC0) Self-Calibration Start Control)

The ADCALST0[2:0] bits control the start of the self-calibration operation of the A/D converter Unit 0. When any bit of ADCALST0[2:0] is set to 1, the self-calibration operation corresponding to each bit is started. When multiple bits of ADCALST0[2:0] are set to 1 simultaneously, each self-calibration operation is performed in the order as following:

1. Internal circuit calibration
2. A/D converter gain/offset calibration
3. Channel-specific sample-and-hold circuit gain/offset calibration

Note: The self-calibration operation corresponding to the bits set to 0 in ADCALST0[2:0] is skipped.

The ADCALST0[2:0] bits must be written when all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0 to 2));
Writing to the ADCALST0[2:0] bits are prohibited when A/D converters are operating.

ADCALST1[1:0] Bits (A/D Converter Unit 1 (ADC1) Self-Calibration Start Control)

The ADCALST1[1:0] bits control the start of the self-calibration operation of the A/D converter Unit 1. When any bit of ADCALST1[1:0] is set to 1, the self-calibration operation corresponding to each bit is started. When multiple bits of ADCALST1[1:0] are set to 1 simultaneously, each self-calibration operation is performed in the order as following:

1. Internal circuit calibration
2. A/D converter gain/offset calibration

Note: The self-calibration operation corresponding to the bits set to 0 in ADCALST1[1:0] is skipped.

The ADCALST1[1:0] bits must be written when all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0 to 2));
Writing to the ADCALST1[1:0] bits are prohibited when A/D converters are operating.

ADCALST2[1:0] Bits (A/D Converter Unit 2 (ADC2) Self-Calibration Start Control)

The ADCALST2[1:0] bits control the start of the self-calibration operation of the A/D converter Unit 2. When any bit of ADCALST2[1:0] is set to 1, the self-calibration operation corresponding to each bit is started. When multiple bits of ADCALST2[1:0] are set to 1 simultaneously, each self-calibration operation is performed in the order as following:

1. Internal circuit calibration
2. A/D converter gain/offset calibration

Note: The self-calibration operation corresponding to the bits set to 0 in ADCALST2[1:0] is skipped.

The ADCALST2[1:0] bits must be written when all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0 to 2));
Writing to the ADCALST2[1:0] bits are prohibited when A/D converters are operating.

60.2.86 A/D Conversion Synchronous Software Start Register (ADSYSTR)

Address(es): ADC_B.ADSYSTR 8743 8C10h, ADC_B_NS.ADSYSTR 0743 8C10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADSYST11	ADSYST10	ADSYST9	ADSYST8	ADSYST7	ADSYST6	ADSYST5	ADSYST4	ADSYST3	ADSYST2	ADSYST1	ADSYST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	ADSYST0 to ADSYST11	Scan Group n: A/D Conversion Start	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Start the A/D conversion of scan group n	W
b31 to b12	—	Reserved	The write value should be 0.	W

n = 0 to 11

The ADSYSTR register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversions of several scan groups simultaneously by software.

ADSYSTn Bit (Scan Group n: A/D Conversion Start) (n = 0 to 11)

The ADSYSTn bit controls the start of the A/D conversion of scan group n. When the ADSYSTn bit is set to 1, the A/D conversion of scan group n is started. Setting 0 to the ADSYSTn bit has no effect on the operation. If the ADSYSTn bits for multiple scan groups are set to 1 at the same time, A/D conversion begins in sequence according to the priority levels of the scan groups.

60.2.87 A/D Conversion Software Start Register n (ADSTRn) (n = 0 to 11)

Address(es): ADC_B.ADSTR0 8743 8C20h, ADC_B.ADSTR1 8743 8C24h, ADC_B.ADSTR2 8743 8C28h, ADC_B.ADSTR3 8743 8C2Ch, ADC_B.ADSTR4 8743 8C30h, ADC_B.ADSTR5 8743 8C34h, ADC_B.ADSTR6 8743 8C38h, ADC_B.ADSTR7 8743 8C3Ch, ADC_B.ADSTR8 8743 8C40h, ADC_B.ADSTR9 8743 8C44h, ADC_B.ADSTR10 8743 8C48h, ADC_B.ADSTR11 8743 8C4Ch, ADC_B_NS.ADSTR0 0743 8C20h, ADC_B_NS.ADSTR1 0743 8C24h, ADC_B_NS.ADSTR2 0743 8C28h, ADC_B_NS.ADSTR3 0743 8C2Ch, ADC_B_NS.ADSTR4 0743 8C30h, ADC_B_NS.ADSTR5 0743 8C34h, ADC_B_NS.ADSTR6 0743 8C38h, ADC_B_NS.ADSTR7 0743 8C3Ch, ADC_B_NS.ADSTR8 0743 8C40h, ADC_B_NS.ADSTR9 0743 8C44h, ADC_B_NS.ADSTR10 0743 8C48h, ADC_B_NS.ADSTR11 0743 8C4Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADST	Scan Group n A/D Conversion Start	0: No effect 1: Start the A/D conversion of scan group n	W
b31 to b1	—	Reserved	The write value should be 0.	W

The ADSTRn register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversion of a scan group by software.

If A/D conversions of several scan groups are to be started simultaneously by software, the ADSYSTR register should be used.

ADST Bit (Scan Group n A/D Conversion Start)

The ADST bit controls the start of the A/D conversion of scan group n. When the ADST bit is set to 1, the A/D conversion of the scan group n is started. Setting 0 to the ADST bit has no effect on the operation.

60.2.88 A/D Conversion Stop Register (ADSTOPR)

Address(es): ADC_B.ADSTOPR 8743 8C60h, ADC_B_NS.ADSTOPR 0743 8C60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTOP2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADSTOP1	—	—	—	—	—	—	—	ADSTOP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADSTOP0	A/D Converter Unit 0 Force Stop	0: No effect 1: Force stop the operation of A/D converter unit 0	W
b7 to b1	—	Reserved	The write value should be 0.	W
b8	ADSTOP1	A/D Converter Unit 1 Force Stop	0: No effect 1: Force stop the operation of A/D converter unit 1	W
b15 to b9	—	Reserved	The write value should be 0.	W
b16	ADSTOP2	A/D Converter Unit 2 Force Stop	0: No effect 1: Force stop the operation of A/D converter unit 2	W
b31 to b17	—	Reserved	The write value should be 0.	W

The ADSTOPR register forces each A/D converter to stop the operation. When the A/D converter is stopped with this register, the A/D conversion results with that A/D converter are not guaranteed.

ADSTOP0 Bit (A/D Converter Unit 0 Force Stop)

The ADSTOP0 bit forces the A/D converter unit 0 to stop the A/D conversion operation. Setting 0 to the ADSTOP0 bit has no effect on the operation. When the ADSTOP0 bit is set to 1, the A/D converter unit 0 is forced to stop the operation. If the A/D converter unit 0 is forced to stop, the A/D conversion results with the A/D converter unit 0 are not guaranteed.

ADSTOP1 Bit (A/D Converter Unit 1 Force Stop)

The ADSTOP1 bit forces the A/D converter unit 1 to stop the A/D conversion operation. Setting 0 to the ADSTOP1 bit has no effect on the operation. When the ADSTOP1 bit is set to 1, the A/D converter unit 1 is forced to stop the operation. If the A/D converter unit 1 is forced to stop, the A/D conversion results with the A/D converter unit 1 are not guaranteed.

ADSTOP2 Bit (A/D Converter Unit 2 Force Stop)

The ADSTOP2 bit forces the A/D converter unit 2 to stop the A/D conversion operation. Setting 0 to the ADSTOP2 bit has no effect on the operation. When the ADSTOP2 bit is set to 1, the A/D converter unit 2 is forced to stop the operation. If the A/D converter unit 2 is forced to stop, the A/D conversion results with the A/D converter unit 2 are not guaranteed.

60.2.89 A/D Conversion Status Register (ADSR)

Address(es): ADC_B.ADSR 8743 8C80h, ADC_B_NS.ADSR 0743 8C80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	CALACT2	CALACT1	CALACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—		ADACT2	ADACT1	ADACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADACT0	A/D Converter Unit 0 (ADC0) A/D Conversion Status	0: ADC0 is not in A/D conversion 1: ADC0 is in A/D conversion	R
b1	ADACT1	A/D Converter Unit 1 (ADC1) A/D Conversion Status	0: ADC1 is not in A/D conversion 1: ADC1 is in A/D conversion	R
b2	ADACT2	A/D Converter Unit 2 (ADC2) A/D Conversion Status	0: ADC2 is not in A/D conversion 1: ADC2 is in A/D conversion	R
b15 to b3	—	Reserved	These bits are read as 0.	R
b16	CALACT0	A/D Converter Unit 0 (ADC0): Calibration Status	0: ADC0 is not in the calibration operation 1: ADC0 is in the calibration operation	R
b17	CALACT1	A/D Converter Unit 1 (ADC1): Calibration Status	0: ADC1 is not in the calibration operation 1: ADC1 is in the calibration operation	R
b18	CALACT2	A/D Converter Unit 2 (ADC2): Calibration Status	0: ADC2 is not in the calibration operation 1: ADC2 is in the calibration operation	R
b31 to b19	—	Reserved	These bits are read as 0.	R

The ADSR register indicates the status for each A/D converter operation.

60.2.90 Scan Group Status Register (ADGRSR)

Address(es): ADC_B.ADGRSR 8743 8C84h, ADC_B_NS.ADGRSR 0743 8C84h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ACTGR11	ACTGR10	ACTGR9	ACTGR8	ACTGR7	ACTGR6	ACTGR5	ACTGR4	ACTGR3	ACTGR2	ACTGR1	ACTGR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	ACTGR0 to ACTGR11	Scan Group n Status*1	The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n is idle 1: Scan group n is in the scanning operation	R
b31 to b12	—	Reserved	These bits are read as 0.	R

n = 0 to 11

Note 1. In the group priority operation, if the scanning operation of a low priority group is interrupted, the ACTGRn bit of the corresponding scan group is set to 1.

The ADGRSR register indicates the operating status for each scan group.

60.2.91 Scan End Status Register (ADSCANENDSR)

Address(es): ADC_B.ADSCANENDSR 8743 8D50h, ADC_B_NS.ADSCANENDSR 0743 8D50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SCENDF11	SCENDF10	SCENDF9	SCENDF8	SCENDF7	SCENDF6	SCENDF5	SCENDF4	SCENDF3	SCENDF2	SCENDF1	SCENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	SCENDF0 to SCENDF11	Scan Group n Scan End Flag*1	The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n has not been scanned 1: End of scan for scan group n is detected	R
b31 to b12	—	Reserved	These bits are read as 0.	R

n = 0 to 11

Note 1. If the A/D conversion operation is stopped with the ADSTOPR register, the SCENDFn bit of the scan group in which the scan operation was stopped is not changed. (It is not set to 1)

The ADSCANENDSR register indicates whether the scanning operation for each scan group has been done. Each flag can be cleared in ADSCANENDSCR register.

60.2.92 Scan End Status Clear Register (ADSCANENDSCR)

Address(es): ADC_B.ADSCANENDSCR 8743 8D54h, ADC_B_NS.ADSCANENDSCR 0743 8D54h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SCENDC11	SCENDC10	SCENDC9	SCENDC8	SCENDC7	SCENDC6	SCENDC5	SCENDC4	SCENDC3	SCENDC2	SCENDC1	SCENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	SCENDC0 to SCENDC11	Scan Group n Scan End Flag Clear	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADSCANENDSR.SCENDFn is cleared	W
b31 to b12	—	Reserved	The write value should be 0.	W

n = 0 to 11

The ADSCANENDSCR register clears the scan end flag for the scan group n.

60.2.93 A/D Conversion Error Status Register (ADERSR)

Address(es): ADC_B.ADERSR 8743 8C88h, ADC_B_NS.ADERSR 0743 8C88h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERF2	ADERF1	ADERF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADERF0	A/D Converter Unit 0 (ADC0) Error Flag	0: Error is not detected 1: Error is detected	R
b1	ADERF1	A/D Converter Unit 1 (ADC1) Error Flag	0: Error is not detected 1: Error is detected	R
b2	ADERF2	A/D Converter Unit 2 (ADC2) Error Flag	0: Error is not detected 1: Error is detected	R
b31 to b3	—	Reserved	These bits are read as 0.	R

The ADERSR register indicates whether the operating error has occurred in each A/D converter. Each flag can be cleared in ADERSCR register.
The A/D conversion data is not guaranteed in the A/D converter operation with an error detected.

60.2.94 A/D Conversion Error Status Clear Register (ADERSCR)

Address(es): ADC_B.ADERSCR 8743 8C8Ch, ADC_B_NS.ADERSCR 0743 8C8Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERCLR2	ADERCLR1	ADERCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADERCLR0	A/D Converter Unit 0 Error Flag Clear	0: No effect 1: ADERSR.ADERF0 is cleared	W
b1	ADERCLR1	A/D Converter Unit 1 Error Flag Clear	0: No effect 1: ADERSR.ADERF1 is cleared	W
b2	ADERCLR2	A/D Converter Unit 2 Error Flag Clear	0: No effect 1: ADERSR.ADERF2 is cleared	W
b31 to b3	—	Reserved	The write value should be 0.	W

The ADERSCR register clears the error flag for A/D converter unit 0, unit 1 or unit 2.

60.2.95 A/D Converter Calibration End Status Register (ADCALENDSR)

Address(es): ADC_B.ADCALENDSR 8743 8C98h, ADC_B_NS.ADCALENDSR 0743 8C98h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDF2	CALENDF1	CALENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CALENDF0	A/D Converter Unit 0 Calibration End Flag	0: End of the calibration is not detected 1: End of the calibration is detected	R
b1	CALENDF1	A/D Converter Unit 1 Calibration End Flag	0: End of the calibration is not detected 1: End of the calibration is detected	R
b2	CALENDF2	A/D Converter Unit 2 Calibration End Flag	0: End of the calibration is not detected 1: End of the calibration is detected	R
b31 to b3	—	Reserved	These bits are read as 0.	R

The ADCALENDSR register indicates the end of the calibration operation for the A/D converter unit 0, unit 1 or unit 2. Each flag can be cleared in ADCALENDSCR register.

60.2.96 A/D Converter Calibration End Status Clear Register (ADCALENDSCR)

Address(es): ADC_B.ADCALENDSCR 8743 8C9Ch, ADC_B_NS.ADCALENDSCR 0743 8C9Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDC2	CALENDC1	CALENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CALENDC0	A/D Converter Unit 0 Calibration End Flag Clear	0: No effect 1: ADCALENDSR.CALENDF0 is cleared	W
b1	CALENDC1	A/D Converter Unit 1 Calibration End Flag Clear	0: No effect 1: ADCALENDSR.CALENDF1 is cleared	W
b2	CALENDC2	A/D Converter Unit 2 Calibration End Flag Clear	0: No effect 1: ADCALENDSR.CALENDF2 is cleared	W
b31 to b3	—	Reserved	The write value should be 0.	W

The ADCALENDSCR register clears the calibration end flag for the A/D converter unit 0, unit 1 or unit 2.

60.2.97 A/D Conversion Overflow Error Status Register (ADOVFERSR)

Address(es): ADC_B.ADOVFERSR 8743 8CA0h, ADC_B_NS.ADOVFERSR 0743 8CA0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEF2	ADOVFEF1	ADOVFEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADOVFEF0	A/D Converter Unit 0 (ADC0) Overflow Error Flag	0: ADC0 overflow error is not detected 1: ADC0 overflow error is detected	R
b1	ADOVFEF1	A/D Converter Unit 1 (ADC1) Overflow Error Flag	0: ADC1 overflow error is not detected 1: ADC1 overflow error is detected	R
b2	ADOVFEF2	A/D Converter Unit 2 (ADC2) Overflow Error Flag	0: ADC2 overflow error is not detected 1: ADC2 overflow error is detected	R
b31 to b3	—	Reserved	These bits are read as 0.	R

The ADOVFERSR register indicates whether the overflow error has occurred in an A/D conversion using A/D converter unit 0, unit 1 or unit 2. Each flag can be cleared in ADOVFERSCR register.

60.2.98 A/D Conversion Overflow Channel Status Register 0 (ADOVFCHSR0)

Address(es): ADC_B.ADOVFCHSR0 8743 8CA4h, ADC_B_NS.ADOVFCHSR0 0743 8CA4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	OVFCHF31	OVFCHF30	OVFCHF29	OVFCHF28	OVFCHF27	OVFCHF26	OVFCHF25	OVFCHF24	OVFCHF23	OVFCHF22	OVFCHF21	OVFCHF20	OVFCHF19	OVFCHF18	OVFCHF17	OVFCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVFCHF15	OVFCHF14	OVFCHF13	OVFCHF12	OVFCHF11	OVFCHF10	OVFCHF9	OVFCHF8	OVFCHF7	OVFCHF6	OVFCHF5	OVFCHF4	OVFCHF3	OVFCHF2	OVFCHF1	OVFCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	OVFCHF0 to OVFCHF31	Analog Channel n: Overflow Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Overflow is not detected 1: Overflow is detected	R

n = 0 to 31

The ADOVFCHSR0 register indicates whether the overflow has occurred in the A/D conversion of the analog channel n. Each flag can be cleared in ADOVFCHSCR0 register.

60.2.99 A/D Conversion Overflow Channel Status Register 1 (ADOVFCHSR1)

Address(es): ADC_B.ADOVFCHSR1 8743 8CA8h, ADC_B_NS.ADOVFCHSR1 0743 8CA8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	OVFCHF35	OVFCHF34	OVFCHF33	OVFCHF32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	OVFCHF32 to OVFCHF35	Analog Channel n: Overflow Flag	The suffix number of each bit symbol corresponds to the analog channel number n. 0: Overflow is not detected 1: Overflow is detected	R
b31 to b4	—	Reserved	These bits are read as 0.	R

n = 32 to 35

The ADOVFCHSR1 register indicates whether the overflow has occurred in the A/D conversion of the analog channel n. Each flag can be cleared in ADOVFCHSCR1 register.

60.2.100 Extended Analog A/D Conversion Overflow Status Register (ADOVFEXSR)

Address(es): ADC_B.ADOVFEXSR 8743 8CB0h, ADC_B_NS.ADOVFEXSR 0743 8CB0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OVFEXF18	OVFEXF17	OVFEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OVFEXF9	OVFEXF8	—	OVFEXF6	OVFEXF5	OVFEXF4	—	OVFEXF2	OVFEXF1	OVFEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVFEXF0	ADC0 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b1	OVFEXF1	ADC1 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b2	OVFEXF2	ADC2 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	OVFEXF4	Temperature Sensor Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b5	OVFEXF5	Internal Reference Voltage Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b6	OVFEXF6	VBATT 1/6 Voltage Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b7	—	Reserved	This bit is read as 0.	R
b8	OVFEXF8	D/A Converter 0 Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b9	OVFEXF9	D/A Converter 1 Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b15 to b10	—	Reserved	These bits are read as 0.	R
b16	OVFEXF16	Channel-dedicated Sample-and-hold Circuit Unit 0 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b17	OVFEXF17	Channel-dedicated Sample-and-hold Circuit Unit 1 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b18	OVFEXF18	Channel-dedicated Sample-and-hold Circuit Unit 2 Self-diagnosis Channel: Overflow Flag	0: Overflow is not detected 1: Overflow is detected	R
b31 to b19	—	Reserved	These bits are read as 0.	R

The ADOVFEXSR register indicates whether the overflow occurred when the A/D conversion for the extended analog function is performed. Each flag can be cleared in ADOVFEXSCR.

60.2.101 A/D Conversion Overflow Error Status Clear Register (ADOVFERSCR)

Address(es): ADC_B.ADOVFERSCR 8743 8CB4h, ADC_B_NS.ADOVFERSCR 0743 8CB4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEC2	ADOVFEC1	ADOVFEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADOVFEC0	A/D Converter Unit 0 Overflow Error Flag Clear	0: No effect 1: ADOVFERSR.ADOVFEF0 is cleared	W
b1	ADOVFEC1	A/D Converter Unit 1 Overflow Error Flag Clear	0: No effect 1: ADOVFERSR.ADOVFEF1 is cleared	W
b2	ADOVFEC2	A/D Converter Unit 2 Overflow Error Flag Clear	0: No effect 1: ADOVFERSR.ADOVFEF2 is cleared	W
b31 to b3	—	Reserved	The write value should be 0.	W

The ADOVFERSCR register clears the overflow error flag for the A/D converter unit 0, unit 1 or unit 2.

60.2.102 A/D Conversion Overflow Channel Status Clear Register 0 (ADOVFCHSCR0)

Address(es): ADC_B.ADOVFCHSCR0 8743 8CB8h, ADC_B_NS.ADOVFCHSCR0 0743 8CB8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	OVFCHC31	OVFCHC30	OVFCHC29	OVFCHC28	OVFCHC27	OVFCHC26	OVFCHC25	OVFCHC24	OVFCHC23	OVFCHC22	OVFCHC21	OVFCHC20	OVFCHC19	OVFCHC18	OVFCHC17	OVFCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVFCHC15	OVFCHC14	OVFCHC13	OVFCHC12	OVFCHC11	OVFCHC10	OVFCHC9	OVFCHC8	OVFCHC7	OVFCHC6	OVFCHC5	OVFCHC4	OVFCHC3	OVFCHC2	OVFCHC1	OVFCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	OVFCHC0 to OVFCHC31	Analog Channel n: Overflow Flag Clear	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADOVFCHSR0.OVFCHFn is cleared	W

n = 0 to 31

The ADOVFCHSCR0 register clears the overflow flag for the analog channel n.

60.2.103 A/D Conversion Overflow Channel Status Clear Register 1 (ADOVFCHSCR1)

Address(es): ADC_B.ADOVFCHSCR1 8743 8CBCh, ADC_B_NS.ADOVFCHSCR1 0743 8CBCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	OVFCHC35	OVFCHC34	OVFCHC33	OVFCHC32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	OVFCHC32 to OVFCHC35	Analog Channel n: Overflow Flag Clear	The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADOVFCHSR1.OVFCHFn is cleared	W
b31 to b4	—	Reserved	The write value should be 0.	W

n = 32 to 35

The ADOVFCHSCR1 register clears the overflow flag for the analog channel n.

60.2.104 Extended Analog A/D Conversion Overflow Status Clear Register (ADOVFEXSCR)

Address(es): ADC_B.ADOVFEXSCR 8743 8CC4h, ADC_B_NS.ADOVFEXSCR 0743 8CC4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OVFEXC18	OVFEXC17	OVFEXC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OVFEXC9	OVFEXC8	—	OVFEXC6	OVFEXC5	OVFEXC4	—	OVFEXC2	OVFEXC1	OVFEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVFEXC0	ADC0 Self-diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF0 is cleared	W
b1	OVFEXC1	ADC1 Self-diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF1 is cleared	W
b2	OVFEXC2	ADC2 Self-diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF2 is cleared	W
b3	—	Reserved	The write value should be 0.	W
b4	OVFEXC4	Temperature Sensor Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF4 is cleared	W
b5	OVFEXC5	Internal Reference Voltage Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF5 is cleared	W
b6	OVFEXC6	VBATT 1/6 Voltage Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF6 is cleared	W
b7	—	Reserved	The write value should be 0.	W
b8	OVFEXC8	D/A Converter 0 Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF8 is cleared	W
b9	OVFEXC9	D/A Converter 1 Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF9 is cleared	W
b15 to b10	—	Reserved	The write value should be 0.	W
b16	OVFEXC16	Channel-Dedicated Sample-and-Hold Circuit Unit 0 Self-Diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF16 is cleared	W
b17	OVFEXC17	Channel-Dedicated Sample-and-Hold Circuit Unit 1 Self-Diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF17 is cleared	W
b18	OVFEXC18	Channel-Dedicated Sample-and-Hold Circuit Unit 2 Self-Diagnosis Channel: Overflow Flag Clear	0: No effect 1: ADOVFEXSR.OVFEXF18 is cleared	W
b31 to b19	—	Reserved	The write value should be 0.	W

The ADOVFEXSCR register clears the overflow flag for the extended analog function channels.

60.2.105 FIFO Control Register (ADFIFOCR)

Address(es): ADC_B.ADFIFOCR 8743 84C0h, ADC_B_NS.ADFIFOCR 0743 84C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	FIFOCE11	FIFOCE10	FIFOCE9	FIFOCE8	FIFOCE7	FIFOCE6	FIFOCE5	FIFOCE4	FIFOCE3	FIFOCE2	FIFOCE1	FIFOCE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FIFOEN11	FIFOEN10	FIFOEN9	FIFOEN8	FIFOEN7	FIFOEN6	FIFOEN5	FIFOEN4	FIFOEN3	FIFOEN2	FIFOEN1	FIFOEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FIFOEN0 to FIFOEN11	Scan Group n FIFO Enable	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO function 1: Enable scan group n FIFO function	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	FIFOCE0 to FIFOCE11	Scan Group n FIFO Clear Enable at Scan Start/Resume	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO clear function 1: Enable scan group n FIFO clear function	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADFIFOCR register enables/disables the FIFO function of scan group n.

FIFOENn Bit (Scan Group n FIFO Enable) (n = 0 to 11)

The FIFOENn bit enables or disables the FIFO function of scan group n. When the FIFOENn bit is set to 1, the FIFO function of scan group n is enabled, and the result of A/D conversion is stored to the FIFO. When the FIFOENn bit is set to 0, the FIFO function of scan group n is disabled.
The A/D conversion result can also be read from the A/D data or extended data registers.

FIFOCEn Bit (Scan Group n FIFO Clear Enable at Scan Start/Resume) (n = 0 to 11)

The FIFOCEn bit controls whether the scan group n FIFO clear function is enabled or disabled. When the FIFOCEn bit is set to 1, FIFO clear for scan group n is enabled and the FIFO is cleared when the scan group n scan starts/resumes. When the FIFOCEn bit is set to 0, FIFO clear for scan group n is disabled.

60.2.106 FIFO Interrupt Control Register (ADFIFOINTCR)

Address(es): ADC_B.ADFIFOINTCR 8743 84C4h, ADC_B_NS.ADFIFOINTCR 0743 84C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FIFOIE11	FIFOIE10	FIFOIE9	FIFOIE8	FIFOIE7	FIFOIE6	FIFOIE5	FIFOIE4	FIFOIE3	FIFOIE2	FIFOIE1	FIFOIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FIFOIE0 to FIFOIE11	Scan Group n FIFO Interrupt Enable	The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO interrupt 1: Enable scan group n FIFO interrupt	R/W
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 11

The ADFIFOINTCR register enables/disables the FIFO data read request interrupt and FIFO overflow interrupt for scan group n.

60.2.107 FIFO Interrupt Generation Level Register 0 (ADFIFOINTLR0)

Address(es): ADC_B.ADFIFOINTLR0 8743 84C8h, ADC_B_NS.ADFIFOINTLR0 0743 84C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV0[4:0]	Scan Group 0 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV1[4:0]	Scan Group 1 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR0 register specifies the generation timing of the FIFO data read request interrupt for the scan group 0 and 1.

60.2.108 FIFO Interrupt Generation Level Register 1 (ADFIFOINTLR1)

Address(es): ADC_B.ADFIFOINTLR1 8743 84CCh, ADC_B_NS.ADFIFOINTLR1 0743 84CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV3[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV2[4:0]	Scan Group 2 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV3[4:0]	Scan Group 3 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR1 register specifies the generation timing of the FIFO data read request interrupt for the scan group 2 and 3.

60.2.109 FIFO Interrupt Generation Level Register 2 (ADFIFOINTLR2)

Address(es): ADC_B.ADFIFOINTLR2 8743 84D0h, ADC_B_NS.ADFIFOINTLR2 0743 84D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV5[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV4[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV4[4:0]	Scan Group 4 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV5[4:0]	Scan Group 5 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR2 register specifies the generation timing of the FIFO data read request interrupt for the scan group 4 and 5.

60.2.110 FIFO Interrupt Generation Level Register 3 (ADFIFOINTLR3)

Address(es): ADC_B.ADFIFOINTLR3 8743 84D4h, ADC_B_NS.ADFIFOINTLR3 0743 84D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV7[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV6[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV6[4:0]	Scan Group 6 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV7[4:0]	Scan Group 7 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR3 register specifies the generation timing of the FIFO data read request interrupt for the scan group 6 and 7.

60.2.111 FIFO Interrupt Generation Level Register 4 (ADFIFOINTLR4)

Address(es): ADC_B.ADFIFOINTLR4 8743 84D8h, ADC_B_NS.ADFIFOINTLR4 0743 84D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV9[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV8[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV8[4:0]	Scan Group 8 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV9[4:0]	Scan Group 9 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR4 register specifies the generation timing of the FIFO data read request interrupt for the scan group 8 and 9.

60.2.112 FIFO Interrupt Generation Level Register 5 (ADFIFOINTLR5)

Address(es): ADC_B.ADFIFOINTLR5 8743 84DCh, ADC_B_NS.ADFIFOINTLR5 0743 84DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV11[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOILV10[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOILV10[4:0]	Scan Group 10 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	FIFOILV11[4:0]	Scan Group 11 FIFO Interrupt Output Timing Setting	These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 15.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR5 register specifies the generation timing of the FIFO data read request interrupt for the scan group 10 and 11.

60.2.113 FIFO Status Register 0 (ADFIFOSR0)

Address(es): ADC_B.ADFIFOSR0 8743 8CD0h, ADC_B_NS.ADFIFOSR0 0743 8CD0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST0[4:0]	Number of Vacant Stages in FIFO for Scan Group 0		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST1[4:0]	Number of Vacant Stages in FIFO for Scan Group 1		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR0 register indicates the number of vacant stages in FIFO for scan group 0 and 1.

60.2.114 FIFO Status Register 1 (ADFIFOSR1)

Address(es): ADC_B.ADFIFOSR1 8743 8CD4h, ADC_B_NS.ADFIFOSR1 0743 8CD4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST3[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST2[4:0]	Number of Vacant Stages in FIFO for Scan Group 2		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST3[4:0]	Number of Vacant Stages in FIFO for Scan Group 3		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR1 register indicates the number of vacant stages in FIFO for scan group 2 and 3.

60.2.115 FIFO Status Register 2 (ADFIFOSR2)

Address(es): ADC_B.ADFIFOSR2 8743 8CD8h, ADC_B_NS.ADFIFOSR2 0743 8CD8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST5[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST4[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST4[4:0]	Number of Vacant Stages in FIFO for Scan Group 4		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST5[4:0]	Number of Vacant Stages in FIFO for Scan Group 5		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR2 register indicates the number of vacant stages in FIFO for scan group 4 and 5.

60.2.116 FIFO Status Register 3 (ADFIFOSR3)

Address(es): ADC_B.ADFIFOSR3 8743 8CDCh, ADC_B_NS.ADFIFOSR3 0743 8CDCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST7[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST6[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST6[4:0]	Number of Vacant Stages in FIFO for Scan Group 6		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST7[4:0]	Number of Vacant Stages in FIFO for Scan Group 7		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR3 register indicates the number of vacant stages in FIFO for scan group 6 and 7.

60.2.117 FIFO Status Register 4 (ADFIFOSR4)

Address(es): ADC_B.ADFIFOSR4 8743 8CE0h, ADC_B_NS.ADFIFOSR4 0743 8CE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST9[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST8[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST8[4:0]	Number of Vacant Stages in FIFO for Scan Group 8		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST9[4:0]	Number of Vacant Stages in FIFO for Scan Group 9		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR4 register indicates the number of vacant stages in FIFO for scan group 8 and 9.

60.2.118 FIFO Status Register 5 (ADFIFOSR5)

Address(es): ADC_B.ADFIFOSR5 8743 8CE4h, ADC_B_NS.ADFIFOSR5 0743 8CE4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	FIFOST11[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIFOST10[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	FIFOST10[4:0]	Number of Vacant Stages in FIFO for Scan Group 10		R
b15 to b5	—	Reserved	These bits are read as 0.	R
b20 to b16	FIFOST11[4:0]	Number of Vacant Stages in FIFO for Scan Group 11		R
b31 to b21	—	Reserved	These bits are read as 0.	R

The ADFIFOSR5 register indicates the number of vacant stages in FIFO for scan group 10 and 11.

60.2.119 FIFO Data Clear Register (ADFIFODCR)

Address(es): ADC_B.ADFIFODCR 8743 8CF0h, ADC_B_NS.ADFIFODCR 0743 8CF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FIFODC11	FIFODC10	FIFODC9	FIFODC8	FIFODC7	FIFODC6	FIFODC5	FIFODC4	FIFODC3	FIFODC2	FIFODC1	FIFODC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FIFODC0 to FIFODC11	Scan Group n FIFO Data Clear	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Clear the data of scan group n FIFO	W
b31 to b12	—	Reserved	The write value should be 0.	W

n = 0 to 11

The ADFIFODCR register clears the data in FIFO for each scan group.

FIFODCn Bit (Scan Group n FIFO Data Clear) (n = 0 to 11)

The FIFODCn bit clears the data in FIFO for scan group n. Writing 1 to the FIFODCn clears the data in the FIFO of scan group n. Writing 0 has no effect on the operation, and the data in FIFO is not cleared

60.2.120 FIFO Error Status Register (ADFIFOERSR)

Address(es): ADC_B.ADFIFOERSR 8743 8CF4h, ADC_B_NS.ADFIFOERSR 0743 8CF4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	FIFOFLF11	FIFOFLF10	FIFOFLF9	FIFOFLF8	FIFOFLF7	FIFOFLF6	FIFOFLF5	FIFOFLF4	FIFOFLF3	FIFOFLF2	FIFOFLF1	FIFOFLF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FIFOOVF11	FIFOOVF10	FIFOOVF9	FIFOOVF8	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FIFOOVF0 to FIFOOVF11	Scan Group n FIFO Overflow Flag	The suffix number of each bit symbol corresponds to the scan group number n. 0: No overflow 1: FIFO overflow is detected	R
b15 to b12	—	Reserved	These bits are read as 0.	R
b27 to b16	FIFOFLF0 to FIFOFLF11	Scan Group n FIFO Data Read Request Flag	The suffix number of each bit symbol corresponds to the scan group number n. 0: FIFO Data Read Request is not detected. 1: FIFO Data Read Request is detected.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

n = 0 to 11

The ADFIFOERSR register indicates the status of the FIFO. Each flag can be cleared in ADFIFOERSCR.

FIFOOVFn Bit (Scan Group n FIFO Overflow Flag) (n = 0 to 11)

The FIFOOVFn bit indicates whether the overflow occurred in the FIFO of the scan group n.
When FIFOOVFn = 0, an overflow is not detected in the FIFO. When FIFOOVFn = 1, an overflow occurred in the FIFO and the A/D conversion result cannot be stored to the FIFO. FIFOOVFn can be cleared in ADFIFOERSCR.

FIFOFLFn Bit (Scan Group n FIFO Data Read Request Flag) (n = 0 to 11)

The FIFOFLFn bit indicates whether a FIFO data read request in scan group n has been detected. The setting and clearing conditions of FIFOFLFn are shown as follows:
[Setting condition]

- When the condition of ADFIFOSRm.FIFOSTn[4:0] ≤ ADFIFOINTLRm.FIFOILVn[4:0] is detected.

[Clearing condition]

- When ADFIFOERSCR.FIFOFLCn is written to 1 under the condition of ADFIFOSRm.FIFOSTn[4:0] > ADFIFOINTLRm.FIFOILVn[4:0].
- When the condition of ADFIFOSRm.FIFOSTn[4:0] > ADFIFOINTLRm.FIFOILVn[4:0] is occurred by the read access to ADFIFODRn register with DMAC or DTC.

Note: m = 0 to 5, n = 0 to 11

60.2.121 FIFO Error Status Clear Register (ADFIFOERSCR)

Address(es): ADC_B.ADFIFOERSCR 8743 8CF8h, ADC_B_NS.ADFIFOERSCR 0743 8CF8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	FIFOFLC11	FIFOFLC10	FIFOFLC9	FIFOFLC8	FIFOFLC7	FIFOFLC6	FIFOFLC5	FIFOFLC4	FIFOFLC3	FIFOFLC2	FIFOFLC1	FIFOFLC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FIFOOVFC11	FIFOOVFC10	FIFOOVFC9	FIFOOVFC8	FIFOOVFC7	FIFOOVFC6	FIFOOVFC5	FIFOOVFC4	FIFOOVFC3	FIFOOVFC2	FIFOOVFC1	FIFOOVFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

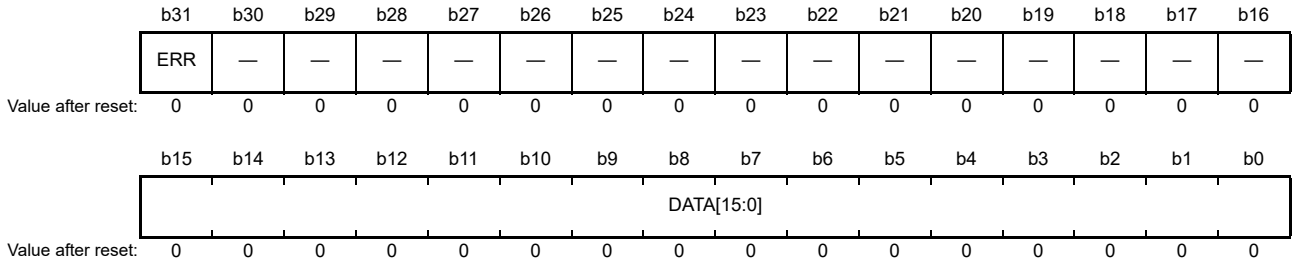
Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FIFOOVFC0 to FIFOOVFC11	Scan Group n FIFO Overflow Flag Clear	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOOVFn is cleared	W
b15 to b12	—	Reserved	The write value should be 0.	W
b27 to b16	FIFOFLC0 to FIFOFLC11	Scan Group n FIFO Data Read Request Flag Clear	The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOFLFn is cleared	W
b31 to b28	—	Reserved	The write value should be 0.	W

n = 0 to 11

ADFIFOERSCR clears the FIFO overflow flags and the FIFO data read request flags for scan group n.

60.2.122 A/D Data Register n (ADDRn) (n = 0 to 35)

Address(es): ADC_B.ADDR0 8743 A000h, ADC_B.ADDR1 8743 A004h, ADC_B.ADDR2 8743 A008h, ADC_B.ADDR3 8743 A00Ch, ADC_B.ADDR4 8743 A010h, ADC_B.ADDR5 8743 A014h, ADC_B.ADDR6 8743 A018h, ADC_B.ADDR7 8743 A01Ch, ADC_B.ADDR8 8743 A020h, ADC_B.ADDR9 8743 A024h, ADC_B.ADDR10 8743 A028h, ADC_B.ADDR11 8743 A02Ch, ADC_B.ADDR12 8743 A030h, ADC_B.ADDR13 8743 A034h, ADC_B.ADDR14 8743 A038h, ADC_B.ADDR15 8743 A03Ch, ADC_B.ADDR16 8743 A040h, ADC_B.ADDR17 8743 A044h, ADC_B.ADDR18 8743 A048h, ADC_B.ADDR19 8743 A04Ch, ADC_B.ADDR20 8743 A050h, ADC_B.ADDR21 8743 A054h, ADC_B.ADDR22 8743 A058h, ADC_B.ADDR23 8743 A05Ch, ADC_B.ADDR24 8743 A060h, ADC_B.ADDR25 8743 A064h, ADC_B.ADDR26 8743 A068h, ADC_B.ADDR27 8743 A06Ch, ADC_B.ADDR28 8743 A070h, ADC_B.ADDR29 8743 A074h, ADC_B.ADDR30 8743 A078h, ADC_B.ADDR31 8743 A07Ch, ADC_B.ADDR32 8743 A080h, ADC_B.ADDR33 8743 A084h, ADC_B.ADDR34 8743 A088h, ADC_B.ADDR35 8743 A08Ch, ADC_B_NS.ADDR0 0743 A000h, ADC_B_NS.ADDR1 0743 A004h, ADC_B_NS.ADDR2 0743 A008h, ADC_B_NS.ADDR3 0743 A00Ch, ADC_B_NS.ADDR4 0743 A010h, ADC_B_NS.ADDR5 0743 A014h, ADC_B_NS.ADDR6 0743 A018h, ADC_B_NS.ADDR7 0743 A01Ch, ADC_B_NS.ADDR8 0743 A020h, ADC_B_NS.ADDR9 0743 A024h, ADC_B_NS.ADDR10 0743 A028h, ADC_B_NS.ADDR11 0743 A02Ch, ADC_B_NS.ADDR12 0743 A030h, ADC_B_NS.ADDR13 0743 A034h, ADC_B_NS.ADDR14 0743 A038h, ADC_B_NS.ADDR15 0743 A03Ch, ADC_B_NS.ADDR16 0743 A040h, ADC_B_NS.ADDR17 0743 A044h, ADC_B_NS.ADDR18 0743 A048h, ADC_B_NS.ADDR19 0743 A04Ch, ADC_B_NS.ADDR20 0743 A050h, ADC_B_NS.ADDR21 0743 A054h, ADC_B_NS.ADDR22 0743 A058h, ADC_B_NS.ADDR23 0743 A05Ch, ADC_B_NS.ADDR24 0743 A060h, ADC_B_NS.ADDR25 0743 A064h, ADC_B_NS.ADDR26 0743 A068h, ADC_B_NS.ADDR27 0743 A06Ch, ADC_B_NS.ADDR28 0743 A070h, ADC_B_NS.ADDR29 0743 A074h, ADC_B_NS.ADDR30 0743 A078h, ADC_B_NS.ADDR31 0743 A07Ch, ADC_B_NS.ADDR32 0743 A080h, ADC_B_NS.ADDR33 0743 A084h, ADC_B_NS.ADDR34 0743 A088h, ADC_B_NS.ADDR35 0743 A08Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA[15:0]	A/D Conversion Data		R
b30 to b16	—	Reserved	These bits are read as 0.	R
b31	ERR	A/D Conversion Data Error Status	0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADDRn registers are read-only registers to read A/D conversion results.

DATA[15:0] Bits (A/D Conversion Data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the analog input channel n.

The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 49) to which the analog input channel n is assigned.

ERR Bit (A/D Conversion Data Error Status)

The ERR bit indicates the error status in the A/D conversion for the analog input channel n.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

60.2.123 A/D Extended Analog Data Register n (ADEXDRn) (n = 0 to 2, 4 to 6, 8, 9, 16 to 18)

Address(es): ADC_B.ADEXDR0 8743 A180h, ADC_B.ADEXDR1 8743 A184h, ADC_B.ADEXDR2 8743 A188h, ADC_B.ADEXDR4 8743 A190h, ADC_B.ADEXDR5 8743 A194h, ADC_B.ADEXDR6 8743 A198h, ADC_B.ADEXDR8 8743 A1A0h, ADC_B.ADEXDR9 8743 A1A4h, ADC_B.ADEXDR16 8743 A1C0h, ADC_B.ADEXDR17 8743 A1C4h, ADC_B.ADEXDR18 8743 A1C8h, ADC_B_NS.ADEXDR0 0743 A180h, ADC_B_NS.ADEXDR1 0743 A184h, ADC_B_NS.ADEXDR2 0743 A188h, ADC_B_NS.ADEXDR4 0743 A190h, ADC_B_NS.ADEXDR5 0743 A194h, ADC_B_NS.ADEXDR6 0743 A198h, ADC_B_NS.ADEXDR8 0743 A1A0h, ADC_B_NS.ADEXDR9 0743 A1A4h, ADC_B_NS.ADEXDR16 0743 A1C0h, ADC_B_NS.ADEXDR17 0743 A1C4h, ADC_B_NS.ADEXDR18 0743 A1C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR	—	—	—	—	DIAGSR[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA[15:0]	A/D Conversion Data		R
b23 to b16	—	Reserved	These bits are read as 0.	R
b26 to b24	DIAGSR[2:0]	Self-Diagnosis Status*1		R
b30 to b27	—	Reserved	These bits are read as 0.	R
b31	ERR	A/D Conversion Error Status	0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

Note 1. Only the ADEXDR0-2, ADEXDR16-18 register has the DIAGSR[2:0] bits. The DIAGSR[2:0] bits in the ADEXDRn register except for the ADEXDR0-2, ADEXDR16-18 register are reserved bits.

The ADEXDRn registers are read-only registers to read A/D conversion results of the extended analog function. The ADEXDRn registers and the extended analog functions are as follows:

- ADEXDR0: ADC0 self-diagnosis
- ADEXDR1: ADC1 self-diagnosis
- ADEXDR2: ADC2 self-diagnosis
- ADEXDR4: Temperature sensor
- ADEXDR5: Internal reference voltage
- ADEXDR6: VBATT 1/6 voltage
- ADEXDR8: D/A converter channel 0 output
- ADEXDR9: D/A converter channel 1 output
- ADEXDR16: Channel-dedicated sample-and-hold circuit unit 0 self-diagnosis channel
- ADEXDR17: Channel-dedicated sample-and-hold circuit unit 1 self-diagnosis channel
- ADEXDR18: Channel-dedicated sample-and-hold circuit unit 2 self-diagnosis channel

DATA[15:0] Bits (A/D Conversion Data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the extended analog function channel. The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 49) to which the extended analog function channel is assigned.

DIAGSR[2:0] Bits (Self-Diagnosis Status)

The DIAGSR[2:0] bits indicate the voltage setting of the self-diagnosis channel. The DIAGSR[2:0] bits in the

ADEXDR0, ADEXDR1, and ADEXDR2 registers indicate the setting of the ADSGDCRm.DIAGVAL[2:0] bits (m = 0 to 11). The DIAGSR[2:0] bits in the ADEXDR16, ADEXDR17, and ADEXDR18 registers indicate the setting of the ADSHSDCRm.SHDIAGn[2:0] bits (m = 0, 1; n = 0 to 11) of the scan group to which the channel-dedicated sample-and-hold circuit is assigned.
The DIAGSR[2:0] bits are available in the ADEXDR0, ADEXDR1, ADEXDR2, ADEXDR16, ADEXDR17 and ADEXDR18 registers. The DIAGSR[2:0] bits in ADEXDRn except above are reserved.

ERR Bit (A/D Conversion Error Status)

The ERR bit indicates the error status in the A/D conversion for the extended analog functions.
When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

60.2.124 FIFO Data Register n (ADFIFODRn) (n = 0 to 11)

Address(es): ADC_B.ADFIFODR0 8743 A200h, ADC_B.ADFIFODR1 8743 A204h, ADC_B.ADFIFODR2 8743 A208h, ADC_B.ADFIFODR3 8743 A20Ch, ADC_B.ADFIFODR4 8743 A210h, ADC_B.ADFIFODR5 8743 A214h, ADC_B.ADFIFODR6 8743 A218h, ADC_B.ADFIFODR7 8743 A21Ch, ADC_B.ADFIFODR8 8743 A220h, ADC_B.ADFIFODR9 8743 A224h, ADC_B.ADFIFODR10 8743 A228h, ADC_B.ADFIFODR11 8743 A22Ch, ADC_B_NS.ADFIFODR0 0743 A200h, ADC_B_NS.ADFIFODR1 0743 A204h, ADC_B_NS.ADFIFODR2 0743 A208h, ADC_B_NS.ADFIFODR3 0743 A20Ch, ADC_B_NS.ADFIFODR4 0743 A210h, ADC_B_NS.ADFIFODR5 0743 A214h, ADC_B_NS.ADFIFODR6 0743 A218h, ADC_B_NS.ADFIFODR7 0743 A21Ch, ADC_B_NS.ADFIFODR8 0743 A220h, ADC_B_NS.ADFIFODR9 0743 A224h, ADC_B_NS.ADFIFODR10 0743 A228h, ADC_B_NS.ADFIFODR11 0743 A22Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR	CH[6:0]							—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA[15:0]	A/D Conversion Data		R
b23 to b16	—	Reserved	These bits are read as 0.	R
b30 to b24	CH[6:0]	A/D Conversion Channel Number		R
b31	ERR	A/D Conversion Data Error Status	0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADFIFODRn register is used to read data stored in the FIFO of scan group n.

DATA[15:0] Bits (A/D Conversion Data)

The DATA[15:0] bits are read-only bits that read the A/D conversion data stored in the scan group n FIFO.

CH[6:0] Bits (A/D Conversion Channel Number)

The CH[6:0] bits indicate the channel number of the A/D conversion data that is read from the DATA[15:0] bits.

ERR Bit (A/D Conversion Data Error Status)

The ERR bit indicates the error status in the A/D conversion data of the analog channel indicated by the CH[6:0] bits. When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

60.3 Operation

60.3.1 A/D Conversion Clock

A/D conversion clock (ADCLK) is the operation clock of ADC. The A/D converters (ADC0, ADC1, and ADC2) are operated and controlled by ADCLK as the basic clock. Figure 60.5 shows clock structure of ADC.
ADCLK is generated from the clock source and the division ratio selected in ADCLKCR register. The frequency of ADCLK should be set so that PCLKA ≥ ADCLK. Also, the frequency of ADCLK should be set so that it is within the guaranteed operating range specified in the electrical characteristics. For details, see section 67, Electrical Characteristics.

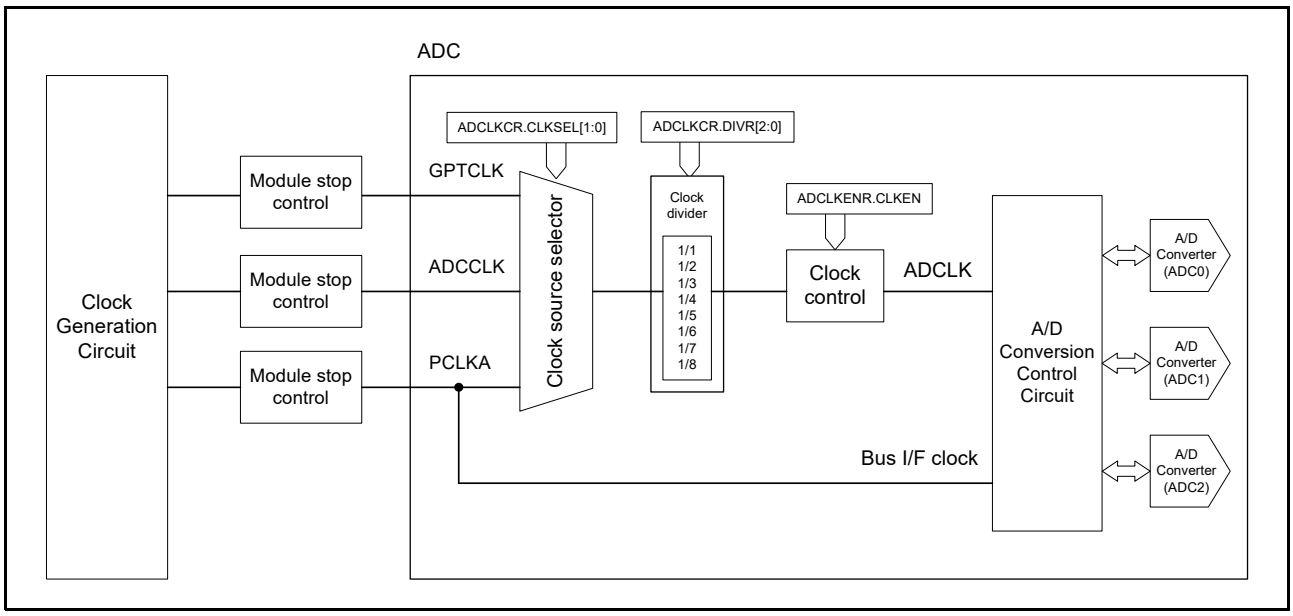


Figure 60.5 Clock Structure

60.3.2 A/D Converter Operation Mode

60.3.2.1 SAR Mode

In SAR mode, A/D converter operates as a nyquist type A/D converter. Analog channels are sampled one time and converted analog to digital by successive approximation register (SAR) method. The features of SAR mode are shown in the following:
[Features of SAR mode]

- Fast conversion: A/D conversion data can be acquired with the shortest group delay.
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Single-ended input and differential input supported (digital filter functions not supported)

For details about operation of A/D conversion in SAR mode, see section 60.3.7, Scan Mode. For restrictions on SAR mode, see section 60.10.14, Restrictions on SAR Mode.

60.3.2.2 Oversampling Mode

In oversampling mode, A/D converter operates as oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by noise shaping successive approximation register method (NS-SAR). The features of oversampling mode are shown in the following:
[Features of oversampling mode]

- High accuracy conversion: A/D conversion data can be acquired with noise shaping technology and digital filter function.*1
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital filter function must be used in oversampling mode. It is prohibited to operate without digital filter function in oversampling mode.

For details about operation of A/D conversion in oversampling mode, see section 60.3.7, Scan Mode. For restrictions on oversampling mode, see section 60.10.15, Restrictions on Oversampling Mode.

60.3.2.3 Hybrid Mode

Hybrid mode has both features of SAR mode and oversampling mode. In hybrid mode, A/D converter operates as oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by noise shaping successive approximation register (NS-SAR) method. In the hybrid mode, while switching channels for each sampling, multiple analog channels are oversampled. The features of hybrid mode are shown in the following:

[Features of hybrid mode]

- High accuracy conversion: A/D conversion data can be acquired with noise shaping technology and digital filter function.*1
- Fast conversion: A/D conversion data can be acquired with the shortest group delay when in the continuous scan mode and after initial delay time.
- A/D conversion of up to 4 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital filter function must be used in hybrid mode. It is prohibited to operate without digital filter function in hybrid mode.

For details about operation of A/D conversion in hybrid mode, see section 60.3.7, Scan Mode. For restrictions on hybrid mode, see section 60.10.16, Restrictions on Hybrid Mode.

60.3.3 Single-Ended Input and Differential Input

ADC supports single-ended input and differential input. The selection of single-ended input or differential input is specified in ADCHCRn.AINMD (n = 0 to 49) bit. For the analog channels that support differential input, see Table 60.4 and Table 60.5.

Single-ended Input

In single-ended input mode, the difference between the voltage of the analog channel (signal source) and the analog reference ground voltage (VREFL0) is A/D-converted.

Differential Input

In differential input mode, the even-numbered analog channel is used as the non-inverting input (+) (AINP) and the odd-numbered analog channel is used as the inverting input (−) (AINN), then the difference voltage between the non-inverting input (+) and the inverting input (−) (AINP − AINN) is A/D-converted.*1

For A/D conversion with differential input mode, select the non-inverting input (+) channel*2 of the target of differential pair in ADCHCRn.CNVCS[6:0] (n = 0 to 49) bits, and select the differential input mode in ADCHCRn.AINMD (n = 0 to 49) bit.

It is prohibited to perform A/D conversion for channels that do not support differential input with differential input mode. In that case, A/D conversion result is not guaranteed.

Note 1. The differential input pair is a combination of channels with analog channel number 2i and 2i + 1 (i = 0, 1, 2, 3, ...), and they are named ANxxxP and ANxxxN (xxx = 000, 002, 004, ...). The combination of discontinuous analog channel numbers or the combination of analog channel numbers 2i − 1 and 2i cannot be used as a

differential input pair.

Note 2. It is the even-numbered analog channel which is named as ANxxxP (xxx = 000, 002, 004, …).

60.3.4 Analog Channel

The analog channel is the source of the analog signal that is targeted for A/D conversion. The analog channels consist of the following elements:

- Analog input channels: A/D conversion channels for the analog input from MCU's I/O pin as the signal source.
- Extended analog channels: A/D conversion channels for the source of the analog signal inside the MCU.

To perform A/D conversion for the analog channels, assignments to the virtual channel and scan group are required. For each element, refer to each item.

60.3.5 Virtual Channel

The virtual channel is a group of registers that stores the A/D conversion configurations for an analog channel. The virtual channels can be specified the configurations for A/D conversion such as the selection of the analog channels, the optional settings for A/D conversion, the data processing method of the A/D conversion data, the assignment to scan groups, and so on.

To perform the A/D conversion of the analog channel, the analog channel should be assigned to any of the virtual channels. Then that virtual channel should be assigned to one of the scan groups. See Figure 60.6 for the relationship between the analog channels and the virtual channels and scan groups. Figure 60.7 shows the A/D conversion order in Figure 60.6 settings.

The virtual channel can be assigned to only one scan group. If the same analog channel is to be converted within the different scan group, or the same analog channel is to be converted several times within the same scan group, assign the analog channel to several virtual channels. When performing the A/D conversion more than once using more than one virtual channel within the same scan group for the same analog channel, the FIFO function should be used in combination.

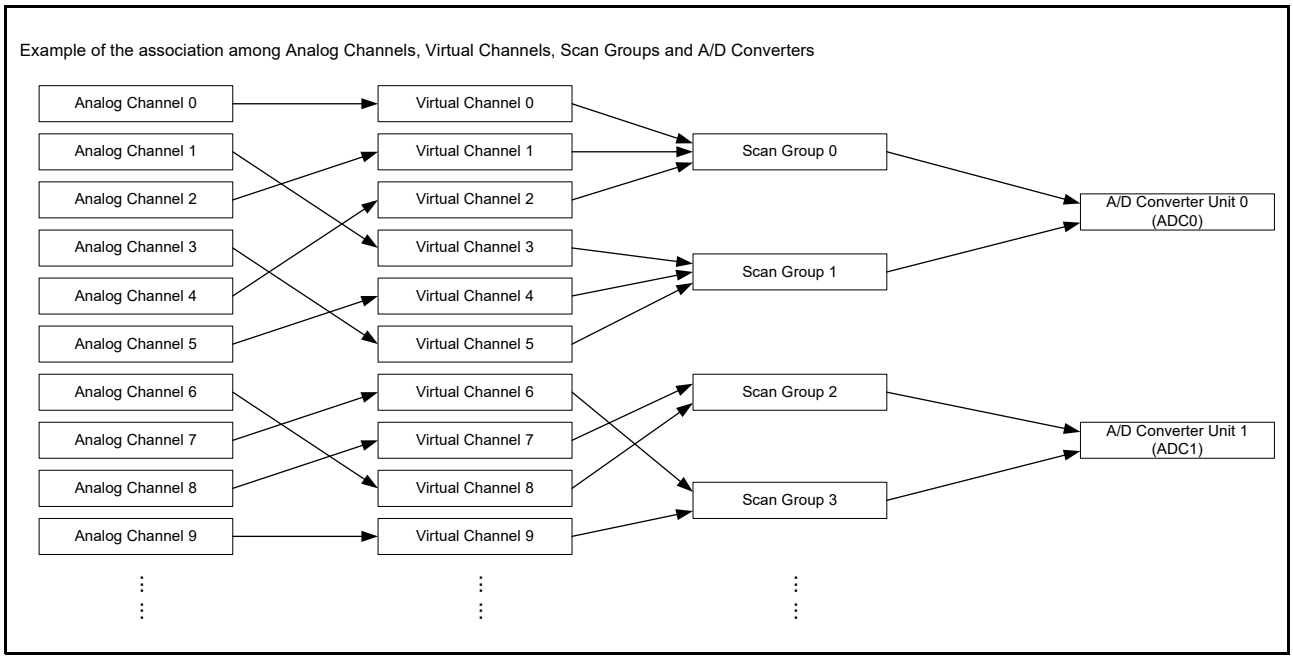


Figure 60.6 The Concept of Virtual Channel

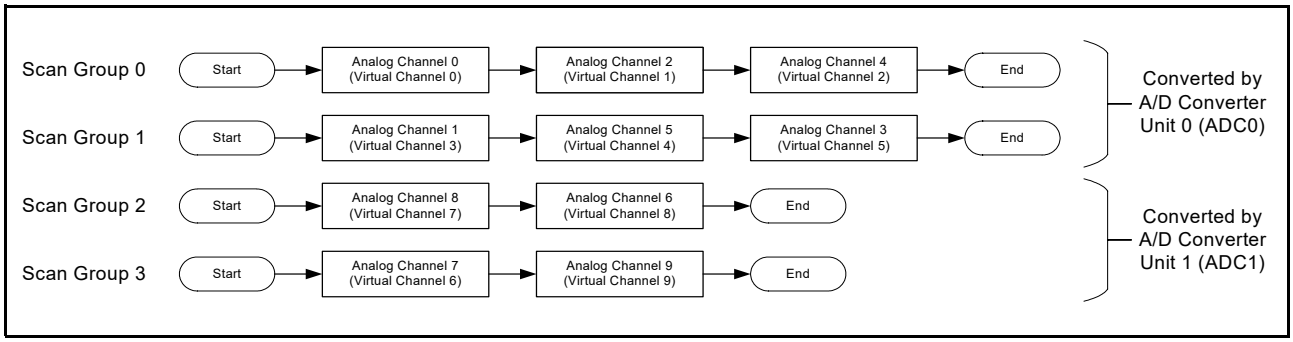


Figure 60.7 Example of A/D Conversion Order at the Scanning Operation

60.3.6 Scan Group

The scan group is a group of the analog channels that perform A/D conversion in a scanning operation. To perform the A/D conversion using the scan group, configure the following steps:

- Assign the analog channel for the A/D conversion to the virtual channel.
- Assign the virtual channel to the scan group.
- Assign the scan group to A/D converter.

See Figure 60.6 and Figure 60.7 for the relationship among the analog channels, the virtual channels, and the scan groups. A single scan group can be assigned virtual channels in the following ranges, according to its operation mode.

- SAR mode: up to 8 virtual channels per scan group
- Oversampling mode: up to 8 virtual channels per scan group
- Hybrid mode: from 2 to 4 virtual channels per scan group

If more than the maximum number of virtual channels are allocated to one scan group, A/D conversion is performed on the 8th (SAR mode or oversampling mode) or 4th (Hybrid mode) channels in ascending order of virtual channel numbers. Subsequent channels are not subject to A/D conversion.

Notes on the scan groups

The virtual channels assigned to one scan group should be assigned the analog channels that can be converted by the same A/D converter. If any analog channels cannot be converted by the specified A/D converter are included, the A/D conversion result of those channels are undefined.

In hybrid mode, you must assign at least 2 virtual channels to a scan group. Operation is not guaranteed when any virtual channel is not assigned or only 1 virtual channel is assigned.

60.3.7 Scan Mode

The scanning operation is the operation of sequential A/D conversion for the analog channels. The modes of the scanning operation are the followings:

- Single scan mode
- Continuous scan mode
- Background continuous scan mode
- Same channel continuous scan mode
- Fixed channels continuous scan mode

The scan modes supported by each operation mode are shown in Table 60.6.

Table 60.6 Scan Modes Supported by Each Operation Mode

	Operation Mode of A/D Converter		
Scan Mode	**SAR Mode**	**Oversampling Mode**	**Hybrid Mode**
Single scan mode	✓	✓	✓
Continuous scan mode	✓	✓	✓
Background continuous scan mode	—	—	✓
Same channel continuous scan mode	—	✓	—
Fixed channels continuous scan mode	—	—	✓ Only for fixed channels*1

✓: available, —: not available

Note 1. A/D conversion on non-fixed channels proceeds in SAR mode.

The scanning operations are performed on a scan group basis. When the scanning operation is started, A/D conversion is performed for each analog channel based on the virtual channel setting. If the scanning operation starts for 2 or more scan group assigned to the same A/D converter at the same time, the scanning operation of the group with the lowest scan group number is performed.

The A/D conversion order of the analog channel in the scanning operation is the ascending order of the virtual channel numbers assigned to the scan group. (The channels with the lowest virtual channel numbers are performed the A/D conversion at the forward end of the scan group. The channels with larger virtual channel numbers are performed the A/D conversion at the backward of the scan group.)

See Figure 60.6 and Figure 60.7 for the relationship of the conversion order of the respective analog channel in the scanning operation.

The scanning operations in each operation mode are shown in the following.

60.3.7.1 Single Scan Mode

- In single scan mode, one scan group is scanned once for each A/D conversion start trigger input.
- Each time an A/D conversion start trigger is input, A/D conversion is performed once for each analog channel assigned to that scan group.

60.3.7.2 Continuous Scan Mode

- The continuous scan mode repeats the scanning operation of one scan group.
- When an A/D conversion start trigger is input, A/D conversion of each analog channel assigned to that scan group is repeated until A/D conversion stop processing is performed.

60.3.7.3 Background Continuous Scan Mode

- The background continuous scan mode repeats the scanning operation of a single scanning group.
- Scanning operation starts when the first A/D conversion start trigger is input.
- A/D conversions of each analog channel assigned to that scan group are repeated until A/D conversion stop processing is performed. (Continuous scanning operation)
- When an A/D conversion start trigger is input during continuous scanning operation, the A/D conversion data at that point is output. (Unless an A/D conversion start trigger is input, the A/D conversion data are not output and the values of the A/D data registers and FIFO data registers are not updated.)

60.3.7.4 Same Channel Continuous Scan Mode

- In same channel continuous scan mode, A/D conversion on a single analog channel assigned to a desired scan group repeatedly proceeds. Changing the target channel for conversion in response to an A/D conversion start trigger stops the current A/D conversion and restarts A/D conversion for the new channel following the change.
- Scanning operation on the first analog channel assigned to a scan group starts in response to input of the first A/D conversion start trigger.
- The A/D conversion repeatedly proceeds until input of the next A/D conversion start trigger (continuous scanning operation).
- A/D conversion on the next analog channel assigned to the scan group starts in response to input of that next A/D conversion start trigger.
- The A/D conversion on the channel repeatedly proceeds until the input of a further A/D conversion start trigger. After that, the above operations are repeated in response to the input of A/D conversion start triggers.
- When an A/D conversion start trigger is input while A/D conversion on the last analog channel assigned to the scan group is repeated, A/D conversion is terminated and then restarted on the first channel in the group.
- In this mode, data outputs can be skipped so that only one result of conversion is output for every four or eight A/D conversions. See section 60.4.11, A/D Conversion Data Output Decimation for details.

60.3.7.5 Fixed Channels Continuous Scan Mode

- In fixed channels continuous scan mode, A/D conversion in hybrid mode proceeds on the up to three analog channels specified as the fixed channels and A/D conversion in SAR mode proceeds on the analog channels specified as non-fixed channels.
- A/D conversion proceeds on all of the fixed channels and then on one non-fixed channel in one round of scanning operation.
- A/D conversion is repeated on all of the fixed channels and one non-fixed channel as stated above while the target non-fixed channel is changed on each round until a next A/D conversion start trigger is input (continuous scanning operation).
- The output of converted data does not start until the number of times A/D conversion has been completed matches the number of taps in the digital filters for the fixed channels.
- The scan groups and virtual channels to which the individual fixed channels are assigned are specified per A/D converter unit. Only the settings listed in Table 60.7 are specifiable.
- For the constraints on the settings for this mode, see Table 60.8.

Table 60.7 Virtual Channel Numbers Which Should be Assigned as the Individual Fixed Channels

Unit	ADC0	ADC1	ADC2
Scan Group	Scan Group 0	Scan Group 1	Scan Group 2
First fixed channel	Virtual Channel 0	Virtual Channel 4	Virtual Channel 8
Second fixed channel	Virtual Channel 1	Virtual Channel 5	Virtual Channel 9
Third fixed channel	Virtual Channel 2	Virtual Channel 6	Virtual Channel 10

Table 60.8 Constraints on the Settings for the Fixed Channels Continuous Scan Mode

Description	Conditions
Number of fixed channels	ADSWNRm.SWFIXm = 1, 2, 3 (m = 0 to 2)
Number of non-fixed channels	ADSWNRm.SWNOFIXm = 1 (m = 0 to 2)
Number of conversions	ADSWNRm.SWNUMm ≥ number of fixed channels + number of non-fixed channels (m = 0 to 2) (The surplus of channels over the number of fixed channels + number of non-fixed channels corresponds to the number of idle channels.)
Scan groups of virtual channels to which the fixed channels are assigned	ADCHCRn.SGSEL[4:0] = 00h (scan group not selected) (n is replaced with the number of a destination scan group for assignment listed in Table 60.7.)
Filter settings for fixed channels	ADDOPCRAp.DFSEL[2:0]: Select filters with the same number of taps for the fixed channels in the same scan group. (p is replaced with the number of a destination virtual channel for assignment listed in Table 60.7.)
Number of sampling states for virtual channels to which individual non-fixed channels are assigned	ADSSTRx.SSTy[9:0] Set the same value for non-fixed channels.
Calculation of A/D converted data on fixed and non-fixed channels	Setting the addition or averaging function is prohibited.
Self-calibration	Self-calibration is required in both fixed channels continuous scan mode and SAR mode with single scan mode.

60.3.8 Example of the Scanning Operation

The scanning operations in each operation mode are shown in the following.

60.3.8.1 SAR Mode – Single Scan Mode

Table 60.9 and Figure 60.8 show the operation example in SAR mode – Single scan mode.

Table 60.9 Example of the Scanning Operation in SAR Mode – Single Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 11)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 11) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.

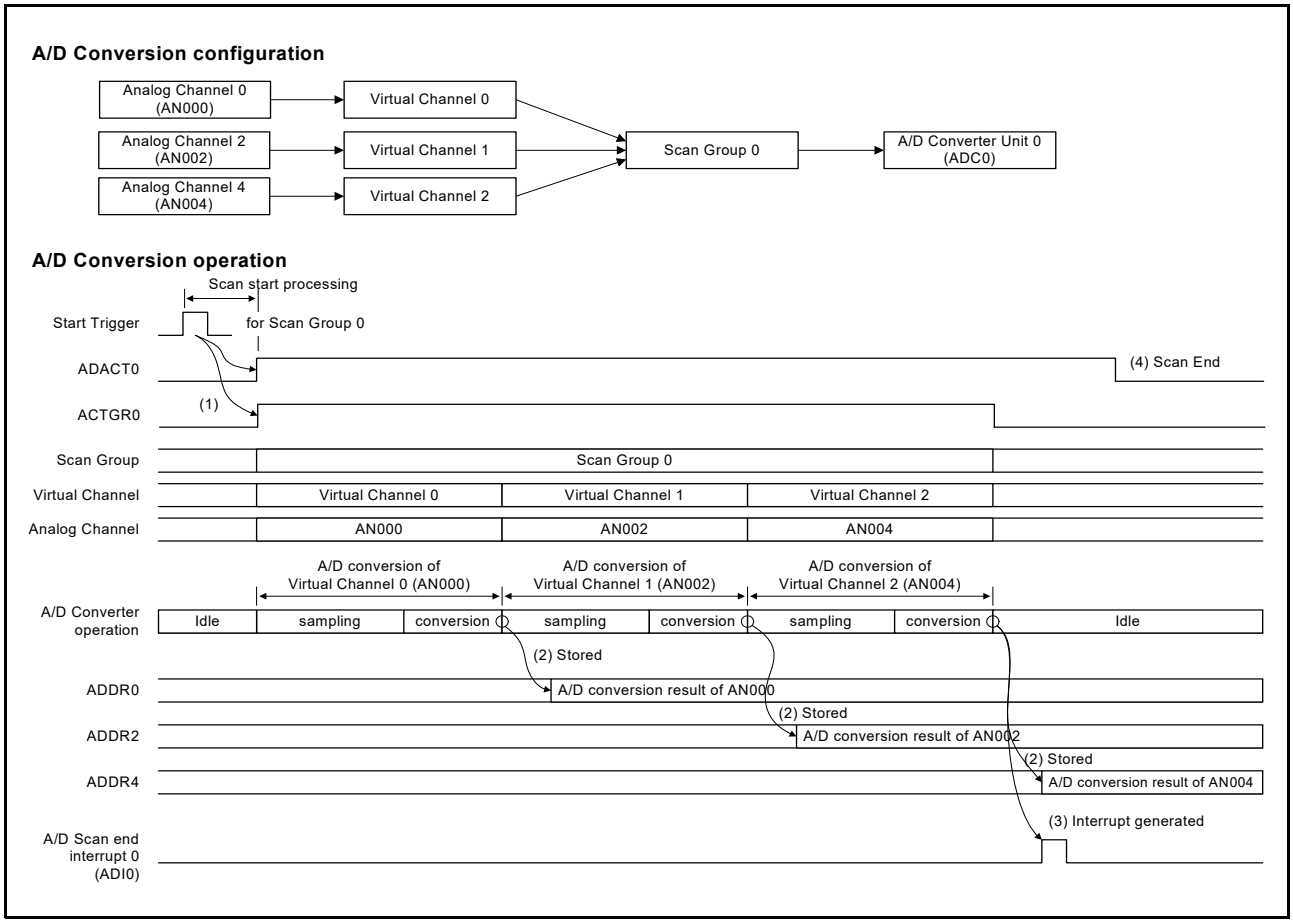


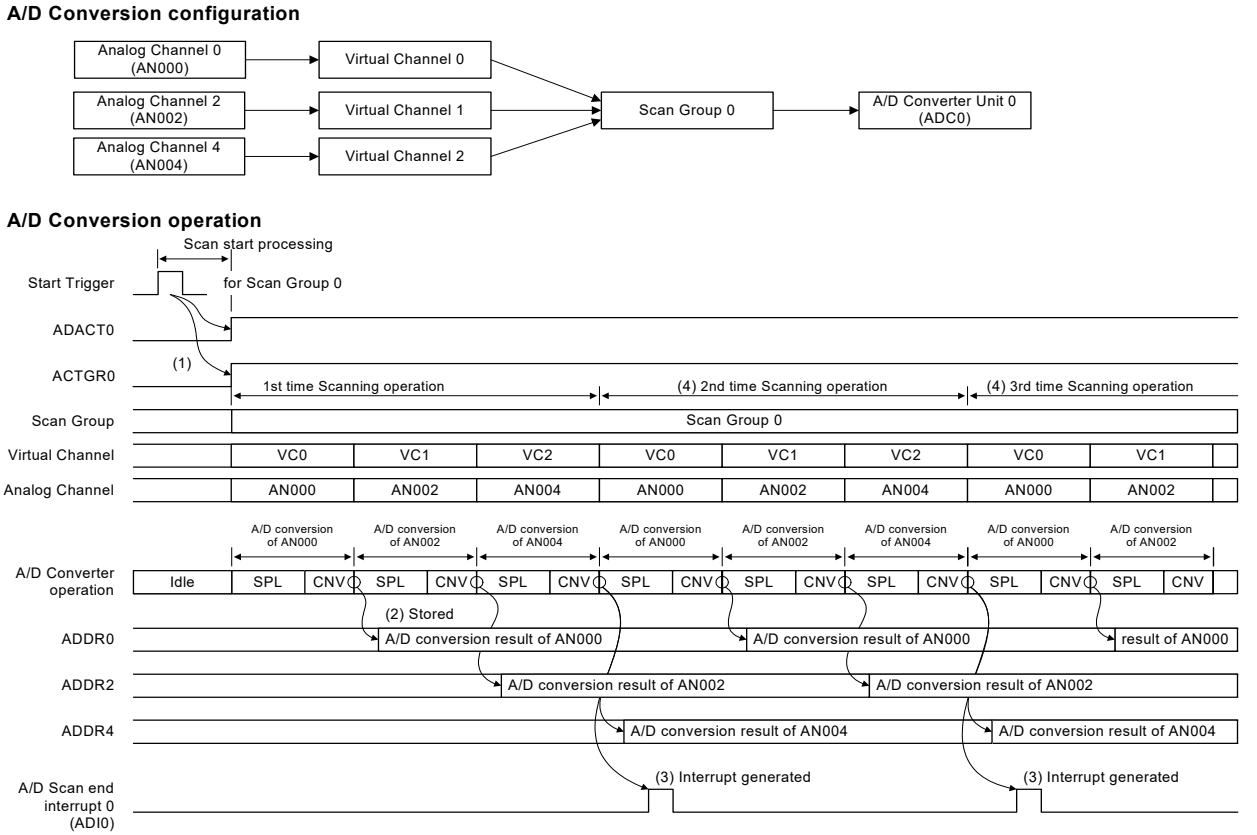
Figure 60.8 Example of the Scanning Operation in SAR Mode – Single Scan Mode

60.3.8.2 SAR Mode – Continuous Scan Mode

Table 60.10 and Figure 60.9 show the operation example in SAR mode – Continuous scan mode.

Table 60.10 Example of the Scanning Operation in SAR Mode – Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 11) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 11)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	Until the A/D conversion stop process is performed, Step 2 to 3 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 60.5.4, Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

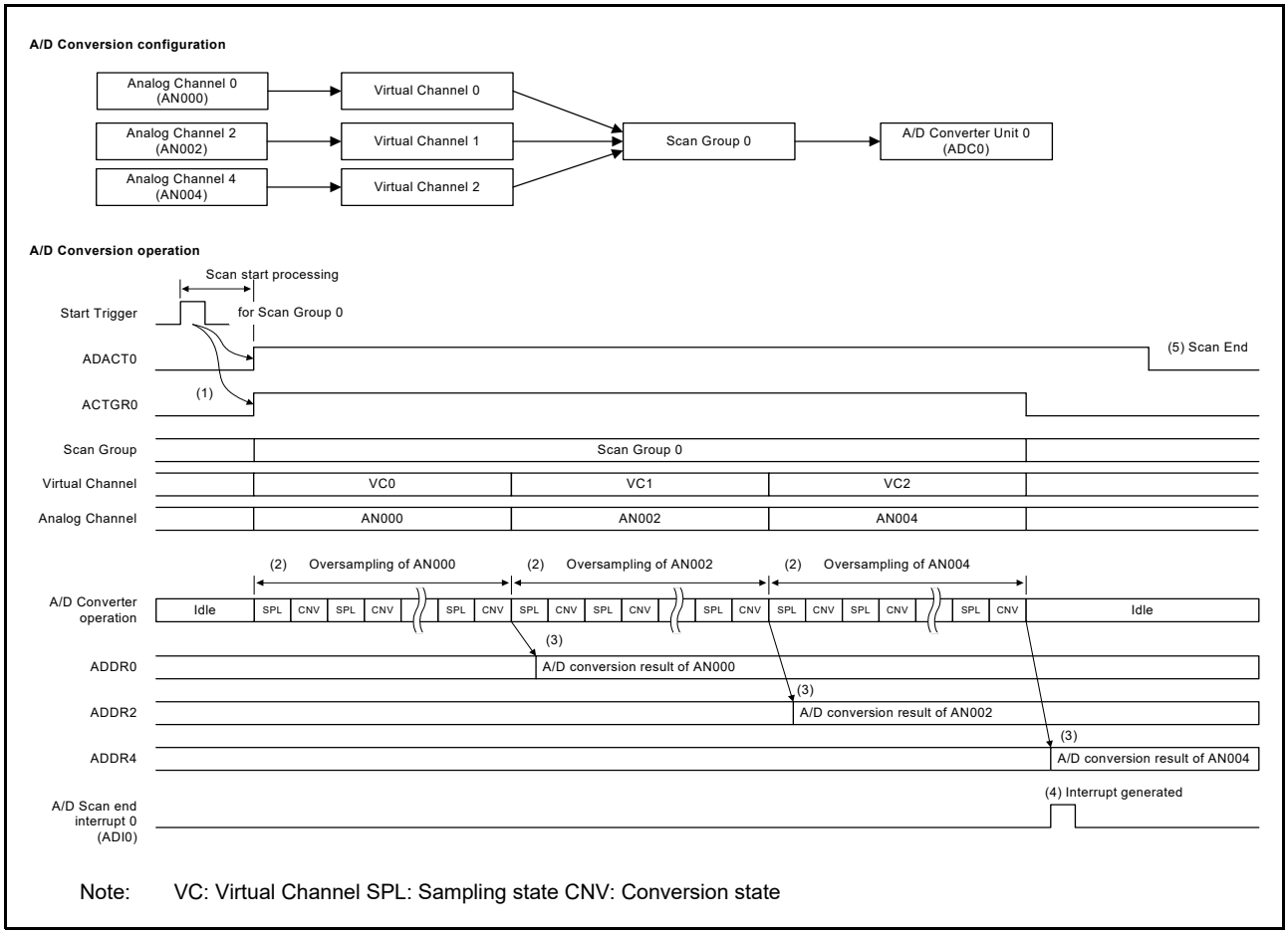
Figure 60.9 Example of the Scanning Operation in SAR Mode – Continuous Scan Mode

60.3.8.3 Oversampling Mode – Single Scan Mode

Table 60.11 and Figure 60.10 show the operation example in oversampling mode – single scan mode.

Table 60.11 Example of the Scanning Operation in Oversampling Mode – Single Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 11) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2)) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

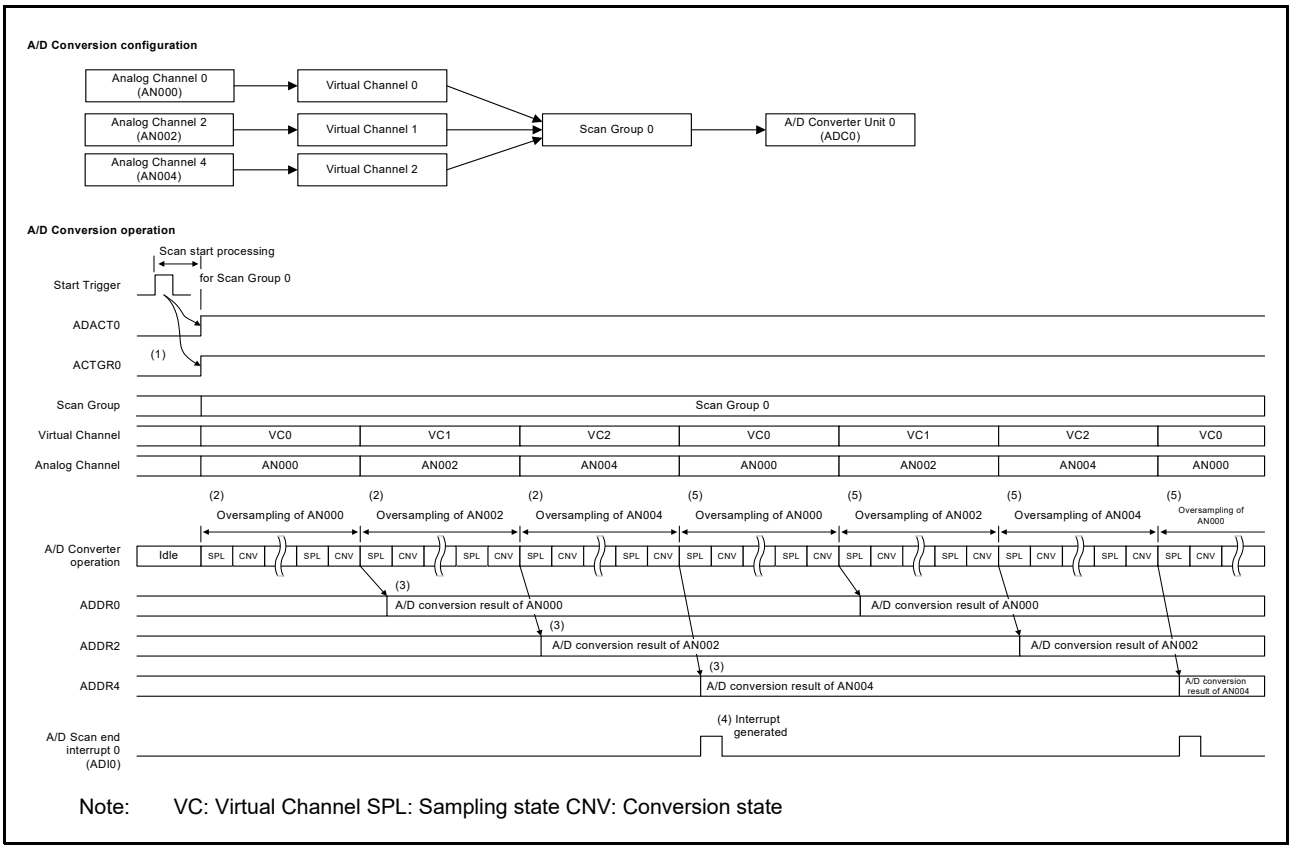
Figure 60.10 Example of the Scanning Operation in Oversampling Mode – Single Scan Mode

60.3.8.4 Oversampling Mode – Continuous Scan Mode

Table 60.12 and Figure 60.11 show the operation example in oversampling mode – continuous scan mode.

Table 60.12 Example of the Scanning Operation in Oversampling Mode – Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 11) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2)) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	Until the A/D conversion stop process is performed, Step 2 to 4 are repeated, and the scanning operation continues. To stop A/D conversion, you must follow to section 60.5.4, Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

Figure 60.11 Example of the Scanning Operation in Oversampling Mode – Continuous Scan Mode

60.3.8.5 Oversampling Mode – Same Channel Continuous Scan Mode

Table 60.13 and Figure 60.12 show the operation example in oversampling mode – same channel continuous scan mode.

Table 60.13 Example of the Scanning Operation in Oversampling Mode – Same Channel Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 11) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2)) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Every time A/D conversion on a single channel is completed, the result of A/D conversion is stored in an A/D data register i (ADDRi (i = 0 to 35)) or A/D extended analog data register j (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to18)). When the FIFO function is in use, the result of A/D conversion is also stored in a FIFO data register k (ADFIFODRk (k = 0 to 11)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to the given scan group is generated in response to completion of A/D conversion on a single channel.
4	The scanning operation continues until input of the next trigger. To stop the A/D conversion, follow the procedure shown in section 60.5.4, Force Stops the A/D Conversion Operation.

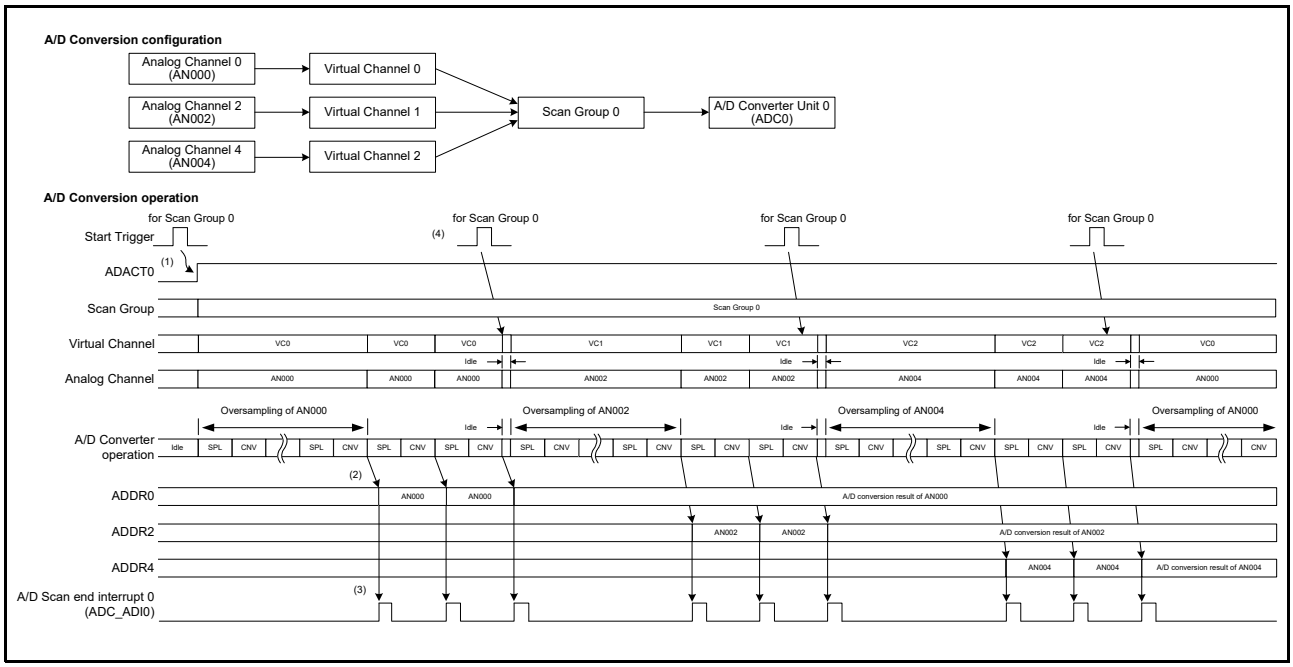


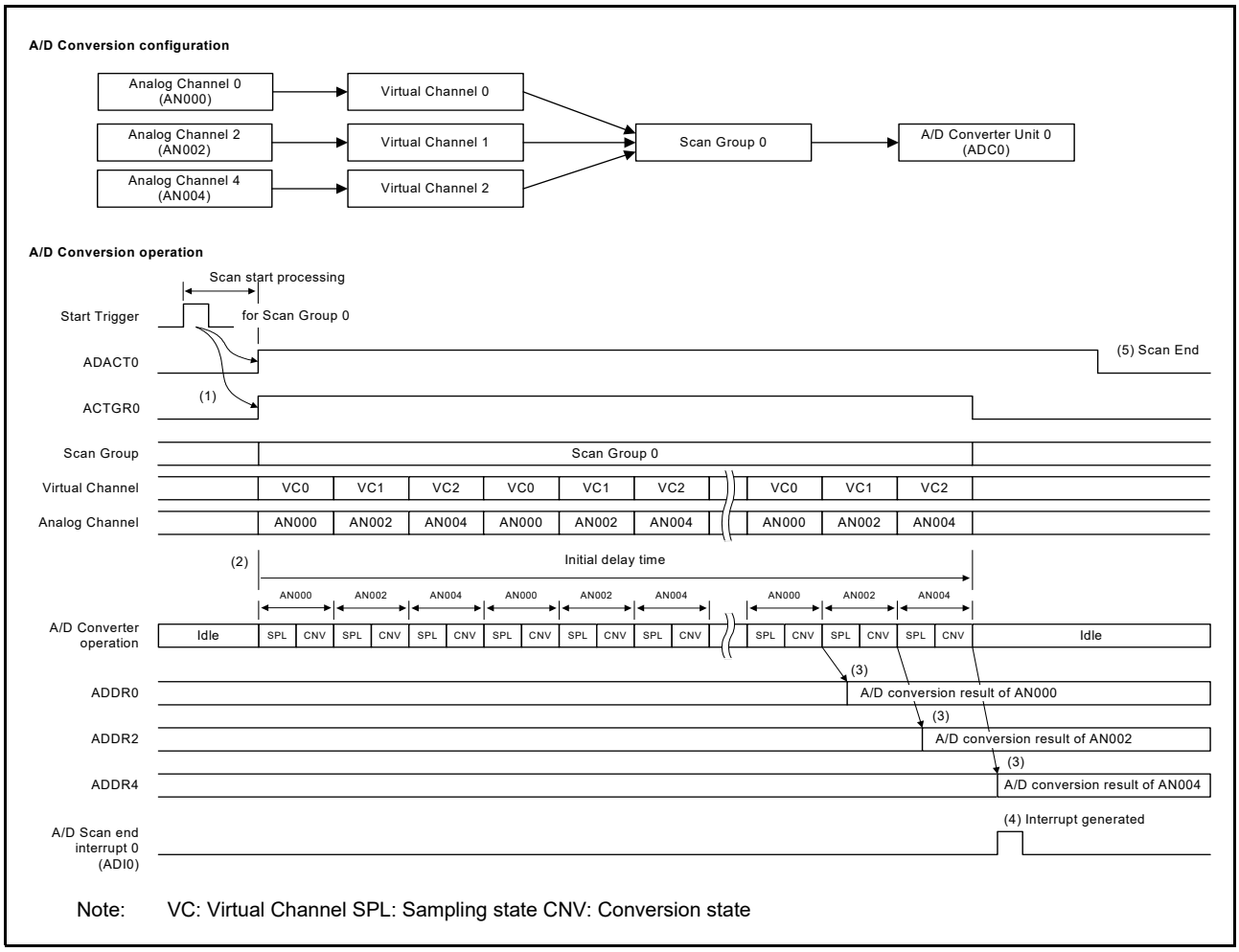
Figure 60.12 Example of the Scanning Operation in Oversampling Mode – Same Channel Continuous Scan Mode

60.3.8.6 Hybrid Mode – Single Scan Mode

Table 60.14 and Figure 60.13 show the operation example in hybrid mode – single scan mode.

Table 60.14 Example of the Scanning Operation in Hybrid Mode – Single Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 11) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state. And the oversampled data in the digital filter are also discarded at the end of the scanning operation.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

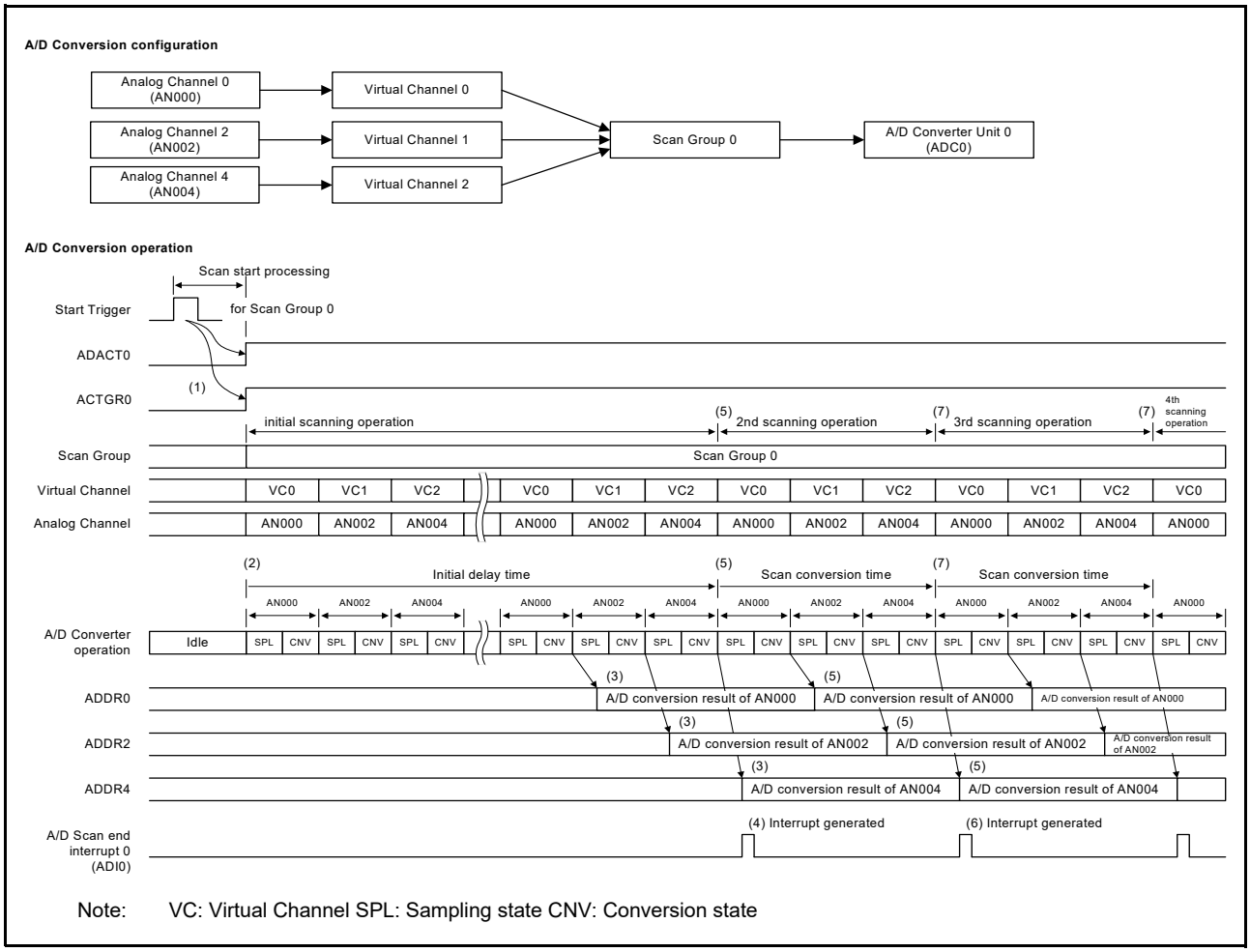
Figure 60.13 Example of the Scanning Operation in Hybrid mode – Single Scan Mode

60.3.8.7 Hybrid Mode – Continuous Scan Mode

Table 60.15 and Figure 60.14 show the operation example in hybrid mode – continuous scan mode.

Table 60.15 Example of the Scanning Operation in Hybrid Mode – Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 11) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	In scanning operation after the initial delay time has elapsed, each time oversampling is performed for an analog channel, the data in the digital filter is updated. The updated A/D conversion data for an analog channel can be output for each one time oversampling or for each multiple times oversampling corresponding to the number of A/D-converted value addition/ averaging times that is performed. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
6	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
7	Thereafter, until the A/D conversion stop process is performed, Step 5 to 6 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 60.5.4, Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

Figure 60.14 Example of the Scanning Operation in Hybrid Mode – Continuous Scan Mode

60.3.8.8 Hybrid Mode – Background Continuous Scan Mode

Table 60.16 and Figure 60.15 show the operation example in hybrid mode – background continuous scan mode.

Table 60.16 Example of the Scanning Operation in Hybrid Mode – Background Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 11) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0 to 2) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data becomes in ready to be output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time. In the background continuous scan operation, A/D conversion data can be obtained after the initial delay time has elapsed from the start of the scanning operation.)
4	In scanning operation after the initial delay time has elapsed, each time oversampling is performed for an analog channel, the data in the digital filter is updated. The updated A/D conversion data for an analog channel can be output for each one time oversampling or for each multiple times oversampling corresponding to the number of A/D-converted value addition/ averaging times that is performed.
5	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 11)).
6	If scan end interrupt is set to enable, scan end interrupt is generated.
7	Thereafter, until the A/D conversion stop process is performed, Background continuous scanning operation (Step 4) is repeated. And whenever the A/D conversion start trigger is input during Background continuous scanning operation, the A/D conversion data is output (Step 5 and Step 6). To stop A/D conversion, you must follow to section 60.5.4, Force Stops the A/D Conversion Operation.

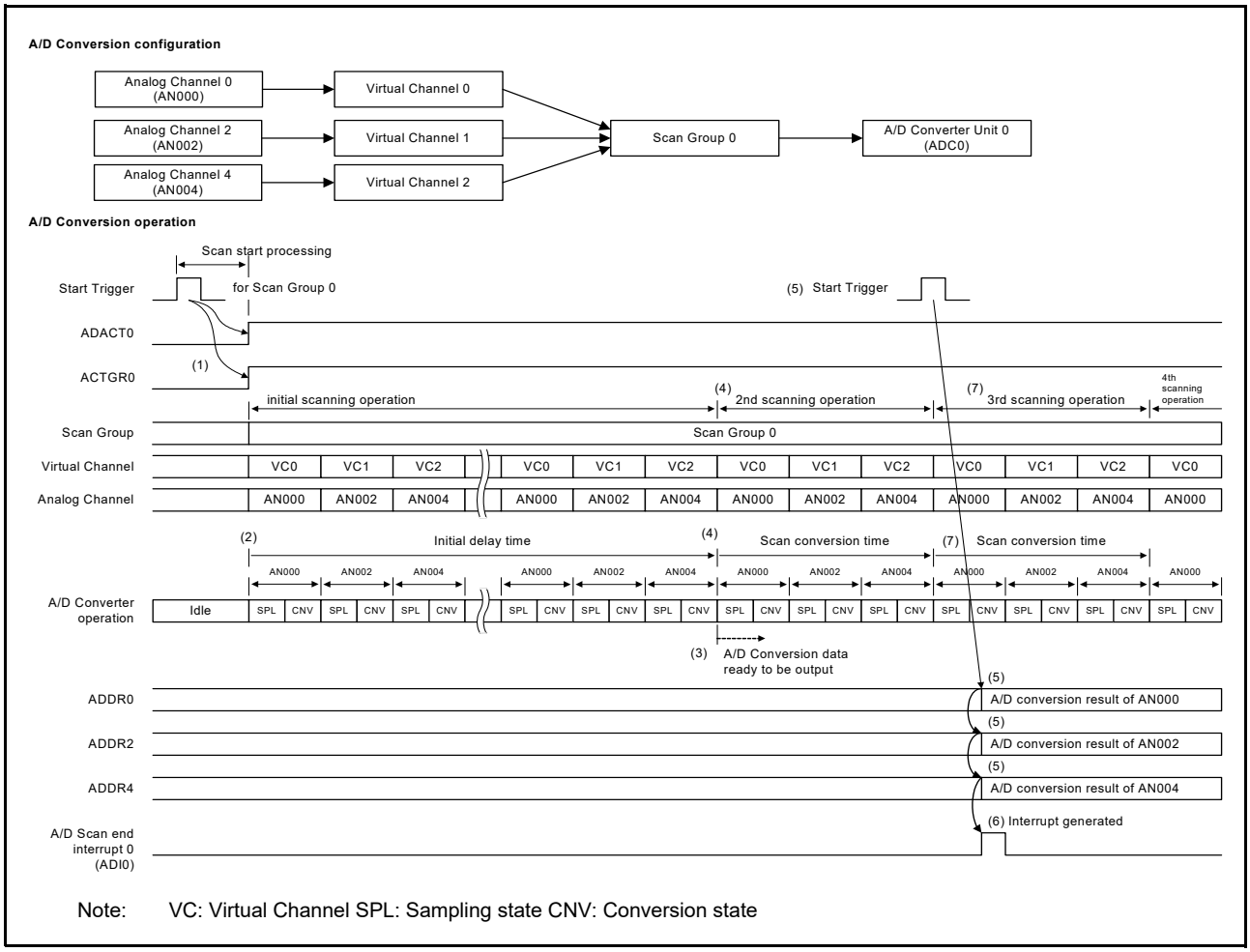


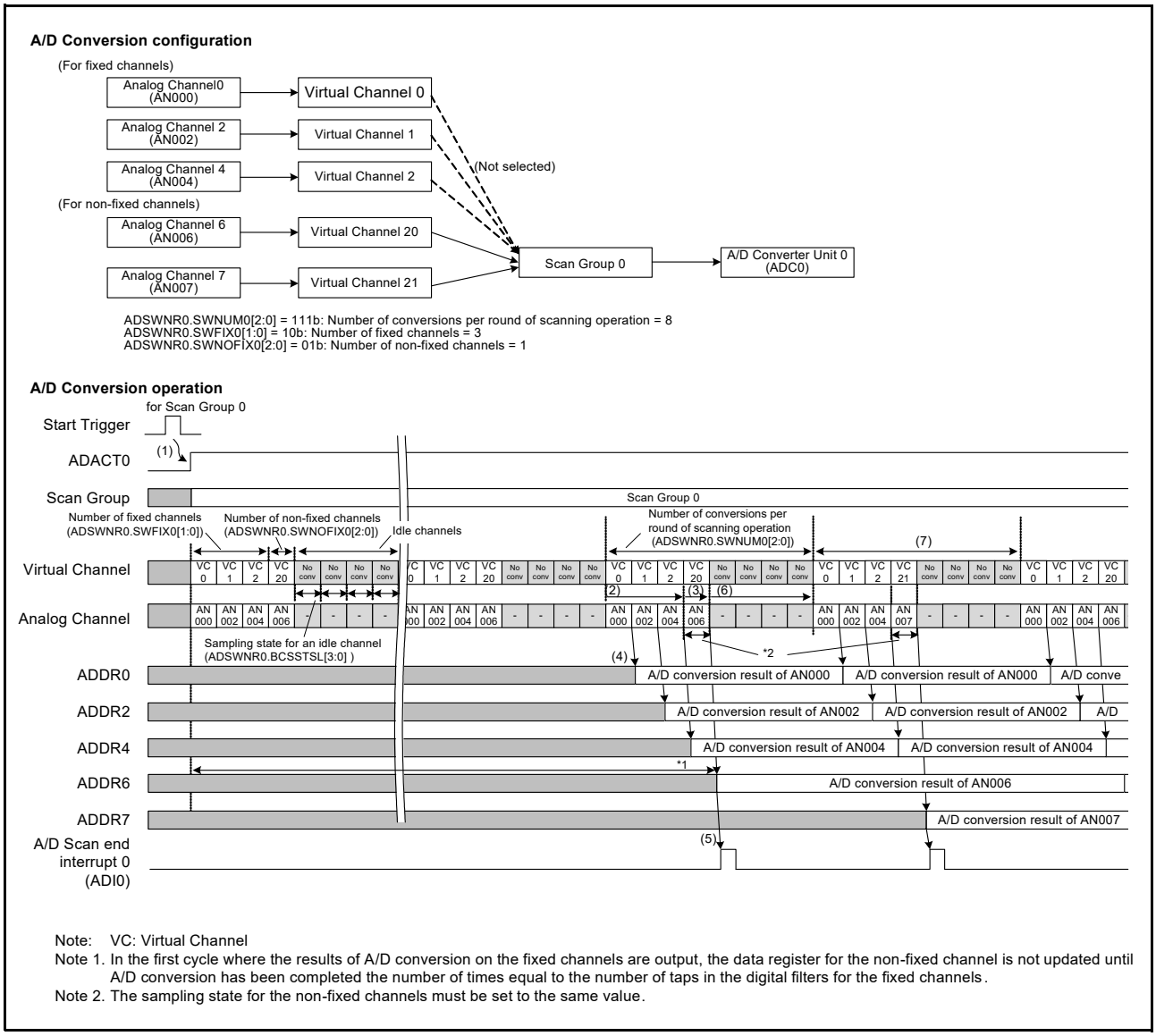
Figure 60.15 Example of the Scanning Operation in Hybrid Mode – Background Continuous Scan Mode

60.3.8.9 Hybrid Mode – Fixed Channels Continuous Scan Mode

Table 60.17 and Figure 60.16 show the operation example in hybrid mode – fixed channels continuous scan mode.

Table 60.17 Example of the Scanning Operation in Hybrid Mode – Fixed Channels Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	A/D conversion in hybrid mode proceeds on one to three fixed channels.
3	A/D conversion in SAR mode proceeds on one non-fixed channel. The non-fixed channel for conversion is selected from among the virtual channels assigned to the given scan group in ascending order of the channel number.
4	Every time A/D conversion on a single analog channel is completed, the result of A/D conversion is stored in an A/D data register i (ADDRi (i = 0 to 35)) or A/D extended analog data register j (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to18)). When the FIFO function is in use, the result of A/D conversion is also stored in a FIFO data register k (ADFIFODRk (k = 0 to 11)).
5	If the scan end interrupt is enabled, the scan end interrupt for the given scan group is generated after the period for the round of scanning.
6	The period until the end of the round of scanning also includes the times for conversion on idle channels.
7	Until the A/D conversion stop processing proceeds, steps 2 to 6 are repeated, and the scanning operation continues. Specifically, A/D conversion is repeated on the same fixed channels and on one non-fixed channel, which is selected from among those assigned to the given scan group in ascending order of the channel number. When the last channel number is reached, the target for conversion of a non-fixed channel returns to the first channel and this operation is repeated. To stop the A/D conversion, follow the procedure shown in section 60.5.4, Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
Note 1. In the first cycle where the results of A/D conversion on the fixed channels are output, the data register for the non-fixed channel is not updated until A/D conversion has been completed the number of times equal to the number of taps in the digital filters for the fixed channels.
Note 2. The sampling state for the non-fixed channels must be set to the same value.

Figure 60.16 Example of the Scanning Operation in Hybrid Mode – Fixed Channels Continuous Scan Mode

60.3.9 Self-Calibration

ADC has a built-in self-calibration function. Self-calibration function is used to calibrate the variations of characteristics caused by the chip-by-chip variations. Self-calibration function has the following operations.

1. Internal Circuit Calibration
   - Adjusts the operation of A/D converter's internal circuitry.
2. Gain and Offset Calibration
   - Measure the A/D converter's Gain Error and Offset Error.
   - The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
     For details, see section 60.4.3.1, Gain Error and Offset Error Calibration.
   - This self-calibration should be performed after Internal-circuit self-calibration is completed.
3. Channel-dedicated sample-and-hold circuit Gain and Offset Calibration
   - Measure the Gain Error and the Offset Error when using the channel-dedicated sample-and-hold circuit.
   - The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
     For details, see section 60.4.3.1, Gain Error and Offset Error Calibration.
   - This self-calibration should be performed after A/D converter Gain/Offset self-calibration is completed.

60.3.9.1 Conditions under Which Self-Calibration is Required

The conditions under which self-calibration is required are shown in Table 60.18. Under the conditions shown in Table 60.18, self-calibration should be performed before using the A/D converter. If A/D converter is operating, stop all A/D converters and perform the self-calibration again. If the required self-calibration is not performed, the A/D conversion result is not guaranteed.

Table 60.18 List of Conditions under Which Self-Calibration should be Performed

Conditions under Which Self-Calibration is Required	Internal-Circuit Calibration	Gain/Offset Calibration	Channel-Dedicated Sample-and-Hold Circuit Gain and Offset Calibration*1
After reset release	✓	✓	✓
After releasing the module stop	✓	✓	✓
When returning from software standby mode or deep software standby mode	✓	✓	✓
When changed the ADCLK setting (When clock source or frequency is changed)	✓	✓	✓
When changed the operation mode or the scan mode of the A/D converter (When changed ADMDR.ADMDm bit (m = 0 to 2))	✓	✓*2	✓
When changed A/D successive approximation time (When changed ADCNVSTR.CSTm bit (m = 0 to 2))	✓	✓	✓
When changed the operation setting of the channel-dedicated sample-and-hold circuit*1	—	—	✓

✓: available, —: not available

Note 1. When any of ADSHCR0.SHENn bits are set to 1, or any of ADSHCR0.SHENn bits are changed, perform the channel-dedicated sample-and-hold circuits self-calibration (n = 0 to 2). Not required if the channel-dedicated sample-and-hold circuits are not used.

Note 2. When the fixed channels continuous scan mode is in use, the gain/offset calibration in SAR mode with single scan mode is also required.

60.3.9.2 Self-Calibration Procedure

Table 60.19 shows the self-calibration procedures.

Table 60.19 Procedure for Self-Calibration

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0)
2	Waiting for all A/D converters to stop	Check that all A/D converters are stopped. If A/D conversion is operating, wait until the A/D converter stops by either of the following: • Wait until all A/D conversions are complete • Force the A/D converter to stop the operation using the ADSTOPR register. For details on forced stopping of A/D conversion operation, see section 60.5.4, Force Stops the A/D Conversion Operation.
3	Setting the number of states for self-calibration	Set the number of states for the self-calibration of A/D converter to ADCALSTCR register. The values in ADCALSTCR register are commonly used for ADC0, ADC1, and ADC2. Set the value to be set in ADCALSTCR register to satisfy the range defined in section 67, Electrical Characteristics.
4	Clearing the error status flag	If error flags are detected, clear them before performing the self-calibration. (A/D converter error flag, A/D converter overflow flag, etc.)
5	Performing the self-calibration for ADC0	Performs the self-calibration for ADC0. • Internal-circuit calibration • Gain/offset calibration After starting the self-calibration, wait until it is completed.
6	Performing the self-calibration for ADC1	Performs the self-calibration for ADC1 in the same as Step 5.
7	Performing the self-calibration for ADC2	Performs the self-calibration for ADC2 in the same as Step 5.
8	Setting the self-calibration of channel-dedicated sample-and-hold circuit*1	Make settings for self-calibration of channel-dedicated sample-and-hold circuit. Enable channel-dedicated sample-and-hold circuit. Sets the number of states for channel-dedicated sample-and-hold self-calibration (ADCALSTCR register*2 and ADCALSHCR register).
9	Performing channel-dedicated sample-and-hold circuit self-calibration for SH0 to SH2.*1	Performs the self-calibration for channel-dedicated sample-and-hold circuit (SH0 to SH2) connected to ADC0. • Channel-dedicated sample-and-hold circuit gain/offset calibration after starting self-calibration, wait until it is completed.
10	Error status check*3	Check for errors in self-calibration. If no operating error occurs, self-calibration is complete.

Note: In order to increase the effectiveness of self-calibration, it is recommended to perform self-calibration for the A/D converters one by one.

Note: When the fixed channels continuous scan mode is to be used, execute steps 1 to 10 for non-fixed channels in SAR mode with single scan mode, and then change the scan mode to the fixed channels continuous scan mode and execute steps 1 to 10.

Note 1. Not required if the channel-dedicated sample-and-hold circuits are not used.

Note 2. When the self-calibration is performed for the channel-dedicated sample-and-hold circuitry, the number of sampling states at the self-calibration of the A/D converter (ADCALSTCR.CALADSST[9:0] bits) may also need to be changed. Set the value that satisfies the electrical characteristics.

Note 3. If the A/D converter error is detected by the self-calibration, it may be set outside the operation guaranteed range specified in the electric characteristics. Check the operation setting.

60.3.9.3 Restrictions on Self-Calibration

(1) Prohibition of the scanning operation during self-calibration

Do not start the scanning operation during self-calibration. When performing the scanning operation, start the scanning operation after self-calibration is completed. Operation is not guaranteed when the scanning operation is started during self-calibration.

(2) Prohibition of additional writes to the ADCALSTR register during self-calibration

After the self-calibration is started, writing to ADCALSTR register is prohibited until self-calibration is completed. Write to ADCALSTR register after self-calibration is completed. Operation is not guaranteed if this restriction is violated.

(3) Restrictions on self-calibration for channel-dedicated sample-and-hold circuit

To perform self-calibration of the channel-dedicated sample-and-hold circuits, set the corresponding SHENn bit in ADSHCR0 (n = 0 to 2) to 1 for the channel-dedicated sample-and-hold circuit unit to be used.

(4) Restrictions on the self-calibration

Self-calibration should be performed for the A/D converters one by one. During the self-calibration operation, the other A/D converters should be in idle (not in scan operation or self-calibration operation).
If this restriction is violated, the A/D conversion accuracy will be degraded due to noise during the self-calibration operation. In this case, the A/D converter characteristics are not guaranteed.
In addition, in order to enhance the effect of self-calibration, self-calibration operation should be performed under conditions with as little noise as possible.

(5) Restrictions on setting of the number of status for self-calibration

The number of states to be set in the ADCALSTCR and ADCALSHCR registers should be set to satisfy the values specified in the Electrical Characteristics. In addition, the number of states involved in self-calibration should be set to satisfy the following restrictions.
If these restrictions are violated, A/D conversion results are not guaranteed.

[Restrictions on setting of ADCALSTCR register]

- ADCALSTCR.CALADSST[9:0] bits
  - According to the type of self-calibration, set the values to satisfy those specified in the electrical characteristics.
  - If the setting value differs according to the type of self-calibration, change the register setting value for each self-calibration.
- ADCALSTCR.CALADCST[5:0] bits
  - Set the same value as ADCNVSTR.CSTm[5:0] (m = 0 to 2) bits. (ADCALSTCR.CALADCST[5:0] = ADCNVSTR.CSTm[5:0])

[Restrictions on setting of ADCALSHCR register] (only when channel-dedicated sample-and-hold circuit is used)

- ADCALSHCR.CALSHSST[7:0] bits
  - Set the same value as ADSHSTR0.SHSST[7:0] bits.
- ADCALSHCR.CALSHHST[2:0] bits
  - Set the same value as ADSHSTR0.SHHST[2:0] bits.

60.3.10 Analog Input Channel

For the correspondence between the analog input pins and the analog channel numbers, the A/D converters, and channel-dedicated sample-and-hold circuit, see Table 60.4.

60.3.11 Extended Analog Function

Extended analog function performs A/D conversion on the internal signal source of MCU. The available sources for extended analog function are shown in the following:

- Self-diagnosis voltage
- Internal reference voltage
- Temperature sensor
- D/A converter output
- VBATT1/6 voltage monitoring output

60.3.12 Self-Diagnosis Function

This A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally.

60.3.12.1 Self-Diagnosis

This A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally. Table 60.20 shows the operation mode in which the self-diagnostic function is available.

Table 60.20 List of Operation Mode for which the Self-Diagnostic Function is Available

Operation Mode – Scan Mode	Self-Diagnosis Function	
	With the Channel-Dedicated Sample-and-Hold Circuit Not in Use	With the Channel-Dedicated Sample-and-Hold Circuit in Use
SAR mode – Single scan mode	✓	✓
SAR mode – Continuous scan mode	✓	—
Oversampling mode – Single scan mode	✓	—
Oversampling mode – Continuous scan mode	—	—
Oversampling mode – Same channel continuous scan mode	—	—
Hybrid mode – Single scan mode	✓	—
Hybrid mode – Continuous scan mode	—	—
Hybrid mode – Background continuous scan mode	—	—
Hybrid mode – Fixed channels continuous scan mode	—	—

✓: available, —: not available

The procedure for performing self-diagnosis of A/D converter is shown in the following.

1. Assign the self-diagnosis channel to one of the virtual channels and configure the virtual channel settings.
2. Assign the virtual channel that the self-diagnosis channel is selected to one of the scan groups.
3. Set the self-diagnosis voltage in DIAGVAL[2:0] bits of ADSGDCRn (n = 0 to 11) register corresponding to the scan group assigned in 2 above. When the channel-dedicated sample-and-hold circuits are to be used, use the

SHDIAGx[2:0] bits in the ADSHSDCRy register (y = 0, 1; x = 0 to 11) to set the voltage for use in self-diagnosis.

4. Start the scanning operation by inputting the A/D conversion start trigger. The self-diagnosis operation is executed at the A/D conversion of the self-diagnosis channel.

60.3.12.2 Self-Diagnosis Mode

ADC has a total of nine self-diagnosis modes: three modes determined by the circuit to be used and three modes with different expected values for each of the selectable circuits. Proceed with self-diagnosis in an appropriate mode for the circuit you are using. We recommend proceeding with self-diagnosis with all three types of expected value.

- Self-diagnosis modes 1 to 3: Self-diagnosis only proceeds for the A/D converter itself.
- Self-diagnosis modes 4 to 6: Self-diagnosis proceeds for the channel-dedicated sample-and-hold circuit in differential input mode in combination with the A/D converter.
- Self-diagnosis modes 7 to 9: Self-diagnosis proceeds for the channel-dedicated sample-and-hold circuit in single-ended input mode in combination with the A/D converter.

Table 60.21, Table 60.22 lists the operation settings when the self-diagnosis function is used. When using the self-diagnosis function, follow the setting in Table 60.21, Table 60.22. If self-diagnosis operation is performed with settings other than those shown in Table 60.21, Table 60.22, the self-diagnosis result (A/D conversion result) is not guaranteed.

Table 60.21 List of Operation Setting and Expected Value for Self-Diagnosis Function

	Register Setting Value				Self-Diagnosis Operation
	ADSGDCRx	ADCHCRy		ADDOPCRCy	Expected Value of A/D Conversion Data*1, *2
Self-Diagnosis Mode	DIAGVAL[2:0]	CNVCS[6:0]	AINMD	SIGNSEL	
Self-diagnosis mode 1	100b	ADC0: 60h ADC1: 61h ADC2: 62h	1	0	0000h
Self-diagnosis mode 2	101b		1	0	8000h
Self-diagnosis mode 3	110b		1	0	7FFFh

Note: x = 0 to 8, y = 0 to 49

Note 1. Expected values indicate ideal values for results of A/D conversion to be stored in the ADEXDRn.DATA[15:0] (n = 0 to 2, 4 to 6, 8, 9, 16 to 18) or ADFIFODRx.DATA[15:0] bits. A result of A/D conversion is no less than the ideal value + 1 when a positive error is generated in terms of the precision of conversion and no greater than the ideal value - 1 when a negative error is generated.

Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.

Table 60.22 List of Operation Setting and Expected Value for Self-Diagnosis Function (with the Channel-Dedicated Sample-and-Hold Circuit in Use)

	Register Setting Value					Self-Diagnosis Operation
	ADSHSDCRy	ADCHCRy		ADSHCR0	ADDOPCRCy	Expected Value of A/D Conversion Data*1, *2
Self-Diagnosis Mode	SHDIAGx[2:0]	CNVCS[6:0]	AINMD	SHMDz	SIGNSEL	
Self-diagnosis mode 4	001b	SH0: 70h SH1: 71h SH2: 72h	1	1	0	0000h
Self-diagnosis mode 5	010b		1	1	0	8000h
Self-diagnosis mode 6	011b		1	1	0	7FFFh
Self-diagnosis mode 7	100b		1	1	0	0000h
Self-diagnosis mode 8	101b		1	1	0	8000h
Self-diagnosis mode 9	110b		1	1	0	7FFFh

Note: x = 0 to 11, y = 0, 1, z = 0 to 2

Note 1. Expected values indicate ideal values for results of A/D conversion to be stored in the ADEXDRn.DATA[15:0] (n = 0 to 2, 4 to 6, 8, 9, 16 to 18) or ADFIFODRx.DATA[15:0] bits. A result of A/D conversion is no less than the ideal value + 1 when a positive error is generated in terms of the precision of conversion and no greater than the ideal value - 1 when a negative error is generated.

Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.

60.3.12.3 Notes on Self-Diagnosis Function

(1) Notes on data format during self-diagnosis

When self-diagnosis modes 4 to 9 is performed, if a data format other than 16-bit length is set, an overflow occurs due to A/D conversion data rounding, and the A/D conversion data error is detected. To perform self-diagnosis, select the 16-bit data format.

(2) Self-diagnosis of channel-dedicated sample-and-hold circuits

Setting normal conversion and self-diagnosis for a single channel-dedicated sample-and-hold circuit in a scan group at the same time is prohibited.

60.3.13 Internal Reference Voltage

ADC can perform A/D conversion of internal reference voltage. Internal reference voltage outputs a constant voltage that is independent of the power supply voltage (VCC, VSS, AVCC0, AVSS0, VREFH0 and VREFL0). The monitoring of internal reference voltage can be used to detect variations in the analog reference supply voltage (VREFH0 and VREFL0), or abnormalities or faults in the system.
When performing A/D conversion of internal reference voltage, it is required to set the internal reference voltage monitor enable register (ADREFCR) and assign the analog channel of internal reference voltage to any of the virtual channels. See Table 60.5 for the correspondence between internal reference voltage function and the analog channels. For the sampling time when A/D conversion of internal reference voltage is performed, see section 67, Electrical Characteristics.

60.3.14 Temperature Sensor

ADC can perform A/D conversion of temperature sensor. The chip temperature can be estimated from the A/D conversion result of temperature sensor. For more details on temperature sensor, see section 62, Temperature Sensor (TEMPS).
When A/D conversion of temperature sensor is performed, temperature sensor should be configured and the analog channel of temperature sensor should be assigned to any of the virtual channels. See Table 60.5 for the correspondence between temperature sensor and analog channels. For the sampling time when A/D conversion of temperature sensor is performed, see section 67, Electrical Characteristics.

60.3.15 D/A Converter

ADC can perform A/D conversion of the voltage that is being output by a D/A converter. For details on the functionality of the D/A converter, see section 61, 12-Bit D/A Converter (DAC12R).
When A/D conversion is performed on the output of D/A converter (DA0 to DA1), the D/A converter should be configured and the analog channel of the D/A converter should be assigned to one of the virtual channels.
For the sampling time when A/D conversion is performed on the D/A converter output voltage, see section 67, Electrical Characteristics.

60.3.16 VBATT 1/6 Voltage Monitoring Output

ADC can perform A/D conversion of the VBATT 1/6 voltage monitoring output. The VBATT 1/6 voltage monitoring output is for the input voltage on the VBATT pin. For details on the VBATT pin, see section 12, Battery Backup Function (VBATTBa).
A/D converting the VBATT 1/6 voltage monitoring output requires setting the VBATTMNSELR.VBTTMNSEL bit to 1 and assigning an analog channel for the VBATT 1/6 voltage monitoring output to any of the virtual channels. For the correspondence between the VBATT 1/6 voltage monitoring output and analog channel, see Table 60.5.
For the sampling time when A/D conversion of the VBATT 1/6 voltage monitoring output is performed, see section 67, Electrical Characteristics.

60.3.17 Channel-Dedicated Sample-and-hold Circuit

ADC has a built-in channel-dedicated sample-and-hold circuit. By using multiple channel-dedicated sample-and-hold circuits, it is possible to sample multiple analog inputs simultaneously.
The conditions under which channel-dedicated sample-and-hold circuit can be used are shown in Table 60.23.

Table 60.23 Operation Mode in which Channel-Dedicated Sample-and-Hold Circuits can be Used

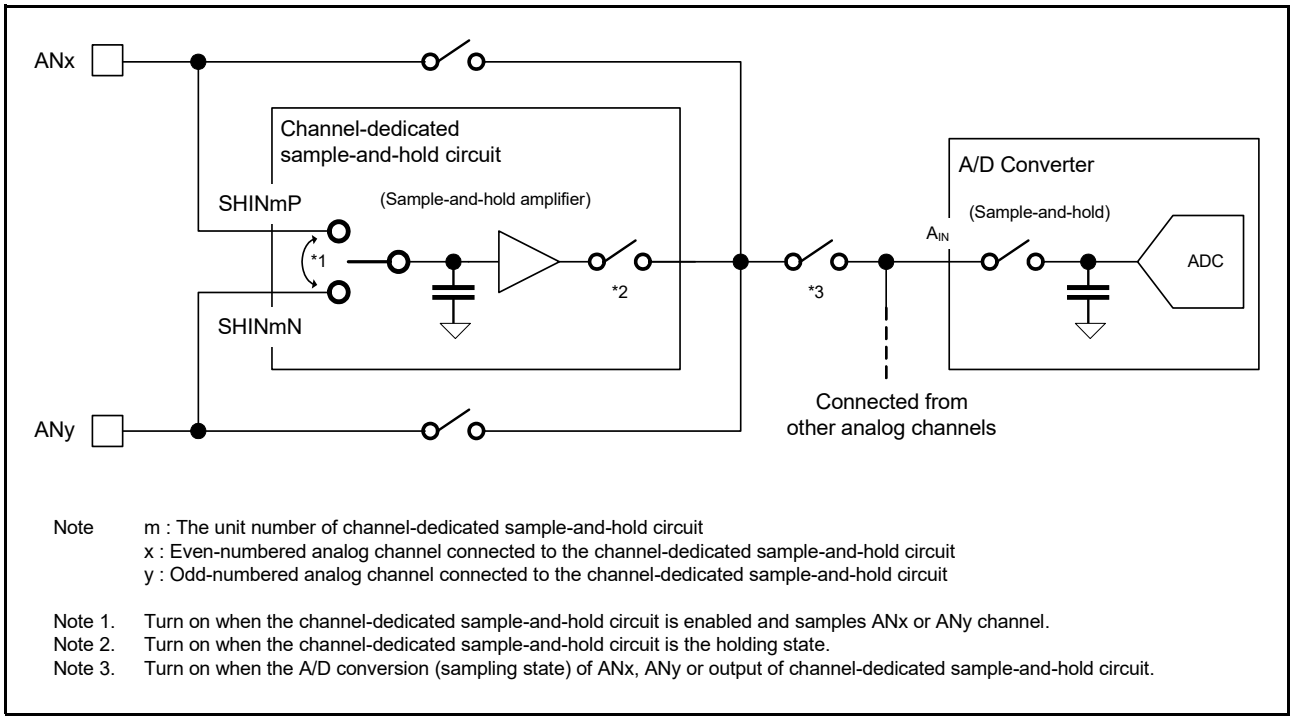
Operation Mode	Channel-Dedicated Sample-and-Hold Circuits
SAR mode – Single scan mode	✓*1
SAR mode – Continuous scan mode	—
Oversampling mode – Single scan mode	—
Oversampling mode – Continuous scan mode	—
Oversampling mode – Same channel continuous scan mode	—
Hybrid mode – Single scan mode	✓
Hybrid mode – Continuous scan mode	✓
Hybrid mode – Background continuous scan mode	✓
Hybrid mode – Fixed channels continuous scan mode	✓*2

✓: available, —: not available
Note 1. The constant sampling mode can be used.
Note 2. In this mode, fixed channels must satisfy the following conditions: the number of fixed channels is set to three and the channel-dedicated sample-and-hold circuits are specified for each of the fixed channels.

60.3.17.1 Configuration of Channel-Dedicated Sample-and-Hold Circuit

(1) Configuration in single-ended input mode

This section describes the internal configuration when using channel-dedicated sample-and-hold circuit with single-ended inputs.
Figure 60.17 shows the configuration of channel-dedicated sample-and-hold circuit in single-ended input mode.
Channel-dedicated sample-and-hold circuits are connected to two analog input channels. When channel-dedicated sample-and-hold circuit is operating in Single-ended input mode, one of the two analogue inputs (ANx or ANy) is sampled and held in one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a single-ended input signal source. (ADCHCRn.AINMD = 0 (n = 0 to 49))
If you want to perform A/D conversion of both ANx and ANy with single-ended inputs while the channel-specific sample-and-hold circuit is enabled, ANx and ANy must be assigned to different scan groups for conversion.
If you want to assign ANx and ANy to the same scan group, A/D convert one by using the channel-dedicated sample-and-hold circuit and the other with the channel-dedicated sample-and-hold circuit bypassed.



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

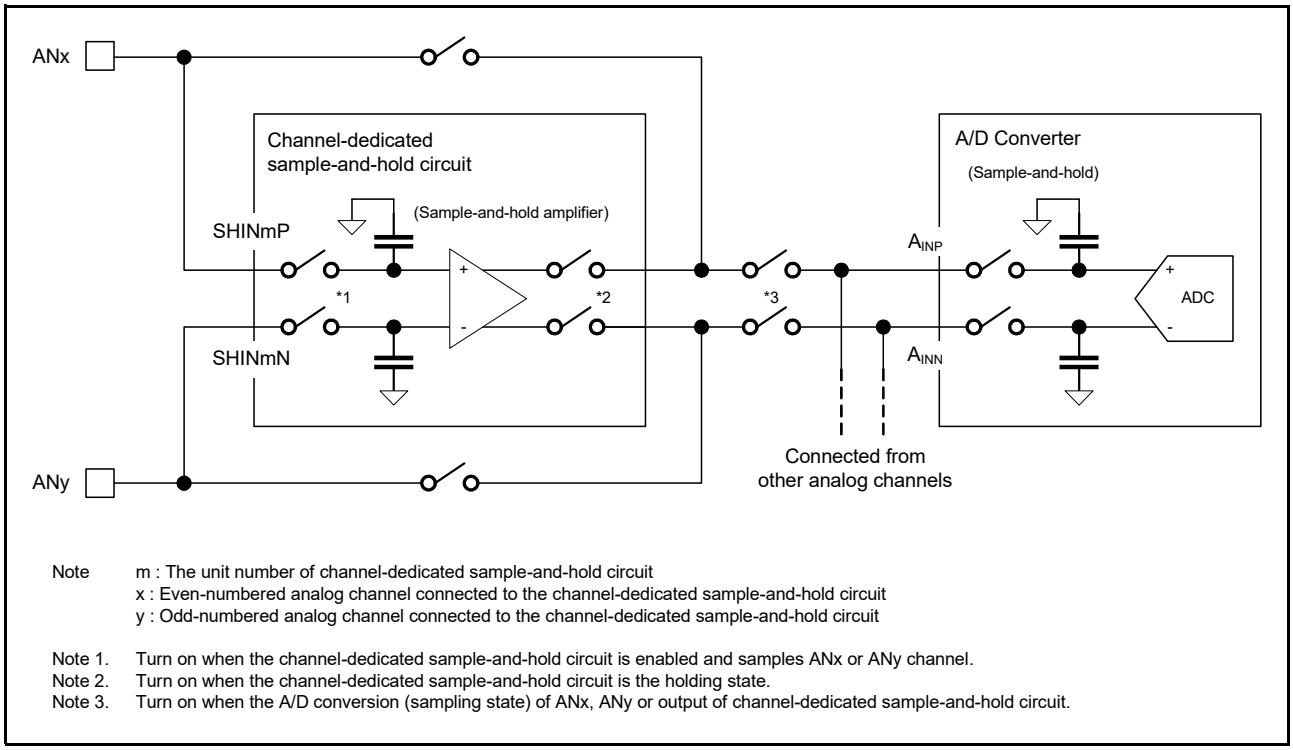
Figure 60.17 Internal Configuration of Channel-Dedicated Sample-and-Hold Circuit in Single-Ended Input Mode

(2) Configuration in differential input mode

This section describes the internal configuration when channel-dedicated sample-and-hold circuit is used with differential inputs.

Figure 60.18 shows the configuration of channel-dedicated sample-and-hold circuit in differential input mode.

When channel-dedicated sample-and-hold circuit is operating in differential input mode, the even analog channel is connected to the non-inverting input (+) (SHINmP) and the odd-numbered analog channel is connected to the inverting input (−) (SHINmN). Two analog inputs (differential input) are sampled and held in one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a differential input signal source. (ADCHCRn.AINMD = 1 (n = 0 to 49))



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

Figure 60.18 Internal Configuration of Channel-dedicated Sample-and-hold Circuit in Differential Input Mode

60.3.17.2 Operation in SAR Mode with Channel-Dedicated Sample-and-Hold Circuit

(1) Basic operation in SAR mode – Single scan mode (single-ended input mode)

This paragraph describes the basic operation of channel-dedicated sample-and-hold circuit in SAR mode. The configuration and behavior of example operation show in Table 60.24, Table 60.25 and Figure 60.19.

Table 60.24 Configuration of Example Operation for Channel-Dedicated Sample-and-Hold Circuit in SAR Mode – Single Scan Mode

Item	Configuration
A/D converter	Set ADC0 to SAR mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and single-ended input mode. (AN000, AN002 and AN004 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002, AN004 and AN006 to virtual channel 0 to 3, and set to Single-ended input mode.
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.25 Example Operation of Channel-Dedicated Sample-and-Hold Circuit in SAR Mode – Single Scan Mode

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by each channel-dedicated sample-and-hold circuit and on other analog input channels (AN006).
4	When A/D conversion of each channel completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.

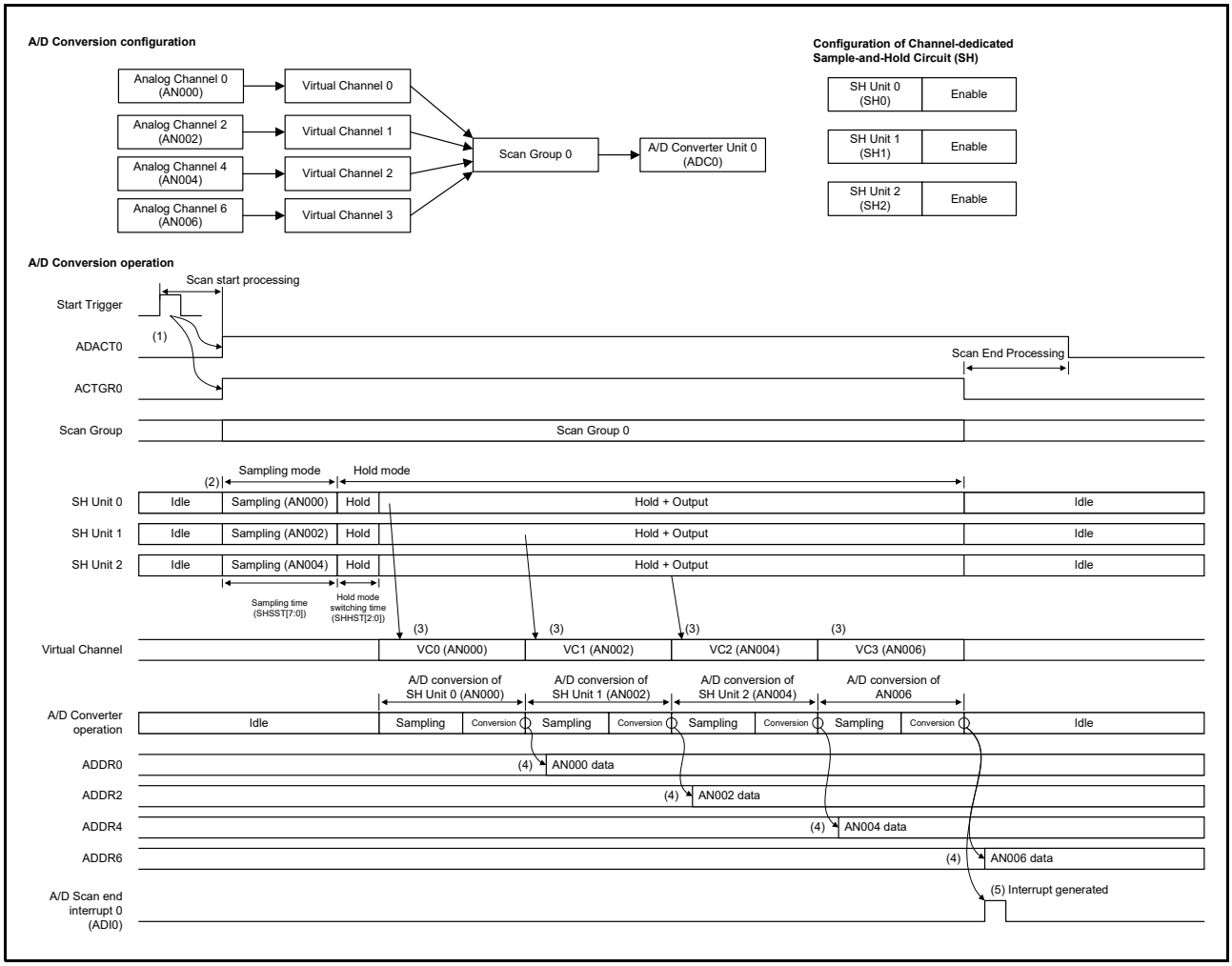


Figure 60.19 Basic Operation of Channel-Dedicated Sample-and-Hold Circuit in SAR Mode

(2) Advanced operation in SAR mode – Single scan mode

This paragraph describes the operation of A/D conversion that sharing one channel-dedicated sample-and-hold circuit between two analog channels. The operation shown in this paragraph is possible only when all of the following conditions are met.

[Operating Conditions]

- Operation mode: SAR mode – Single scan mode
- Input method: Single-ended input mode
- Others:

The following shows an example of A/D conversions of even-numbered analog channels and odd-numbered analog channels using channel-dedicated sample-and-hold circuit in SAR mode – single scan mode. The configuration example is shown in Table 60.26, and the operation detail is shown in Table 60.27 and Figure 60.20.

Table 60.26 Configuration Example of Advanced Operation for Channel-Dedicated Sample-and-Hold Circuit

Item	Configuration
A/D converter	Set ADC0 to SAR mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and single-ended input mode. (AN000, AN002 and AN004 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to single- ended input mode. • Assign AN001, AN003 and AN005 to virtual channel 3 to 5, and set to single- ended input mode.
Scan group*1	• Scan group 0: Assign AN000, AN002, AN004 (virtual channels 0 to 2). • Scan group 1: Assign AN001, AN003, AN005 (virtual channels 3 to 5). • Set scan group 0 and scan group 1 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.27 Example Advanced Operation of Channel-Dedicated Sample-and-Hold Circuit

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.
6	After scanning of scan group 0 is completed, the scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
7	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN001, AN003, AN005) and holds them.
8	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If scan end interrupt for scan group 1 is set to enabled, a scan end interrupt occurs.

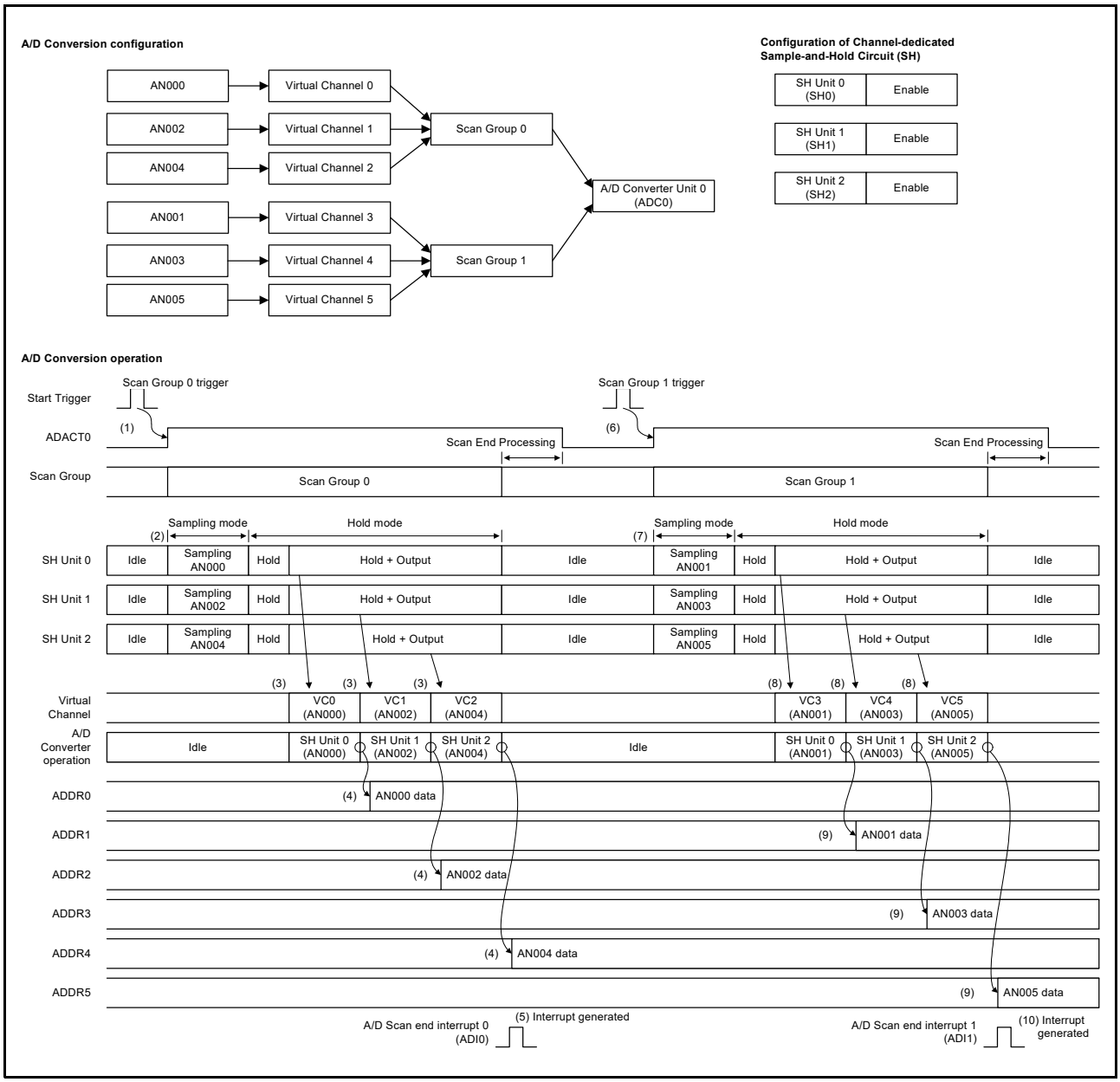


Figure 60.20 The Example Operation when Channel-Dedicated Sample-and-Hold Circuit is Shared with Two Analog Channels in SAR Mode

(3) Advanced operation 2 in SAR mode with single scan mode

The following passages describe the operation of A/D conversion when one of two analog channels is used with a channel-dedicated sample-and-hold circuit and the other bypasses the channel-dedicated sample-and-hold circuit. The operation shown in the passages is possible only when all of the following conditions are met.

[Operating Conditions]

- Operation mode: SAR mode – Single scan mode
- Input method: Single-ended input mode
- Other condition: Settings to use and to bypass the channel-dedicated sample-and-hold circuits in single-ended input mode (ADSHSBPCR.SHSBPn = 0, ADSHSBPCR.SHSBNn = 1 (n = 0 to 2))

The following tables and figure give an example of A/D conversion on even-numbered analog channels with the channel-dedicated sample-and-hold circuits in use and odd-numbered analog channels with the channel-dedicated sample-and-hold circuits bypassed in SAR mode with single scan mode. The steps for the example configuration are listed in Table 60.28, and details on the operation are given in Table 60.29 and Figure 60.21.

Table 60.28 Configuration Example of Advanced Operation for Channel-Dedicated Sample-and-Hold Circuit 2

Item	Configuration
A/D converter	Set ADC0 to SAR mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and single-ended input mode. Make the settings to bypass the channel-dedicated sample-and-hold circuits in single-ended input mode for odd-numbered analog channels. (AN000, AN002, and AN004 signals are A/D converted with the channel-dedicated sample-and-hold circuits in use and AN001, AN003, and AN005 signals are A/D converted with the channel-dedicated sample-and-hold circuits bypassed.)
Virtual channel	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to single- ended input mode. • Assign AN001, AN003 and AN005 to virtual channel 3 to 5, and set to single- ended input mode.
Scan group	• Scan group 0: Assign AN000, AN002, AN004, AN001, AN003, AN005 (virtual channels 0 to 5). • Set scan group 0 to convert with ADC0.

Table 60.29 Example Advanced Operation of Channel-Dedicated Sample-and-Hold Circuit 2

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
4	Sampling of the odd-numbered analog inputs (AN001, AN003, AN005) is started and A/D conversion then proceeds.
5	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
6	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.

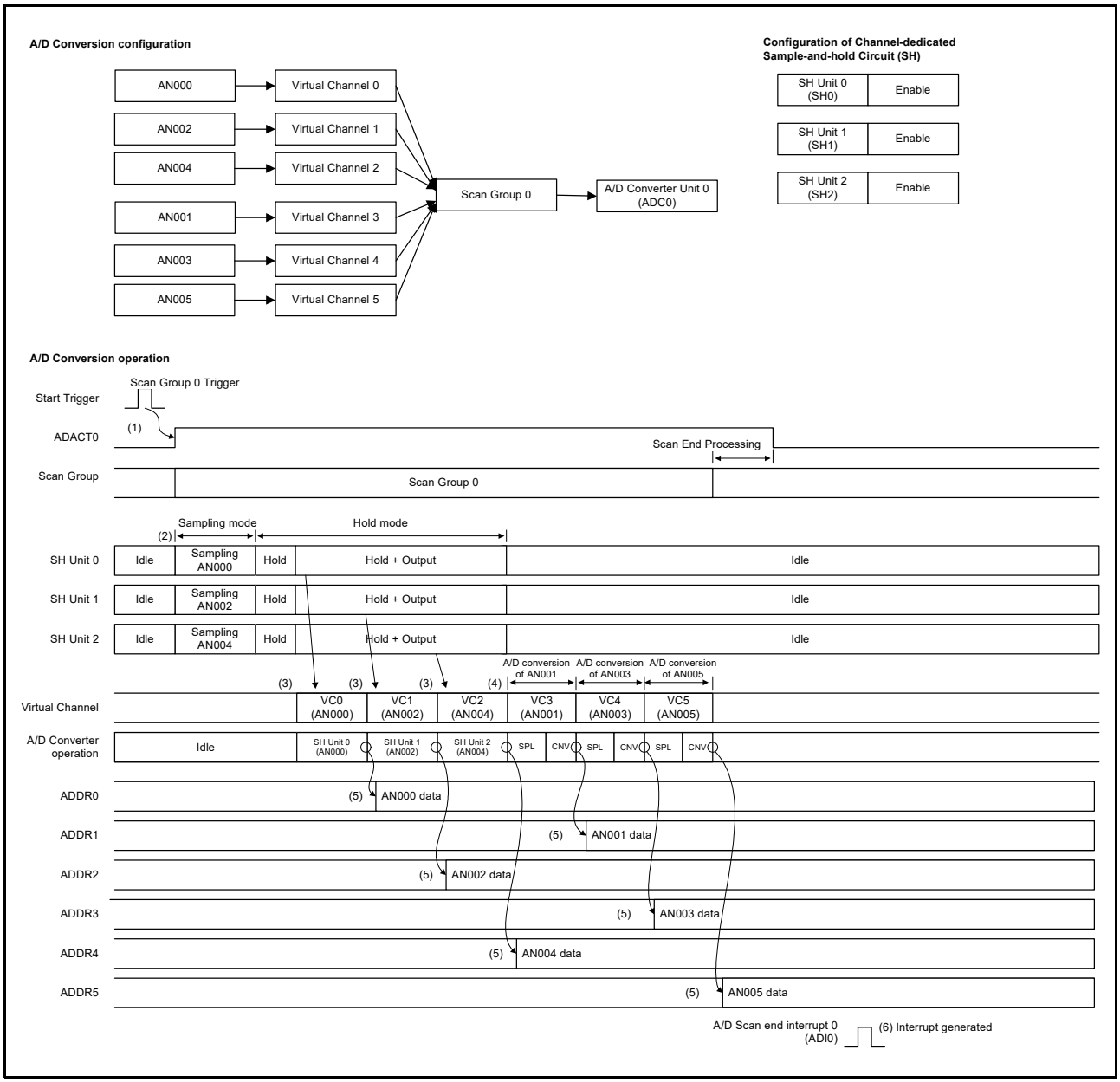


Figure 60.21 Example of A/D Conversion Operations in SAR Mode When the Channel-Dedicated Sample-and-Hold Circuits are Used with the Even-Numbered Analog Channels but Bypassed for the Odd-Numbered Analog Channels

(4) SAR mode (with constant sampling enabled)

When the constant sampling function is enabled, the channel-dedicated sample-and-hold circuits start sampling before A/D conversion proceeds. The circuits simultaneously proceed with sampling on multiple channels, so the results of A/D conversion are for voltages that were captured at the same time.
When the constant sampling function is in use, the channel-dedicated sample-and-hold circuits are assigned to the virtual channels specified for use in conversion. Only the settings listed in Table 60.30 are specifiable.

Table 60.30 Virtual Channel Numbers Which Should be Assigned as the Individual Fixed Channels

Item	Configuration
Channel-dedicated sample-and-hold circuit 0	Virtual channel 0
Channel-dedicated sample-and-hold circuit 1	Virtual channel 1
Channel-dedicated sample-and-hold circuit 2	Virtual channel 2

A comparison of operations with the constant sampling function enabled and disabled is given in Figure 60.22.

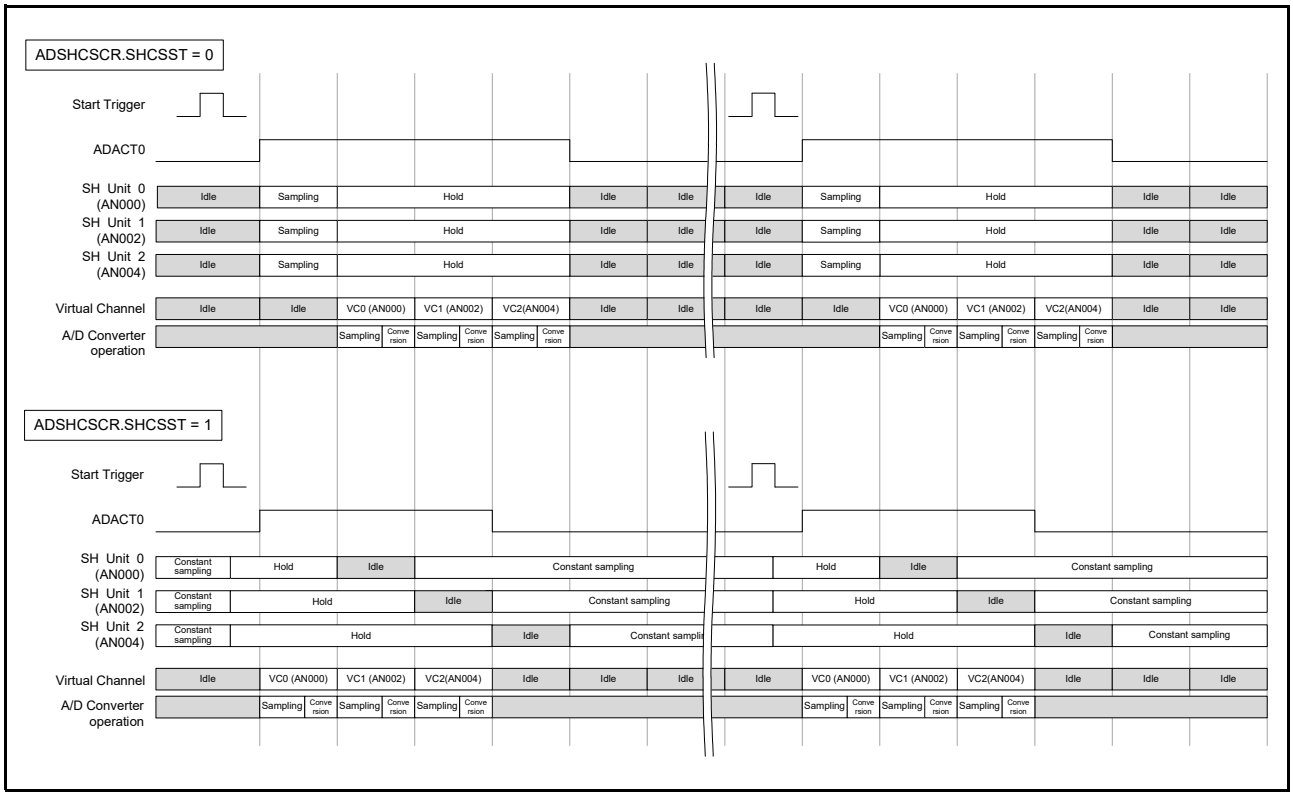


Figure 60.22 Comparison of Operations with the Constant Sampling Function Disabled (SHCSST = 0) and Enabled (SHCSST = 1)

The steps in an example of making the settings for operation are listed in Table 60.31, and an example of operation is given in Table 60.32 and Figure 60.23.

Table 60.31 Example of Making the Settings for Operation with the Constant Sampling Function Enabled

Item	Configuration
A/D converter	Set ADC0 to SAR mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and single-ended input mode. (AN000, AN002 and AN004 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.) Enable constant sampling of SH0 to SH2.
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Single-ended input mode.
Scan group*1	• Assign virtual channel 0 to 2 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.32 Example of Operation with the Constant Sampling Function Enabled

Step	Operation
1	Setting the SHCSST0 to SHCSST2 bits in the ADSHCSCR register to 1 starts constant sampling by the channel-dedicated sample-and-hold circuits (SH0 to SH2).
2	When a trigger for scan group 0 is input, holding of analog input voltages on the AN000, AN002, and AN004 pins starts.
3	After the stabilization time of the channel-dedicated sample and hold circuit has elapsed, A/D conversion is performed on the voltage held by the channel-dedicated sample and hold circuit.
4	When A/D conversion of each channel completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn). The channel-specific sample-and-hold circuitry resumes full-time sampling after an idle period.
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.
6	Setting the SHCSST0 to SHCSST2 bits in the ADSHCSCR register to 0 stops constant sampling by the channel-dedicated sample-and-hold circuits (SH0 to SH2).

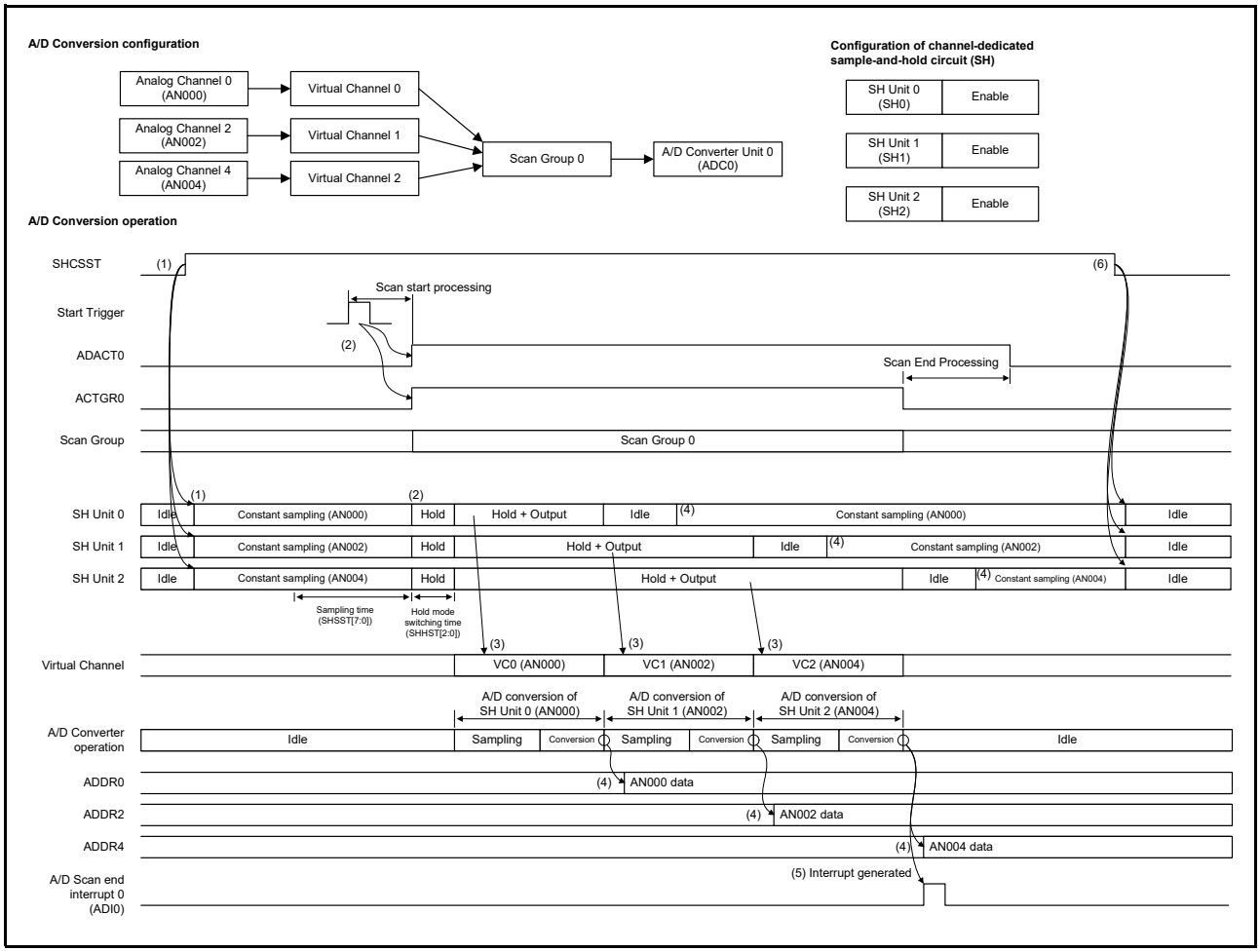


Figure 60.23 Constant Sampling Operation in SAR Mode

(5) Advanced operation (in differential input mode) with the disconnection detection assist function in use in SAR mode

When the differential input mode is selected for the channel-dedicated sample-and-hold circuits, using the differential bypass function allows the detection of disconnection by the disconnection detection assist function of the A/D converter (in single-ended input mode). This is intended to avoid the complicated self-calibration operations required in changing the channel-dedicated sample-and-hold circuits from differential input mode to single-ended input mode because the disconnection detection assist function is only supported in single-ended input mode.

The table below lists the steps in an example of making the settings for operation. In this example, A/D conversion with the use of the channel-dedicated sample-and-hold circuit proceeds in differential input mode in scan group 0, and the disconnection detection assist function is then executed in scan group 1 without changing the input mode.

Table 60.33 Example of Settings for Using the Disconnection Detection Assist Function of the A/D Converter with the Channel-Dedicated Sample-and-Hold Circuit (in Differential Input Mode)

Item	Configuration
A/D converter	Set ADC0 to SAR mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to enabled and differential input mode. (AN000 and AN001 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel	• Assign AN000 to virtual channel 0, and set to differential input mode. (This will function AN000 as the non-inverting inputs (+) of the differential input, and AN001 as the inverting inputs (–) of the differential input.) • Assign AN000 and AN001 to virtual channel 1, 2, and set to Single-ended input mode.
Scan group	• Scan group 0: Virtual channel 0 • Scan group 1: Virtual channel 1, 2 • Make the settings to use ADC0 for A/D conversion of scan groups 0 and 1. • Scan group 0: Make the setting to use the channel-dedicated sample-and-hold circuits (ADSHDBPCR.SHDBP0 = 0) • Scan group 1: Make the setting to bypass the channel-dedicated sample-and-hold circuits (ADSHDBPCR.SHDBP1 = 1) • Scan group 1: Enable the disconnection detection assist function.

60.3.17.3 Operation in Hybrid Mode with Channel-Dedicated Sample-and-Hold Circuit

(1) Basic operation in hybrid mode – Single scan mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in hybrid mode – single scan mode. Since differential input mode is supported in hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to differential input mode. The configuration and behavior of example operation show in Table 60.34, Table 60.35 and Figure 60.24.

Table 60.34 Configuration of Example Operation for Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Single Scan Mode

Item	Configuration
A/D converter	Set ADC0 to hybrid mode – single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (–) of the differential input.)
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.35 Example Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Single Scan Mode

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	A/D converter and channel-dedicated sample-and-hold circuit enter the idle state when scanning operation is completed. The oversampling data in digital filter are discarded at the end of the scanning operation.

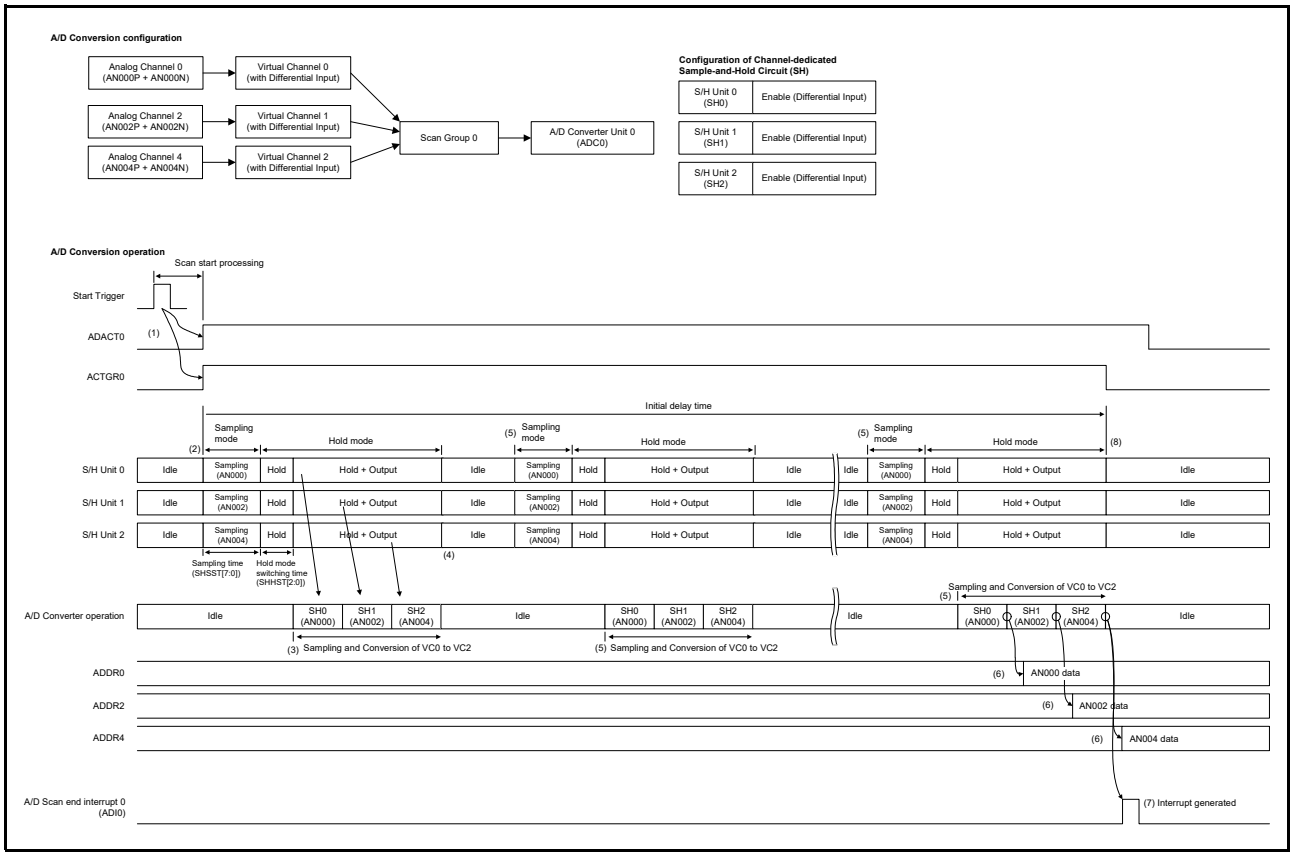


Figure 60.24 Basic Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Single Scan Mode

(2) Basic operation in hybrid mode – Continuous scan mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in hybrid mode – continuous scan mode. Since differential input mode is supported in hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to differential input mode. The configuration and behavior of example operation show in Table 60.36, Table 60.37 and Figure 60.25.

Table 60.36 Example Operation Setting of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Continuous Scan Mode

Item	Configuration
A/D converter	Set ADC0 to hybrid mode – continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (–) of the differential input.)
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.37 Example Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Continuous Scan Mode

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 11)).
9	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
10	Thereafter, until the forcibly stop process is performed, Step 8 to 9 are repeated, and the scanning operation continues. To perform the forcibly stop, follow to section 60.5.4, Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

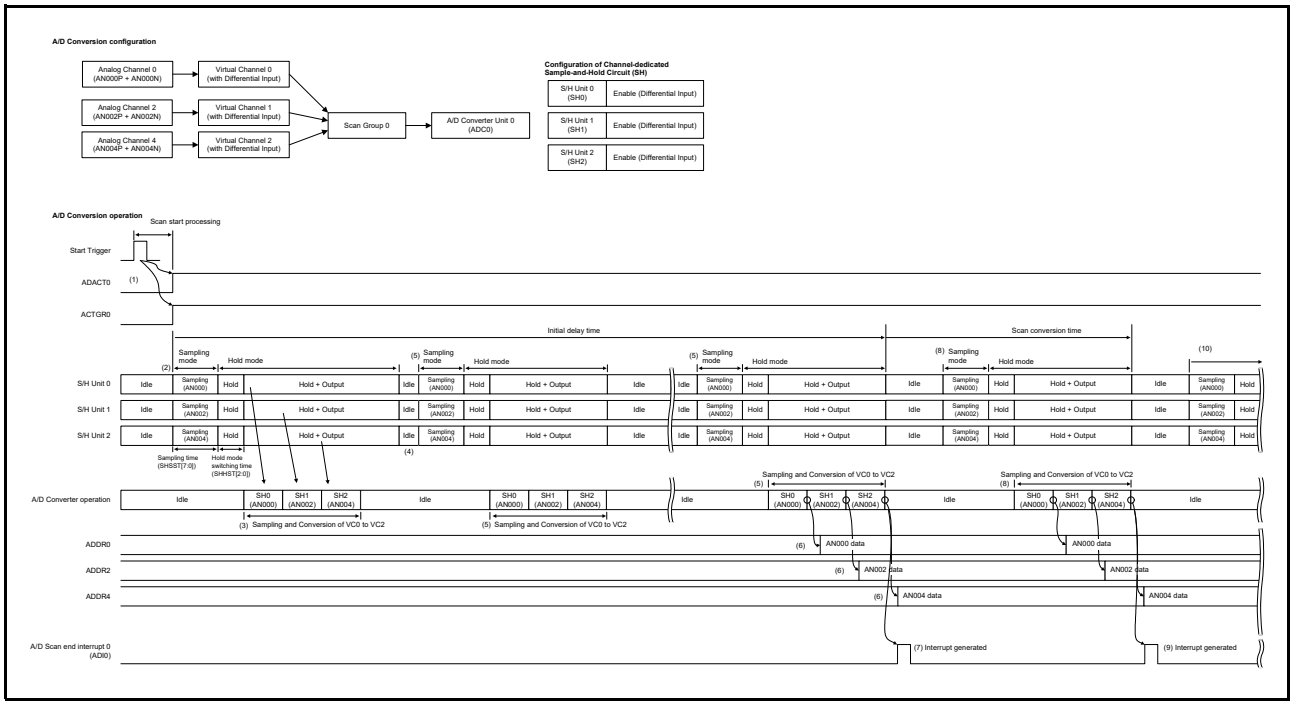


Figure 60.25 Basic Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Continuous Scan Mode

(3) Basic operation in hybrid mode – Background continuous scan mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in hybrid mode – background continuous scan mode. Since differential input mode is supported in hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to differential input mode. The configuration and behavior of example operation show in Table 60.38, Table 60.39 and Figure 60.26.

Table 60.38 Configuration of Example Operation for Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Background Continuous Scan Mode

Item	Configuration
A/D converter	Set ADC0 to hybrid mode – Background continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (–) of the differential input.)
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.39 Example Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Background Continuous Scan Mode

Step	Operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2) .
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated. After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, first A/D conversion data becomes in ready to be output. Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each time oversampling is performed for each analog channel, the data in the digital filter is updated. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data becomes in ready to be output.
7	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 35), ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)). When FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 11)). Also, if scan end interrupt is set to enable, scan end interrupt is generated. If FIFO interrupt is set to enable, FIFO data read request is generated when the condition for generating it is met.
8	Thereafter, until the forcibly stop process is performed, background continuous scanning operation (Step 6) is repeated. And whenever the A/D conversion start trigger is input during background continuous scanning operation, the A/D conversion data is output (Step 7). To perform the forcibly stop, follow to section 60.5.4, Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

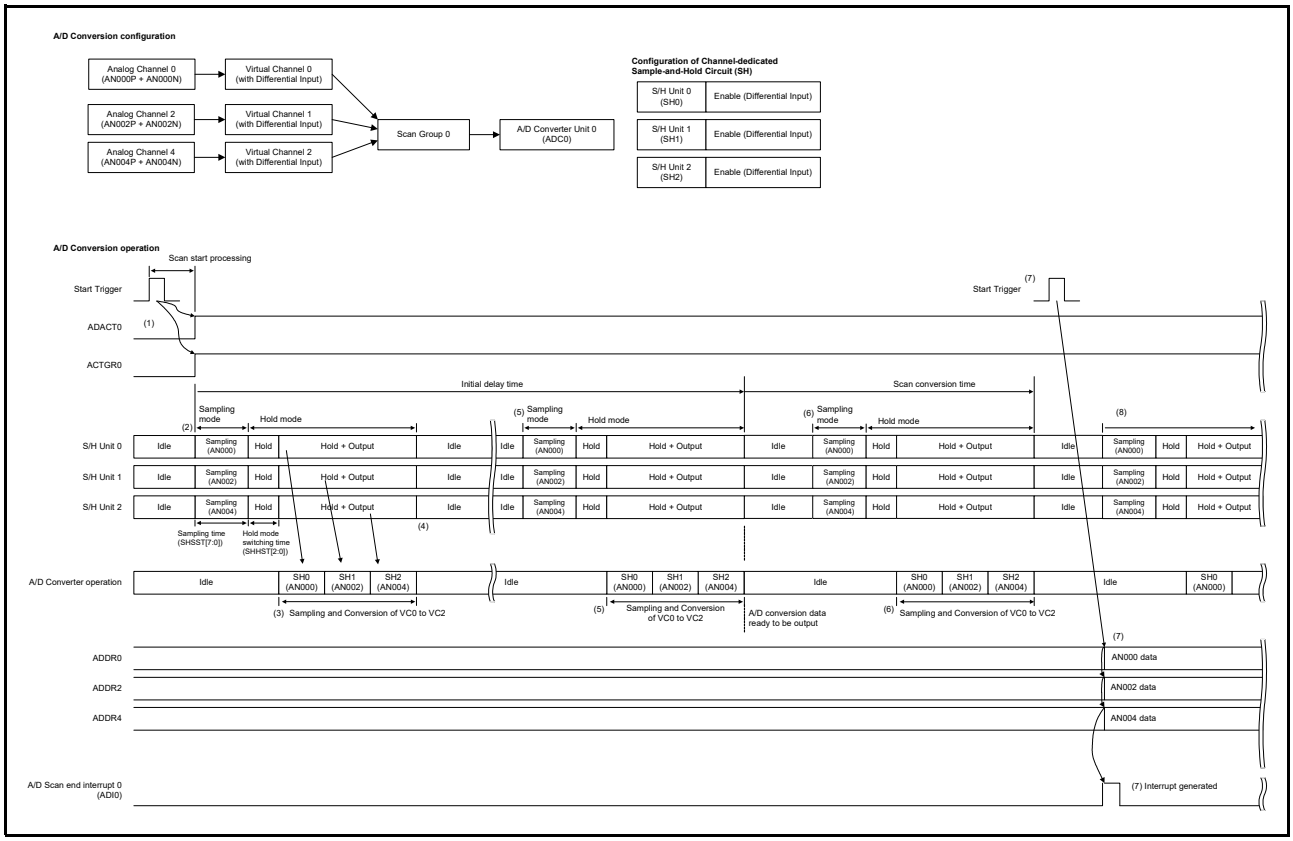


Figure 60.26 Basic Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Background Continuous Scan Mode

(4) Basic operation in hybrid mode with fixed channels continuous scan mode

The following passages describe the basic operation of the channel-dedicated sample-and-hold circuits in hybrid mode with fixed channels continuous scan mode. The steps in an example of making the settings for operation are listed in Table 60.40 and an example of operation is given in Table 60.41. This mode can only be used under the following conditions: the number of fixed channels is three and use of the channel-dedicated sample-and-hold circuits is only specified for the fixed channels.

Table 60.40 Configuration of Example Operation for Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Fixed Channels Continuous Scan Mode

Item	Configuration
A/D converter	Set ADC0 to hybrid mode – Fixed channels continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (–) of the differential input.)
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

Table 60.41 Example Operation of Channel-Dedicated Sample-and-Hold Circuit in Hybrid Mode – Fixed Channels Continuous Scan Mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	The three channel-dedicated sample-and-hold circuits (SH0 to SH2) specified as the fixed channels start sampling the analog inputs (AN000, AN002, AN004) and hold the voltages on those pins.
3	The output signals of the channel-dedicated sample-and-hold circuits (SH0 to SH2) are A/D converted in hybrid mode.
4	A/D conversion in SAR mode proceeds on one non-fixed channel. The non-fixed channel for conversion is selected from among the virtual channels assigned to the given scan group in ascending order of the channel number.
5	After A/D conversion on the three fixed channels and a single non-fixed channel, the channel-dedicated sample-and-hold circuits release the held voltages and enter the idle state.
6	Every time A/D conversion on a single analog channel is completed, the result of A/D conversion is stored in an A/D data register i (ADDRi (i = 0 to 35)) or A/D extended analog data register j (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to18)). When the FIFO function is in use, the result of A/D conversion is also stored in a FIFO data register k (ADFIFODRk (k = 0 to 11)).
7	If the scan end interrupt is enabled, the scan end interrupt for the given scan group is generated after the period for the round of scanning.
8	The period until the end of the round of scanning also includes the times for conversion on idle channels.
9	Until the A/D conversion stop processing proceeds, steps 2 to 6 are repeated, and the scanning operation continues. Specifically, A/D conversion is repeated on the same fixed channels and on one non-fixed channel, which is selected from among those assigned to the given scan group in ascending order of the channel number. When the last channel number is reached, the target for conversion of a non-fixed channel returns to the first channel and this operation is repeated. To stop the A/D conversion, follow the procedure shown in section 60.5.4, Force Stops the A/D Conversion Operation.

60.3.17.4 Restrictions on Channel-Dedicated Sample-and-Hold Circuit

When using channel-dedicated sample-and-hold circuit, observe the following restrictions:

(1) Limitations on usage of channel-dedicated sample-and-hold circuit

Channel-dedicated sample-and-hold circuit is only available in the following:

- SAR mode – Single scan mode
- Hybrid mode – Single scan mode
- Hybrid mode – Continuous scan mode
- Hybrid mode – Background continuous scan mode
- Hybrid mode – Fixed channels continuous scan mode*1

In the operation mode other than above, channel-dedicated sample-and-hold circuit is prohibited.

Set the sampling time of the channel-dedicated sample-and-hold circuit to at least 3.5 times the sum of the following times:

- Hold mode switching time
- Sampling times of the A/D converter units that use the channel-dedicated sample-and-hold circuits in the same scan group
- A/D successive approximation times set for the virtual channel numbers to which the channel-dedicated sample-and-hold circuits are assigned

This relationship is expressed by the following equation:

$$ADSHSTR0.SHSST[7:0] \geq (ADSHSTR0.SHHST[2:0] + \Sigma_{i=1}^{n} ADSSTRx.SSTy[9:0] + \Sigma_{i=1}^{n-1} ADCNVSTR.CSTm[5:0]\ ) \times 3.5$$

(x = 0 to 7, y = 0 to 15, m = 0 to 2, i: Number of rounds of A/D conversion on the virtual channels that use the channel-dedicated sample-and-hold circuits in the same scan group (1 to n))

Note 1. Only fixed channels are supported when fixed channels are set to 3 channels and all fixed channels enable channel dedicated sample-and-hold circuit. Non-fixed channels cannot use channel-dedicated sample-and-hold circuit.

(2) Restrictions on self-calibration operation of channel-dedicated sample-and-hold circuit

Self-calibration operation is required when channel-dedicated sample-and-hold circuit is used. For more details on self-calibration procedures for channel-dedicated sample-and-hold circuit and the restrictions on it, see section 60.3.9, Self-Calibration.

(3) Restrictions on assigning virtual channels and scan groups

For analog channels using channel-dedicated sample-and-hold circuit, assign them to virtual channels so that A/D conversion is performed at the front of the scan group. Specifically, assign an analog channel that uses channel-dedicated sample-and-hold circuit from the virtual channel that has the smallest virtual channel number among the virtual channels assigned to that scan group. (Assign the analog channels of channel-dedicated sample-and-hold circuit to the front virtual channels in ascending order of the virtual channel number.)
If this restriction is violated, the operation of channel-dedicated sample-and-hold circuit and A/D converter is not guaranteed.

(4) Prohibition of multiple assignments of channels using the same channel-dedicated sample-and-hold circuit in the same scan group

It is prohibited to assign more than one virtual channel that uses the same channel-dedicated sample-and-hold circuit to a single scan group. Specific restrictions are the followings:

[Restrictions]

- When not using the bypass function, the virtual channel whose signal source is selected as the even-numbered analog channel connected to channel- dedicated sample-and-hold circuit and the virtual channel whose signal source is selected as the odd-numbered analog channel must not be assigned to the same scan group.
- It is prohibited to assign the even-numbered analog channels or odd-numbered analog channels connected to channel- dedicated sample-and-hold circuit to multiple virtual channels, and assign them to the same scan group.

Operation is not guaranteed if this restriction is violated.
To avoid this restriction, assign virtual channels that use the same channel-dedicated sample-and-hold circuit to separate scan groups. This restriction can be avoided by preventing A/D conversion from occurring more than once for the same channel-dedicated sample-and-hold circuit during one scanning operation. For the example workaround for this restriction, see (2) Advanced operation in SAR mode – Single scan mode in section 60.3.17.2, Operation in SAR Mode with Channel-Dedicated Sample-and-Hold Circuit in SAR Mode.

(5) Restriction on the differential bypass function

When a scan group uses the channel-dedicated sample-and-hold circuits with the use of differential bypass inputs, the disconnection detection assist function and constant sampling function for use with the channel-dedicated sample-and-hold circuits cannot be used in any of the channel-dedicated sample-and-hold circuits, including in cases where they are in use by another scan group.

(6) Restrictions on group priority operation

Only a scan group with the highest priority can use the channel-dedicated sample-and-hold circuits in group priority operation.

(7) Restrictions on the constant sampling function

The constant sampling function can only be used in SAR mode with single scan mode.
Set the constant sampling function after self-calibration and before A/D conversion.
The constant sampling function is only available for the virtual channels listed in Table 60.30.
Set the sampling time of the channel-dedicated sample-and-hold circuit which uses the constant sampling function as follows.

- When synchronization between units is disabled:
  ADSHSTR0.SHSST[7:0] = 04h (initial value)
- When synchronization between units is enabled:
  Set the sampling time and hold mode switching time of the channel-dedicated sample-and-hold circuit so that the sum of the two is equal to the synchronous operation period. In addition, set the synchronous operation period to at least 6 cycles of ADCLK, which is the minimum value of sum of the sampling time and hold mode switching time.
  ADSHSTR0.SHSST[7:0] + ADSHSTR0.SHHST[2:0] = ADSYCR.ADSYCYC[7:0]
  ADSYCR.ADSYCYC[7:0] ≥ 6 ADCLK

When the constant sampling function is to be used, the sampling period of the channel-dedicated sample-and-hold circuit is from completion of conversion of the signal from a channel-dedicated sample-and-hold circuit until conversion of the next signal from the channel-dedicated sample-and-hold circuit starts. Therefore, in cases of the input of successive conversion start triggers, secure sufficient intervals between the triggers to ensure that the sampling period of the channel-dedicated sample-and-hold circuit is secured.

(8) Restriction on hybrid mode with fixed channels continuous scan mode

The fixed channels continuous scan mode for the channel-dedicated sample-and-hold circuits can only be used under the following conditions: the number of fixed channels is three and use of the channel-dedicated sample-and-hold circuits is only specified for the fixed channels.

60.3.18 Disconnection Detection Assist Function

The disconnection detection assist function is the function that fixes the charges of the sampling capacitance of the A/D converter or of the channel-dedicated sample-and-hold circuits to a specified condition prior to starting A/D conversion. This function can be used to detect the wiring disconnection connected to analog input.
Table 60.42 shows the operation modes in which the disconnection detection assist function is available. In addition, when this function is to be used, the use of the high accuracy mode is recommended.

Table 60.42 List of Operation Mode for which Disconnection Detection Assist Function is Available

Operation Mode – Scan Mode	Disconnection Detection Assist Function	
	With the Channel-Dedicated Sample-and-Hold Circuit Not in Use	With the Channel-Dedicated Sample-and-Hold Circuit in Use
SAR mode – Single scan mode	✓	✓
SAR mode – Continuous scan mode	✓	—
Oversampling mode – Single scan mode	—	—
Oversampling mode – Continuous scan mode	—	—
Oversampling mode – Same channel continuous scan mode	—	—
Hybrid mode – Single scan mode	✓	—
Hybrid mode – Continuous scan mode	✓	—
Hybrid mode – Background continuous scan mode	—	—
Hybrid mode – Fixed channels continuous scan mode	—	—

✓: Available, —: Not Available

Figure 60.27 shows an example operation of A/D conversion when disconnection detection assist function is used.
Figure 60.28 shows an example operation of A/D conversion when disconnection detection assist function and channel-dedicated sample-and-hold circuit are used.
Figure 60.29 shows an example of the disconnection detection when precharge is selected, and Figure 60.30 shows an example of the disconnection detection when discharge is selected.

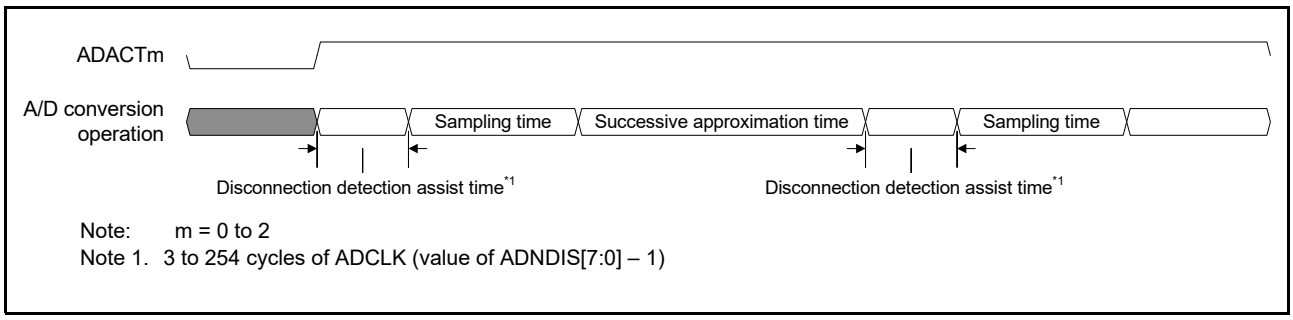
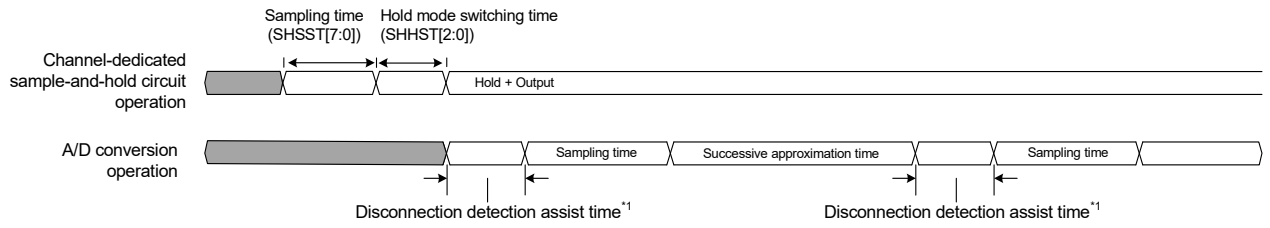
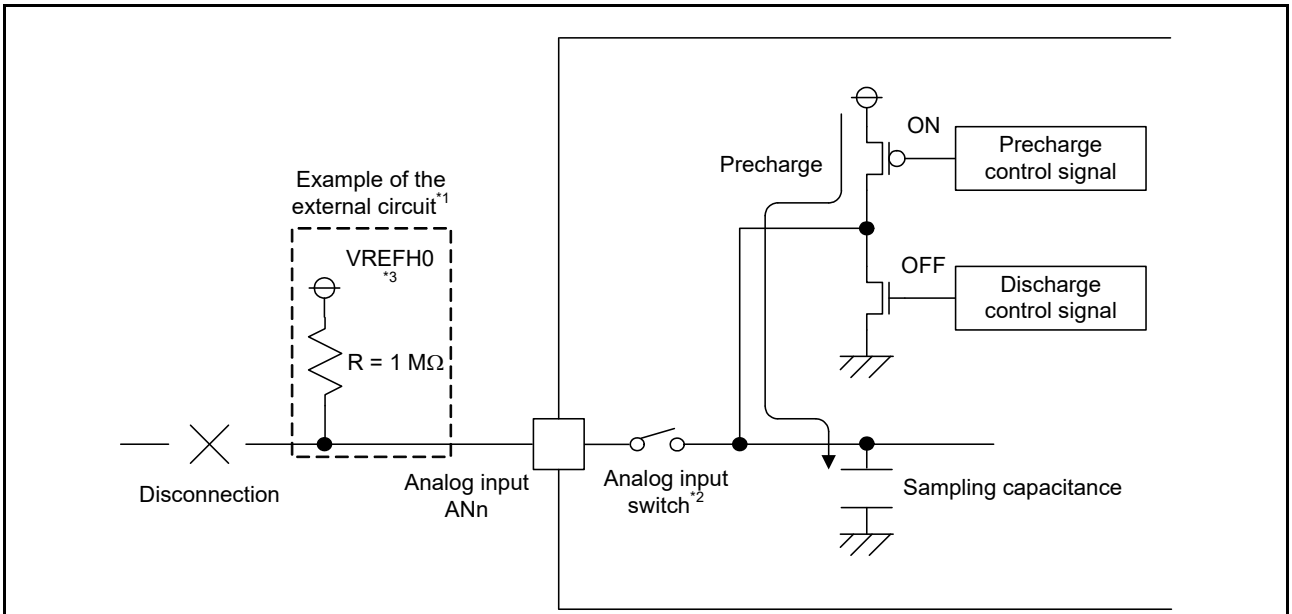


Figure 60.27 Operation of A/D Conversion when Disconnection Detection Assist Function is Used



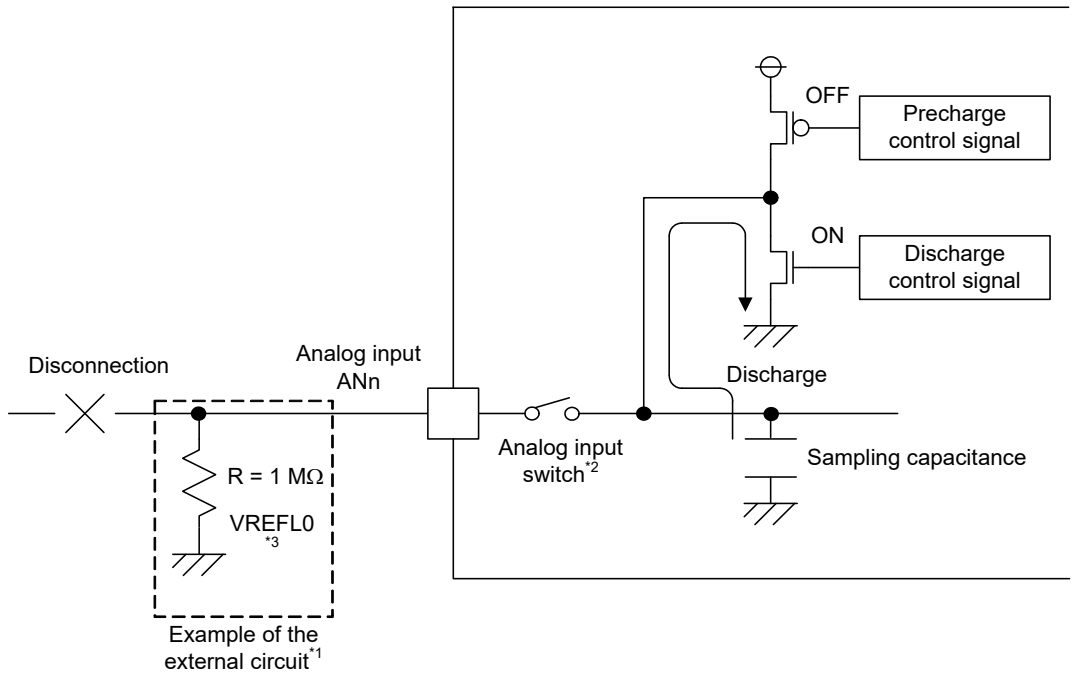
Note 1. 3 to 254 cycles of ADCLK (value of ADNDIS[7:0] – 1)

Figure 60.28 Operation of A/D Conversion when the Disconnection Detection Assist Function and Channel-Dedicated Sample-and-Hold Circuit are Used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when precharge, and turn on when sampling the Analog input ANn.
Note 3. For the 320-pin LFBGA, use VREFH00, VREFH01, and VREFH02 for ADC0, ADC1, and ADC2, respectively.

Figure 60.29 Example of Disconnection Detection when Precharge is Selected



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when discharge and turn on when sampling the Analog input ANn.
Note 3. For the 320-pin LFBGA, use VREFL00, VREFL01, and VREFL02 for ADC0, ADC1, and ADC2, respectively.

Figure 60.30 Example of Disconnection Detection when Discharge is Selected

60.3.19 Group Priority Operation

Group priority operation function is the function that performs the scanning operation (A/D conversion) for a scan group with a higher priority based on the priority of scan group.
The priority of scan groups is group 0 > group 1 > … > group n. (n: the number of maximum scan group - 1)
Table 60.43 shows the conditions when group priority operation can be used and the corresponding register settings. Set the operation according to Table 60.43.

Table 60.43 Group Priority Operation Usable Conditions and Corresponding Register Settings

Operating Conditions		Group Priority Operation				
Operation mode of A/D converter		Operation	Setting for ADGSPCR register*1			
			PGSm	RSCNm	LGRRSm	GRPm
SAR mode	Single scan mode	✓	1	1	1	0
	Continuous scan mode	✓	1	1	1	1
Oversampling mode		—	0	0	0	0
Hybrid mode		—	0	0	0	0

✓: Available, —: Not Available
m = 0 to 2
Note 1. Settings other than the specified value are prohibited.

60.3.19.1 Group Priority Operation in SAR Mode – Single Scan Mode

In group priority operation in SAR mode – Single scan mode, the single scanning operation is performed according to the priority of scan group.
If an A/D conversion start trigger for the high priority group is input during a scanning operation of the low priority group, the scanning operation of the low priority group is interrupted, and the scanning operation of the high priority group is started. After the scanning operation of the high priority group is completed, the scanning operation of the low priority group is resumed.
When an A/D conversion start trigger of the low priority group during a scanning operation of the high priority group is input, the scanning operation of the low priority group is performed after the scanning operation of the high priority group is completed.

Example of the Group Priority Operation in SAR Mode – Single Scan Mode

Table 60.44 and Figure 60.31 show the example of group priority operation in SAR mode – Single scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high priority) and analog channel 1 to 3 (AN001 to AN003) are assigned to scan group 1 (low priority).

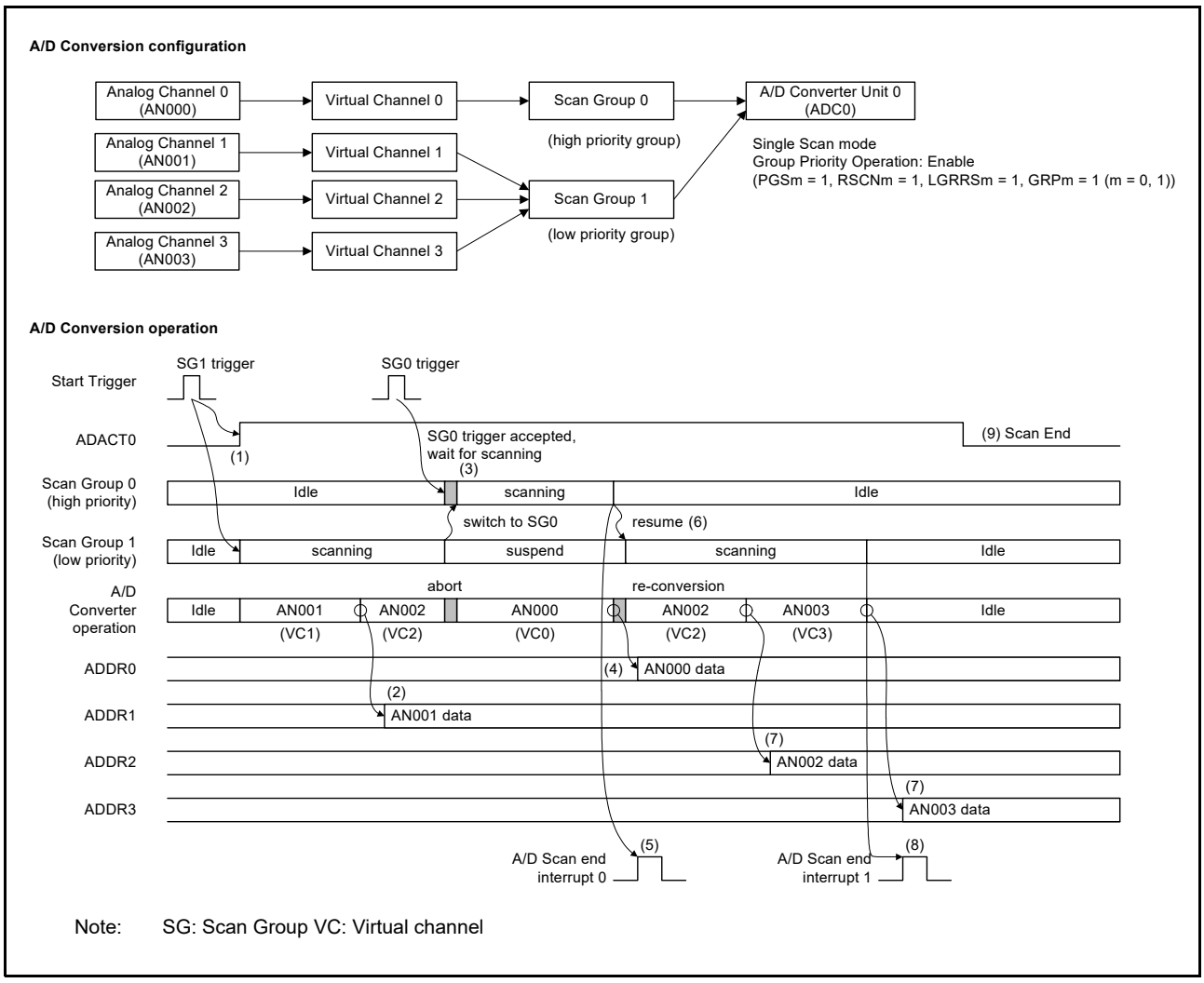
Table 60.44 Example of the Group Priority Operation in SAR Mode – Single Scan Mode (2 groups) (1/2)

Step	Operation
1	The scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started.*1
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel.*1
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).

Table 60.44 Example of the Group Priority Operation in SAR Mode – Single Scan Mode (2 groups) (2/2)

Step	Operation
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	When all scanning operations are completed, ADSR.ADACTm (m = 0, 1) bits are cleared and the A/D converter enters the idle status.

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 60.3.19.3, Restrictions on Group Priority Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 60.31 Example of Group Priority Operation in SAR Mode – Single Scan Mode (2 Groups)

60.3.19.2 Group Priority Operation in SAR Mode – Continuous Scan Mode

In group priority operation in SAR mode – Continuous scan mode, the continuous scanning operation is performed for the scan group (continuous-scan group) in which scanning operation was first started. If an A/D conversion start trigger of another scan group (interrupt-scan group) is input during the continuous scanning operation of the continuous-scan group, the following operations are performed according to the priority of the interrupt-scan group.

1. Priority: When Interrupt-scan group (high) > Continuous-scan group (low)

- The scanning operation of the continuous-scan group is interrupted, and the single scanning operation of the interrupt-scan group is started.
- The continuous scanning operation of the continuous-scan group is restarted after the single scanning operation of the interrupt-scan group is completed.

2. Priority: When Continuous-scan group (high) > Interrupt-scan group (low)

- This operation is prohibited. Operation is not guaranteed.

In case 1. above, if an A/D conversion start trigger for another scan group (interrupt-scan group B) is input during the single scanning operation of the interrupt-scan group (interrupt-scan group A), the following operation is performed according to the priority of interrupt-scan group B.

3. Priority: When Interrupt-scan group B (high) > Interrupt-scan group A (middle) > Continuous-scan group (low)

- The single scanning operation of interrupt-scan group A is suspended, and the single scanning operation of interrupt-scan group B is started.
- After the single scanning operation of interrupt-scan group B is completed, the single scanning operation of interrupt-scan group A is resumed.
- After the scanning operation of interrupt-scan group A is completed, the continuous scanning operation of the continuous-scan group is resumed.

4. Priority: When interrupt-scan group A (high) > interrupt-scan group B (medium) > continuous-scan group (low)

- After the single scanning operation of interrupt-scan group A is completed, the single scanning operation of interrupt-scan group B is started.
- After the single scanning operation of interrupt-scan group B is completed, the continuous scanning operation of the continuous-scan group is resumed.

5. Priority: When interrupt-scan group A (high) > continuous-scan group (medium) > interrupt-scan group B (low)

- This operation is prohibited. Operation is not guaranteed.

Example operations are shown in the followings:

(1) Example of two-group priority operation in SAR mode – Continuous scan mode

Table 60.45 and Figure 60.32 show the example of group priority operation in SAR mode – Continuous scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high-priority group) and analog channel 1 to 3 (AN001 to AN003) is assigned to scan group 1 (low-priority group).

Table 60.45 Example of the Group Priority Operation in SAR Mode – Continuous Scan Mode (2 Groups) (1/2)

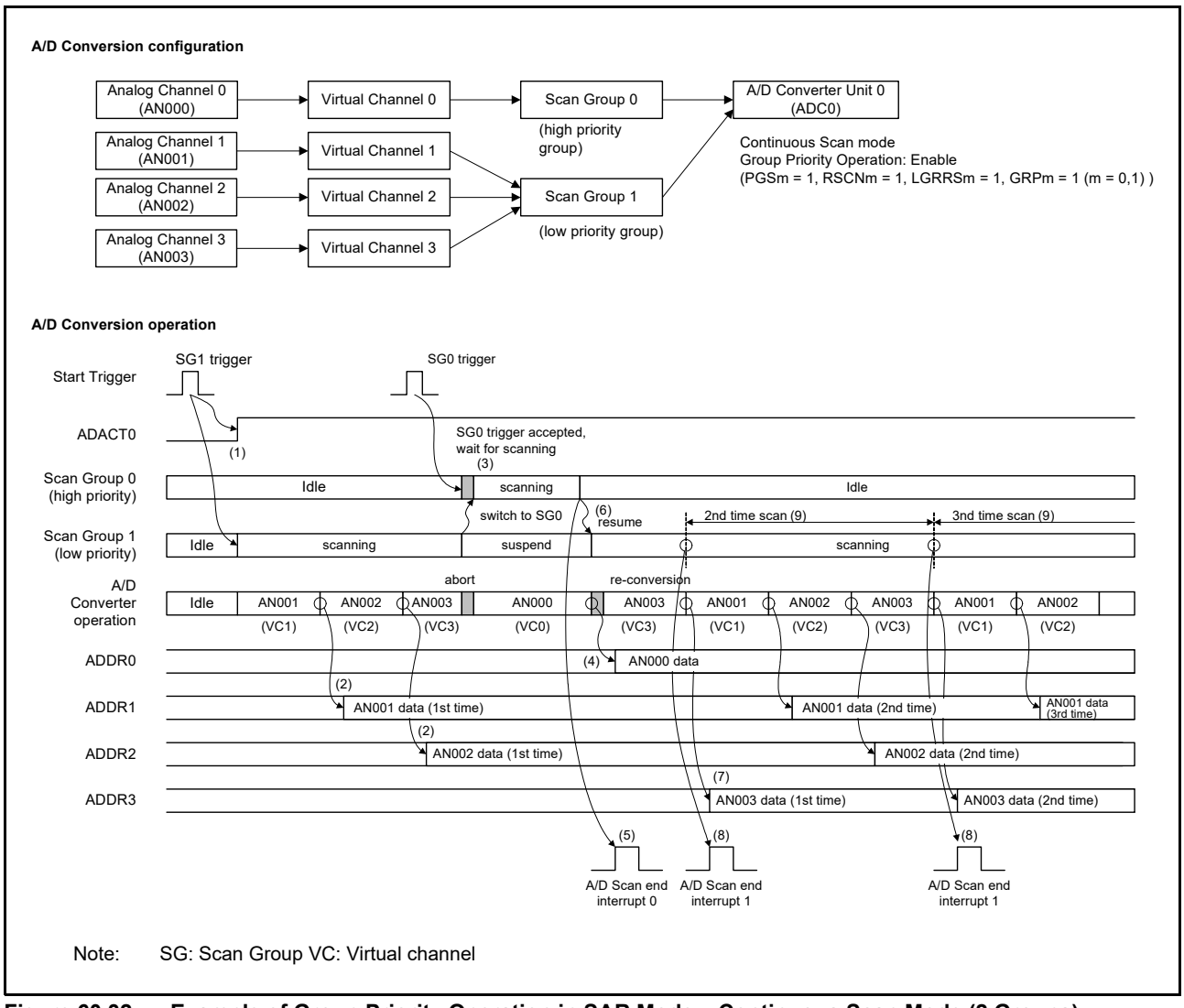
Step	Operation
1	The continuous scanning operation of scan group 1 (low priority group) is started by the trigger input of scan group 1.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and scanning operation of scan group 0 is started.*1
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel.*1

Table 60.45 Example of the Group Priority Operation in SAR Mode – Continuous Scan Mode (2 Groups) (2/2)

Step	Operation
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	Scan group 1 continues the continuous scanning operation.*2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 60.3.19.3, Restrictions on Group Priority Operation.

Note 2. To stop the continuous scanning operation, follow the procedures described in section 60.5.4, Force Stops the A/D Conversion Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 60.32 Example of Group Priority Operation in SAR Mode – Continuous Scan Mode (2 Groups)

(2) Example of three-groups priority operation in SAR mode – Continuous scan mode

Table 60.46 and Figure 60.33 show the example of group priority operation when analog channel 0 (AN000) is assigned to scan group 0 (high priority), analog channels 1 and 2 (AN001 and AN002) are assigned to scan group 1 (middle priority), and analog channel 3 to 5 (AN003 to AN005) are assigned to scan group 2 (low priority).

Table 60.46 Example of the Group Priority Operation in SAR Mode – Continuous Scan Mode (3 Groups)

Step	Operation
1	The continuous scanning operation of scan group 2 is started by the trigger input of scan group 2.
2	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 1 is input during A/D conversion on scan group 2, the scanning operation of scan group 2 is suspended and the scanning operation of scan group 1 is started.*1
4	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started.*1
6	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
7	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
8	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is restarted from the A/D conversion incomplete channel.*1
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
11	Scan group 2 is resumed after scanning of scan group 1 is completed. The scanning operation is resumed from the A/D conversion incomplete channel.*1
12	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
13	If a scan end interrupt for scan group 2 is enabled, a scan end interrupt occurs.
14	Scan group 1 continues the continuous scanning operation.*2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 60.3.19.3, Restrictions on Group Priority Operation.

Note 2. To stop the continuous scanning operation, follow the procedures described in section 60.5.4, Force Stops the A/D Conversion Operation.

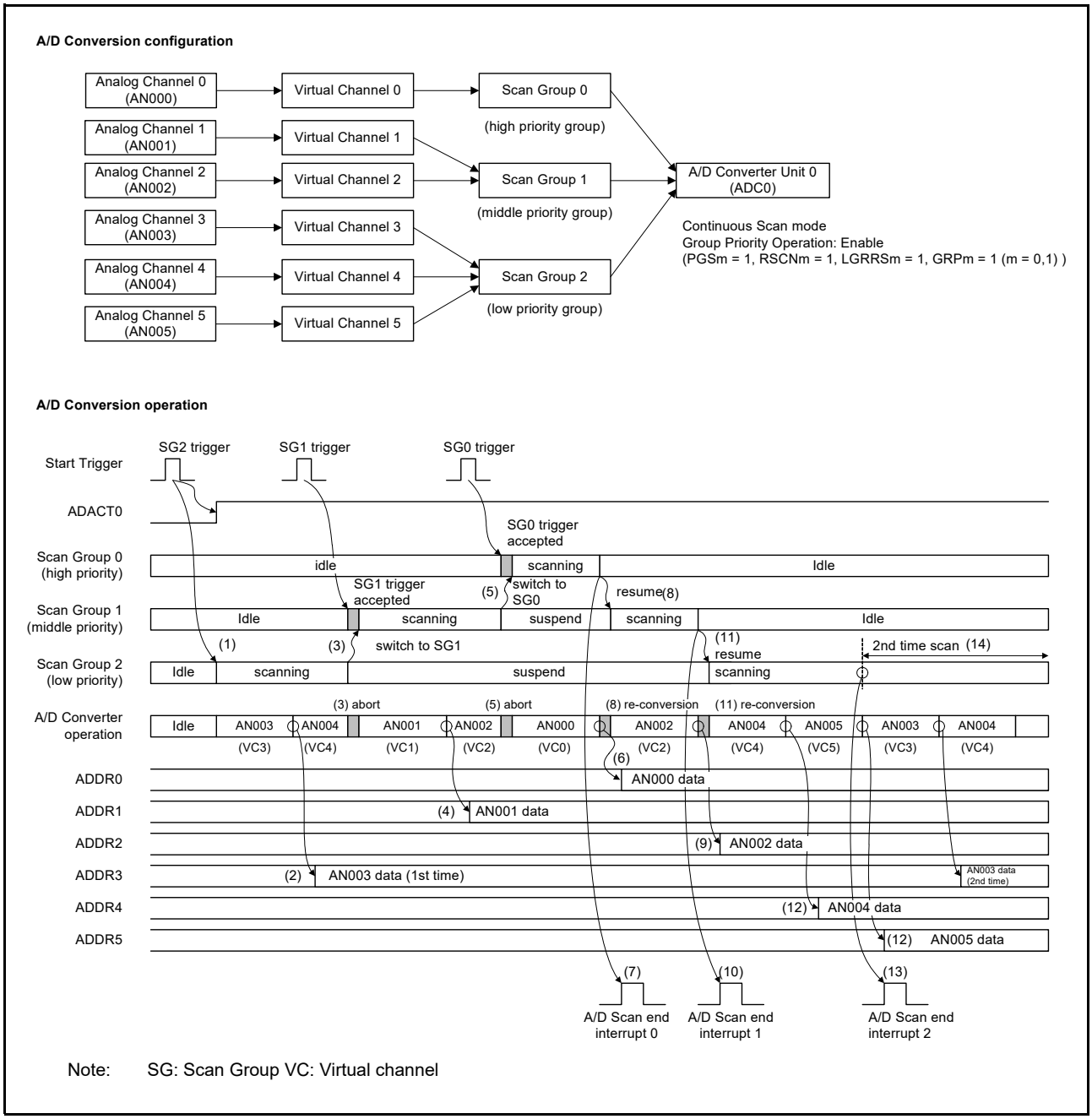


Figure 60.33 Example of Group Priority Operation in SAR Mode – Continuous Scan Mode (3 Groups)

60.3.19.3 Restrictions on Group Priority Operation

(1) Channel-dedicated sample-and-hold circuit restrictions

Only a scan group with the highest priority can use the channel-dedicated sample-and-hold circuits in group priority operation.

(2) Restrictions on the operation mode and operation setting of the A/D converter

Group priority operation is only available in SAR mode; Group priority operation is prohibited in oversampling mode or hybrid mode.

For details, see Table 60.43.

If these restrictions are violated, operation is not guaranteed.

(3) Restrictions on group priority operation in SAR mode – Continuous scan mode

In group priority operation in SAR mode – continuous scan mode, do not input A/D conversion start triggers for scan groups that have a lower priority than the scan group that started continuous scanning operation. If this restriction is violated, the operation is not guaranteed.

(4) Restrictions on trigger input for the same scan group

Input the A/D conversion start triggers for the same scan group in group priority operation for the following periods.

[When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)]

- 2 ADCLK + 2 PCLK cycles after the scan end flag is set*1

[When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)]

- Synchronous operation period cycle × 2 after the scan end flag is set*1

Note: m = 0 to 2

Note 1. ADSCANENDSR.SCENDFn = 1 (n = 0 to 11)

If A/D conversion start triggers are input without waiting for the above period, the A/D conversion start triggers may be lost. To accept A/D conversion start triggers reliably, input the trigger after the above period has elapsed.

(5) Restrictions on suspend and resume for scanning operation of low-priority groups

When an A/D conversion start trigger for the high priority group is input during scanning of the low priority group in group priority operation, the scanning operation of the low priority group is interrupted at the timing shown in the following:

[Scan interruption timing of low priority group]

- When synchronous operation is enabled
  - After the A/D conversion start trigger of the high-priority group is accepted, the scanning operation of the low-priority group is suspended after waiting until the next synchronous operation period timing.
- When synchronous operation is disabled
  - The scanning operation of the low-priority group is immediately suspended after the A/D conversion start trigger of the high-priority group is accepted.

If A/D conversion is being performed on channels for which A/D-converted value addition/averaging function is to be used at the timing of scan interruption of a low-priority group, the midway result of A/D-converted value addition/averaging value at that time is discarded. When the scanning operation of the low-priority group is resumed, A/D conversion starts over from the beginning of the number of times specified for A/D-converted value addition/averaging function.

If the scanning operation is suspended by group priority operation, ADGRSR.ACTGRn (n = 0 to 11) bit of the suspended scan group remains set to 1.

60.3.20 Synchronous Operation

Synchronous operation is the function that controls the operations of multiple A/D converters so that they are synchronized. The synchronization signal can be selected from between an internal signal of the ADC and the external signal from the GPT.

60.3.20.1 Synchronous Operation Examples

(1) Basic synchronous operation

The basic operation of synchronous operation is shown in the following. The configuration example is shown in Table 60.47, and the detail of operation is shown in Table 60.48 and Figure 60.34.

Table 60.47 Configuration Example of Basic Synchronous Operation

Item	Configuration
A/D converter	• ADC0: Set to SAR mode – single scan mode • ADC1: Set to SAR mode – single scan mode
Virtual channel	• Virtual channel 0 to 2: Assign AN000 to AN002 and set to single-ended input mode. • Virtual channel 8 to 10: Assign AN008 to AN010 and set to single-ended input mode.
Scan group	[Scan Group 0] • Assign AN000 to AN002 (virtual channel 0 to 2) • Set to convert with ADC0 [Scan Group 1] • Assign AN008 to AN010 (virtual channel 8 to 10) • Set to convert with ADC1
Synchronous operation setting	• Enable ADC0 synchronous operation • Enable ADC1 synchronous operation • Set the synchronous operation period cycles

Table 60.48 Example of Basic Synchronous Operation

Step	Detail Behavior of Example Operation
1	When a trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts.
3	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
5	During synchronous operation, the A/D converter starts the sampling operation in accordance with synchronous operation period.

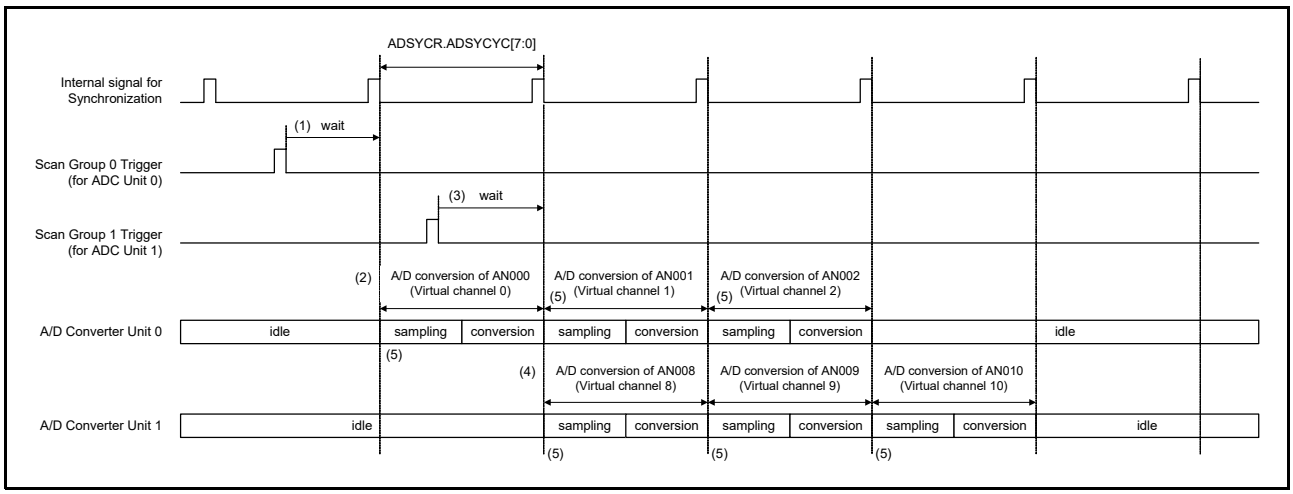


Figure 60.34 Example of Synchronous Operation in SAR Mode – Single Scan Mode

(2) Example of synchronous operation using an external signal (the synchronization signal from the GPT)

Table 60.49 Configuration Example

Item	Configuration
A/D converter	• ADC0: Set to SAR mode – single scan mode • ADC1: Set to SAR mode – single scan mode
Virtual channel	• Virtual channel 0 to 2: Assign AN000 to AN002 and set to single-ended input mode. • Virtual channel 8 to 10: Assign AN008 to AN010 and set to single-ended input mode.
Scan group	[Scan Group 0] • Assign AN000 to AN002 (virtual channel 0 to 2) • Set to convert with ADC0 [Scan Group 1] • Assign AN008 to AN010 (virtual channel 8 to 10) • Set to convert with ADC1
Synchronous operation setting	• Enable ADC0 synchronous operation • Enable ADC1 synchronous operation • Set the synchronous operation period cycles • Enable synchronization with an external signal (the synchronization signal from the GPT)

Table 60.50 Example of Basic Synchronous Operation

Step	Detail Behavior of Example Operation
1	When a GPT trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts.
3	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
5	During synchronous operation, the A/D converter starts the sampling operation in accordance with synchronous operation period.

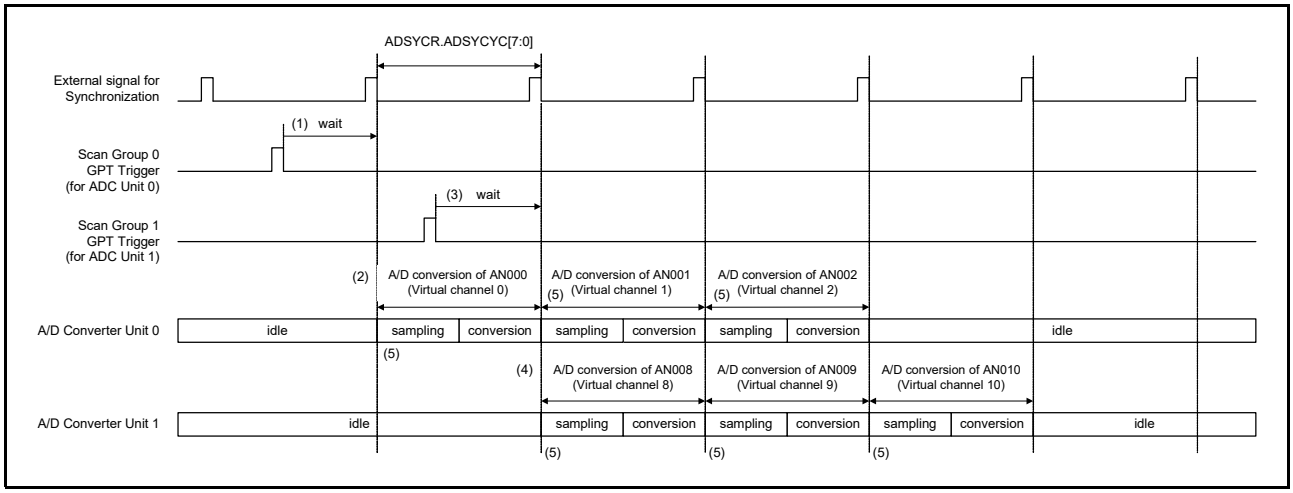


Figure 60.35 Example of Synchronous Operation Using an External Signal (the Synchronization Signal from the GPT)

60.3.20.2 Restrictions on Synchronous Operation

When synchronous operation is used, observe the following restrictions. If the following restrictions are violated, operations are not guaranteed.

(1) Basic restrictions

- Set the synchronous operation period to a value larger than successive approximation time of ADCm (m = 0 to 2).
  - ADSYCR.ADSYCYC[7:0] ≥ ADCNVSTR.CSTm[5:0] + 1 (m = 0 to 2)
- Set so that the sum of the sampling times of the analog channels and successive approximation time of ADCm (m = 0 to 2) is an integral multiple of the synchronous operation period.
  - ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0] = ADSYCR.ADSYCYC[7:0] × i
    (x = 0 to 7, y = 0 to 15, m = 0 to 2, i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
- Set the synchronous operation period to a value larger than successive approximation time at self-calibration.*1
  - ADSYCR.ADSYCYC[7:0] ≥ ADCALSTCR.CALADCST[5:0] + 1
- Set the sampling time and successive approximation time at self-calibration to be integral multiples of the synchronous operation period.*1
  - ADCALSTCR.CALADSST[9:0] + ADCALSTCR.CALADCST[5:0] = ADSYCR.ADSYCYC[7:0] × i (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))

(2) Restrictions when using channel-dedicated sample-and-hold circuit

- Set the synchronous operation period to a value greater than the hold mode switching time of channel-dedicated sample-and-hold circuit.
  - ADSYCR.ADSYCYC[7:0] ≥ ADSHSTR0.SHHST[2:0] + 1
- Set the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time to be an integral multiple of the synchronous operation period.
  - ADSHSTR0.SHSST[7:0] + ADSHSTR0.SHHST[2:0] = ADSYCR.ADSYCYC[7:0] × i (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
- Set the hold mode switching time of channel-dedicated sample-and-hold circuit so that it is the same as successive approximation time of ADC0.
  - ADSHSTR0.SHHST[2:0] = ADCNVSTR.CSTm[5:0]
- Set the synchronous operation period to a value greater than the hold mode switching time in channel-dedicated sample-and-hold circuit's self-calibration operation.*1
  - ADSYCR.ADSYCYC[7:0] ≥ ADCALSHCR.CALSHHST[2:0] + 1
- Set so that the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time in self-calibration operation is an integral multiple of the synchronous operation period.*1
  - ADCALSHCR.CALSHSST[7:0] + ADCALSHCR.CALSHHST[2:0] = ADSYCR.ADSYCYC[7:0] × i (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))

(3) Restrictions when using disconnection detection assist function

- Set Disconnection Detection Assist period so that it is the same as the synchronous operation period.
  - ADSGDCRn.ADNDIS[7:0] = ADSYCR.ADSYCYC[7:0] (n = 0 to 11)

(4) Restrictions when using an external signal (the synchronization signal from the GPT)

- Select GPTCLK as the clock source for ADCLK.
- Release the GPT module-stop state using the Module Stop Register E (MSTPCRE) before enabling using an external signal (the synchronization signal from the GPT) and selecting the clock source for ADCLK.

(5) Restriction when using high accuracy mode

- The synchronous operation period should be set as follows.
  - ADSYCR.ADSYCYC[7:0] × i = ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0]
    (x = 0 to 7, y = 0 to 15, m = 0 to 2, i: 1, 2 × n (n = 1, 2, 3…))

Note 1. In relation to this restriction, the restrictions on the settings of self-calibration should be satisfied at the same time.
For restrictions on the setting of self-calibration, see section 60.3.9.3, Restrictions on Self-Calibration.

60.3.21 A/D Conversion Operation Monitor Function

The timing of A/D conversion operations can be monitored by observing the internal operation signals output from pins of the A/D converter. When this function is to be used, see section 22, I/O Ports to specify the signals to be monitored. The following signals can be monitored.

- ADSYNC

ADSYNC is the synchronization signal for A/D conversion operations. This signal indicates the synchronous timing of the synchronization function for A/D conversion. When this function is enabled, an A/D conversion operation is executed in synchronization with this signal. When this function is disabled, this signal is generated but ignored in A/D conversion operations. See section 60.3.20, Synchronous Operation for more details about synchronous operation of A/D conversion.

- ADCNVn (n = 0 to 2)

ADCNVn indicates the period of the A/D conversion of A/D converter units 0, 1, and 2. This signal is asserted during A/D conversion.

- ADSTn (n = 0 to 2)

ADSTn indicates the period of A/D scanning of A/D converter units 0, 1, and 2. This signal is asserted during A/D scanning.

Figure 60.36 shows the operation timing of each signal.

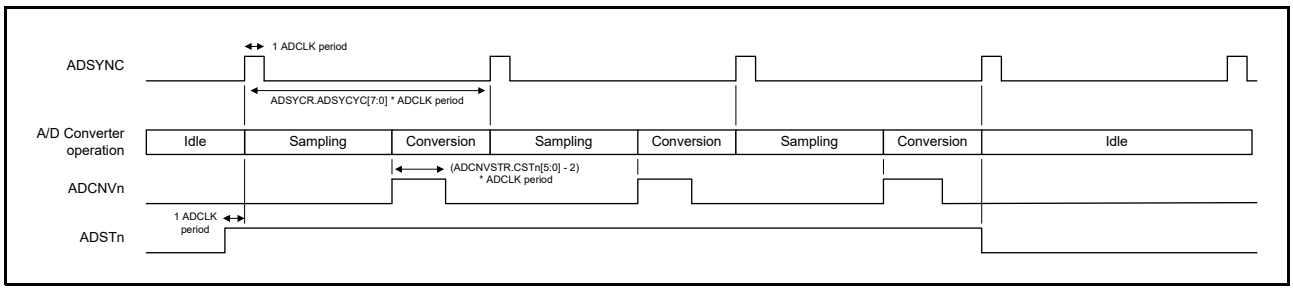


Figure 60.36 Monitor Signal Waveforms

60.4 A/D Conversion Data

This section describes the internal processing for A/D conversion data. For details about the operation of A/D conversion, see section 60.3, Operation.

60.4.1 Internal Data Processing Flow

Figure 60.37 shows the internal processing-flow of A/D conversion data.
The A/D conversion data output from the A/D converter is temporarily stored in the data buffer. And then, various data processing is performed based on the register setting. If the function that corresponds to each data processing is not used, its data processing is skipped.

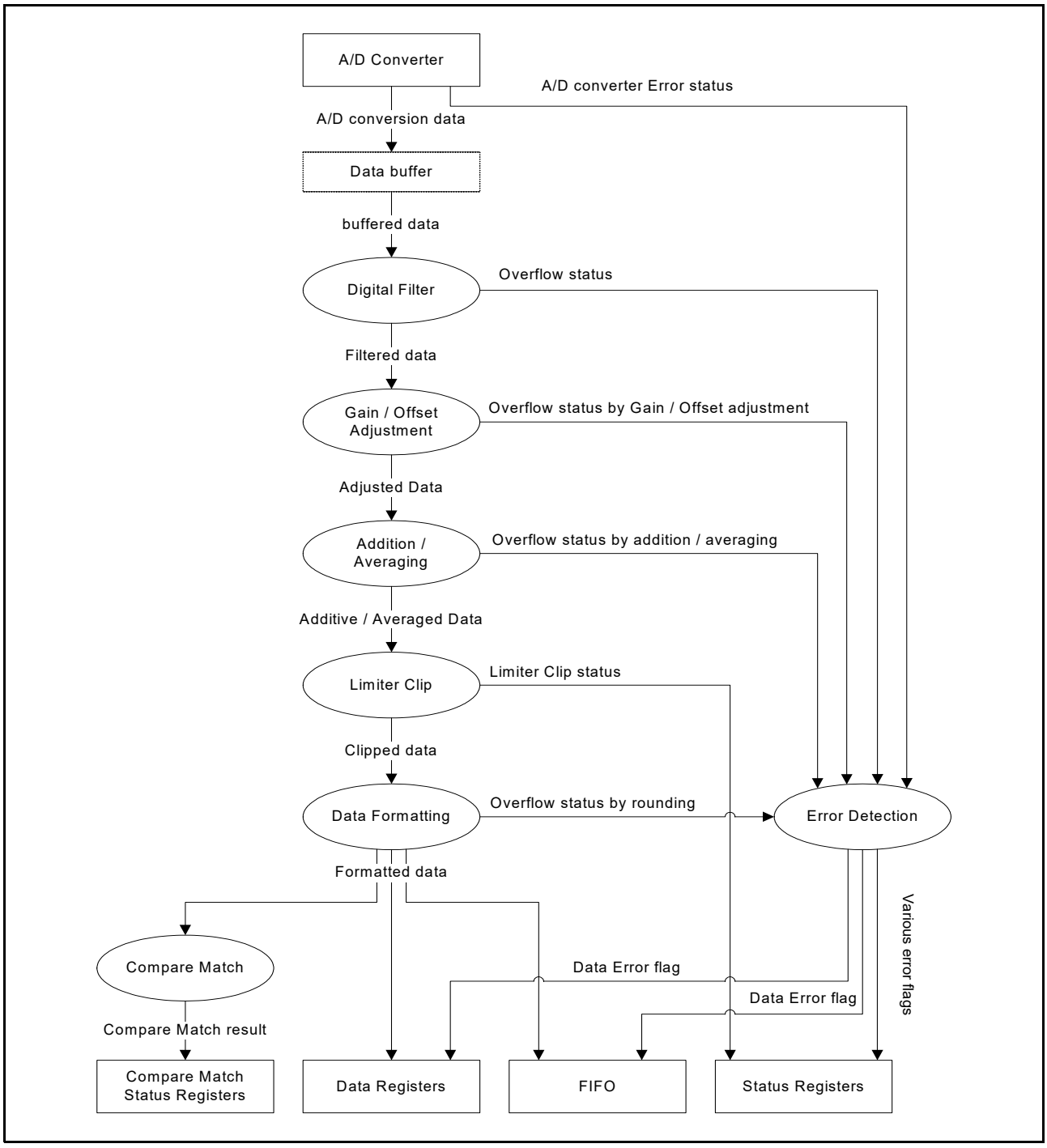


Figure 60.37 Internal Data Flow Diagram of A/D Conversion Data

60.4.2 Digital Filter Function

ADC has 4 digital filters per 1 unit of A/D converter. High-precision A/D conversion results can be obtained by using a digital filter. The digital filter function must be used in oversampling mode or hybrid mode. The digital filter function in SAR mode is prohibited.

60.4.2.1 Configuration and Characteristics

The digital filter consists of a FIR type filter with 22 taps. The digital filter is set by ADDOPCRAn.DFSEL[2:0] (n = 0 to 49) and ADDFSRm (m = 0 to 2) registers. The configuration of the digital filter is shown in Figure 60.38.
The type of the digital filter (FIR filter) can be selected from the following. For the characteristics of the digital filters, see section 67, Electrical Characteristics.
[List of digital filter characteristics]

- Sinc filter

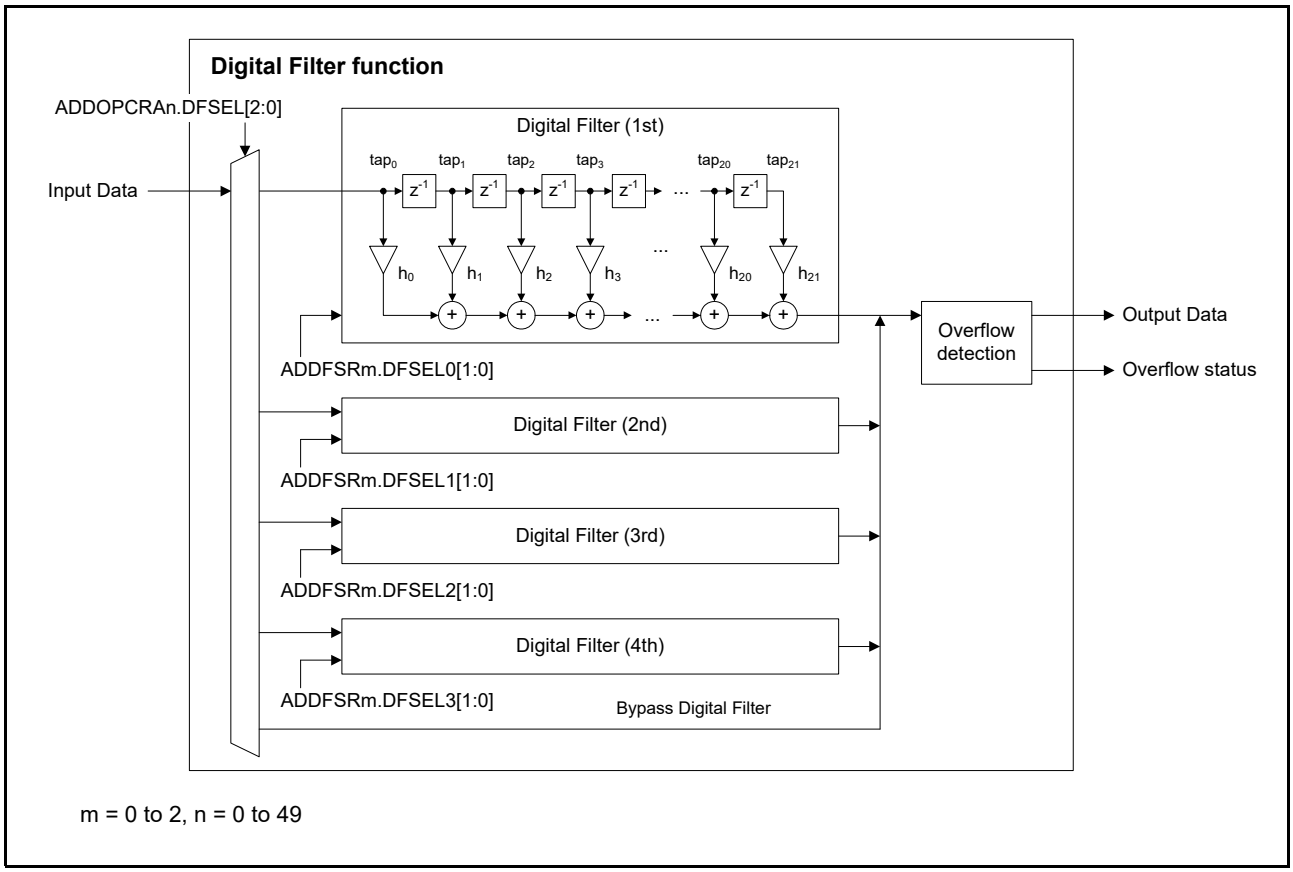


Figure 60.38 Block Diagram of Digital Filter

60.4.2.2 Operation of Digital Filter Function

When the digital filter function is enabled, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.
The operation of the digital filters in oversampling mode and hybrid mode is shown in the followings.

(1) Digital filter operation in oversampling mode

In oversampling mode, oversampling (A/D conversion) is performed continuously for one analog channel. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.
In oversampling mode, the data of the taps in the digital filter is discarded when the A/D conversion data is output to the next data processing. However, when A/D-converted value addition/averaging function is used, the data of taps in the digital filter is kept until the data required to calculate the addition value or average value of the A/D converted value is collected. When the A/D-converted value addition/averaging value is calculated, the tap data in the digital filter is discarded.

(2) Digital filter operation in hybrid mode

Hybrid mode can process oversampled data (A/D converted data) of up to four analog channels in parallel by using several digital filter circuits at the same time. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.
In hybrid mode – single scan mode, taps in the digital filter are discarded at the end of scanning operation.
In hybrid mode – continuous scan mode, the data of the taps in the digital filter continues to be updated as long as the continuous scanning operation is continued. Therefore, after the data is filled in all taps in the digital filter, a new calculation result is output for each oversampling. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.
The operation of the digital filter in hybrid mode – background continuous scan mode is the same as in hybrid mode – continuous scan mode. During the background continuous scanning operation, the digital filters and other data processing continue to take place in the background. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.

60.4.3 Calibration and Adjustment

In this process, the following processing is performed for the results output from the A/D converter or for the data output from Digital Filter.

1. Calibrate Gain Error and Offset Error
2. Adjust User's Gain
3. Adjust User's Offset

For details on each process, refer to each item.

60.4.3.1 Gain Error and Offset Error Calibration

This process calibrates the A/D converter's Gain Error and Offset Error due to by chip-by-chip characteristic variations. Calibration for the A/D converter's Gain Error and Offset Error is performed by calculation based on error data measured inside the ADC by self-calibration operation.

Using gain error and offset error calibration when the input to the A/D converter is near the VREFH0 upper limit or VREFL0 lower limit, an overflow of A/D conversion data may occur.

60.4.3.2 User's Gain Adjustment

User's Gain adjusting function multiplies the A/D conversion data by an arbitrary coefficient value.

The coefficient value of User's Gain is set in ADUGTRn.UGAIN[15:0] (n = 0 to 7). User's Gain can be set for each virtual channel. The User's Gain Table to be used is selected in ADDOPCRAm.GAINSEL[3:0] (m = 0 to 49). The coefficient value of User's Gain is the sum of the gain corresponding to the bits that are set to 1 in ADUGTRn.UGAIN[15:0]. Table 60.51 shows the gain factors corresponding to the respective bit in ADUGTRn.UGAIN[15:0]. An example of setting User's Gain is shown in Table 60.52. Figure 60.39 and Figure 60.40 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Gain adjustment function is used.

The overflow of A/D conversion data may occur when User's Gain adjusting function is used.

Table 60.51 List of Gains Corresponding to Each Gain Setting Bit in User's Gain Setting Table Register

UGAIN[15:8]	Gain Value	UGAIN[7:0]	Gain Value
b15	2^{1} = 2.0	b7	2^{-7} = 7.813E–03
b14	2^{0} = 1.0	b6	2^{-8} = 3.906E–03
b13	2^{-1} = 0.5	b5	2^{-9} = 1.953E–03
b12	2^{-2} = 0.25	b4	2^{-10} = 9.766E–04
b11	2^{-3} = 0.125	b3	2^{-11} = 4.883E–04
b10	2^{-4} = 0.0625	b2	2^{-12} = 2.441E–04
b9	2^{-5} = 0.03125	b1	2^{-13} = 1.221E–04
b8	2^{-6} = 1.563E-02	b0	2^{-14} = 6.104E–04

Table 60.52 Example of User's Gain Settings (1/2)

ADUGTRn.UGAIN[15:0] (n = 0 to 7)	Gain Value
0000h	x0.0000
:	:
0400h	x0.0625
:	:
0800h	x0.1250
:	:
1000h	x0.2500
:	:

Table 60.52 Example of User's Gain Settings (2/2)

ADUGTRn.UGAIN[15:0] (n = 0 to 7)	Gain Value
2000h	x0.5000
:	:
4000h (initial value)	x1.0000
:	:
8000h	x2.0000
:	:
FFFFh	x3.9999

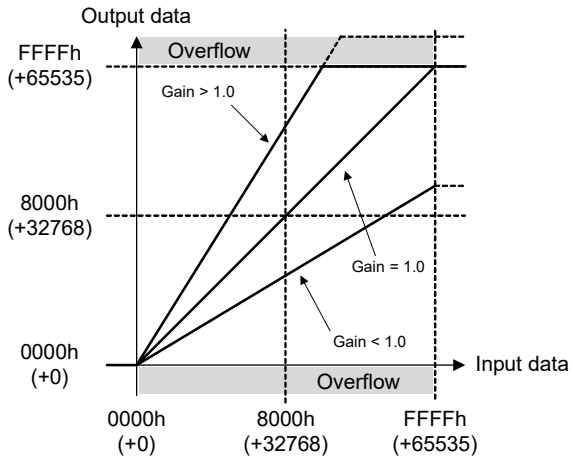


Figure 60.39 User's Gain Adjustment (16-Bit Data Length Format and Unsigned Data Format)

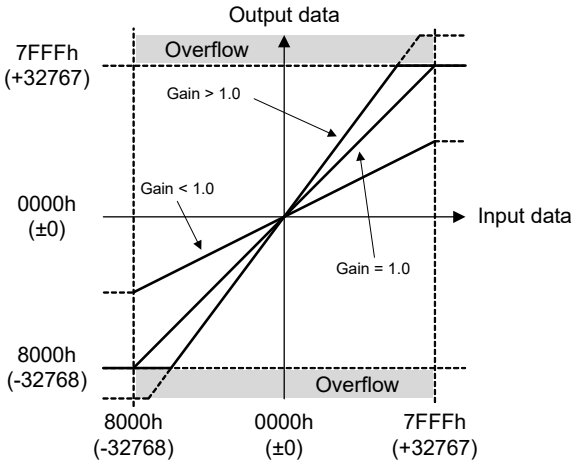


Figure 60.40 User's Gain Adjustment (16-Bit Data Length Format and Signed Data Format)

60.4.3.3 User's Offset Adjustment

User's Offset adjustment function adds or subtracts any constant value to or from the A/D conversion data.
The constant value of User's Offset is set in ADUOFTRn.UOFSET[15:0] (n = 0 to 7). User's Offset can be set for each virtual channel. User's Offset Table to be used is selected in ADDOPCRAm.OFSETSEL[3:0] (m = 0 to 49). Table 60.53 shows the relationship between the register setting value of User's Offset and the offset value. Figure 60.41 and Figure 60.42 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Offset adjustment function is used.
When User's Offset adjusting function is used, overflow of A/D conversion data may occur.

Table 60.53 Relationship between User's Offset Register Value and Offset Value

ADUOFTRn.UOFSET[15:0] (n = 0 to 7)	Offset Value (16-bit Data Length Format)
7FFFh	+32767
7FFEh	+32766
7FFDh	+32765
:	:
0003h	+3
0002h	+2
0001h	+1
0000h (initial value)	0
FFFFh	−1
FFFEh	−2
FFFDh	−3
:	:
8002h	−32766
8001h	−32767
8000h	−32768

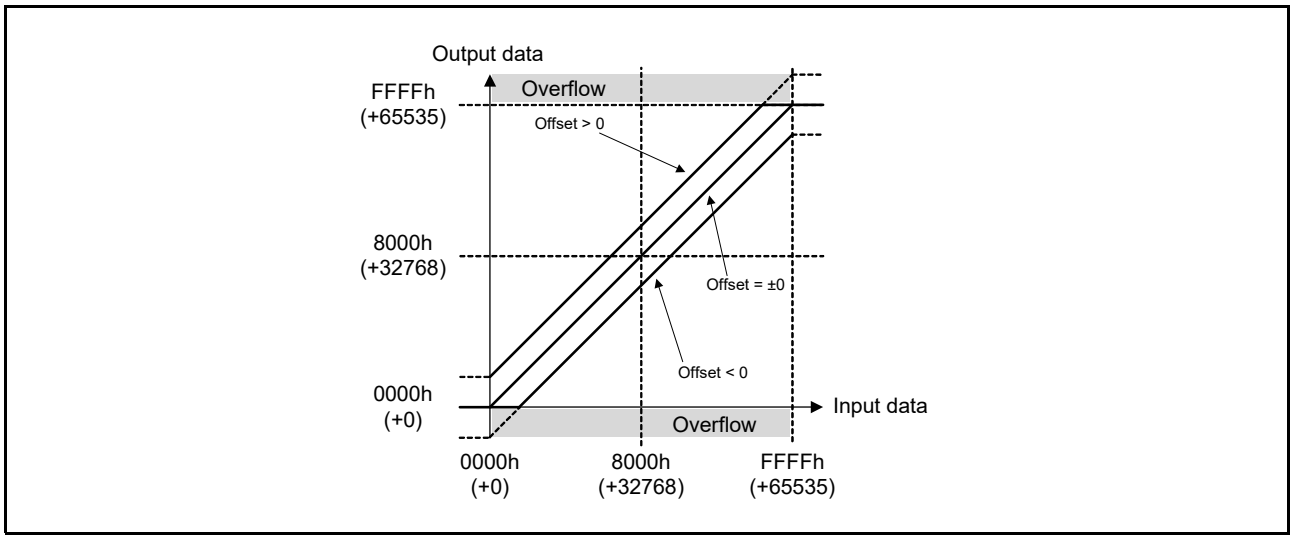


Figure 60.41 User's Offset Adjustment (16-Bit Data Length Format and Unsigned Data Format

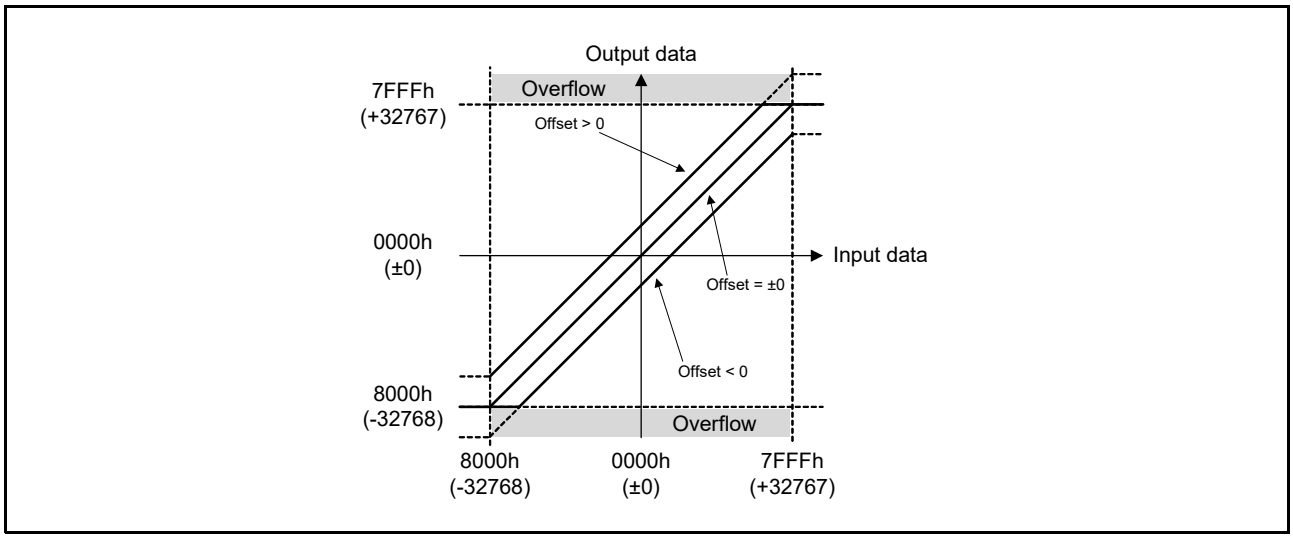


Figure 60.42 User's Offset Adjustment (16-Bit Data Length Format and Signed Data Format)

60.4.4 A/D-Converted Value Addition/Averaging Function

A/D-converted value addition/averaging function is a function to calculate the sum value or average value of the result of A/D conversion of analog channels continuously for the specified number of times. A/D-converted value addition/averaging function can be configured for each virtual channel in ADDOPCRBn.AVEMD[1:0] bits and ADDOPCRBn.ADC[3:0] bits (n = 0 to 49).

A/D-converted value addition/averaging function inputs the output of the previous data processing. When the specified number of times of data is input, the added value or average value is calculated, and the result is output to the next data processing. When A/D-converted value addition/averaging function is used, the A/D converter repeats the A/D conversion until the data is collected for the number of times specified by the ADDOPCRBn.ADC[3:0] (n = 0 to 49) bits. One calculation result (total value or average value) is output every time the specified number of times of data is input to A/D-converted value addition/averaging. Figure 60.43 shows the operation of A/D-converted value addition/averaging function in SAR mode or oversampling mode. Figure 60.44 shows the operation of A/D-converted value addition/averaging function in hybrid mode.

When the A/D-converted value addition/averaging function is used, overflow of the A/D-converted data may occur. However, A/D conversion overflow may not be detected under certain conditions. For more information on A/D conversion overflow, see section 60.6.2, A/D Conversion Overflow.

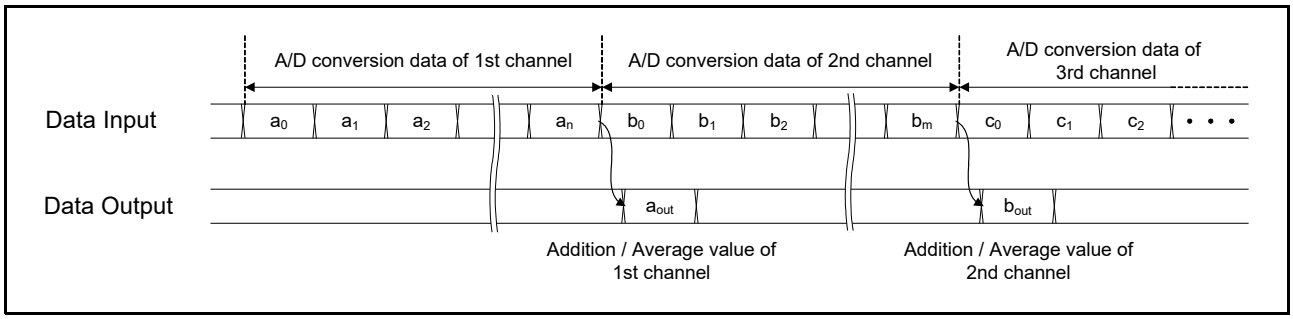


Figure 60.43 Example Operation of A/D-Converted Value Addition/Averaging Function (SAR Mode or Oversampling Mode)

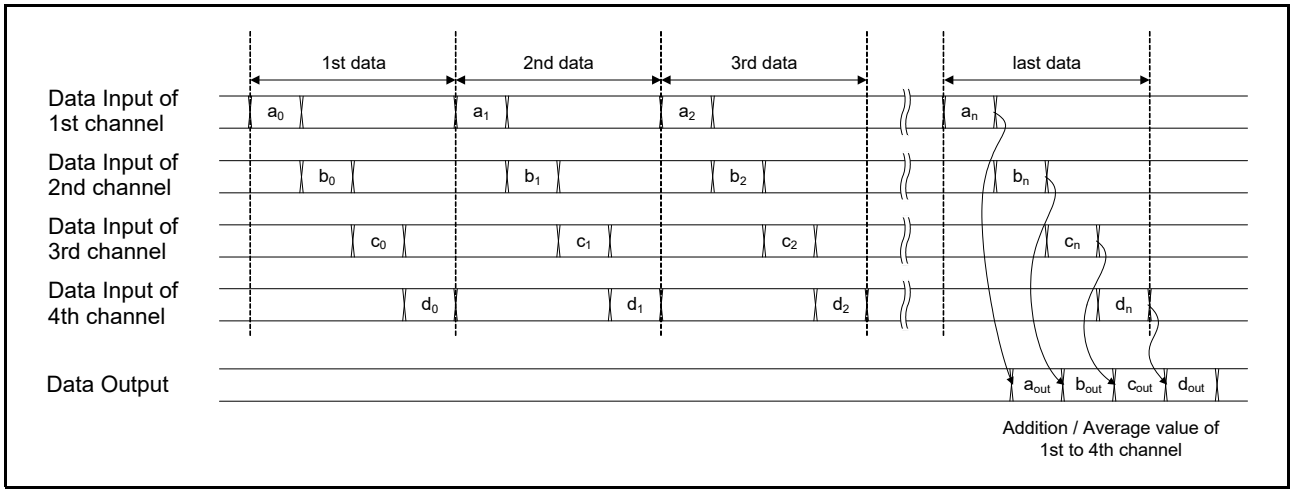


Figure 60.44 Example Operation of A/D-Converted Value Addition/Averaging Function (Hybrid Mode)

60.4.5 Limiter Clip Function

Limiter Clip Function is the function to set the upper and lower limits of A/D conversion data. When the A/D conversion data exceeds the specified upper limit value, it is clipped to the upper limit value. If the A/D conversion data falls below the specified lower limit, it is clipped to the lower limit value. Figure 60.45 shows an example of Limiter Clip Function operation.

The upper and lower limits of Limiter Clip Function can be specified in ADLIMTRn.LIMU[15:0] (n = 0 to 7) and ADLIMTRn.LIML[15:0] (n = 0 to 7). The Limiter Clip Function can be selected use or not (enable or disable) for each virtual channel in ADDOPCRCm.LIMTBLS[3:0] (m = 0 to 49).

Limiter Clip Function is processed by 16-bit length. When 14-/12-/10-bit is selected as the data length of the A/D conversion data (when ADDOPCRCm.ADPRC[1:0] = 01b, 10b, or 11b (m = 0 to 49)), the rounding to the specified data length is performed after the clipping by Limiter Clip Function. For details, see section 60.4.6, Data Formatting process.

The upper and lower limits of the limiter clip are treated as signed or unsigned depending on the setting of the sign selection bit (ADDOPCRCm.SIGNSEL (m = 0 to 49)) in the A/D conversion data.

For Limiter Clip Function, the upper limit value should be greater than the lower limit value (ADLIMTRn.LIMU[15:0] > ADLIMTRn.LIML[15:0] (n = 0 to 7)). If the upper limit value is less than or equal to the lower limit value, the A/D conversion data always becomes 0000h.

If a limiter clip occurs, the flags are set in the following status registers:

- ADLIMGRSR: The flag is set to the bit corresponding to the scan group where the limiter clip occurred.
- ADLIMCHSR0: If a limiter clip occurs during A/D conversion of an analog input channel, the flag is set to the corresponding bit.
- ADLIMCHSR1: If a limiter clip occurs during A/D conversion of an Extended Analog input channels, the flag is set to the corresponding bit.
- ADLIMEXSR: If a limiter clip occurs during A/D conversion of Extended Analog Function, the flag is set to the corresponding bit.

To clear the flags in the status registers, write 1 to the corresponding bit in ADLIMGRSCR, ADLIMCHSCR0, ADLIMCHSCR1, ADLIMEXSCR.

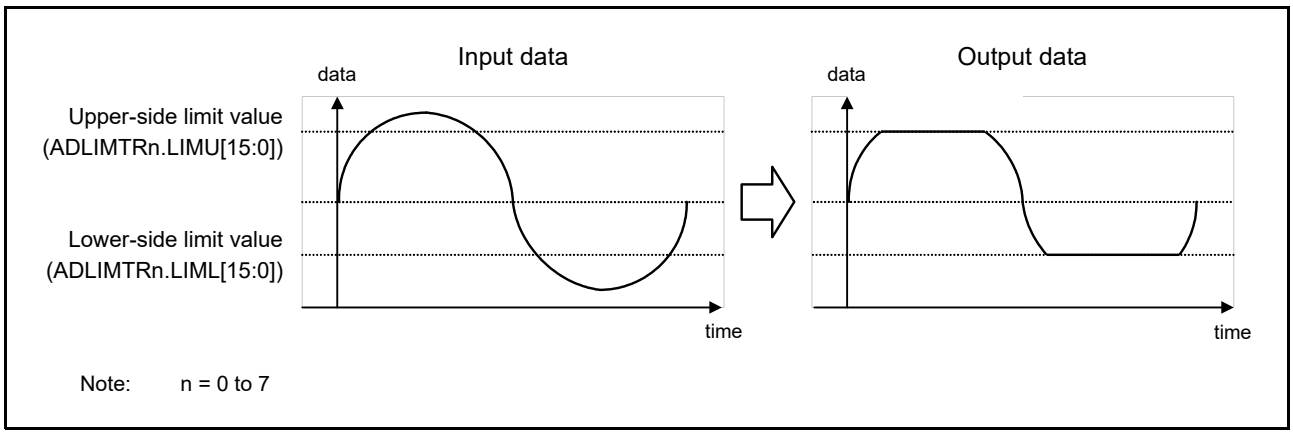


Figure 60.45 Example of Limiter Clip Function Operation

60.4.6 Data Formatting process

In this process, the following formatting is performed on the A/D conversion data.

- Signed/Unsigned data processing
- Data Rounding process

The data formatted in this process is stored to DATA[15:0] bits of the A/D data register (ADDRi (i = 0 to 35)), the extended A/D data register (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18), and the FIFO data register (ADFIFODRk (k = 0 to 11)) as the result of A/D conversion.
For details on the data format, see section 60.4.7, Data Format.

60.4.6.1 Signed/Unsigned Data Process

The A/D conversion data is formatted into signed or unsigned data format based on the setting of the ADDOPCRCn.SIGNSEL (n = 0 to 49) bits. Select the signed data format (SIGNSEL = 0) for differential input mode and the unsigned format (SIGNSEL = 1) for single-ended input mode.

60.4.6.2 Data Rounding Process

The A/D conversion data is processed as a fraction of the data length based on the setting of the ADDOPCRCn.ADPRC[1:0] (n = 0 to 49) bits. If 14-/12-/10-bit data format is selected (ADPRC[1:0] = 01b, 10b, 11b), the lower bits of the A/D conversion data will be rounding. (If the most significant bit of the digit to be cut off is 0, the digit is rounded down; if it is 1, the digit is rounded up.) If the 16-bit data format is selected (ADPRC[1:0] = 00b), the A/D conversion data is not rounded.

60.4.7 Data Format

This section describes the data format of the A/D conversion result stored in the DATA[15:0] bits of the A/D data register (ADDRi (i = 0 to 35)), extended A/D data register (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)), and FIFO data register (ADFIFODRk (k = 0 to 11)).

Note: The data format does not guarantee the resolution and accuracy of the A/D converter itself. See section 67, Electrical Characteristics of the A/D converter.

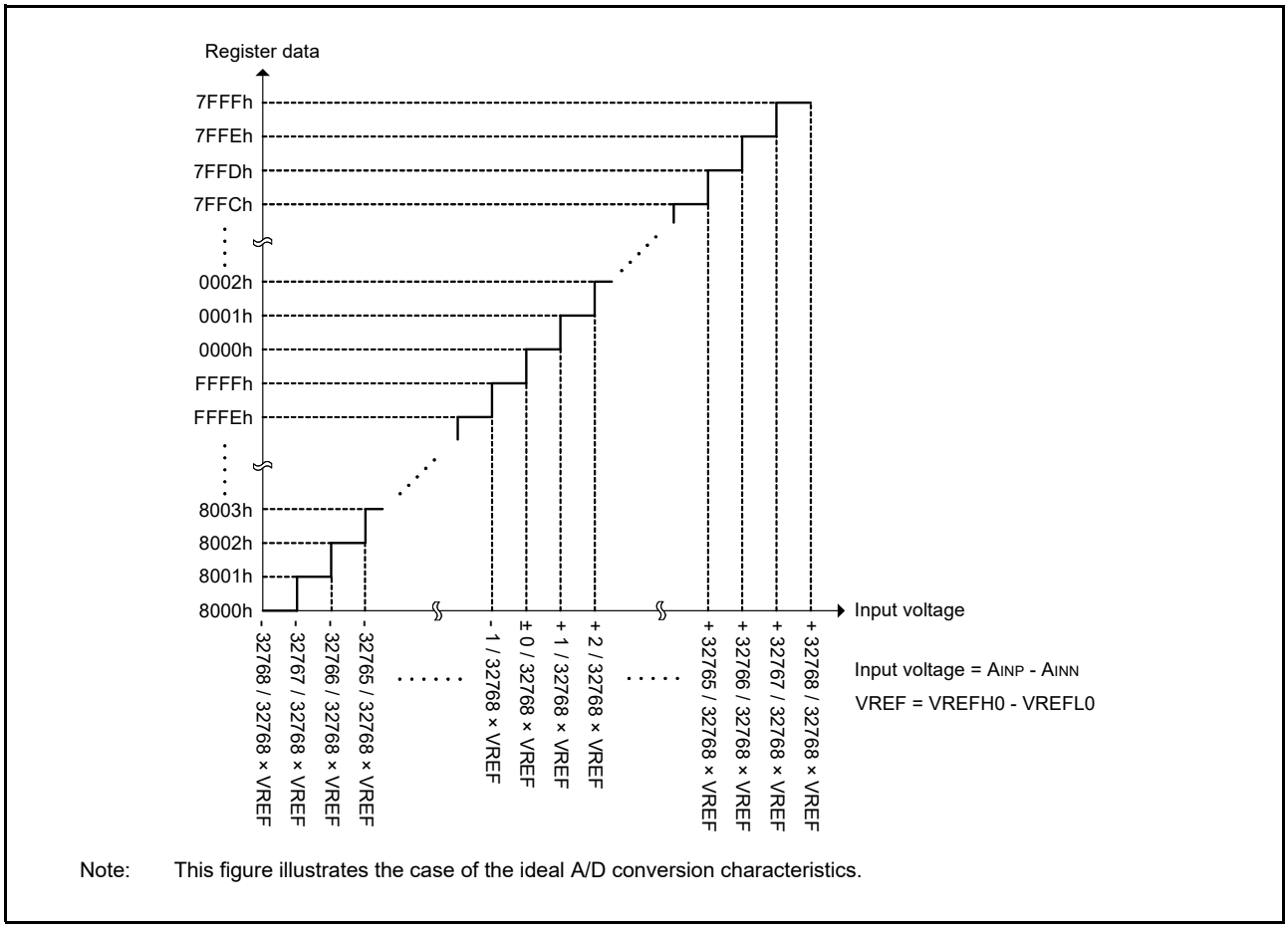
(1) 16-bits data length format

Table 60.54 shows the alignment of A/D conversion data on 16-bit data format. Figure 60.46 and Figure 60.47 show the data range for 16-bit data format.

For 16-bit data length signed data format, the differential input voltage is A/D-converted in the range of 8000h (-VREFH0) to 7FFFh (+VREFH0). In 16-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0000h (VREFL0) to FFFFh (VREFH0).

Table 60.54 Alignment of A/D Conversion Result Data (16-Bit Data Length)

Bit number:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit symbol:	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.46 Data Range of A/D Conversion Result (16-Bit Data Length, Signed Data, Differential Input)

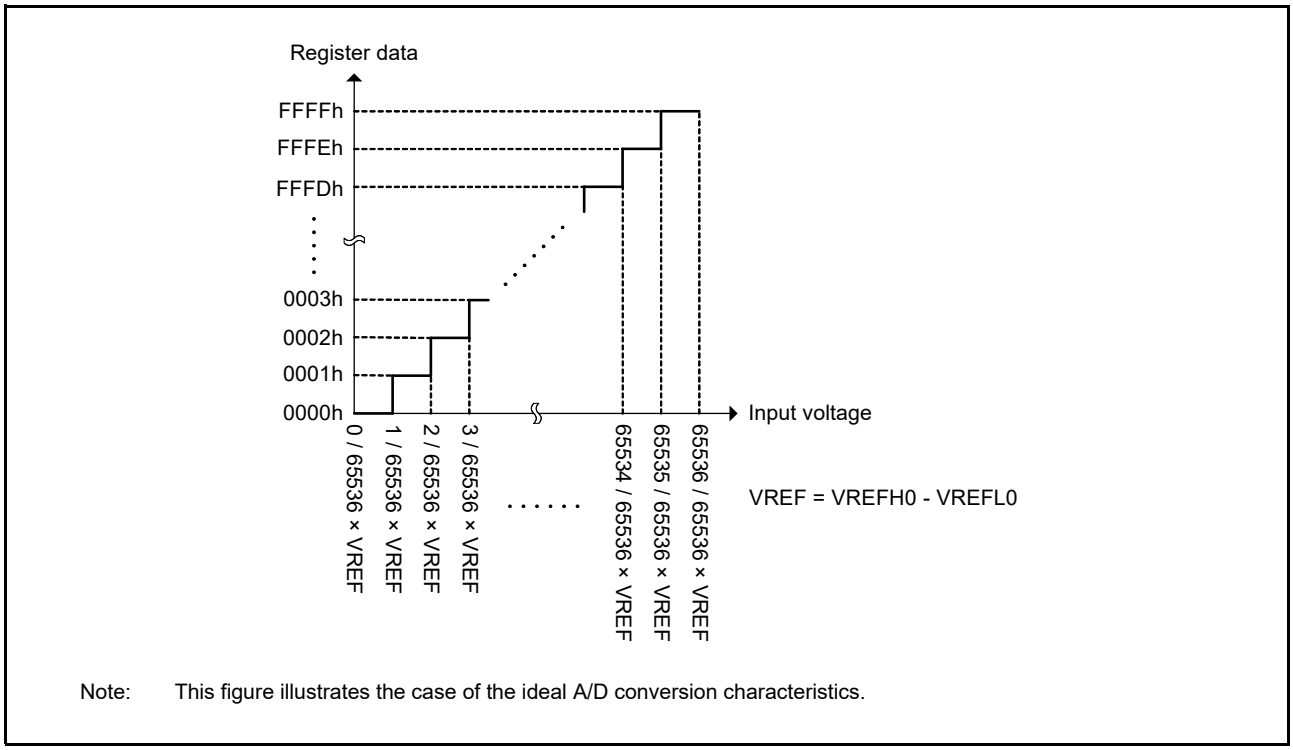


Figure 60.47 Data Range of A/D Conversion Result (16-Bit Data Length, Unsigned Data, Single-Ended Input)

(2) 14-bits data length format

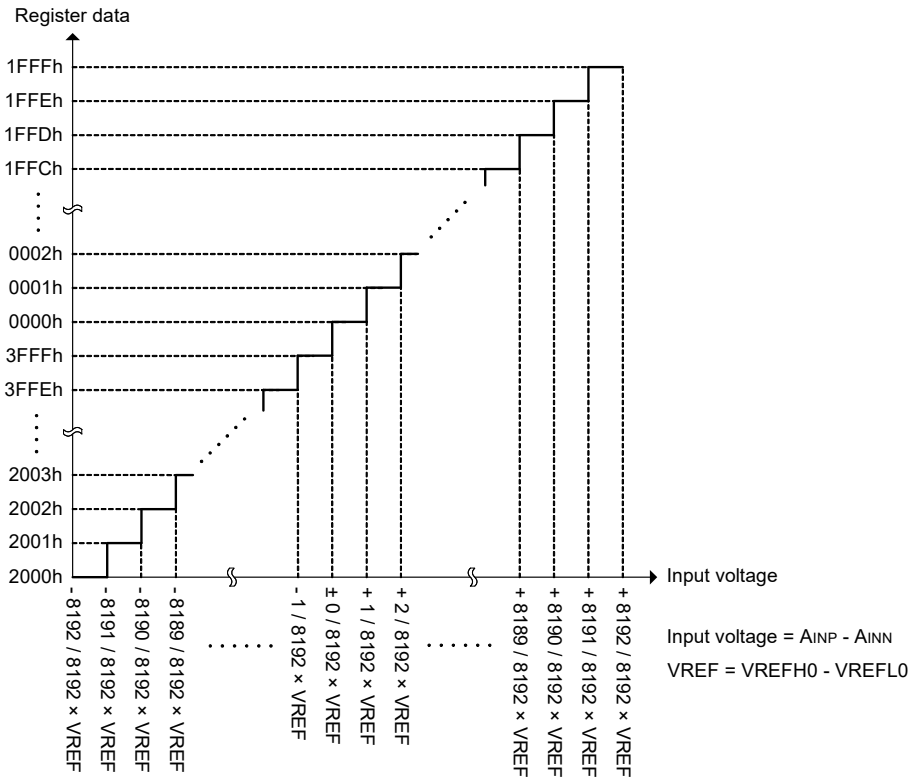
Table 60.55 shows the alignment of A/D conversion data on 14-bit data format. Figure 60.48 and Figure 60.49 show the data range for 14-bit data format.

For 14-bit data length signed data format, the differential input voltage is A/D-converted in the range of 2000h (-VREFH0) to 1FFFh (+VREFH0). In 14-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0000h (VREFL0) to 3FFFh (VREFH0).

For 14-bit data length, the upper 2 bits (bit15 to bit14) are always 0.

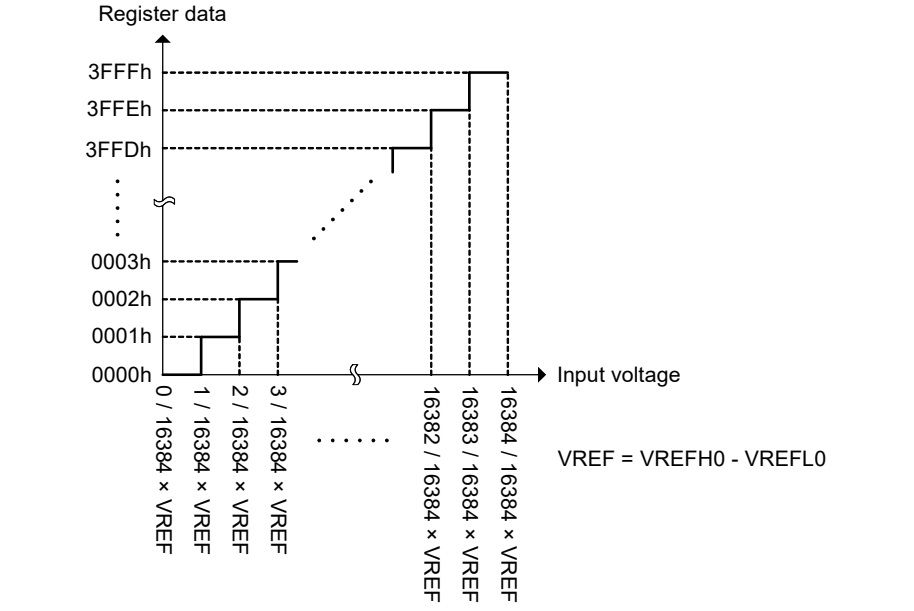
Table 60.55 Alignment of A/D Conversion Result Data (14-Bit Data Length)

Bit number:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit symbol:	0	0	DATA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.48 Data Range of A/D Conversion Result (14-Bit Data Length, Signed Data, Differential Input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.49 Data Range of A/D Conversion Result (14-Bit Data Length, Unsigned Data, Single-Ended Input)

(3) 12-bits data length format

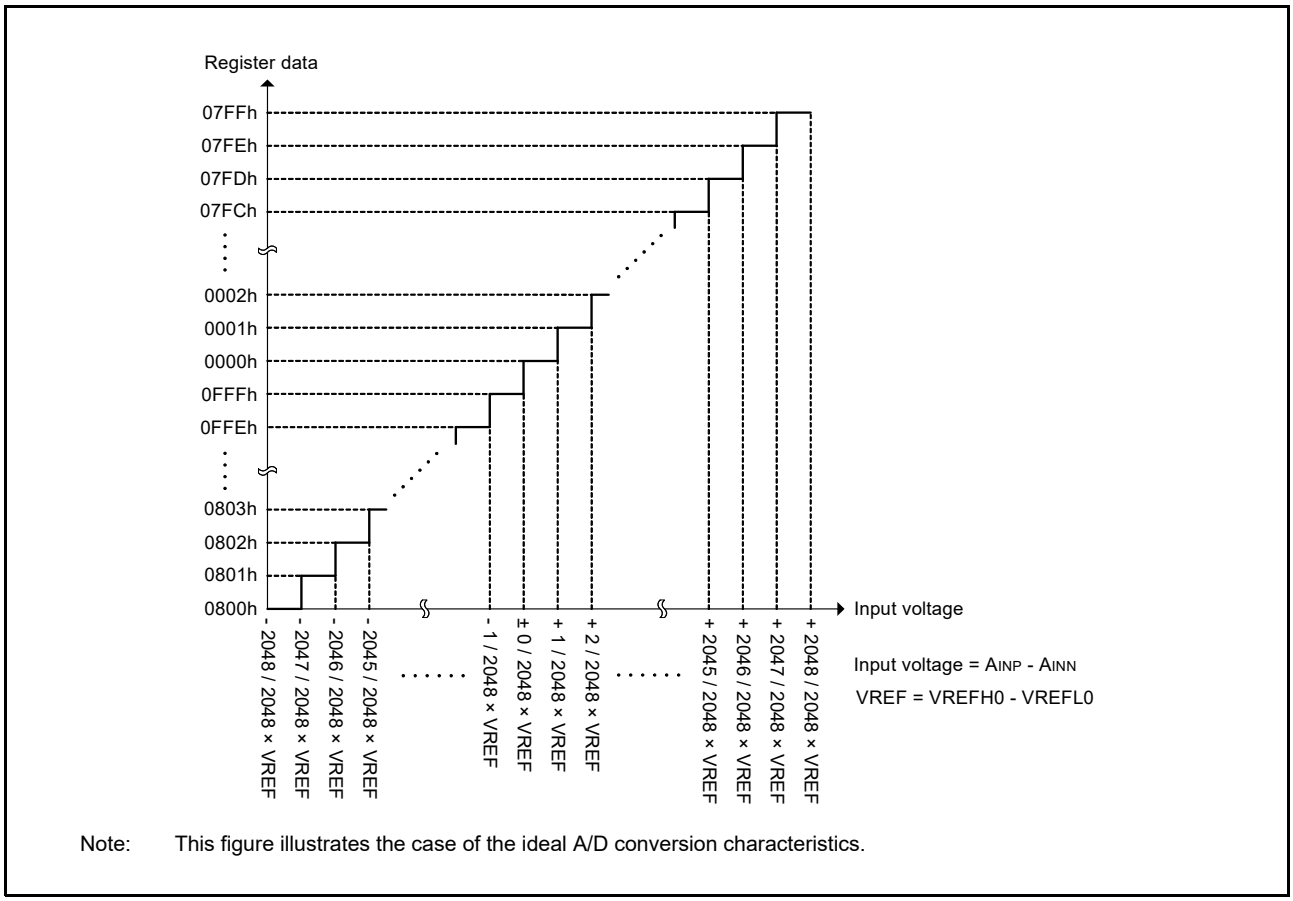
Table 60.56 shows the alignment of A/D conversion data on 12-bit data format. Figure 60.50 and Figure 60.51 show the data range for 12-bit data format.

For 12-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0800h (-VREFH0) to 07FFh (+VREFH0). In 12-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0000h (VREFL0) to 0FFFh (VREFH0).

For 12-bit data length, the upper 4 bits (bit15 to bit12) are always 0.

Table 60.56 Alignment of A/D Conversion Result Data (12-Bit Data Length)

Bit number:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit symbol:	0	0	0	0	DATA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.50 Data Range of A/D Conversion Result (12-Bit Data Length, Signed Data, Differential Input)

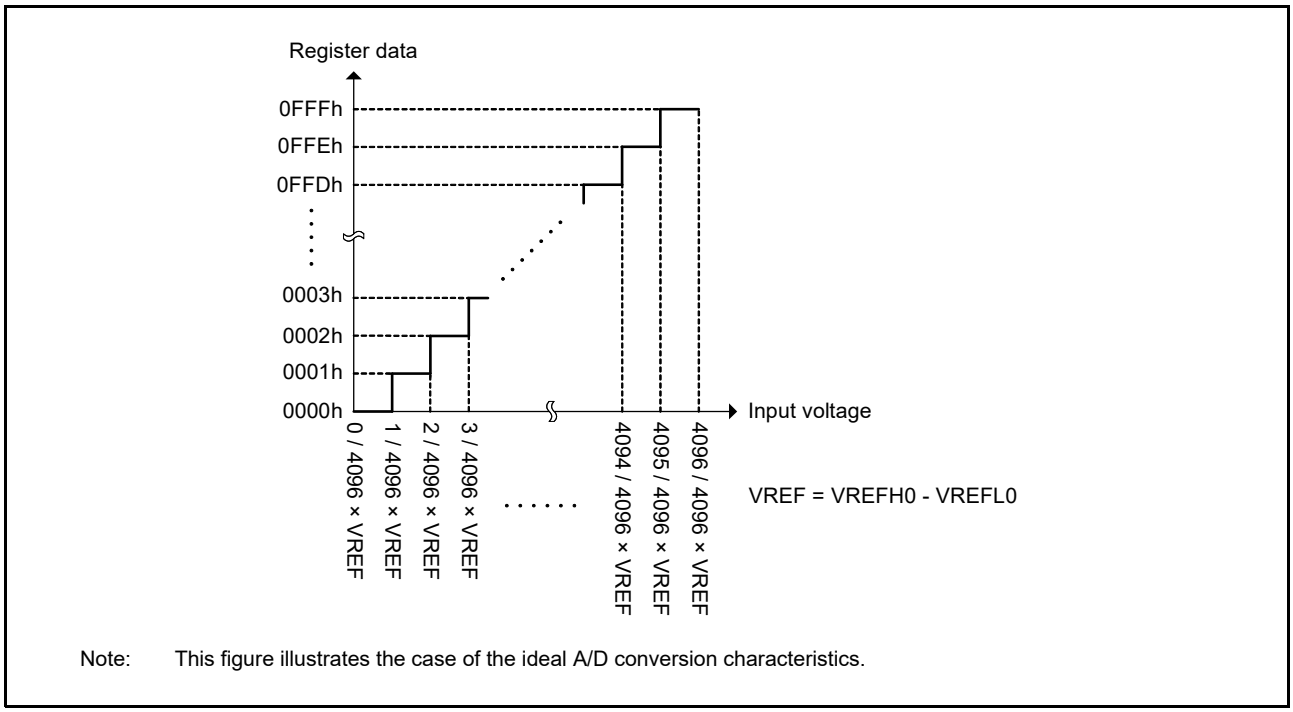


Figure 60.51 Data Range of A/D Conversion Result (12-Bit Data Length, Unsigned Data, Single-Ended Input)

(4) 10-bits data length format

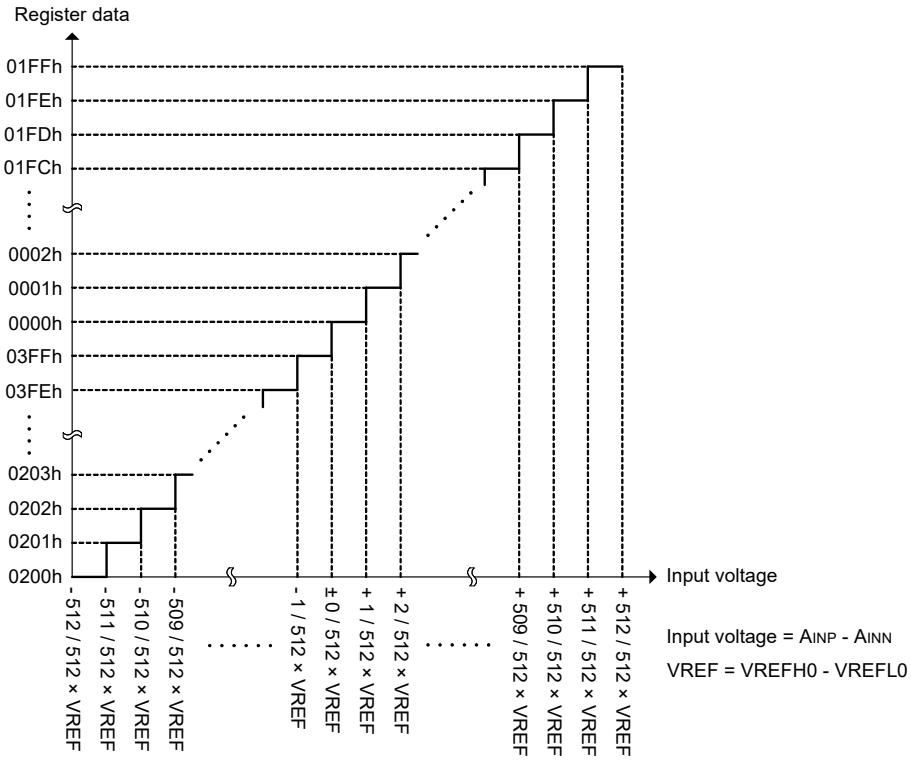
Table 60.57 shows the alignment of A/D conversion data on 10-bit data format. Figure 60.52 and Figure 60.53 show the data range for 10-bit data format.

For 10-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0200h (-VREFH0) to 03FFh (+VREFH0). In 10-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0000h (VREFL0) to 03FFh (VREFH0).

For 10-bit data length, the upper 6 bits (bit15 to bit10) are always 0.

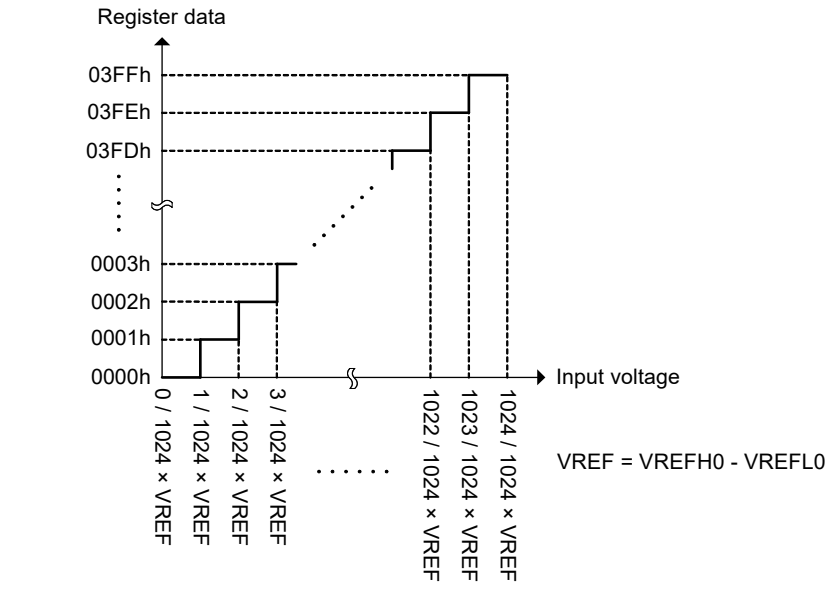
Table 60.57 Alignment of A/D Conversion Result Data (10-Bit Data Length)

Bit number:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit symbol:	0	0	0	0	0	0	DATA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.52 Data Range of A/D Conversion Result (10-Bit Data Length, Signed Data, Differential Input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 60.53 Data Range of A/D Conversion Result (10-Bit Data Length, Unsigned Data, Single-Ended Input)

60.4.8 Compare Match Function

60.4.8.1 Compare Match

The compare match function compares the A/D conversion result with the reference value set in the compare match table register (ADCMPTBRn (n = 0 to 7). The compare match function compares the A/D conversion data after the data formatting process.

(1) Compare match mode

The comparison mode for detecting the compare match is selected in ADCMPMDRm.CMPMDn[1:0] (m = 0, 1. n = 0 to 7). The comparison mode can be selected from the following four modes.

1. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value.
2. Compare match is detected when the A/D conversion value is less than or equal to the specified lower limit.
3. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value, or less than or equal to the lower limit value.
4. Compare match is detected when the A/D conversion value is within the specified upper and lower limit values (when the value is lower than or equal to the upper limit value and greater than or equal to the lower limit value).

Figure 60.54 shows an example of compare match detection.

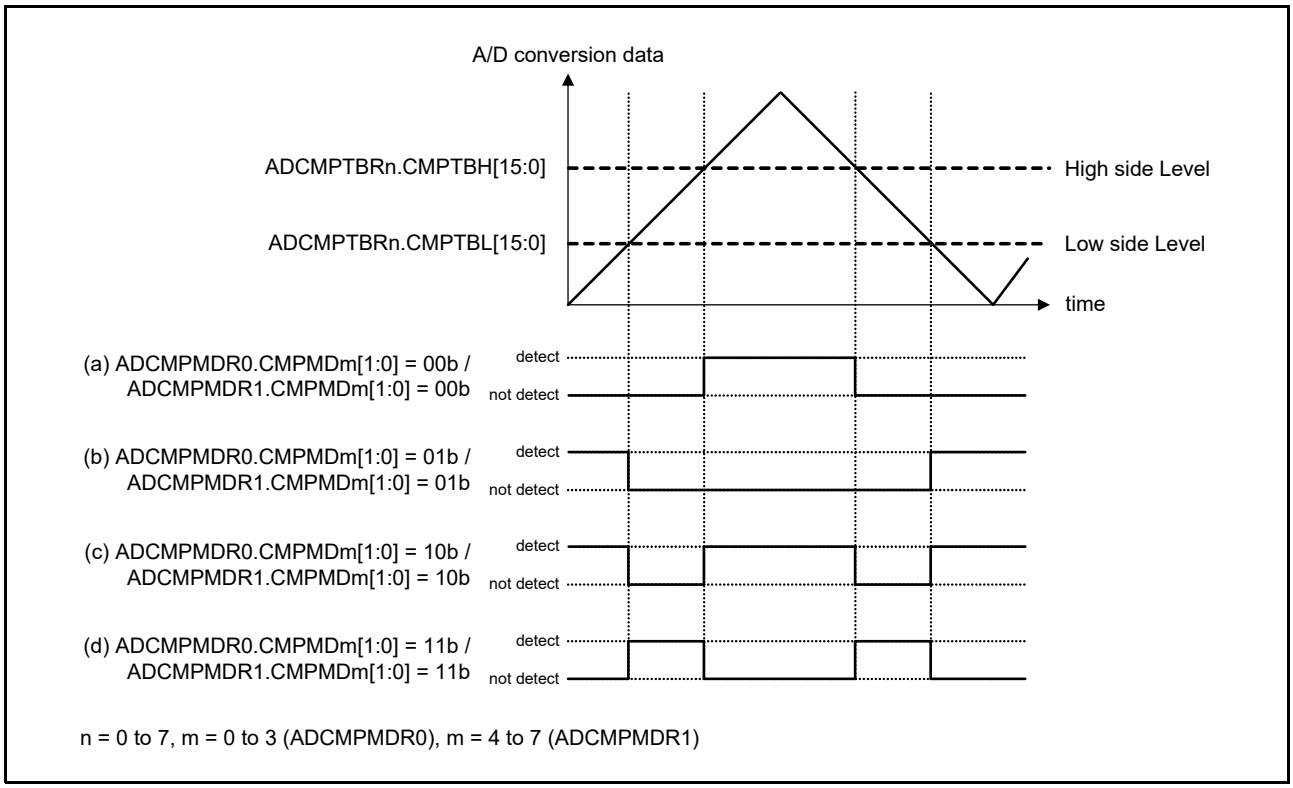


Figure 60.54 Compare Match Detection

(2) Compare match detection

To use the compare match function, enable the compare match table used in ADCMPENR register, and select the compare match table in ADDOPCRBn.CMPTBLEm (n = 0 to 49, m = 0 to 7) for each virtual channel.
When A/D conversion is performed on a virtual channel for which the compare match function is enabled, a compare match is judged based on the A/D conversion result of its channel.
When a compare match is detected, the flags are set in the following status registers:

- ADCMPTBSR: The flag is set to the bit corresponding to the compare match table used when a compare match is detected.
- ADCMPCHSR0, ADCMPCHSR1: The flag is set to the bit corresponding to the analog input channel that detected the compare match.
- ADCMPEXSR: The flag is set to the bit corresponding to Extended Analog Function where a compare match is detected.

To clear the flag of each status register, write 1 to the corresponding bit of ADCMPTBSCR, ADCMPCHSCR0, ADCMPCHSCR1, ADCMPEXSCR.
If compare match interrupt is enabled in ADCMPINTCR register and a compare match using compare match table 0 to 7 is detected, the corresponding interrupt is generated.

60.4.8.2 Composite Compare Match

The composite compare match function generates interrupts and ELC events by combining the comparison results of multiple compare match tables. The combinations and conditions for the compare match table for the composite compare match function can be set using ADCCMPCRn (n = 0 to 7).
Composite compare match interrupt is generated when a compare match is detected that matches the specified condition. Also, if a compare match is not detected, a composite compare unmatch interrupt is generated.
Figure 60.55 shows the relationship between composite compare match and compare match function. Table 60.58 shows the relationship between the composite compare match function and the control register.

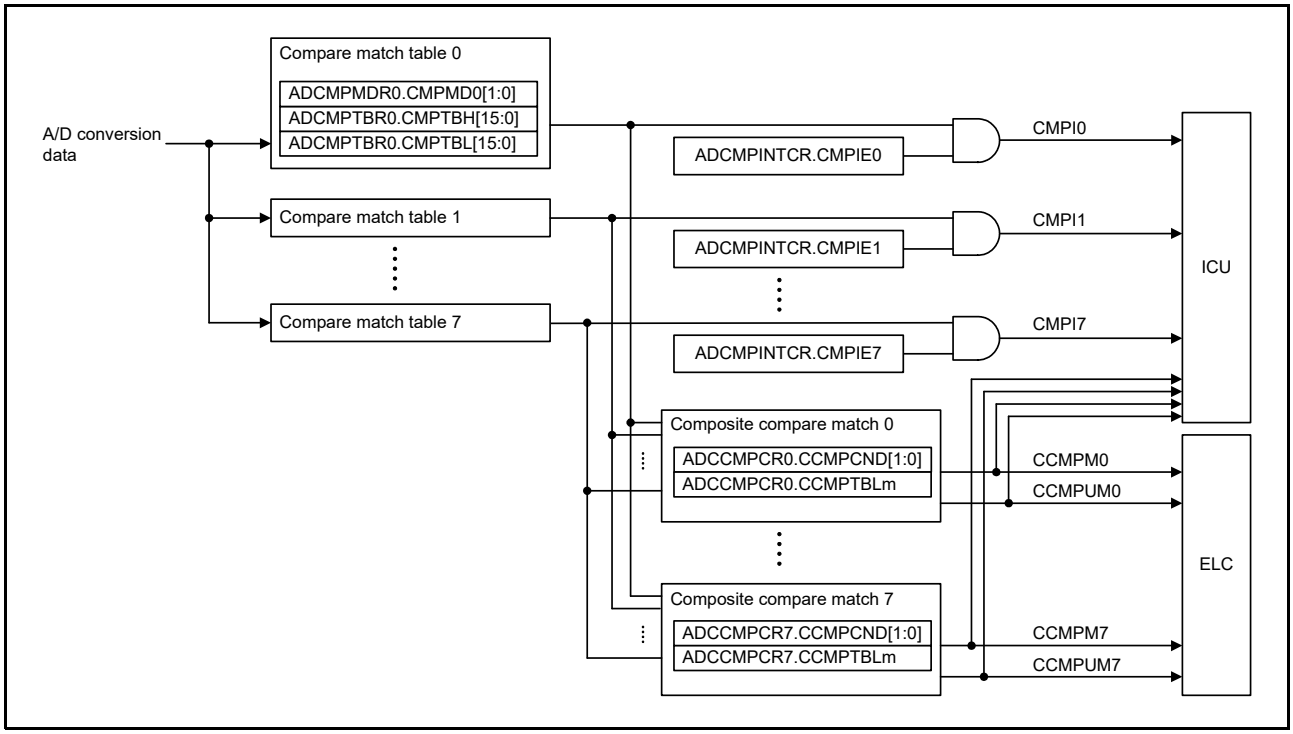


Figure 60.55 Relationship between Composite Compare Match Function and Compare Match Function

Table 60.58 Composite Compare Match and Control Register (m = 0 to 7)

Function Name	Enable Composite Compare Function	Composite Compare Mode Selection	Composite Compare Match Interrupt
Composite compare match 0	ADCCMPCR0.CCMPTBLm	ADCCMPCR0.CCMPCND[1:0]	CCMPM0, CCMPUM0
Composite compare match 1	ADCCMPCR1.CCMPTBLm	ADCCMPCR1.CCMPCND[1:0]	CCMPM1, CCMPUM1
Composite compare match 2	ADCCMPCR2.CCMPTBLm	ADCCMPCR2.CCMPCND[1:0]	CCMPM2, CCMPUM2
Composite compare match 3	ADCCMPCR3.CCMPTBLm	ADCCMPCR3.CCMPCND[1:0]	CCMPM3, CCMPUM3
Composite compare match 4	ADCCMPCR4.CCMPTBLm	ADCCMPCR4.CCMPCND[1:0]	CCMPM4, CCMPUM4
Composite compare match 5	ADCCMPCR5.CCMPTBLm	ADCCMPCR5.CCMPCND[1:0]	CCMPM5, CCMPUM5
Composite compare match 6	ADCCMPCR6.CCMPTBLm	ADCCMPCR6.CCMPCND[1:0]	CCMPM6, CCMPUM6
Composite compare match 7	ADCCMPCR7.CCMPTBLm	ADCCMPCR7.CCMPCND[1:0]	CCMPM7, CCMPUM7

60.4.9 Data Registers

A/D data register (ADDRi (i = 0 to 35)) and Extended A/D data register (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to 18)) store the A/D conversion data of the analog channels corresponding to each register. These registers are updated (overwritten) when A/D conversion for the analog channel corresponding to each register is completed.

60.4.10 FIFO Function

FIFO consists of 16-stage registers and can hold up to 16 A/D conversion data. One FIFO is implemented for each scan group. The A/D conversion data stored in FIFO can be read from ADFIFODRn register (n = 0 to 11).
FIFO acts as a ring buffer. Writing and reading A/D conversion data to and from FIFO is controlled by the write pointer and read pointer inside FIFO. The block diagram of FIFO is shown in Figure 60.56.

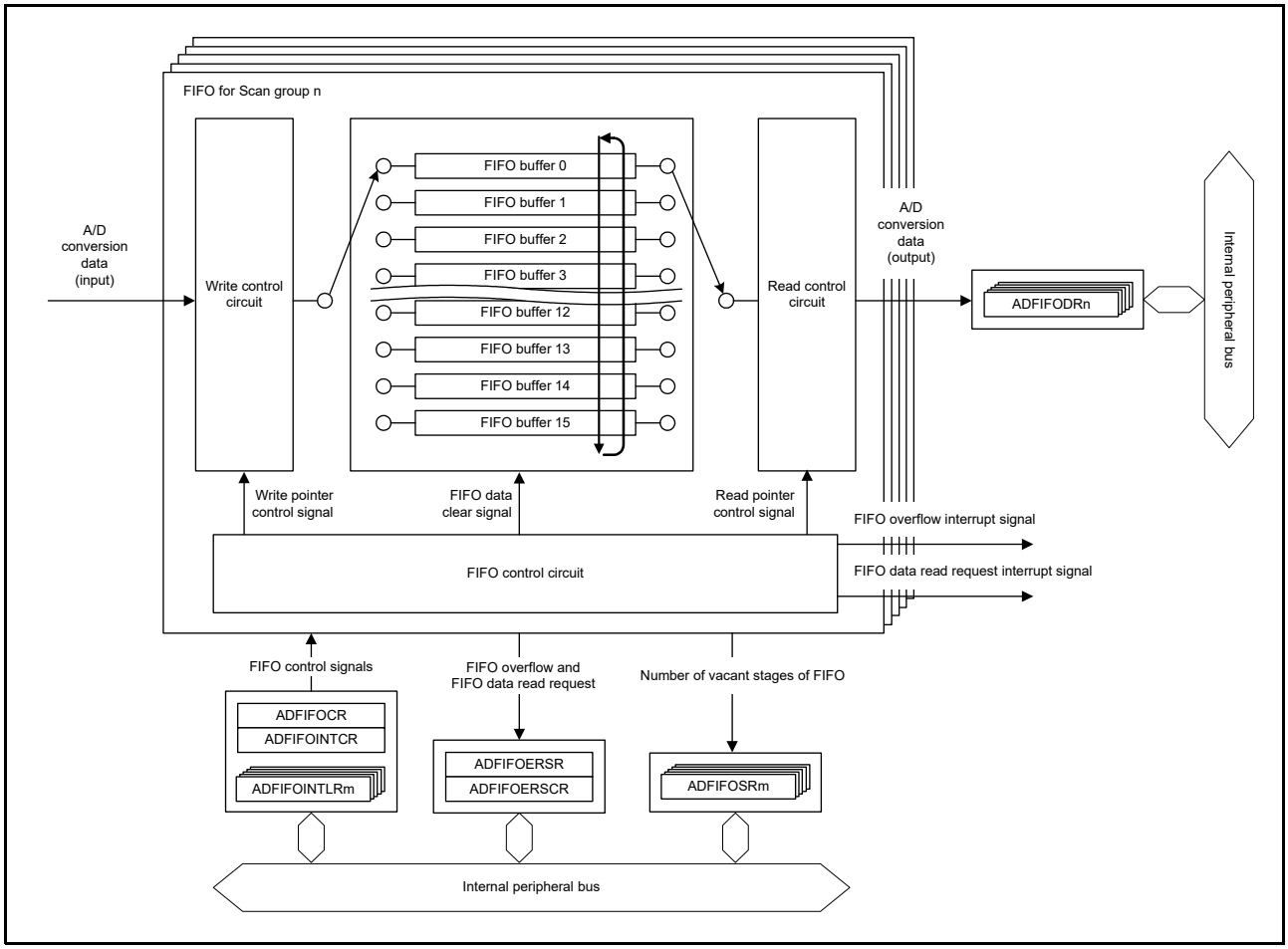


Figure 60.56 The Block Diagram of FIFO (m = 0 to 5, n = 0 to 11)

60.4.10.1 Operation When Writing A/D Conversion Data to FIFO

The A/D conversion data is stored sequentially to the register pointed by the write pointer. When A/D conversion data is written to a register in FIFO, the write pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[4:0] (m = 0 to 5, n = 0 to 11) bits) is decreased 1.
A FIFO overflow occurs when FIFO is full (ADFIFOSRm.FIFOSTn[4:0] bits are 00000b) and an additional A/D conversion data is written to FIFO. When a FIFO overflow occurs, the A/D conversion data is not written and the write pointer does not change.

60.4.10.2 Operation When Reading A/D Conversion Data from FIFO

When A/D conversion data is read from FIFO data register (ADFIFODRn (n = 0 to 11)), the data in the register pointed to by FIFO read pointer is read. When A/D conversion data is read from FIFO, the read pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[4:0] (m = 0 to 5, n = 0 to 11) bits) is increased 1.
If FIFO is empty (ADFIFOSRm.FIFOSTn[4:0] bits are 11111b) and the reading from the FIFO is occurred, an invalid data is read. (If FIFO is read after a reset or after FIFO data is cleared, 00000000h is read. If data had been written to register in FIFO, the previous data is read.) The read pointer does not change in the read operation when FIFO is empty.
Read A/D conversion data from FIFO data register (ADFIFODRn) by 32-bit access. For details, see section 60.4.10.6, Usage Notes on FIFO.

60.4.10.3 FIFO Data Read Request and Overflow

When A/D conversion data is written to or read from FIFO, the write pointer or read pointer changes. Depending on the state of the write pointer and the read pointer, the number of vacant stages in FIFO is reflected in ADFIFOSRm.FIFOSTn[4:0] (m = 0 to 5, n = 0 to 11).
FIFO data read request flag is set (ADFIFOERSR.FIFOFLFn = 1 (n = 0 to 11)) when the number of vacant stages become less than or equal to the value specified in ADFIFOINTLRm.FIFOILVn[4:0] (m = 0 to 5, n = 0 to 11) bit. If a FIFO data read request occurs, FIFO data read request interrupt can be generated.
When FIFO is full (ADFIFOSRm.FIFOSTn[4:0] bits are 00000b) and additional A/D conversion data are written, FIFO overflow flag is set (ADFIFOERSR.FIFOOVFn = 1 (n = 0 to 11)). If a FIFO overflow occurs, FIFO data overflow interrupt can be generated.

60.4.10.4 Clearing FIFO Data

The data stored in the registers in FIFO can be cleared by writing to ADFIFODCR.FIFODCn (n = 0 to 11) bit. When writing 1 to ADFIFODCR.FIFODCn bit, the registers, read pointer, and write pointer in FIFO of the scan group corresponding to FIFODCn bit are initialized. Clearing FIFO data using FIFODCn bit should be performed while A/D conversion is not being performed.

60.4.10.5 Clearing FIFO Data by Starting/Resuming Scanning

For clearing the unnecessary data stored in the registers in FIFO at the start of a scan group or at the restart of scanning after the stop of the previous scanning in group priority operation, write to ADFIFOCR.FIFOCEn (n = 0 to 11) bit. Clearing can be enabled or disabled individually for each scan group. When writing 1 to ADFIFOCR.FIFOCEn (n = 0 to 11) bit, the registers, read pointer, and write pointer in FIFO of the scan group corresponding to FIFOCEn bit are initialized.

60.4.10.6 Usage Notes on FIFO

Reading the A/D conversion data from the FIFO data register (ADFIFODRn (n = 0 to 11)) should be performed with a single 32-bit access; access to each bit of ADFIFODRn, 16-bit read access, and 8-bit read access are prohibited. If these restrictions are violated, the operation of FIFO is not guaranteed. (If these restrictions are violated, the read pointer may not change normally, and correct data may not be able to read. Alternatively, the read pointer may advance unintentionally, and A/D conversion data may be lost.)

If you want to restart scanning and rescan from the first channel during group priority operation, enable FIFO clearing (ADFIFOCR.FIFOCEn = 1 (n = 0 to 11)) for scan groups other than the highest priority group.

When FIFO clear is enabled using ADFIFOCR.FIFOCEn (n = 0 to 11), all data remaining in the FIFO at the start of scan is cleared. Control the scan start timing so that the FIFO is not cleared before the necessary FIFO data is read. When the clearing FIFO data function by starting/resuming scanning is enabled using ADFIFOCR.FIFOCEn (n = 0 to 11), all data remaining in the FIFO at the start of scan, or all data before the stop of the previous scanning at the restart of a scan, is cleared. Note that the usage of the clearing FIFO data function by starting/resuming scanning is prohibited for other than SAR mode and group priority operation.

60.4.11 A/D Conversion Data Output Decimation

In oversampling mode with same channel continuous scan mode, the output of A/D conversion data can be decimated (decimation). The A/D conversion data output interval can be set for each unit using ADDECCR.DCIMm (m = 0 to 2). When A/D conversion data output is decimated using this function, the interrupt output is decimated as well. When this function is used in addition/average mode, the A/D conversion data output interval is the period when the specified number of times the data has been added or averaged and then decimated by this function. In addition/average mode, the data to be decimated by this function is the data before addition/averaging. Figure 60.57 shows an example of operation when only one result of conversion is output for every four A/D conversions. Figure 60.58 shows an example of operation when only one result of conversion is output for every four A/D conversions with 4-time addition.

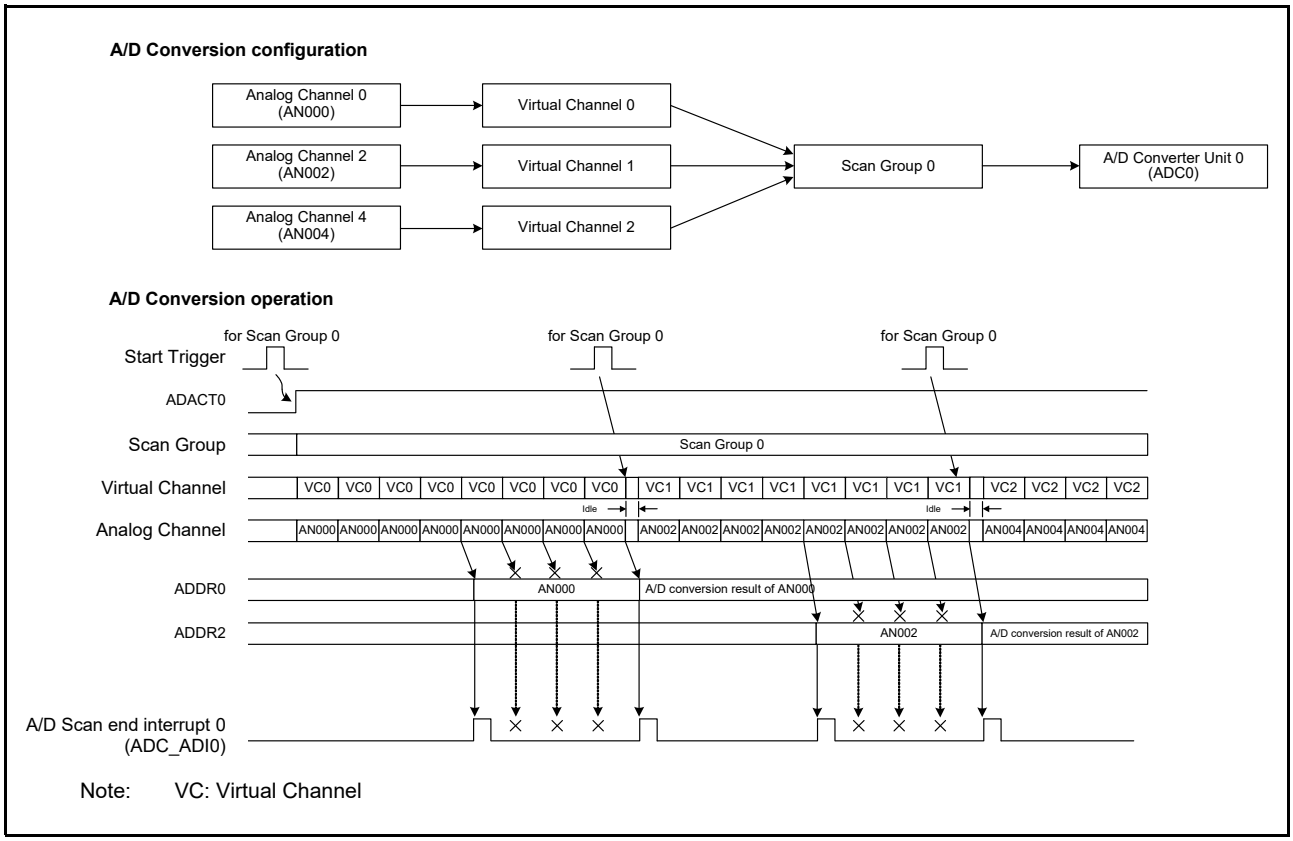


Figure 60.57 Example of the Decimation Function Operation When Only One Result of Conversion Is Output for Every Four A/D Conversions

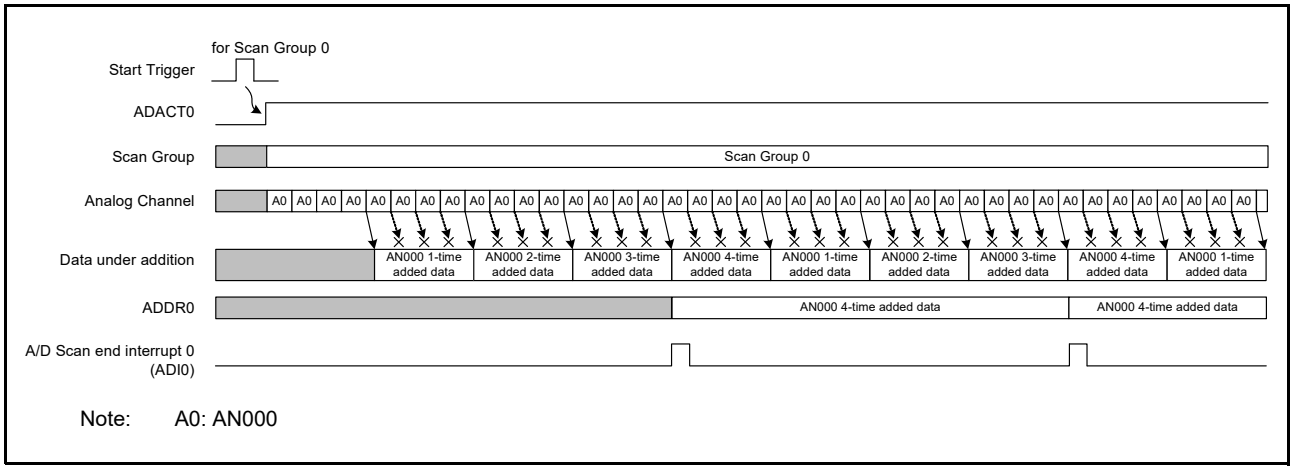


Figure 60.58 Example of the Decimation Function Operation When Only One Result of Conversion Is Output for Every Four A/D Conversions with 4-time Addition in the Addition Mode

60.4.12 A/D Conversion Data Error Detection

The ERR bit in A/D data register (ADDRi (i = 0 to 35)), Extended A/D data register (ADEXDRj (j = 0 to 2, 4 to 6, 8, 9, 16 to18)), and FIFO data register (ADFIFODRk (k = 0 to 11)) is a flag that indicates an A/D conversion data error. An A/D conversion data error indicates that an invalid A/D conversion data has been detected. If an A/D conversion data error occurs (ERR = 1), the A/D conversion data cannot be guaranteed.
A/D conversion data error is generated by the following factors.

- A/D converter error
- A/D conversion overflow

For more details about these errors, see section 60.6, Error Detection.

60.5 Start and Stop Control of A/D Conversion

60.5.1 Software Trigger

A/D conversion of scan group n can be started by writing 1 to ADSTRn.ADST (n = 0 to 11) bit or ADSYSTR.ADSYSTn (n = 0 to 11) bit.
ADSTRn.ADST bit is used to start A/D conversion individually for each scan group. ADSYSTR.ADSYSTn bit is used to start A/D conversion on the scan group assigned to ADC0, ADC1 or ADC2 at the same time.
Except when group priority operation, writing to ADSTRn.ADST bit or ADSYSTR.ADSYSTn bit for the scan groups, which use A/D converter that is in A/D conversion, is ignored.

60.5.2 Peripheral Module Triggers

A/D conversion can be started by triggers from the peripheral modules listed in the followings.

- ELC Trigger
- GPT Trigger
- External Trigger (I/O port)

To perform A/D conversion by a trigger from a peripheral module, configure the trigger for each scan group and enable the trigger input from the peripheral module in ADTRGENR register.

60.5.2.1 ELC Trigger

A/D conversion can be started by an event (ELC event) from Event Link Controller. To start A/D conversion using an ELC event, configure the scan group using the ELC event in ADTRGELCn (n = 0 to 11) register and enable the A/D conversion start trigger in ADTRGENR register. Table 60.59 shows the correspondence between ADTRGELCn register and ELC event.

Table 60.59 Correspondence between ELC Trigger Enable Register and ELC Event

Register Bit	Event Name
ADTRGELCn.TRGELC0	ELC_AD00
ADTRGELCn.TRGELC1	ELC_AD01
ADTRGELCn.TRGELC2	ELC_AD02
ADTRGELCn.TRGELC3	ELC_AD10
ADTRGELCn.TRGELC4	ELC_AD11
ADTRGELCn.TRGELC5	ELC_AD12
ADTRGELCn.TRGELC6	ELC_AD20
ADTRGELCn.TRGELC7	ELC_AD21
ADTRGELCn.TRGELC8	ELC_AD22

n = 0 to 11

60.5.2.2 GPT Trigger

A/D conversion can be started by an interrupt source from the general-purpose PWM Timer (GPT). To start A/D conversion using an interrupt source from the GPT, configure the scan group using the GPT interrupt source in ADTRGGPTn (n = 0 to 11) register and enable the A/D conversion start trigger in ADTRGENR register. Table 60.60 shows the correspondence between ADTRGGPTn register and GPT interrupt sources.

Table 60.60 Correspondence between GPT Trigger Enable Registers and GPT Interrupt Sources

Register Bit	Event Name
ADTRGGPTn.TRGGPTA0	GPT0_ADTRGA
ADTRGGPTn.TRGGPTA1	GPT1_ADTRGA
ADTRGGPTn.TRGGPTA2	GPT2_ADTRGA
ADTRGGPTn.TRGGPTA3	GPT3_ADTRGA
ADTRGGPTn.TRGGPTA4	GPT4_ADTRGA
ADTRGGPTn.TRGGPTA5	GPT5_ADTRGA
ADTRGGPTn.TRGGPTA6	GPT6_ADTRGA
ADTRGGPTn.TRGGPTA7	GPT7_ADTRGA
ADTRGGPTn.TRGGPTA8	GPT8_ADTRGA
ADTRGGPTn.TRGGPTA9	GPT9_ADTRGA
ADTRGGPTn.TRGGPTA10	GPT10_ADTRGA
ADTRGGPTn.TRGGPTA11	GPT11_ADTRGA
ADTRGGPTn.TRGGPTA12	GPT12_ADTRGA
ADTRGGPTn.TRGGPTA13	GPT13_ADTRGA
ADTRGGPTn.TRGGPTA14	GPT14_ADTRGA
ADTRGGPTn.TRGGPTA15	GPT15_ADTRGA
ADTRGGPTn.TRGGPTB0	GPT0_ADTRGB
ADTRGGPTn.TRGGPTB1	GPT1_ADTRGB
ADTRGGPTn.TRGGPTB2	GPT2_ADTRGB
ADTRGGPTn.TRGGPTB3	GPT3_ADTRGB
ADTRGGPTn.TRGGPTB4	GPT4_ADTRGB
ADTRGGPTn.TRGGPTB5	GPT5_ADTRGB
ADTRGGPTn.TRGGPTB6	GPT6_ADTRGB
ADTRGGPTn.TRGGPTB7	GPT7_ADTRGB
ADTRGGPTn.TRGGPTB8	GPT8_ADTRGB
ADTRGGPTn.TRGGPTB9	GPT9_ADTRGB
ADTRGGPTn.TRGGPTB10	GPT10_ADTRGB
ADTRGGPTn.TRGGPTB11	GPT11_ADTRGB
ADTRGGPTn.TRGGPTB12	GPT12_ADTRGB
ADTRGGPTn.TRGGPTB13	GPT13_ADTRGB
ADTRGGPTn.TRGGPTB14	GPT14_ADTRGB
ADTRGGPTn.TRGGPTB15	GPT15_ADTRGB

n = 0 to 11

60.5.2.3 External Trigger

A/D conversion can be started by the input from the external trigger pin (ADTRG0, ADTRG1 and ADTRG2). To start A/D conversion using an external trigger, configure the scan group using the external trigger in ADTRGEXTn (n = 0 to 11) register and enable the A/D conversion start trigger in ADTRGENR register.

External trigger is active low. Before setting to enable the external trigger (ADTRG0, ADTRG1 and ADTRG2), you should input the High level to the external trigger pin.

60.5.3 Trigger Delay

The trigger delay function adds a delay to the A/D conversion start trigger in order to adjust the A/D conversion start timing for each scan group. This function adds a delay to the internal trigger for each scan group generated by accepting an external trigger, ELC trigger or GPT trigger. This function cannot be used to add a delay to a software trigger.
The delay value to be added to the internal trigger is set for each scan group in ADTRGDLRi (i = 0 to 5) register. The delay value to be added is the register setting value × A/D conversion clock (ADCLK) cycle.

60.5.4 Force Stops the A/D Conversion Operation

The scanning operation of the A/D converter can be forcibly stopped by writing 1 to ADSTOPR.ADSTOPm (m = 0 to 2) bit during scanning operation. If the scanning operation is forcibly stopped, the A/D conversion data is not guaranteed.

60.5.4.1 Forced Stop Procedure

Table 60.61 shows the procedure for forced stop of A/D conversion operation. Observe Table 60.61 when performing the forced stop. If this procedure is violated, the A/D converter may not be able to stop and then it may not operate normally. In that case, a reset is required for recovery.

Table 60.61 Procedure for Forced Stop of A/D Conversion Operation

No.	Step Name	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0)
2	Waiting time	After setting the above Step 1, a wait time is required to safely stop the A/D converter. Proceed to the next processing after the specified waiting time has elapsed. For details on the wait time, see section 60.5.4.2, Waiting Time after Disabling Trigger Input for Forced Stop Processing.
3	Checking the A/D converter status	Check if the A/D converter is operating after the waiting time of the above Step 2 has elapsed. If the A/D converter is running (ADSR.ADACTm = 1), proceed to Step 4. If the A/D converter is stopped (ADSR.ADACTm = 0), no further processing is required (proceed to Step 6).
4	Forced stop of A/D converter	Forcibly stop the A/D converter by ADSTOPR register. (Write ADSTOPR.ADSTOPm = 1)
5	Waiting for A/D converter to stop	Wait until the A/D converter stops. (ADSR.ADACTm = 0)
6	End	Processing of forced stop is completed.

m = 0 to 2, n = 0 to 11

Do not write to ADSYSTR register or ADSTRn register while this forced stop procedure is in progress.

60.5.4.2 Waiting Time after Disabling Trigger Input for Forced Stop Processing

In the forced stop procedure shown in Table 60.61, after disabling the trigger input from the peripheral module, a wait time is required to perform the forced stop (Step 2 of Table 60.61). This wait time is the time that must be secured to perform a forced stop safely.
The wait time after disabling the trigger input for forced stop processing is calculated as follows.
[Waiting time after disabling trigger input in forced stop processing]

- When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)
  - (ADTRGDLRi.TRGDLYn[7:0] setting value + ADSYCR.ADCSYCYC[7:0] setting value × 2) × t_{ADCLK}
- When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)
  - (4 + ADTRGDLRi.TRGDLYn[7:0] setting value) × t_{ADCLK}

Note: i = 0 to 5, n = 0 to 11, m = 0 to 2, t_{ADCLK}: ADCLK period

After the waiting time calculated by the above expressions has elapsed, proceed to the Step 3 in Table 60.61. There are two ways to generate the waiting time: (a) Use the timer function built into MCU. (b) Read the status register multiple times. If you choose (b), the number of register reads to generate the waiting time required is calculated by the following equation:

[Number of register reads to generate the waiting time]

N_{RD} = (Wait time after disabling trigger input ÷ (t_{PCLK} × N_{RDCYC})) + 1

N_{RD}: Number of register reads required to generate the trigger input disable wait time (truncate after the decimal point)

t_{PCLK}: The period of the Bus I/F clock (PCLKA)

N_{RDCYC}: Minimum number of register read access cycles (see section 5, I/O Registers.)

60.6 Error Detection

60.6.1 A/D Converter Error

An A/D converter error is detected when an abnormal operation of the A/D converter occurs. If an A/D converter error is detected, the A/D conversion result is not guaranteed.
An A/D converter error occurs in the following cases:

[A/D Converter Error Generation Conditions]

- When A/D conversion is performed without self-calibration.
- When the operating frequency of ADCLK exceeds the guaranteed operating frequency range specified in the electric characteristics.
- When the successive approximation time of the A/D converter exceeds the guaranteed operating range specified in the electric characteristics.
- When an accidental abnormal operation occurs in the A/D converter due to external factors.

If an A/D converter error is detected, the flag is set in ADERSR.ADERFn (n = 0 to 2) bit. To clear the A/D converter error flag, write 1 to the corresponding bit in ADERSCR register.

Notes on A/D Converter Error

The A/D converter error is an auxiliary function used to detect abnormal operation of the A/D converter. It does not guarantee the reliable detection of abnormal operation. Even under the A/D converter error occurrence conditions described above, an A/D converter error may not be detected depending on the individual chip differences and chip operating conditions.

60.6.2 A/D Conversion Overflow

A/D conversion overflow is detected when A/D conversion data exceeds the range that can be handled in the specified data format. If A/D conversion overflow occurs, the A/D conversion data becomes the upper or lower limit value of the specified data format. For the range that can be handled as A/D conversion data, see section 60.4.7, Data Format.
A/D conversion overflow is detected in the following cases:

- When the input to the A/D converter exceeds the range between VREFH0 to VREFL0
  - Single-ended input: When the input to the A/D converter exceeds VREFH0 or falls below VREFL0
  - Differential input: When the differential input to the A/D converter exceeds +VREFH0 or falls below -VREFH0
- When overflow occurs by the internal processing (calculation) for the A/D conversion data shown in the following:
  - Gain error and Offset error calibration
  - User's Gain/User's Offset adjustment function
  - When using A/D-converted value addition function
  - Data formatting process

When an A/D conversion overflow is detected, the flag is set in the following status register:

- ADOVFERSR: The flag is set to the bit corresponding to the A/D converter that detected the A/D conversion overflow.
- ADOVFCHSR0, ADOVFCHSR1: The flag is set to the bit corresponding to the analog input channel on which A/D conversion overflow is detected.
- ADOVFEXSR: The flag is set to the bit corresponding to the extended analog channel on which A/D conversion overflow is detected.

To clear the flags in the status registers, write 1 to the corresponding bit in ADOVFERSCR, ADOVFCHSCR0, ADOVFCHSCR1, ADOVFEXSCR.

60.6.3 FIFO Overflow

If A/D-converted data is written (added) to FIFO while FIFO is full, a FIFO overflow is detected. For more details about FIFO overflow, see section 60.4.10, FIFO Function.

60.7 Procedure for Setting up and Changing

60.7.1 Initial Setup Procedure

Table 60.62 shows the initial setup procedure.

Table 60.62 Initial Setup Procedure

No.	Step	Description
1	Release module-stop	Release the module-stop bit for ADC in MSTPCR register.*1
2	I/O port configuration	Set ASEL bit of the pin used as analog input to 1.
3	Synchronous operation configuration	Set the synchronous operation function. The synchronous operation function is disabled at the initial value of the ADSYCR register after reset released. If the synchronous operation function is used, be sure to enable the synchronous operation function (ADSYCR.ADSYDISm = 1 (m = 0 to 2))*2.
4	ADCLK configuration	Set the clock source and division ratio for ADCLK. Then, set ADCLKENR.CLKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1)*3.
5	A/D conversion configuration	Configure the A/D conversion settings.
6	Wait for operation stabilization	Wait until the operating stabilization times specified in the Electrical Characteristics have elapsed.
7	Self-calibration	Self-calibration must be executed prior to starting A/D conversion. Set up for self-calibration and execute it. For details, see section 60.3.9, Self-Calibration.
8	Enabling constant sampling	When the constant sampling function is to be used, enable the function after self-calibration is completed.
9	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
10	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note: In either of the following cases, release the GPT module-stop state using the Module Stop Register E (MSTPCRE) before the corresponding step.
- An external signal (the synchronization signal from the GPT) is to be used in Step 3.
- GPTCLK is to be selected in Step 4.
For details, see section 11, Low Power Consumption.

Note 1. When the ADC units need to be stopped individually, set the corresponding ADUSLPCRm.ADUSLPm bits to 1 (m = 0 to 2).
Note 2. When an external signal (the synchronization signal from the GPT) is to be used, set the ADSYCR.ADSYEXTEN bit to 1.
Note 3. When an external signal (the synchronization signal from the GPT) is to be used, confirm that the setting of the ADCLKSR.SYLOCKSR bit is 1.

60.7.2 Procedure for Changing ADCLK Settings

Table 60.63 shows the procedure for changing ADCLK setting. Table 60.64 shows the procedure for changing ADCLK setting while an external signal (the synchronization signal from the GPT) is in use.

Table 60.63 Procedure for Changing ADCLK Setting (1/2)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 11))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 60.5.4, Force Stops the A/D Conversion Operation.
3	Stop ADCLK supplies	Set ADCLKENR.CLKEN bit to 0. Then, wait for ADCLK to stop (ADCLKSR.CLKSR = 0).
4	Change ADCLK setting	Change the clock-source and the division ratio for ADCLK.*1
5	Started supplying ADCLK	Set ADCLKENR.CLKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1).

Table 60.63 Procedure for Changing ADCLK Setting (2/2)

No.	Step	Description
6	Change the A/D conversion configuration	Change the following settings according to the changed ADCLK frequency. • Successive approximation time for A/D converter • Number of sampling states for A/D conversion • Number of sampling states and hold mode switching states for channel- dedicated sample-and-hold circuit*2 • Number of states for self-calibration operation (A/D converter and channel- dedicated sample-and-hold circuit*2) • Synchronous operation period*3 • Disconnection detection assist period*4 If other settings related to A/D conversion are also to be changed, change them in this step.
7	Wait for operation stabilization	Wait until the operating stabilization times specified in the section 67, Electrical Characteristics have elapsed.
8	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 60.3.9, Self-Calibration.
9	Enabling constant sampling	When the constant sampling function is to be used, enable the function after self-calibration is completed.
10	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
11	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. When GPTCLK is selected, release the GPT module-stop state using the Module Stop Register E (MSTPCRE). For details, see section 11, Low Power Consumption.
Note 2. Setting is not required when channel-dedicated sample-and-hold circuit is not used.
Note 3. No change is required when synchronous operation setting is disabled (ADSYCR, ADSYDISm = 1 (m = 0 to 2))
Note 4. Setting is not required when disconnection detection assist function is not used.

Table 60.64 Procedure for Changing ADCLK Setting While an External Signal (the Synchronization Signal from the GPT) is in Use (ADSYCR.ADSYEXTEN = 1) (1/2)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 11))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 60.5.4, Force Stops the A/D Conversion Operation.
3	Stop of the synchronization signal from the GPT	Stop the synchronization signal from the GPT. After stopping the signal, wait for at least 3 ADCLK.
4	Stop ADCLK supplies	Set ADCLKENR.CLKEN bit to 0. Then, wait for ADCLK to stop (ADCLKSR.CLKSR = 0).
5	Change ADCLK setting	Change the clock-source and the division ratio for ADCLK.*1
6	Start of the synchronization signal from the GPT	Start the output of the synchronization signal from the GPT.
7	Started supplying ADCLK	Set ADCLKENR.CLKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1)*2.
8	Confirmation of synchronization with an external signal (the synchronization signal from the GPT)	Confirm that the setting of the ADCLKSR.SYLOCKSR bit is 1.
9	Change the A/D conversion configuration	Change the following settings according to the changed ADCLK frequency. • Successive approximation time for A/D converter • Number of sampling states for A/D conversion • Number of sampling states and hold mode switching states for channel- dedicated sample-and-hold circuit*3 • Number of states for self-calibration operation (A/D converter and channel- dedicated sample-and-hold circuit*3) • Synchronous operation period • Disconnection detection assist period*4 If other settings related to A/D conversion are also to be changed, change them in this step.

Table 60.64 Procedure for Changing ADCLK Setting While an External Signal (the Synchronization Signal from the GPT) is in Use (ADSYCR.ADSYEXTEN = 1) (2/2)

No.	Step	Description
10	Wait for operation stabilization	Wait until the operating stabilization times specified in the section 67, Electrical Characteristics have elapsed.
11	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 60.3.9, Self-Calibration.
12	Enabling constant sampling	When the constant sampling function is to be used, enable the function after self-calibration is completed.
13	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
14	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. When GPTCLK is selected, release the GPT module-stop state using the Module Stop Register E (MSTPCRE). For details, see section 11, Low Power Consumption.
Note 2. The time until the supply of the clock is resumed is longer than that while the setting of the ADSYCR.ADSYEXTEN bit is 0 by up to 1 ADCLK.
Note 3. Setting is not required when channel-dedicated sample-and-hold circuit is not used.
Note 4. Setting is not required when disconnection detection assist function is not used.

60.7.3 Procedure for Changing the Settings of the A/D Converter

Table 60.65 shows the procedure for changing the A/D conversion configuration while an external signal (the synchronization signal from the GPT) is in use. If there is a change in ADCLK setting, follow the procedures in section 60.7.2, Procedure for Changing ADCLK Settings.

Table 60.65 Procedure for Changing the Settings of the A/D Converter while an External Signal (the Synchronization Signal from the GPT) is in Use (ADSYCR.ADSYEXTEN = 1)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 11))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 60.5.4, Force Stops the A/D Conversion Operation.
3	Stop of the synchronization signal from the GPT	Stop the synchronization signal from the GPT. After stopping the signal, wait for at least 3 ADCLK.
4	Change the A/D conversion configuration	Change the A/D conversion settings (except the settings for ADCLK).
5	Wait for operation stabilization	Wait until the operating stabilization times specified in the section 67, Electrical Characteristics have elapsed.
6	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 60.3.9, Self-Calibration.
7	Enabling constant sampling	When the constant sampling function is to be used, enable the function after self-calibration is completed.
8	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
9	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

60.7.4 Procedure for the A/D Converter Entering to and Returning from the Software Standby Mode

Table 60.66 lists the procedure for the A/D converter returning from the software standby mode while an external signal (the synchronization signal from the GPT) is in use.

Table 60.66 Procedure for the A/D Converter Returning from the Software Standby Mode while the Synchronization Signal from the GPT is in Use (ADSYCR.ADSYEXTEN = 1)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 11))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 60.5.4, Force Stops the A/D Conversion Operation.
3	Stop of the synchronization signal from the GPT	Stop the synchronization signal from the GPT. After stopping the signal, wait for at least 3 ADCLK.
4	Stop ADCLK supplies	Set ADCLKENR.CLKEN bit to 0. Then, wait for ADCLK to stop (ADCLKSR.CLKSR = 0).
5	Entering to the software standby mode	Enter the software standby mode.
6	Returning from the software standby mode	Release the software standby mode.
7	ADCLK configuration	Set the clock source and division ratio for ADCLK.*1
8	Start of the synchronization signal from the GPT	Start the output of the synchronization signal from the GPT.
9	Started supplying ADCLK	Set ADCLKENR.CLKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1)*2.
10	Confirmation of synchronization with an external signal (the synchronization signal from the GPT)	Confirm that the setting of the ADCLKSR.SYLOCKSR bit is 1.
11	A/D conversion configuration	Change the following settings according to the changed ADCLK frequency. • Successive approximation time for A/D converter • Number of sampling states for A/D conversion • Number of sampling states and hold mode switching states for channel- dedicated sample-and-hold circuit*3 • Number of states for self-calibration operation (A/D converter and channel- dedicated sample-and-hold circuit*3) • Synchronous operation period • Disconnection detection assist period*4 If other settings related to A/D conversion are also to be changed, change them in this step.
12	Wait for operation stabilization	Wait until the operating stabilization times specified in the ssection 67, Electrical Characteristics have elapsed.
13	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 60.3.9, Self-Calibration.
14	Enabling constant sampling	When the constant sampling function is to be used, enable the function after self-calibration is completed.
15	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
16	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. When GPTCLK is selected, release the GPT module-stop state using the Module Stop Register E (MSTPCRE). For details, see section 11, Low Power Consumption.
Note 2. The time until the supply of the clock is resumed is longer than that while the setting of the ADSYCR.ADSYEXTEN bit is 0 by up to 1 ADCLK.
Note 3. Setting is not required when channel-dedicated sample-and-hold circuit is not used.
Note 4. Setting is not required when disconnection detection assist function is not used.

60.8 Interrupt Sources

Table 60.67 lists ADC interrupt sources.

For more details about interrupts, see section 15, Interrupt Controller (ICUM) . For more details about ELC events, see section 21, Event Link Controller (ELC).

Table 60.67 Interrupt Sources (1/3)

Interrupt Request	Symbol	Description	Status Flag
A/D converter error interrupt	ERR0	Generated when abnormal operation of ADC0 is detected.	ADERSR.ADERF0
	ERR1	Generated when abnormal operation of ADC1 is detected.	ADERSR.ADERF1
	ERR2	Generated when abnormal operation of ADC2 is detected.	ADERSR.ADERF2
A/D converter self-calibration end interrupt	CALEND0	Generated when self-calibration operation of ADC0 is finished.	ADCALENDSR.CALENDF0
	CALEND1	Generated when self-calibration operation of ADC1 is finished.	ADCALENDSR.CALENDF1
	CALEND2	Generated when self-calibration operation of ADC2 is finished.	ADCALENDSR.CALENDF2
A/D scan end interrupt	ADI0	Generated at the scan end of scan group 0	ADSCANENDSR.SCENDF0
	ADI1	Generated at the scan end of scan group 1	ADSCANENDSR.SCENDF1
	ADI2	Generated at the scan end of scan group 2	ADSCANENDSR.SCENDF2
	ADI3	Generated at the scan end of scan group 3	ADSCANENDSR.SCENDF3
	ADI4	Generated at the scan end of scan group 4	ADSCANENDSR.SCENDF4
	ADI5	Generated at the scan end of scan group 5	ADSCANENDSR.SCENDF5
	ADI6	Generated at the scan end of scan group 6	ADSCANENDSR.SCENDF6
	ADI7	Generated at the scan end of scan group 7	ADSCANENDSR.SCENDF7
	ADI8	Generated at the scan end of scan group 8	ADSCANENDSR.SCENDF8
	ADI9	Generated at the scan end of scan group 9	ADSCANENDSR.SCENDF9
	ADI10	Generated at the scan end of scan group 10	ADSCANENDSR.SCENDF10
	ADI11	Generated at the scan end of scan group 11	ADSCANENDSR.SCENDF11
Limiter clip interrupt	LIMCLPI	Generated when a limiter clip using limiter table 0 to 7 is detected for A/D conversion.	ADLIMGRSR.LIMGRFn ADLIMCHSR0.LIMCHFi ADLIMCHSR1.LIMCHFi ADLIMEXSR.LIMEXFj
A/D conversion overflow interrupt	RESOVF0	Generated when the overflow is detected in A/D conversion result with ADC0.	ADOVFERSR.ADOVFEF0 ADOVFCHSR0.OVFCHFi ADOVFCHSR1.OVFCHFi ADOVFEXSR.OVFEXFj
	RESOVF1	Generated when the overflow is detected in A/D conversion result with ADC1.	ADOVFERSR.ADOVFEF1 ADOVFCHSR0.OVFCHFi ADOVFCHSR1.OVFCHFi ADOVFEXSR.OVFEXFj
	RESOVF2	Generated when the overflow is detected in A/D conversion result with ADC2.	ADOVFERSR.ADOVFEF2 ADOVFCHSR0.OVFCHFi ADOVFCHSR1.OVFCHFi ADOVFEXSR.OVFEXFj

Table 60.67 Interrupt Sources (2/3)

Interrupt Request	Symbol	Description	Status Flag
Compare match interrupt	CMPI0	Generated when a compare match with compare match table 0 is detected.	ADCMPTBSR.CMPTBF0
	CMPI1	Generated when a compare match with compare match table 1 is detected.	ADCMPTBSR.CMPTBF1
	CMPI2	Generated when a compare match with compare match table 2 is detected.	ADCMPTBSR.CMPTBF2
	CMPI3	Generated when a compare match with compare match table 3 is detected.	ADCMPTBSR.CMPTBF3
Compare match interrupt	CMPI4	Generated when a compare match with compare match table 4 is detected.	ADCMPTBSR.CMPTBF4
	CMPI5	Generated when a compare match with compare match table 5 is detected.	ADCMPTBSR.CMPTBF5
	CMPI6	Generated when a compare match with compare match table 6 is detected.	ADCMPTBSR.CMPTBF6
	CMPI7	Generated when a compare match with compare match table 7 is detected.	ADCMPTBSR.CMPTBF7
Composite compare match interrupt	CCMPM0 CCMPM1 CCMPM2 CCMPM3 CCMPM4 CCMPM5 CCMPM6 CCMPM7	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF7
	CCMPUM0 CCMPUM1 CCMPUM2 CCMPUM3 CCMPUM4 CCMPUM5 CCMPUM6 CCMPUM7	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is not detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF7

Table 60.67 Interrupt Sources (3/3)

Interrupt Request	Symbol	Description	Status Flag
FIFO data read request interrupt	FIFOREQ0	Generated when the number of vacant stages in FIFO for scan group 0 become less than or equal to the specified value.	ADFIFOSR0.FIFOST0[4:0]
	FIFOREQ1	Generated when the number of vacant stages in FIFO for scan group 1 become less than or equal to the specified value.	ADFIFOSR0.FIFOST1[4:0]
	FIFOREQ2	Generated when the number of vacant stages in FIFO for scan group 2 become less than or equal to the specified value.	ADFIFOSR1.FIFOST2[4:0]
	FIFOREQ3	Generated when the number of vacant stages in FIFO for scan group 3 become less than or equal to the specified value.	ADFIFOSR1.FIFOST3[4:0]
	FIFOREQ4	Generated when the number of vacant stages in FIFO for scan group 4 become less than or equal to the specified value.	ADFIFOSR2.FIFOST4[4:0]
	FIFOREQ5	Generated when the number of vacant stages in FIFO for scan group 5 become less than or equal to the specified value.	ADFIFOSR2.FIFOST5[4:0]
	FIFOREQ6	Generated when the number of vacant stages in FIFO for scan group 6 become less than or equal to the specified value.	ADFIFOSR3.FIFOST6[4:0]
	FIFOREQ7	Generated when the number of vacant stages in FIFO for scan group 7 become less than or equal to the specified value.	ADFIFOSR3.FIFOST7[4:0]
	FIFOREQ8	Generated when the number of vacant stages in FIFO for scan group 8 become less than or equal to the specified value.	ADFIFOSR4.FIFOST8[4:0]
FIFO data read request interrupt	FIFOREQ9	Generated when the number of vacant stages in FIFO for scan group 9 become less than or equal to the specified value.	ADFIFOSR4.FIFOST9[4:0]
	FIFOREQ10	Generated when the number of vacant stages in FIFO for scan group 10 become less than or equal to the specified value.	ADFIFOSR5.FIFOST10[4:0]
	FIFOREQ11	Generated when the number of vacant stages in FIFO for scan group 11 become less than or equal to the specified value.	ADFIFOSR5.FIFOST11[4:0]
FIFO data overflow interrupt	FIFOOVF	Generated when FIFO overflow is detected in any of FIFO for scan group 0 to 11.	ADFIFOERSR.FIFOOVF0 to ADFIFOERSR.FIFOOVF11

Note: n = 0 to 11
i = 0 to 31 (ADLIMCHSR0, ADOVFCHSR0), i = 32 to 35 (ADLIMCHSR1, ADOVFCHSR1)
j = 0 to 2, 4 to 6, 8, 9, 16 to 18

(1) A/D converter error interrupt

A/D converter error interrupt can be generated when an A/D converter error is detected.

A/D converter error interrupt (ERRm (m = 0 to 2)) is generated when ADERINTCR.ADEIEm (m = 0 to 2) bit is 1 and ADERSR.ADERFm (m = 0 to 2) bit is 1.

(2) A/D converter self-calibration end interrupt

A/D converter self-calibration end interrupt can be generated at the end of A/D converter self-calibration operation.

A/D converter self-calibration end interrupt (CALENDm (m = 0 to 2)) is generated when ADCALINTCR.CALENDIEm (m = 0 to 2) bit is 1 and ADCALENDSR.CALENDFm (m = 0 to 2) bit is 1.

(3) A/D scan end interrupt

A/D scan end interrupt can be generated at the end of scanning operation of scan group n (n = 0 to 11).

A/D scan end interrupt for scan group 0 to 11 (ADI0 to ADI11) are generated when ADINTCR.ADIEn (n = 0 to 11) bit is set to 1 and ADSCANENDSR.SCENDFn (n = 0 to 11) bit is set to 1.
However, A/D scan end interrupt is not generated when A/D conversion operation (scanning operation) is forcibly stopped by ADSTOPR register.

(4) Limiter clip interrupt

Limiter clip interrupt (LIMCLPI) can be generated when a limiter clip with limiter table i (i = 0 to 7) is detected.
Limiter clip interrupt is generated when either ADLIMINTCR.LIMIEi (i = 0 to 11) bit is 1 and ADLIMGRSR.LIMGRFi (i = 0 to 11) bit is 1 is satisfied.

(5) A/D conversion overflow interrupt

A/D conversion overflow interrupt can be generated when A/D conversion overflow in the A/D conversion result using ADCm (m = 0 to 2) is detected. For details about A/D conversion overflow, see section 60.6.2, A/D Conversion Overflow.
A/D conversion overflow interrupt (RESOVFm (m = 0 to 2)) is generated when ADOVFINTCR.ADOVFIEm (m = 0 to 2) bit is 1 and ADOVFERSR.ADOVFEFm (m = 0 to 2) bit is 1.

(6) Compare match interrupt

Compare match interrupt can be generated when a compare match is detected.
Compare match interrupt (CMPIj (j = 0 to 7)) is generated when ADCMPINTCR.CMPIEj (j = 0 to 7) bit is 1 and ADCMPTBSR.CMPTBFj (j = 0 to 7) bit is 1.

(7) Composite compare match interrupt, Composite compare unmatch interrupt

Composite compare match interrupt (CCMPMk (k = 0 to 7)), compare unmatch interrupt (CCMPUMk (k = 0 to 7)) can be generated by combining the comparison results of two or more compare match tables. For details about composite compare match, see section 60.4.8.2, Composite Compare Match.

(8) FIFO data read request interrupt

FIFO data read request interrupt can be generated when the number of vacant stages in FIFO become less than or equal to the specified value.
FIFO data read request interrupt for scan group 0 to 11 (FIFOREQ0 to FIFOREQ11) are generated when ADFIFOINTCR.FIFOIEn (n = 0 to 11) is set to 1 and ADFIFOSRm.FIFOSTn[4:0] ≤ ADFIFOINTLRm.FIFOILVn[4:0] (m = 0 to 5, n = 0 to 11).
However, FIFO read request interrupt is not generated while the ADFIFOERSR.FIFOFLFn (n = 0 to 11) bit corresponding to its interrupt source is set to 1.

(9) FIFO data overflow interrupt

FIFO data overflow interrupt (FIFOOVF) can be generated when a FIFO overflow is detected in any of FIFO in scan groups 0 to 11.

60.9 Scan Conversion Time

60.9.1 Scan Start Processing Time

The processing time from the input of the A/D conversion start trigger to the start of the scanning operation is shown in Table 60.68 and Figure 60.59.

Table 60.68 Scan Start Processing Time

Item		Symbol	Processing Time
Peripheral module trigger input processing time*1		t_{D_TRG}	[When an external trigger is selected.] • 5 ADSRCCLK + (ADTRGDLRr.TRGDLYn[7:0] + 1) × ADCLK [Other than above] • 1 ADSRCCLK + (ADTRGDLRr.TRGDLYn[7:0] + 1) × ADCLK
Software trigger input processing time		t_{D_SW}	[When ADCLK = PCLKA/1]*2 • Number of access cycles to I/O registers*3 + 1 PCLKA [Other than above] • (Number of access cycles to I/O registers*3) + 1 PCLKA + (1 to 3 ADCLK)
Internal trigger processing time		t_{D_ITRG}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) • 6 ADCLK [Idle start] • 7 to 13 ADCLK [Suspend start]*4 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) • 7 ADCLK [Idle start] • (3 + ADSYCR.ADSYCYC[7:0]) × ADCLK to (2 + 2 × ADSYCR.ADSYCYC[7:0]) × ADCLK [Suspend start]*5
Wait time for synchronous operation		t_{D_SYOP}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) • 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) • 0 to ADSYCR.ADSYCYC[7:0] × ADCLK
Total scan start processing time	Started by peripheral module trigger	t_{D_ADST}	$t_{D_TRG} + t_{D_ITRG} + t_{D_SYOP}$
	Started by software trigger		$t_{D_SW} + t_{D_ITRG} + t_{D_SYOP}$

Note: ADSRCCLK: Clock source of ADCLK
n = 0 to 11, m = 0 to 2, r = 0 to 5

Note 1. It does not include the input delay time from the source of the peripheral module trigger to ADC.

Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

Note 3. It is the number of access cycles to ADSTRn register (n = 0 to 11) or ADSYSTR register. For more details about the number of access cycles for I/O registers, see section 5, I/O Registers.

Note 4. When constant sampling is enabled, it is 7 to (6 + ADSHSTRm.SHSST[7:0]) × ADCLK. When constant sampling is enabled, by setting the sampling time setting of the channel-dedicated sample-and-hold circuit to ADSHSTRm.SHSST[7:0] = 04h (reset value), the effect of the ADSHSTRm.SHSST[7:0] setting on the conversion start time can be ignored.

Note 5. When constant sampling is enabled, (3 + ADSYCR.ADSYCYC[7:0]) × ADCLK to (2 + ADSHSTRm.SHSST[7:0] + ADSYCR.ADSYCYC[7:0]) × ADCLK. When constant sampling is enabled, by setting the sampling time setting of the channel dedicated sample-and-hold circuit to ADSHSTRm.SHSST[7:0] = 04h (reset value), the effect of the ADSHSTRm.SHSST[7:0] setting on the conversion start time can be ignored.

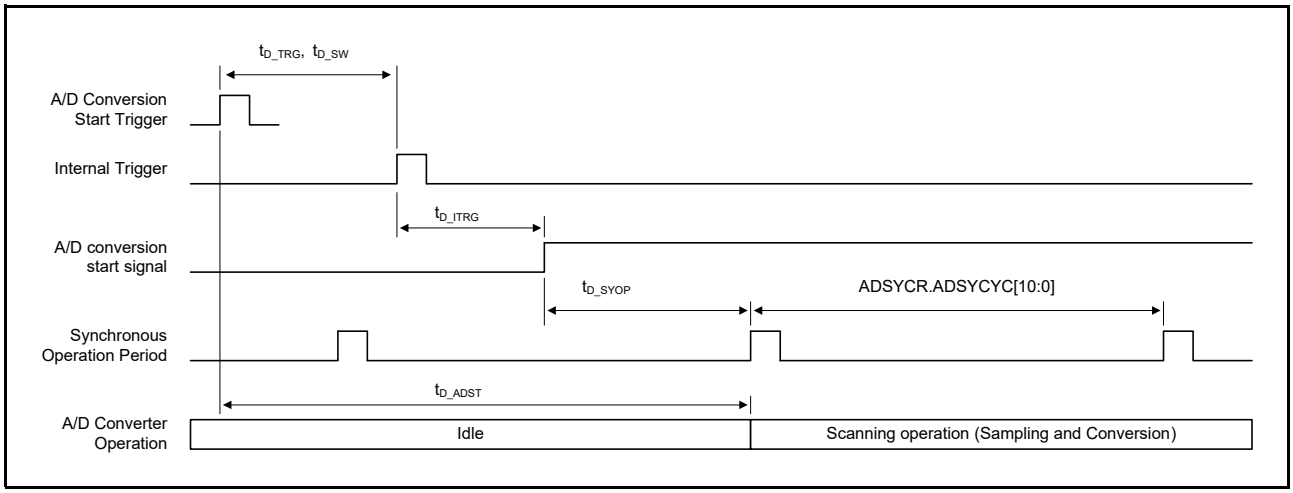


Figure 60.59 Scan Start Processing Time

60.9.2 Conversion Processing Time

The various processing times in the conversion operation are shown in Table 60.69 and Figure 60.60 to Figure 60.64.

Table 60.69 A/D Convert Processing Time (1/2)

Item			Symbol	Processing Time
Channel- dedicated sample-and-hold processing time	Sampling time		t_{SH_SPL}	ADSHSTR0.SHSST[7:0] × ADCLK
	Hold mode switching time		t_{SH_HLD}	ADSHSTR0.SHHST[2:0] × ADCLK
	Sampling mode switching time (only when hybrid mode)		t_{SH_D}	(ADSYCR.ADSYCYC[7:0] - 1) × ADCLK
A/D disconnection detect assist processing time			t_{DDA}	[When disconnection detect assist function is disabled] • 0 [When disconnection detect assist function is enabled] • ADSGDCRn.ADNDIS[7:0] × ADCLK
A/D conversion time	Sampling time		t_{AD_SPL}	ADSSTRp.SSTq[9:0] × ADCLK
	Successive approximation time		t_{AD_CNV}	ADCNVSTR.CSTm[5:0] × ADCLK
A/D conversion data processing time	SAR mode (not using digital filter function)	When ADCLK = PCLKA/1 is set*1	t_{ADDP}	[When A/D-converted value addition/averaging function is not used] • 6 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 7 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 7 ADCLK + (5 to 6 PCLKA) [When A/D-converted value addition/averaging function is used] • 8 ADCLK + (5 to 6 PCLKA)
	Oversampling mode or hybrid mode (using with digital filter function)	When ADCLK = PCLKA/1 is set*1		[When A/D-converted value addition/averaging function is not used] • 8 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 9 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 9 ADCLK + 5 to 6 PCLKA [When A/D-converted value addition/averaging function is used] • 10 ADCLK + 5 to 6 PCLKA
Total A/D conversion time (SAR mode)*3	Channel conversion time*2		t_{ADCH_S}	$(t_{DDA} + t_{AD_SPL} + t_{AD_CNV}) \times N_{ADC}$
	Scan conversion time*4		t_{SCAN_S}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{ADCH_S} [When channel-dedicated sample-and-hold circuit is used] • $t_{SH_SPL} + t_{SH_HLD} + \Sigma t_{ADCH_S}$
Total A/D conversion time (oversampling mode)*3	Oversampling period		t_{OV_OS}	$(t_{DDA} + t_{AD_SPL} + t_{AD_CNV})$
	Channel conversion time		t_{ADCH_O}	$(t_{DDA} + t_{AD_SPL} + t_{AD_CNV}) \times (N_{TAP} + N_{ADC})$
	Scan conversion time*5		t_{SCAN_O}	Σt_{ADCH_O}

Table 60.69 A/D Convert Processing Time (2/2)

Item			Symbol	Processing Time
Total A/D conversion time (hybrid mode)*3	Channel-dedicated sample-and-hold processing time in hybrid mode		t_{HY_SH}	$t_{SH_SPL} + t_{SH_HLD} + t_{SH_D}$
	Oversampling period		t_{HY_OS}	$(t_{DDA} + t_{AD_SPL} + t_{AD_CNV})$
	Scan conversion time	Initial delay	t_{HY_ID}	[When channel-dedicated sample-and-hold circuit is not used] • $(N_{TAP} + N_{ADC}) \times \Sigma t_{HY_OS}$*6 [When channel-dedicated sample-and-hold circuit is used] • $(t_{HY_SH} \times (N_{ADC} + N_{TAP}) - t_{SH_D}) + (N_{TAP} + N_{ADC}) \times \Sigma t_{HY_OS}$*6
		After initial delay time	t_{SCAN_HY}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{HY_OS}*6 [When channel-dedicated sample-and-hold circuit is used] • $t_{HY_SH} + \Sigma t_{HY_OS}$*6

n = 0 to 11, m = 0 to 2, p = 0 to 7, q = 0 to 15

Note: N_{ADC}: This value is the number of the times of the addition/averaging according to the setting value of ADDOPCRBx.ADC[3:0] (x = 0 to 49). If A/D-converted value addition/averaging function is not used, this value is 1.
N_{TAP}: This value is the number of TAP of the digital filter that selected in ADDOPCRAx.DFSEL[2:0] (x = 0 to 49) and ADDFSRm.DFSELy[1:0] (m = 0 to 2, y = 0 to 4).
N_{SGCH}: This value is the number of the channels in the scan group.

Note 1. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

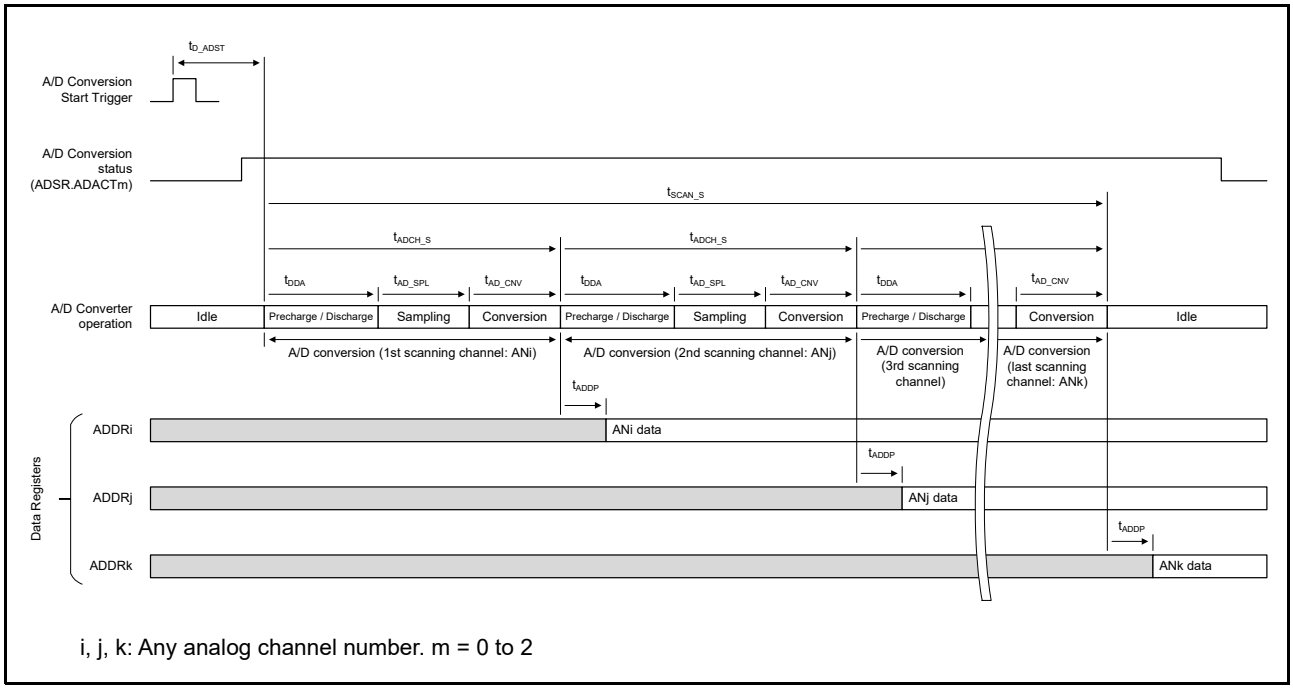
Note 2. It does not include the channel-dedicated sample-and-hold processing time.

Note 3. It does not include the A/D conversion data processing time.

Note 4. This is the sum of the channel conversion times (t_{ADCH_N}) that calculated from the conversion settings for each analog channel assigned to the scan group. If channel-dedicated sample-and-hold circuits are used, channel-dedicated sample-and-hold processing times (t_{SH_SPL} and t_{SH_HLD}) are also added.

Note 5. This is the sum of the channel conversion times (t_{ADCH_O}) that calculated from the conversion settings for each analog channel assigned to the scan group.

Note 6. This is the sum of the oversampling period (t_{HY_OS}) that calculated from the conversion settings for each analog channel assigned to the scan group.



i, j, k: Any analog channel number. m = 0 to 2

Figure 60.60 A/D Conversion Processing Time (SAR Mode)

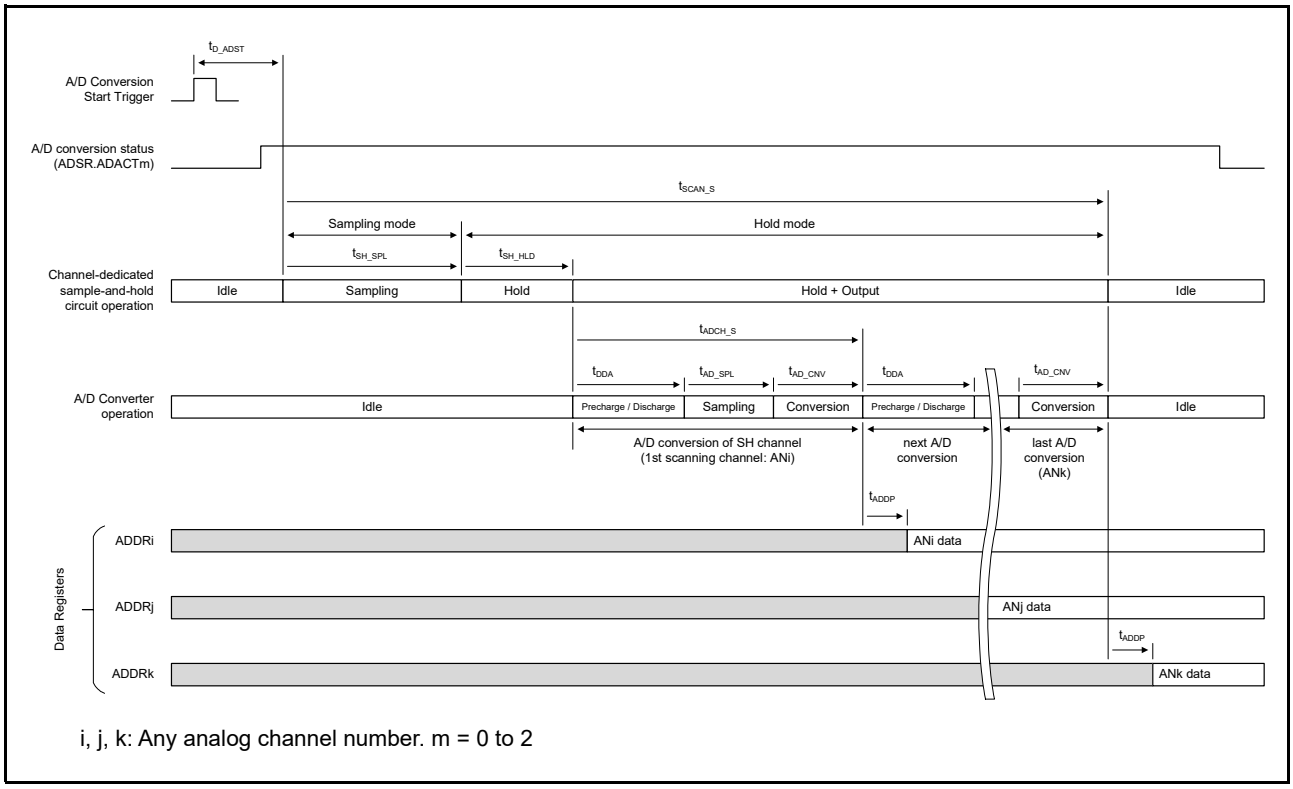


Figure 60.61 A/D Conversion Processing Time (SAR Mode with Channel-Dedicated Sample-and-Hold Circuit)

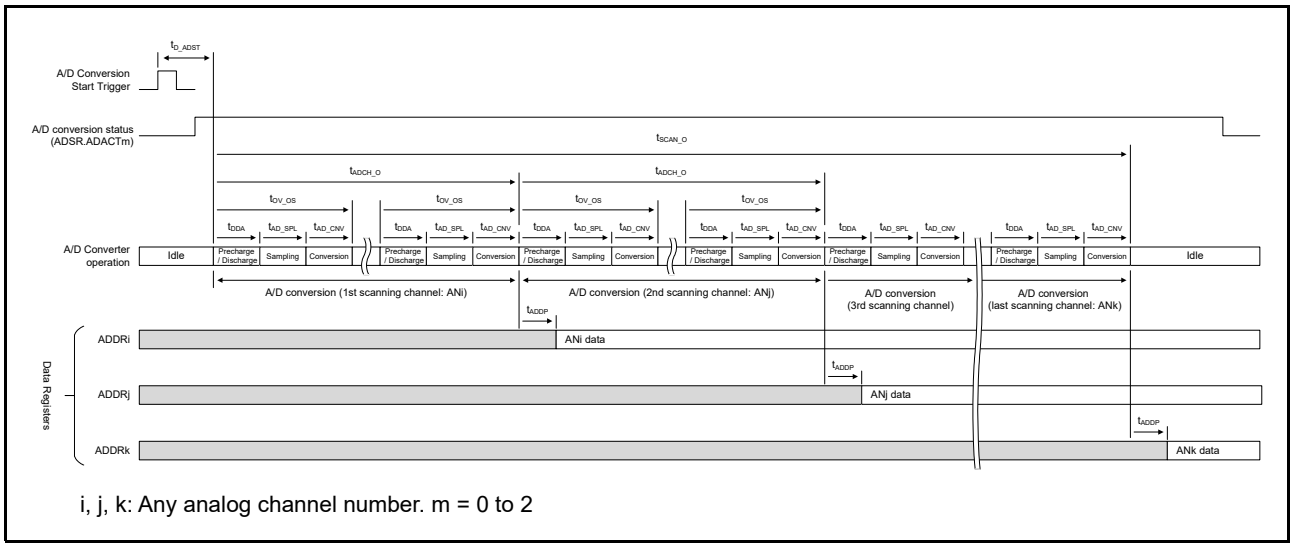
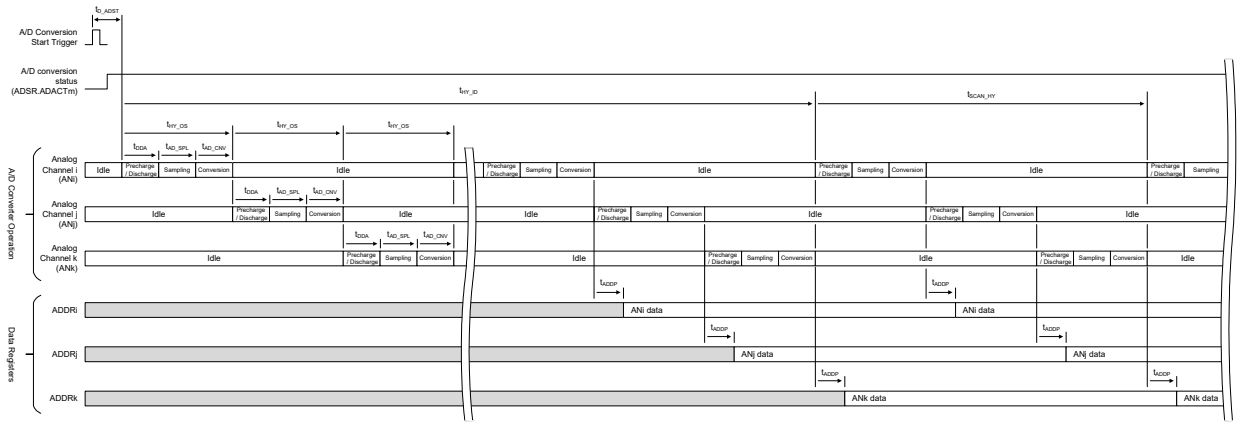
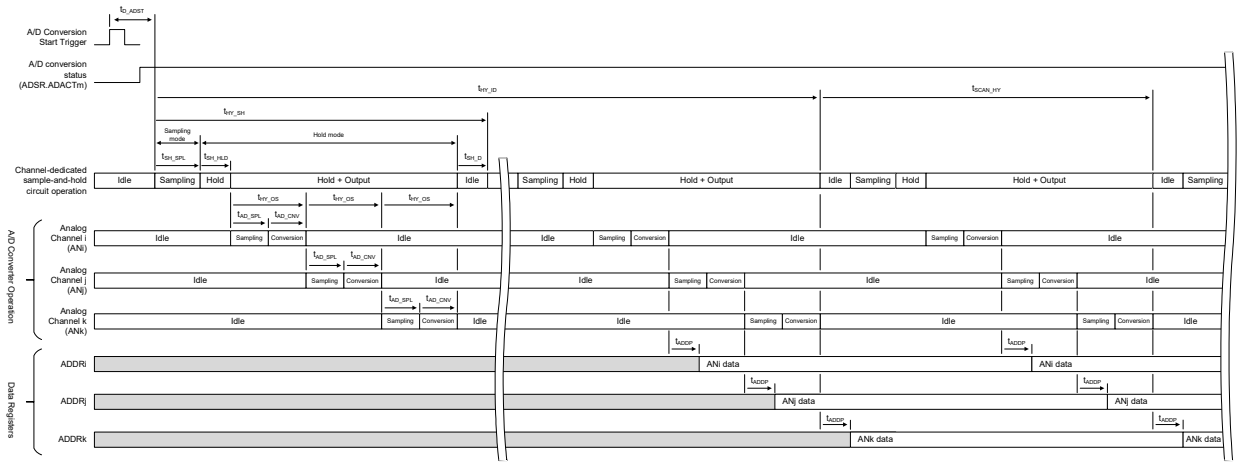


Figure 60.62 A/D Conversion Processing Time (Oversampling Mode)



i, j, k: Any analog channel number. m = 0 to 2

Figure 60.63 A/D Conversion Processing Time (Hybrid Mode)



i, j, k: Any analog channel number. m = 0 to 2

Figure 60.64 A/D Conversion Processing Time (Hybrid Mode with Channel-Dedicated Sample-and-Hold Circuit)

60.9.3 Scan End Processing Time

The scan end processing time and the forcibly stop processing time are shown in Table 60.70, Figure 60.65, Figure 60.66.

Table 60.70 Scan End Processing Time and the Forcibly Stop Processing Time

Item		Symbol	Processing Time
Scan end processing time	Until reflected in the status register*1	t_{ED1}	t_{ADDP}
	Until output the FIFO data read request interrupt	t_{ED2}	t_{ED1} + 2 PCLKA
Forcibly stop processing time	Forcibly stop trigger input processing time	t_{STOP_TRG}	[When ADCLK = PCLKA/1 is set]*2 • Number of access cycles to I/O registers*3 + 1 PCLKA [Other than above] • (Number of access cycles to I/O registers*3) + 1 PCLKA + (3 to 4 ADCLK)
	Wait time for synchronous operation	t_{STOP_SYNC}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) • 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) • 0 to ADSYCR.ADSYCYC[7:0] × ADCLK
	Forcibly stop processing time	t_{STOP}	[When ADCLK = PCLKA/1 is set]*2 (ADSYCR.ADSYDISm = 1) • 4 PCLKA (ADSYCR.ADSYDISm = 0) • 3 PCLKA [Other than above] (ADSYCR.ADSYDISm = 1) • 4 PCLKA + (2 to 3 PCLKA) (ADSYCR.ADSYDISm = 0) • 3 PCLKA + (2 to 3 PCLKA)

m = 0 to 2

Note 1. This is the time it takes for ADSCANENDSR.SCENDFn = 1 (n = 0 to 11) or until an end-of-scan interrupt is generated.

Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

Note 3. It is the number of access cycles to ADSTOPR register. For more details about the number of access cycles for I/O registers, see section 5, I/O Registers.

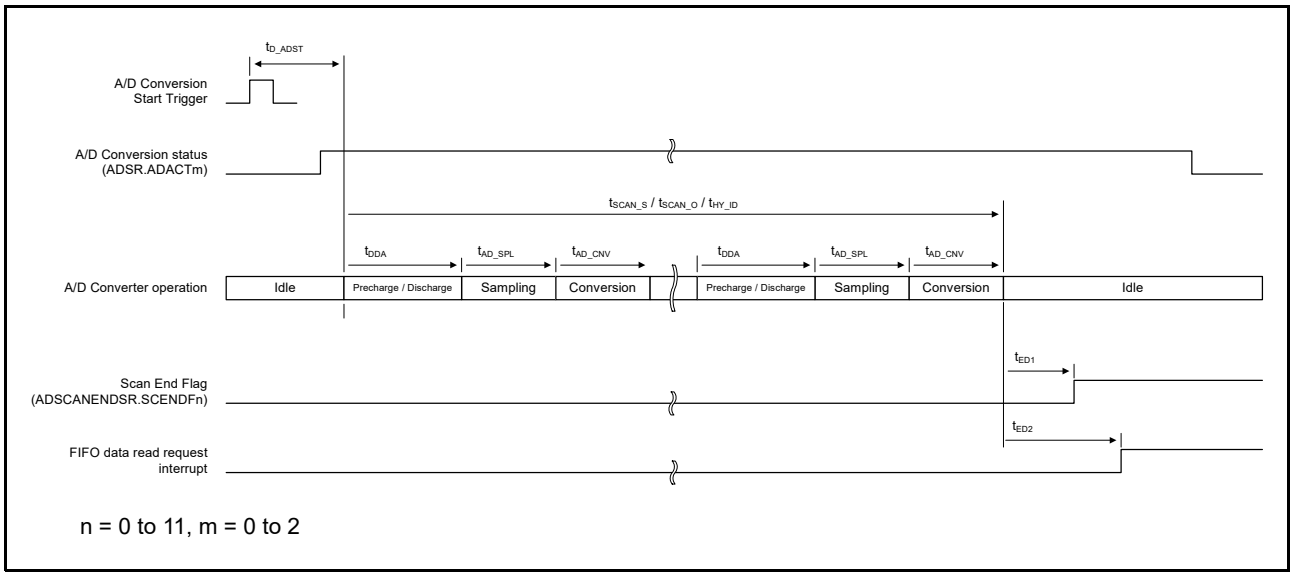


Figure 60.65 Scan End Processing Time (SAR/Oversampling/Hybrid Mode – Single Scan Mode)

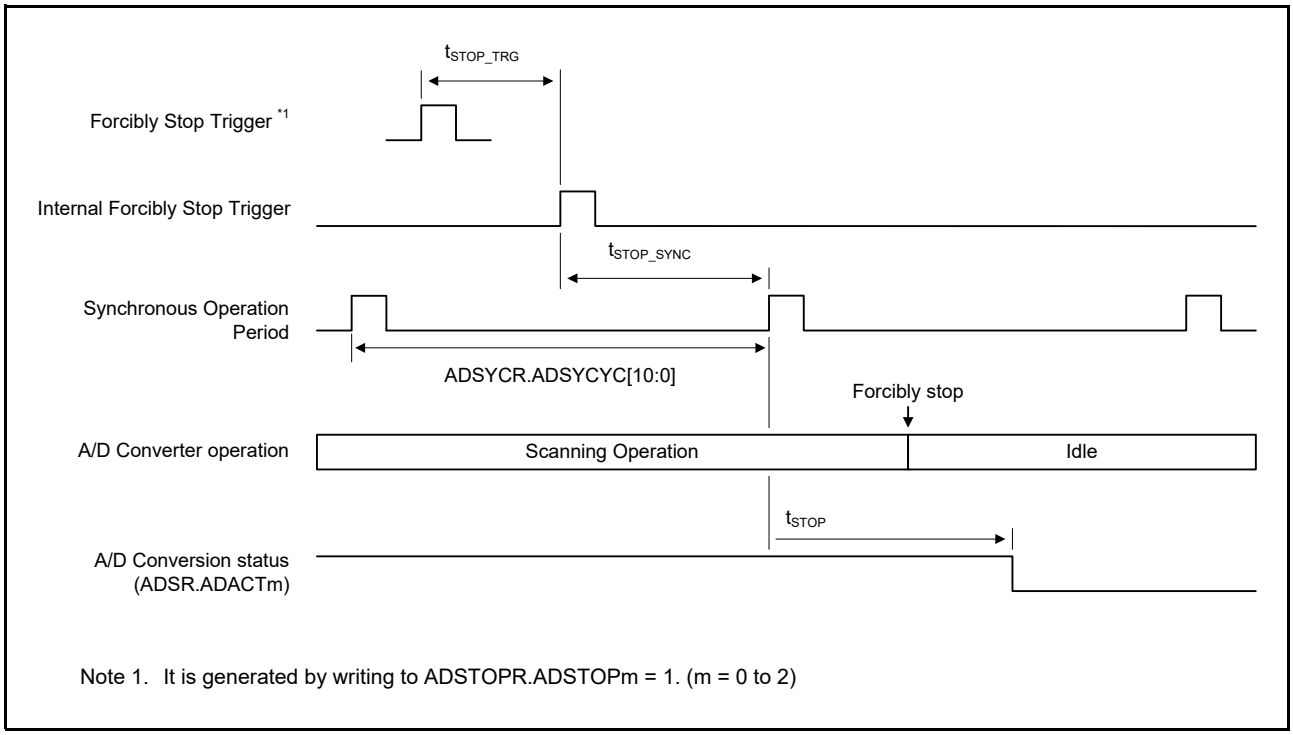


Figure 60.66 Forcibly Stop Processing Time (SAR/Oversampling/Hybrid Mode – Single Scan Mode)

60.10 Usage Notes

60.10.1 Prohibition of Changing the Operation Settings During A/D Conversion Operation

The registers related to the operation setting of the A/D converter should be set while all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0 to 2)). Changing (writing) of registers except for those listed below is prohibited during A/D conversion. If the operation setting is changed during A/D conversion, operation is not guaranteed.

[Registers that can be written while the A/D converter is operating]

- Status clear registers
  - Status clear registers related to A/D converter operation
    (ADERSCR, ADCALENDSCR, ADSCANENDSCR)
  - A/D conversion overflow status clear registers
    (ADOVFERSCR, ADOVFCHSCR0, ADOVFCHSCR1, ADOVFEXSCR)
  - Limiter clip status clear register
    (ADLIMGRSCR, ADLIMCHSCR0, ADLIMCHSCR1, ADLIMEXSCR)
  - Compare match status clear register
    (ADCMPTBSCR, ADCMPCHSCR0, ADCMPCHSCR1, ADCMPEXSCR)
  - FIFO error status clear register
    (ADFIFOERSCR)
- Software trigger register (ADSYSTR, ADSTRn (n = 0 to 11))
- A/D converter stop register (ADSTOPR)
- A/D converter start trigger enable register (ADTRGENR)*1

Note 1. Writing during operation is only permitted for disabling the trigger input (ADTRGENR.STTRGENn = 0 (n = 0 to 11)) in order to stop the A/D conversion. To avoid unintended operation, do not change the trigger input to Enable (ADTRGENR.STTRGENn = 1) during operation.

60.10.2 Usage Notes on Forced Stop of A/D Conversion

If you want to forcibly stop the A/D conversion operation, follow the procedure in section 60.5.4, Force Stops the A/D Conversion Operation.

60.10.3 Usage Notes on A/D Data Registers

When A/D conversion is performed multiple times on the same analog channel, A/D data register ADDRn (n = 0 to 35) or A/D extended analog data register ADEXDRm (m = 0 to 2, 4 to 6, 8, 9, 16 to 18) corresponding to that analog channel is overwritten at the later A/D conversion.
If you intend to keep the data for each A/D conversion for the same analog channel, perform one of the following methods.

- Read out data from registers at each scanning operation
  - At the end of scanning each scan group, read out the A/D conversion data of the analog channel that was subject to A/D conversion from ADDRn or ADEXDRm.
  - This method is effective if A/D conversion of the same analog channel is not performed more than once in the same scan group, and A/D conversion data can be read from the registers before the next scanning operation.
- Use FIFO function to hold data for each A/D conversion
  - By using FIFO, multiple A/D conversion data for the same analog channel can be kept.

– Read out the A/D conversion data before FIFO overflow.
– For the notes on using FIFOs, see section 60.4.10.6, Usage Notes on FIFO.

60.10.4 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ADC operation. The ADC is initially stopped after a reset. The registers become accessible on release from the module-stop state.
See section 60.10.6, Restrictions on Entering and Releasing the Low-Power States in connection with this usage note.

60.10.5 Settings for the Sleep Function

ADC0, ADC1, and ADC2 can individually be set to sleep operation mode using the A/D converter unit sleep control registers ADUSLPCRm (m = 0 to 2). Set these registers first after reset released in accordance with section 60.7.1, Initial Setup Procedure. For the units in the sleep state, do not execute self-calibration nor input A/D conversion start triggers. For the units in self-calibration or in A/D conversion, prevent the corresponding A/D converter unit sleep control registers m (m = 0 to 2) from setting the units to sleep operation mode.

60.10.6 Restrictions on Entering and Releasing the Low-Power States

(1) Module-stop state

Before entering the module-stop state, be sure to stop A/D conversion. When releasing the module-stop state again, make sure the shutdown time specified in the Electrical Characteristics has elapsed before releasing the module stop state. Then, follow to section 60.7.1, Initial Setup Procedure.

(2) Software standby mode

For entering and releasing the software standby mode, follow to section 60.7.4, Procedure for the A/D Converter Entering to and Returning from the Software Standby Mode.

(3) Low-power states

Configure the A/D conversion not to start during further entering a low-power state after entering the module-stop state.

60.10.7 Notes on Board Design

(1) Protection circuit

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, take the following measures.

- Insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0.
- Connect a protection circuit to protect the analog input pins.

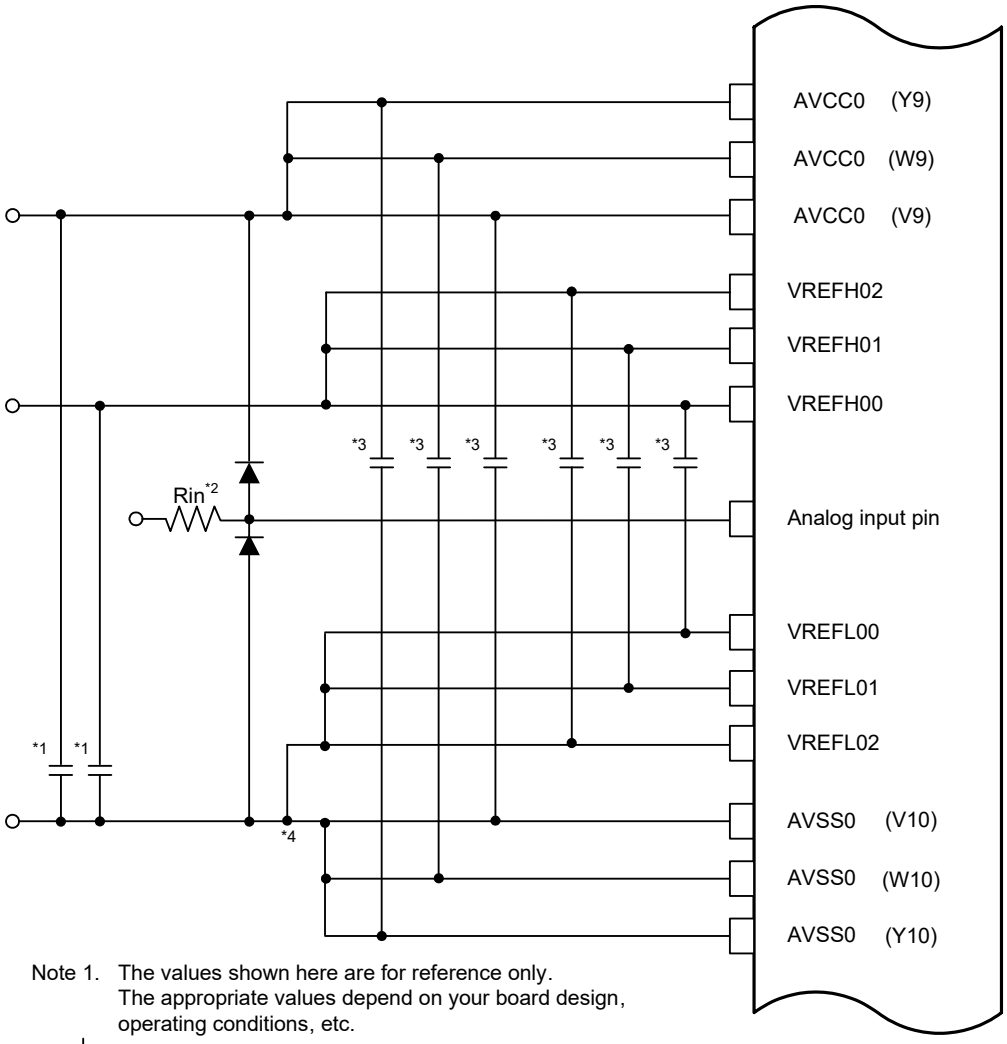
An example of a protection circuit is shown in Figure 60.67.

(2) Board design to ensure A/D conversion accuracy

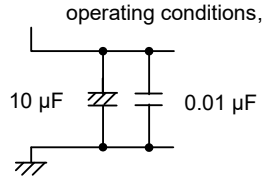
In order to ensure the accuracy of A/D conversion, design the board considering the following:

- Analog circuits and digital circuits should be separated from each other as far as possible.
- Analog circuit signal lines and digital circuit signal lines should not intersect or be placed near each other.
- The analog input, analog reference power supply (VREFH0), analog reference ground (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0).

- The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).
- Place a capacitor for noise filters between AVCC0 pin and AVSS0 pin and between VREFH0 pin and VREFL0 pin close to the pin and connect them. Also, connect AVSS0 and VREFL0 pins to the analog ground on the board as close to the pins as possible. Figure 60.67 and Figure 60.68 show examples of connections. As shown in Figure 60.67, connect individual capacitors to each pin on 320-pin products.

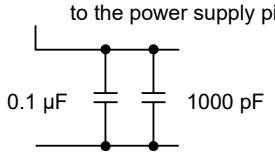


Note 1. The values shown here are for reference only.
The appropriate values depend on your board design, operating conditions, etc.



Note 2. Rin: Signal source impedance

Note 3. To ensure the better A/D conversion accuracy, these capacitors must be placed as close as possible to the appropriate pins. Place the 1000-pF capacitor closer to the power supply pin than the 0.1-µF capacitor.



Note 4. Connect to the analog ground of the board at the position as close as possible to the capacitor for noise filtering that ensure the A/D conversion accuracy.

Figure 60.67 Example of Protection Circuit and Noise Prevention (320-pin products)

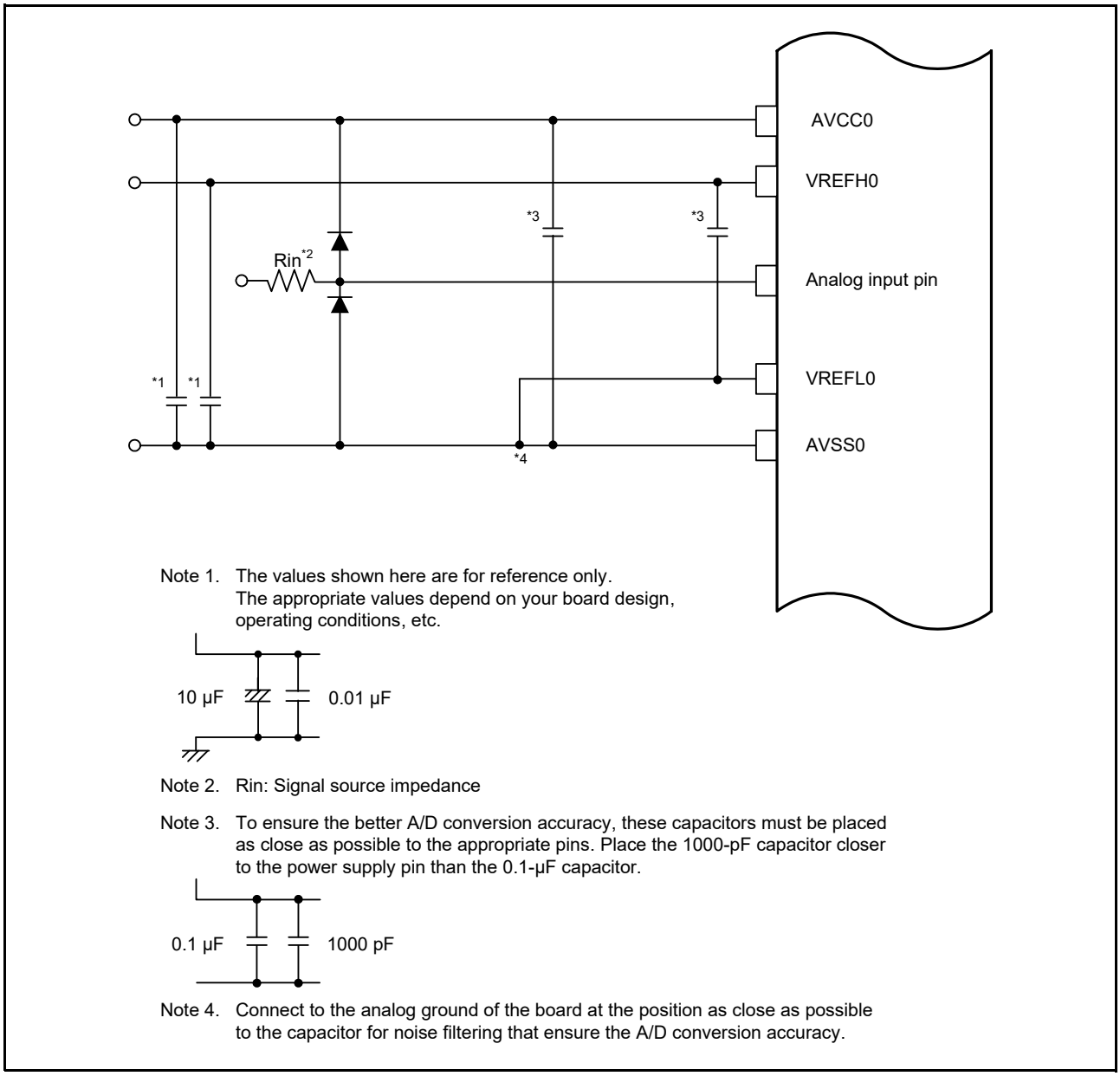


Figure 60.68 Example of Protection Circuit and Noise Prevention (Products Other than 320-pin)

60.10.8 Notes on Synchronous Operation

Synchronous operation function is disabled in the initial state after reset released. If using synchronous operation function with an external signal (the synchronization signal from the GPT), follow the restrictions described in section 60.3.20, Synchronous Operation.

60.10.9 Notes on Channel-Dedicated Sample-and-Hold Circuit

When using channel-dedicated sample-and-hold circuit, follow the restrictions. For details, see section 60.3.17.4, Restrictions on Channel-Dedicated Sample-and-Hold Circuit.

60.10.10 Restrictions on Analog Channel Shared Among Multiple A/D Converters

To avoid degradation of the accuracy of the A/D conversion result, A/D conversion of the same analog channel (same analog signal source) from both ADC1 and ADC2 is prohibited, except for the self-diagnosis channel.
If this restriction is violated, the A/D conversion results are not guaranteed because the A/D conversion accuracy of the target analog channel may deteriorate significantly.

60.10.11 Notes on A/D Conversion Start Trigger

(1) Notes on A/D conversion trigger for the same scan group

In modes other than same channel continuous scan mode or background continuous scan mode, A/D conversion start triggers for the same scan group are not accepted until scanning operation of the scan group is completed (in this case, A/D conversion start triggers are ignored).
An A/D conversion trigger for the same scan group should be input after the scanning operation of its scan group is completed (ADSCANENDSR.SCENDFn = 1 (n = 0 to 11)) and after a 6 PCLKA clock cycles or more has elapsed.

(2) Prohibition on calibration and A/D conversion

It is prohibited to start calibration and A/D conversion in the ADC unit during sleep operation. In addition, it is prohibited to set the ADC unit to Sleep mode during calibration and A/D conversion operations.

60.10.12 Notes on Self-Calibration

Observe the restrictions on self-calibration function. For details, see section 60.3.9, Self-Calibration.

60.10.13 Notes on Group Priority Operation

Observe the restrictions when using group priority operation. For details, see section 60.3.19, Group Priority Operation.

60.10.14 Restrictions on SAR Mode

(1) Notes on resolution and accuracy

In SAR mode, the A/D converter functions as a 12-bit A/D converter. Although the data output from the A/D converter is internally extended to 16-bit data length for calculation processing, even if 14-bit or 16-bit length data format is selected, the resolution and accuracy exceeding 12 bits are not guaranteed for that A/D conversion data. Select 14-bit or 16-bit length data format in SAR mode if you want to increase the computational resolution, such as when using A/D-converted value addition/averaging function or User's Gain/User's Offset function.

(2) Restrictions on digital filter function

The digital filter is prohibited in SAR mode. Operation is not guaranteed when the digital filter function is enabled.

60.10.15 Restrictions on Oversampling Mode

(1) Restrictions on digital filter function

In Oversampling mode, the use of the digital filter function is mandatory. If A/D conversion is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.

(2) Restrictions on trigger interval in same channel continuous scan mode

In oversampling mode with same channel continuous scan mode, A/D conversion start trigger should be input after the following interval or more.
[Required Interval for A/D Conversion Start Trigger Input]

- When trigger delay function is not used, or software trigger: 8 ADCLK, or more
- When trigger delay function is used: (8 + (the setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK, or more

Note: m = 0 to 5, n = 0 to 11

If this restriction is violated, the A/D conversion start trigger is not accepted and is ignored.

60.10.16 Restrictions on Hybrid Mode

In hybrid mode, observe the following restrictions.

(1) Prohibition of scanning operation with only one channel

In hybrid mode, be sure to assign two or more virtual channels to one scan group (up to 4 virtual channels). Scanning operation with only one virtual channel assigned to one scan group is prohibited. If this restriction is violated, operation is not guaranteed.

(2) Restrictions on virtual channel configuration

In hybrid mode, do not assign the same analog channel to multiple virtual channels that are assigned to the same scan group. If this restriction is violated, A/D conversion result is not guaranteed.

(3) Restrictions on digital filter configuration

In hybrid mode, the use of the digital filter function is mandatory. If A/D conversion in hybrid mode is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.
In addition, since multiple digital filters are used simultaneously in hybrid mode, the digital filters used for virtual channels in the same scan group must be set to be exclusive. The digital filters selected by the ADDOPCRAn.DFSEL[2:0] (n = 0 to 49) bits must be set to be exclusive among virtual channels in the same scan group. If the same digital filter is selected from multiple virtual channels in the same scan group, operation is not guaranteed.

(4) Restrictions on A/D-converted value addition/averaging function

When using A/D-converted value addition/averaging function in hybrid mode, set the number of addition/averaging times for all virtual channels in the same scan group to be the same. If this restriction is violated, operation is not guaranteed.

(5) Restrictions on trigger interval in background continuous scan mode

In Hybrid mode – background continuous scan mode, A/D conversion start trigger should be input after the following interval or more.
[Required interval for A/D conversion start trigger input]

- When trigger delay function is not used, or software trigger: 8 ADCLK, or more
- When trigger delay function is used: (8 + (the setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK, or more

Note: m = 0 to 5, n = 0 to 11
If this restriction is violated, the A/D conversion start trigger is not accepted and is ignored.

(6) Restrictions on fixed channels continuous scan mode

In hybrid mode with fixed channels continuous scan mode, observe the constraints on the settings. For details, see section 60.3.7.5, Fixed Channels Continuous Scan Mode.

60.10.17 Restrictions on the High Accuracy Mode

The high accuracy mode can be set using the A/D converter conversion accuracy mode register. In the high accuracy mode, observe the following restrictions. If the following restrictions are violated, operations are not guaranteed.

- A/D conversion modes with which the high accuracy mode can be set
  - SAR mode
  - Hybrid mode
- Restrictions on the sampling time of the analog channel of the ADCm (m = 0 to 2) for which the high accuracy mode is set
  - Set the value no less than 1.6 times the successive approximation time.
  - ADSSTRx.SSTy[9:0] ≥ ADCNVSTR.CSTm [5:0] ×1.6 (x = 0 to 9, y = 0 to 11, m = 0 to 2)
  - Set the number of states no less than four.
  - ADSSTRx.SSTy[9:0] ≥ 04h (x = 0 to 9, y = 0 to 11)
- When the high accuracy mode is set, the synchronous operation period should be set as follows.
  - ADSYCR.ADSYCYC[7:0] × i = ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0]
    (x = 0 to 7, y = 0 to 15, m = 0 to 2, i: 1, 2 × n (n = 1, 2, 3…))

The use of the high accuracy mode is recommended for the following function.

- Disconnection detection assist function

60.10.18 Restrictions on Composite Compare Match

The selection of a common compare match table for multiple ADCs is prohibited for composite compare match. If selected, the composite compare match interrupt is asserted when the condition is satisfied at any ADC. When the condition is not satisfied for all ADCs, the composite compare unmatch interrupt is asserted.

60.10.19 Estimating the Sampling Error and Sampling Time

This section describes how to estimate the sampling error and sampling time.
The sampling time required in A/D conversion for externally input analog signal is determined by the time for charging the sampling capacitance in A/D converter. The simplified model of the circuit is shown in Figure 60.69.

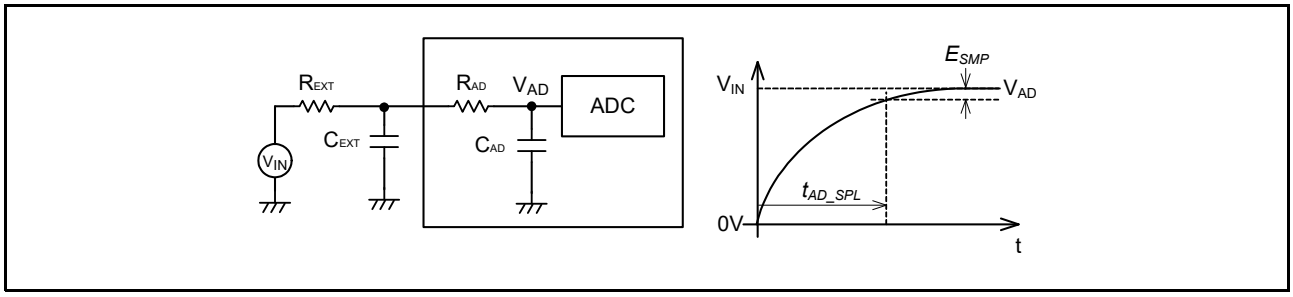


Figure 60.69 Simplified Model of Sample-and-Hold Circuit for A/D Converter

The sampling error can be roughly estimated by the following equation. Set the sampling time (t_{AD_SPL}) to ensure that the sampling error (E_{SMP}) falls within the expected range. The range of E_{SMP} is 0 to 1; the minimum value (0) indicates that there is no error and the error becomes the maximum (1) when the input voltage does not converge at all and V_{in} = VREFH0.

[Estimation Equation for Sampling Error]

$$E_{SMP}(t) = \frac{\frac{k_1 C_{AD}}{C_{EXT} + k_1 C_{AD}}}{1 - \frac{C_{EXT}}{C_{EXT} + k_1 C_{AD}} e^{\frac{-t_{SCAN_*} + t_{AD_SPL}/k_2 \left(1 - \frac{\tau_{EXT}}{\tau}\right)}{\tau_{EXT}}}} e^{\frac{-t_{AD_SPL}/k_2}{\tau}}$$

$$\tau = R_{EXT}(C_{EXT} + k_1 C_{AD}) + R_{AD} k_1 C_{AD}$$

$$\tau_{EXT} = R_{EXT} C_{EXT}$$

E_{SMP}: Estimated value of sampling error
t_{AD_SPL}: Sampling time
C_{EXT}: External capacitance (pin capacitance + PCB parasitic capacitance)
C_{AD}: Internal sampling capacitance
R_{EXT}: External input signal source impedance
R_{AD}: Internal resistance
k_1, k_2: Correction coefficients according to operation mode
t_{SCAN_*}: Channel scan interval (the sum of the scan conversion time) (* = S, O, HY)

Figure 60.70 shows the definition of the channel scan interval.

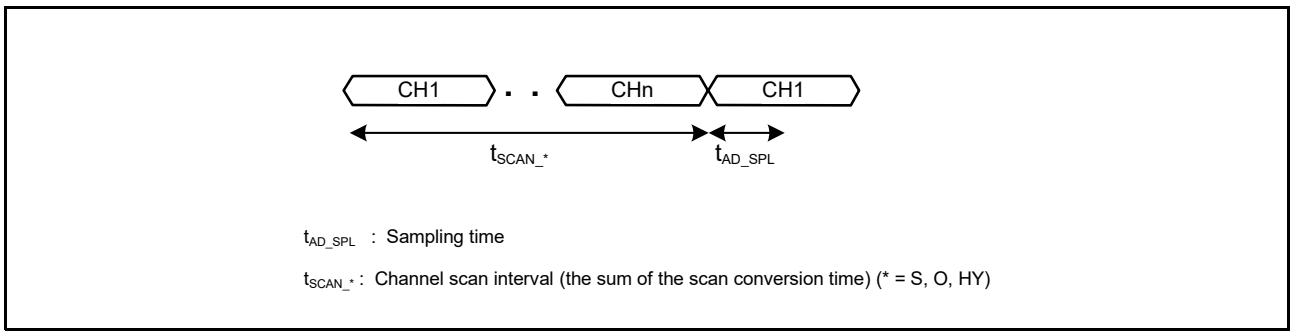


Figure 60.70 Definition of the Channel Scan Interval

The typical values for the parameters are as follows. These are reference values.

[Internal Sampling Capacitance (C_{AD}): 7 pF]

Table 60.71 k_1, k_2

Power Supply	Scan Mode	k_1 High-Speed Channels (AN000 to AN015) Low-Speed Channels (AN016 to AN035)	k_2
AVCC0: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0: 2.7 V to AVCC	SAR mode (normal mode)	1.3	1.0
	SAR mode (high accuracy mode)	1.3	1.8
	Oversampling mode	1.0	1.0
	Hybrid mode	1.3	1.8
AVCC0: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0: 1.62 V to AVCC	SAR mode (normal mode)	1.2	1.0
	SAR mode (high accuracy mode)	1.2	1.8
	Oversampling mode	1.0	1.0
	Hybrid mode	1.2	1.8

Table 60.72 R_{AD}

Power Supply	R_{AD}(Ω)	
	High-Speed Channels (AN000 to AN015)	Low-Speed Channels (AN016 to AN035)
AVCC0: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0: 2.7 V to AVCC	700	3000
AVCC0: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0: 1.62 V to AVCC	1500	5000

[Usage Notes]

- Set the sampling time to a value greater than or equal to the minimum value specified in the electrical characteristics.
- The equation described in this section is simplified in view of general use case. It does not guarantee the accurate sampling time. You should only use it to roughly estimate the sampling time.

60.10.20 Restriction on A/D Conversion Overflow in Single-Ended Input Mode

To use the A/D conversion overflow to determine whether the input to the A/D converter is lower than VREFL0 in single-ended input mode, make the following settings to disable both the user offset adjustment function and A/D-converted value addition or average function.

1. Disable the user offset adjustment function: Set the ADUOFTRn.UOFSET[15:0] bits to 0000h.
2. Disable the A/D-converted value addition or average function: Set the ADDOPCRBn.AVEMD[1:0] bits to 00b.

Note that if the user offset adjustment function is required, the limiter clip interrupt can be used instead of the overflow interrupt to detect an input lower than VREFL0 while the user offset adjustment function is enabled but the A/D-converted value addition or average function is disabled. In such a case, set the lower-side limit of the limiter clip function (ADLIMTRn.LIML[15:0] bits) to the code value corresponding to an input voltage that is below VREFL0 but cannot be detected as an overflow due to the user offset setting.

61. 12-Bit D/A Converter (DAC12R)

61.1 Overview

This MCU includes two channels of 12-bit D/A converters.
Table 61.1 lists the specifications of the 12-bit D/A converter, Figure 61.1 shows a block diagram of the 12-bit D/A converter.

Table 61.1 Specifications of 12-Bit D/A Converter

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Output Control	Conversion result can be output to external pin, internal peripheral module (A/D converter), or both.
Event link function (input)	D/A conversion can be started in response to the input of an event signal.
Low power consumption function	Module stop state can be set.
Protect Zone filter	Security attribution can be set for each channel*1.

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

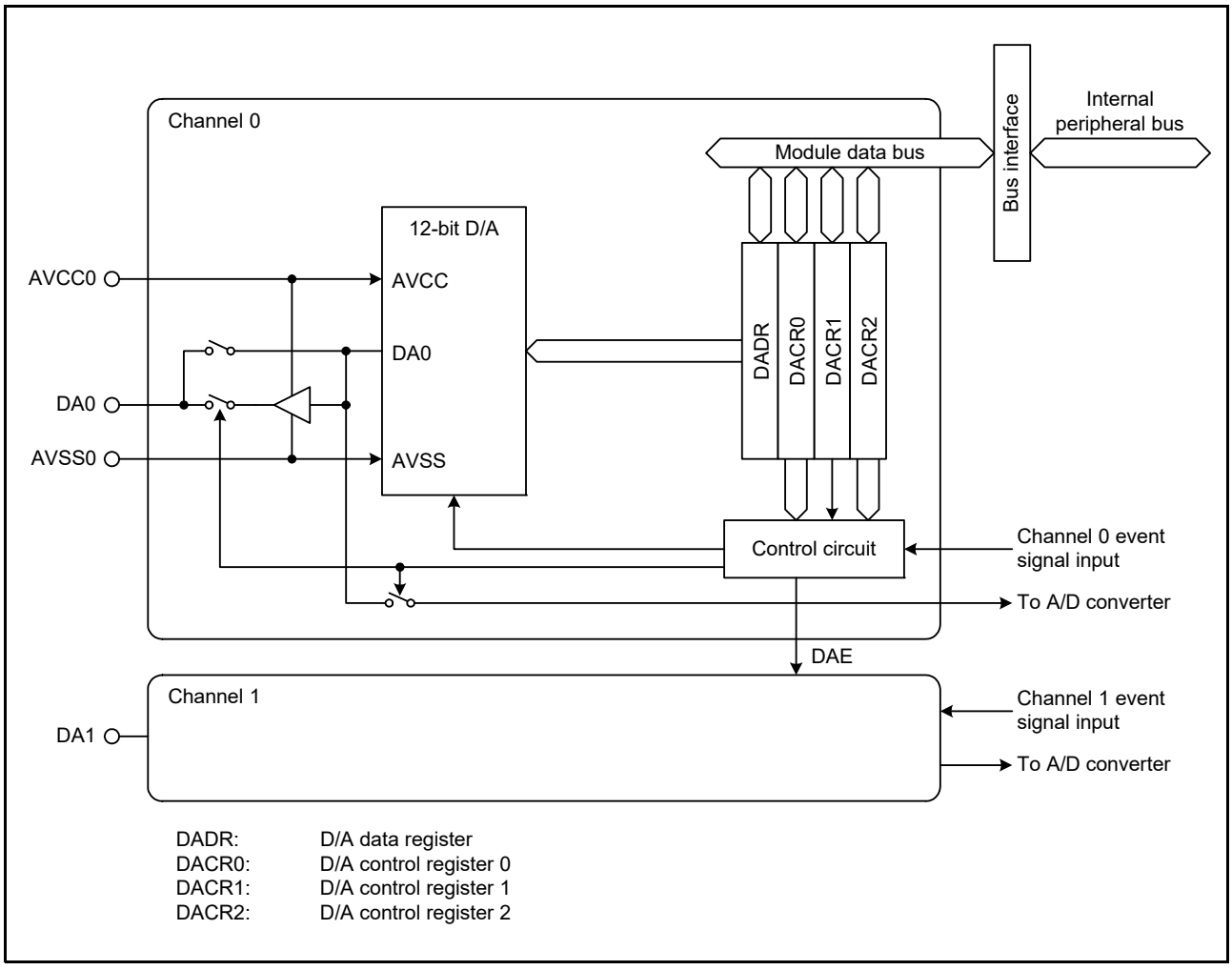


Figure 61.1 Block Diagram of 12-Bit D/A Converter

Table 61.2 lists the pin configuration of the 12-bit D/A converter.

Table 61.2 Pin Configuration of 12-Bit D/A Converter

Pin Name	I/O	Function
AVCC0	Input	Analog power supply pin and reference voltage input pin
AVSS0	Input	Analog ground pin and reference ground pin
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

61.2 Register Descriptions

61.2.1 D/A Data Register (DADR)

Address(es): DAC12R0.DADR 8783 3000h, DAC12R1.DADR 8783 3100h, DAC12R0_NS.DADR 0783 3000h, DAC12R1_NS.DADR 0783 3100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DADR register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Access this register in 16-bit units.

When the DACR1.DPSEL bit is 0, the upper 4 bits are reserved. When the DPSEL bit is 1, the lower 4 bits are reserved. These reserved bits are read as 0. The write value should be 0.

When using the output buffer amplifier, the range of setting values is 0E0h to F1Fh when the DACR2.OFSSEL bit is 0, and 1F8h to E07h when the OFSSEL bit is 1.

When analog output is enabled, the value of the DADR register is converted and output to the analog output pin.

61.2.2 D/A Control Register 0 (DACR0)

Address(es): DAC12R0.DACR0 8783 3004h, DAC12R1.DACR0 8783 3104h, DAC12R0_NS.DACR0 0783 3004h, DAC12R1_NS.DACR0 0783 3104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAOUTDIS	DAOUTEN	—	DAOE	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DAE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DACEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DACEN	D/A Conversion Enable	0: D/A conversion is disabled. 1: D/A conversion is enabled.	R/W
b14 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	DAE	Collective D/A Conversion Enable*1, *2	0: D/A conversion of channels 0 and 1 is controlled by the DACEN bit for each channel. 1: D/A conversion of channels 0 and 1 is enabled collectively.	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	DAOE	D/A Output Enable	0: Analog output from the DAn pin is disabled (n = 0, 1). 1: Analog output from the DAn pin is enabled.	R/W
b29	—	Reserved	This bit is read as 0. The write value should be 0.	R
b30	DAOUTEN	DAn Pin Output Enable*3	0: Analog output from the DAn pin follows the setting of the DAOUTDIS bit. 1: Analog output from the DAn pin is enabled regardless of the setting of the DAOUTDIS bit.	R/W
b31	DAOUTDIS	DAn Pin Output Disable*3	0: Analog output from the DAn pin is enabled and analog output to internal peripheral modules is disabled. 1: Analog output from the DAn pin is disabled and analog output to internal peripheral modules is enabled.	R/W

Note 1. This bit is reserved on channel 1. This bit is read as 0. The write value should be 0.
Note 2. When using this bit, ensure that the security attributes for channels 0 and 1 are the same.
Note 3. This bit should be rewritten only when the DACEN bit for the same channel and DAE bit for channel 0 are both 0.

DACEN Bit (D/A Conversion Enable)

The DACEN bit controls the D/A conversion and output of the results of conversion. This bit is only valid when the DAE bit for channel 0 is set to 0.

When this bit is set to 0, D/A conversion is disabled and D/A output is also disabled. When this bit is set to 1, D/A conversion is performed, but D/A output is controlled by the DAOE, DAOUTEN, and DAOUTDIS bits.

The DACEN bit can also be set to 1 using the event link function.

DAE Bit (Collective D/A Conversion Enable)

This bit is used to control the D/A conversion for both channel 0 and channel 1 collectively. To control the D/A conversion for each channel individually, set the DAE bit for channel 0 to 0 and use the DACEN bit instead.

Table 61.3 Operation of D/A Converters

Bit Settings			Operation of D/A converters	
Channel 0		Channel 1		
DAE	DACEN	DACEN	Channel 0	Channel 1
0	0	0	Conversion stops. Analog output disabled (Hi-Z).	Conversion stops. Analog output disabled (Hi-Z).
0	1	0	D/A conversion is performed.	Conversion stops. Analog output disabled (Hi-Z).
0	0	1	Conversion stops. Analog output disabled (Hi-Z).	D/A conversion is performed.
0	1	1	D/A conversion is performed.	D/A conversion is performed.
1	x	x	D/A conversion is performed.	D/A conversion is performed.

DAOE Bit (D/A Output Enable)

This bit is used to enable analog output from the DAn pin (n = 0, 1). This bit does not affect analog output to internal peripheral modules.

DAOUTEN Bit (DAn Pin Output Enable)

This bit is used to enable analog output from the DAn pin even when the DAOUTDIS bit is set to 1 (n = 0, 1). Using this bit allows analog output from the DAn pins to be enabled while keeping analog output to internal peripheral modules enabled.

DAOUTDIS Bit (DAn Pin Output Disable)

This bit is used to disable analog output from the DAn pin (n = 0, 1). Setting this bit to 1 enables analog output to the internal peripheral modules.

Table 61.4 Analog Voltage Output Destination Specification (when the DACEN or DAE bit is set to 1)

Bit Settings				Analog Outputs	
DAOUTDIS	DAOUTEN	DAOE	DACR2.DAAMP	DAn Pin	Internal Peripheral Modules
0	x	0	x	Output disabled (Hi-Z)	Output disabled
0	x	1	0	Unbuffered output	Output disabled
0	x	1	1	Buffered output	Output disabled
1	0	x	x	Output disabled (Hi-Z)	Unbuffered output
1	1	0	x	Output disabled (Hi-Z)	Unbuffered output
1	1	1	x	Unbuffered output	Unbuffered output

61.2.3 D/A Control Register 1 (DACR1)

Address(es): DAC12R0.DACR1 8783 3008h, DAC12R1.DACR1 8783 3108h, DAC12R0_NS.DACR1 0783 3008h, DAC12R1_NS.DACR1 0783 3108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DPSEL	Format Select	0: Data is right-justified. 1: Data is left-justified.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

DPSEL Bit (Format Select)

This bit specifies the location of valid data in the DADR register.

Write data to bits 11 to 0, when this bit is set to 0. Write data to bits 15 to 4, when this bit is set to 1.

61.2.4 D/A Control Register 2 (DACR2)

Address(es): DAC12R0.DACR2 8783 300Ch, DAC12R1.DACR2 8783 310Ch, DAC12R0_NS.DACR2 0783 300Ch, DAC12R1_NS.DACR2 0783 310Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAAMP	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OFSSEL	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	OFSSEL	Offset Calibration Value Select *1	0: Offset calibration value for normal voltage (AVCC0 ≥ 2.7 V) 1: Offset calibration value for low voltage (AVCC0 < 2.7 V)	R/W
b30 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	DAAMP	Output Buffer Amplifier Control *2	0: Output buffer amplifier is not used. 1: Output buffer amplifier is used.	R/W

Note 1. Set the same value in this bit for all channels.
Note 2. This bit should be rewritten only when the DACEN bit for the same channel and DAE bit for channel 0 are both set to 0.

OFSSEL Bit (Offset Calibration Value Select)

This bit is used to select the optimum offset calibration value based on the operating voltage of the D/A converter. When using AVCC0 below 2.7 V, set this bit to 1.

DAAMP Bit (Output Buffer Amplifier Control)

This bit selects whether the D/A converter output is output to the DAn pin via an output buffer amplifier (n = 0, 1). It is only valid when the DACR0.DAOUTDIS bit is 0. When the DAOUTDIS bit is 1, the setting of this bit is ignored and the output buffer amplifier is not used. Refer to Table 61.4 for details.

61.3 Operation

The 12-bit D/A converter features independent circuits for each channel, allowing the D/A conversion enable/disable status and the output destinations of the D/A conversion result to be set per channel. Channels 0 and 1 can be enabled or disabled for D/A conversion simultaneously. This section explains operation using channel 0 as an example.

61.3.1 D/A Conversion

When the DACR0.DAE or DACR0.DACEN bit is 1, D/A conversion is enabled and the results of conversion are output from the destination specified by the DAOE, DAOUTEN, and DAOUTDIS bits in the DACR0 register. Figure 61.2 shows the timing of this operation.
An example of the setting procedure for D/A conversion is shown below.

When Outputting from the DA0 Pin Bypassing the Output Buffer Amplifier

(1) Set the DACR0.DAOE bit to 1, the DAOUTDIS bit to 0, and the DACR2.DAAMP bit to 0.
(2) Set the data format in the DACR1.DPSEL bit.
(3) Set the DADR register to 0000h.
(4) Setting the DACR0.DACEN bit to 1 initiates D/A conversion, and after the recovery time ($t_{DSLPUP1}$) has elapsed, the voltage at the DA0 pin stabilizes at 0 V. Note that if a value other than 0000h is set in step (3), additional conversion time (t_{DCONV1}) is required before the output stabilizes.
(5) Updating the DADR register immediately initiates re-conversion, and after the conversion time (t_{DCONV1}) has elapsed, the output at the DA0 pin stabilizes at the voltage corresponding to the set value. The output of the D/A conversion results is held at this level until the DADR register is updated or the DACEN bit or DAOE bit is set to 0. The output voltage is expressed by the following formula:

$$\frac{\text{Value of the DADR register}}{4096} \times \text{AVCC0}$$

(6) Setting the DACEN bit to 0 stops D/A conversion and disables analog output.

When Outputting from the DA0 Pin Via the Output Buffer Amplifier

(1) Set the DACR0.DAOUTDIS bit to 0 and the DACR2.DAAMP bit to 1.
(2) Set the data format in the DACR1.DPSEL bit.
(3) When the DACR2.OFSSEL bit is 0, set the DADR register to 00E0h (for right-justified format) or 0E00h (for left-justified format). When the OFSSEL bit is 1, set the DADR register to 01F8h (for right-justified format) or 1F80h (for left-justified format).
(4) Setting the DACR0.DACEN bit to 1 initiates D/A conversion.
(5) After the output buffer amplifier startup time (t_{DISOUT}) has elapsed, setting the DACR0.DAOE bit to 1 enables the output from the DA0 pin, and after the recovery time ($t_{DSLPUP2}$) has elapsed, the voltage at the DA0 pin stabilizes at approximately 0 V. Note that if a value other than the above is set in step (3), additional conversion time (t_{DCONV2}) is required before the output stabilizes.
(6) Updating the DADR register immediately initiates re-conversion, and after the conversion time (t_{DCONV2}) has elapsed, the output at the DA0 pin stabilizes at the voltage corresponding to the set value. The output of the D/A conversion results is held at this level until the DADR register is updated or the DACEN bit or DAOE bit is set to 0. The output voltage is expressed by the following formula:

$$\frac{\text{Value of the DADR register}}{4096} \times \text{AVCC0}$$

Note that when using the output buffer amplifier, the lower and upper limits of the output voltage do not reach 0 V and AVCC0, respectively. For details, refer to section 67, Electrical Characteristics.
(7) Setting the DACEN bit to 0 stops D/A conversion and disables analog output.

When Outputting to Internal Peripheral Modules

(1) Set the DACR0.DAOUTDIS bit to 1, and the DACR2.DAAMP bit to 0.

(2) Set the data format in the DACR1.DPSEL bit.

(3) Set the DADR register to 0000h.

(4) Setting the DACR0.DACEN bit to 1 initiates D/A conversion, and after the recovery time ($t_{DSLPUP1}$) has elapsed, the output voltage stabilizes at 0 V. Note that if a value other than 0000h is set in step (3), additional conversion time (t_{DCONV3}) is required before the output stabilizes.

(5) Updating the DADR register immediately initiates re-conversion, and after the conversion time (t_{DCONV3}) has elapsed, the output stabilizes at the voltage corresponding to the set value. The output of the D/A conversion results is held at this level until the DADR register is updated or the DACEN bit is set to 0. The output voltage is expressed by the following formula:

$$\frac{\text{Value of the DADR register}}{4096} \times \text{AVCC0}$$

(6) Setting the DACEN bit to 0 stops D/A conversion and disables analog output.

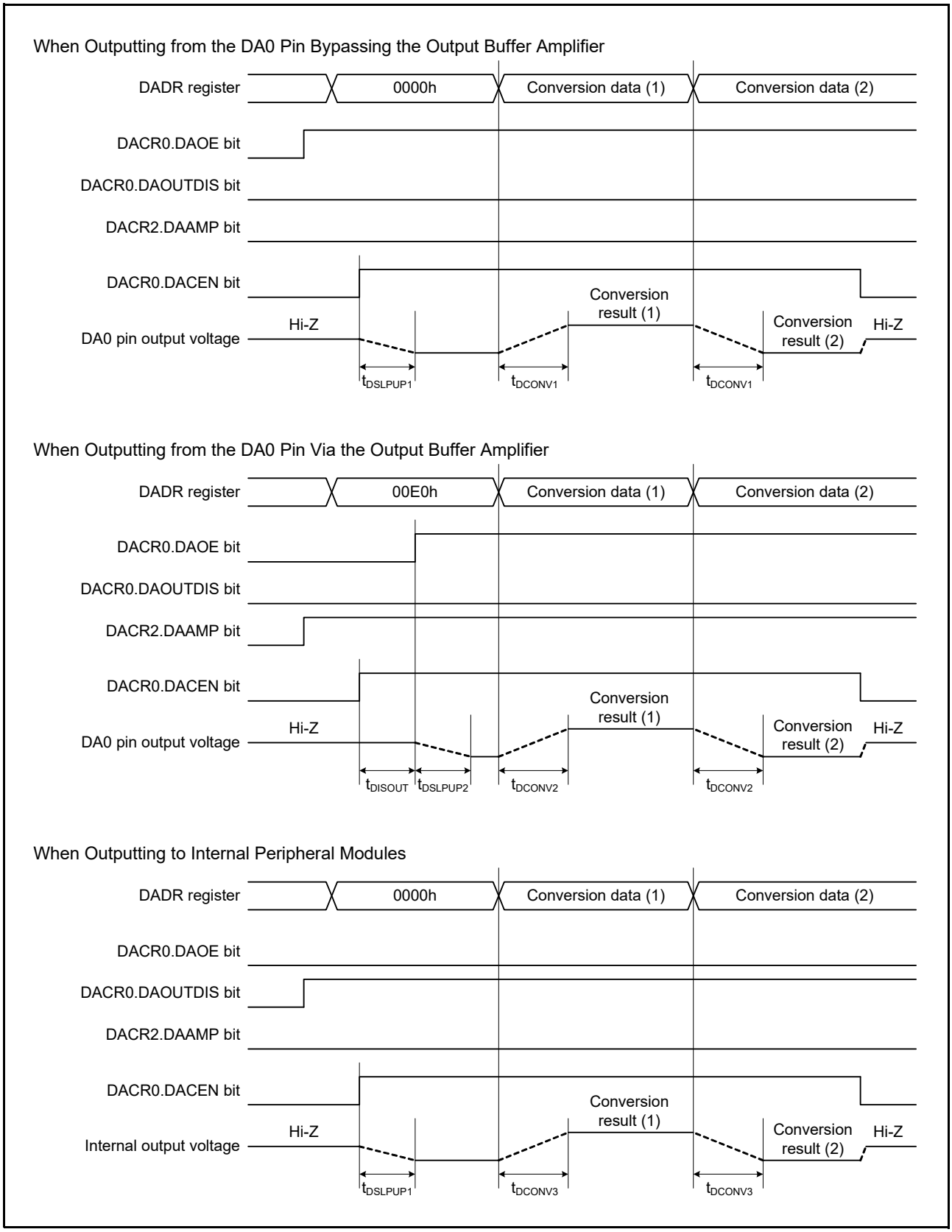


Figure 61.2 Example of 12-Bit D/A Converter Operation

61.4 Linked Operation with the Use of the Event Link Function

61.4.1 Action in Response to Receiving an Event Signal

The example of setting procedure for using event link operation is described below.
The event signal can only control the DACR0.DACEN bit; it cannot control the DAOE bit. Therefore, depending on the data for D/A conversion, an undefined voltage may be output before the D/A conversion output voltage has settled.

(1) Set the DACR0.DACEN bit to 0, DAE bit to 0, DAOE bit to 1, DAOUTDIS bit to 0, and the DACR2.DAAMP bit to 0.
(2) Set the data for D/A conversion in the DADR register.
(3) Set the number for the event signal to be linked in the ELSR46 register (DAC12R0) or ELSR47 register (DAC12R1) of the ELC.
(4) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(5) Set the event output source module and start it. After the event is output from the module, the DACR0.DACEN bit becomes 1, and D/A conversion starts.
(6) Set the ELSR46 and ELSR47 registers to 0000h to stop event link operation of 12-bit D/A converter. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

61.4.2 Usage Notes on Event Link Operation

- When the event link function is used, set the DACR0.DACEN, DAE, and DACR2.DAAMP bits to 0.
- If an event signal is asserted during writing to the DACR0.DACEN bit, setting 1 in the bit in response to the event signal takes priority.

61.5 Usage Notes

61.5.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The 12-bit D/A converter is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

61.5.2 Operation of the 12-Bit D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the 12-bit D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable output from the 12-bit D/A converter by setting the DACR0.DACEN and DAE bits to 0.

61.5.3 Operation of the 12-Bit D/A Converter in Software Standby Mode or Deep Software Standby Mode

When the MCU enters software standby mode or deep software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode or deep software standby mode, disable output from the 12-bit D/A converter by setting the DACR0.DACEN and DAE bits to 0.

61.5.4 Notes on Output from the DAn Pin

When outputting D/A conversion result from the DAn pin, set the security attribute of the port multiplexed with the DAn pin to either the same security attribute as the corresponding channel, or to non-secure (n = 0, 1). If the channel is set to non-secure and the pin is set to secure, the D/A conversion result will not be output.

61.5.5 Notes on Simultaneous Output to the DAn Pin and Internal Peripheral Modules

If both the DACR0.DAOUTDIS and DAOUTEN bits are set to 1 to output the D/A conversion result simultaneously to the DAn pin and internal peripheral modules, the D/A conversion characteristics specified in the electrical characteristics may not be met (n = 0, 1). A full evaluation is recommended before use when using simultaneous output.

62. Temperature Sensor (TEMPS)

62.1 Overview

This MCU has a built-in temperature sensor. The temperature sensor outputs a voltage proportional to temperature. By converting the output voltage of the temperature sensor into a digital value with a 16-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.
Table 62.1 lists the specifications of the temperature sensor, and Figure 62.1 shows a block diagram of the temperature sensor.

Table 62.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 16-bit A/D converter.
Low-power consumption function	Module stop state can be set.
Temperature Sensor Calibration Data	Reference data measured for each chip at factory shipment is stored.
Abnormal temperature detection	Abnormal high/low temperatures can be detected and a reset can be issued.
Protect Zone filter	Security attribution can be set.

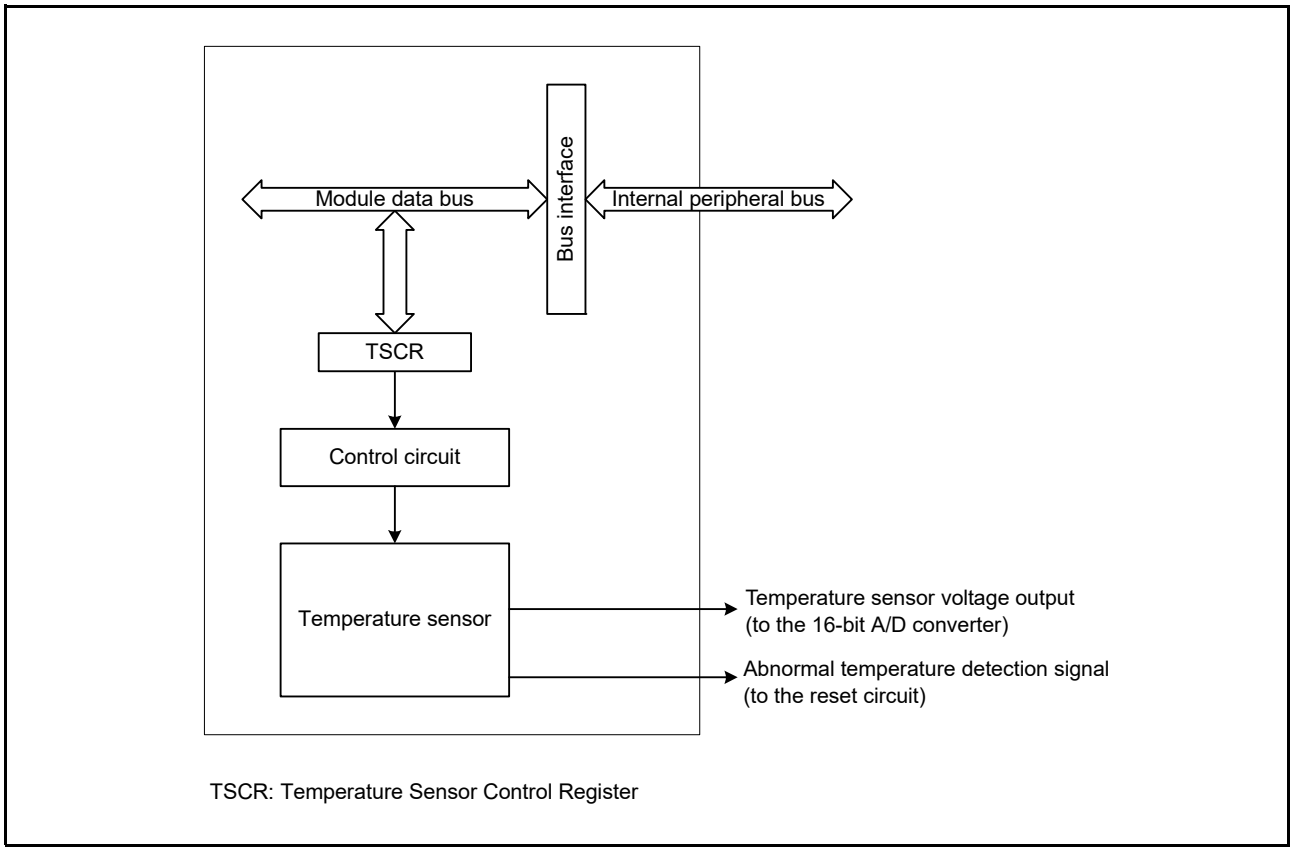


Figure 62.1 Block Diagram of Temperature Sensor

62.2 Register Descriptions

62.2.1 Temperature Sensor Control Register (TSCR)

Address(es): TEMPS.TSCR 8783 5000h, TEMPS_NS.TSCR 0783 5000h

	b7	b6	b5	b4	b3	b2	b1	b0
	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TSOE	Temperature Sensor Output Enable	0: Disables output from the temperature sensor to the 16-bit A/D converter. 1: Enables output from the temperature sensor to the 16-bit A/D converter	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSEN	Temperature Sensor Enable	0: Stops the temperature sensor. 1: Starts the temperature sensor.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The settings of TSCR register have the timing restrictions shown in Figure 62.4.

62.2.2 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): TEMPSCONST.TSCDR 86C1 2DA0h, TEMPSCONST_NS.TSCDR 06C1 2DA0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment. The TSCDR register is a 32-bit read-only register and should be read in 32-bit units.
Temperature sensor calibration data is a digital value obtained using the 16-bit A/D converter to convert the voltage output by the temperature sensor under the condition of Tj = 125°C and AVCC0 = 3.3 V.
The voltage V1 output by the temperature sensor under the condition of Tj = 125°C can be calculated from the value of the TSCDR register according to the formula below.

V1 = 3.3 × value of the TSCDR register/4096 [V]

Note that the AVCC0 voltage does not affect voltage V1.

62.2.3 Temperature Sensor Calibration Data Register 2 (TSCDR2)

Address(es): TEMPSCONST.TSCDR2 86C1 2DA4h, TEMPSCONST_NS.TSCDR2 06C1 2DA4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

The TSCDR2 register stores temperature sensor calibration data measured for each chip at factory shipment. The TSCDR2 register is a 32-bit read-only register and should be read in 32-bit units.
Temperature sensor calibration data is a digital value obtained using the 16-bit A/D converter to convert the voltage output by the temperature sensor under the condition of Tj = –40°C and AVCC0 = 3.3 V.
The voltage V1 output by the temperature sensor under the condition of Tj = –40°C can be calculated from the value of the TSCDR2 register according to the formula below.

V1 = 3.3 × value of the TSCDR2 register/4096 [V]

Note that the AVCC0 voltage does not affect voltage V1.

62.3 Using the Temperature Sensor

The temperature sensor outputs a voltage proportional to temperature. By converting this voltage into a digital value with a 16-bit A/D converter and converting it into temperature, the internal temperature of the MCU can be obtained.

62.3.1 Preparation for Using the Temperature Sensor

Perform a calibration of the temperature sensor as shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
Slope: Temperature slope of the temperature sensor (V/°C); Slope = (V2 – V1)/(T2 – T1)

Since there are individual differences in temperature sensors, it is recommended to prepare a temperature slope by performing trial measurements at two different temperatures as shown below.
Use the 16-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
Again, using the 16-bit A/D converter, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature slope (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature slope given in Table 67.130 of section 67, Electrical Characteristics, use the 16-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this calibration gives less accurate temperatures than two-point calibration.

T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
Slope: Temperature slope given in Table 67.130 ÷ 1000 (V/°C)

In this MCU, the TSCDR register holds the temperature value (CAL_{125}) measured by the temperature sensor under the condition Tj = 125°C and AVCC0 = 3.3 V, and the TSCDR2 register holds the temperature value (CAL_{-40}) measured by the temperature sensor measured under the condition of Tj = –40°C and AVCC0 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
If V1 is calculated from CAL_{125} or CAL_{-40},

V1 = 3.3 × CAL_{125} or CAL_{-40}/4096 [V]
Using this, the measured temperature can be calculated according to the formula below.

T = (Vs – V1)/Slope + 125 or –40 [°C]

T: Measured temperature (internal temperature of the MCU) (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Tj = 125°C or –40°C and AVCC0 = 3.3 V (V)
Slope: Temperature slope given in Table 67.130 ÷ 1000 (V/°C)

The slope can also be obtained by using the value of the TSCDR or TSCDR2 register as the second result of measurement.

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 62.2.

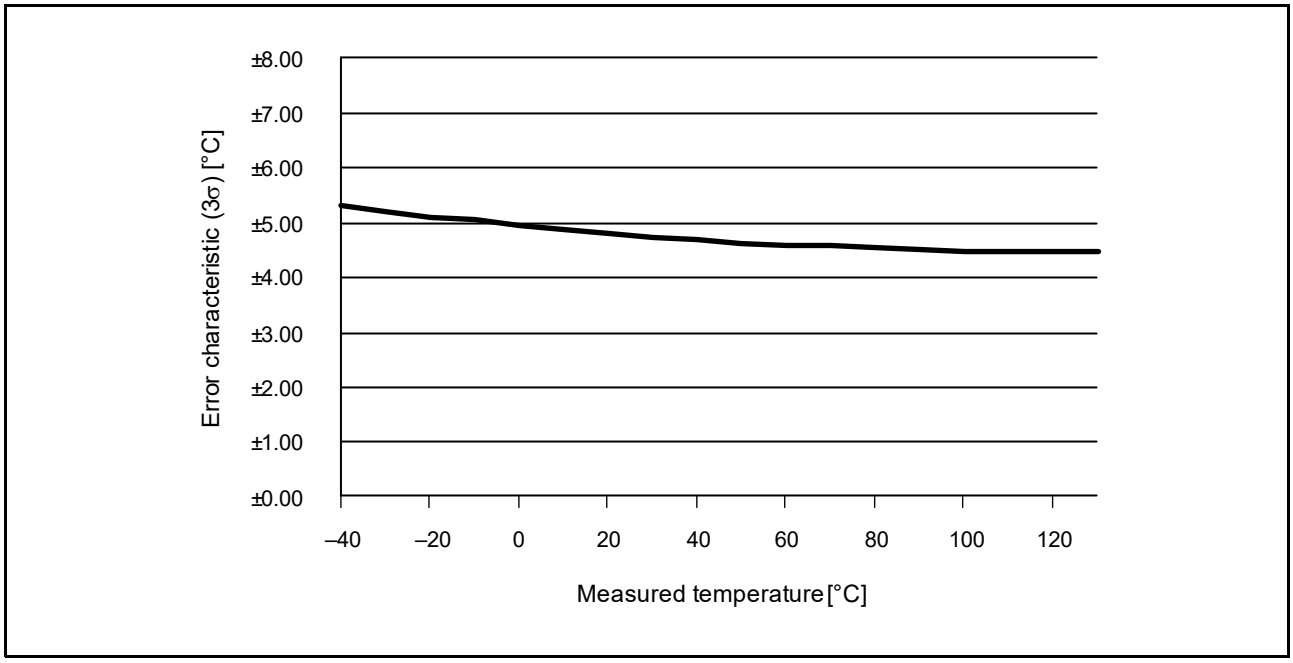
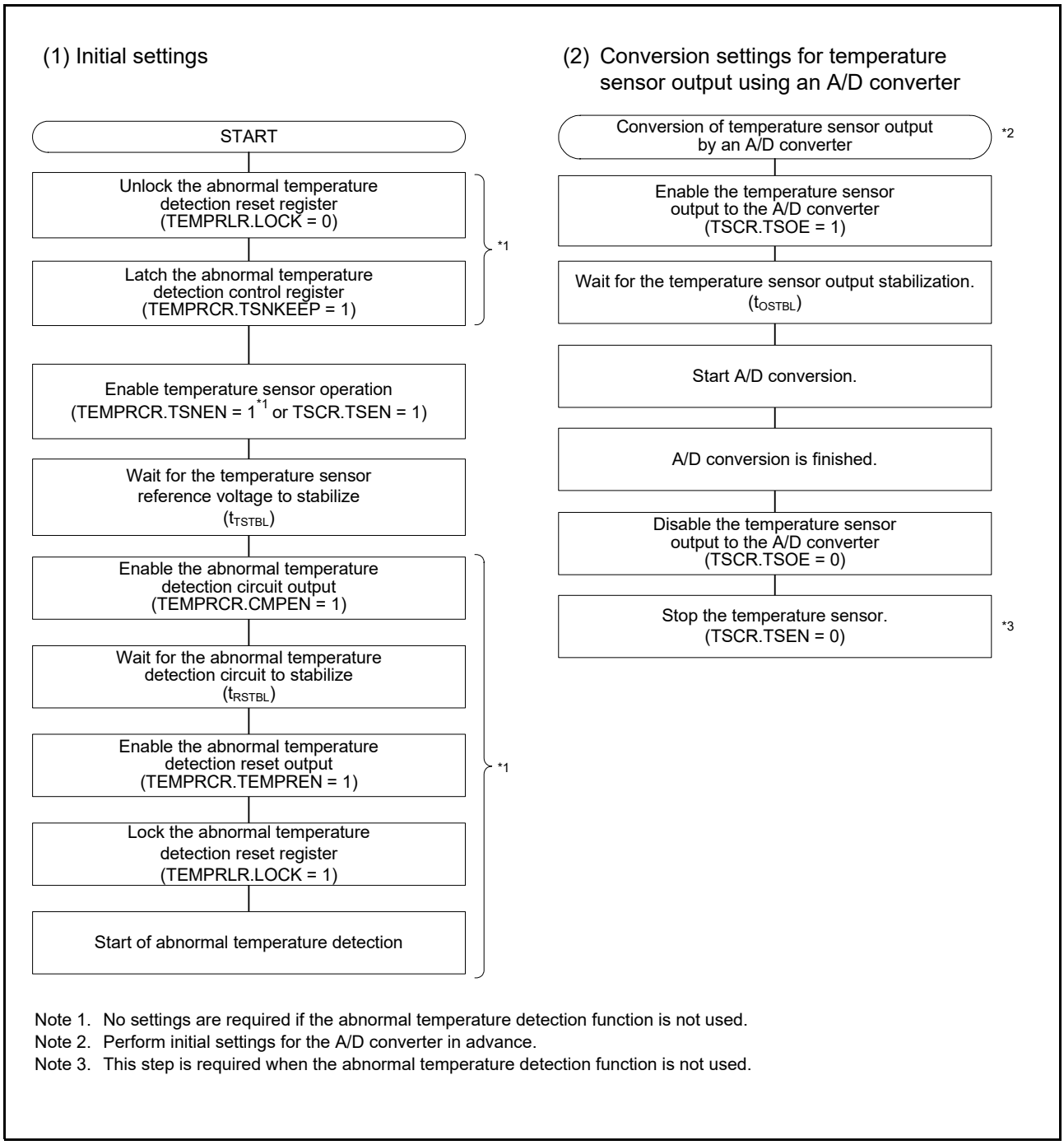


Figure 62.2 Error in the Measured Temperature

62.3.2 Procedure for Using the Temperature Sensor

Figure 62.3 shows the procedure for using the temperature sensor.

To use a temperature sensor, perform the initial settings (1) first. When converting temperature sensor output using an A/D converter, perform the initial settings (1) and then perform the procedure (2) to convert the temperature sensor output using the A/D converter.
For details on how to configure the A/D converter, see section 60, 16-Bit A/D Converter (ADC16H).
For details on registers TEMPRLR and TEMPRCR, see section 6, Resets.



Note 1. No settings are required if the abnormal temperature detection function is not used.
Note 2. Perform initial settings for the A/D converter in advance.
Note 3. This step is required when the abnormal temperature detection function is not used.

Figure 62.3 Procedure for Using the Temperature Sensor

62.3.3 Timing of A/D Conversion of Temperature Sensor Output

Figure 62.4 shows the timing from the start of temperature-sensor operation until the completion of A/D conversion when only the output from the temperature sensor is to be A/D converted and conversion is in single-scan mode. The times shown in the figure are described in Table 62.2.

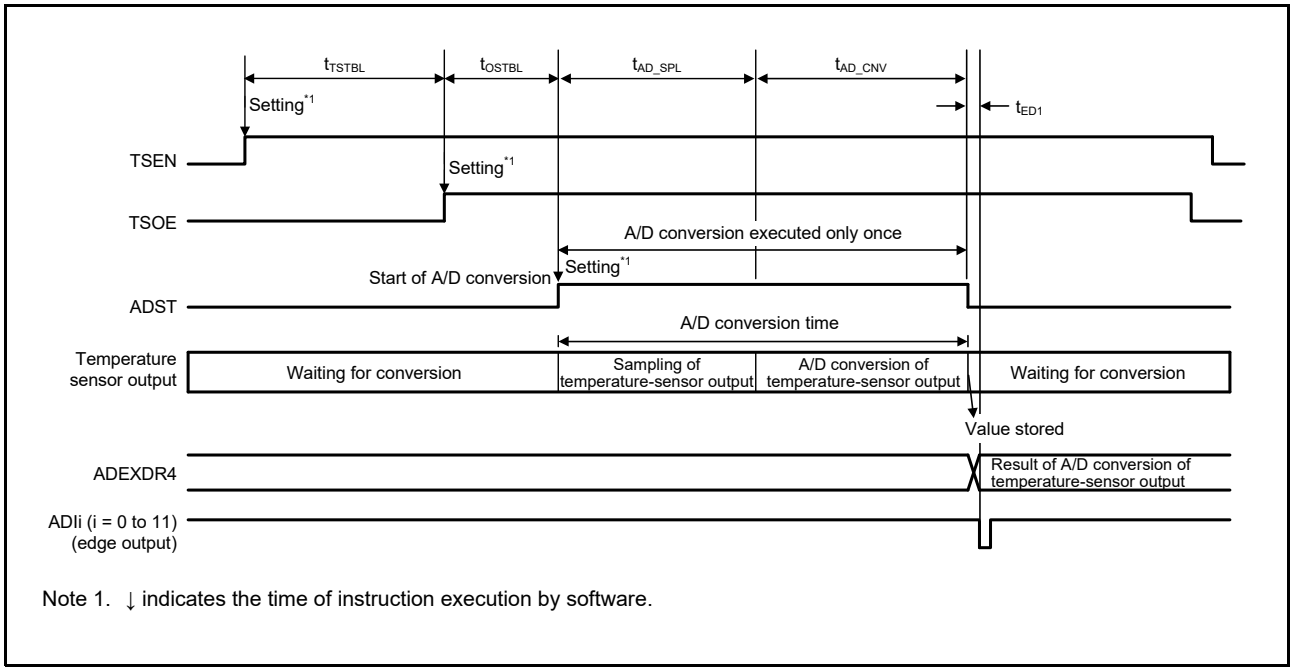


Figure 62.4 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 62.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Temperature-sensor reference-voltage stabilization wait time	t_{TSTBL}	Refer to section 67, Electrical Characteristics.
Temperature-sensor output stabilization wait time	t_{OSTBL}	Refer to section 67, Electrical Characteristics.
Abnormal temperature detection circuit stabilization waiting time	t_{RSTBL}	Refer to section 67, Electrical Characteristics.
16-bit A/D converter input sampling time	t_{AD_SPL}	Refer to “Table 60.69, A/D Convert Processing Time” in “section 60.9.2, Conversion Processing Time”.
16-bit A/D converter successive approximation time	t_{AD_CNV}	Refer to “Table 60.69, A/D Convert Processing Time” in “section 60.9.2, Conversion Processing Time”.
Scan end processing time	t_{ED1}	Refer to “Table 60.70, Scan End Processing Time and the Forcibly Stop Processing Time” in “section 60.9.3, Scan End Processing Time”.

62.4 Usage Note

62.4.1 Module-Stop Function Setting

The corresponding bit in module stop control register D (MSTPCRD) can be used to enable and disable the temperature sensor. The initial setting is for the temperature sensor to be stopped. The register becomes accessible by releasing from the module-stop state. For details, refer to section 11, Low Power Consumption.

62.4.2 Settings for the Software Standby Mode

Before placing the MCU in the software standby mode, stop A/D conversion and set the TSCR.TSOE bit to 0. After that, set the TSCR.TSEN bit to 0 to stop the temperature sensor. Figure 62.5 shows the procedure for a transition to the software standby mode, and Figure 62.6 shows the procedure for return from the software standby mode.

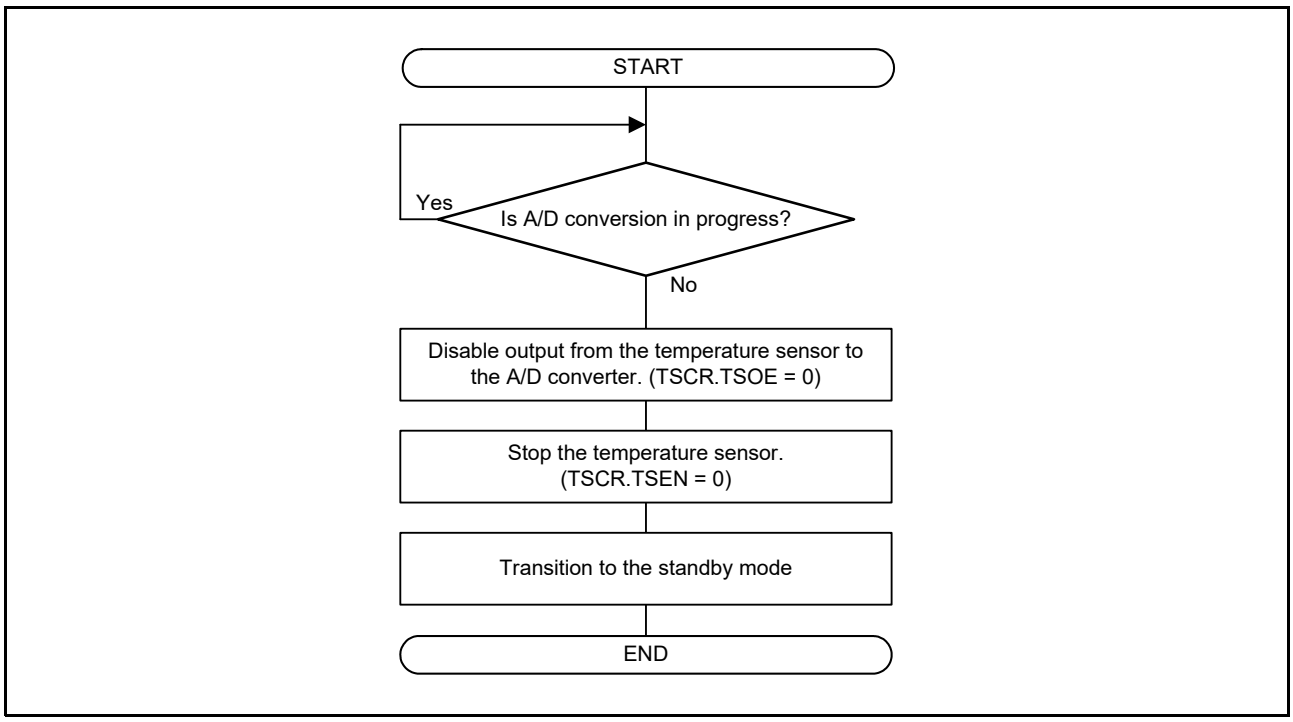


Figure 62.5 Procedure for a Transition to the Software Standby Mode

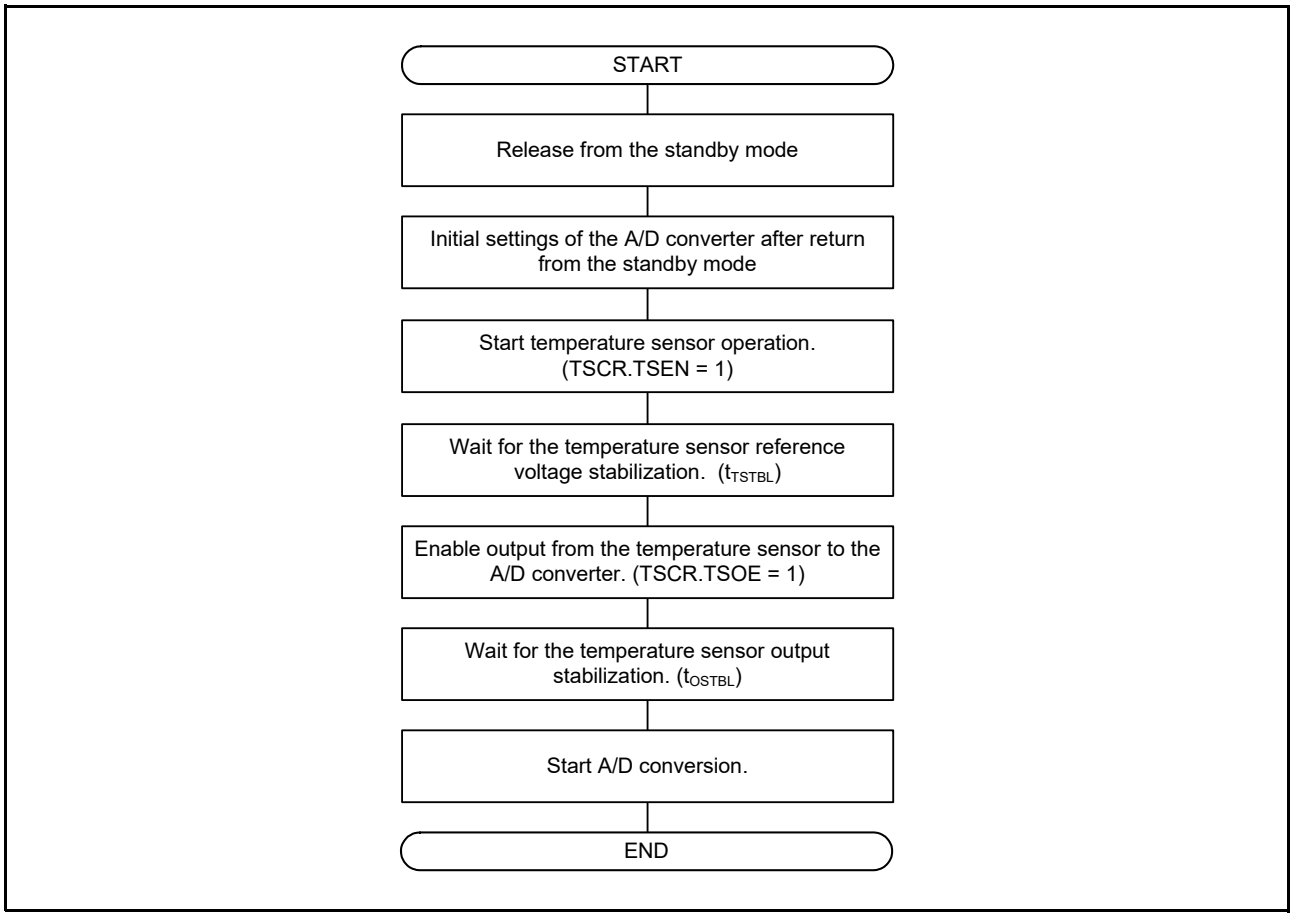


Figure 62.6 Procedure for Return from the Software Standby Mode

63. Data Operation Circuit (DOCA)

63.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16- or 32-bit values.
Table 63.1 lists the specifications of the DOC and Figure 63.1 is a block diagram of the DOC.
An interrupt can be generated if the result of 16- or 32-bit comparison meets one of the set interrupt conditions or if the result of the addition or subtraction result of 16- or 32-bit values is an overflow or underflow.

Table 63.1 DOC Specifications

Item	Description
Data operation function	• Comparison of 16- or 32-bit values (equal or not equal, greater or less than, or within or beyond a range) • Addition or subtraction of 16- or 32-bit values
Lower power consumption function	The DOC can be placed in a module-stop state.
Interrupts	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.
Event link function (output)	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.
Protect Zone filter	Security attribute can be set.*1

Note 1. When this module is set to the secure attribute, every register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this module is set to the non-secure attribute, every register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

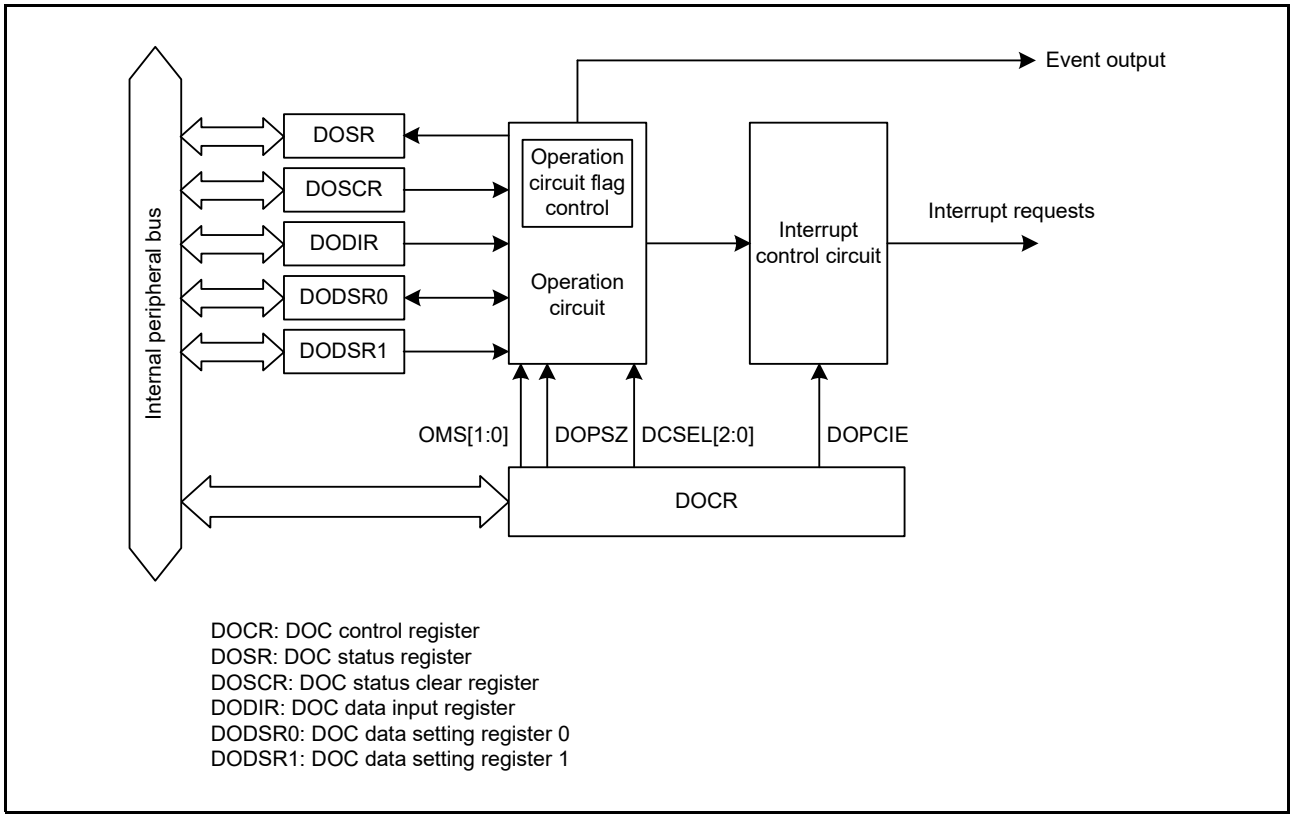


Figure 63.1 DOC Block Diagram

63.2 Register Descriptions

63.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 8741 1000h, DOC_NS.DOCR 0741 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	DOPCIE	DCSEL[2:0]			DOPSZ	—	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DOPSZ	Data Operation Size Select	0: 16-bit width 1: 32-bit width	R/W
b6-b4	DCSEL[2:0]	Detection Condition Select*1	b6 b4 0 0 0: Not equal to (DODIR ≠ DODSR0) 0 0 1: Equal to (DODIR = DODSR0) 0 1 0: Less than (DODIR < DODSR0) 0 1 1: Greater than (DODIR > DODSR0) 1 0 0: Within the range (DODSR0 < DODIR < DODSR1) 1 0 1: Beyond the range (DODIR < DODSR0, DODSR1 < DODIR) Settings other than above are prohibited.	R/W
b7	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR register specifies the operation of DOC, or enabling or disabling of the interrupt.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the DOC.

DOPSZ Bit (Data Operation Size Select)

This bit selects the size of the data operation.

DCSEL[2:0] Bits (Detection Condition Select)

This bit is valid only when data comparison mode is selected.
This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

63.2.2 DOC Status Register (DOSR)

Address(es): DOC.DOSR 8741 1004h, DOC_NS.DOSR 0741 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DOPCF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSR register indicates the results of data operation.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The setting of the DOCR.OMS[1:0] bits is 00b (data comparison mode) and the result of data comparison meets the condition selected by the DOCR.DCSEL[2:0] bits.
- The setting of the DOCR.OMS[1:0] bits is 01b (data addition mode) and the result of addition was greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The setting of the DOCR.OMS[1:0] bits is 10b (data subtraction mode) and the result of subtraction is below 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

63.2.3 DOC Status Clear Register (DOSCR)

Address(es): DOC.DOSCR 8741 1008h, DOC_NS.DOSCR 0741 1008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DOPCFCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	W
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSCR register is for clearing the DOPCF flag. It is always read as 00h.

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOSR.DOPCF flag.

63.2.4 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 8741 100Ch, DOC_NS.DODIR 0741 100Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODIR register is a readable and writable register that holds values for use in operations.
Access the DODIR register with the data operation size selected by the DOCR.DOPSZ bit.

63.2.5 DOC Data Setting Register 0 (DODSR0)

Address(es): DOC.DODSR0 8741 1010h, DOC_NS.DODSR0 0741 1010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR0 register is a readable and writable register that holds values for use in comparison or the results of other operations.
Access the DODSR0 register with the data operation size selected by the DOCR.DOPSZ bit.
In data comparison mode, store the standard value for use in comparison in this register. When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected, specify the lower boundary of the range.
In data addition or data subtraction mode, this register holds the results of operations.

63.2.6 DOC Data Setting Register 1 (DODSR1)

Address(es): DOC.DODSR1 8741 1014h, DOC_NS.DODSR1 0741 1014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR1 register is a readable and writable register that holds a value for use in range comparison.
Access the DODSR1 register with the data operation size selected by the DOCR.DOPSZ bit.
When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected in data comparison mode, specify the upper boundary of the range.
This register is only used when either ‘within the range’ or ‘beyond the range’ is selected.

63.3 Operation

63.3.1 Data Comparison Mode

Figure 63.2 to Figure 63.7 show an example of the steps involved in data comparison mode operation by the DOC*1. An example of operation with the operation size of 32 bits is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode. At the same time, write to the DOCR.DCSEL[2:0] bits to select the condition to be detected.
(2) Specify the standard values for comparison in the DODSR0 and DODSR1 registers.*2
(3) Write the value for comparison in the DODIR register.
(4) If the value written to the DODIR register satisfies the condition set in the DOCR.DCSEL[2:0] bits, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Comparison is made to proceed at the same time a value is written to the DODIR register. Writing values to the DODSR0 and DODSR1 registers does not make comparison proceed.
Note 2. Setting of the DODSR1 register is only required when either 'within the range' or 'beyond the range' is selected. Set a greater value for the DODSR1 register than that of the DODSR0 register.

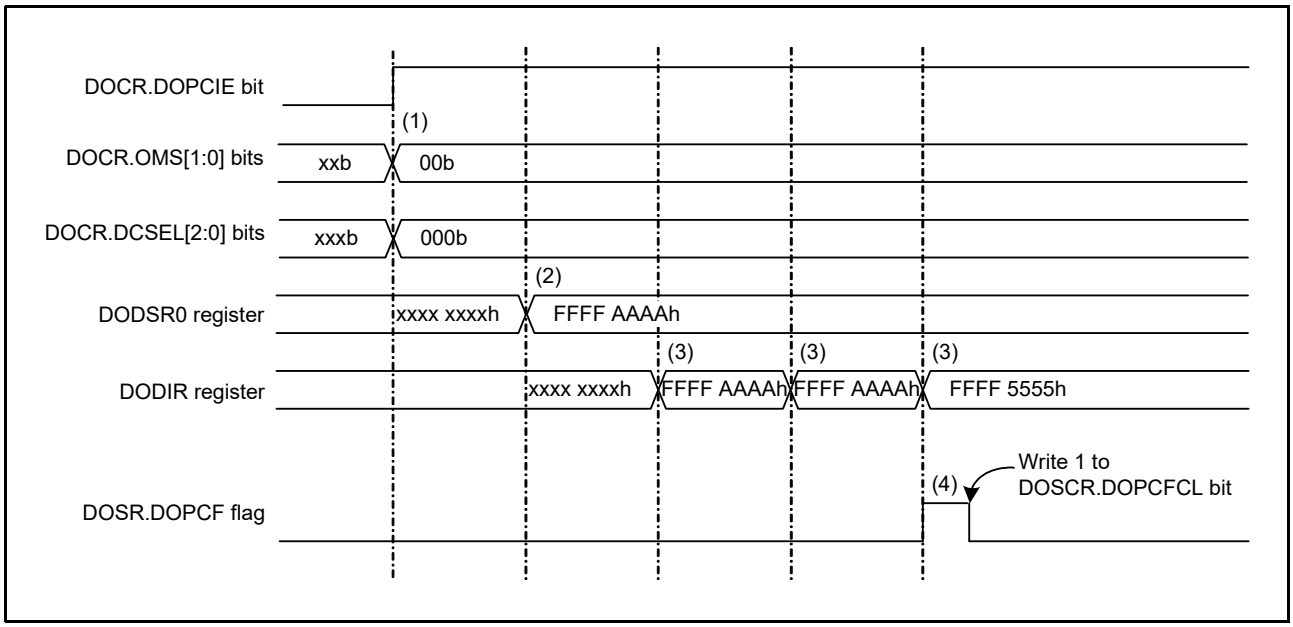


Figure 63.2 Example of the Operation of 'Not Equal to' as the Detection Condition in Data Comparison Mode

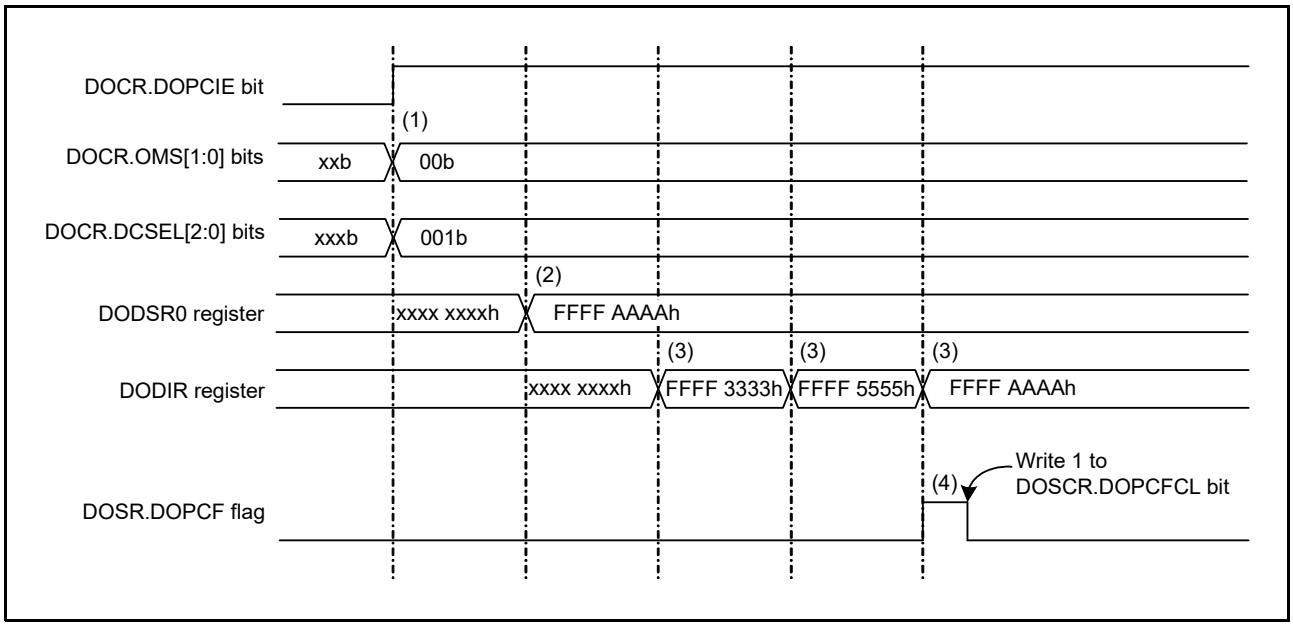


Figure 63.3 Example of the Operation of 'Equal to' as the Detection Condition in Data Comparison Mode

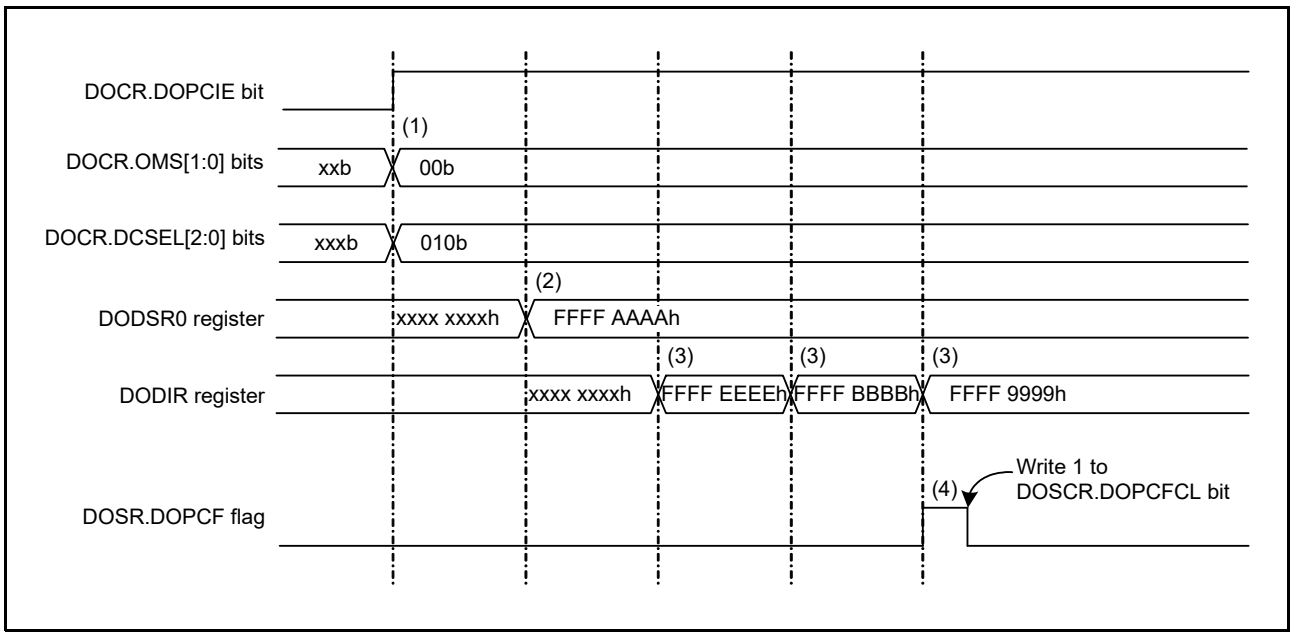


Figure 63.4 Example of the Operation of 'Less Than' as the Detection Condition in Data Comparison Mode

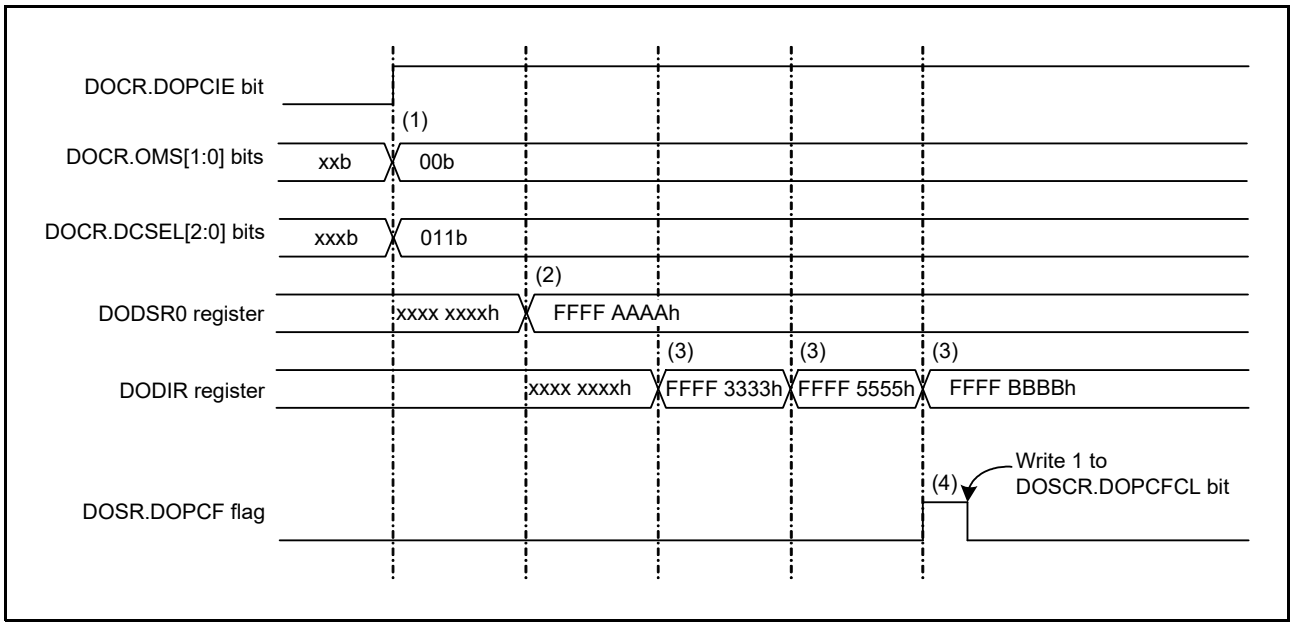


Figure 63.5 Example of the Operation of 'Greater Than' as the Detection Condition in Data Comparison Mode

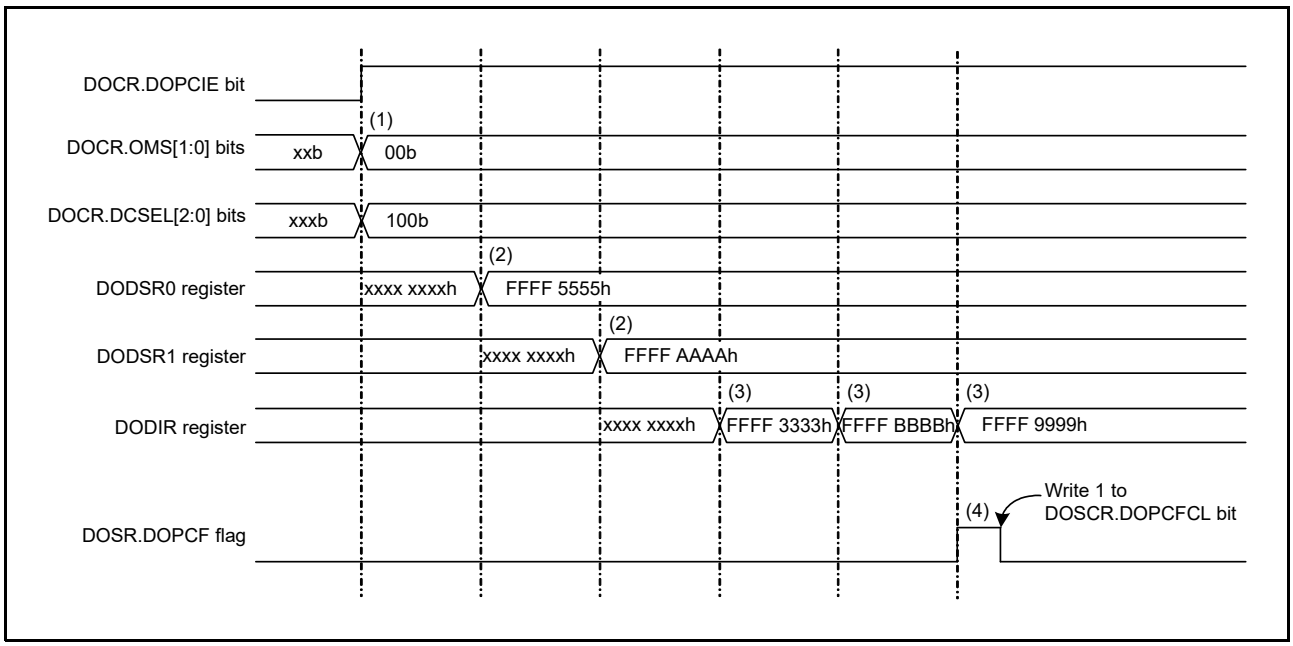


Figure 63.6 Example of the Operation of 'Within the Range' as the Detection Condition in Data Comparison Mode

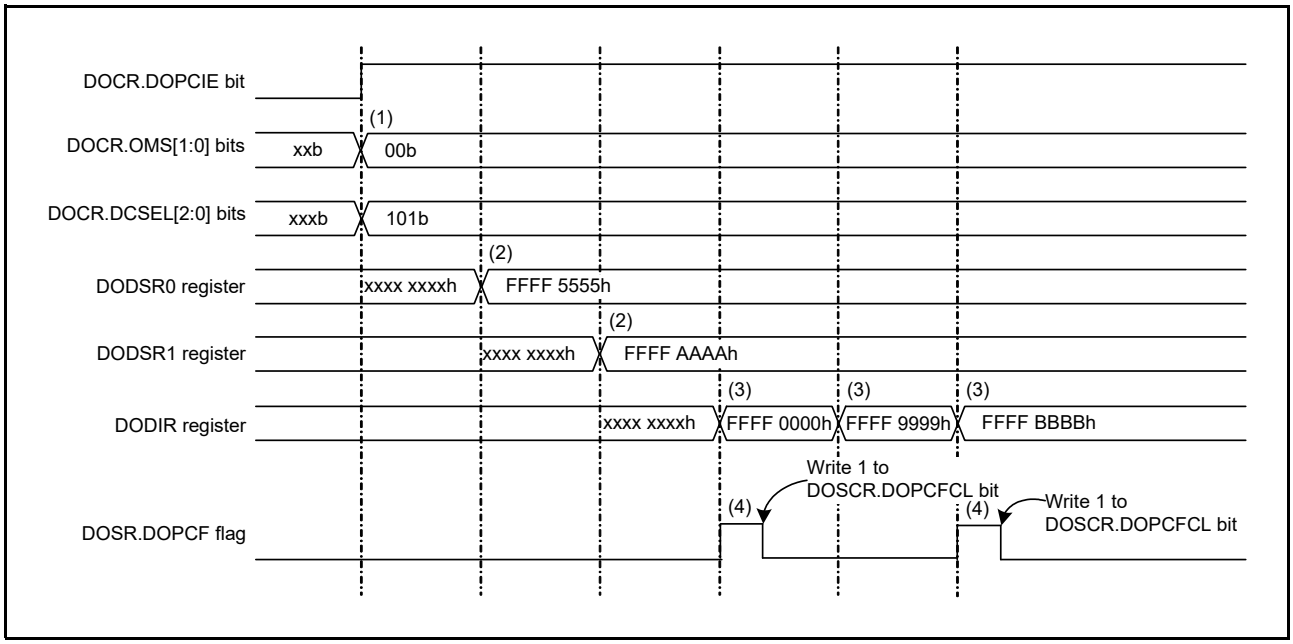


Figure 63.7 Example of the Operation of 'Beyond the Range' as the Detection Condition in Data Comparison Mode

63.3.2 Data Addition Mode

Figure 63.8 shows an example of the steps involved in data addition mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFF FFFFh, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Addition is made to proceed at the same time as a value is written to the DODIR register. Writing a value to the DODSR0 register does not make addition proceed.

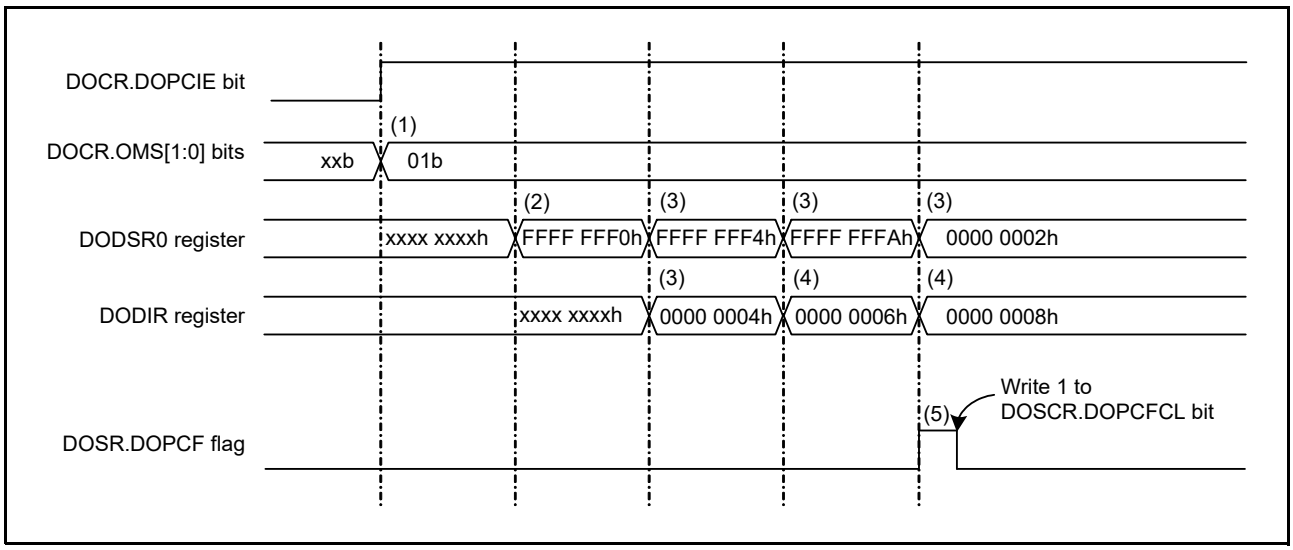


Figure 63.8 Example of Operation in Data Addition Mode

63.3.3 Data Subtraction Mode

Figure 63.9 shows an example of the steps involved in data subtraction mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000 0000h, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Subtraction is made to proceed at the same time a value is written to the DODIR register. Writing a value to the DODSR0 register does not make subtraction proceed.

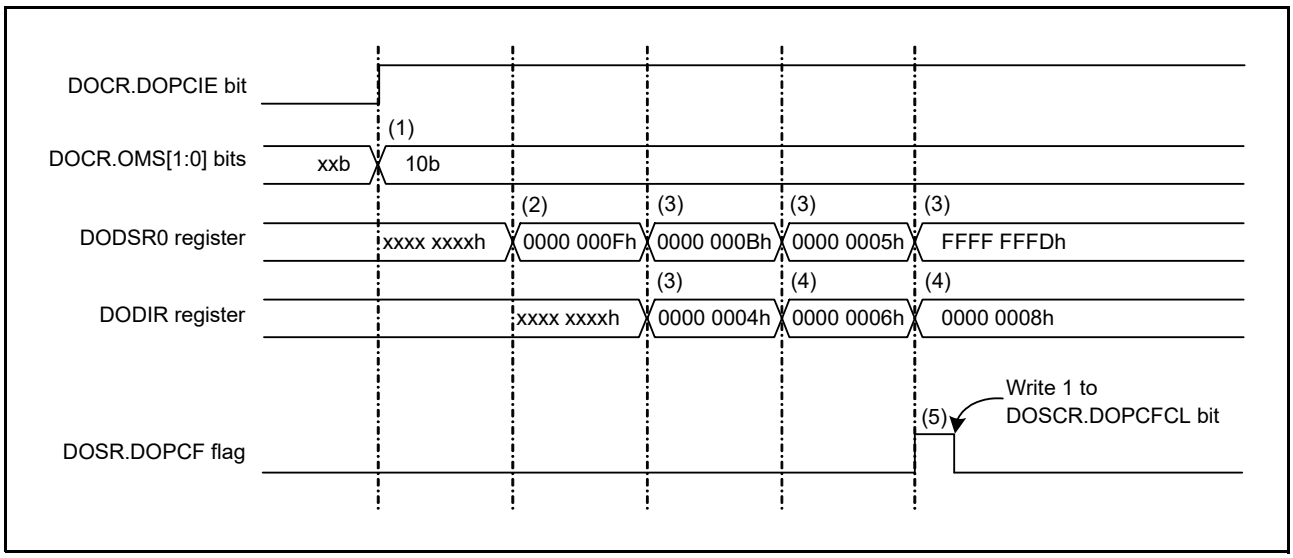


Figure 63.9 Example of Operation in Data Subtraction Mode

63.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOSR.DOPCF flag becomes 1 when the interrupt source condition is satisfied, and an interrupt request is also issued if the DOCR.DOPCIE bit is 1.
Table 63.2 lists the details of the interrupt request.

Table 63.2 Interrupt Request from DOC

Interrupt Request	Data Operation Result Flag	Interrupt Generation Timing
Data operation circuit interrupt (DOPCI)	DOPCF	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1). • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

63.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The result of data comparison meets the detection condition.
- The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

63.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. When an interrupt source condition is satisfied while the interrupt is enabled, the interrupt request signal is issued to the CPU.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

63.6 Usage Note

63.6.1 Module Stop Function Setting

Operation of the DOC can be enabled or disabled by setting the MSTPC13 bit in module stop control register C (MSTPCRC). The DOC is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

64. SRAM

64.1 Overview

The MCU provides an on-chip, SRAM module with parity checking or error correction code (ECC).
Table 64.1 lists the specifications of the SRAM.

Table 64.1 Specifications of the SRAM

Item		SRAM0	SRAM1
Capacity		1 Mbyte	512 Kbytes
Address Range	Secure Area	8200 0000h to 820F FFFFh	8210 0000h to 8217 FFFFh
	Non-secure Area	0200 0000h to 020F FFFFh	0210 0000h to 0217 FFFFh
Access		Supports wait state function. Wait states are not inserted into the access cycle by default. If the ICLK frequency is greater than 150 MHz, a wait state is required. If the ICLK frequency is less than or equal to 150 MHz, a wait state is not required. For details, see section 64.3.6, Access Cycle	
Data Retention		Data retention is not available on deep software standby.	
Low Power Consumption		The SRAM0 and SRAM1 are independently transferable to the module stop state.	
Error Checking		SEC-DED (Single-Error Correction and Double-Error Detection Code: 64-bit data with 8-bit ECC code)	
Security		Protect Zone Filter is integrated for memory access and register access. Access to the memory space is controlled by setting the memory Security Attribution. And access to register space is controlled by setting the register Security Attribution. See section 56, Security Features.	

64.2 Register Descriptions

64.2.1 SRAMn Security Attribute Boundary Address Register (SRAMSABARn) (n = 0, 1)

Address(es): CPSCU.SRAMSABAR0 8701 8400h, CPSCU.SRAMSABAR1 8701 8404h,
CPSCU_NS.SRAMSABAR0 0701 8400h, CPSCU_NS.SRAMSABAR1 0701 8404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the PRCR_S.PRC4 bit.

This register specifies the offset of the boundary between the secure and non-secure areas of each SRAMn (n = 0, 1).
Write the absolute address to this register. The register must be accessed in 32-bit units.
When writing to this register, writing from b31 to b21 is ignored and the write value to b12 to b0 must be 0.
The region lower than the boundary address is marked as secure area, and higher than or equal to the boundary address is marked as non-secure area.
The boundary address is as follows:
8200 0000h + SRAMSABARn (secure area)
0200 0000h + SRAMSABARn (non-secure area)

SRAMSABAR0

When the offset of the boundary address is 0000 0000h, whole SRAM0 is marked as non-secure area.
When the offset of the boundary address is 0010 0000h or greater, whole SRAM0 is marked as secure area.

SRAMSABAR1

When the offset of the boundary address is 0010 0000h or less, whole SRAM1 is marked as non-secure area.
When the offset of the boundary address is 0018 0000h or greater, whole SRAM1 is marked as secure area.

64.2.2 SRAM Security Attribution Register (SRAMSAR)

Address(es): CPSCU.SRAMSAR 8701 8010h, CPSCU_NS.SRAMSAR 0701 8010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SRAM WTSA	—	—	—	—	—	—	SRAMSA1	SRAMSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SRAMSA0	SRAM0-Related Register Security Attribution	0: Secure 1: Non-secure	R/W
b1	SRAMSA1	SRAM1-Related Register Security Attribution	0: Secure 1: Non-secure	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SRAMWTSA	SRAMWTSC Security Attribution	0: Secure 1: Non-secure	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: This register is write-protected by the PRCR_S.PRC4 bit.

This register controls the security attributes of SRAM registers.

SRAMSA0 Bit (SRAM0-Related Register Security Attribution)

This bit is used to specify the security attributes of the following SRAM0-related registers and bits.

- SRAMCR0
- SRAMECCRGN0
- SRAMESCLR.CLR00, SRAMESCLR.CLR01

SRAMSA1 Bit (SRAM1-Related Register Security Attribution)

This bit is used to specify the security attributes of the following SRAM1-related registers and bits.

- SRAMCR1
- SRAMECCRGN1
- SRAMESCLR.CLR10, SRAMESCLR.CLR11

SRAMWTSA Bit (SRAMWTSC Security Attribution)

This bit is used to specify the security attributes of the SRAMWTSC register.

64.2.3 SRAM Protection Control Register for Secure (SRAMPRCR_S)

Address(es): SRAM.SRAMPRCR_S 8701 C000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PR	Register Write Control	0: Writing to registers are disable. 1: Writing to registers are enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KW[7:0]	Write Key Code	Key code protection to the PR bit	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note: It is necessary to write by 16-bit access. 8-bit write access is prohibited. When 8-bit write access is executed, operation is not guaranteed.

This register controls writing to SRAM registers that are specified as secure.

PR Bit (Register Write Control)

This bit controls the write mode of the SRAMWTSC, SRAMCRn, and SRAMECCRGNn registers that are specified as secure by the SRAMSAR register.

When this bit is set to 1, writing to SRAMWTSC, SRAMCRn, and SRAMECCRGNn that are specified as secure is permitted.

Simultaneously with writing this bit, A5h must also be written to the KW[7:0] bits.

KW[7:0] Bits (Write Key Code)

These bits enable or disable writing to the PR bit.

When writing to the PR bit, write A5h to the KW[7:0] bits at the same time. If a value other than A5h is written, the PR bit is not updated. When read, it returns 00h.

64.2.4 SRAM Protection Control Register for Non-Secure (SRAMPRCR_NS)

Address(es): SRAM_NS.SRAMPRCR_NS 0701 C004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PR	Register Write Control	0: Writing registers are disable. 1: Writing registers are enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KW[7:0]	Write Key Code	Key code protection to the PR bit	R/W

Note: This register is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: It is necessary to write by 16-bit access. 8-bit write access is prohibited. When 8-bit write access is executed, operation is not guaranteed.

This register controls writing to SRAM registers that are specified as non-secure.

PR Bit (Register Write Control)

This bit controls the write mode of the SRAMWTSC, SRAMCRn, and SRAMECCRGNn registers that are specified as non-secure by the SRAMSAR register.

When this bit is set to 1, writing to SRAMWTSC, SRAMCRn, and SRAMECCRGNn that are specified as non-secure is permitted.

When writing to this bit, A5h must also be written to the KW[7:0] bits.

KW[7:0] Bits (Write Key Code)

These bits enable or disable writing to the PR bit.

When writing to the PR bit, write A5h to the KW[7:0] bits at the same time. If a value other than A5h is written, the PR bit is not updated. When read, it returns 00h.

64.2.5 SRAM Wait State Control Register (SRAMWTSC)

Address(es): SRAM.SRAMWTSC 8701 C008h, SRAM_NS.SRAMWTSC 0701 C008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	WTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WTEN	SRAM wait enable	0: Do not add wait state in the access cycle to SRAMs. 1: Add wait state in the access cycle to SRAMs.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note: This register can be written only when the PR bit of the SRAMPRCR_S or SRAMPRCR_NS register is 1.

This register sets whether or not to insert wait cycles into SRAM access cycles.

WTEN Bit (SRAM wait enable)

This bit sets the wait cycle insertion to the access cycle of SRAM. When this bit is set to 1, one wait cycle is inserted to the read access cycle of SRAM. When this bit is 1 and there are consecutive accesses to the same area*1 and the same mat*2, one wait cycle is inserted in the second access cycle. When access frequency is greater than 150 MHz, set this bit to 1.

Note 1. The area of SRAM is divided to 128KB units.
Note 2. The mat is divided to 0h to 7h and 8h to Fh in the lower 4-bits of the address.

For example, in the case of SRAMn, the same area and same mat means 16 areas in the figure below. When one area is accessed consecutively, one wait cycle is inserted.
Figure 64.1 shows the mat configuration of each SRAM.

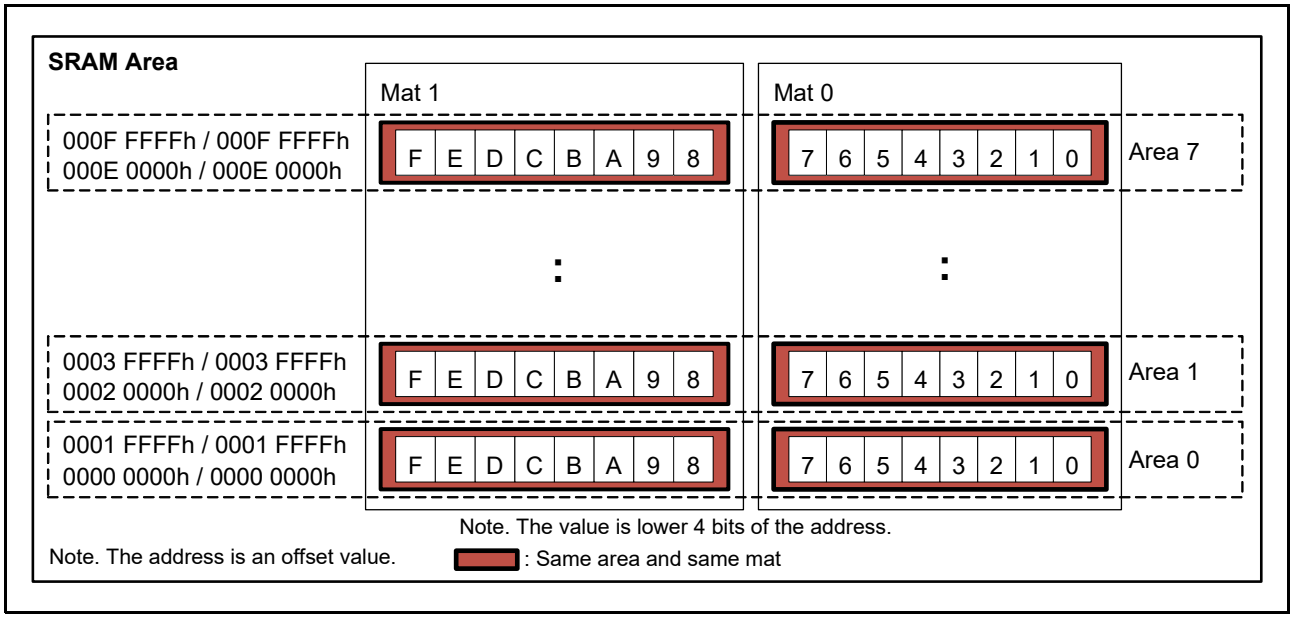


Figure 64.1 Mat configuration of each SRAMs

64.2.6 SRAM Control Register n (SRAMCRn) (n = 0, 1)

Address(es): SRAM.SRAMCR0 8701 C010h, SRAM.SRAMCR1 8701 C014h,
SRAM_NS.SRAMCR0 0701 C010h, SRAM_NS.SRAMCR1 0701 C014h

	b7	b6	b5	b4	b3	b2	b1	b0
	TSTBYP	—	—	E1STSEN	ECCMOD[1:0]		—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OAD	Operation after error detection	0: Interrupt 1: Reset	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b2	ECCMOD[1:0]	ECC Operating Mode Select	b3 b2 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking	R/W
b4	E1STSEN	ECC 1-bit Error Update Enable	0: Disable updating of 1-bit ECC error 1: Enable updating of 1-bit ECC error	R/W
b6 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSTBYP	ECC Bypass Select	0: Disable ECC bypass 1: Enable ECC bypass	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

This register controls the functions related to the ECC operation of SRAMn (n = 0, 1).

OAD Bit (Operation after error detection)

Select whether to request a reset or an interrupt when an ECC error is detected.
For information about handling the interrupt (also selectable as a non-maskable interrupt), refer to section 15, Interrupt Controller (ICUM).

E1STSEN Bit (ECC 1-bit Error Update Enable)

This bit enables or disables updating of the ECC 1-bit error status bit in the SRAMESR register, in response to a 1-bit error.
This bit also functions as an interrupt or a reset mask.

TSTBYP Bit (ECC Bypass Select)

This bit enables direct access to ECC code by bypassing the ECC function. The ECC Bypass function is used along with setting the ECCMOD[1:0] bits to 00b.
The ECC must be accessed the same address. The ECC code is assigned to the lower 8 bits. When writing the ECC code, the bits above the ECC bits are ignored. When reading the ECC code, the bits above the ECC bits are invalid.
For details about ECC test, refer to section 64.3.3, ECC Decoder Testing.

64.2.7 SRAM ECC Region Control Register 0 (SRAMECCRGN0)

Address(es): SRAM.SRAMECCRGN0 8701 C030h, SRAM_NS.SRAMECCRGN0 0701 C030h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ECCRGN[3:0]			
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ECCRGN[3:0]	ECC target region select	b3 b0 0 0 0 0: No ECC target area 0 0 0 1: 0200 0000h to 0201 FFFFh 8200 0000h to 8201 FFFFh (128 Kbytes) 0 0 1 0: 0200 0000h to 0203 FFFFh 8200 0000h to 8203 FFFFh (256 Kbytes) 0 0 1 1: 0200 0000h to 0205 FFFFh 8200 0000h to 8205 FFFFh (384 Kbytes) 0 1 0 0: 0200 0000h to 0207 FFFFh 8200 0000h to 8207 FFFFh (512 Kbytes) 0 1 0 1: 0200 0000h to 0209 FFFFh 8200 0000h to 8209 FFFFh (640 Kbytes) 0 1 1 0: 0200 0000h to 020B FFFFh 8200 0000h to 820B FFFFh (768 Kbytes) 0 1 1 1: 0200 0000h to 020D FFFFh 8200 0000h to 820D FFFFh (896 Kbytes) 1 0 0 0: 0200 0000h to 020F FFFFh 8200 0000h to 820F FFFFh (1 Mbyte) Settings other than above are prohibited. The ECC target area selected by these bits can be controlled by the SRAMCR0 register.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

This register selects the ECC target area of SRAM0.

64.2.8 SRAM ECC Region Control Register 1 (SRAMECCRGN1)

Address(es): SRAM.SRAMECCRGN1 8701 C034h, SRAM_NS.SRAMECCRGN1 0701 C034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ECCRGN[3:0]			
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ECCRGN[3:0]	ECC target region select	b3 b0 0 0 0 0: No ECC target area 0 0 0 1: 0210 0000h to 0211 FFFFh 8210 0000h to 8211 FFFFh (128 Kbytes) 0 0 1 0: 0210 0000h to 0213 FFFFh 8210 0000h to 8213 FFFFh (256 Kbytes) 0 0 1 1: 0210 0000h to 0215 FFFFh 8210 0000h to 8215 FFFFh (384 Kbytes) 0 1 0 0: 0210 0000h to 0217 FFFFh 8210 0000h to 8217 FFFFh (512 Kbytes) Settings other than above are prohibited. The ECC target area selected by these bits can be controlled by the SRAMCR1 register.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note: This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

This register selects the ECC target area of SRAM1.

64.2.9 SRAM Error Status Register (SRAMESR)

Address(es): SRAM.SRAMESR 8701 C040h, SRAM_NS.SRAMESR 0701 C040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ERR11	ERR10	ERR01	ERR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERR00	SRAM0 1-bit ECC Error Status	0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
b1	ERR01	SRAM0 2-bit ECC Error Status	0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
b2	ERR10	SRAM1 1-bit ECC Error Status	0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
b3	ERR11	SRAM1 2-bit ECC Error Status	0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
b15 to b4	—	Reserved	These bits are read as 0	R

This register is the SRAM ECC error status register.
This register is cleared by the corresponding bit in the SRAMESCLR register, or by any reset other than a bus error reset or memory error reset. If an error and a clear occur at the same time, clear takes priority.

ERRn0 Flags (SRAMn 1-bit ECC Error Status) (n = 0, 1)

This flag indicates whether or not a 1-bit ECC error has occurred in SRAMn.
When the ECC function is enabled, error checking is enabled (SRAMCRn.ECCMOD[1:0] = 11b), and the SRAMCRn.E1STSEN bit is set to 1, the ERRn0 flag becomes 1 when a 1-bit error is detected. When the ERRn0 flag becomes 1, a reset or interrupt request occurs according to the SRAMCRn.OAD bit.

ERRn1 Flags (SRAMn 2-bit ECC Error Status) (n = 0, 1)

This flag indicates whether or not a 2-bit ECC error has occurred in SRAMn.
When the ECC function is enabled and error checking is performed (SRAMCRn.ECCMOD[1:0] = 11b), if a 2-bit error is detected, the ERRn1 flag becomes 1. When the ERRn1 flag becomes 1, a reset or interrupt request occurs according to the SRAMCRn.OAD bit.

64.2.10 SRAM Error Status Clear Register (SRAMESCLR)

Address(es): SRAM.SRAMESCLR 8701 C048h, SRAM_NS.SRAMESCLR 0701 C048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CLR11	CLR10	CLR01	CLR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR00	SRAM0 1-bit ECC Error Status Clear	Setting this bit to 1 clears the SRAMESR.ERR00 flag. The read value is always 0.	R/W*1
b1	CLR01	SRAM0 2-bit ECC Error Status Clear	Setting this bit to 1 clears the SRAMESR.ERR01 flag. The read value is always 0.	R/W*1
b2	CLR10	SRAM1 1-bit ECC Error Status Clear	Setting this bit to 1 clears the SRAMESR.ERR10 flag. The read value is always 0.	R/W*1
b3	CLR11	SRAM1 2-bit ECC Error Status Clear	Setting this bit to 1 clears the SRAMESR.ERR11 flag. The read value is always 0.	R/W*1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The security attributes for this register can be set for each bit.
When a bit is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, but no Protect Zone access errors are generated.
When a bit is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, but no Protect Zone access errors are generated.

Note 1. Only 1 can be written.

This register clears the ECC error status flags in the SRAMESR register.

If a flag is set due to an error and is also cleared by the SRAMESCLR register at the same time, the clear takes precedence.

64.2.11 SRAMn Error Address m Register (SRAMEARnm) (n = 0, 1; m = 0, 1)

Address(es): SRAM.SRAMEAR00 8701 C050h, SRAM.SRAMEAR01 8701 C054h, SRAM.SRAMEAR10 8701 C060h, SRAM.SRAMEAR11 8701 C064h, SRAM_NS.SRAMEAR00 0701 C050h, SRAM_NS.SRAMEAR01 0701 C054h, SRAM_NS.SRAMEAR10 0701 C060h, SRAM_NS.SRAMEAR11 0701 C064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is a 32-bit read-only register that stores the offset of the error address when an SRAM error occurs.
It is cleared by clearing the ECC error status flag using the corresponding bit in the SRAMESCLR register, or by any reset other than a bus error reset or memory error reset. If the offset of the error address due to an error is stored and cleared at the same time, the clear takes priority.
The error address is as follows.
8200 0000h + SRAMEARnm (secure area)
0200 0000h + SRAMEARnm (non-secure area)

For SRAMEARn0 (n = 0, 1)

This register stores the offset of the error address where a 1-bit ECC error was detected. It holds the offset of the address where the first error occurred. It is cleared by clearing the 1-bit ECC error status flag by the SRAMESCLR.CLRn0 bit.

For SRAMEARn1 (n = 0, 1)

This register stores the offset of the error address where a 2-bit ECC error was detected. It holds the offset of the address where the first error occurred. It is cleared by clearing the 2-bit ECC error status flag by the SRAMESCLR.CLRn1 bit.

64.3 Operation

64.3.1 ECC Function

Enabling and disabling of ECC error correction can be selected through the SRAMCRn.ECCMOD[1:0] bits setting. In the initial state, ECC error correction is disabled. The ECC used in this device is SEC-DED (Single-Error Correction and Double-Error Detection Code).
When ECCs are enabled, an 8-bit ECC code is appended to 64-bit data for writing. For reading, 72-bit (data: 64 bits, ECC code: 8 bits) data is read out from the SRAM.

ECC Function with Error Checking Enable

When a 1-bit error occurs, error correction is performed, and when the SRAMCRn.E1STSEN bit is 1, the SRAMESR.ERRn0 flag is set to 1. When a 2-bit error occurs, error detection is performed, and the SRAMESR.ERRn1 flag is set to 1, but error correction is not performed.

ECC Function without Error Checking Enable

When a 1-bit error occurs, error correction is performed, but the SRAMESR.ERRn0 flag is not updated even if the SRAMCRn.E1STSEN bit is 1. When a 2-bit error occurs, error detection is performed, but the SRAMESR.ERRn1 flag is not updated, and error correction is not performed.

ECC Function Disabled

Even if a 1-bit or 2-bit error occurs, error correction and error detection are not performed. Therefore, the SRAMESR.ERRn0 and SRAMESR.ERRn1 flags are not updated.

When updating data after an error occurs, only 64-bit data writes are supported. Since the SRAM data is undefined after power-on or after waking up from deep software standby mode, an ECC error will occur when accessing the SRAM if ECC enabled and error check is selected. Therefore, before using the ECC function, enable ECC and perform an initial write with 64-bit data size, or enable ECC and perform an initial write without error check, for the area to be used in the SRAM.

64.3.2 ECC Error Interrupt Function

When ECC function with error checking is enabled, error checking is possible in the SRAMn. In the case of a 2-bit error, the ERRn1 flag in the SRAMESR register set to 1. In the case of a 1-bit error, the ERRn0 flag in the SRAMESR register set to 1 when SRAMCRn.E1STSEN bit is 1.
When the ECC 1-bit error is to be masked, it is necessary to set the SRAMCRn.E1STSEN bit to 0 to disable updating to 1 of the SRAMESR.ERRn0 flag. An ECC error will not be generated while ECC function is disabled or ECC function without error checking is enabled.
ECC error may result in interrupt or reset. When an ECC error occurs, a reset request is output if the SRAMCRn.OAD bit is 1. If the OAD bit is 0, an interrupt request is output to the ICU.
ECC Error interrupt occurs when one flag in the SRAMESR register is set. ECC Error interrupt continues to occur until the SRAMESR register flag is cleared.

64.3.3 ECC Decoder Testing

This MCU allows users to test the ECC decoder.
To test your ECC decoder, follow the steps below.

1. Wait for the previous access to complete.
2. Write A501h to the SRAMPRCR_S or SRAMPRCR_NS register and enable writing to SRAM-related registers.
3. Write 08h to the SRAMCRn register and set ECC function without error checking enabled, and ECC bypass disabled.
4. Wait for 08h to be written in the SRAMCRn register.
5. Writes 8 bytes of data to the target address. The 8-bit ECC code is updated automatically.
6. Wait for the previous access to complete.
7. Write 80h to the SRAMCRn register, set ECC disabled and ECC bypass enabled.
8. Wait for 80h to be written in the SRAMCRn register.
9. Read the target address and get the 8-bit ECC code.
10. To generate a 1-bit/2-bit ECC error, invert the 1-bit/2 bit of the data read in step 9 and write the data back to the target address.
11. Wait for the previous access to complete.
12. Write 1Ch to the SRAMCRn register and set the ECC function with error checking enabled, ECC 1-bit error information update enabled, and ECC bypass disabled.
13. Wait for 1Ch to be written in the SRAMCRn register.
14. Read the target address.
15. Wait for the previous access to complete.
16. Confirm the generation of the ECC error by the SRAMESR register.

64.3.4 Interrupt Source

The only SRAM interrupt source is an ECC error. The SRAMCRn.OAD bit can be used to set whether an ECC error causes a reset or an interrupt request. An SRAM interrupt request is generated when any error status flag in the SRAMESR register is set to 1, and remains active until the corresponding error status flag is cleared. If a common memory error occurs (NMISR.CMST = 1 or RSTSR1.CMRF = 1), read the SRAMESR register and check the cause of the SRAM interrupt. Table 64.2 lists the SRAM interrupt source.

Table 64.2 SRAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
ECCERR	ECC error (SRAMn with ECC)	Not possible	Not possible

64.3.5 Wait State

When the ICLK frequency is greater than 150 MHz, set the SRAMWTSC.WTEN bit to 1, as the wait cycle must be inserted into SRAM0 and SRAM1. When the wait cycles are not inserted, operation is not guaranteed.
The relationship between SRAM access weight setting and ICLK frequency is as follows:
[ICLK frequency]

- 300 MHz ≥ ICLK > 150 MHz : 1 wait
- 150 MHz ≥ ICLK : No wait

64.3.6 Access Cycle

The following shows the delay information of the access cycle.

- When the ECC function is enabled, writing through access that is not aligned or writing through access that is aligned but with a unit of less than 64 bits is delayed by two cycles relative to normal writing.
- Wait cycles are inserted according to the setting of the SRAMWTSC.WTEN bit. If the SRAMWTSC.WTEN bit is 1, reads are delayed by one cycle.

64.4 Usage Notes

64.4.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to the SRAM.
When the MSTPCRA.MSTPA0 bit is set to 1, the clock supplied to SRAM0 will be stopped, and when the MSTPCRA.MSTPA1 bit is set to 1, the clock supplied to SRAM1 will be stopped. When the clock supply stops, the SRAM enters the module-stop state.
After reset, SRAM is in operation.
In the module stop state, SRAM cannot be accessed. Do not enter the module stop state while accessing SRAM. In the module stop state, access to the SRAM is prohibited. If accessed, operation cannot be guaranteed.
For details on the MSTPCRA register, refer to section 11, Low Power Consumption.

64.4.2 Instruction Fetch from SRAM Area

When executing programs using SRAM, initialize the SRAM area so that the CPU can properly prefetch data. ECC error might occur if the CPU prefetches from an area that is not initialized.
The SRAM must be initialized at an 8-byte boundary. When enabling the ECC function and executing a program on SRAM, initialize the SRAM taking into consideration that the CPU will perform instruction prefetching. Since instruction prefetching is performed up to 32 bytes, initialize an extra 24 to 31 bytes from the last address of the program.

64.4.3 Notes when using the SRAM error check function

Data in SRAM are undefined when the power is turned on. Therefore, an ECC error occurs if the data are read before initialization.
SRAM is read in 8-byte (64-bit) units. Initialize at an 8-byte boundary.

65. MRAM

This MCU incorporates a code MRAM, data MRAM, option-setting memory, and user extra MRAM. Instructions and operands can be stored in the code MRAM. Data can only be stored in the data MRAM. For option-setting memory, see section 7, Option-Setting Memory (OFSM). The user program and data can be stored in the user extra MRAM as required by the user's application. In this section, the data MRAM, option-setting memory, and user extra MRAM are collectively referred to as extra MRAM.

65.1 Overview

Table 65.1 lists the specification of MRAM, and Figure 65.1 shows a block diagram of the MRAM related modules. The programming sequencer controls programming of the code MRAM and extra MRAM. Two program sequencers are available: CPSEQ for the code MRAM and EPSEQ for the extra MRAM. The MRAM application command interface (MACI) controls EPSEQ for option-setting memory and user extra MRAM according to the specified MACI commands.

Table 65.1 Specifications of MRAM (1/2)

Item	Code MRAM	Extra MRAM		
		Data MRAM	Option-Setting Memory	User Extra MRAM
Memory capacity	User area: 1 Mbyte max	Data area: 48 Kbytes	See section 7, Option-Setting Memory (OFSM)	User area: 128 Kbytes GPOTP: 96 bytes
Switch to operate as OTP	Not possible	Not possible	See section 7, Option-Setting Memory (OFSM)	Possible
Read cycle	See section 65.5.2, Code MRAM Frequency Notifications Register (MRCFREQ)	See section 65.5.3, Extra MRAM Frequency Notifications Register (MREFREQ)		
Programming method	Write page buffer (32 bytes)	Write page buffer (16 bytes)	MACI command	
Protection	Protects against erroneous rewriting of the MRAM			
Dual bank function	The dual-bank structure makes a safe update possible in cases where programming is suspended. • Linear mode: The code MRAM is used as one area. • Dual mode: The code MRAM is divided into two areas (MRC0 area = bank 0 and MRC1 area = bank 1).	Not available		
Block swap function	Block swap block size: 8 Kbytes Block swapping allows safe updates to code in operations by exchanging working sectors	Not available		
Background operations (BGOs)	• The code MRAM can be read while the extra MRAM is being programmed. • The extra MRAM can be read while the code MRAM is being programmed. • A bank of the code MRAM can be read while another bank of the code MRAM is being programmed.			
Units of programming	1 to 32 bytes	1 to16 bytes	16 bytes	
MACI command	Not available	Not available	Program: 16 bytes Forced stop Status clear Configuration set: 16 bytes Increment counter: 1 bit Read counter: 8 bytes	

Table 65.1 Specifications of MRAM (2/2)

Item	Code MRAM	Extra MRAM		
		Data MRAM	Option-Setting Memory	User Extra MRAM
Security function	• Protect against illicit tampering with or reading out of data in MRAM • OFS area protection • Startup area select setting protection - BTFLG and MSUACR register are protected by the FSPR bit. - BTSIZE is protected by the FSPRS bit. • Permanent block protect setting protection - User area is permanently protected from programming operation by permanent block protect function. • MRAM protection for Protect Zone - Protection for MRAM area (program) - Protection for MRAM area (read) - Protection for register • Extra MRAM program mode entry protection • Anti-rollback counter • HUK protection			
Safety function	Software protection • MACI command protection by MENTRYR register • User area protection by MRCPCS and MRCPCNS registers • User area and data area protection by the block protect setting Error protection • Error is detected when unintended commands or prohibited settings occur. The MACI command is not accepted after an error detection. Startup area protection • The startup area select function allows the user to safely update the startup program, which is to be started after a reset. The size of the startup area can be selected from 8 Kbytes, 16 Kbytes or 32 Kbytes. ECC support for double-bit error correction and triple-bit error detection.			
Interrupt request	• MRCRDI (code MRAM ECC error) Enabled by the INTENBTC and INTENBDC bits • MRERDI (Extra MRAM ECC error) Enabled by the INTENBTE and INTENBDE bits • MRCPRI (code MRAM access error) Enabled by the MRCAEIE bit • MRDPRI (data MRAM access error) Enabled by the MRDAEIE bit • MREPRI (Extra MRAM access error) Enabled by the CMDLKIEE and MREAEIE bits • MRDYI (Extra MRAM ready) Enabled by the MRDYIE bit • MRPRRI (MRAM program ready) Enabled by the MRCPRRIE or MRDPRRIE bit			
Address translation	• The startup area select function is supported in linear mode. • The bank swap function is supported in dual mode. • The block swap function is supported in linear mode.			

Note 1. Once switched to operate as OTP (One Time Programmable), the specified area of the MRAM cannot be rewritten.

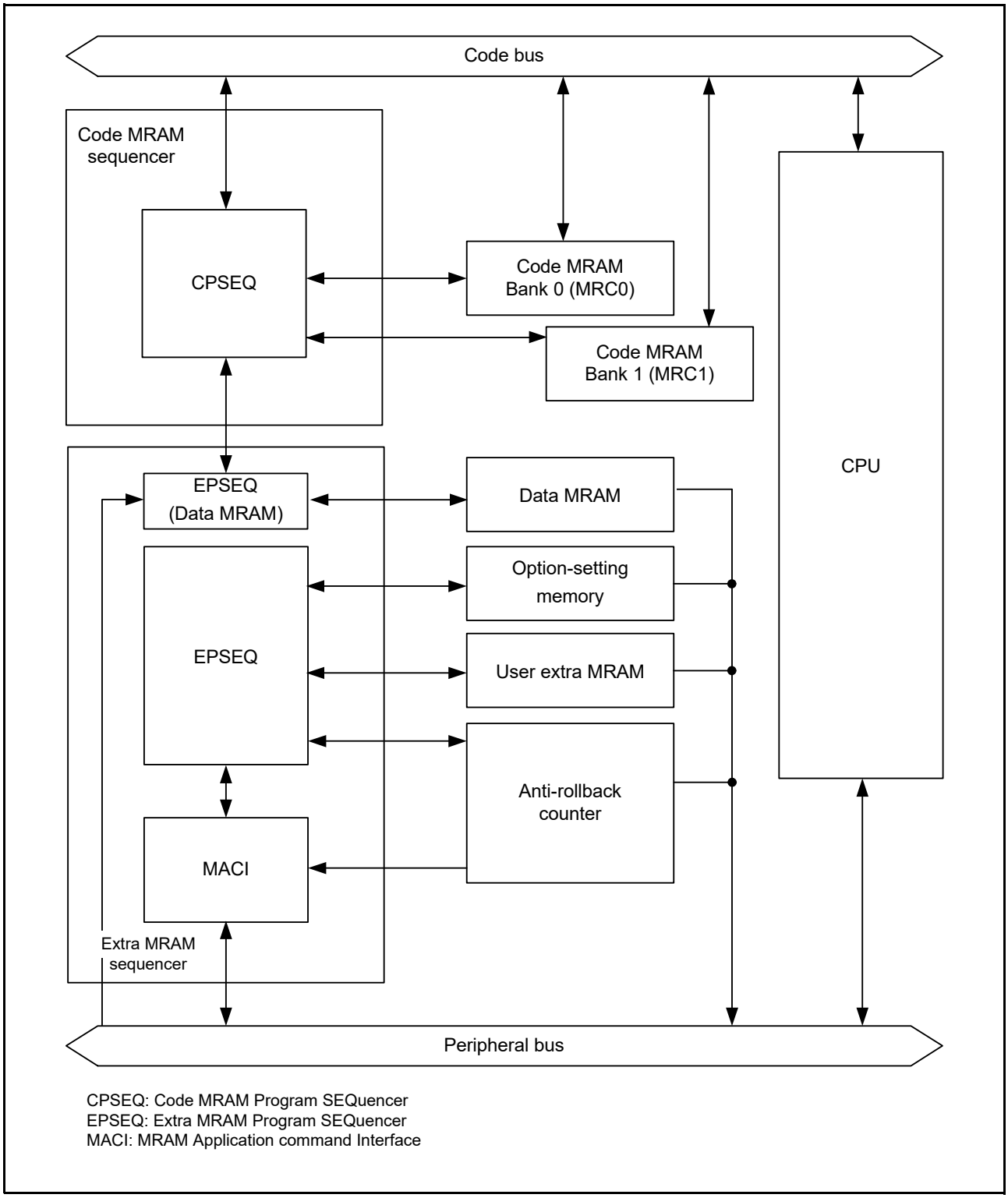


Figure 65.1 Block Diagram of MRAM Related Modules

65.2 Structure of Memory

Figure 65.2 shows the memory map of code MRAM in linear mode. Figure 65.3 shows the memory map of code MRAM in dual mode. The code MRAM can be used as 2 bank areas by using the dual bank function. This dual-bank structure allows a safe update of a program while a user program is running.

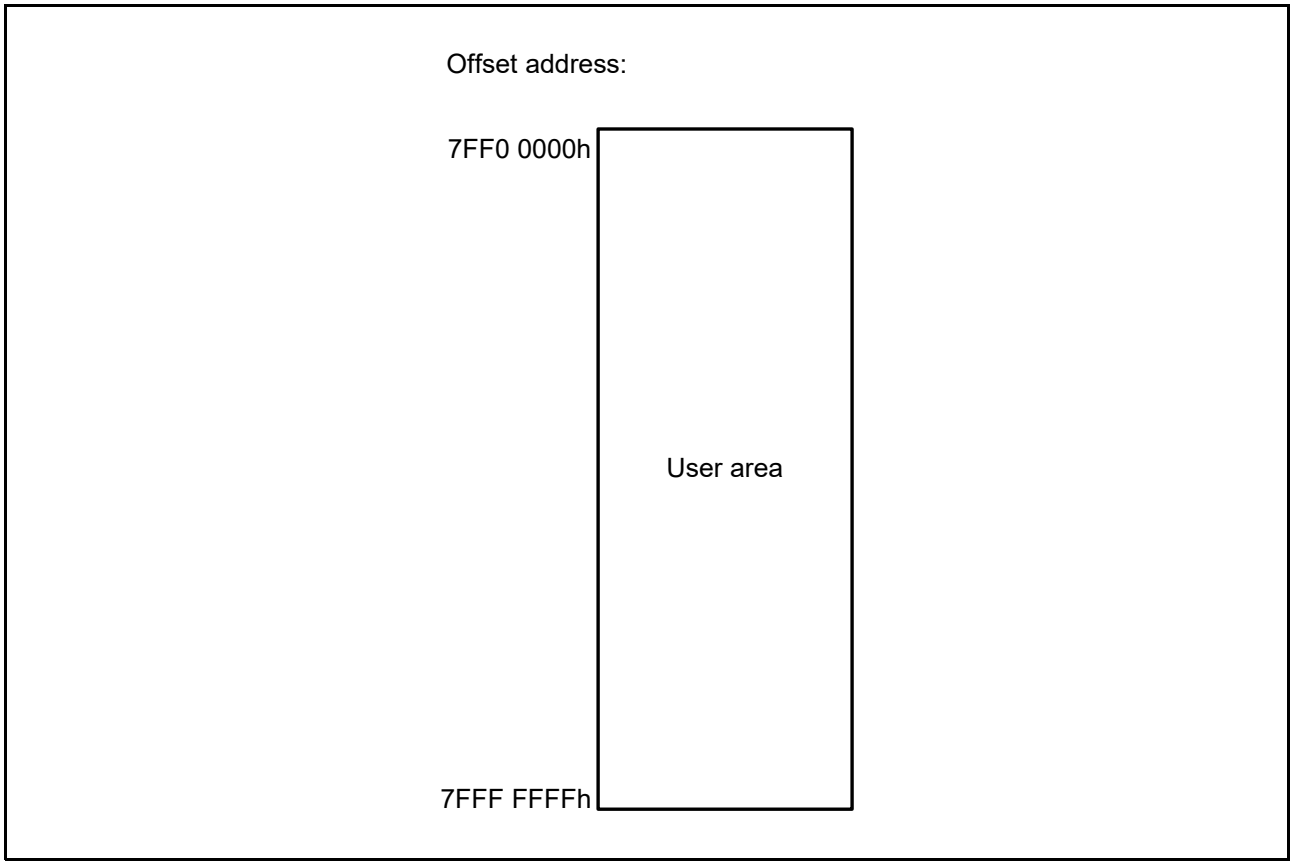


Figure 65.2 Memory Map of the Code MRAM in Linear Mode

Table 65.2 Read and Programming Address by Product for the Code MRAM (Linear Mode)

Code MRAM Memory Capacity	Address
1 Mbyte product	Secure area: FFF0 0000h to FFFF FFFFh MCR1: FFF0 0000h to FFF7 FFFFh MCR0: FFF8 0000h to FFFF FFFFh Non-secure area: 7FF0 0000h to 7FFF FFFFh MCR1: 7FF0 0000h to 7FF7 FFFFh MCR0: 7FF8 0000h to 7FFF FFFFh
512 Kbytes product	Secure area: FFF8 0000h to FFFF FFFFh MCR1: FFF8 0000h to FFFB FFFFh MCR0: FFFC 0000h to FFFF FFFFh Non-secure area: 7FF8 0000h to 7FFF FFFFh MCR1: 7FF8 0000h to 7FFB FFFFh MCR0: 7FFC 0000h to 7FFF FFFFh

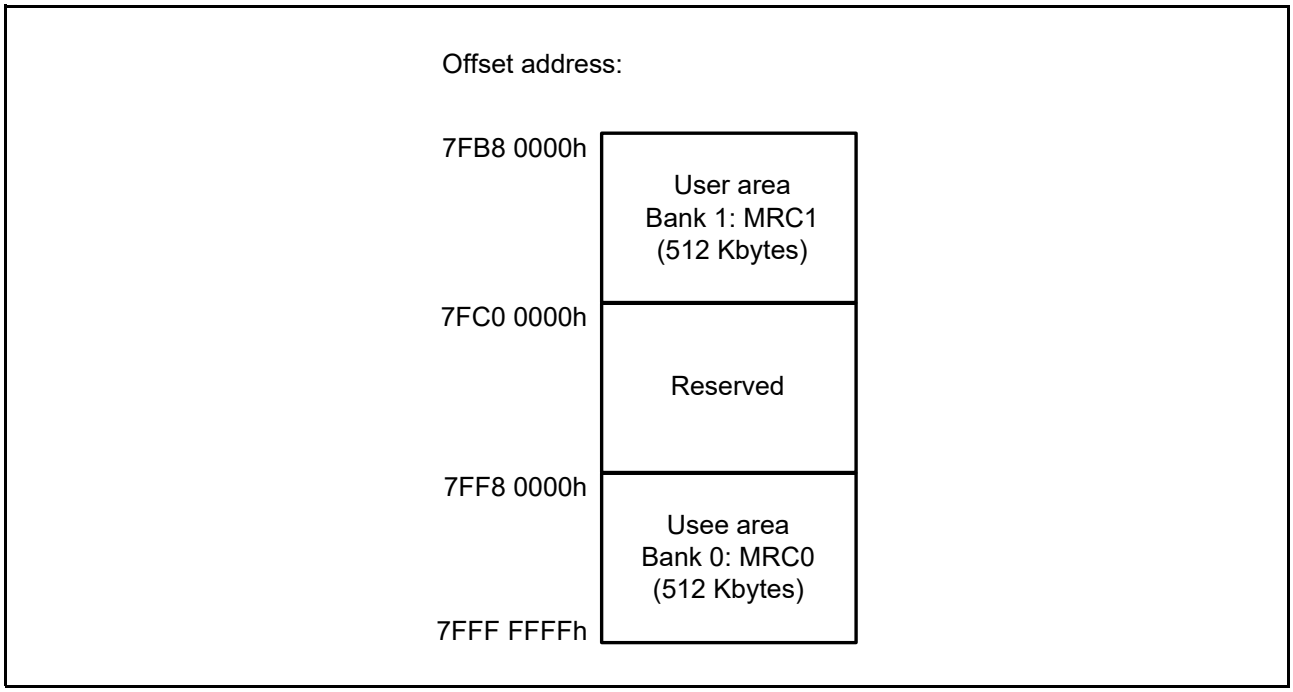


Figure 65.3 Memory Map of the Code MRAM in Dual Mode

Table 65.3 Read and Programming Address by Product for the Code MRAM (Dual Mode (BANKSEL.BANKSWP[2:0] = 111b))

Code MRAM Memory Capacity	Address
Products with 1 Mbyte of MRAM, Bank 1	Secure area (MRC1): FFB8 0000h to FFBF FFFFh Non-secure area (MRC1): 7FB8 0000h to 7FBF FFFFh
Products with 1 Mbyte of MRAM, Bank 0	Secure area (MRC0): FFF8 0000h to FFFF FFFFh Non-secure area (MRC0): 7FF8 0000h to 7FFF FFFFh
Products with 512 Kbytes of MRAM, Bank 1	Secure area (MRC1): FFBC 0000h to FFBF FFFFh Non-secure area (MRC1): 7FBC 0000h to 7FBF FFFFh
Products with 512 Kbytes of MRAM, Bank 0	Secure area (MRC0): FFFC 0000h to FFFF FFFFh Non-secure area (MRC0): 7FFC 0000h to 7FFF FFFFh

Figure 65.4 shows the memory map of the data MRAM and user extra MRAM. The user extra MRAM can only be accessed as a secure area. The user extra MRAM is divided into a 128-Kbyte user area and a 96-byte general-purpose OTP area (GPOTP).

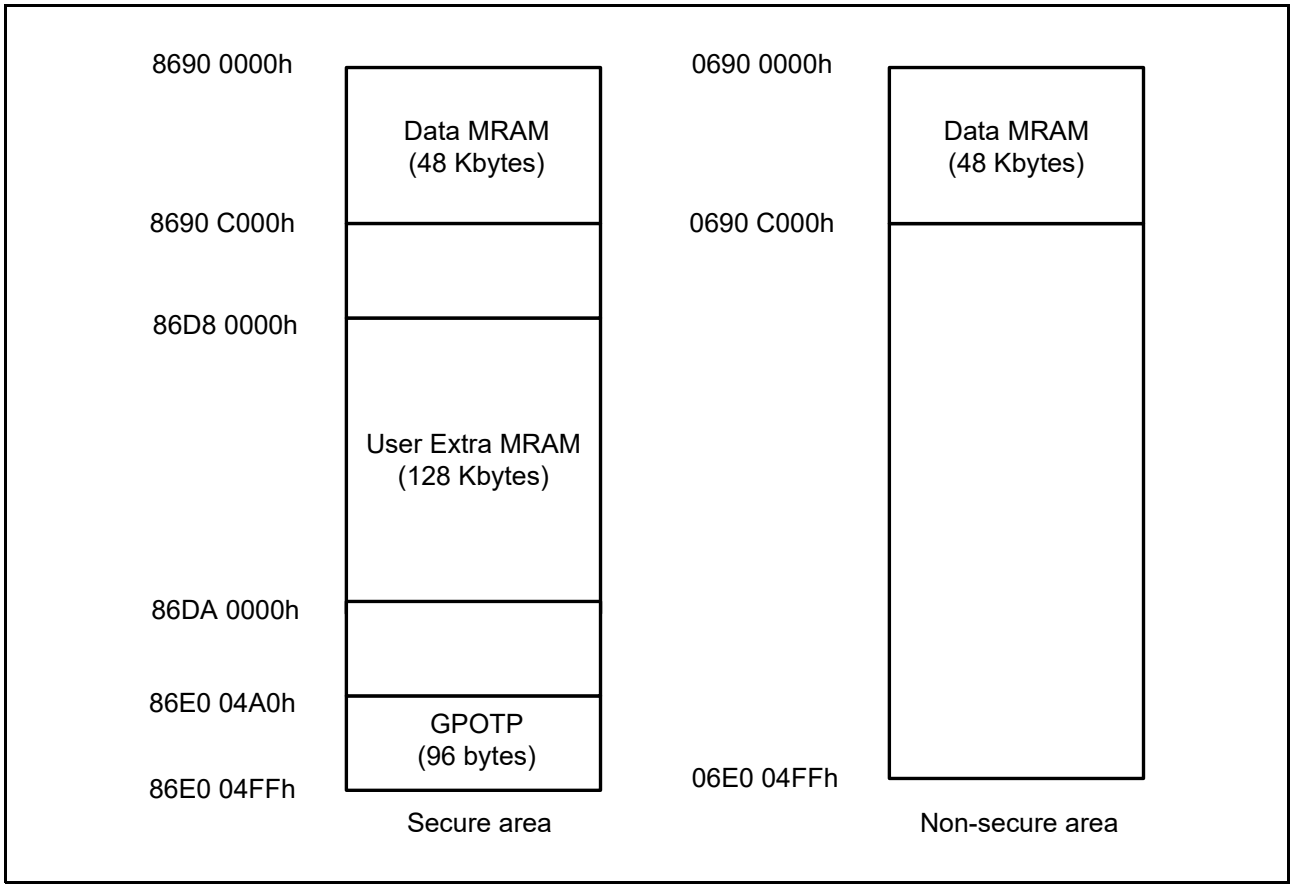


Figure 65.4 Memory Map of the Data MRAM and User Extra MRAM

65.3 Address Space

Using the hardware interface with MRAM requires access to all registers of the hardware. This is so that MACI commands can be issued. Table 65.4 to Table 65.6 list the base addresses.

Table 65.4 The Base Address of the Extra MRAM

Base Address Symbol	Address	Secure Alias
BASE_ME	8600 0000h or 0600 0000h	Secure or non-secure
BASE_MES	8600 0000h	Secure
BASE_MEN	0600 0000h	Non-secure

Table 65.5 The Base Address of the Code MRAM

Base Address Symbol	Address	Secure Alias
BASE_MC	FF00 0000h or 7F00 0000h	Secure or non-secure

Table 65.6 MACI Command Issuing Area

Area	Address	Capacity	Secure Alias
MACI command issuing area	8708 0000h	4 bytes	Secure
	0708 0000h		Non-secure

65.4 MRAM Read and Program Function

65.4.1 Prefetch Buffer

Prefetch buffer speeds up CPU instruction fetches from bus master to code MRAM.

Table 65.7 Prefetch Buffer Overview

Function	Implementation
Target Region	Linear mode: 7FF0 0000h to 7FFF FFFFh (1 Mbyte) FFF0 0000h to FFFF FFFFh (1 Mbyte) Dual mode: 7FB8 0000h to 7FBF FFFFh (512 Kbytes) 7FF8 0000h to 7FFF FFFFh (512 Kbytes) FFB8 0000h to FFBF FFFFh (512 Kbytes) FFF8 0000h to FFFF FFFFh (512 Kbytes)
Target Bus Master	CPU instruction fetch
Capacity	64 Bytes/CPU
Associativity	Full associative
	256 bits/entry (256 bits aligned data), 2 entries
Access Cycle	Hit: 0 wait Miss: See section 65.5.2, Code MRAM Frequency Notifications Register (MRCFREQ)

65.4.2 Code MRAM Programming Procedure

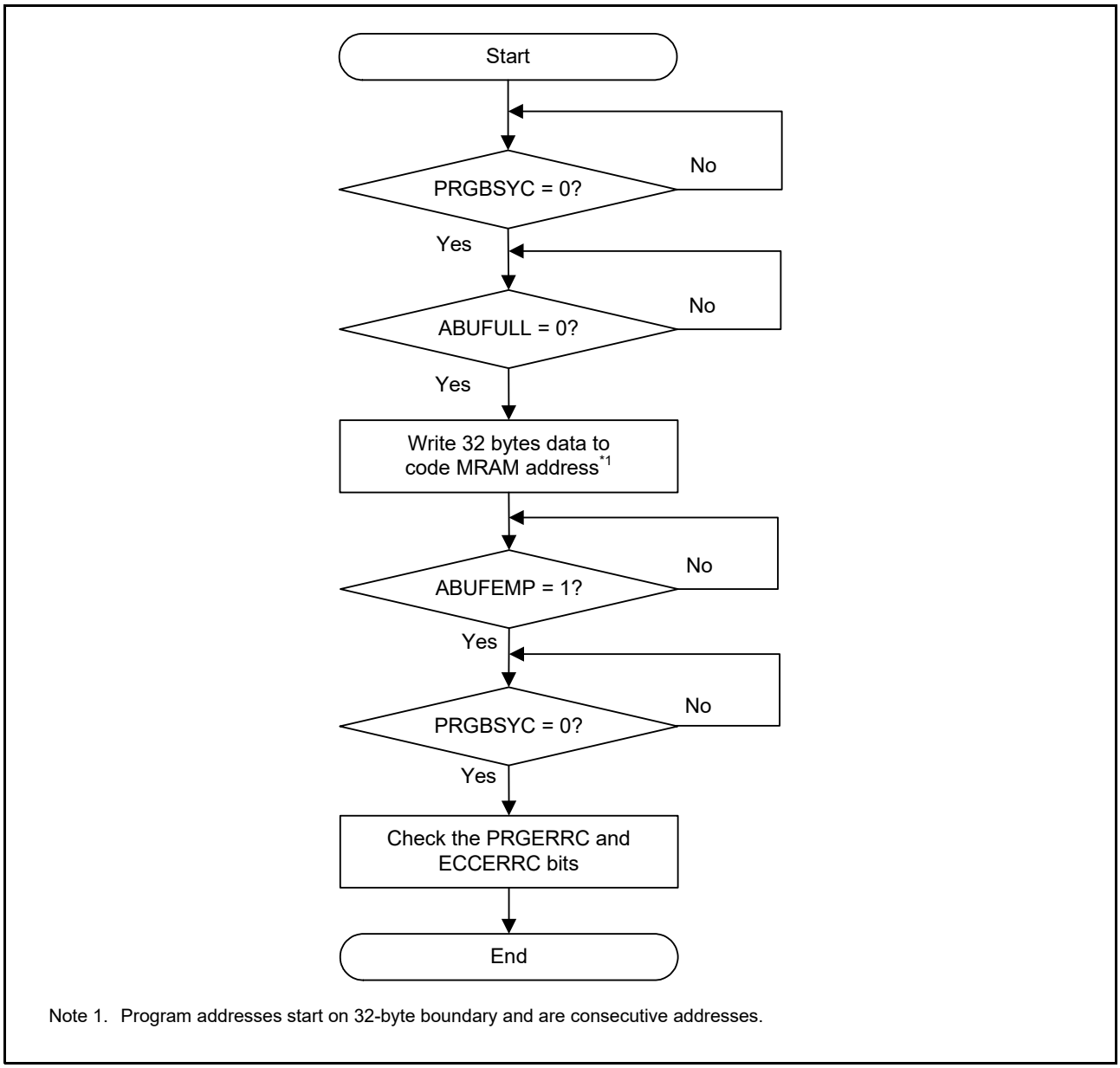
Program data for code MRAM is entered to program buffer by bus access and is written to code MRAM in 32 bytes units. It is recommended the following to avoid system bus stall while code MRAM program processing.

- Only one bus master can perform code MRAM write access
- Program data is 32 bytes or less in the same 32 bytes boundary.

The following describes the procedure for programming the MRC0 area. When programming the MRC1 area, replace the bit names in the description with those assigned to the MRC1 area. Figure 65.5 and Figure 65.6 show the procedure for programming the MRC0 area. Figure 65.7 and Figure 65.8 show the procedure for programming the MRC1 area. Before starting to program the MRC0 area, check the PRGBSYC and ABUFFULL flags in the MRCPS register to ensure that no other programming is in progress. The data to be written to the MRC0 area are stored in the 32-byte program data buffer and are written to the MRC0 area when any of the following conditions is satisfied.

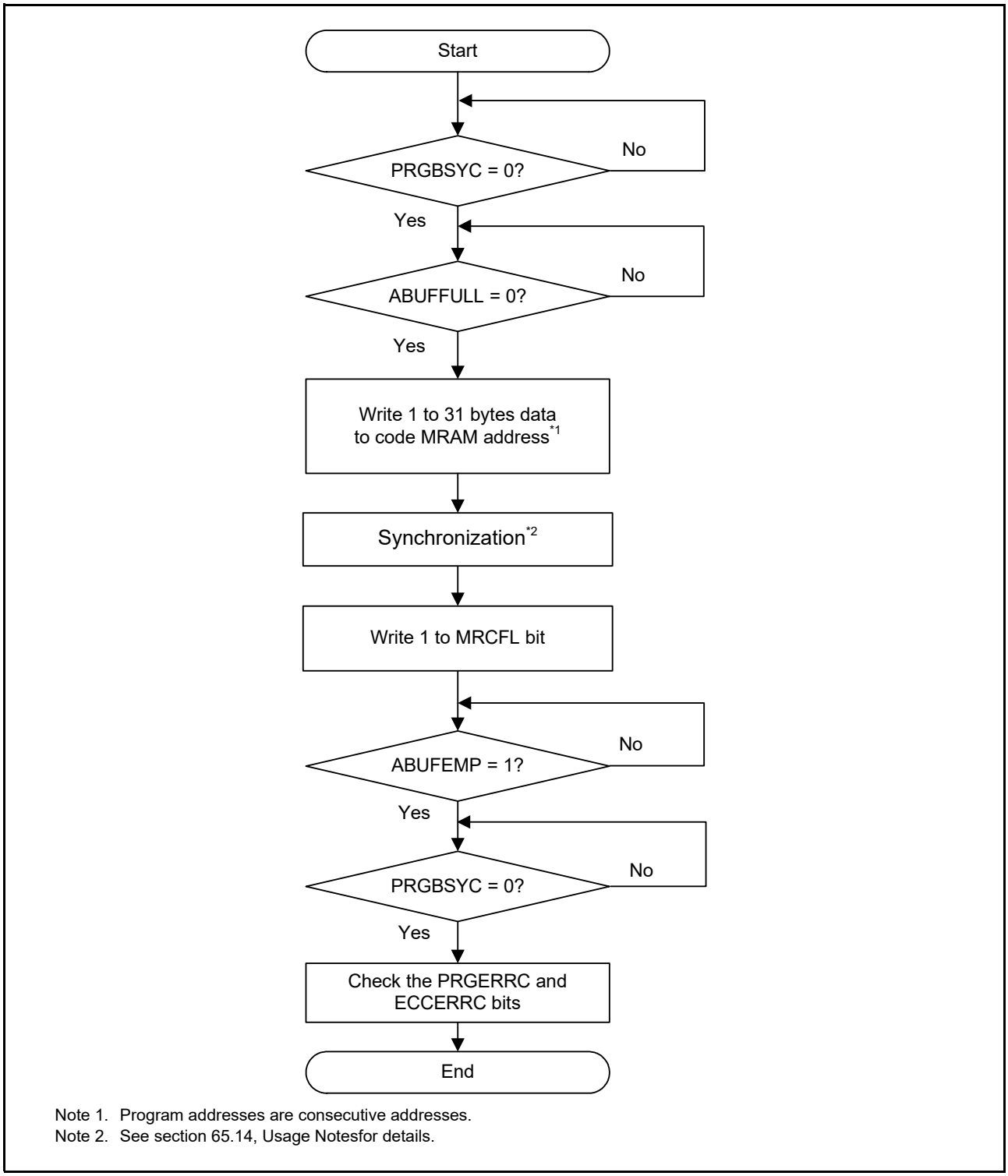
- Program data buffer is full (Figure 65.5)
- The address of the MRC0 area in the code MRAM to be written to is beyond the current 32-byte boundary.
- MRCFL bit in MRCFLR register is set to 1 (Figure 65.6)

Program processing error status can be confirmed by PRGERRC and ECCERRC bits in MRCPS register.



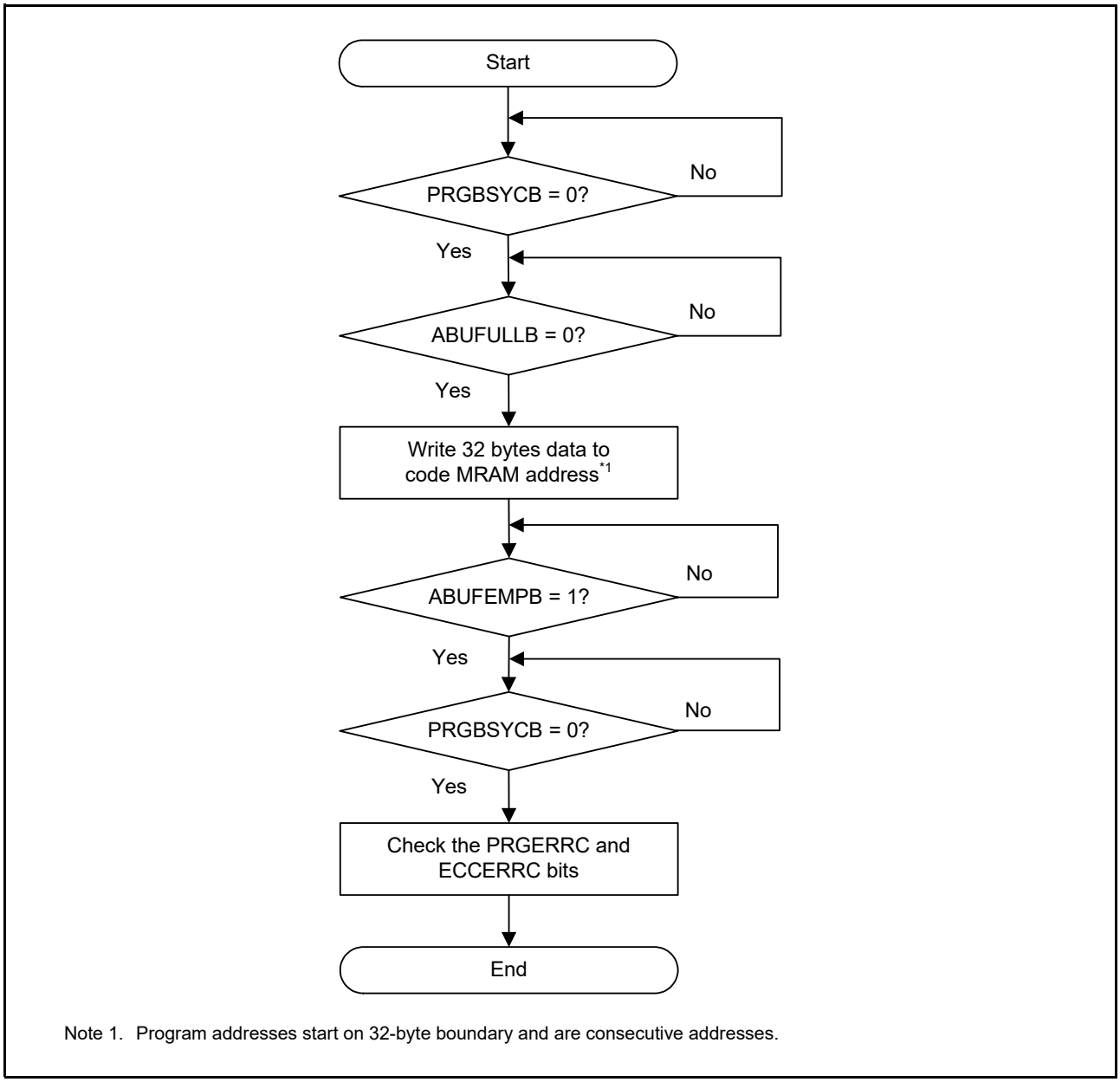
Note 1. Program addresses start on 32-byte boundary and are consecutive addresses.

Figure 65.5 Programming Procedure in MRC0 Area (32-Byte Programming)



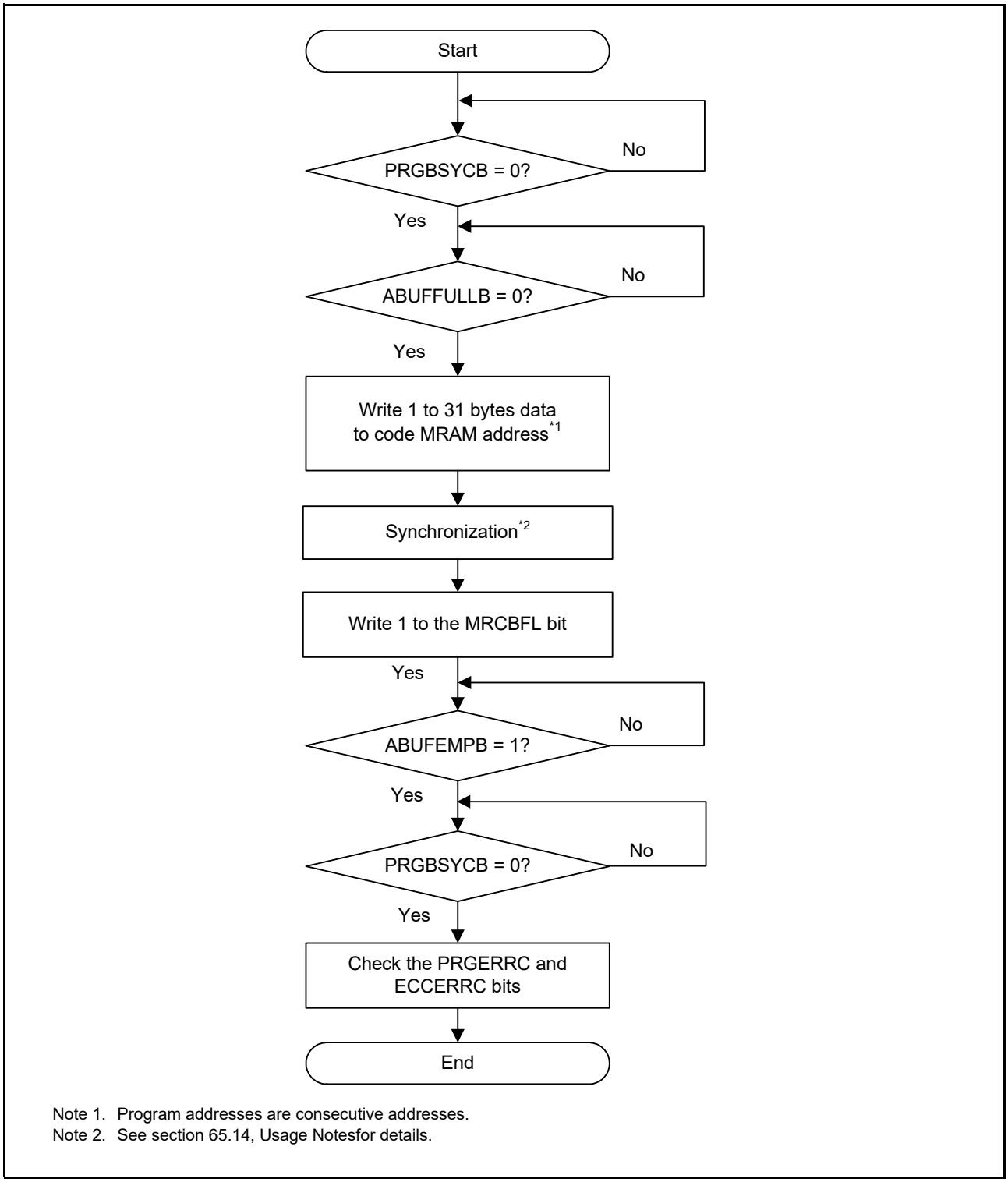
Note 1. Program addresses are consecutive addresses.
Note 2. See section 65.14, Usage Notesfor details.

Figure 65.6 Programming Procedure in MRC0 Area (Smaller than 32-Byte Programming)



Note 1. Program addresses start on 32-byte boundary and are consecutive addresses.

Figure 65.7 Programming Procedure in MRC1 Area (32-Byte Programming)



Note 1. Program addresses are consecutive addresses.
Note 2. See section 65.14, Usage Notesfor details.

Figure 65.8 Programming Procedure in MRC1 Area (Smaller than 32-Byte Programming)

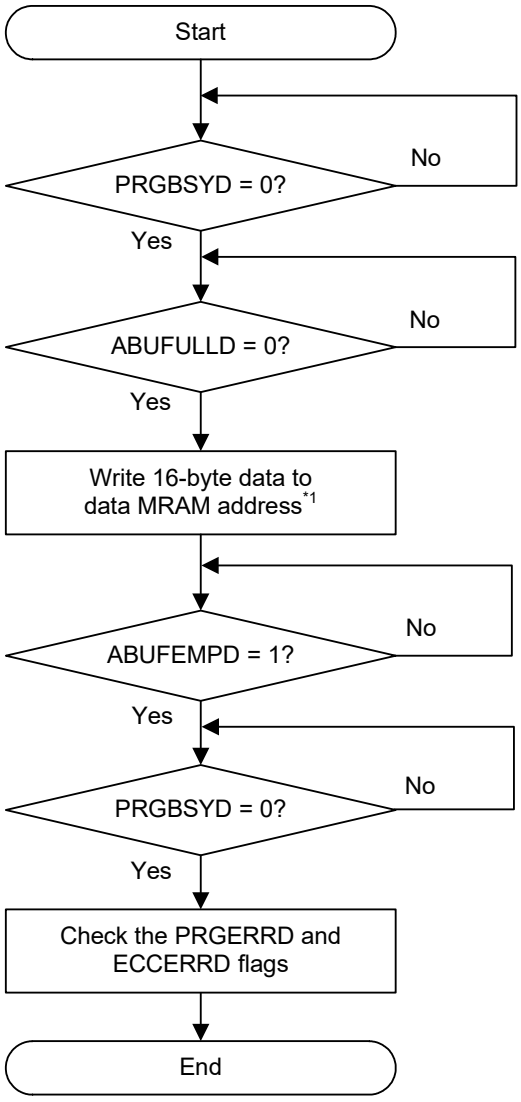
65.4.3 Data MRAM Programming Procedure

Program data for the data MRAM is input to the program buffer via bus access and written to the data MRAM in 16-byte units. The followings are recommended to avoid system bus stalls during the data MRAM programming operations:

- Only one bus master can perform write access to the data MRAM.
- Program data should be 16 bytes or less on the same 16-byte boundary.

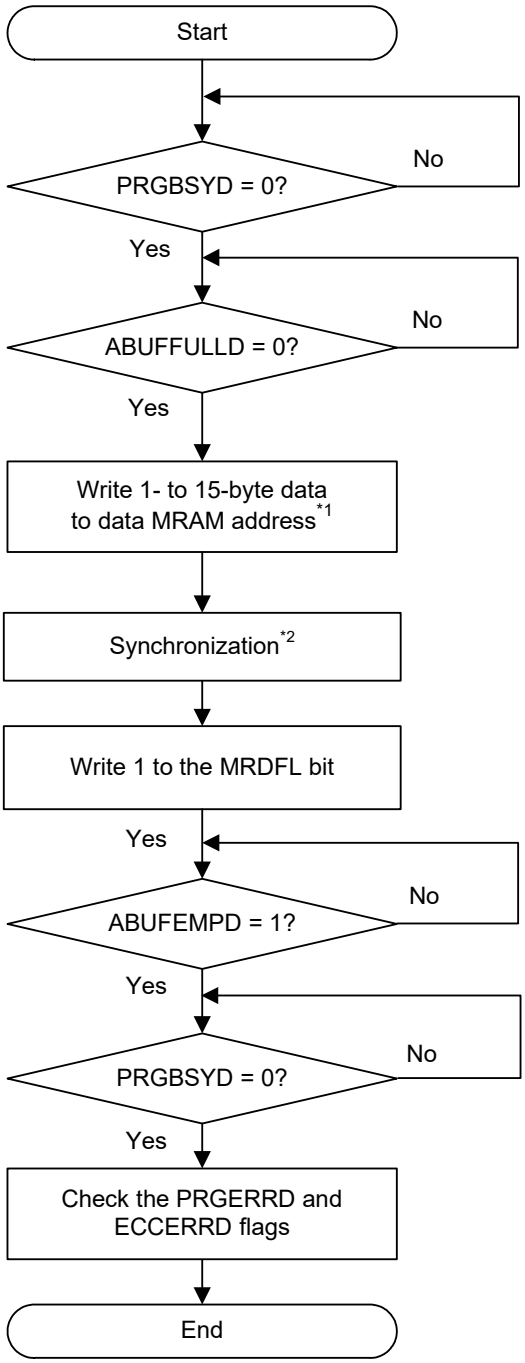
The programming procedures of data MRAM are shown in Figure 65.9 and Figure 65.10. Before writing to data MRAM, confirm that other program processing is not performed by the PRGBSYD and ABUFFULLD flags in the MRDPS register. All program processing is performed by writing to the data MRAM and writing 1 to the MRDFL bit in the MRDFLR register. Completion of programming can be confirmed by checking the MRDFL bit in the MRDFLR register and the ABUFEMPD and PRGBSYD flags in the MRDPS register.

The program processing error status can be confirmed by the PRGERRD and ECCERRD flags in the MRDPS register.



Note 1. Program addresses start on 16-byte boundary and are consecutive addresses.

Figure 65.9 Programming Procedure in the Data MRAM (16-Byte Programming)



Note 1. Program addresses are consecutive addresses.
Note 2. See section 65.14, Usage Notes for details.

Figure 65.10 Programming Procedure in the Data MRAM (Smaller than16-Byte Programming)

65.4.4 MRAM Programming Procedure with MRAM Program Ready Interrupt

Figure 65.11, Figure 65.12, and Figure 65.13 show the procedure for programming the MRC or MRD using a DMA transfer start interrupt request. If the DMAC performs a write to MRAM during a code MRAM or data MRAM programming process, the system bus stalls until the code MRAM or data MRAM programming is complete.

Therefore, to perform efficient software operations (including the DMAC), it is recommended that you operate the DMAC according to the procedures in Figure 65.11, Figure 65.12, and Figure 65.13. However, these steps may still cause the system bus to stall when:

- Another bus master performs programming to the code MRAM or data MRAM regardless of the interrupt request
- Performing more than one DMA write transfer per interrupt request

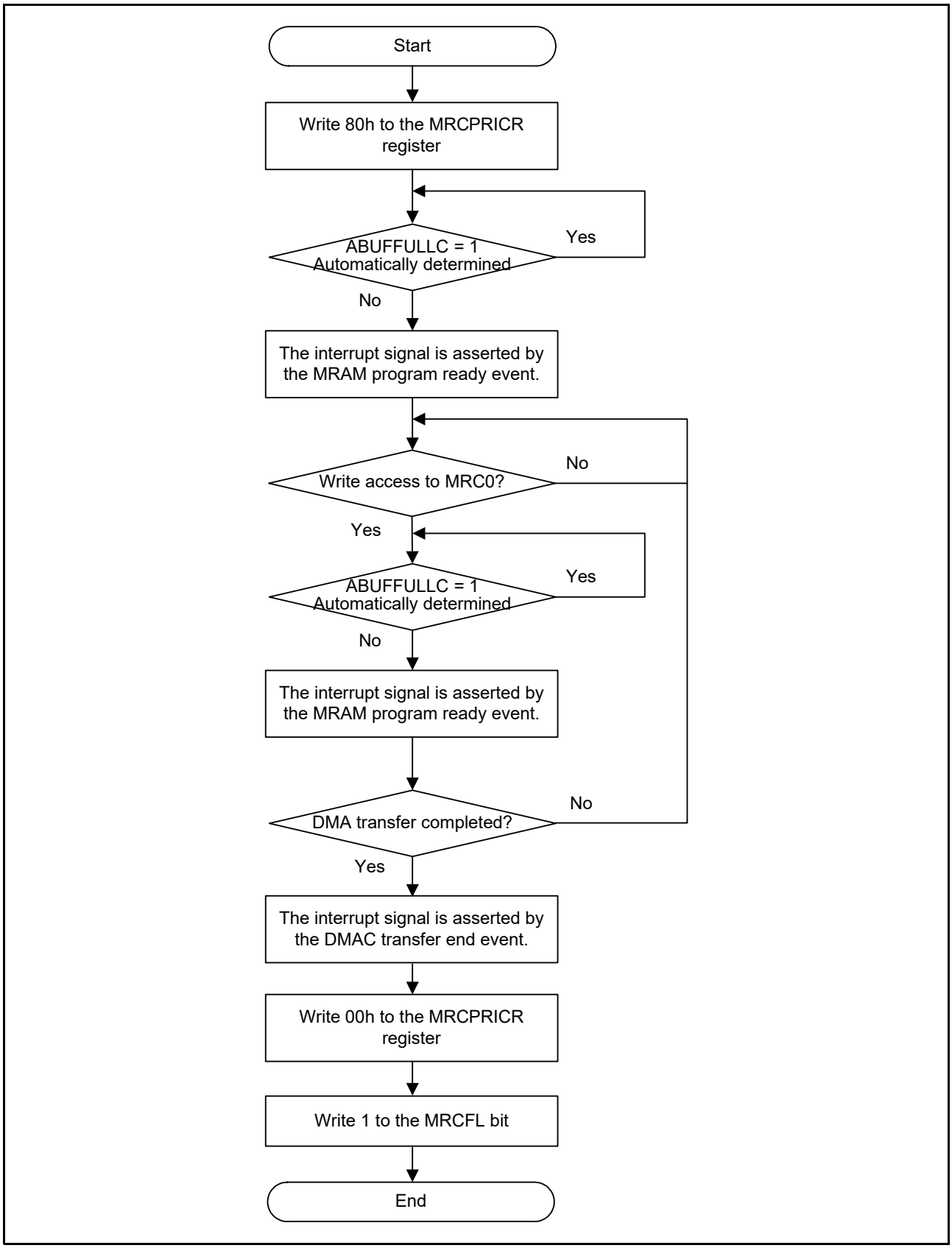


Figure 65.11 Code MRAM (MRC0) Programming Procedure with DMAC

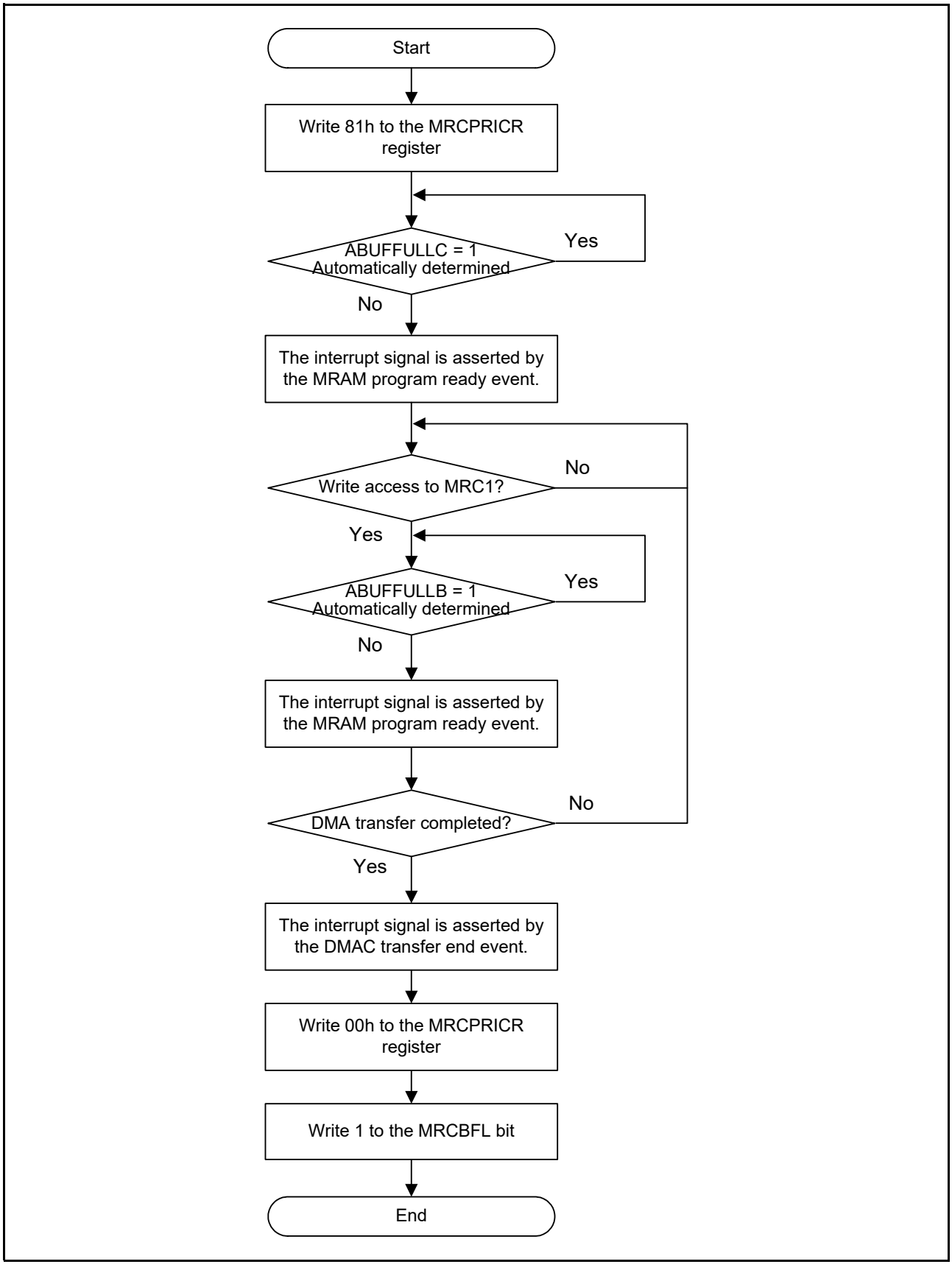


Figure 65.12 Code MRAM (MRC1) Programming Procedure with DMAC

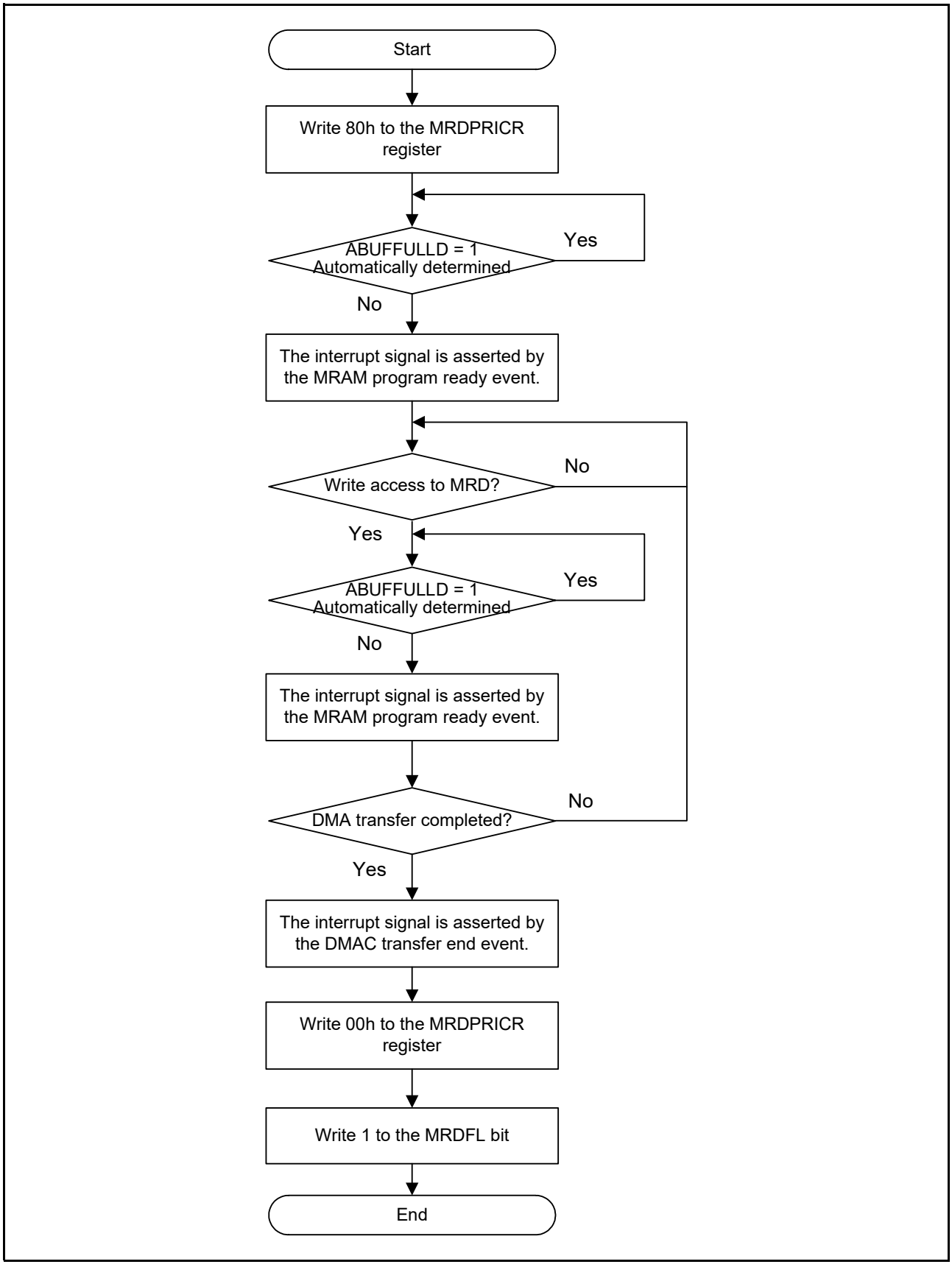
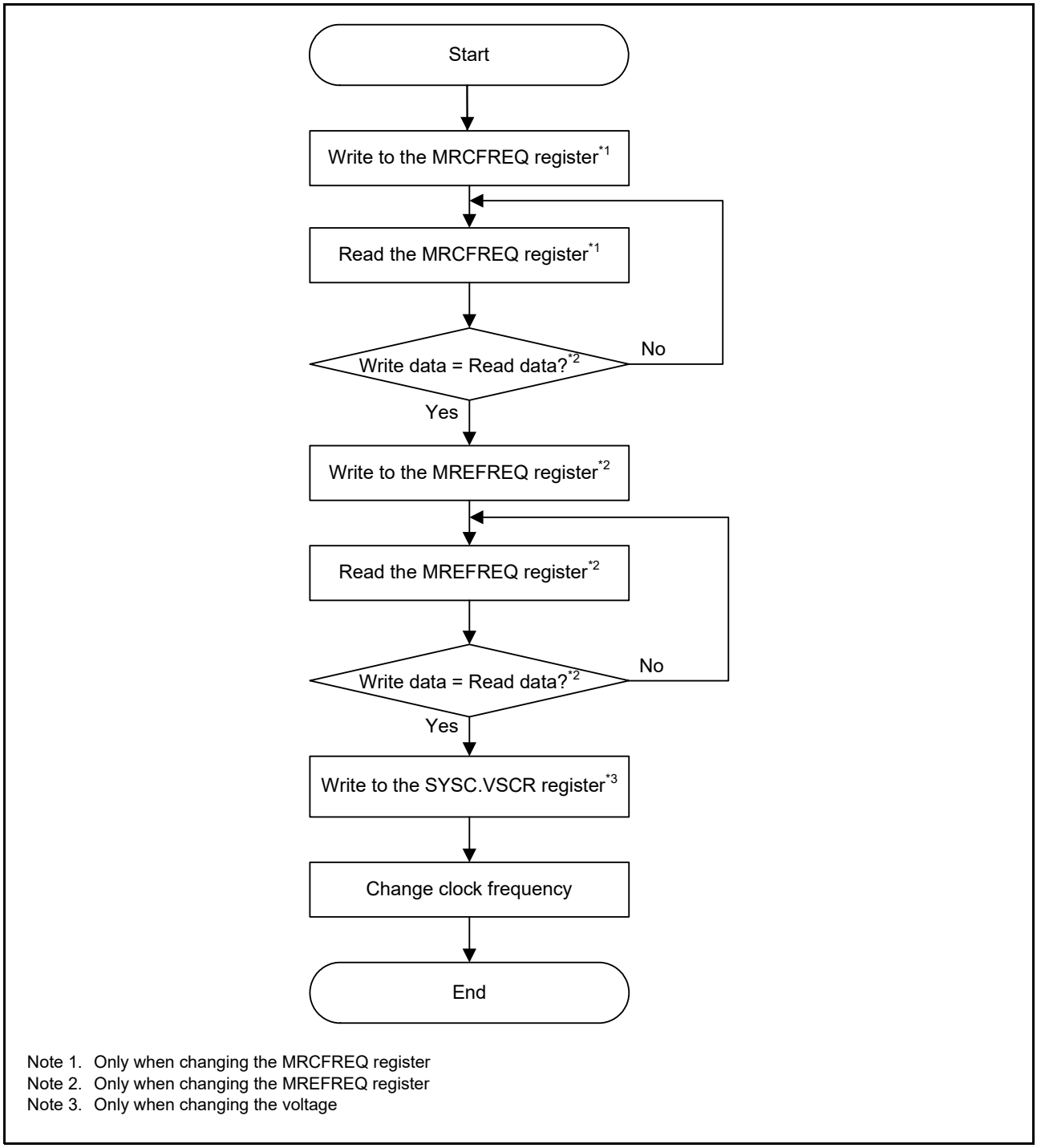


Figure 65.13 Data MRAM Programming Procedure with DMAC

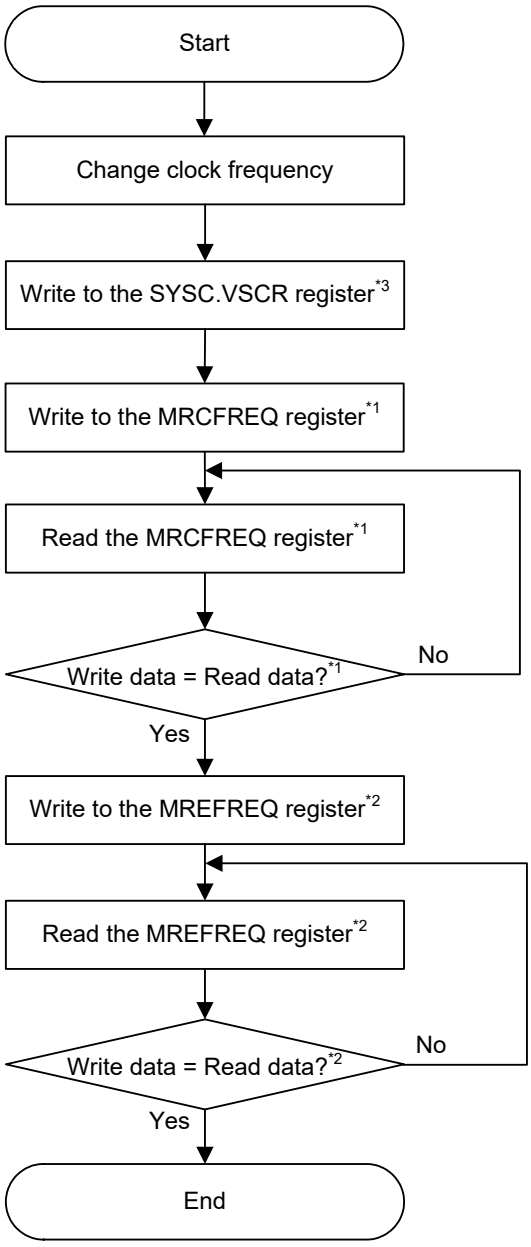
65.4.5 Frequency Change Procedure

The frequency setting is not reflected even if the register is written while programming or reading MRAM. To change the clock frequency, check that the register value is set to the write value before changing it. The frequency change procedures for code MRAM and extra MRAM are shown in Figure 65.14 and Figure 65.15.



Note 1. Only when changing the MRCFREQ register
Note 2. Only when changing the MREFREQ register
Note 3. Only when changing the voltage

Figure 65.14 Frequency Up Procedure



Note 1. Only when changing the MRCFREQ register
Note 2. Only when changing the MREFREQ register
Note 3. Only when changing the voltage

Figure 65.15 Frequency Down Procedure

65.5 Register Descriptions

65.5.1 Code MRAM Prefetch Buffer Enable Register (MRCPFB)

Address(es): MRAM.MRCPFB 8709 C000h, MRAM_NS.MRCPFB 0709 C000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MPFBEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPFBEN	Code MRAM Prefetch Buffer Enable	0: Prefetch buffer is disabled 1: Prefetch buffer is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

MPFBEN Bit (Code MRAM Prefetch Buffer Enable)

This bit is used to enable or disable code MRAM prefetch buffer.

65.5.2 Code MRAM Frequency Notifications Register (MRCFREQ)

Address(es): MRAM.MRCFREQ 8709 C004h, MRAM_NS.MRCFREQ 0709 C004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MRCMHZ[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	MRCMHZ[9:0]	Code MRAM Frequency Notifications	Set the operating frequency of the code MRAM by rounding it up in MHz and converting it to a binary number. MRCMHZ[9:0] ≤ 064h: No cycles of waiting 064h < MRCMHZ[9:0] ≤ 0C8h: 1 cycle of waiting 0C8h < MRCMHZ[9:0] ≤ 12Ch: 2 cycles of waiting Settings other than above are prohibited.	R/W*1
b23 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	Key Code	Key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 32-bit units with the KEY[7:0] bits set to 1Eh.
Note 2. Written values are not retained by these bits. These bits are read as 0.

The MRCFREQ register is used to set the read and program frequency for the code MRAM. For a faster clock frequency, set the MRCMHZ[9:0] bits before changing the clock frequency. To slow down the clock frequency, set the MRCMHZ[9:0] bits after changing the clock frequency. For information on the frequency setting, see section 65.4.5, Frequency Change Procedure.

MRCMHZ[9:0] Bits (Code MRAM Frequency Notifications)

These bits specify the operating frequency of the code MRAM. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits. If the operating frequency is 32.768 KHz or less, set the MRCMHZ[9:0] bits to 000h.

Example 1: The desired frequency is 249.4 MHz.
Round up the first decimal place of 249.4 to obtain the nearest whole number (= 250). Specify the value in binary (0FAh) in these bits.

Example 2: The desired frequency is 0.4 MHz.
Round up the first decimal place of 0.4 to obtain the nearest whole number (= 1). Specify the value in binary (001h) in these bits.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the MRCMHZ[9:0] bits.

65.5.3 Extra MRAM Frequency Notifications Register (MREFREQ)

Address(es): MRAM.MREFREQ 8709 C008h, MRAM_NS.MREFREQ 0709 C008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MREMHZ[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	MREMHZ[7:0]	Extra MRAM Frequency Notifications	Set the operating frequency of the extra MRAM by rounding it up in MHz and converting it to a binary number. MREMHZ[7:0] ≤ 64h: No cycles of waiting 64h < MREMHZ[7:0] ≤ 96h: 1 cycle of waiting Settings other than above are prohibited.	R/W*1
b23 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	Key Code	Key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 32-bit units with the KEY[7:0] bits set to E1h.
Note 2. Written values are not retained by these bits. These bits are read as 0.

The MREFREQ register is used to set the read and program frequency of extra MRAM. For a faster clock frequency, set the MREMHZ[7:0] bits before changing the clock frequency. To slow down the clock frequency, set the MREMHZ[7:0] bits after changing the clock frequency. For information on the frequency setting, see section 65.4.5, Frequency Change Procedure.

MREMHZ[7:0] Bits (Extra MRAM Frequency Notifications)

These bits specify the operating frequency of the extra MRAM. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits. If the operating frequency is 32.768 KHz or less, set the MREMHZ[7:0] bits to 000h.

Example 1: The desired frequency is 124.4 MHz.
Round up the first decimal place of 124.4 to obtain the nearest whole number (= 125). Specify the value in binary (7Dh) in these bits.

Example 2: The desired frequency is 0.4 MHz.
Round up the first decimal place of 0.4 to obtain the nearest whole number (= 1). Specify the value in binary (01h) in these bits.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the MREMHZ[7:0] bits.

65.5.4 Code MRAM ECC Decoder Control Register (MRCDECC)

Address(es): MRAM.MRCDECC 8709 C010h, MRAM_NS.MRCDECC 0709 C010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	ECCSELC	DECDISC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECDISC	Code MRAM ECC Decoder Disable	0: ECC decoder is enabled 1: ECC decoder is disabled	R/W*1
b1	ECCSELC	Code MRAM ECC Bits Select	0: ECC bits unselect 1: ECC bits select	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to this bit is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 5Ah.
Note 2. Written values are not retained by these bits. These bits are read as 0.

The MRCDECC register is used to test the ECC decoder circuits for the code MRAM. For information on the ECC decoder circuits test flow, see section 65.9.6, Fault Correction and Detection.

DECDISC Bit (Code MRAM ECC Decoder Disable)

This bit selects to enable or disable the ECC decoder circuit in code MRAM.
[Setting condition]

- When 1 is written while the write enabling condition is satisfied.

[Clearing condition]

- When 0 is written while the write enabling condition is satisfied.

ECCSELC Bit (Code MRAM ECC Bits Select)

When the setting of the ECCSELC bit is 1, the ECC bits of code MRAM are output to the [144:128] and [16:0] bits and 0 is output to the [255:145] and [127:17] bits of the read data. When an DEC error occurs, the corrected data is output. Read data is not guaranteed when a TED error is detected.
[Setting condition]

- When 1 is written while the write enabling condition is satisfied.

[Clearing condition]

- When 0 is written while the write enabling condition is satisfied.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the DECDISC and ECCSELC bits.

65.5.5 Code MRAM Read Access Error Interrupt Enable Register (MRCRAEINT)

Address(es): MRAM.MRCRAEINT 8709 C014h, MRAM_NS.MRCRAEINT 0709 C014h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	INTENBTC	INTENBDC
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTENBDC	Code MRAM DEC Error Interrupt Enable	0: Generation of an MRCRDI interrupt request is disabled when the MRCRAES.DECERRC flag is set to 1. 1: Generation of an MRCRDI interrupt request is enabled when the MRCRAES.DECERRC flag is set to 1.	R/W
b1	INTENBTC	Code MRAM TED Error Interrupt Enable	0: Generation of an MRCRDI interrupt request is disabled when the MRCRAES.TEDERRC flag is set to 1. 1: Generation of an MRCRDI interrupt request is enabled when the MRCRAES.TEDERRC flag is set to 1.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRCRAEINT register enables or disables generation of a code MRAM ECC error (MRCRDI) interrupt request.

INTENBDC Bit (Code MRAM DEC Error Interrupt Enable)

This bit enables or disables the generation of an MRCRDI interrupt request when a code MRAM read access violation occurs causing the DECERRC flag in the MRCRAES register to become 1.

INTENBTC Bit (Code MRAM TED Error Interrupt Enable)

This bit enables or disables the generation of an MRCRDI interrupt request when a code MRAM read access violation occurs causing the TEDERRC flag in the MRCRAES register to become 1.

65.5.6 Code MRAM Read Access Error Status Register (MRCRAES)

Address(es): MRAM.MRCRAES 8709 C018h, MRAM_NS.MRCRAES 0709 C018h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TEDERRC	DECERRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECERRC	Code MRAM DEC Error Detect Flag	0: DEC error not detected 1: DEC error detected	R/W *1
b1	TEDERRC	Code MRAM TED Error Detect Flag	0: TED error not detected 1: TED error detected	R/W *1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. 0 can be written to clear the flag only after 1 has been read.

DECERRC Flag (Code MRAM DEC Error Detect Flag)

This bit indicates the result of DEC error check for read data.

[Setting condition]

- Error is detected to read data of bus masters (This bit is not set when an error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

TEDERRC Flag (Code MRAM TED Error Detect Flag)

This bit indicates the result of TED error check for read data.

[Setting condition]

- Error is detected to read data of bus masters (This bit is not set when an error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

65.5.7 Code MRAM TED Error Address Register (MRCRTEA)

Address(es): MRAM.MRCRTEA 8709 C01Ch, MRAM_NS.MRCRTEA 0709 C01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRCRTEA[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRCRTEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0.	R
b31 to b5	MRCRTEA[26:0]	Code MRAM Read Access TED Error Address	Indicates address when TED error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register stores the address when a TED error occurs during a code MRAM read. The address of ECC error is stored when the MRCRAES.TEDERRC flag is set to 1. The MRCRTEA[26:0] bits do not capture the error address when a TED error occurs until the MRCRAES.TEDERRC flag becomes 0.

MRCRTEA[26:0] Bits (Code MRAM Read Access TED Error Address)

[Setting condition]

- When the TEDERRC flag in the MRCRAES register changes from 0 to 1.

65.5.8 Code MRAM DEC Error Address Register (MRCRDEA)

Address(es): MRAM.MRCRDEA 8709 C020h, MRAM_NS.MRCRDEA 0709 C020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRCRDEA[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRCRDEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0.	R
b31 to b5	MRCRDEA[26:0]	Code MRAM Read Access DEC Error Address	Indicates address when DEC error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register stores the address when a DEC error occurs during a code MRAM read. The address of ECC error is stored when the MRCRAES.DECERRC flag is set to 1. The MRCRDEA[26:0] bits do not capture the error address when a DEC error occurs until the MRCRAES.DECERRC flag becomes 0.

MRCRDEA[26:0] Bits (Code MRAM Read Access DEC Error Address)

[Setting condition]

- When the DECERRC flag in the MRCRAES register changes from 0 to 1.

65.5.9 Extra MRAM ECC Decoder Control Register (MREDECC)

Address(es): MRAM.MREDECC 8709 C030h, MRAM_NS.MREDECC 0709 C030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	ECCSELE	DECDISE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECDISE	Extra MRAM ECC Decoder Disable	0: ECC decoder enable 1: ECC decoder disable	R/W*1
b1	ECCSELE	Extra MRAM ECC Bits Select	0: ECC bits unselect 1: ECC bits select	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to this bit is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to A5h.
Note 2. Written values are not retained by these bits. These bits are read as 0.

DECDISE Bit (Extra MRAM ECC Decoder Disable)

When this bit is set to 1 and the MRE is read, the data before error correction is output.

ECCSELE Bit (Extra MRAM ECC Bits Select)

If the ECCSELE bit is 1, the ECC bits of the MRE are output to bits [16:0] as read data, and 0 is output to bits [127:17]. If a DEC error occurs, the data after the error correction is output. If a TED error is detected, the read data is not guaranteed.

65.5.10 Extra MRAM Read Access Error Interrupt Enable Register (MRERAEINT)

Address(es): MRAM.MRERAEINT 8709 C034h, MRAM_NS.MRERAEINT 0709 C034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	INTENBTE	INTENBDE
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTENBDE	Extra MRAM DEC Error Interrupt Enable	0: Generation of an MRERDI interrupt request is disabled when the MRERAES.DECERRE flag is set to 1. 1: Generation of an MRERDI interrupt request is enabled when the MRERAES.DECERRC flag is set to 1.	R/W
b1	INTENBTE	Extra MRAM TED Error Interrupt Enable	0: Generation of an MRERDI interrupt request is disabled when the MRERAES.TEDERRE flag is set to 1. 1: Generation of an MRERDI interrupt request is enabled when the MRERAES.TEDERRE flag is set to 1.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRERAEINT register enables or disables generation of a Extra MRAM ECC error (MRERDI) interrupt request.

INTENBDE Bit (Extra MRAM DEC Error Interrupt Enable)

This bit enables or disables generation of an MRERDI interrupt request when a extra MRAM read access violation occurs causing the DECERRE flag in the MRERAES register to become 1.

INTENBTE Bit (Extra MRAM TED Error Interrupt Enable)

This bit enables or disables generation of an MRERDI interrupt request when a extra MRAM read access violation occurs causing the TEDERRE flag in the MRERAES register to become 1.

65.5.11 Extra MRAM Read Access Error Status Register (MRERAES)

Address(es): MRAM.MRERAES 8709 C038h, MRAM_NS.MRERAES 0709 C038h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TEDERRE	DECERRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECERRE	Extra MRAM DEC Error Detected	0: DEC error not detected 1: DEC error detected	R/W*1
b1	TEDERRE	Extra MRAM TED Error Detected	0: TED error not detected 1: TED error detected	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. 0 can be written to clear the flag only after 1 has been read.

DECERRE Flag (Extra MRAM DEC Error Detected)

This bit indicates the result of DEC error check for read data.

[Setting condition]

- Error is detected to read data of bus masters (This flag is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this flag.

TEDERRE Flag (Extra MRAM TED Error Detected)

This bit indicates the result of TED error check for read data.

[Setting condition]

- Error is detected to read data of bus masters (This flag is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this flag.

65.5.12 Extra MRAM TED Error Address Register (MRERTEA)

Address(es): MRAM.MRERTEA 8709 C03Ch, MRAM_NS.MRERTEA 0709 C03Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRERTEA[27:12]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRERTEA[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b31 to b4	MRERTEA[27:0]	Extra MRAM Read Access TED Error Address	Indicates address when TED error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register stores the address when a TED error occurs during an extra MRAM read. The address of ECC error is stored when the MRERAES.TEDERRE flag is set to 1. The MRERTEA[27:0] bits do not capture the error address when a TED error occurs until the MRERAES.TEDERRE flag becomes 0.

MRERTEA[27:0] Bits (Extra MRAM Read Access TED Error Address)

[Setting condition]

- When the TEDERRE flag in the MRERAES register changes from 0 to 1.

65.5.13 Extra MRAM DEC Error Address Register (MRERDEA)

Address(es): MRAM.MRERDEA 8709 C040h, MRAM_NS.MRERDEA 0709 C040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRERDEA[27:12]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRERDEA[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b31 to b4	MRERDEA[27:0]	Extra MRAM Read Access DEC Error Address	Indicates address when DEC error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

This register stores the address when a DEC error occurs during an extra MRAM read. The address of ECC error is stored when the MRERAES.DECERRE flag is set to 1. The MRERDEA[27:0] bits do not capture the error address when a DEC error occurs until the MRERAES.DECERRE flag becomes 0.

MRERDEA[27:0] Bits (Extra MRAM Read Access DEC Error Address)

[Setting condition]

- When the DECERRE flag in the MRERAES register changes from 0 to 1.

65.5.14 MRAM Security Attribution Register (MSAR)

Address(es): MRAM.MSAR 8709 C100h, MRAM_NS.MSAR 0709 C100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRCPSEQSA	MREPSEQSA	MRCPSA	MRDPSA	MACITRSA	MACICMRSA	MACICMISA	—	—	—	—	MPFBENSA	MRCFREQSA	MRCECCSA	MREFREQSA	MREECCSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MREECCSA	Extra MRAM ECC Register Security Attribution	0: Secure 1: Non-secure	R/W
b1	MREFREQSA	MREFREQ Register Security Attribution	0: Secure 1: Non-secure	R/W
b2	MRCECCSA	Code MRAM ECC Register Security Attribution	0: Secure 1: Non-secure	R/W
b3	MRCFREQSA	MRCFREQ Register Security Attribution	0: Secure 1: Non-secure	R/W
b4	MPFBENSA	MRCPFB Register Security Attribution	0: Secure 1: Non-secure	R/W
b8 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	MACICMISA	MACI Command Issuing Security Attribution	0: Secure 1: Non-secure	R/W
b10	MACICMRSA	MACI Command Registers Security Attribution	0: Secure 1: Non-secure	R/W
b11	MACITRSA	MACI Transfer Security Attribution	0: Secure 1: Non-secure	R/W
b12	MRDPSA	Data MRAM Program Register Security Attribution	0: Secure 1: Non-secure	R/W
b13	MRCPSA	Code MRAM Program Register Security Attribution	0: Secure 1: Non-secure	R/W
b14	MREPSEQSA	EPSEQ Area Register Security Attribution	0: Secure 1: Non-secure	R/W
b15	MRCPSEQSA	CPSEQ Area Register Security Attribution	0: Secure 1: Non-secure	R/W

Note: This register can only be written by secure transactions. Writes by non-secure transactions are ignored, but no Protect Zone access errors are generated. It can be read by any transaction.
Note: Write access is invalid when PRCR_S.PRC4*1 = 0
Note 1. Refer to the target specifications for the register write protection function of the PRCR_S.PRC4 bit.

MREECCSA Bit (Extra MRAM ECC Register Security Attribution)

This bit sets the security attribute of the MREDECC, MRERAEINT, MRERAES, MRERTEA, and MRERDEA registers.

MREFREQSA Bit (MREFREQ Register Security Attribution)

This bit sets the security attribute of the MREFREQ register.

MRCECCSA Bit (Code MRAM ECC Register Security Attribution)

This bit sets the security attribute of the MRCDECC, MRCRAEINT, MRCRAES, MRCRTEA, and MRCRDEA registers.

MRCFREQSA Bit (MRCFREQ Register Security Attribution)

This bit sets the security attribute of the MRCFREQ register.

MPFBENSA Bit (MRCPFB Register Security Attribution)

This bit sets the security attribute of the MCPFB register.

MACICMISA Bit (MACI Command Issuing Security Attribution)

This bit sets the security attribute of the MRAM access area.

MACICMRSA Bit (MACI Command Registers Security Attribution)

This bit sets the security attribute of the MASTAT, MREPAEINT, MRDYIE, MSADDR, MSTATR, MENTRYR, MSUINITR, MCMDR, and MSUASMON registers.

MACITRSA Bit (MACI Transfer Security Attribution)

This bit sets the security attribute of the MCTRCNTR, and MCTRSTATR registers.

MRDPSA Bit (Data MRAM Program Register Security Attribution)

This bit sets the security attribute of the MRDPS, MRDPAEINT, MRDPEA, and MRDPRICR registers.

MRCPSA Bit (Code MRAM Program Register Security Attribution)

This bit sets the security attribute of the MRCPCNS, MRCBPROTNS, MRCPS, MRCPAEINT, MRCPEA, and MRCPRICR registers.

MREPSEQSA Bit (EPSEQ Area Register Security Attribution)

This bit sets the security attribute of the MRPSC and MREEECC registers.

MRCPSEQSA Bit (CPSEQ Area Register Security Attribution)

This bit sets the security attribute of the MRCEECC register.

65.5.15 Extra MRAM Zeroization Status Register (MREZS)

Address(es): MRAM.MREZS 8709 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	WHUKEXE	WHUKZF
Value after reset:	1	1	1	1	1	1	0	1/0*1

Bit	Symbol	Bit Name	Description	R/W
b0	WHUKZF	W-HUK Zero Status Flag	0: The W-HUK has been zeroized. 1: The W-HUK has not been zeroized.	R
b1	WHUKEXE	W-HUK Zeroization Executing Status Flag	0: W-HUK zeroization not performed 1: W-HUK zeroization in progress	R
b7 to b2	—	Reserved	These bits are read as 1.	R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. The initial value is changed by the status of W-HUK.

WHUKZF Flag (W-HUK Zero Status Flag)

See section 65.8.6, HUK Protection for the update conditions of the WHUKZF flag.

[Setting condition]

- When W-HUK is released from the reset state without zeroing.

[Clearing condition]

- When W-HUK zeroization is complete.

WHUKEXE Flag (W-HUK Zeroization Executing Status Flag)

[Setting condition]

- When W-HUK zeroization is started.

[Clearing condition]

- When W-HUK zeroization is complete.

65.5.16 Extra MRAM Zeroization Control Register (MREZC)

Address(es): MRAM.MREZC 8709 C404h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	WHUKZE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	WHUKZE[2:0]	W-HUK Zeroization Execute	0XX, X0X, XX0: W-HUK zeroization not performed 111: W-HUK zeroization performed	R/W *1, *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W *3

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 3Ch.

Note 2. The MREZC register cannot be written in the following conditions.
- When the W-HUK pseudo-zeroing operation in Table 65.34 has already been completed and pseudo-zeroing is to be started again by register writing without any system resets.
- If the operation of W-HUK zeroing in Table 65.34 has already been completed.

Note 3. Written values are not retained by these bits. These bits are read as 0.

WHUKZE[2:0] Bits (W-HUK Zeroization Execute)

0 cannot be written to these bits.

[Setting condition]

- 1 is written to while write enabling conditions are satisfied.

[Clearing condition]

- When W-HUK zeroization is complete.

65.5.17 Extra MRAM Access Status Register (MASTAT)

Address(es): MRAM.MASTAT 8709 E010h, MRAM_NS.MASTAT 0709 E010h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CMDLK	MREAE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	MREAE	Extra MRAM Access Violation Flag	0: No extra MRAM access violation has occurred. 1: A extra MRAM access violation has occurred.	R/W *1
b4	CMDLK	Command Lock Flag	0: The MRAM sequencer is not in the command-locked state. 1: The MRAM sequencer is in the command-locked state.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. 0 can be written to clear the flag only after 1 has been read.

The MASTAT register indicates whether extra MRAM access violation has occurred. If any of the CMDLK and MREAE flags is set to 1, the extra MRAM sequencer enters the command-locked state (see section 65.9.2, Error Protection). To release it from the command-locked state, the status clear or forced stop command must be issued to the extra MRAM sequencer.

MREAE Flag (Extra MRAM Access Violation Flag)

This flag indicates whether extra MRAM access violation occurred. When this flag is set to 1, the ILGLERR flag in the MSTATR register is set to 1, placing the extra MRAM sequencer in the command-locked state.

[Setting conditions]

In general, commands are issued in extra MRAM program mode as described below.

- When the area of extra MRAM beyond the address specified by the MSADDR register is treated as reserved
- When an MRAM sequencer command is issued to the non-secure attribute extra MRAM

[Clearing conditions]

- When 0 is written to this flag after this flag is set to 1.
- When the extra MRAM sequencer starts processing the status clear or forced stop command.

CMDLK Flag (Command Lock Flag)

This flag indicates that the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- When the extra MRAM sequencer detects an error and enters the command-locked state.

[Clearing condition]

- When the extra MRAM sequencer starts processing the status clear or forced stop command.

65.5.18 Extra MRAM Access Error Interrupt Enable Register (MPAEINT)

Address(es): MRAM.MPAEINT 8709 E014h, MRAM_NS.MPAEINT 0709 E014h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CMDLKIE	MREAEIE	—	—	—
Value after reset:	0	0	0	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	MREAEIE	Extra MRAM Access Violation Interrupt Enable	0: Generation of an MREPRI interrupt request is disabled when the MASTAT.MREAE flag is set to 1. 1: Generation of an MREPRI interrupt request is enabled when the MASTAT.MREAE flag is set to 1.	R/W
b4	CMDLKIE	Command Lock Interrupt Enable	0: Generation of an MREPRI interrupt request is disabled when the MASTAT.CMDLK flag is set to 1. 1: Generation of an MREPRI interrupt request is enabled when the MASTAT.CMDLK flag is set to 1.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MPAEINT register enables or disables generation of an extra MRAM access error (MREPRI) interrupt request.

MREAEIE Bit (Extra MRAM Access Violation Interrupt Enable)

This bit enables or disables generation of an MREPRI interrupt request when a extra MRAM access violation occur causing the MREAE flag in the MASTAT register to become 1.

CMDLKIE Bit (Command Lock Interrupt Enable)

This bit enables or disables generation of an MREPRI interrupt request when the extra MRAM sequencer enters the command-locked state causing the CMDLK flag in the MASTAT register to become 1.

65.5.19 Extra MRAM Ready Interrupt Enable Register (MRDYIE)

Address(es): MRAM.MRDYIE 8709 E018h, MRAM_NS.MRDYIE 0709 E018h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MRDYIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRDYIE	Extra MRAM Ready Interrupt Enable	0: Generation of an MRDYI interrupt request is disabled 1: Generation of an MRDYI interrupt request is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRDYIE register enables or disables generation of an extra MRAM ready (MRDYI) interrupt.

MRDYIE Bit (Extra MRAM Ready Interrupt Enable)

This bit is used to enable or disable generation of an MRDYI interrupt request when the MRDY flag in the MSTATR register is changed from 0 to 1 on completion of processing by the extra MRAM sequencer of program and configuration set command.

65.5.20 MACI Command Start Address Register (MSADDR)

Address(es): MRAM.MSADDR 8709 E030h, MRAM_NS.MSADDR 0709 E030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSADDR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MSADDR[31:0]	Start Address for MACI Command Processing	Start Address for MACI Command Processing	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Writing to these bits is possible only when the MSTATR.MRDY flag is 1. Writing to these bits while the MSTATR.MRDY flag = 0 is ignored. The value of the lower 4 bits (b3 to b0) is fixed to 0000b.

The MSADDR register specifies the address where the target area for command processing starts when the MACI command for program or configuration set is issued.
The MSADDR register is initialized when the MSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

MSADDR[31:0] Bits (Start Address for MACI Command Processing)

These bits specify the start address for MACI command processing. Bits [30:25] are ignored in MACI command processing for extra MRAM. Bit [31] is used for the security area for extra MRAM sequencer command. When bit [31] is 1, the extra MRAM sequencer can update the secure area with secure access. When bit [31] is 0, the extra MRAM sequencer can update the non-secure area with secure and non-secure access. Bits corresponding to address bits of lower order than the corresponding boundary listed above are also ignored. Table 65.8 shows the address boundary for each command.
See the user's manual for information on the addresses of the extra MRAM.
See Table 65.17 and Table 65.18 for information on the addresses of program and configuration set commands.

Table 65.8 Address Boundary for Each of the Commands

Command	Address Boundary
Program (extra MRAM)	16-byte
Configuration set (extra MRAM)	16-byte

65.5.21 MRAM Counter Select Register (MCNTSELR)

Address(es): MRAM.MCNTSELR 8709 E048h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CNTSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CNTSEL[2:0]	Counter Select	b2 b0 0 0 0: Anti-rollback counter is not selected. 0 0 1: ARCS is selected. Counter size is 1024 bits. 0 1 0: ARCOEMBL is selected. Counter size is 64 bits. 0 1 1: Anti-rollback counter is not selected. 1 x x: ARCNS is selected. Counter configuration is 256 or 1024 bits. See Table 65.9 and Table 65.10 for details on counter select condition.	R/W *1, *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. This bit can be written when MSTATR.MRDY = 1. Writing to this bit is ignored when MSTATR.MRDY = 0.
Note 2. Writing to these bits is possible only when this register is written in 8-bit units.

CNTSEL[2:0] Bits (Counter Select)

The CNTSEL[2:0] bits specifies 3 types of anti-rollback counter (ARCS, ARCNS or ARCOEMBL) in the increment counter or read counter command.

The MCNTSELR value is initialized when the SUINIT bit in MSUINITR is set to 1.

Before using the ARCNS counter, it is necessary to set whether the counter is to be used with 4 lines × 256 bits or 1 line × 1024 bits using the program command. The bits to be set are CNFARCNS[1:0] bits. If the CNFARCNS[1] bit is set to 1, the ARCNS counter cannot be used. Once the counter type is decided, it cannot be changed.

Table 65.9 Select Condition of ARCNS (256 Bits) in Increment/Read Counter Command

MCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARCNS0[255:0]	Counter number is 4. Counter size is 256 bits.
101	ARCNS1[255:0]	
110	ARCNS2[255:0]	
111	ARCNS3[255:0]	

Table 65.10 Select Condition of ARCNS (1024 Bits) in Increment Counter Command

MCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARCNS[1023:0]	Counter number is 1. Counter size is 1024 bits.
101		
110		
111		

65.5.22 MRAM Counter Data Register n (MCNTDTRn) (n = 0, 1)

Address(es): MRAM.MCNTDTR0 8709 E04Ch, MRAM.MCNTDTR1 8709 E050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CNTRDAT[31:16]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CNTRDAT[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	CNTRDAT[31:0]	Counter Read Data	32 bits (LSB side of 64-bit counter value) of read data are output to MCNTDTR0. 32 bits (MSB side of 64-bit counter value) of read data are output to MCNTDTR1.	R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

MCNTDTR0 and MCNTDTR1 indicate the 64-bit value that was read from the ARCOEMBL anti-rollback counter by the read counter command.

CNTRDAT[31:0] Bits (Counter Read Data)

[Condition for read data]

- MACI has accepted the read counter command.

[Clearing conditions]

- MACI has accepted extra MRAM sequencer command except for the read counter command.
- Extra MRAM sequencer enters command lock state.
- W-HUK zeroization is started.

65.5.23 MRAM Configuration Update Transfer Control Register (MCTRCNTR)

Address(es): MRAM.MCTRCNTR 8709 E060h, MRAM_NS.MCTRCNTR 0709 E060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	TRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRTRG	Transfer Start Trigger	0: Configuration update transfer process has not started. 1: Configuration update transfer process started	R/W *1, *2, *3
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Written values are not retained in these bits. These bits are read as 0.
Note 2. This bit can be written when the MCTRSTATR register is 00h. Writing to this bit is ignored when the MCTRSTATR register is other than 00h or when W-HUK zeroing is in progress.
Note 3. Writing to this bit is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 45h.

The MCTRCNTR register is used to start the configuration update transfer process.

TRTRG Bit (Transfer Start Trigger)

The TRTRG bit is used to start the configuration update transfer process.

[Setting condition]

- When the MCTRCNTR register is in write-enabled state and “4501h” is written to the MCTRCNTR register

65.5.24 MRAM Configuration Update Transfer Status Register (MCTRSTATR)

Address(es): MRAM.MCTRSTATR 8709 E06Ch, MRAM_NS.MCTRSTATR 0709 E06Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRMD	—	TRBUSY
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRBUSY	Transfer Busy Status	0: No configuration update transfer is in progress. 1: Configuration update transfer is in progress.	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	TRMD	Transfer Mode Setting Status	0: The MENTRYR register is in a setting that allows data transfer. 1: The MENTRYR register is not in a setting that allows data transfer.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

TRBUSY Bit (Transfer Busy Status)

Indicates configuration update transfer processing status.

[Setting condition]

- 1 is written to the MCTRCNTR.TRTRG bit.

[Clearing condition]

- Configuration update transfer processing is complete.

TRMD Bit (Transfer Mode Setting Status)

Indicates whether the MENTRYR register is in a state that allows a configuration update transfer to be started.

[Setting condition]

- The MENTRYR register is set to 0080h.
- When “AA80h” is written to the MENTRYR register during data MRAM programming processing

[Clearing condition]

- The MENTRYR register is set to 0000h.

65.5.25 Extra MRAM Status Register (MSTATR)

Address(es): MRAM.MSTATR 8709 E080h, MRAM_NS.MSTATR 0709 E080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	ILGCOMERR	—	—	OTERR	PZFERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRDY	ILGLERR	—	PRGERR	—	—	—	—	—	—	CFGSETERR	PROTERR	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b4	PROTERR	Protection Error Flag	0: Programming of and configuration settings for the protected area have been successfully completed. 1: An error occurred during programming of or configuration settings for the protected area.	R
b5	CFGSETERR	Configuration Set Error Flag	0: Configuration setting has been completed successfully. 1: An error has occurred during configuration setting.	R
b11 to b6	—	Reserved	These bits are read as 0.	R
b12	PRGERR	Programming Error Flag	0: Programming has been completed successfully. 1: An error has occurred during programming.	R
b13	—	Reserved	This bit is read as 0.	R
b14	ILGLERR	Illegal Command Error Flag	0: The extra MRAM sequencer has not detected an illegal MACI command or illegal extra MRAM access. 1: The extra MRAM sequencer has detected an illegal MACI command or illegal extra MRAM access.	R
b15	MRDY	Extra MRAM Ready Flag	0: Program, Configuration set, Increment Counter or Read Counter command processing is in progress. 1: None of the above is in progress.	R
b18 to b16	—	Reserved	These bits are read as 0.	R
b19	PZFERR	Protect Zone Filter Error Flag	0: A Protect Zone filter error has not been detected. 1: A Protect Zone filter error has been detected.	R
b20	OTERR	Other Error Flag	0: An error has not been detected. 1: An error has been detected.	R
b21	—	Reserved	This bit is undefined when read.	R
b22	—	Reserved	This bit is read as 0.	R
b23	ILGCOMERR	Illegal Command Error Flag	0: An illegal MACI command error has not been detected. 1: An illegal MACI command error has been detected.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MSTATR register indicates the state of the extra MRAM sequencer.

PROTERR Flag (Protection Error Flag)

The update area is permanently protected, and this flag indicates that there is a TED error in the data to be updated. If the PROTERR flag is 1, the extra MRAM sequencer is in a command lock state.
[Setting condition]

- Occurrence of an error while the extra MRAM sequencer was programming or making configuration settings for the protected area

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

CFGSETERR Flag (Configuration Set Error Flag)

This flag indicates the result of the extra MRAM configuration setting. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- When the extra MRAM sequencer reads data not to update from configuration area during configuration setting.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

PRGERR Flag (Programming Error Flag)

This flag indicates the result of the extra MRAM programming. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- See Table 65.42, Error Protection Types.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates that the extra MRAM sequencer has detected an illegal MACI command or extra MRAM access. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- See Table 65.42, Error Protection Types.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

MRDY Flag (Extra MRAM Ready Flag)

This flag indicates the command processing state of the extra MRAM sequencer.

[Setting conditions]

- After the extra MRAM sequencer completes command processing.
- After the extra MRAM sequencer starting to process a status clear or forced stop command
- When W-HUK zeroing is in processing

[Clearing conditions]

- After the extra MRAM sequencer has received the MACI command.
- The extra MRAM sequencer has received program and configuration set commands, and after the first write to the MACI command- issuing area
- For other commands, after the first write to the MACI command- issuing area.

PZFERR Flag (Protect Zone Filter Error Flag)

This flag indicates the Protect Zone filter error in the MACI command. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- See Table 65.42, Error Protection Types.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

OTERR Flag (Other Error Flag)

This flag indicates that an MACI command has been issued when the condition of accepting commands is not satisfied. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- See Table 65.42, Error Protection Types.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

ILGCOMERR Flag (Illegal Command Error Flag)

This flag indicates that the extra MRAM sequencer has detected an illegal MACI command. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- See Table 65.42, Error Protection Types.

[Clearing condition]

- After the extra MRAM sequencer starting to process a status clear or forced stop command

65.5.26 Extra MRAM Program Mode Entry Register (MENTRYR)

Address(es): MRAM.MENTRYR 8709 E084h, MRAM_NS.MENTRYR 0709 E084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								MENTRY	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MENTRY	Extra MRAM Program Mode Entry	0: Extra MRAM is in the read mode. 1: Extra MRAM is in the program mode.	R/W *1, *2, *3
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*4

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. These bits can be written when the MRDY flag in the MSTATR register is 1. Writing to these bits are ignored when the MRDY flag is 0.

Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to AAh.

Note 3. The MENTRY bit cannot be set to 1 during configuration update, the MENTRYR register cannot be written to during W-HUK zeroing.

Note 4. Written values are not retained by these bits. These bits are read as 0.

The MENTRYR register is used to specify extra MRAM program mode. To enable the extra MRAM sequencer to accept MACI commands, the MENTRY bit must be set to 1 to put the extra MRAM sequencer into program mode.
The MENTRYR register is initialized when the MSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

MENTRY Bit (Extra MRAM Program Mode Entry)

This bit specifies extra MRAM program mode.
[Setting condition]

- When the MENTRYR register is in write-enabled state and the MENTRYR register is in the state of 0000h, AA80h is written to the MENTRYR register.

[Clearing conditions]

- The MENTRYR register is written in 8-bit units when the MSTATR.MRDY flag is set to 1.
- When the MSTATR.MRDY flag is set to 1 and a value other than AAh is specified for the KEY[7:0] bits and written to the MENTRYR register in 16-bit units
- Write AA00h to the MENTRYR register while writing to the MENTRYR register is enabled.
- Write to the MENTRYR register while writing is enabled and its value is other than 0000h.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the MENTRY bit.

65.5.27 Extra MRAM Sequencer Set-Up Initialization Register (MSUINITR)

Address(es): MRAM.MSUINITR 8709 E08Ch, MRAM_NS.MSUINITR 0709 E08Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SUINIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SUINIT	Set-Up Initialization	0: The extra MRAM sequencer setting register values of the MSADDR, MENTRYR, and MCNTSELR registers are retained. 1: Initializes the extra MRAM sequencer settings in the MSADDR, MENTRYR, and MCNTSELR registers	R/W *1, *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W *3

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. This bit can be written when the MRDY flag in the MSTATR register is 1. Writing to this bit is ignored when the MRDY flag is 0.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 2Dh.
Note 3. Written values are not retained by these bits. These bits are read as 0.

The MSUINITR register is used for initialization of the extra MRAM sequencer settings.

SUINIT Bit (Set-Up Initialization)

This bit initializes the following extra MRAM sequencer set-up registers.

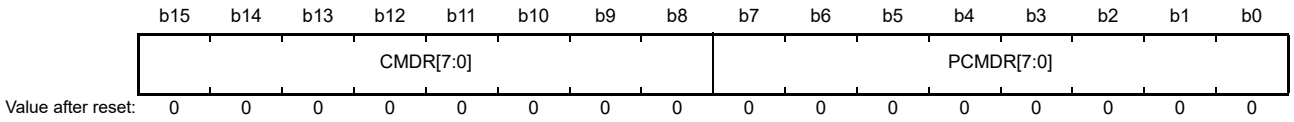
- MSADDR
- MENTRYR
- MCNTSELR

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the SUINIT bit.

65.5.28 MACI Command Register (MCMDR)

Address(es): MRAM.MCMDR 8709 E0A0h, MRAM_NS.MCMDR 0709 E0A0h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Pre-command Flag	The command just before the latest command is stored.	R
b15 to b8	CMDR[7:0]	Command Flag	The latest command is stored.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MCMDR register indicates the command accepted by MACI.

PCMDR[7:0] Flags (Pre-command Flag)

These flags indicate the command received immediately before the latest command received by MACI.

CMDR[7:0] Flags (Command Flag)

These flags indicate the latest command received by MACI.

Table 65.11 States of MCMDR after Receiving Commands

Command	CMDR[7:0] Flags	PCMDR[7:0] Flags
Program	E8h	Previous Command
Status Clear	50h	Previous Command
Forced Stop	B3h	Previous Command
Configuration set	40h	Previous Command
Increment Counter	D0h	35h
Read Counter	D0h	39h

65.5.29 MRAM Startup Area Select Monitor Register (MSUASMON)

Address(es): MRAM.MSUASMON 8709 E0DCh, MRAM_NS.MSUASMON 0709 E0DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	BTSIZE[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1	0/1	0/1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	FSPRS	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0/1	0/1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0.	R
b14	FSPRS	Protection Status Flag of Startup Area Size Selection	0: Protected state 1: Non-protected state	R
b15	FSPR	Protection Status Flag of Startup Area Selection	0: Protected state 1: Non-protected state	R
b28 to b16	—	Reserved	These bits are read as 0.	R
b30, b29	BTSIZE[1:0]	Startup Area Size Select	b30 b29 0 0: Reserved 0 1: Startup area size is set to 32 Kbytes 1 0: Startup area size is set to 16 Kbytes 1 1: Startup area size is set to 8 Kbytes	R
b31	BTFLG	Startup Area Select Flag*1	0: The startup area is the alternate block (block 1) 1: The startup area is the default block (block 0)	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. If the MSUACR.SAS[1:0] bits are changed to 1xb, the startup area follows the setting of the MSUACR.SAS[1:0] bits regardless of the setting of the SAS2.BTFLG bit.

The MSUASMON register is a register that contains a flag that indicates the write protection status for startup area size selection and startup area selection, and that indicates the startup area size selection value and the startup area selection flag value. When a reset or configuration setting command is executed, MACI transfers data from the extra MRAM to the MSUASMON register, and the extra MRAM settings become effective.

FSPRS Flag (Protection Status Flag of Startup Area Size Selection)

This flag indicates whether or not protection is enabled against writing configuration setting commands for the startup area size selection setting.

FSPR Flag (Protection Status Flag of Startup Area Selection)

This flag indicates whether or not protection is enabled against writing configuration setting commands for to startup area selection and writing to the MSUACR register.

BTSIZE[1:0] Bits (Startup Area Size Select)

These bits indicate the size of the startup area.

BTFLG Flag (Startup Area Select Flag)

This flag indicates whether the address of the startup area is exchanged for the boot swap function or not.

65.5.30 MRAM Startup Area Control Register (MSUACR)

Address(es): MRAM.MSUACR 8709 E0E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	SAS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SAS[1:0]	Startup Area Select	b1 b0 0 x: Startup area is selected by the SAS2.BTFLG bit 1 0: Startup area is temporarily switched to the default area regardless of the setting of the SAS2.BTFLG bit 1 1: Startup area is temporarily switched to the alternate area regardless of the setting of the SAS2.BTFLG bit	R/W *1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W *3

x: Don't care.

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. Writing is possible only when the MSUASMON.FSPR flag is 1. Writing is ignored when the MSUASMON.FSPR flag is 0.

Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 66h.

Note 3. Written values are not retained by these bits. These bits are read as 0.

The MSUACR register is used to switch the startup area between the default area (block 0) and the alternative area (block 1) using the boot swap function.

SAS[1:0] Bits (Startup Area Select)

These bits are used to switch the startup area between the default area (block 0) and the alternate area (block 1).

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the SAS [1:0] bits.

65.5.31 MACI Reset Transfer Status Register (MRSTTRST)

Address(es): MRAM.MRSTTRST 8709 E0F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RSTTRER
Value after reset:	0	0	0	0	0	0	0	x

x: Don’t care

Bit	Symbol	Bit Name	Description	R/W
b0	RSTTRER	MACI Reset Transfer Status	0: ECC error has not been detected. 1: ECC error has been detected.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.

RSTTRER Bit (MACI Reset Transfer Status)

This bit indicates the MACI reset transfer status.

65.5.32 MRAM Program Speed Control Register (MRPSC)

Address(es): MRAM.MRPSC 8709 E800h, MRAM_NS.MRPSC 0709 E800h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MHSPEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MHSPEN	MRAM High Speed Program Mode Enable	0: Normal program mode 1: High speed program mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

MHSPEN Bit (MRAM High Speed Program Mode Enable)

This bit is used to enable or disable high speed program mode of code MRAM or extra MRAM. After setting this bit to 1 and programming code MRAM or extra MRAM, the mode is changed to high-speed program mode. To change to normal program mode, set this bit to 0, read the MRPSC register once, and program or read the MRAM.

65.5.33 Extra MRAM ECC Encoder Control Register (MREEECC)

Address(es): MRAM.MREEECC 8709 E804h, MRAM_NS.MREEECC 0709 E804h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	ECCBYPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECCBYPE	Extra MRAM ECC encoder outputs bypass enable	0: ECC bypass is disabled. 1: ECC bypass is enabled.	R/W*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	The ECCBYPE bit protection key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to EFh.
Note 2. Written values are not retained by these bits. These bits are read as 0.

The MREEECC register is used to enable the bypass function of the ECC encoder output of the extra MRAM.
It is used to test the ECC decoder circuit of the extra MRAM. For information on the ECC decoder circuits test flow, see section 65.9.6, Fault Correction and Detection.

ECCBYPE Bit (Extra MRAM ECC encoder outputs bypass enable)

This bit enables the function to bypass the ECC encoder output of the extra MRAM.
When the ECCBYPE bit is 0, the ECC bits of the extra MRAM are the ECC encoder output.
When the ECCBYPE bit is 1, the ECC bits of the extra MRAM are bits [16:0] of the write data.

KEY[7:0] Bits (Key Code)

The KEY[7:0] bits control permission and prohibition of writing to the ECCBYPE bit.
To write to the ECCBYPE bit, write EFh to the KEY[7:0] at the same time. If a value other than EFh is written, the ECCBYPE bit will not be updated. When read, it will return 00h.

65.5.34 Code MRAM Program Control Register for Non-Secure Application (MRCPCNS)

Address(es): MRAM.MRCPCNS 8709 F000h, MRAM_NS.MRCPCNS 0709 F000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	MRCPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRCPEN	Code MRAM Program Enable	0: Code MRAM non-secure areas are not programmable 1: Code MRAM non-secure areas are programmable	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W *2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 86h.
Note 2. Written values are not retained by these bits. These bits are read as 0.

MRCPEN Bit (Code MRAM Program Enable)

This bit is used to enable programming to the non-secure area.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the MRCPEN bit.

65.5.35 Code MRAM Program Control Register for Secure Application (MRCPCS)

Address(es): MRAM.MRCPCS 8709 F004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	MRCPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRCPEN	Code MRAM Program Enable	0: Code MRAM secure areas are not programmable 1: Code MRAM secure areas are programmable	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W *2

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 68h.
Note 2. Written values are not retained by these bits. These bits are read as 0.

MRCPEN Bit (Code MRAM Program Enable)

This bit is used to enable programming to the secure area.

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the MRCPEN bit.

65.5.36 Code MRAM Block Protection Register for Non-Secure Application (MRCBPROTNS)

Address(es): MRAM.MRCBPROTNS 8709 F008h, MRAM_NS.MRCBPROTNS 0709 F008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	BPCN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BPCN	Code MRAM Block Protection Cancel	0: Code MRAM block protection is enabled 1: Code MRAM block protection is disabled	R/W *1, *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	BPCN bit protection key code	R/W *3

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. This bit can be written when the MRCPCNS.MRCPEN bit is 1. Writing when the MRCPCNS.MRCPEN bit is 0 is ignored.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 78h.
Note 3. Written values are not retained by these bits. These bits are read as 0.

The MRCBPROTNS register is used to disable the block protect function for non-secure. When the block protect setting is locked by permanent block setting, it cannot be disabled by this register.

BPCN Bit (Code MRAM Block Protection Cancel)

This bit disables the block protection function for the non-secure code MRAM area.

[Setting condition]

- When the MRCPCNS.MRCPEN bit is 1, 7801h is written to the MRCBPROTNS register.

[Clearing conditions]

- When the MRCPCNS.MRCPEN bit is 1, 7800h is written to the MRCBPROTNS register.
- When 0 is written to the MRCPCNS.MRCPEN bit.

KEY[7:0] Bits (Key Code)

This bit enables or disables writing to the BPCN bit.

When writing to the BPCN bit, write 78h to the KEY[7:0] bits at the same time. If a value other than 78h is written, the BPCN bit will not be updated. When read, it will return 00h.

65.5.37 Code MRAM Block Protection Register for Secure Application (MRCBPROTS)

Address(es): MRAM.MRCBPROTS 8709 F00Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	BPCN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BPCN	Code MRAM Block Protection Cancel	0: Code MRAM block protection is enabled 1: Code MRAM block protection is disabled	R/W *1, *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	BPCN1 bit protection key code	R/W *3

Note: This register is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. This bit can be written when the MRCPCS.MRCPEN bit is 1. Writing when the MRCPCS.MRCPEN bit is 0 is ignored.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to B1h.
Note 3. Written values are not retained by these bits. These bits are read as 0.

The MRCBPROTS register is used to disable the block protect function for secure. When the block protect setting is locked by permanent block setting, it cannot be disabled by this register.

BPCN Bit (Code MRAM Block Protection Cancel)

This bit disables the block protection function for the secure code MRAM area.
[Setting condition]

- When the MRCPCS.MRCPEN bit is 1, B101h is written to the MRCBPROTS register.

[Clearing condition]

- When the MRCPCS.MRCPEN bit is 1, B100h is written to the MRCBPROTS register.
- When 0 is written to the MRCPCS.MRCPEN bit.

KEY[7:0] Bits (Key Code)

This bit enables or disables writing to the BPCN bit.
When writing to the BPCN bit, write B1h to the KEY[7:0] bits at the same time. If a value other than B1h is written, the BPCN bit will not be updated. When read, it will return 00h.

65.5.38 Code MRAM Program Status Register (MRCPS)

Address(es): MRAM.MRCPS 8709 F010h, MRAM_NS.MRCPS 0709 F010h

	b7	b6	b5	b4	b3	b2	b1	b0
	PRGBSYC	ABUFFULL	ABUFEMP	PRGBSYCB	ABUFFULLB	ABUFEMPB	ECCERRC	PRGERRC
Value after reset:	0	0	1	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRGERRC	Programming Error Flag	0: Programming has been completed successfully 1: An error has occurred during programming	R/W *1
b1	ECCERRC	ECC Error Flag	0: Programming has been completed successfully 1: An error has occurred during programming	R/W *1
b2	ABUFEMPB	MRC1 Address Buffer Empty Flag	0: The address buffer is not empty. Write data can be flushed. 1: The address buffer is empty. Write data cannot be flushed.	R
b3	ABUFFULLB	MRC1 Address Buffer Full Flag	0: Address buffer is not full. Write transactions are allowed. 1: Address buffer is full. Write transactions are not allowed.	R
b4	PRGBSYCB	MRC1 Programming Status Flag	0: Programming is not in progress 1: Programming is in progress	R
b5	ABUFEMP	MRC0 Address Buffer Empty Flag	0: The address buffer is not empty. Write data can be flushed. 1: The address buffer is empty. Write data cannot be flushed.	R
b6	ABUFFULL	MRC0 Address Buffer Full Flag	0: Address buffer is not full. Write transactions are allowed. 1: Address buffer is full. Write transactions are not allowed.	R
b7	PRGBSYC	MRC0 Programming Status Flag	0: Programming is not in progress 1: Programming is in progress	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. 0 can be written to clear the flag only after 1 has been read.

The MRCPS register indicates the status of the code MRAM programming sequencer.

PRGERRC Flag (Programming Error Flag)

This flag indicates the result of programming to the code MRAM. Writing 1 to this flag is ignored.

[Setting condition]

- An error has occurred during programming and the ECCERRC flag is 0.

[Clearing condition]

- 0 is written after reading 1 from this flag.

ECCERRC Flag (ECC Error Flag)

This flag indicates that an uncorrectable error has occurred when the programming data from the code bus is less than 32 bytes. Programming data to code MRAM is generated from data in the target address and data from code bus. If a TED error occurs when reading data from the target address, programming data to code MRAM cannot be correctly generated if the data from code bus is less than 32 bytes. To perform programming to an address with this flag set to 1, all 32 bytes data must be input from code bus.
This flag can only be written to 0 and writing 1 is ignored.
[Setting condition]

- Uncorrectable error occurred when programming data from code bus is less than 32 bytes and PRGERRC flag is 0.

[Clearing condition]

- 0 is written after reading 1 from this bit.

ABUFEMPB Flag (MRC1 Address Buffer Empty Flag)

This flag indicates the status of the programming address buffer for the code MRAM on the MRC1 side. When the address buffer is empty, this flag becomes 1. When this flag is 1, writing to the MRCFLR.MRCBFL bit is not possible, so programming data cannot be flashed.

ABUFFULLB Flag (MRC1 Address Buffer Full Flag)

This flag indicates the status of the address buffer for programming the code MRAM on the MRC1 side. When the address buffer is full, this flag becomes 1. When this flag is 1, new write transactions are not accepted until programming to the MRC1 area starts or ends.

PRGBSYCB Flag (MRC1 Programming Status Flag)

This flag indicates the programming status of the code MRAM on the MRC1 side. If CPSEQ executes the code MRAM programming processing, this flag is set to 1. When this flag is 1, a new read transaction to the MRC1 area cannot be accepted until the MRC1 area programming finishes.

ABUFEMP Flag (MRC0 Address Buffer Empty Flag)

This flag indicates the status of the programming address buffer for the code MRAM on the MRC0 side. When the address buffer is empty, this flag becomes 1. When this flag is 1, writing to the MRCFLR.MRCBFL bit is not possible, so programming data cannot be flashed.

ABUFFULL Flag (MRC0 Address Buffer Full Flag)

This flag indicates the status of the address buffer for programming the code MRAM on the MRC0 side. When the address buffer is full, this flag becomes 1. When this flag is 1, new write transactions are not accepted until programming to the MRC0 area starts or ends.

PRGBSYC Flag (MRC0 Programming Status Flag)

This flag indicates the programming status of the code MRAM on the MRC0 side. If CPSEQ executes the code MRAM programming processing, this flag is set to 1. When this flag is 1, a new read transaction to the MRC0 area cannot be accepted until the MRC0 area programming finishes.

65.5.39 Code MRAM Program Access Error Interrupt Enable Register (MRCPAEINT)

Address(es): MRAM.MRCPAEINT 8709 F014h, MRAM_NS.MRCPAEINT 0709 F014h

	b7	b6	b5	b4	b3	b2	b1	b0
	MRCAEIE	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MRCAEIE	Program Access Error Interrupt Enable	0: Generation of an MRCPRI interrupt request is disabled 1: Generation of an MRCPRI interrupt request is enabled	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRCPAEINT register enables or disables generation of a code MRAM access error (MRCPRI) interrupt request.

MRCAEIE Bit (Program Access Error Interrupt Enable)

This bit enables a code MRAM access error (MRCPRI) interrupt request to be generated when the MRCPS.PRGERRC flag or MRCPS.ECCERRC flag is 1.

When the MRCAEIE bit is 1, an MRCPRI interrupt request is generated. When the MRCAEIE bit is 0, an MRCPRI interrupt request is not generated.

65.5.40 Code MRAM Program Error Address Register (MRCPEA)

Address(es): MRAM.MRCPEA 8709 F018h, MRAM_NS.MRCPEA 0709 F018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MCPEA[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MCPEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b5	MCPEA[26:0]	Code MRAM Program Error Address	Indicates address when a code MRAM program error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRCPEA register stores the error address when a code MRAM program error occurs.

MCPEA[26:0] Bits (Code MRAM Program Error Address)

[Update condition]

- When a program error or ECC error occurs while the MRCPS.ECCERRC flag is 0 and the MRCPS.PROGERRC flag is 0, the error address is stored in the MRCPEA register.

65.5.41 Code MRAM Flush Register (MRCFLR)

Address(es): MRAM.MRCFLR 8709 F030h, MRAM_NS.MRCFLR 0709 F030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	MRCBFL	MRCFL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRCFL	Flush Write Data Buffer for MRC0	0: Not execute to flush program data buffer 1: Execute to flush program data buffer	R/W *1, *3
b1	MRCBFL	Flush Write Data Buffer for MRC1	0: Not execute to flush program data buffer 1: Execute to flush program data buffer	R/W *2, *3
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	The MRCFL and MRCBFL bit protection key codes	R/W *4

Note 1. When the ABUFEMP flag in the MRCPS register is 1, it is not possible to write 1 to this bit.
Note 2. When the ABUFEMPB flag in the MRCPS register is 1, it is not possible to write 1 to this bit.
Note 3. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to C3h.
Note 4. Written values are not retained by these bits. These bits are read as 0.

The MRCFLR register is used for writing data less than 32 bytes.

MRCFL Bit (Flush Write Data Buffer for MRC0)

This bit is used for starting the programming operation to the code MRAM on the MRC0 side.
[Setting condition]

- When C301h or C303h is written to the MRCFLR register while the MRCPS.ABUFEMP flag is 0.

[Clearing condition]

- When the MRCPS.ABUFEMP flag is set to 1.

MRCBFL Bit (Flush Write Data Buffer for MRC1)

This bit is used for starting the programming operation to the code MRAM on the MRC1 side.
[Setting condition]

- When C302h or C303h is written to the MRCFLR register while the MRCPS.ABUFEMPB flag is 0.

[Clearing condition]

- When the MRCPS.ABUFEMPB flag is set to 1.

KEY[7:0] Bits (Key Code)

When writing to the MRCFL or MRCBFL bit, write C3h to the KEY[7:0] bits at the same time. If a value other than C3h is written, the MRCFL or MRCBFL bit will not be updated. When read, it will return 00h.

65.5.42 Code MRAM Program Ready Interrupt Control Register (MRCPRICR)

Address(es): MRAM.MRCPRICR 8709 F040h, MRAM_NS.MRCPRICR 0709 F040h

	b7	b6	b5	b4	b3	b2	b1	b0
	MRCPRRIE	—	—	—	—	—	—	MRCINTSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRCINTSEL	Program Ready Interrupt Request Source Select	0: Selects the program ready interrupt for the code MRAM on the MRC0 side 1: Selects the program ready interrupt for the code MRAM on the MRC1 side	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MRCPRRIE	Program Ready Interrupt Enable	0: Generation of program ready interrupt requests is disabled. 1: Generation of program ready interrupt requests is enabled.	R/W *1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. It is prohibited to set the MRCPRICR.MRCPRRIE bit and the MRDPRICR.MRDPRRIE bit to 1 at the same time.

The MRCPRICR register is used to enable or disable the generation of MRAM program ready (MRPRRI) interrupt requests.

MRCINTSEL Bit (Program Ready Interrupt Request Source Select)

This bit is used to select the source of the MRAM program ready (MRPRRI) interrupt.

MRCPRRIE Bit (Program Ready Interrupt Enable)

This bit enables the generation of an MRAM program ready (MRPRRI) interrupt request.
When programming on the MRC0 side (MRCINTSEL bit = 0), a program ready interrupt request can be generated if the MRCPS.ABUFFULLC bit is 0. Writing 1 to this bit generates the first interrupt request. After that, when a write transaction to the MRC0 side is accepted, an interrupt request is generated.
When programming on the MRC1 side (MRCINTSEL bit = 1), a program ready interrupt request can be generated if the MRCPS.ABUFFULLCB bit is 0. Writing 1 to this bit generates the first interrupt request. After that, when a write transaction to the MRC1 side is accepted, an interrupt request is generated.

65.5.43 Data MRAM Program Status Register (MRDPS)

Address(es): MRAM.MRDPS 8709 F410h, MRAM_NS.MRDPS 0709 F410h

	b7	b6	b5	b4	b3	b2	b1	b0
	PRGBSYD	ABUFFULLD	ABUFEMPD	—	—	—	ECCERRD	PRGERRD
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRGERRD	Programming Error Flag	0: Programming has been completed successfully 1: An error has occurred during programming	R/W *1
b1	ECCERRD	ECC Error Flag	0: No error has occurred during programming 1: An error has occurred during programming	R/W *1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ABUFEMPD	Data MRAM Address Buffer Empty Flag	0: The address buffer is not empty. Write data can be flushed. 1: The address buffer is empty. Write data cannot be flushed.	R
b6	ABUFFULLD	Data MRAM Address Buffer Full Flag	0: Address buffer is not full. Write transactions are allowed. 1: Address buffer is full. Write transactions are not allowed.	R
b7	PRGBSYD	Data MRAM Programming Status Flag	0: Programming is not in progress 1: Programming is in progress	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. 0 can be written to clear the flag only after 1 has been read.

The MRDPS register indicates the status of the data MRAM programming sequencer.

PRGERRD Flag (Programming Error Flag)

This flag indicates the result of programming to the data MRAM. Writing 1 to this flag is ignored.

[Setting condition]

- When an error occurs during programming and the ECCERRD flag is 0

[Clearing condition]

- 0 is written after reading 1 from this flag.

ECCERRD Flag (ECC Error Flag)

This flag indicates that an uncorrectable error has occurred when the programming data from the peripheral bus is less than 16 bytes. The programming data for the data MRAM is generated from the data at the target address and the data from the peripheral bus. If a TED error occurs when reading data from the target address and the data from the peripheral bus is less than 16 bytes, the programming data for the data MRAM cannot be generated correctly. When programming an address where this flag is set to 1, all 16 bytes of data must be input from the peripheral bus.

This flag can only be written as 0; writing 1 is ignored.

[Setting condition]

- When an uncorrectable error occurs, and the programming from the peripheral bus is less than 16 bytes, and the PRGERRD flag is 0

[Clearing condition]

- 0 is written after reading 1 from this flag.

ABUFEMPD Flag (Data MRAM Address Buffer Empty Flag)

This flag indicates the status of the programming address buffer for the data MRAM. When the address buffer is empty,

this flag becomes 1. When this flag is 1, writing to the MRDFLR.MRDFL bit is not possible, so programming data cannot be flashed.

ABUFFULLD Flag (Data MRAM Address Buffer Full Flag)

This flag indicates the status of the address buffer for programming the data MRAM. When the address buffer is full, this flag becomes 1. When this flag is 1, new write transactions are not accepted until programming to the data MRAM starts or ends.

PRGBSYD Flag (Data MRAM Programming Status Flag)

This flag indicates the programming status of the data MRAM. If CPSEQ executes the data MRAM programming processing, this flag is set to 1. When this flag is 1, a new read transaction to the data MRAM cannot be accepted until the data MRAM programming finishes.

65.5.44 Data MRAM Program Access Error Interrupt Enable Register (MRDPAEINT)

Address(es): MRAM.MRDPAEINT 8709 F414h, MRAM_NS.MRDPAEINT 0709 F414h

	b7	b6	b5	b4	b3	b2	b1	b0
	MRDAEIE	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MRDAEIE	Data MRAM Program Access Error Interrupt Enable	0: Generation of MRDPRI interrupt requests is disabled. 1: Generation of MRDPRI interrupt requests is enabled.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRDPAEINT register is used to enable or disable the generation of data MRAM access error (MRDPRI) interrupt requests.

MRDAEIE Bit (Data MRAM Program Access Error Interrupt Enable)

This bit is used to enable the generation of data MRAM access error (MRDPRI) interrupt requests.
When the MRDPS.PRGERRD flag or MRDPS.ECCERRD flag is 1, an MRDPRI interrupt request is generated if this bit is 1. If this bit is 0, an MRDPRI interrupt request is not generated.

65.5.45 Data MRAM Program Error Address Register (MRDPEA)

Address(es): MRAM.MRDPEA 8709 F418h, MRAM_NS.MRDPEA 0709 F418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MDPEA[27:12]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MDPEA[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b4	MDPEA[27:0]	Data MRAM Program Error Address	Stores the error address when a data MRAM program error occurs.	R

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MRDPEA register is used to store the error address when a data MRAM program error occurs.

MDPEA[27:0] Bits (Data MRAM Program Error Address)

[Update condition]

- When a program error or ECC error occurs while the MRDPS.PRGERRD flag is 0 and the MRDPS.ECCERRD flag is 0, the error address is stored in the MRDPEA register.

65.5.46 Data MRAM Flush Register (MRDFLR)

Address(es): MRAM.MRDFLR 8709 F430h, MRAM_NS.MRDFLR 0709 F430h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	MRDFL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRDFL	Flush Write Data Buffer for Data MRAM	0: Not execute to flush program data buffer 1: Execute to flush program data buffer	R/W *1, *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	The MRDFL bit protection key code	R/W *3

Note 1. When the ABUFEMPD flag in the MRDPS register is 1, it is not possible to write 1 to this bit.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to C4h.
Note 3. Written values are not retained by these bits. These bits are read as 0.

The MRDFLR register is used for programming data less than 16 bytes.

MRDFL Bit (Flush Write Data Buffer for Data MRAM)

This bit is used for starting data MRAM programming operation.

[Setting condition]

- When C401h is written to the MRDFLR register while the MRDPS.ABUFEMPD flag is 0.

[Clearing condition]

- When the ABUFEMPD flag in the MRDPS register is set to 1.

KEY[7:0] Bits (Key Code)

The KEY[7:0] bits control permission and prohibition of writing to the MRDFL bit.

To write to the MRDFL bit, write C4h to the KEY[7:0] bits at the same time. If a value other than C4h is written, the MRDFL bit will not be updated. When read, it will read 00h.

65.5.47 Data MRAM Program Ready Interrupt Control Register (MRDPRICR)

Address(es): MRAM.MRDPRICR 8709 F440h, MRAM_NS.MRDPRICR 0709 F440h

	b7	b6	b5	b4	b3	b2	b1	b0
	MRDPRRIE	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MRDPRRIE	Data MRAM Program Ready Interrupt Enable	0: Generation of program ready interrupt request is disabled. 1: Generation of program ready interrupt request is enabled.	R/W *1

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

Note 1. It is prohibited to set the MRCPRICR.MRCPRRIE bit and the MRDPRICR.MRDPRRIE bit to 1 at the same time.

The MRDPRICR register is used to enable or disable the generation of MRAM program ready (MRPRRI) interrupt requests.

MRDPRRIE Bit (Data MRAM Program Ready Interrupt Enable)

This bit is used to enable the generation of MRAM program ready (MRPRRI) interrupt requests.
When programming to the data MRAM, a program ready interrupt request can be generated if the MRDPS.ABUFFULLD flag is 0. Writing 1 to this bit generates the first interrupt request. After that, when a write transaction to the data MRAM is accepted, an interrupt request is generated.

65.5.48 Code MRAM ECC Encoder Control Register (MRCEECC)

Address(es): MRAM.MRCEECC 8709 F804h, MRAM_NS.MRCEECC 0709 F804h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	ECCBYPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECCBYPC	Code MRAM ECC encoder outputs bypass enable	0: ECC bypass is disabled. 1: ECC bypass is enabled.	R/W*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	The ECCBYPC bit protection key code	R/W*2

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.
Note 1. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to CDh.
Note 2. Written values are not retained by these bits. These bits are read as 0.

The MRCEECC register is used to enable the bypass function of the ECC encoder output of the code MRAM.
It is used to test the ECC decoder circuit of the code MRAM. For information on the ECC decoder circuits test flow, see section 65.9.6, Fault Correction and Detection.

ECCBYPC Bit (Code MRAM ECC encoder outputs bypass enable)

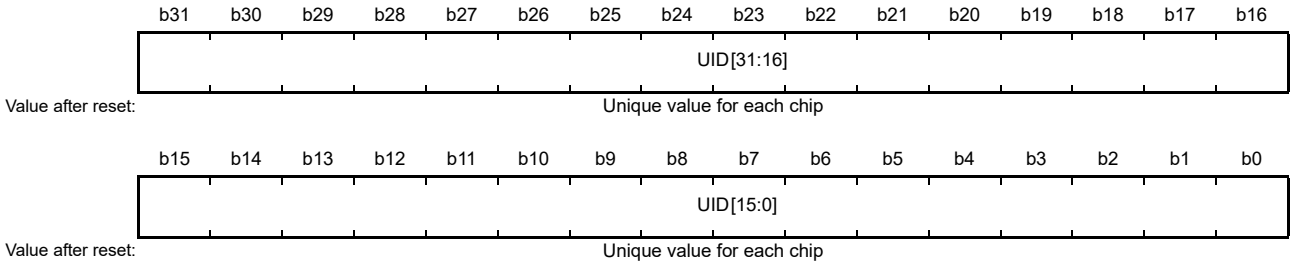
This bit enables the function to bypass the ECC encoder output of the code MRAM.
When the ECCBYPC bit is 0, the ECC bits of the code MRAM are the ECC encoder output.
When the ECCBYPC bit is 1, the ECC bits of the code MRAM are bits [144:128,16:0] of the write data.

KEY[7:0] Bits (Key Code)

The KEY[7:0] bits control permission and prohibition of writing to the ECCBYPC bit.
To write to the ECCBYPC bit, write CDh to the KEY[7:0] at the same time. If a value other than CDh is written, the ECCBYPC bit will not be updated. When read, it will return 00h.

65.5.49 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): NVMCONST.UIDR0 86F0 1000h, NVMCONST.UIDR1 86F0 1004h, NVMCONST.UIDR2 86F0 1008h, NVMCONST.UIDR3 86F0 100Ch, NVMCONST_NS.UIDR0 06F0 1000h, NVMCONST_NS.UIDR1 06F0 1004h, NVMCONST_NS.UIDR2 06F0 1008h, NVMCONST_NS.UIDR3 06F0 100Ch



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	UID	Unique ID	Unique ID for individual identification	R

The UIDRn register is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

65.5.50 Part Numbering Register n (PNRn) (n = 0 to 3)

Address(es): NVMCONST.PNR0 86C1 2C38h, NVMCONST.PNR1 86C1 2C3Ch, NVMCONST.PNR2 86C1 2C40h, NVMCONST.PNR3 86C1 2C44h, NVMCONST_NS.PNR0 06C1 2C38h, NVMCONST_NS.PNR1 06C1 2C3Ch, NVMCONST_NS.PNR2 06C1 2C40h, NVMCONST_NS.PNR3 06C1 2C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PNR[31:16]															
Value after reset:	Unique value for each chip															

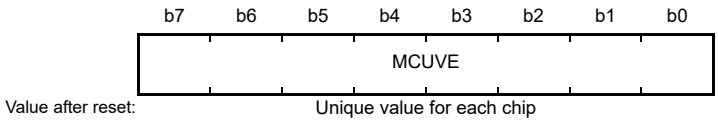
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNR[15:0]															
Value after reset:	Unique value for each chip															

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	PNR	Part Number	Part Number Information	R

The PNRn register is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in Table 1.3, List of Products. The first character of the part number (ASCII code “R”, 52h) is stored in b0 to b7 of the PNR0 register. When reading with the signature request command of the serial programming interface, data is read in order from the lowest address data.

65.5.51 MCU Version Register (MCUVER)

Address(es): NVMCONST.MCUVER 86C1 2C48h, NVMCONST_NS.MCUVER 06C1 2C48h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	MCUVE	MCU Version	MCU Version Information	R

The MCUVER is a read-only register that stores a MCU version. The MCUVER register should be read in 8-bit units.

65.5.52 MRAM-OTP Write Mode Control Register (MWMCR)

Address(es): SYSC.MWMCR 8701 9AB4h, SYSC_NS.MWMCR 0701 9AB4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MWM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MWM[1:0]	MRAM-OTP Write Mode	b1 b0 0 0: Normal speed write mode 0 1: High speed write mode 0 1 0: High speed write mode 1 1 1: Prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When this register is set to the secure attribute, it is accessible by secure transactions. Any access by non-secure transaction is ignored, and a Protect Zone access error is generated.
When this register is set to the non-secure attribute, it is accessible by non-secure transactions. Any access by secure transaction is ignored, and a Protect Zone access error is generated.

The MWMCR register controls the MRAM-OTP write mode when writing data to MRAM-OTP.

MWM[1:0] Bits (MRAM-OTP Write Mode)

1. MWM[1:0] = 00: MRAM-OTP is set to MRAM-OTP normal speed write mode.
   You can set MWM[1:0] to 00 in the case that 1.62V ≤ VCC ≤ 3.63V.
   Refer to the write period of MRAM-OTP normal speed write mode to section 67, Electrical Characteristics.
2. MWM[1:0] = 01: MRAM-OTP is set to MRAM-OTP high speed write mode 0.
   You can set MWM[1:0] to 01 in the case that 2.5V ≤ VCC ≤ 3.63V.
   Refer to the write period of MRAM-OTP high speed write mode 0 to section 67, Electrical Characteristics.
3. MWM[1:0] = 10: MRAM-OTP is set to MRAM-OTP high speed write mode 1.
   You can set MWM[1:0] to 10 in the case that 3.0V ≤ VCC ≤ 3.63V.
   Refer to the write period of MRAM-OTP high speed write mode 1 to section 67, Electrical Characteristics.

65.6 Operating Mode

65.6.1 MRAM Read/Program Mode

The code MRAM and extra MRAM have modes for reading and programming as shown in Figure 65.16.
A transition from code MRAM read mode to code MRAM program mode is started by a request for programming generated under one of the following conditions.

- The program data buffer has become full.
- The last address of the code MRAM where data have been written is beyond the 16-byte boundary of the addresses at which the data stored in the program data buffer are to be programmed.
- The MRCFLR.MRCFL bit is set to 1.

Programming of the code MRAM automatically begins after a transition to code MRAM program mode. When programming of the code MRAM is completed, operation automatically switches from code MRAM program mode to code MRAM read mode.

Transitions between extra MRAM read mode and extra MRAM program mode are initiated by changing the value of the MENTRYR register, by issuing a request for programming of the data MRAM, or by W-HUK zeroization (see section 65.8.6, HUK Protection).
The MRAM is in extra MRAM read mode when the MENTRYR register is set to 0000h, no request for programming of the data MRAM is issued, and W-HUK zeroization is not executed.
MRAM is in extra MRAM program mode in other cases.
Programming of the extra MRAM can be started by using an MACI command while the MENTRYR register is set to 0080h, by issuing a request for programming of the data MRAM while the MENTRYR register is set to 0000h, or by starting W-HUK zeroization.

A transition to extra MRAM program mode for programming of the data MRAM is started by a request for programming generated under one of the following conditions.

- The program data buffer has become full.
- The last address of the data MRAM where data have been written is beyond the 16-byte boundary of the addresses at which the data stored in the program data buffer are to be programmed.
- The MRDFLR.MRDFL bit is set to 1.

Programming of the data MRAM automatically begins after a transition to extra MRAM program mode. When programming of the data MRAM is completed, operation automatically switches from extra MRAM program mode to extra MRAM read mode.

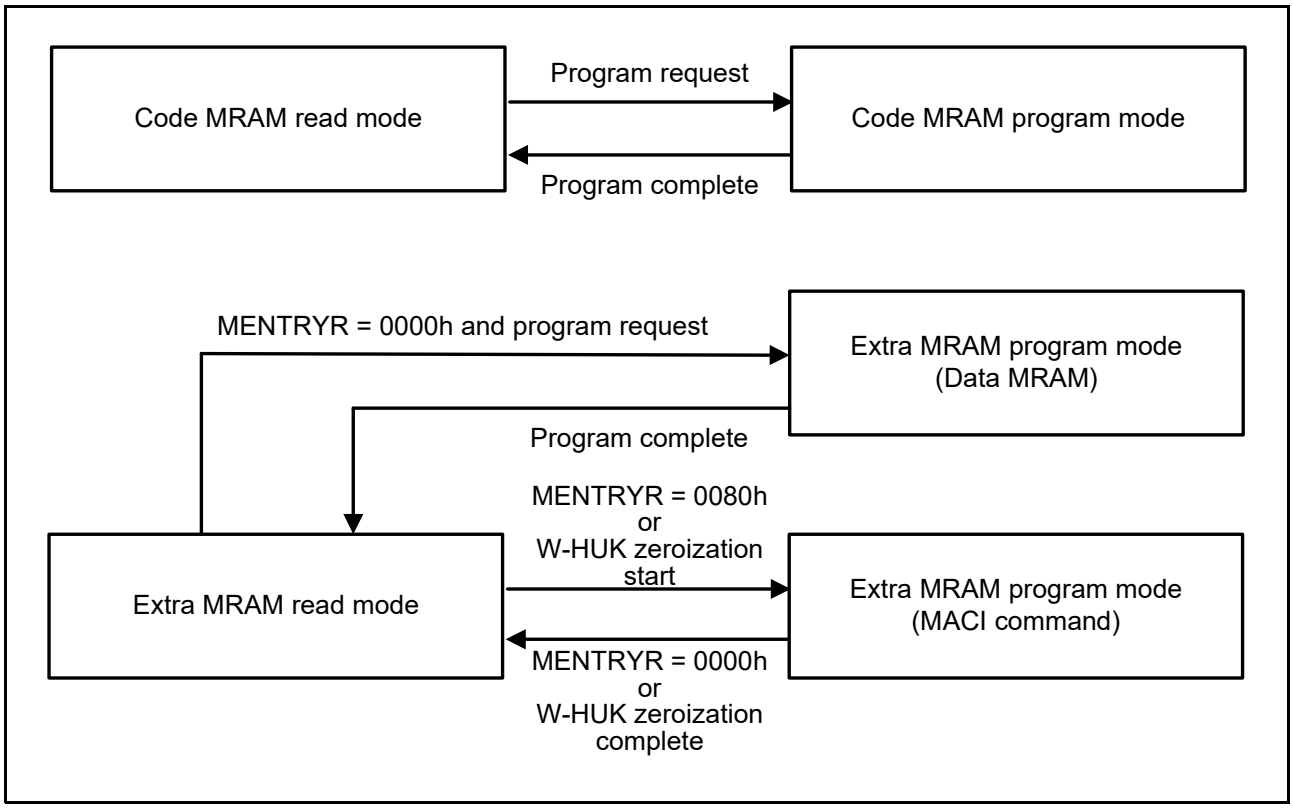


Figure 65.16 MRAM Read/Program Mode Transition

65.6.2 MRAM High-Speed Program Mode

MRAM has normal program mode and high-speed program mode, as shown in Figure 65.17, and the same mode settings apply to both code MRAM program mode and extra MRAM program mode.
In normal program mode, when the MHSPEN bit of the MRPSC register is set to 1 in the “MHSPM disabled” state, the system transitions to the MHSPM startup preparation state.
Executing an MRAM program once in the MHSPM startup preparation state transitions to high-speed program mode.
Setting the MHSPEN bit to 0 transitions from high-speed program mode to normal program mode.
The MRAM program mode enters the MHSPM startup preparation state when the MCU is in standby mode.
The setting of the MHSPEN bit is retained in standby mode, so after wake-up, programming MRAM or extra MRAM once transitions the MRAM program mode to high-speed program mode.

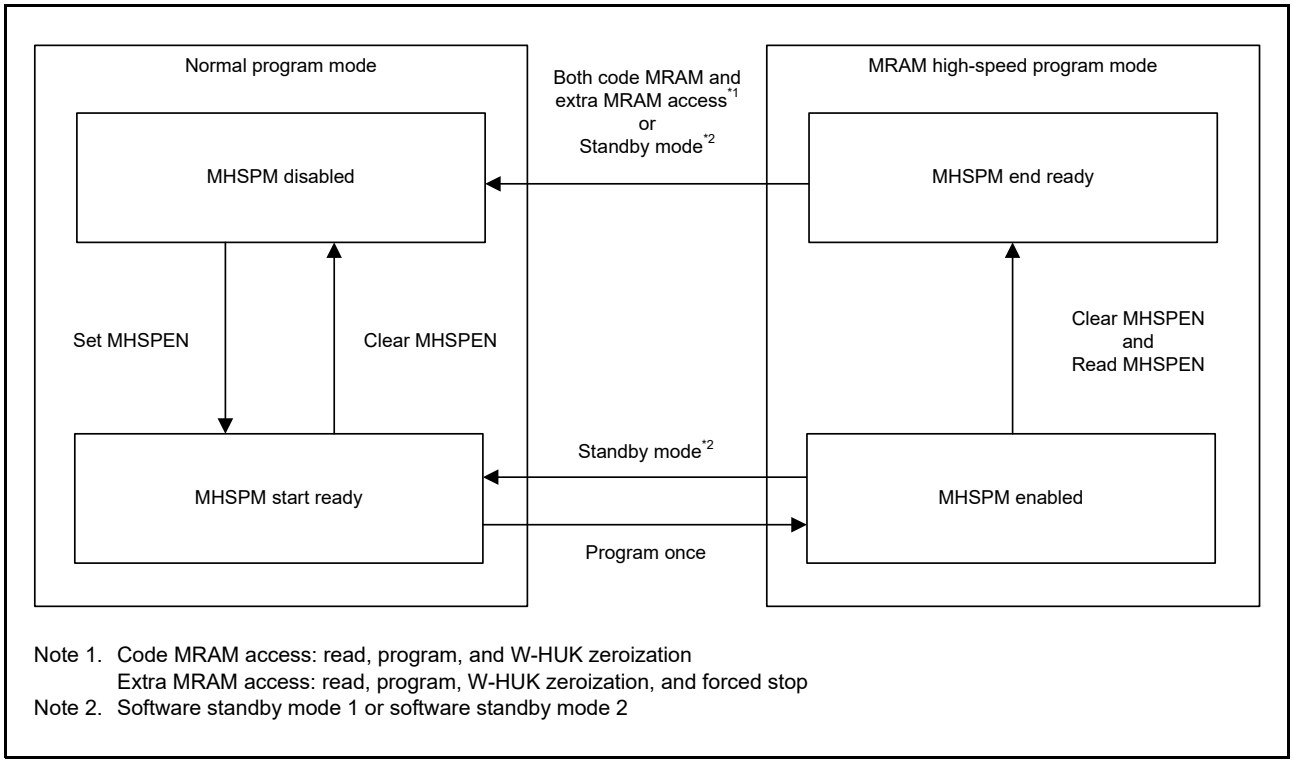


Figure 65.17 High-Speed Program Mode Transition

65.7 MACI Commands

65.7.1 Programmable Area with MACI Commands

Table 65.12 shows the MRAM and OTP areas in extra MRAM that can be programmed with the MACI command. These areas store setting data for security, safety, and system configuration. Each bit (both data and ECC) in OTP area is allowed to be programmed only one time from 1 to 0.

There are two types of OTP areas with ECC and without ECC as shown in Table 65.12. A unit of 16 bytes data can be programmed only once to OTP area with ECC. Reprogramming of different data cannot be retried because some bits of ECC may need to change from 0 to 1. Reliability of OTP area without ECC is secured by redundant data configuration. A unit of 16 bytes data can be programmed multiple times to OTP area without ECC in case that there is no bit changing from 0 to 1.

Table 65.12 Programmable Area with MACI Commands

Address (Offset address: BASE_ME)	Area Name	Memory Type	ECC
00C2 E820h to 00C2 E9FFh	Secure OFS configuration	MRAM	Use
00C4 EC20h to 00C4 EC7Fh	Non-secure OFS configuration	MRAM	Use
00C9 F050h to 00C9 F7FFh	Configuration	MRAM	Use
00D8 0000h to 00D9 FFFFh	User extra MRAM area	MRAM or OTP	Use
00E0 0400h to 00E0 040Fh	FSBL setting	OTP	Use
00E0 0410h to 00E0 049Fh	Start address of measurement report Start address of code certificate	OTP	Use
00E0 04A0h to 00E0 04FFh	General purpose OTP	OTP	Use
00E1 0610h to 00E1 061Fh	Secure OFS OTP memory type setting	OTP	Use
00E1 0620h to 00E1 06EFh	Secure OFS OTP setting	MRAM or OTP	Use
00E1 0710h to 00E1 07FFh	Secure OFS OTP setting	MRAM or OTP	Use
00E1 0A00h to 00E1 0A7Fh	Permanent block protect setting for secure*1	OTP	Don't use
00E1 0A80h to 00E1 0A9Fh	Data MRAM permanent block protection setting for secure	OTP	Don't use
00E2 0810h to 00E2 081Fh	Non-secure OFS OTP memory type setting	OTP	Use
00E2 0820h to 00E2 087Fh	Non-secure OFS OTP setting	MRAM or OTP	Use
00E2 0C00h to 00E2 0C7Fh	Permanent block protect setting for non-secure*2	OTP	Don't use
00E2 0C80h to 00E2 0C9Fh	Data MRAM permanent block protection setting for non-secure	OTP	Don't use
00E3 0E00h to 00E3 0E0Fh	Secure OFS OTP setting protect setting	OTP	Don't use
00E3 0E10h to 00E3 0E1Fh	REVOKE	OTP	Don't use
00E3 0E20h to 00E3 0E2Fh	Zeroization HUK in OTP enable	OTP	Don't use
00E3 0E30h to 00E3 0E3Fh	Anti-rollback counter setting	OTP	Don't use
00E3 0E50h to 00E3 0E5Fh	User Extra MRAM	OTP	Don't use
00E3 0E60h to 00E3 0E6Fh	Secure OFS setting protect setting	OTP	Don't use
00E3 0E80h to 00E3 0E8Fh	Non-secure OFS OTP setting protect setting	OTP	Don't use
00E3 0EE0h to 00E3 0EEFh	Non-secure OFS setting protect setting	OTP	Don't use
00E3 0EF0h to 00E3 0EFFh	Reserved	OTP	Don't use
—	Anti-rollback counter	OTP	Don't use

Note 1. The PBPSm_S[n] bit can be programmed to 0 when the BPSn_S.BPi[3:0] bits are 0000b.
Note 2. The PBPSm_NS[n] bit can be programmed to 0 when the BPSn_NS.BPi[3:0] bits are 0000b.

65.7.2 List of MACI Commands

Table 65.13 List of MACI Commands

MACI Command	Function
Program	This is used to program extra MRAM area except for the configuration area. The program unit is 16 bytes.
Status Clear	This initializes the ILGLERR, PRGERR, ILGCOMERR, OTERR, PZFERR, CFGPRGERR, and PROTERR flags in the MSTATR register and the CMDLK and MREAE flags in the MASTAT register, and the extra MRAM sequencer released from command-locked state.
Forced Stop	This forcibly stops processing of MACI commands and initializes the MSTATR and MASTAT register.
Configuration Set	This is used to program configuration area for setting security function and safety function Units of setting are 16 bytes.
Increment Counter (secure access only)	Anti-rollback counter value can be incremented. The target of counter is selected by the MCNTSELR register. Update data unit is 1 bit.
Read Counter (secure access only)	The value of the OFSM.ARCOEMBL register can be read. Reading data is output to the MCNTDTR0 and MCNTDTR1 registers. Read unit is 64 bit.

The MACI commands are issued by writing to the MACI command-issuing area (4 bytes starting at 8708 0000h). When write access as shown in Table 65.15 proceeds in the specified state, the extra MRAM sequencer executes the processing corresponding to the given command (see section 65.7.3, Relationship Between the Extra MRAM Sequencer State and MACI Commands).

Table 65.14 MACI Command Formats

MACI Commands	Number of Write Access	Write Data to the MACI Command-Issuing Area			
		1st Access	2nd Access	3rd to (N+2)th Access	(N+3)th Access
Program 16-byte programming: N = 8	N + 3	E8h	08h (= N)	WD1 to WDN	D0h
Status Clear	1	50h	—	—	—
Forced Stop	1	B3h	—	—	—
Configuration set N = 8 (16 Bytes)	N + 3	40h	08h (= N)	WD1-WDn	D0h
Increment Counter	2	35h	D0h	—	—
Read Counter	2	39h	D0h	—	—

Note: WDN (N = 1, 2,…): Nth 16-bit data to be programmed.

The extra MRAM sequencer clears the MSTATR.MRDY flag to 0 at the start of processing of a command other than the status clear command and sets this flag to 1 on completion of command processing.
If the setting of the MRDYIE.MRDYIE bit is 1, MRDYI interrupt is generated when the MSTATR.MRDY flag is set to 1.

65.7.3 Relationship Between the Extra MRAM Sequencer State and MACI Commands

The MACI commands are accepted according to the mode/state of the extra MRAM sequencer. MACI commands should be issued after the transition to the extra MRAM program mode and checking of the state of the extra MRAM sequencer. Use the MSTATR and MASTAT registers to check the state of the extra MRAM sequencer. In addition, the occurrence of errors in general can be checked by reading the CMDLK bit in the MASTAT register; its value is the logical OR of the ILGLERR, ILGCOMERR, OTERR, PZFERR, PRGERR, CFGPRGERR, and PROTERR flags in the MSTATR register. Table 65.15 lists the available commands in each operating mode.

Table 65.15 Operating Mode and Available Commands

Operation Mode	MENTRYR	Available Commands
MRE/MRD read mode MRD program mode	0000h	None
Extra MRAM program mode (W-HUK zeroization is not executed)	0080h	Program Status Clear Forced Stop Configuration Increment Counter Read Counter
Extra MRAM program mode (W-HUK zeroization is executed)	0000h/0080h*1	None

Note 1. MRAM is in extra MRAM program mode when W-HUK zeroization is executed regardless of MENTRYR value.

Table 65.16 shows the state of the extra MRAM sequencer and acceptable MACI commands. An appropriate mode is assumed to have been set before the commands are executed.

Table 65.16 Acceptable MACI Commands and the State of the Extra MRAM Sequencer

	Program, Configuration Set, Increment Counter, Read Counter, Forced Stop Command Processing	Command-Locked State (MRDY = 1)	Command-Locked State (MRDY = 0)	W-HUK Zeroization is Executed	Other State
MRDY flag	0	1	0	0/1	1
CMDLK bit	0	1	1	0/1	0
Program	×	×	×	×	✓
Status Clear	×	✓	×	×	✓
Forced Stop	✓	✓	✓	×	✓
Configuration set	×	×	×	×	✓
Increment Counter	×	×	×	×	✓
Read Counter	×	×	×	×	✓

✓: Acceptable
×: Not acceptable (places the sequencer in the command-locked state)

65.7.4 Usage of MACI Commands

This section gives an overview of the usage of MACI commands.

65.7.4.1 Overview of Command Usage in Extra MRAM Program Mode

An overview of MACI command usage in extra MRAM program mode is shown below. For which commands are available in extra MRAM program mode, Table 65.15, Operating Mode and Available Commands.

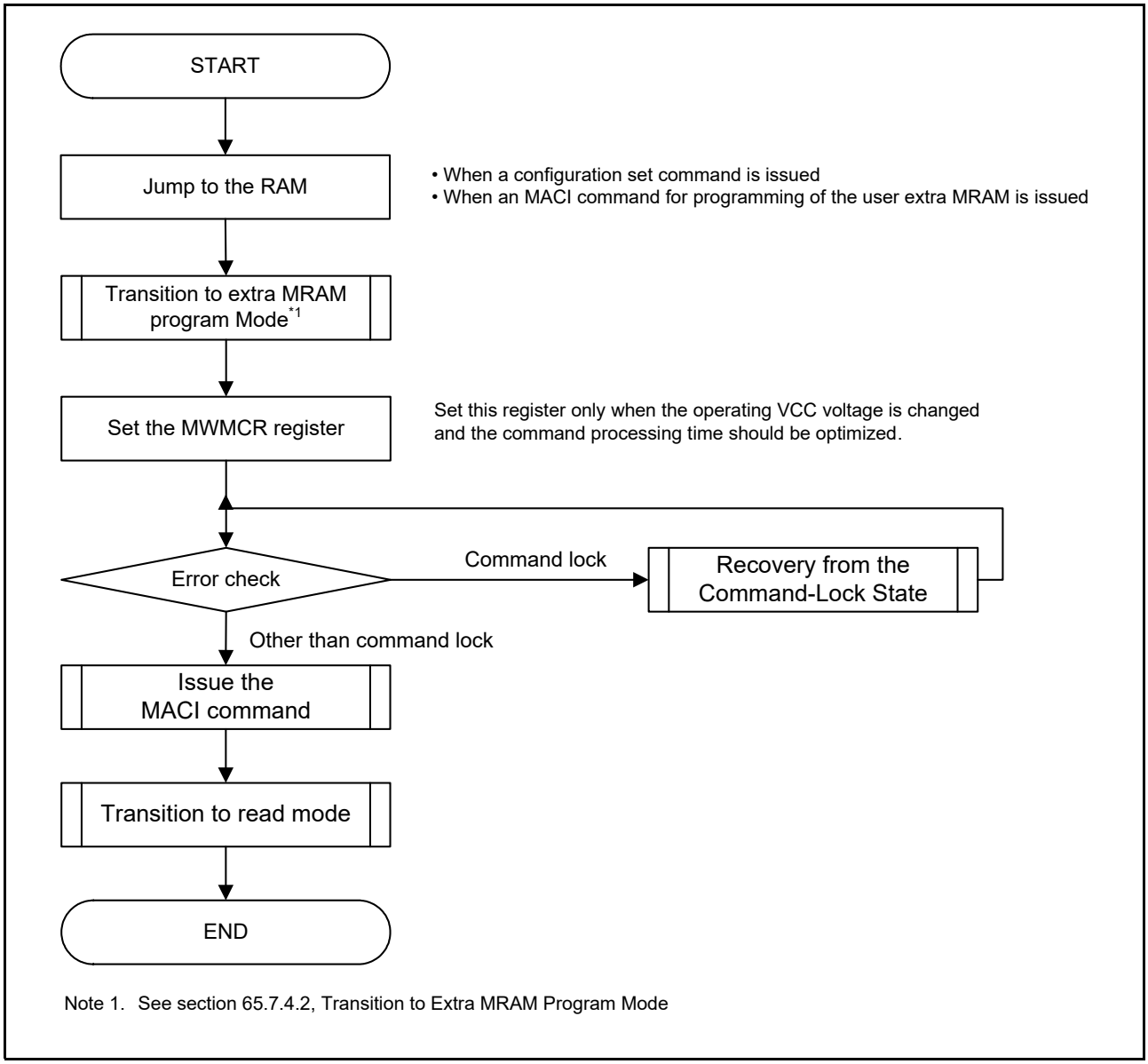


Figure 65.18 Overview of Command Usage in Extra MRAM Program Mode

65.7.4.2 Transition to Extra MRAM Program Mode

To issue the MACI commands for the extra MRAM, a transition to extra MRAM program mode is required. To cause the transition to extra MRAM program mode, set the MENTRY bit in the MENTRYR register to 1.

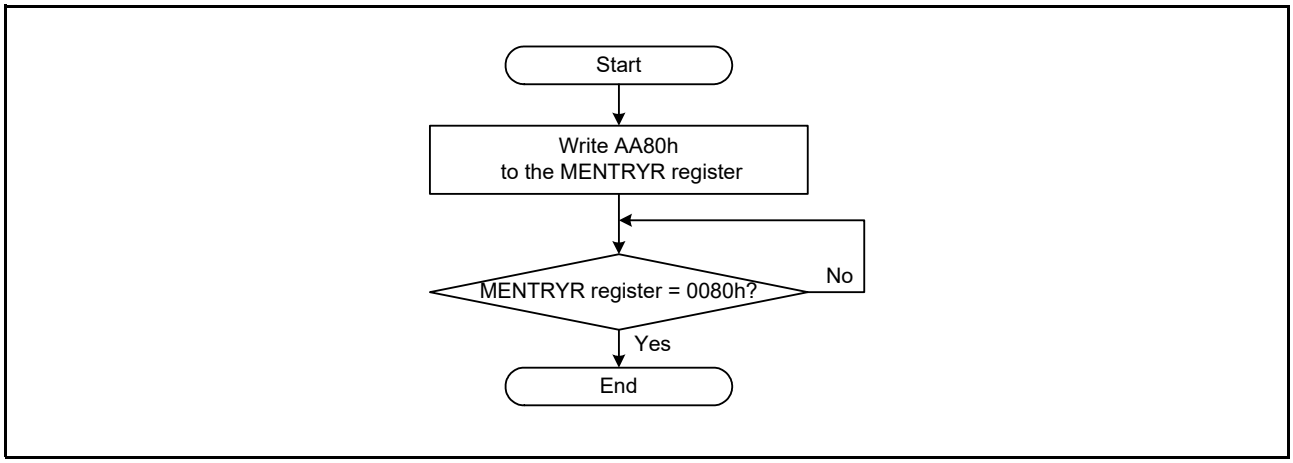
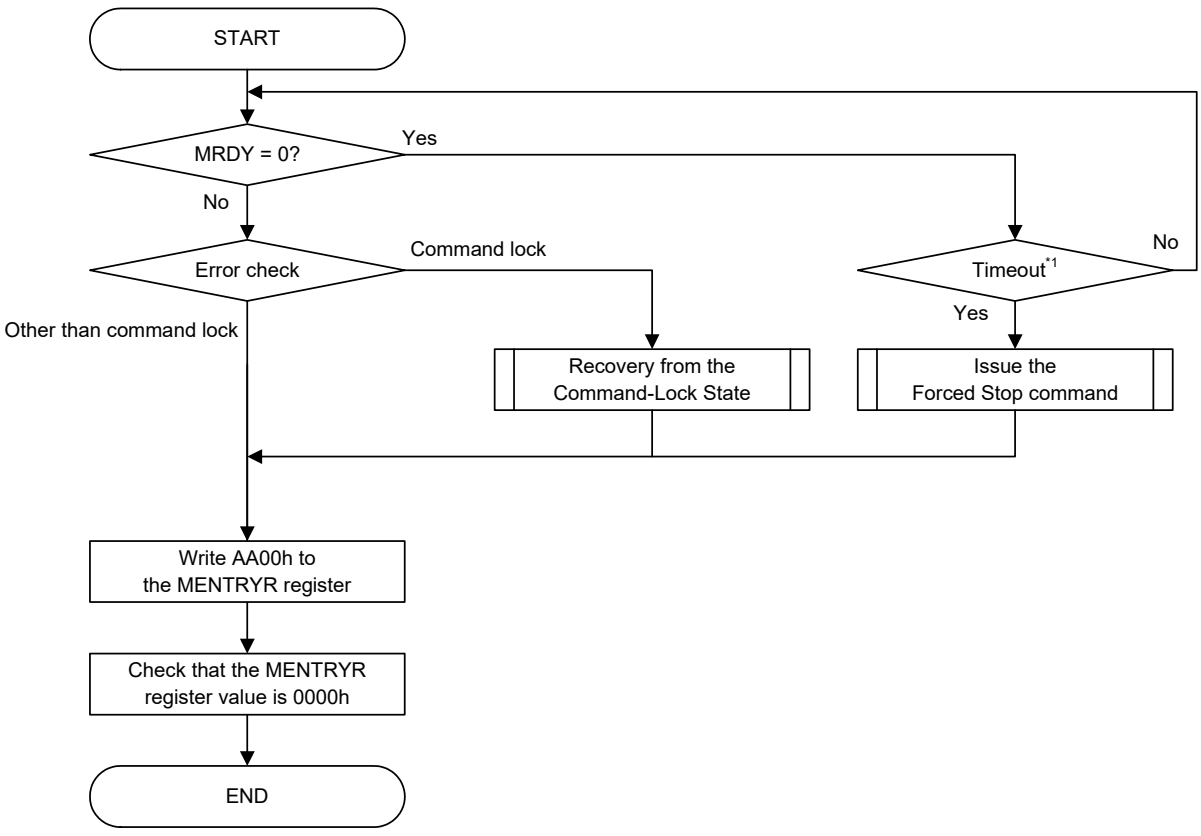


Figure 65.19 Procedure for Transition to Extra MRAM Program Mode

65.7.4.3 Transition to Read Mode

To read the extra MRAM, the MRDY flag in the MSTATR register must be set to 1 or a transition to read mode is required. To cause a transition to read mode, set the MENTRYR register to 0000h. The transition to read mode should be made while processing by the Extra MRAM sequencer is completed and the operation is in a state other than command locked.

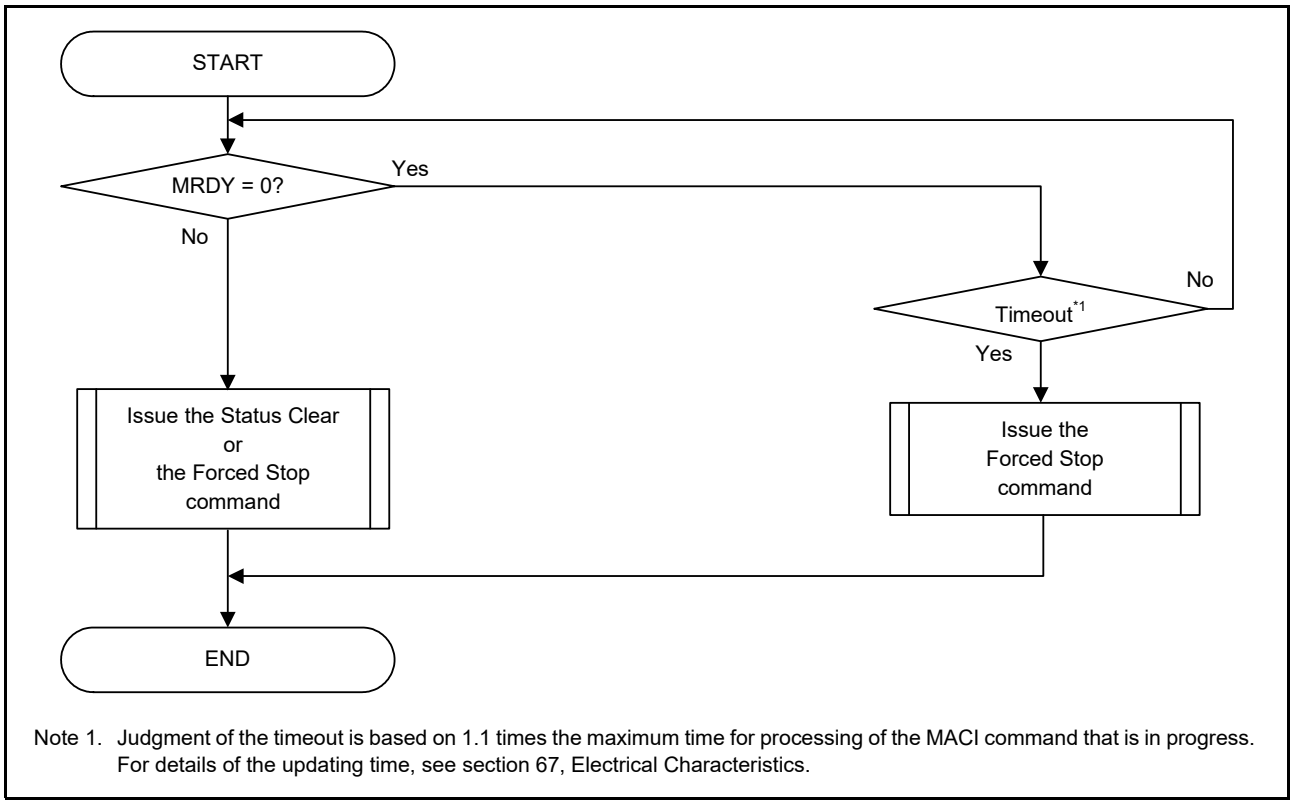


Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the MACI command that is in progress. For details of the updating time, see section 67, Electrical Characteristics.

Figure 65.20 Procedure for Transition to Read Mode

65.7.4.4 Recovery from the Command-Locked State

When the extra MRAM sequencer enters the command-locked state, MACI commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, forced stop command.
If processing is not completed in time beyond the maximum programming time specified in the electrical characteristics, this can be considered a timeout, and the extra MRAM sequencer should be stopped by the forced stop command.

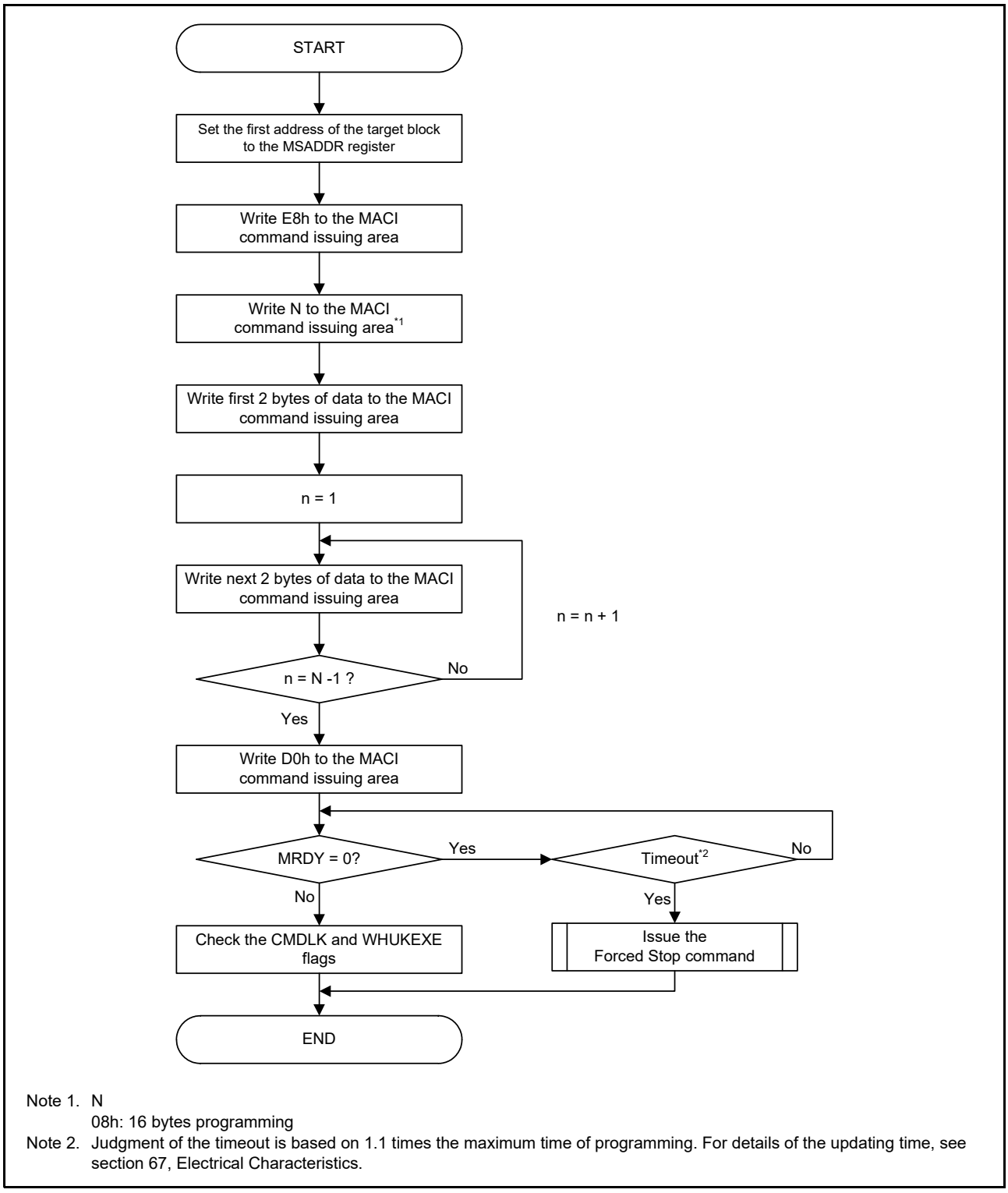


Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the MACI command that is in progress. For details of the updating time, see section 67, Electrical Characteristics.

Figure 65.21 Recovery from the Command-Locked State

65.7.4.5 Program Command

The program command is used for programming to the extra MRAM area, excluding the configuration area. Before issuing the program command, set the first address of the target block in the MSADDR register.
Writing D0h to the MACI command-issuing area at the final access of the MACI command-issuing starts the program command processing.



Note 1. N
08h: 16 bytes programming
Note 2. Judgment of the timeout is based on 1.1 times the maximum time of programming. For details of the updating time, see section 67, Electrical Characteristics.

Figure 65.22 Usage of the Program Command

The correspondence between the possible target data and the address value set in the MSADDR register is shown in Table 65.17. For details on the MSADDR register, see section 65.5.20, MACI Command Start Address Register (MSADDR).

Table 65.17 Address Used by Program Command

Address	MSADDR Register Value	Setting Data	Timing when the Setting is Enabled
BASE_ME+00D8 0000h	BASE_MES + 00D8 0000h to BASE_MES + 00D9 FFF0h	User extra MRAM	—
BASE_ME+00E0 0400h	BASE_MES + 00E0 0400h	FSBL setting	At a reset
BASE_ME+00E0 0410h	BASE_MES + 00E0 0410h to BASE_MES + 00E0 0490h	Start address of measurement report Start address of code certificate	—
BASE_ME+00E0 04A0h	BASE_MES + 00E0 04A0h to BASE_MES + 00E0 04F0h	GPOTP	—
BASE_ME+00E1 0610h	BASE_MES + 00E1 0610h	Secure OFS OTP memory type selection	When a reset or command is executed
BASE_ME+00E1 0620h	BASE_MES + 00E1 0620h to BASE_MES + 00E1 06E0h	Secure OFS OTP setting	At a reset
BASE_ME+00E1 0710h	BASE_MES + 00E1 0710h to BASE_MES + 00E1 07F0h	Secure OFS OTP setting	At a reset
BASE_ME+00E1 0A00h	BASE_MES + 00E1 0A00h to BASE_MES + 00E1 0A70h	Permanent block protect setting for secure	When a reset or command is executed
BASE_ME+00E1 0A80h	BASE_MES + 00E1 0A80h to BASE_MES + 00E1 0A90h	Data MRAM permanent block protection setting for secure application	When a reset or command is executed
BASE_ME+00E2 0810h	BASE_MEN + 00E2 0810h	Non-secure OFS OTP memory type selection	When a reset or command is executed
BASE_ME+00E2 0820h	BASE_MEN + 00E2 0820h to BASE_MEN + 00E2 0870h	Non-secure OFS OTP setting	At a reset
BASE_ME+00E2 0C00h	BASE_MEN + 00E2 0C00h to BASE_MEN + 00E2 0C70h	Permanent block protect setting	When a reset or command is executed
BASE_ME+00E2 0C80h	BASE_MEN + 00E2 0C80h to BASE_MEN + 00E2 0C90h	Data MRAM permanent block protection setting	When a reset or command is executed
BASE_ME+00E3 0E00h	BASE_MES + 00E3 0E00h	Permanent lock setting for secure OFS OTP setting	When a reset or command is executed
BASE_ME+00E3 0E10h	BASE_MES + 00E3 0E10h	REVOKE	When a reset or command is executed
BASE_ME+00E3 0E20h	BASE_MES + 00E3 0E20h	Zeroization HUK in OTP enable	When a reset or command is executed
BASE_ME+00E3 0E30h	BASE_MES + 00E3 0E30h	Anti-rollback counter setting	—
BASE_ME+00E3 0E50h	BASE_MES + 00E3 0E50h	User Extra MRAM setting	When a reset or command is executed
BASE_ME+00E3 0E60h	BASE_MES + 00E3 0E60h	Permanent lock setting for secure OFS configuration setting	When a reset or command is executed
BASE_ME+00E3 0E80h	BASE_MEN + 00E3 0E80h	Permanent lock setting for non-secure OFS OTP setting	When a reset or command is executed
BASE_ME+00E3 0EE0h	BASE_MEN + 00E3 0EE0h	Permanent lock setting for non-secure OFS configuration setting	When a reset or command is executed
BASE_ME+00E3 0EF0h	BASE_MES + 00E3 0EF0h	Reserved	—

65.7.4.6 Status Clear Command

The Status Clear command is used to clear the command-locked state (see section 65.7.4.4, Recovery from the Command-Locked State). To clear the ILGLERR, ILGCOMERR, OTERR, PRGERR, PZFERR, CFGPRGERR, and PROTERR flags in the MSTATR register in the command-locked state, the Status Clear command is available.

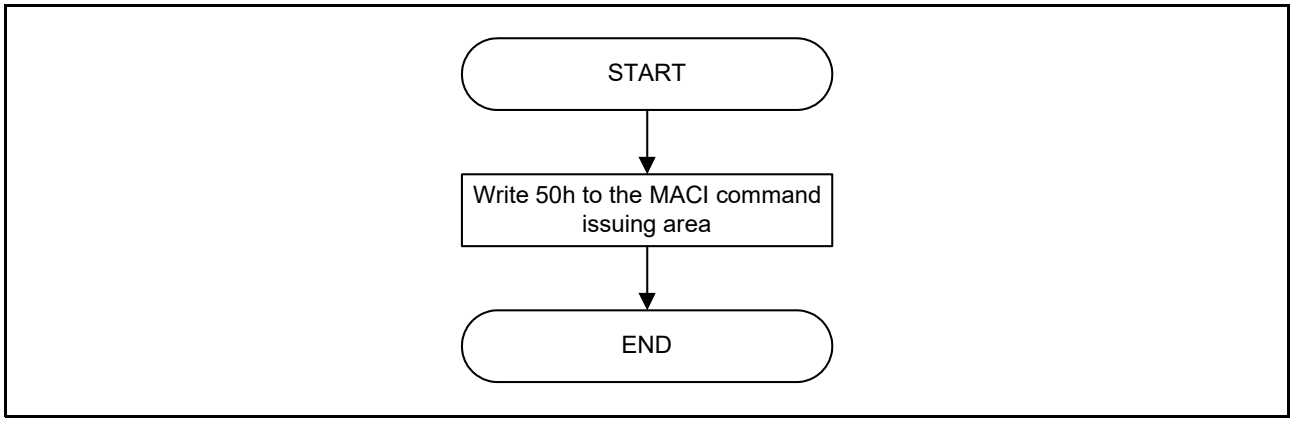
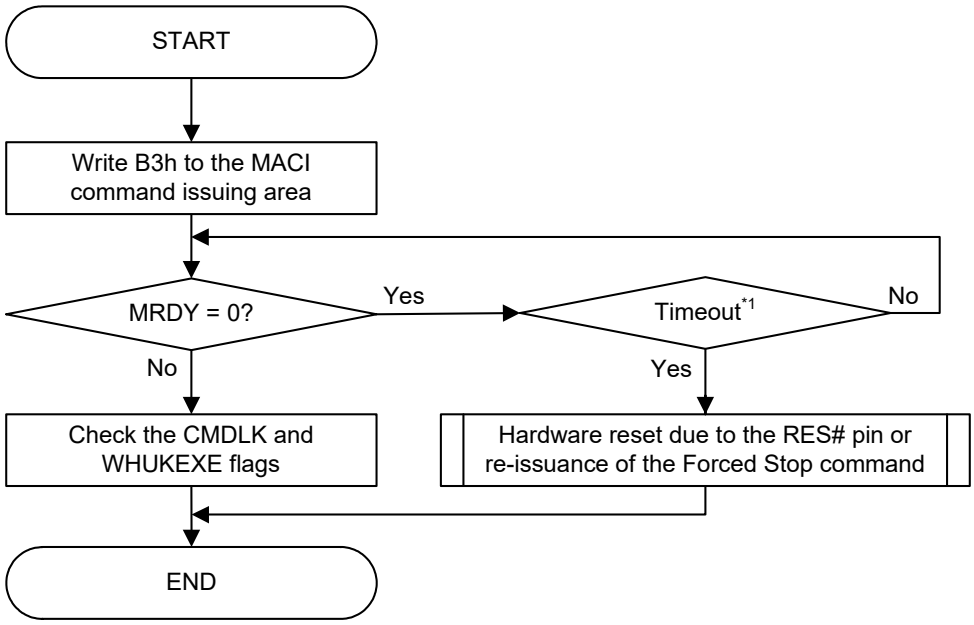


Figure 65.23 Usage of the Status Clear Command

65.7.4.7 Forced Stop Command

The Forced stop command forcibly ends command processing by the extra MRAM sequencer. Although this command halts command processing, values from the programming that was in progress are not guaranteed. Furthermore, resumption of processing is not possible. Processing of programming that was halted by the Forced Stop command is also defined as one from the perspective of programming endurance.
Executing the Forced Stop command also initializes part of the MACI, the whole extra MRAM sequencer, the MSTATR register and the MASTAT register. Accordingly, this command can be used in the procedure for recovery from the command-locked state and in processing in response to a time-out of the extra MRAM sequencer (see section 65.7.4.4, Recovery from the Command-Locked State)

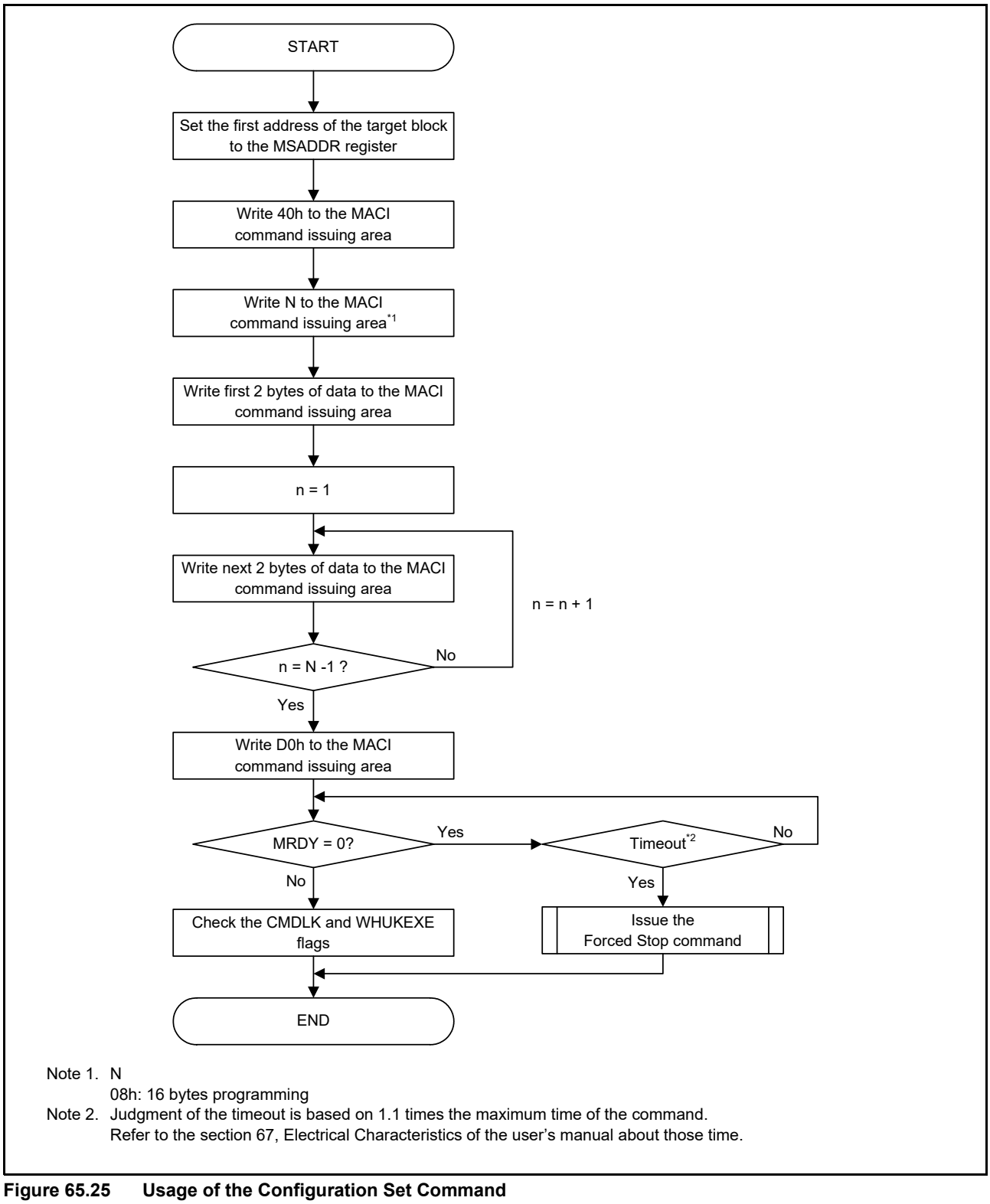


Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the Forced Stop command latency. For details of the updating time, see section 67, Electrical Characteristics.

Figure 65.24 Usage of the Forced Stop Command

65.7.4.8 Configuration Set Command

The configuration set command is used to security function, safety function. Before issuing the configuration set command, set the specified address (shown in Table 65.18) in the MSADDR register. Writing D0h to the MACI command issuing area in the final access for issuing the MACI command starts MACI processing of the configuration set command.



Note 1. N
08h: 16 bytes programming
Note 2. Judgment of the timeout is based on 1.1 times the maximum time of the command.
Refer to the section 67, Electrical Characteristics of the user's manual about those time.

Figure 65.25 Usage of the Configuration Set Command

The correspondence between the possible target data for configuration setting and the address value set in the MSADDR register is shown in Table 65.18. For details on the MSADDR register, see section 65.5.20, MACI Command Start Address Register (MSADDR).

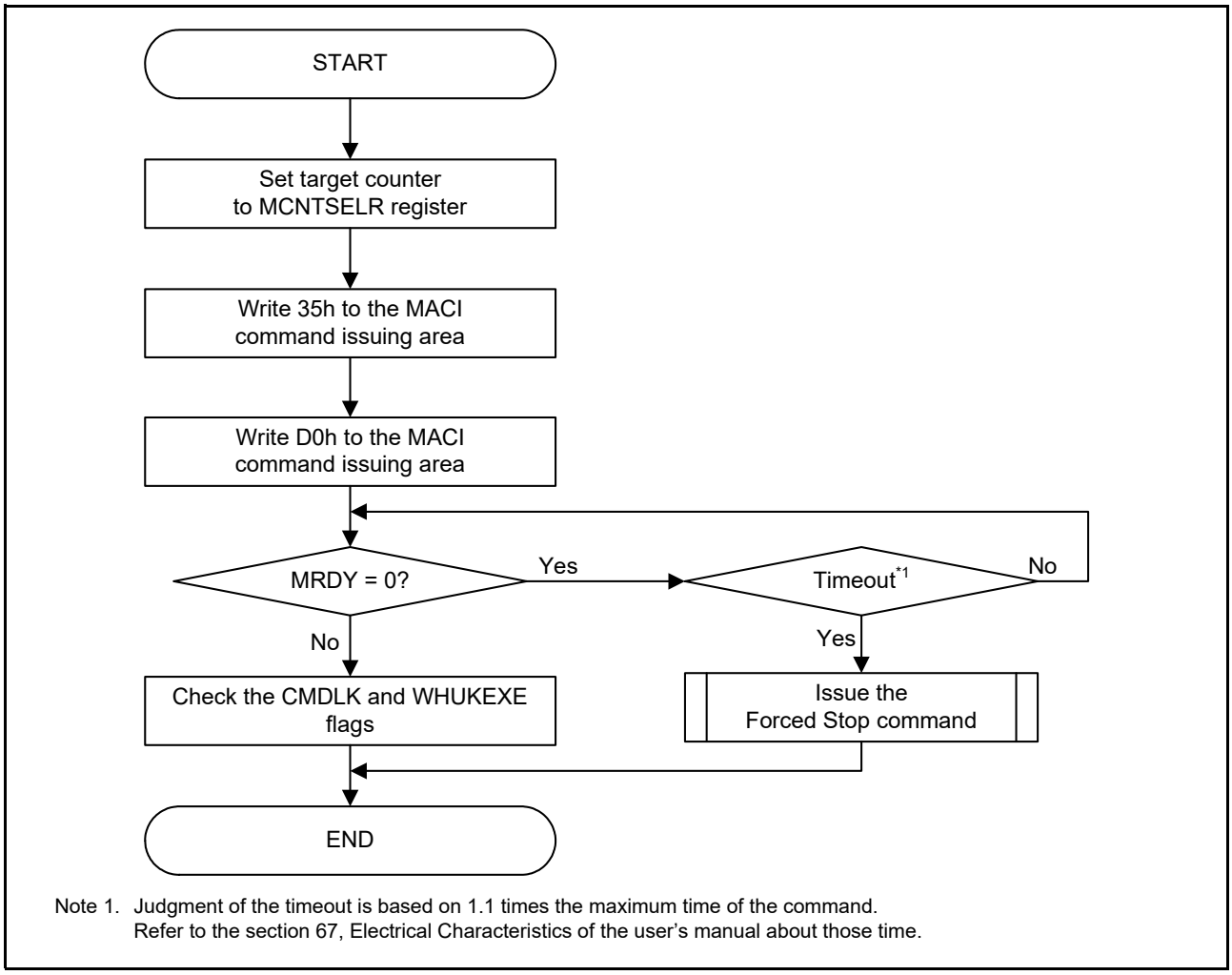
Table 65.18 Address Used by Configuration Set Command

Address	MSADDR Register Value	Setting Data	Operation of Additional Writing	Timing When the Setting is Enabled
BASE_ME + 00C2 E850h	BASE_MES + 00C2 E850h	Bank Mode Select Register (DUALSEL)	Writable*1	At a reset
BASE_ME + 00C2 E870h	BASE_MES + 00C2 E870h	Start-Up Area Setting Register (SAS)	Writable*1	When a reset or command is executed
BASE_ME + 00C2 E874h	BASE_MES + 00C2 E874h	Start-Up Area Setting Register 2 (SAS2)	Writable*1	When a reset or command is executed
BASE_ME + 00C2 E9B0h	BASE_MES + 00C2 E9B0h	Bank Select Register for Secure Application (BANKSEL_S)	Writable*1	At a reset
BASE_ME + 00C2 E9C0h	BASE_MES + 00C2 E9C0h	Block Select Register for Secure Application (BLSEL_S)	Writable*1	At a reset
BASE_ME + 00C2 E9E0h	BASE_MES + 00C2 E9E0h	BANKSEL Attribute Select Register (BANKSEL_SEL)	Writable*1	At a reset
BASE_ME + 00C2 E9F0h	BASE_MES + 00C2 E9F0h	BLSEL Attribute Select Register (BLSEL_SEL)	Writable*1	At a reset
BASE_ME + 00C4 EC60h	BASE_MEN + 00C2 EC60h	Bank Select Register for Non-secure Application (BANKSEL_NS)	Writable*2	At a reset
BASE_ME + 00C4 EC70h	BASE_MEN + 00C2 EC70h	Block Select Register for Non-Secure Application (BLSEL_NS)	Writable*2	At a reset
BASE_ME + 00C9 F050h	BASE_MES + 00C9 F050h	MRAM Security Attribute Boundary Register (MRAMSABR)	Writable	At a reset
BASE_ME + 00C9 F080h	BASE_MES + 00C9 F080h	Data MRAM Block Protect Register 0 for Secure Application (BPSD0_S)	Writable*3	When a reset or command is executed
BASE_ME+00 C9 F100h	BASE_MES + 00C9 F100h	Block Swap Setting Register n for Secure Application (BLKSWP_S)	Writable	At a reset
BASE_ME + 00C9 F200h	BASE_MES + 00C9 F200h to BASE_MES + 00C9 F330h	Block Protect Register n for Secure Application (BPSn_S)	Writable*5	When a reset or command is executed
BASE_ME + 00C9 F480h	BASE_MEN + 00C9 F480h	Data MRAM Block Protect Register 0 for Non-Secure Application (BPSD0_NS)	Writable*4	When a reset or command is executed
BASE_ME + 00C9 F500h	BASE_MEN + 00C9 F500h	Block Swap Setting Register for Non-Secure Application (BLKSWP_NS)	Writable	At a reset
BASE_ME + 00C9 F600h	BASE_MEN + 00C9 F600h to BASE_MEN + 00C9 F730h	Block Protect Register n for Non-Secure Application (BPSn_NS)	Writable*6	When a reset or command is executed

Note 1. This area can be protected by an individual bit in a POFSPSSn register. If the setting of the bit in the POFSPSSn register corresponding to an area for updating is 0, the area cannot be updated by the configuration setting command (n = 0 to 3).
Note 2. This area can be protected by each bit in the POFSPSNS0 register. If the bit in the POFSPSNS0 register corresponding to the update area is 0, it cannot be updated by the configuration set command.
Note 3. Once PBPSD0_S[n] bit is set to 0, the BPSD0_S[n] bit cannot be restored to 1 by using the Configuration set command.
Note 4. Once PBPSD0_NS[n] bit is set to 0, the BPSD0_NS[n] bit cannot be restored to 1 by using the Configuration set command.
Note 5. Once PBPSm_S[n] bit is set to 0, the BPSn_S.BPi[3:0] bits cannot be restored to 1111b by using the Configuration set command.
Note 6. Once PBPSm_NS[n] bit is set to 0, the BPSn_NS.BPi[3:0] bits cannot be restored to 1111b by using the Configuration set command.

65.7.4.9 Increment Counter Command

The Increment Counter command is used to security function, safety function. The increment counter command is used for updating 3 types of anti-rollback counter (ARCS, ARCNS, and ARCOEMBL). The counter update is executed increment processing and cannot be decrement processing.
Writing D0h at 2nd write access of extra MRAM sequencer command triggers MACI to start the increment counter command processing. Completion of command processing can be confirmed by the MRDY flag of MSTATR register.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the command.
Refer to the section 67, Electrical Characteristics of the user's manual about those time.

Figure 65.26 Usage of the Increment Counter Command

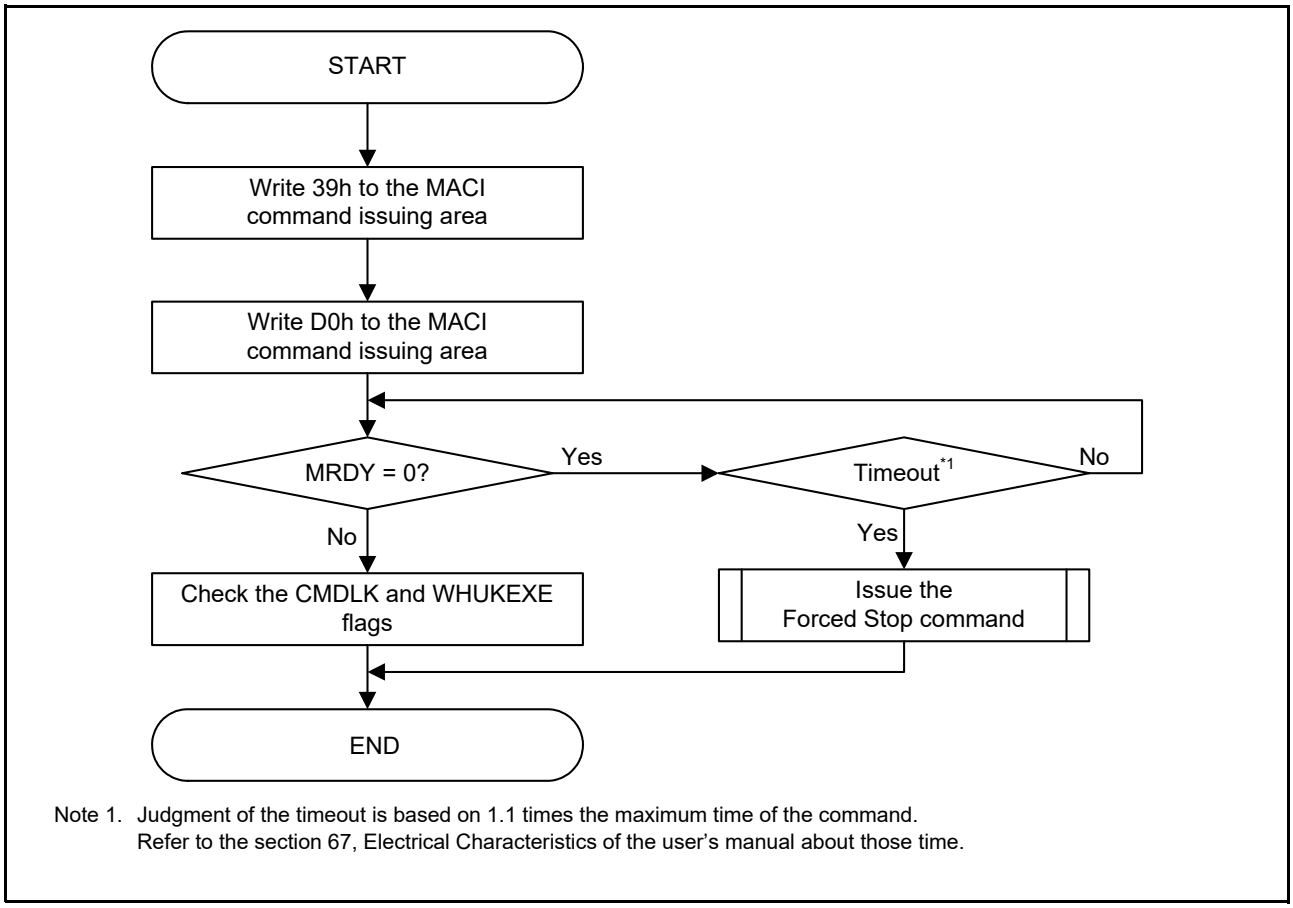
Anti-rollback counter value is not increased the following conditions by the increment counter command.

- The value for updating the ARCOEMBL is equal to or greater than the OEM boot loader version (OEM BL Ver.) output by the RSIP when the ARCCS.ARCMD[1:0] bits are set to 01b (mode1).
- The value for updating the ARCS is equal to or greater than the version value output by the RSIP when the ARCCS.ARCMD[1:0] bits are set to 00b (mode2).
- The ARCCS.ARCMD[1:0] bits are set to an invalid value.
- Target counter value is maximum. (Error does not occur)
- Target counter is ARCNS when ARCCS.CNFARCNS[1:0] is invalid setting.

65.7.4.10 Read Counter Command

The read counter command is used to security function, safety function. The read counter command is used for reading the anti-rollback counter (ARCOEMBL). Extra MRAM sequencer reads 64 bits data, and stores to the MCNTDTR0 and MCNTDTR1 registers.

Writing D0h at 2nd write access of extra MRAM sequencer command triggers MACI to start the read counter command processing. Completion of command processing can be confirmed by the MRDY flag of MSTATR register.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the command.
Refer to the section 67, Electrical Characteristics of the user's manual about those time.

Figure 65.27 Usage of the Read Counter Command

65.8 Security Function

The extra MRAM sequencer supports the following security functions in security enabled product:

- Permanent OFS protect setting
- Security flag for startup area and startup size
- Permanent block protect setting
- MRAM protection for Protect Zone
- Extra MRAM configuration area protection
- Anti-rollback counter
- HUK protection
- OEM_ROOT_KEY protection
- AUTH AL1 KEY, AUTH AL2 KEY, and AUTH RMA KEY protection

65.8.1 Permanent OFS Protect Setting

The secure OFS configuration area, non-secure OFS configuration area, secure OFS OTP area, and non-secure OFS OTP area can be permanently protected by the protection setting bits as listed in Table 65.19 to Table 65.22, respectively. See section 7, Option-Setting Memory (OFSM).

Table 65.19 Example of Protection Settings for the Secure OFS Configuration Area

Protect Register	MSADDR Register Value	Protection Size	Protection Setting Regster	Corresponding Bit	Remarks
DUALSEL	86C2 E820h	4 Bytes	MOFSPS0_S	bit 12	—
SAS	86C2 E820h	4 Bytes	MOFSPS0_S	bit 20	—
SAS2	86C2 E820h	4 Bytes	MOFSPS0_S	bit 21	—
BANKSEL_S	86C2 E9A0h	4 Bytes	MOFSPS3_S	bit 4	—
BLSEL_S	86C2 E9A0h	4 Bytes	MOFSPS3_S	bit 8	—
BANKSEL_SEL	86C2 E9A0h	4 Bytes	MOFSPS3_S	bit 16	—
BLSEL_SEL	86C2 E9A0h	4 Bytes	MOFSPS3_S	bit 20	—

Table 65.20 Example of Protection Settings for the Non-Secure OFS Configuration Area

Protect Register	MSADDR Register Value	Protection Size	Protection Setting Regster	Corresponding Bit	Remarks
BANKSEL_NS	06C4 EC20h	4 Bytes	MOFSPS0_NS	bit 16	—
BLSEL_NS	00C4 EC20h	4 Bytes	MOFSPS0_NS	bit 20	—

Table 65.21 Example of Protection Settings for the Secure OFS OTP Area

Protect Register	MSADDR Register Value	Protection Size	Protection Setting Regster	Corresponding Bit	Remarks
OFS0	86E1 0620h	4 Bytes	SOFSOPS0_S	bit 8	—
OFS2	86E1 0620h	4 Bytes	SOFSOPS0_S	bit 9	—
MDE	86E1 0620h	4 Bytes	SOFSOPS0_S	bit 10	—
OFS1_S	86E1 07A0h	4 Bytes	SOFSOPS3_S	bit 0	—
OFS1_SEL	86E1 07A0h	4 Bytes	SOFSOPS3_S	bit 12	—

Table 65.22 Example of Protection Settings for the Non-Secure OFS OTP Area

Protect Register	MSADDR Register Value	Protection Size	Protection Setting Regster	Corresponding Bit	Remarks
OFS1_NS	06E2 0820h	4 Bytes	SOFSPS0_NS	bit 12	—

65.8.2 Security Flag for Startup Area and Startup Size Select

The security flag (POFSPSS0.POFSPSS0[21] and POFSPSS0.POFSPSS0[20]) for the startup area and startup size are located in the OTP area as the section 65.7.4.5, Program Command. The POFSPSS0[21] and POFSPSS0[20] bits can respectively be monitored by using the FSPR and FSPRS flags in the MSUASMON register. When the POFSPSS0[21] or POFSPSS0[20] bit is 0, issuing a configuration set command to respectively change the BTFLG or BTSIZE bit has no effect. Also, when the POFSPSS0[21] bit is 0, it is not possible to write the startup area select bits (SAS[1:0]) in the MSUACR register.
The POFSPSS0[21] and POFSPSS0[20] bits provide the protection. See section 7, Option-Setting Memory (OFSM).

65.8.3 Permanent Block Protect Setting

The permanent block protect setting disables updating of the block protection setting. The code MRAM and data MRAM can never be updated when the permanent block protect setting is enabled. See section 65.9.1.2, Block Protection of Code MRAM in details of the block protect setting.
The block protect setting and the permanent block protect setting cannot be updated by a configuration set command. Even if a configuration set command is issued, the extra MRAM sequencer does not detect an error. In addition, the protected blocks cannot be updated by a program command.
Figure 65.28, Table 65.23, and Table 65.24 show the protection against updating of the bits in block protect register n for non-secure application (BPSn_NS.BPi[3:0]) and permanent block protect register m for non-secure application (PBPSm_NS[n]).
Figure 65.29, Table 65.25, and Table 65.26 show the protection against updating of the bits in block protect register n for secure application (BPSn_S.BPi[3:0]) and permanent block protect register m for secure application m (PBPSm_S[n]).
Figure 65.30, Table 65.27, and Table 65.28 show the protection against updating of the bits in data MRAM block protect register 0 for non-secure application (BPSD0_NS[n]) and data MRAM permanent block protect register 0 for non-secure application (PBPSD0_NS[n]).
Figure 65.31, Table 65.29, and Table 65.30 show the protection against updating of the bits in data MRAM block protect register 0 for secure application (BPSD0_S[n]) and data MRAM permanent block protect register 0 for secure application (PBPSD0_S[n]).

The setting of the MRAM security attribute boundary register (MRAMSABR) determines which of the BPSn_S and BPSn_NS registers is effective for code MRAM block protection and which of the PBPSm_S and PBPSm_NS registers is effective for code MRAM permanent block protection.
For details of the permanent block protect registers (PBPSm_NS, PBPSm_S, PBPSD0_NS, and PBPSD0_S), see section 7, Option-Setting Memory (OFSM).

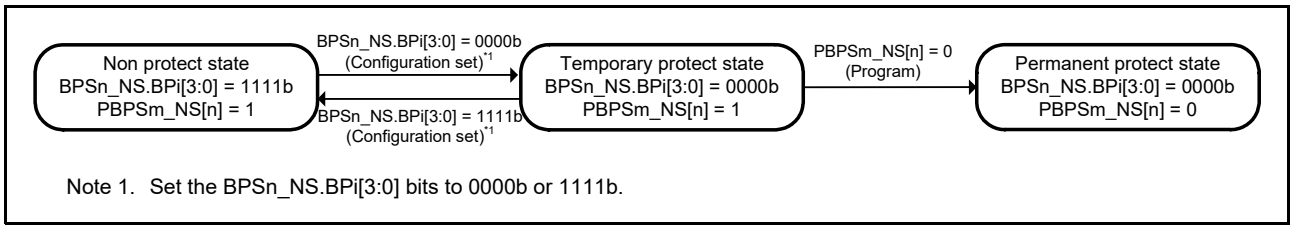


Figure 65.28 Status Transition of Extra MRAM Sequencer by BPSn_NS.BPi[3:0] and PBPSm_NS[n]

Table 65.23 Protection against Updating of BPSn_NS.BPi[3:0]

Current State		Updatable State by Configuration Set Command	
BPSn_NS.BPi[3:0]	**PBPSm_NS[n]**	**BPSn_NS.BPi[3:0] = 1111b**	**BPSn_NS.BPi[3:0] = 0000b**
1111b	1	O	O
1111b	0	—	—
0000b	1	O	O
0000b	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

Table 65.24 Protection against Updating of PBPSm_NS[n]

Current State		Updatable State by Configuration Set Command	
BPSn_NS.BPi[3:0]	**PBPSm_NS[n]**	**PBPSm_NS[n] = 1**	**PBPSm_NS[n] = 0**
0000b	1	O	O
1111b	0	—	—
1111b	1	O	X
0000b	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

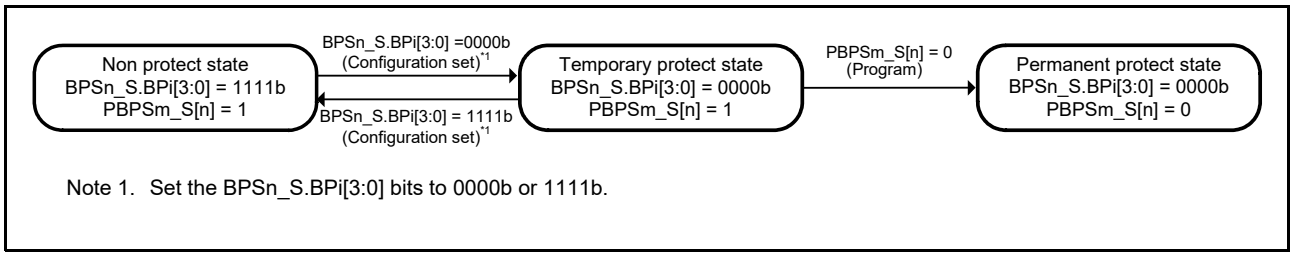


Figure 65.29 Status Transition of Extra MRAM Sequencer by BPSn_S.BPi[3:0] and PBPSm_S[n]

Table 65.25 Protection against Updating of BPSn_S.BPi[3:0]

Current State		Updatable State by Configuration Set Command	
BPSn_S.BPi[3:0]	**PBPSm_S**	**BPSn_S.BPi[3:0] = 1111b**	**BPSn_S.BPi[3:0] = 0000b**
1111b	1	O	O
1111b	0	—	—
0000b	1	O	O
0000b	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

Table 65.26 Protection against Updating of PBPSm_S[n]

Current State		Updatable State by Configuration Set Command	
BPSn_S.BPi[3:0]	**PBPSm_S[n]**	**PBPSm_S[n] = 1**	**PBPSm_S[n] = 0**
0000b	1	O	O
1111b	0	—	—
1111b	1	O	X
0000b	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

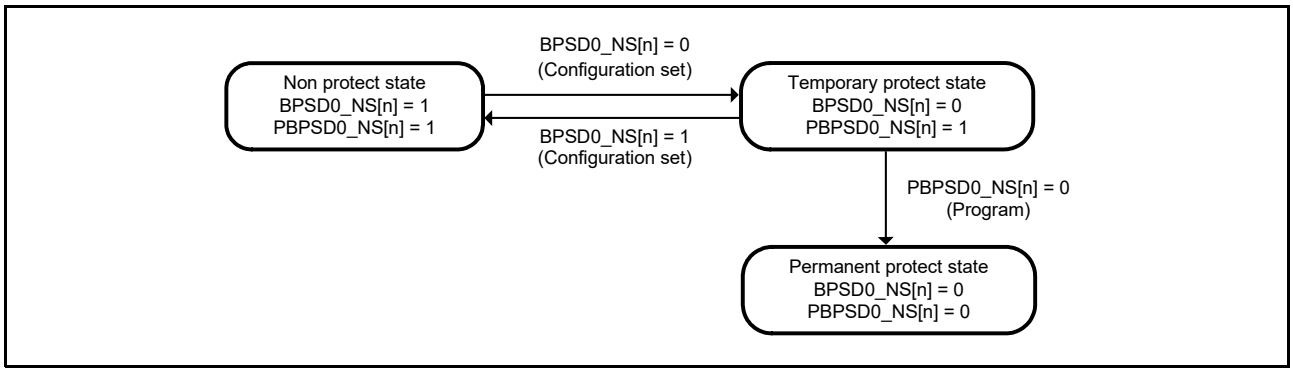


Figure 65.30 Status Transition of Extra MRAM Sequencer by BPSD0_NS[n] and PBPSD0_NS[n]

Table 65.27 Protection against Updating of BPSD0_NS[n]

Current State		Updatable State by Configuration Set Command	
BPSD0_NS[n]	**PBPSD0_NS[n]**	**BPSD0_NS[n] = 1**	**BPSD0_NS[n] = 0**
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

Table 65.28 Protection against Updating of BPSD0_NS[n]

Current State		Updatable State by Configuration Set Command	
BPSD0_NS[n]	**PBPSD0_NS[n]**	**BPSD0_NS[n] = 1**	**BPSD0_NS[n] = 0**
0	1	O	O
1	0	—	—
1	1	O	X
0	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

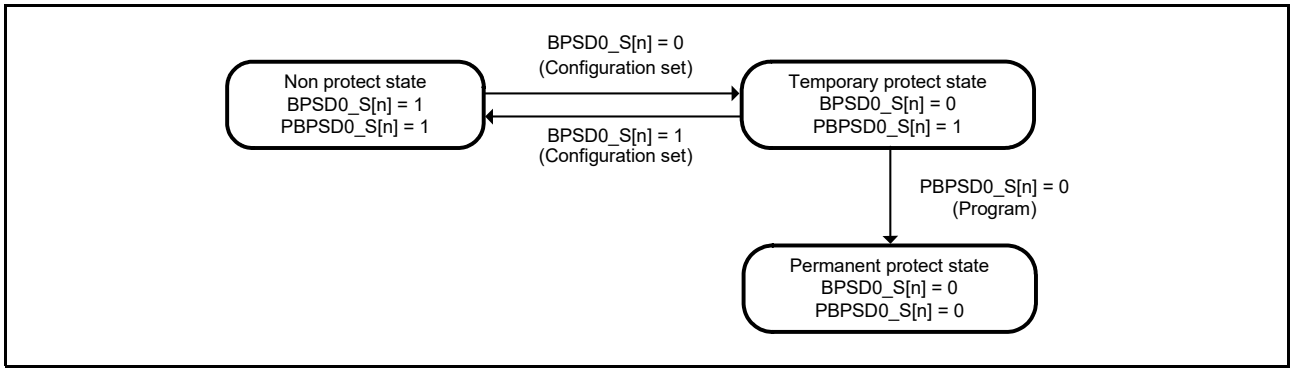


Figure 65.31 Status Transition of Extra MRAM Sequencer by BPSD0_S[n] and PBPSD0_S[n]

Table 65.29 Protection against Updating of BPSD0_S[n]

Current State		Updatable State by Configuration Set Command	
BPSD0_S[n]	**PBPSD0_S[n]**	**BPSD0_S[n] = 1**	**BPSD0_S[n] = 0**
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

Table 65.30 Protection against Updating of PBPSD0_S[n]

Current State		Updatable State by Configuration Set Command	
BPSD0_S[n]	**PBPSD0_S[n]**	**PBPSD0_S[n] = 1**	**PBPSD0_S[n] = 0**
0	1	O	O
1	0	—	—
1	1	O	X
0	0	X	O

O: Updatable by Configuration set command.
X: Not updatable by Configuration set command. (error does not occur.)
—: Not reach to this state.

65.8.4 MRAM Protection for Protect Zone

MRAM has following 4 types of protect function against non-secure access to support the Protect Zone specification for MRAM. Descriptions of this section are focused on extra MRAM sequencer operation.

- Protection for the code MRAM and data MRAM areas (against programming)
- Protection for the extra MRAM area (against programming)
- Protection for the MRAM area (against reading)
- Protection for registers

65.8.4.1 Protection for Code MRAM and Data MRAM Areas (Program)

This function protects code MRAM and data MRAM from bus access such as CPU, DMAC. For details of protection, see section 56, Security Features.

65.8.4.2 Protection for extra MRAM Area (Program)

This function protects secure area of extra MRAM from MRAM sequencer command of non-secure developer. The condition of protection depends on extra MRAM sequencer command, access attribution.
See Table 65.31 in details of protection for extra MRAM area (Program).

Table 65.31 Protection for Extra MRAM Area (Program)

MACI Command		Target Area		Issuing MACI Command by Non-Secure Access	Issuing MACI Command by Secure Access
Program	Extra MRAM	MRAM and OTP area (non-secure area)	Secure area*1	×	×
			Non-secure area*2	✓	✓
		MRAM and OTP area (secure area)	Secure area*1	×	✓
			Non-secure area*2	×	×
Configuration set	Extra MRAM	Configuration area (non-secure area)	Secure area*1	×	×
			Non-secure area*2	✓	✓
		Configuration area (secure area)	Secure area*1	×	✓
			Non-secure area*2	×	×
Increment Counter Read Counter	Extra MRAM	Anti-rollback counter area (secure area only)	—	×	✓

✓: MACI command operation is not prohibited.
×: MACI command operation is prohibited.
Note 1. The secure area is judged when the address of MSADDR[31] is 1.
Note 2. The non-secure area is judged when the address of MSADDR[31] is 0.

Error occurs when the area is selected, and MACI command is executed.
The configuration areas are mapped to both the non-secure areas and secure areas. Figure 65.32 shows the non-secure and secure areas where the configuration areas are mapped.
The extra MRAM sequencer judges that target area is secure region from the MSADDR register setting.

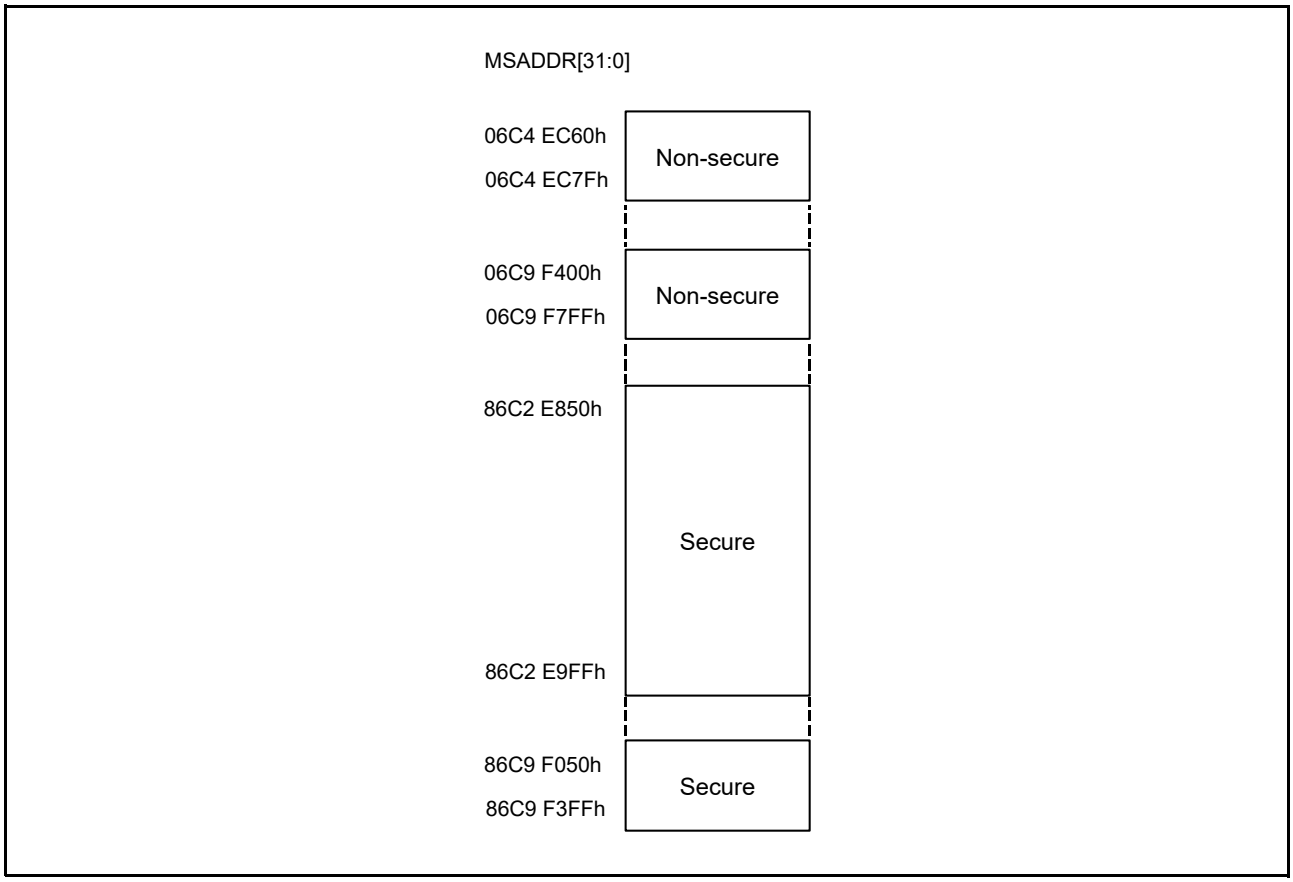


Figure 65.32 Secure/Non-secure Region in Configuration Area

65.8.4.3 Protection for MRAM Area (Read)

This function protects code MRAM, data MRAM, and extra MRAM from bus access (CPU, DMAC, etc.). For details of protection, see section 56, Security Features.

65.8.4.4 Protection for Registers

The MRAM registers have the write access protection against the non-secure access. See Table 65.32 in details of protected the MRAM registers.

Table 65.32 Protected MRAM Registers for Protect Zone (1/2)

Protection Target Register	Secure Attribute Setting	Remarks
MRCEECC	SA register setting (MSAR.MRBCPSA)	section 65.5.48, Code MRAM ECC Encoder Control Register (MRCEECC)
MRPSC, MREEECC	SA register setting (MSAR.MRBEPSA)	section 65.5.32, MRAM Program Speed Control Register (MRPSC) section 65.5.33, Extra MRAM ECC Encoder Control Register (MREEECC)
MRCPCNS, MRCBPROTNS, MRCPS, MRCPAEINT, MRCPEA, MRCPRICR	SA register setting (MSAR.MRCPSA)	section 65.5.34, Code MRAM Program Control Register for Non-Secure Application (MRCPCNS) section 65.5.36, Code MRAM Block Protection Register for Non-Secure Application (MRCBPROTNS) section 65.5.38, Code MRAM Program Status Register (MRCPS) section 65.5.39, Code MRAM Program Access Error Interrupt Enable Register (MRCPAEINT) section 65.5.40, Code MRAM Program Error Address Register (MRCPEA) section 65.5.42, Code MRAM Program Ready Interrupt Control Register (MRCPRICR)
MRDPS, MRDPAEINT, MRDPEA, MRDPRICR	SA register setting (MSAR.MRDPSA)	section 65.5.43, Data MRAM Program Status Register (MRDPS) section 65.5.44, Data MRAM Program Access Error Interrupt Enable Register (MRDPAEINT) section 65.5.45, Data MRAM Program Error Address Register (MRDPEA) section 65.5.47, Data MRAM Program Ready Interrupt Control Register (MRDPRICR)
MSAR	Always secure	section 65.5.14, MRAM Security Attribution Register (MSAR)
MCTRCNTR, MCTRSTATR	SA register setting (MSAR.MACITRSA)	section 65.5.23, MRAM Configuration Update Transfer Control Register (MCTRCNTR) section 65.5.24, MRAM Configuration Update Transfer Status Register (MCTRSTATR)
MRAM Access area	SA register setting (MSAR.MACICOMISA)	section 65.5.14, MRAM Security Attribution Register (MSAR)
MACI*1	SA register setting (MSAR.MACICOMRSA)	section 65.5.14, MRAM Security Attribution Register (MSAR)
MRCPFB	SA register setting (MSAR.MPFBENSA)	section 65.5.1, Code MRAM Prefetch Buffer Enable Register (MRCPFB)
MRCFREQ	SA register setting(MSAR.MRCFREQSA)	section 65.5.2, Code MRAM Frequency Notifications Register (MRCFREQ)
MRCDECC, MRCRAEINT, MRCRAES, MRCRTEA, MRCRDEA	SA register setting (MSAR.MRCECCSA)	section 65.5.4, Code MRAM ECC Decoder Control Register (MRCDECC) section 65.5.5, Code MRAM Read Access Error Interrupt Enable Register (MRCRAEINT) section 65.5.6, Code MRAM Read Access Error Status Register (MRCRAES) section 65.5.7, Code MRAM TED Error Address Register (MRCRTEA) section 65.5.8, Code MRAM DEC Error Address Register (MRCRDEA)
MREFREQ	SA register setting (MSAR.MREFREQSA)	section 65.5.3, Extra MRAM Frequency Notifications Register (MREFREQ)

Table 65.32 Protected MRAM Registers for Protect Zone (2/2)

Protection Target Register	Secure Attribute Setting	Remarks
MREDECC, MRERAEINT, MRERAES, MRERTEA, MRERDEA	SA register setting (MSAR.MREECCSA)	section 65.5.9, Extra MRAM ECC Decoder Control Register (MREDECC) section 65.5.10, Extra MRAM Read Access Error Interrupt Enable Register (MRERAEINT) section 65.5.11, Extra MRAM Read Access Error Status Register (MRERAES) section 65.5.12, Extra MRAM TED Error Address Register (MRERTEA) section 65.5.13, Extra MRAM DEC Error Address Register (MRERDEA)
MREZS, MREZC, MCNTSELR, MCNTDTR0, MCNTDTR1, MRCPCS, MRCBPROTS, MRSTTRST, MSUACR	Always secure	section 65.5.15, Extra MRAM Zeroization Status Register (MREZS) section 65.5.16, Extra MRAM Zeroization Control Register (MREZC) section 65.5.21, MRAM Counter Select Register (MCNTSELR) section 65.5.22, MRAM Counter Data Register n (MCNTDTRn) (n = 0, 1) section 65.5.35, Code MRAM Program Control Register for Secure Application (MRCPCS) section 65.5.37, Code MRAM Block Protection Register for Secure Application (MRCBPROTS) section 65.5.31, MACI Reset Transfer Status Register (MRSTTRST) section 65.5.30, MRAM Startup Area Control Register (MSUACR)

Note 1. The target MACI registers are MASTAT, MPAEINT, MRDYIE, MSADDR, MSTATR, MENTRYR, MSUINITR, MCMDR and MSUASMON.

65.8.5 Anti-Rollback Counter

Anti-rollback counter cannot be decremented and is used in the firmware update. Only secure applications are available. Anti-rollback counter shifts one bit to the left with the increment counter command from the initial value of 0. The least significant bit (bit 0) is set to 1. When anti-rollback counter is of maximum value, extra MRAM sequencer does not change counter by the Increment Counter command (does not occur error).
There are three commands that control the counters. See section 65.7.2, List of MACI Commands for details.
Figure 65.33 shows anti-rollback counter increment image when increment counter command is issued.

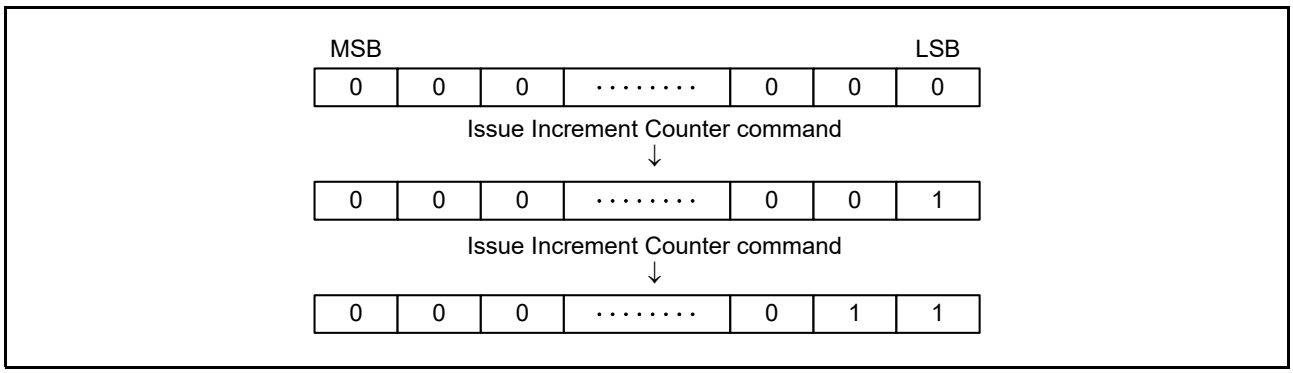


Figure 65.33 Increment Image of Anti-Rollback Counter

Anti-rollback counter has 3 types of counters (ARCS, ARCNS and ARCOEMBL). If ARCCS.ARCMD[1:0] bits are set to 11b, the increment counter command has no effect on any counters.
The ARCNS can be configured as 1 × 1024 bits or 4 × 256 bits by the ARCNS counter configuration select bits (ARCCS.CNFARCNS[1:0]) and cannot be updated to an invalid configuration.
The ARCOEMBL is protected by the RSIP and cannot be updated until the updated OEM boot loader has been successfully validated. After the successful validation, the ARCOEMBL can be updated to the current OEM boot loader version (OEM BL Ver.) that was output by the RSIP. The ARCOEMBL cannot be updated to a value greater than the OEM boot loader version (OEM BL Ver.).

65.8.6 HUK Protection

Zeroization of W-HUK is triggered when one of the ZHUKEN[3:0] bits (zeroization enable bit) is 0 and anti-tamper is detected or 07h is set to the WHUKWE[2:0] bits in MREZC register. Zeroing W-HUK disables the generation of HUK. Table 65.33 shows the combinations of enable bits and request signals. If a zeroing request is received during code MRAM, data MRAM, or extra MRAM program, the executing program is forcibly terminated, and the zeroing process is started. If code MRAM program request is received during zeroing, code MRAM program is started after zeroing is completed. MACI commands are not accepted during zeroing. During W-HUK zeroing, all MACI commands are not accepted and extra MRAM cannot be read.

Table 65.33 Combinations of Request Signals and Enable bits

Zeroization Request Signal				Zeroization Enable Bit				
	RTCIC			ZHUKEN				
WHUKZE[2:0]	2	1	0	3	2	1	0	Function
7h	—	—	—	0	—	—	—	O
7h	—	—	—	1	—	—	—	Oe
Except 7h	—	—	—	—	—	—	—	×
—	1	—	—	—	0	—	—	O
—	1	—	—	—	1	—	—	Oe
—	0	—	—	—	—	—	—	×
—	—	1	—	—	—	0	—	O
—	—	1	—	—	—	1	—	Oe
—	—	0	—	—	—	—	—	×
—	—	—	1	—	—	—	0	O
—	—	—	1	—	—	—	1	Oe
—	—	—	0	—	—	—	—	×

O: W-HUK zeroing.
Oe: W-HUK Pseudo-zeroing
×: Zeroization of W-HUK does not execute
—: Don't care

Table 65.34 shows differences of W-HUK pseudo-zeroing and W-HUK zeroing described in Table 65.33. If the zeroization enable bit is 1 and tamper is detected, the MREZS.WHUKZF bit is set to 0 to complete zeroing without physically zeroing W-HUK (W-HUK pseudo-zeroing). If the zeroization enable bit is 0 and a tamper is detected, physical zeroization of W-HUK is performed and the Zeroized Monitor register is set to 0 (W-HUK zeroing).

Table 65.34 Internal State at the Time of Zeroing Request

Table 65.33 Function	Operation Name	Zeroization Enable Bits	W-HUK	W-HUK Zeroized Monitor Register (MREZS)
Oe	W-HUK Pseudo-zeroing	Disable(1)	Not zeroed	Clear
O	W-HUK zeroing	Enable(0)	Zeroed	Clear flag after zeroization finished

65.8.7 OEM Root Key Protection

The REVOKE bit located in the OTP area is used to delete the OEM root key. When the REVOKE bit is 0, the read data of the target OEM root key is masked to 0.

65.9 Protection Function

65.9.1 Software Protection

Software protection function disables MRAM programming operation according to register and configuration area settings.

65.9.1.1 Code MRAM Program Protection

When MRCPCNS register is set to 0000h, programming of non-secure areas in code MRAM is prohibited. When MRCPCS register is set to 0000h, programming of secure areas in code MRAM is prohibited. Code MRAM program protection has higher priority than block protection and cannot be canceled by MRCBPROTNS or MRCBPROTS register.

65.9.1.2 Block Protection of Code MRAM

Each block in the code MRAM has a block protect register (BPSn_NS or BPSn_S) for that block. When the MRCBPROTNS or MRCBPROTS register is set to 0000h and the BPSn_NS.BPi[3:0] bits for the target block for protection are set to 0000b, access for programming of the non-secure area of the target block in the code MRAM is prohibited.
When the MRCBPROTS register is set to 0000h and the BPSn_S.BPi[3:0] bits for the target block for protection are set to 0000b, access for programming of the secure area of the target block in the code MRAM is prohibited. To program the block for which the BPSn_S.BPi[3:0] or BPSn_NS.BPi[3:0] bits are set to 0000b, set the MRCBPROTNS or MRCBPROTS register to 0001h.
The block protect setting can be locked by the permanent block protect register (PBPSm_S or PBPSm_NS). When a bit in the permanent block protect register or block protect register is set 0, program access to the corresponding block cannot be performed.
The setting of the MRAM security attribute boundary register (MRAMSABR) determines which of the BPSn_S and BPSn_NS registers is effective for the setting of the target block for block protection.
Likewise, the setting of the MRAMSABR register determines which of the PBPSm_S and PBPSm_NS registers is effective for the setting of the target block for permanent block protection.
See section 65.8.3, Permanent Block Protect Setting in details of the permanent block protect setting and block protect setting.
See section 65.5.36, Code MRAM Block Protection Register for Non-Secure Application (MRCBPROTNS) and section 65.5.37, Code MRAM Block Protection Register for Secure Application (MRCBPROTS) for the MRCBPROTNS and MRCBPROTS registers.
For details of block protect setting (BPS_S or BPS_NS), permanent block protect setting block (PBPS_S or PBPS_NS) and code MRAM boundary setting, see section 7, Option-Setting Memory (OFSM).
Table 65.35 to Table 65.40 show the correspondence between the block protect settings and the target blocks in some examples of function settings.

- In linear mode, BPSn_NS.BPi[3:0] or BPSn_S.BPi[3:0] are assigned to the blocks in the code MRAM (e.g., the addresses are the last block address to BASE_MC + 00FF FFFFh).
- BPSn_NS[31:0] and BPSn_S[31:0] are assigned to blocks in the code MRAM according to the settings of the startup area select bit (SAS2.BTFLG) and startup area size setting bits (SAS.BTSIZE[1:0]). For details, see section 65.9.3, Startup Program Protection.
- In dual mode, BPS0_NS.BP0[3:0] to BPS7_NS.BP7[3:0] or BPS0_S.BP0[3:0] to BPS7_S.BP7[3:0] are assigned to the blocks in bank 0 (MRC0) of the code MRAM.
  BPS64_NS.BP0[3:0] to BPS71_NS.BP7[3:0] or BPS64_S.BP0[3:0] to BPS71_S.BP7[3:0] are assigned to the blocks in bank 1 (MRC1) of the code MRAM. The offset addresses for bank 0 are F8 0000h to FF FFFFh and those

for bank 1 are B8 0000h to BF FFFFh in the products with 1-Mbyte code MRAM.
For details, see section 65.9.4, Dual Bank Function.

Table 65.35 to Table 65.40 show example of the block protect setting.

Table 65.35 Example of Block Protect Setting when SAS2.BTFLG bit is 1 and All Sizes (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS15_NS.BP7[3:0] or BPS15_S.BP7[3:0]	F0 0000h to F0 1FFFh	
:	:	
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	

Table 65.36 Example of Block Protect Setting when SAS2.BTFLG bit is 0 and SAS.BTSIZE[1:0] bits are 11b (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS15_NS.BP7[3:0] or BPS15_S.BP7[3:0]	F0 0000h to F0 1FFFh	—
:	:	
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	—
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	The contents of FF C000h to FF DFFFh and FF E000h to FF FFFFh are swapped in this startup area selection setting.
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	The contents of FF E000h to FF FFFFh and FF C000h to FF DFFFh are swapped in this startup area selection setting.

Table 65.37 Example of Block Protect Setting when SAS2.BTFLG bit is 0 and SAS.BTSIZE[1:0] bits are 10b (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS15_NS.BP7[3:0] or BPS15_S.BP7[3:0]	F0 0000h to F0 1FFFh	—
:	:	
BPS0_NS.BP4[3:0] or BPS0_S.BP4[3:0]	FF 6000h to FF 7FFFh	—
BPS0_NS.BP3[3:0] or BPS0_S.BP3[3:0]	FF 8000h to FF 9FFFh	The contents of FF 8000h to FF 9FFFh and FF C000h to FF DFFFh will be swapped in this startup area selection setting.
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	The contents of FF A000h to FF BFFFh and FF E000h to FF FFFFh will be swapped in this startup area selection setting.
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	The contents of FF C000h to FF DFFFh and FF 8000h to FF 9FFFh will be swapped in this startup area selection setting.
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	The contents of FF E000h to FF FFFFh and FF A000h to FF BFFFh will be swapped in this startup area selection setting.

Table 65.38 Example of Block Protect Setting when SAS2.BTFLG bit is 0 and SAS.BTSIZE[1:0] bits are 01b (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS15_NS.BP7[3:0] or BPS15_S.BP7[3:0]	F0 0000h to F0 1FFFh	—
:	:	
BPS1_NS.BP0[3:0] or BPS1_S.BP0[3:0]	FE E000h to FE FFFFh	—
BPS0_NS.BP7[3:0] or BPS0_S.BP7[3:0]	FF 0000h to FF 1FFFh	The contents of FF 0000h to FF 1FFFh and FF 8000h to FF 9FFFh will be swapped in this startup area selection setting.
BPS0_NS.BP6[3:0] or BPS0_S.BP6[3:0]	FF 2000h to FF 3FFFh	The contents of FF 2000h to FF 3FFFh and FF A000h to FF BFFFh will be swapped in this startup area selection setting.
BPS0_NS.BP5[3:0] or BPS0_S.BP5[3:0]	FF 4000h to FF 5FFFh	The contents of FF 4000h to FF 5FFFh and FF C000h to FF DFFFh will be swapped in this startup area selection setting.
BPS0_NS.BP4[3:0] or BPS0_S.BP4[3:0]	FF 6000h to FF 7FFFh	The contents of FF 6000h to FF 7FFFh and FF E000h to FF FFFFh will be swapped in this startup area selection setting.
BPS0_NS.BP3[3:0] or BPS0_S.BP3[3:0]	FF 8000h to FF 9FFFh	The contents of FF 8000h to FF 9FFFh and FF 0000h to FF 1FFFh will be swapped in this startup area selection setting.
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	The contents of FF A000h to FF BFFFh and FF 2000h to FF 3FFFh will be swapped in this startup area selection setting.
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	The contents of FF C000h to FF DFFFh and FF 4000h to FF 5FFFh will be swapped in this startup area selection setting.
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	The contents of FF E000h to FF FFFFh and FF 6000h to FF 7FFFh will be swapped in this startup area selection setting.

Table 65.39 Example of Block Protect Setting when DUALSEL.BANKMD[2:0] bits are 000b and BANKSEL.BANKSWP[2:0] bits are 111b (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS71_NS.BP7[3:0] or BPS71_S.BP7[3:0]	B8 0000h to B8 1FFFh	
:	:	
BPS64_NS.BP7[3:0] or BPS64_S.BP7[3:0]	BF 0000h to BF 1FFFh	
BPS64_NS.BP6[3:0] or BPS64_S.BP6[3:0]	BF 2000h to BF 3FFFh	
BPS64_NS.BP5[3:0] or BPS64_S.BP5[3:0]	BF 4000h to BF 5FFFh	
BPS64_NS.BP4[3:0] or BPS64_S.BP4[3:0]	BF 6000h to BF 7FFFh	
BPS64_NS.BP3[3:0] or BPS64_S.BP3[3:0]	BF 8000h to BF 9FFFh	
BPS64_NS.BP2[3:0] or BPS64_S.BP2[3:0]	BF A000h to BF BFFFh	
BPS64_NS.BP1[3:0] or BPS64_S.BP1[3:0]	BF C000h to BF DFFFh	
BPS64_NS.BP0[3:0] or BPS64_S.BP0[3:0]	BF E000h to BF FFFFh	
—	—	Reserve area
BPS7_NS.BP7[3:0] or BPS7_S.BP7[3:0]	F8 0000h to F8 1FFFh	
:	:	
BPS0_NS.BP7[3:0] or BPS0_S.BP7[3:0]	FF 0000h to FF 1FFFh	
BPS0_NS.BP6[3:0] or BPS0_S.BP6[3:0]	FF 2000h to FF 3FFFh	
BPS0_NS.BP5[3:0] or BPS0_S.BP5[3:0]	FF 4000h to FF 5FFFh	
BPS0_NS.BP4[3:0] or BPS0_S.BP4[3:0]	FF 6000h to FF 7FFFh	
BPS0_NS.BP3[3:0] or BPS0_S.BP3[3:0]	FF 8000h to FF 9FFFh	
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	

Table 65.40 Example of Block Protection Settings when DUALSEL.BANKMD[2:0] bits are 000b and BANKSEL.BANKSWP[2:0] bits are 000b (Code MRAM Size: 1 Mbyte)

Block Protect Setting	Address (Offset address)	Remarks
BPS71_NS.BP7[3:0] or BPS71_S.BP7[3:0]	B8 0000h to B8 1FFFh	The contents of Bank 1 and Bank 0 are swapped.
:	:	
BPS64_NS.BP7[3:0] or BPS64_S.BP7[3:0]	BF 0000h to BF 1FFFh	
BPS64_NS.BP6[3:0] or BPS64_S.BP6[3:0]	BF 2000h to BF 3FFFh	
BPS64_NS.BP5[3:0] or BPS64_S.BP5[3:0]	BF 4000h to BF 5FFFh	
BPS64_NS.BP4[3:0] or BPS64_S.BP4[3:0]	BF 6000h to BF 7FFFh	
BPS64_NS.BP3[3:0] or BPS64_S.BP3[3:0]	BF 8000h to BF 9FFFh	
BPS64_NS.BP2[3:0] or BPS64_S.BP2[3:0]	BF A000h to BF BFFFh	
BPS64_NS.BP1[3:0] or BPS64_S.BP1[3:0]	BF C000h to BF DFFFh	
BPS64_NS.BP0[3:0] or BPS64_S.BP0[3:0]	BF E000h to BF FFFFh	
—	—	Reserve area
BPS7_NS.BP7[3:0] or BPS7_S.BP7[3:0]	F8 0000h to F8 1FFFh	The contents of Bank 0 and Bank 1 are swapped.
:	:	
BPS0_NS.BP7[3:0] or BPS0_S.BP7[3:0]	FF 0000h to FF 1FFFh	
BPS0_NS.BP6[3:0] or BPS0_S.BP6[3:0]	FF 2000h to FF 3FFFh	
BPS0_NS.BP5[3:0] or BPS0_S.BP5[3:0]	FF 4000h to FF 5FFFh	
BPS0_NS.BP4[3:0] or BPS0_S.BP4[3:0]	FF 6000h to FF 7FFFh	
BPS0_NS.BP3[3:0] or BPS0_S.BP3[3:0]	FF 8000h to FF 9FFFh	
BPS0_NS.BP2[3:0] or BPS0_S.BP2[3:0]	FF A000h to FF BFFFh	
BPS0_NS.BP1[3:0] or BPS0_S.BP1[3:0]	FF C000h to FF DFFFh	
BPS0_NS.BP0[3:0] or BPS0_S.BP0[3:0]	FF E000h to FF FFFFh	

65.9.1.3 Block Protection of Data MRAM

Each block in the data area has data MRAM block protect register 0 (BPSD0_S or BPSD0_NS) for that block.
The block protect setting can be locked by data MRAM permanent block protect register 0 (PBPSD0_S or PBPSD0_NS).
Access for programming of a block cannot be performed when the corresponding bits in data MRAM permanent block protect register 0 and data MRAM block protect register 0 are set to 0.
The setting of the MRAM security attribute boundary register (MRAMSABR) determines which of the BPSD0_S and BPSD0_NS registers is effective for the setting of data MRAM block protection and which of the PBPSD0_S and PBPSD0_NS registers is effective for the setting of data MRAM permanent block protection.
See section 65.8.3, Permanent Block Protect Setting. For details of data MRAM block protect register 0 (BPSD0_S or BPSD0_NS), data MRAM permanent block protect register 0 (PBPSD0_S or PBPSD0_NS), and MRAM security attribute boundary register (MRAMSABR), see section 7, Option-Setting Memory (OFSM).
Table 65.41 shows an example of block protect settings for the data MRAM.

Table 65.41 Example of Block Protection Settings for Data MRAM

Data MRAM Block Protect Setting	Address (Offset address)
BPSD0_NS.BP23 or BPSD0_S.BP23	90 B800h to 90 BFFFh
:	:
BPSD0_NS.BP3 or BPSD0_S.BP3	90 1800h to 90 1FFFh
BPSD0_NS.BP2 or BPSD0_S.BP2	90 1000h to 90 17FFh
BPSD0_NS.BP1 or BPSD0_S.BP1	90 0800h to 90 0FFFh
BPSD0_NS.BP0 or BPSD0_S.BP0	90 0000h to 90 07FFh

65.9.2 Error Protection

Error protection detects the issuing of illegal MACI commands, illegal access, and extra MRAM sequencer malfunction. MACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The extra MRAM cannot be programmed while the extra MRAM sequencer is in the command-locked state. For release from the command-locked state, issue the status clear or forced stop command. The status clear command can only be used while the MRDY flag in the MSTATR register is 1. The forced stop command can be used regardless of the value of the MRDY flag. While the CMDLKIE bit in the MPAEINT register is 1, an extra MRAM access error (MREPRI) interrupt is generated if the extra MRAM sequencer enters the command-locked state (the CMDLK bit in the MASTAT register is set to 1).

If the extra MRAM sequencer enters the command-locked state in response to a command, the extra MRAM sequencer continues the processing for programming. If a command is issued in the command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to previous error detection.

Table 65.42 shows error protection types and status bit values after error detection.

Table 65.42 Error Protection Types (1/2)

Error Type	Description	ILGCO MERR	OTERR	PZFERR	ILGLERR	PRGERR	CFGPRGERR	PROTERR	MREAE
Illegal command error	An undefined size has been specified in the first cycle of command. (Not Byte write)	1	0	0	1	0	0	0	0
	An undefined code is written in the first access of the MACI command.	1	0	0	1	0	0	0	0
	The value specified in the last access of the multiple access MACI command is not D0h.	1	0	0	1	0	0	0	0
	The value (N) specified in the second write access of the MACI command in the program command or Configuration set command is wrong.	1	0	0	1	0	0	0	0
	Increment counter or read counter command has been issued with the ARCMD[1:0] bits or MCNTSLER[2:0] bits being set to an incorrect value. (Including the case where the CNFARCNS[1:0] setting is incorrect)	1	0	0	1	0	0	0	0
	The MACI command not acceptable in each mode has been issued (see Table 65.16).	1	0	0	1	0	0	0	0
	The MACI command has been issued when command acceptance conditions are not satisfied (see Table 65.17).	0/1	0/1	0	1	0/1	0/1	0	0/1
	Increment counter command for updating the version to an illegal value has been issued. (see section 65.8.5, Anti-Rollback Counter).	1	0	0	1	0	0	0	0
	A command has been issued against extra MRAM sequencer command illegal status (see Table 65.13).	0/1	0/1	0	1	0/1	0/1	0	0/1
Program error	An error occurs during programming. (When OTP area data is programmed from 0 to 1 or verify error)	0	0	0	0	1	0	0	0
Configuration set error	An error has occurred during configuration set.	0	0	0	0	0	1	0	0
Protection error	The area for updating is permanently protected and the data for updating contains a TED error.	0	0	0	0	0	0	1	0

Table 65.42 Error Protection Types (2/2)

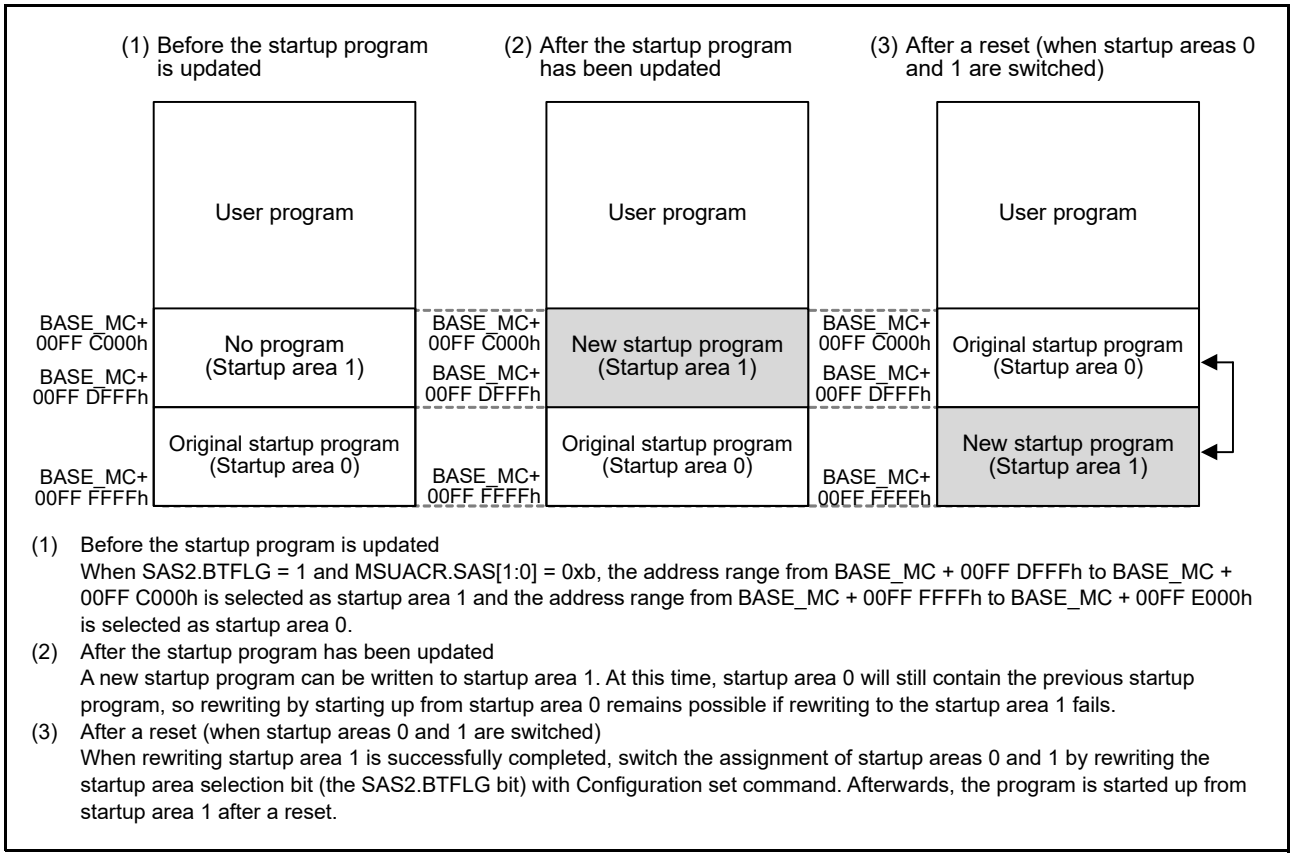
Error Type	Description	ILGCO MERR	OTERR	PZFERR	ILGLERR	PRGERR	CFGPRGERR	PROTERR	MREAE
Access error	Program command has been issued to reserved area in extra MRAM program mode.	0/1	0	0	1	0	0	0	1
	Configuration set command has been issued to a reserved configuration area in extra MRAM program mode.	0/1	0	0	1	0	0	0	1
	Program command or configuration set command has been issued to Protect Zone protection area in extra MRAM program mode.	0/1	0	1	1	0	0	0	1
	Configuration set command has been issued to operation mode protection area in extra MRAM program mode.	0/1	0	0	1	0	0	0	1
Others	The MACI command-issuing area has been accessed in extra MRAM read mode.	0	1	0	1	0	0	0	0
	The MACI command-issuing area has been read in extra MRAM program mode.	0	1	0	1	0	0	0	0

65.9.3 Startup Program Protection

This is a function for protecting the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.
The startup area is 8 Kbytes, 16 Kbytes, or 32 Kbytes in size and is assigned to the user area in the code MRAM. The size of the startup area can be changed by the BTSIZE bits. When resizing the startup area, be careful not to destroy the software in use.
This function uses the values of the SAS2.BTFLG bit, the SAS.BTSIZE bits, and the MSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Table 65.25 to Table 65.30).

In protection of the startup program, the selection of the startup area can be fixed by bit 21 of the MOFSPS0_S register and the size of the startup area can be fixed by bit 20 of the register. However, the POFSPS.POFSPS[21:20] bits are not restored to 1 once the flag is set to 0. Exercise extra caution when handling the POFSPS.POFSPS[21:20] bits.

Assign all startup areas (a total of 64 Kbytes of areas 0 and 1) to the secure area regardless of the BTSIZE bit setting.



(1) Before the startup program is updated
When SAS2.BTFLG = 1 and MSUACR.SAS[1:0] = 0xb, the address range from BASE_MC + 00FF DFFFh to BASE_MC + 00FF C000h is selected as startup area 1 and the address range from BASE_MC + 00FF FFFFh to BASE_MC + 00FF E000h is selected as startup area 0.

(2) After the startup program has been updated
A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.

(3) After a reset (when startup areas 0 and 1 are switched)
When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the SAS2.BTFLG bit) with Configuration set command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 65.34 Concept of Protection of the Startup Program

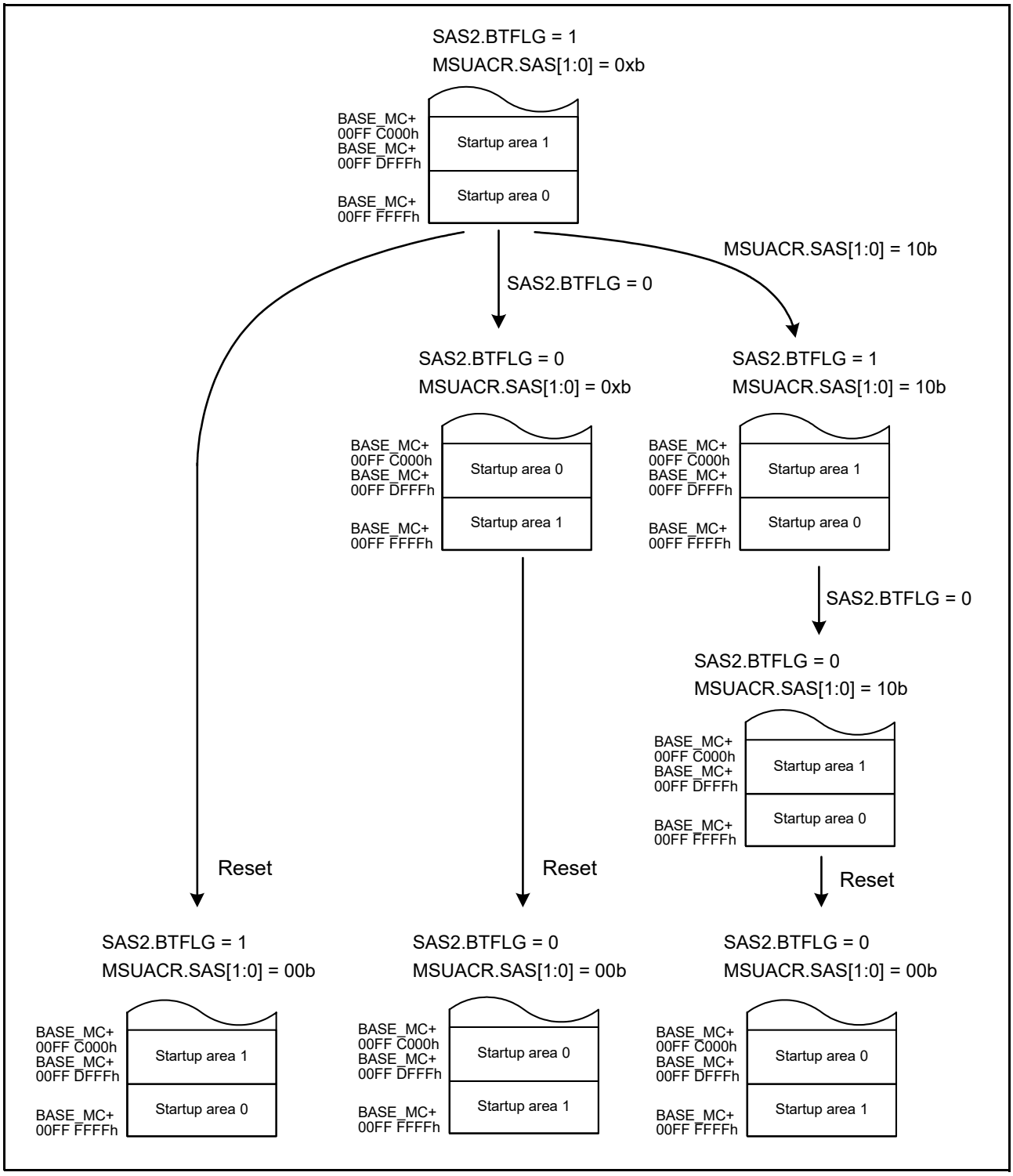


Figure 65.35 Example 1 of Transitions for Startup Program Protection Settings

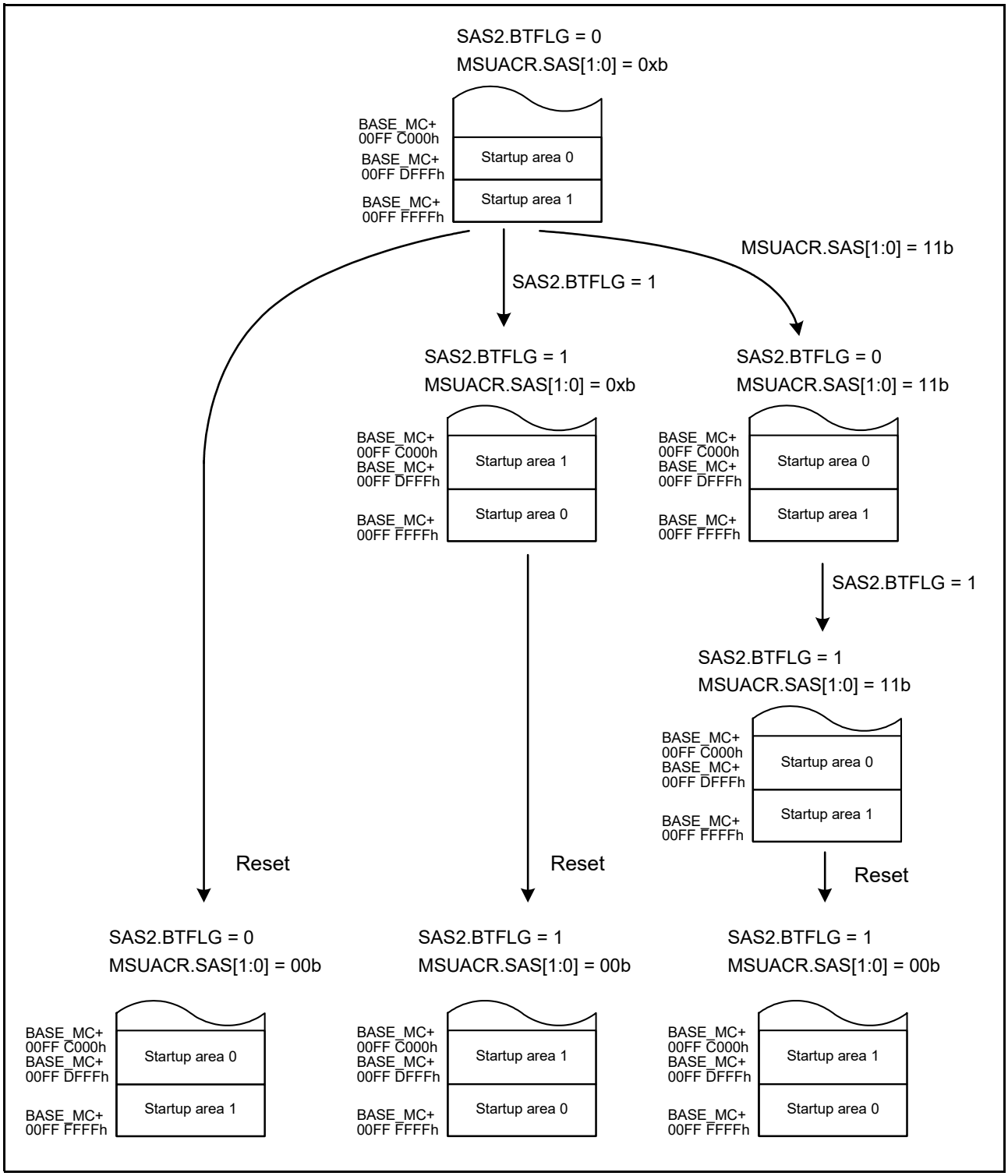


Figure 65.36 Example 2 of Transitions for Startup Program Protection Settings

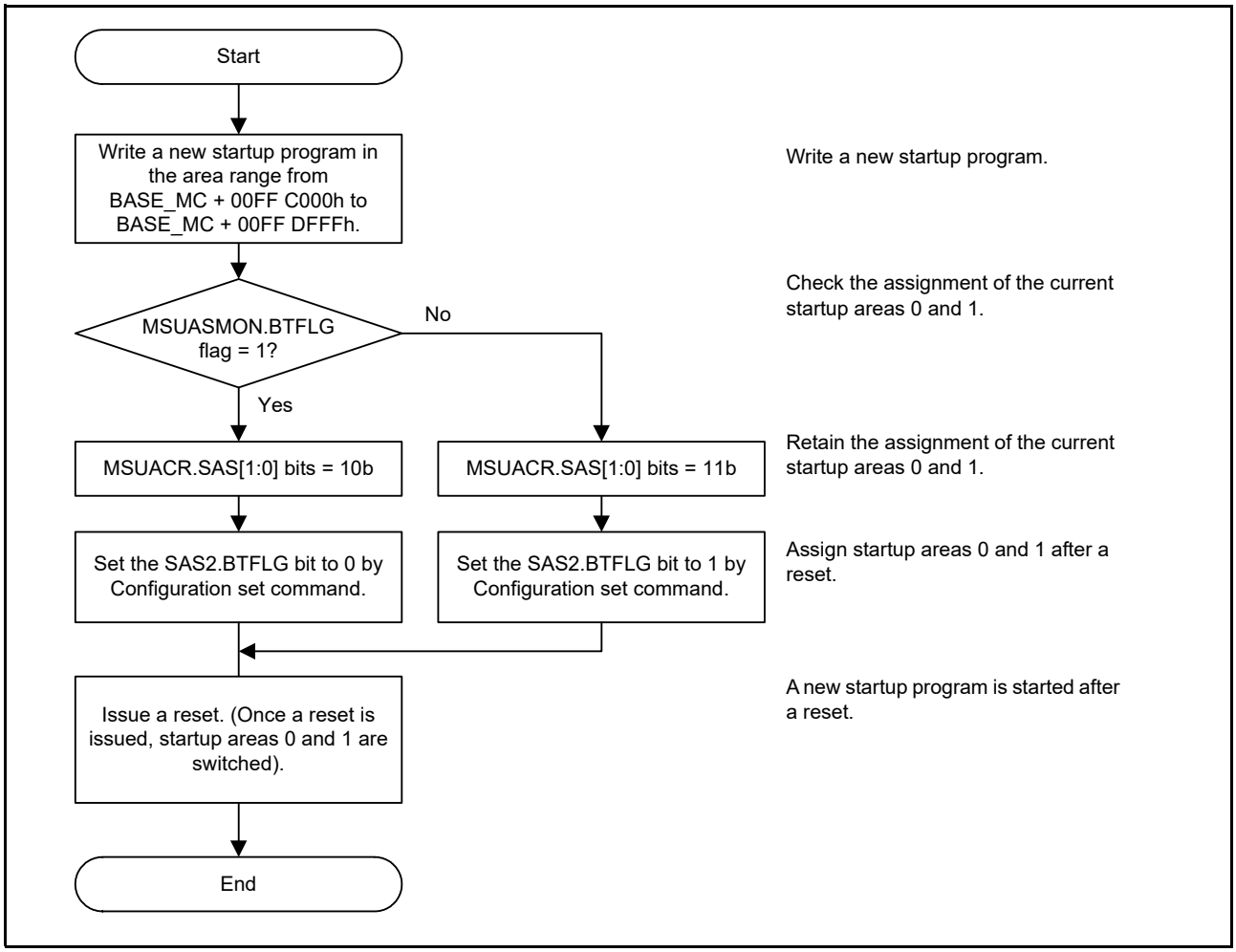


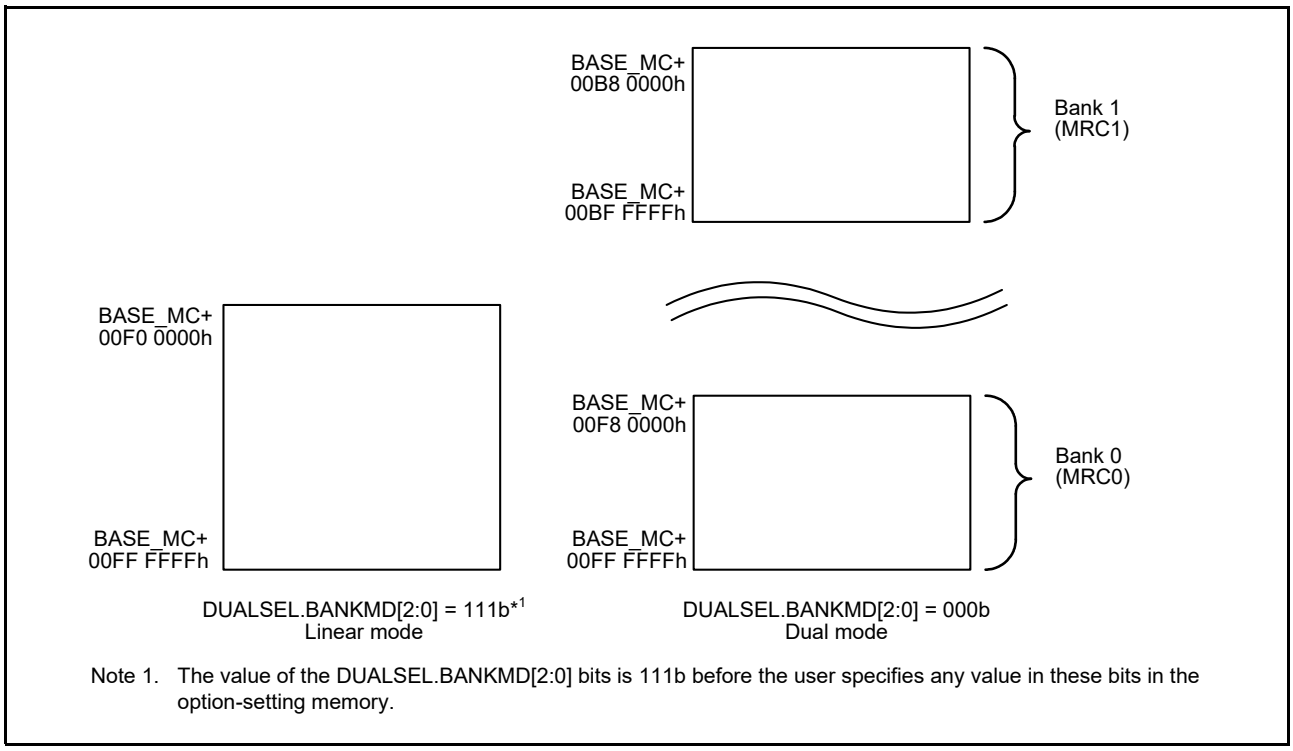
Figure 65.37 Concept of Protection of the Startup Program

65.9.4 Dual Bank Function

The bank mode switching function and startup bank selection function can be used to update the user program while execution of the program is in progress.
These functions provide a way to safely update the program even if rewriting is suspended by a reset.

65.9.4.1 Bank Mode Switching Function

This function can be used to switch operations between two modes; linear mode, in which the user area in the code MRAM is handled as a single area, and dual mode, in which it is handled as two bank areas. Figure 65.38 shows the bank mode switching function. Setting the DUALSEL.BANKMD[2:0] bits in the option-setting memory and then applying a reset switches the bank mode. Selecting dual mode enables the startup bank selection function.
Note that the startup program protect function cannot be used when dual mode is selected (DUALSEL.BANKMD[2:0] = 000b).

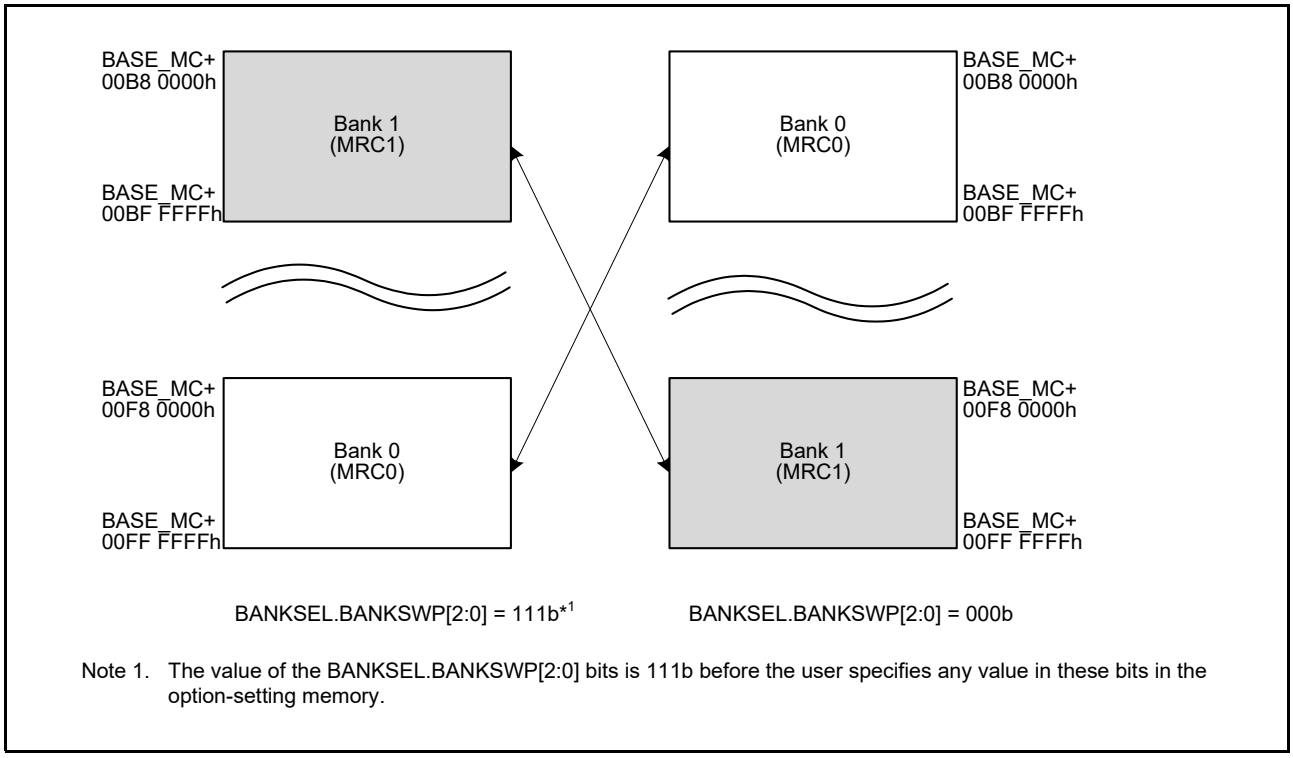


Note 1. The value of the DUALSEL.BANKMD[2:0] bits is 111b before the user specifies any value in these bits in the option-setting memory.

Figure 65.38 Example of Bank Mode Switching (Example in the Product with 1-Mbyte Code MRAM)

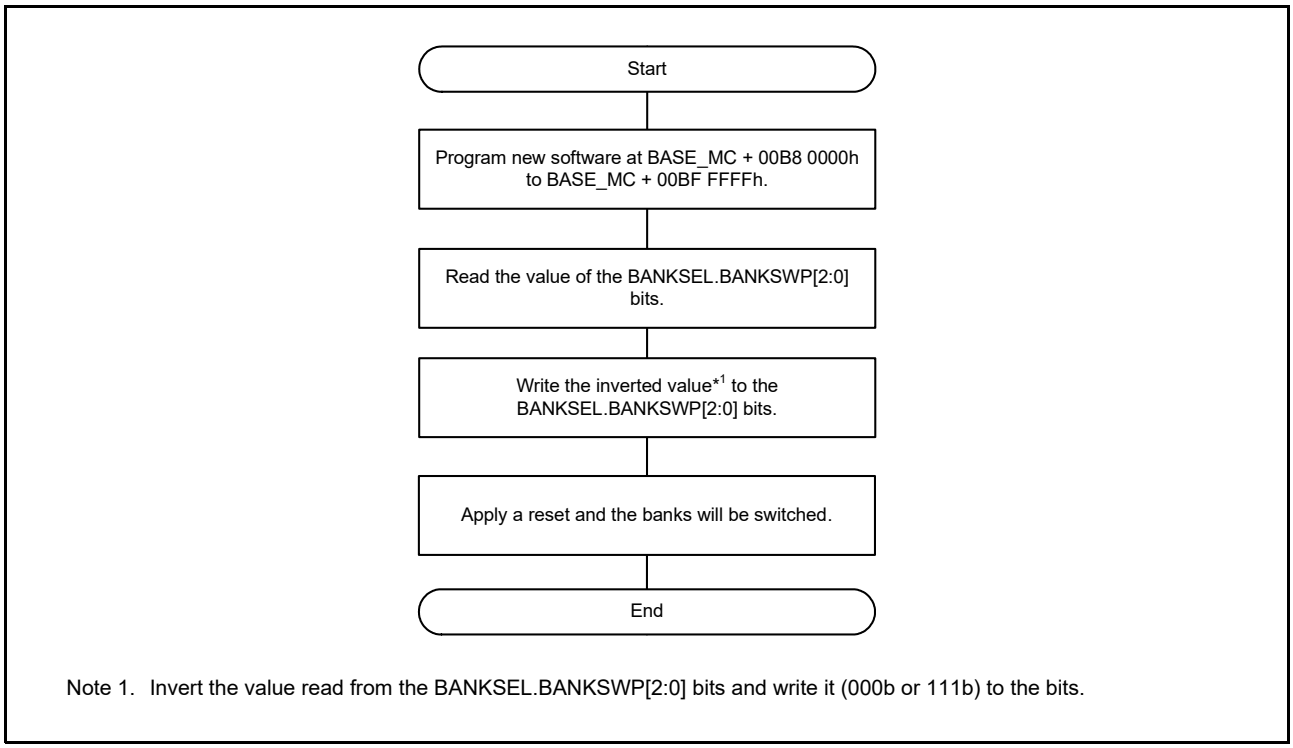
65.9.4.2 Startup Bank Selection Function

This function can be used to select the bank area where program execution is started when dual mode is selected (DUALSEL.BANKMD[2:0] = 000b). This provides a way to safely update the program even if rewriting is suspended by a reset. Figure 65.39 shows the startup bank selection function and Figure 65.40 shows an example of startup bank selection flow. Setting the BANKSEL.BANKSWP[2:0] bits in the option-setting memory and then applying a reset switches the addresses between banks 0 and 1 and starts program execution from the updated area. This function is disabled when linear mode is selected.



Note 1. The value of the BANKSEL.BANKSWP[2:0] bits is 111b before the user specifies any value in these bits in the option-setting memory.

Figure 65.39 Example of Startup Bank Selection (Example in the Product with 1-Mbyte Code MRAM)



Note 1. Invert the value read from the BANKSEL.BANKSWP[2:0] bits and write it (000b or 111b) to the bits.

Figure 65.40 Example of Startup Bank Selection Flow (Example in the Product with 1-Mbyte Code MRAM)

65.9.5 Block Swap Function

This function can be used to update the user program while program execution is in progress. This provides a way to safely update the program even if rewriting is suspended by a reset.

A half or quarter of the code MRAM can be allocated as block swap areas. The total size of the block swap areas can be selected by the BLSEL.BLSWAREA bits. When resizing the block swap areas, be careful not to destroy the software in use.

Be sure to specify the same attribute (secure or non-secure area) to both block swap areas.

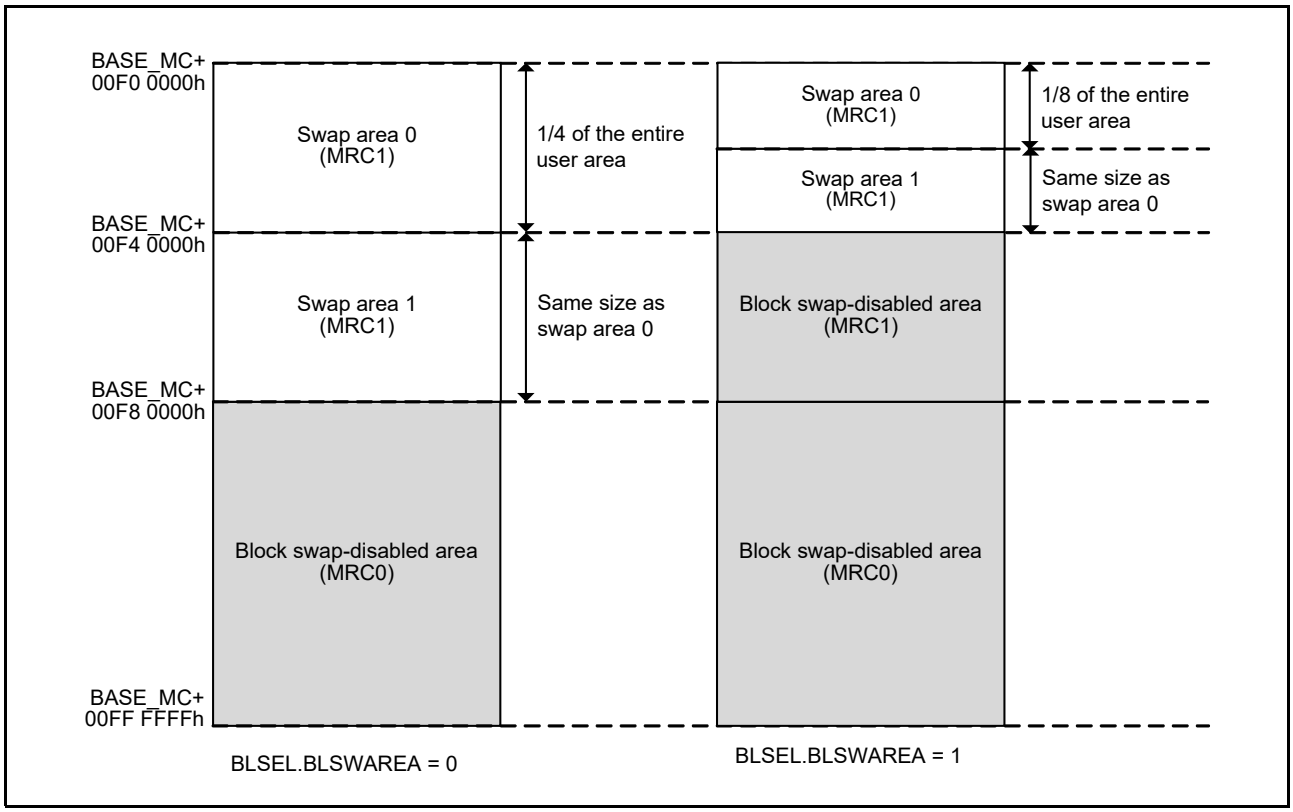


Figure 65.41 Sizes of the Source and Target Areas for Block Swap

65.9.5.1 Block Swap Selection

This function provides a way to safely update the program even if rewriting is suspended by a reset. Figure 65.42 and Figure 65.43 show examples of block swap selection and Figure 65.44 shows an example of block swap selection flow. Setting the BLKSWPn_S.SWPi[3:0] or BLKSWPn_NS.SWPi[3:0] bits in the option-setting memory to 0000b and then applying a reset swaps the addresses of the selected blocks between the source and target areas (swap areas 0 and 1). Note that the block swap selection function cannot be used when dual mode is selected (DUALSEL.BANKMD[2:0] = 000b).

For the correspondence between the BLKSWPn_S.SWPi[3:0] or BLKSWPn_NS.SWPi[3:0] bits and the target blocks (addresses) for block swap, refer to section 7.2.10, Block Swap Setting Register n for Secure Application (BLKSWPn_S) (n = 0 to 3), Block Swap Setting Register n for Non-Secure Application (BLKSWPn_NS) (n = 0 to 3).

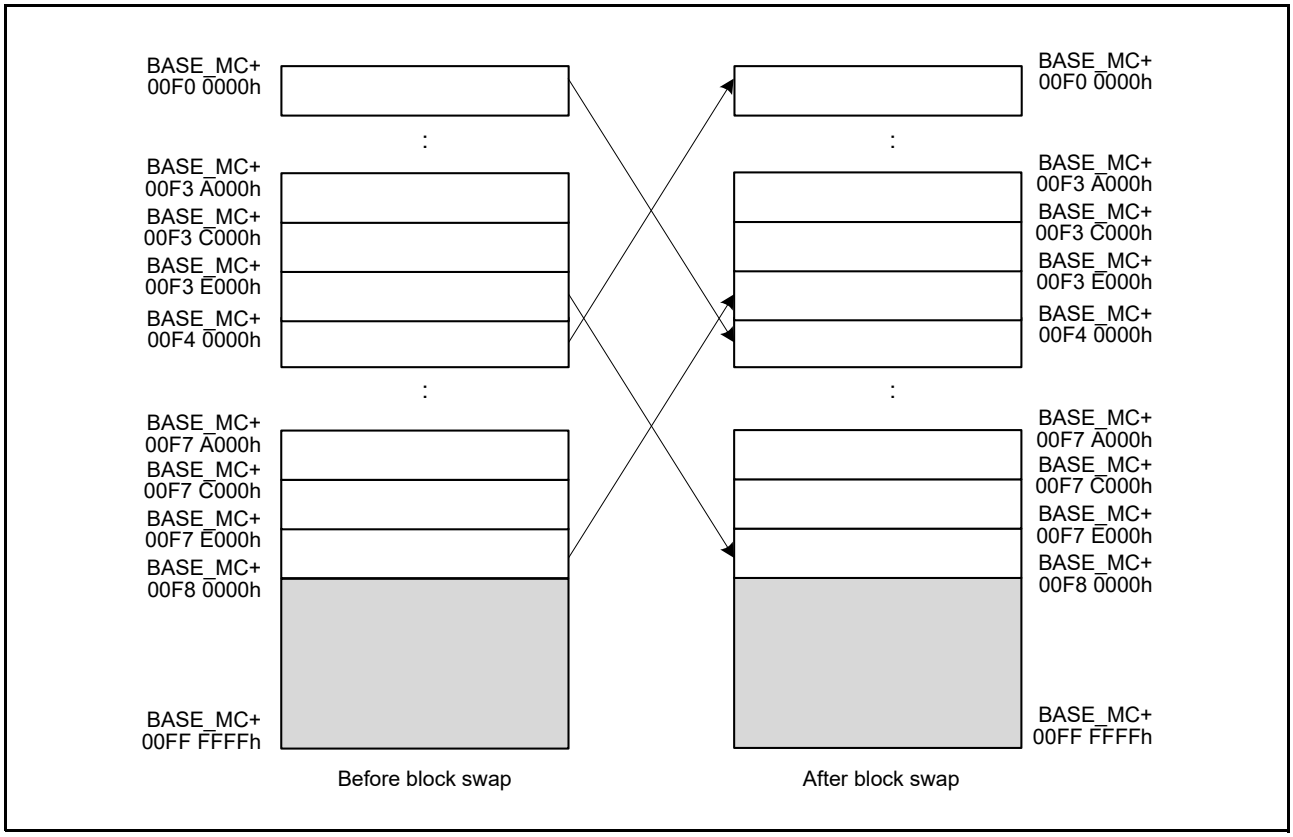


Figure 65.42 Example of Block Swap Selection when the Block Swap Areas are 1/2 (BLSEL.BLSWAREA = 0) of the Code MRAM (Example in the Product with 1-Mbyte Code MRAM)

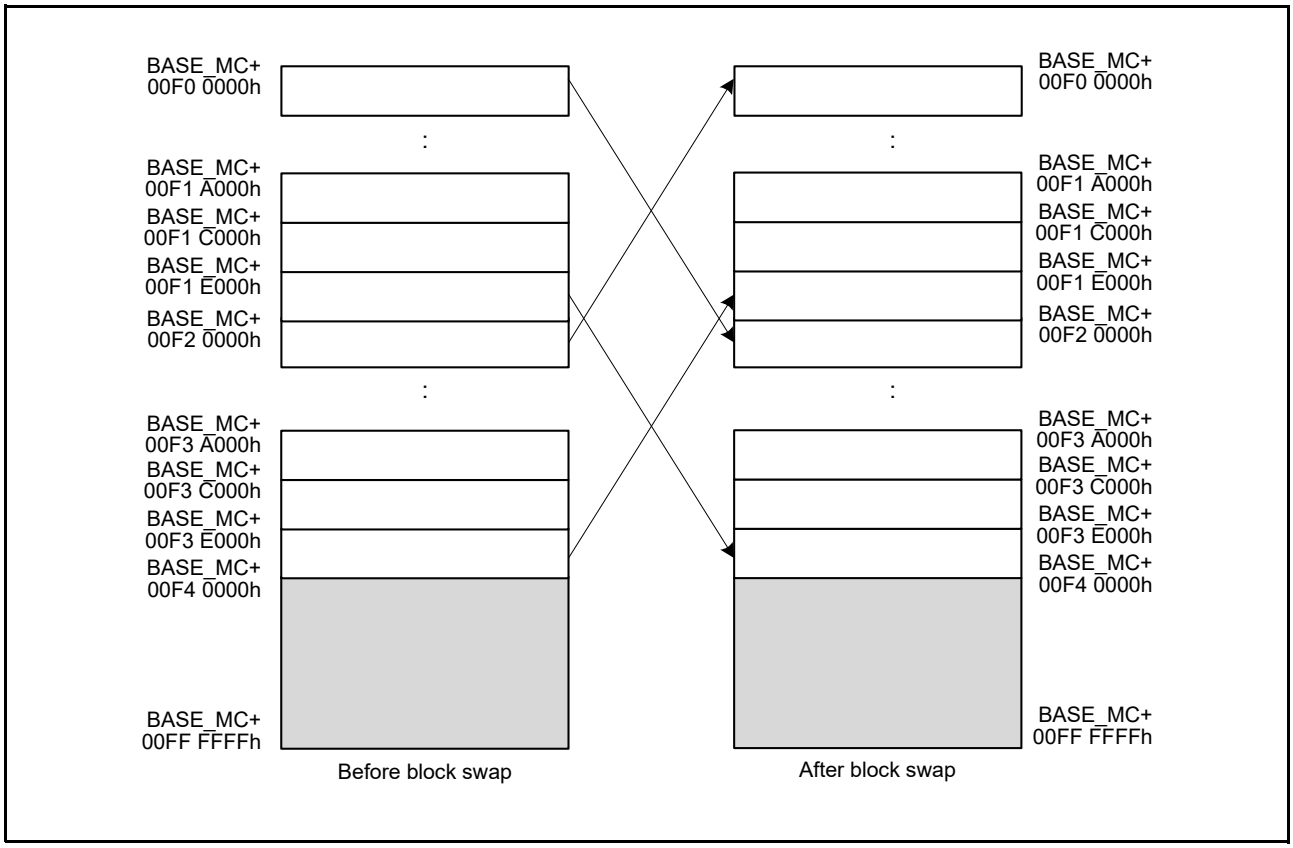


Figure 65.43 Example of Block Swap Selection when the Block Swap Areas are 1/4 (BLSEL.BLSWAREA = 1) of the Code MRAM (Example in the Product with 1-Mbyte Code MRAM)

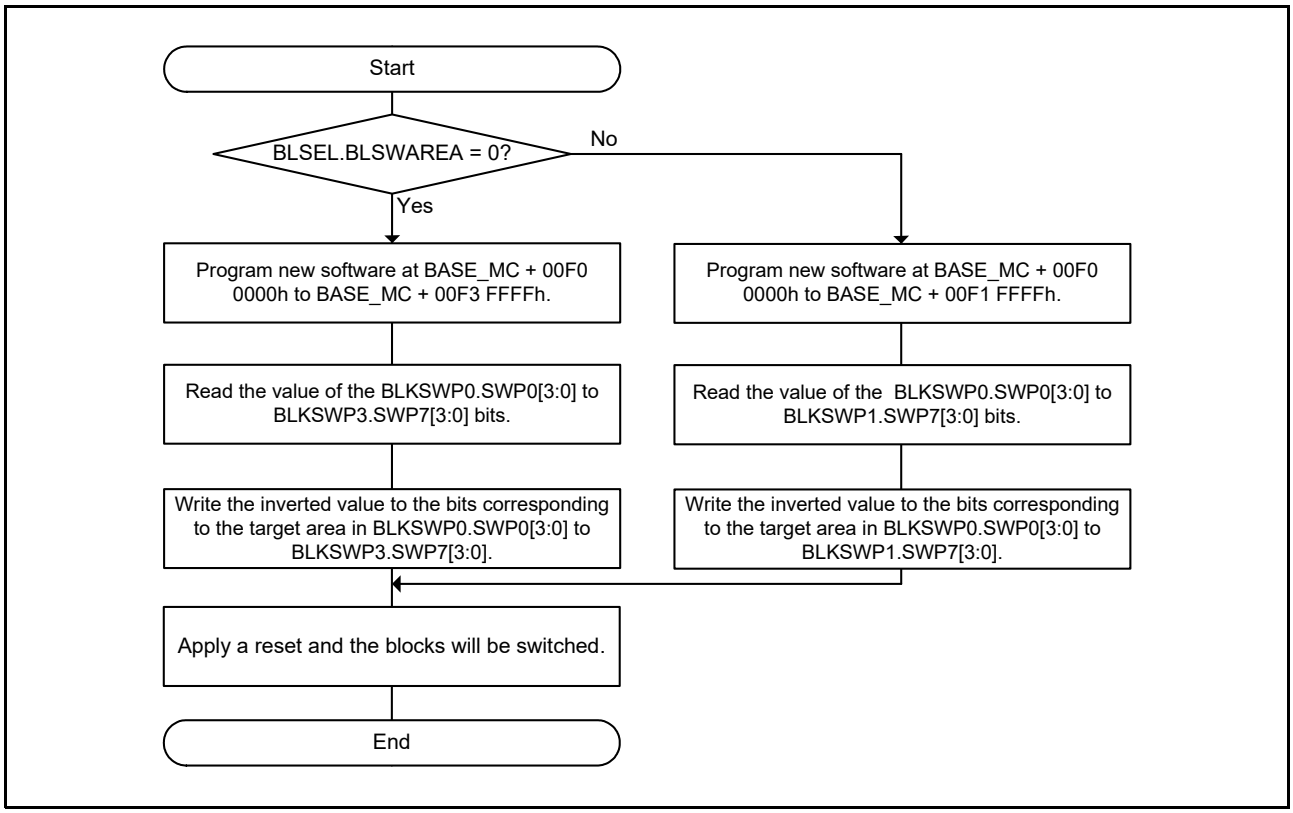


Figure 65.44 Example of Block Swap Selection Flow (Example in the Product with 1-Mbyte Code MRAM)

65.9.6 Fault Correction and Detection

MRAM systems are equipped with ECC circuitry to correct and detect errors. DEC-TED (double bit error correction and triple bit error detection) ECC corrects errors in 128-bit data units. Code MRAM uses two ECC circuits for 256 bit readout, and extra MRAM uses one ECC circuit for 128 bit readout.

If there is a DEC error in the code MRAM reading, error correction is performed and the MRCRAES.DECERRC flag is set to 1. If there is a TED error, the MRCRAES.TEDERRC flag is set to 1, but error correction is not possible. When an error is detected, the error address is stored in the MRCRDEA register for DEC error and in the MRCRTEA register for TED error.

If a DEC error occurs, read the data at the error address, and write back. If a TED error occurs, update all 32 bytes of data and overwrite with the correct data.

If there is a DEC error in the extra MRAM reading, error correction is performed and the MRERAES.DECERRE flag is set to 1. If there is a TED error, the MRERAES.TEDERRE flag is set to 1, but error correction is not possible. When an error is detected, the error address is stored in the MRERDEA register for DEC errors and in the MRERTEA register for TED errors. If a DEC error occurs, read the data at the error address, and write back. If a TED error occurs, update all 32 bytes of data and overwrite with the correct data.

65.9.6.1 Test Method for ECC Circuits

The ECC circuit can be tested by setting the program data and ECC data to desired values. A block diagram of the ECC test circuit is shown in Figure 65.45.

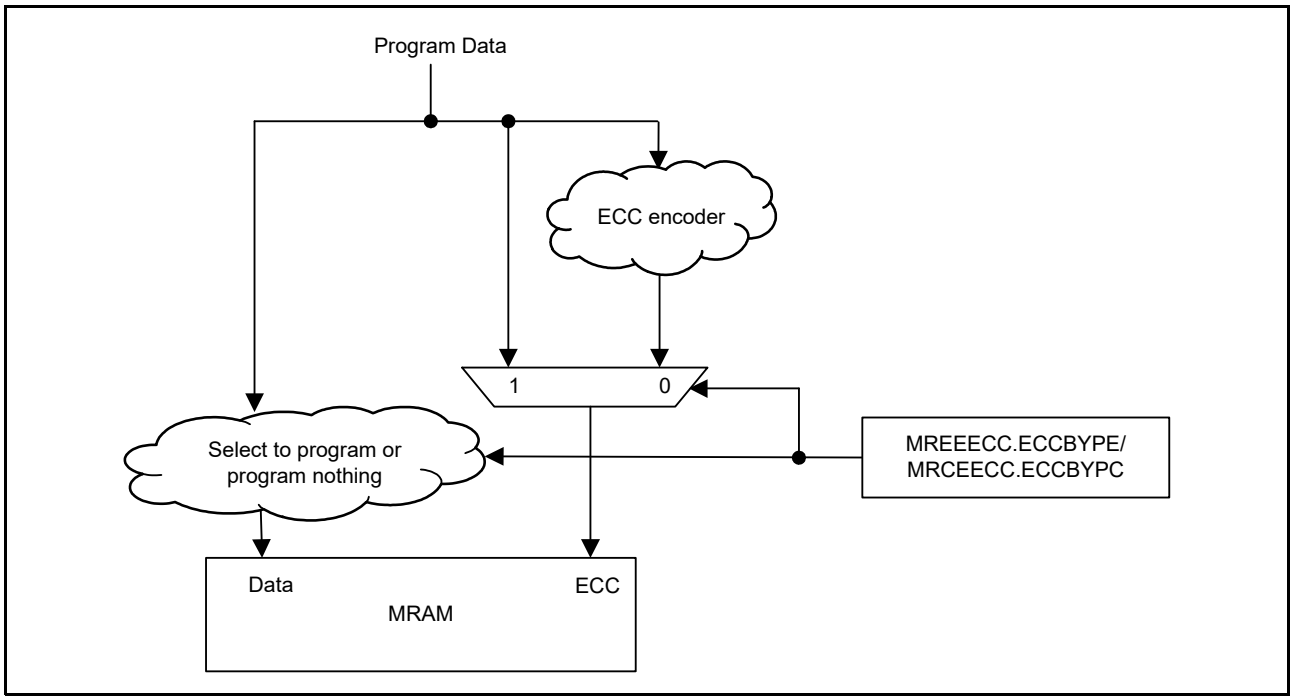


Figure 65.45 Block Diagram of the ECC Test Circuit

Table 65.43 Changes in Data Allocation Depending on the Value of MRCEECC.ECCBYPC

	Code MRAM	
MRCEECC.ECCBYPC	**Data[255:0]**	**ECC[33:0]**
0	Program data[255:0]	ECC code generated from program data[127:0]
1	Program nothing	{Program data[144:128], Program data[16:0]}

Table 65.44 Changes in Data Allocation Depending on the Value of MREEECC.ECCBYPE

	Extra MRAM	
MREEECC.ECCBYPE	**Data[127:0]**	**ECC[16:0]**
0	Program data[127:0]	ECC code generated from program data[127:0]
1	Program nothing	Program data[16:0]

Figure 65.46 shows the ECC decoder test flow for the code MRAM and Figure 65.47 shows the ECC decoder test flow for the data MRAM. The code MRAM has two ECC decoders and ECC errors are merged, so when testing the ECC decoder, test the ECC decoder circuits individually.

Note 1. To eliminate ECC errors in instruction fetching.
Note 2. If you get an error, try the test again at a different address.
Note 3. Invert the bits of the code according to the error condition you want to check (1 to 3)

Replace the bits shown in Table 65.43, Table 65.44with the code you created and write to the same address as DATA.

Note 4. See Table 65.45, Table 65.46 for how to check the result.

Table 65.45 Expected Value of Code MRAM Error Status

	Expected Value		
Check Results	**1-bit Invert**	**2-bit Invert**	**3-bit Invert**
MRCRAES.DECERRC	0	1	0
MRCRAES.TEDERRC	0	0	1

Table 65.46 Expected Value of Data MRAM Error Status

	Expected Value		
Check Results	**1-bit Invert**	**2-bit Invert**	**3-bit Invert**
MRERAES.DECERRE	0	1	0
MRERAES.TEDERRE	0	0	1

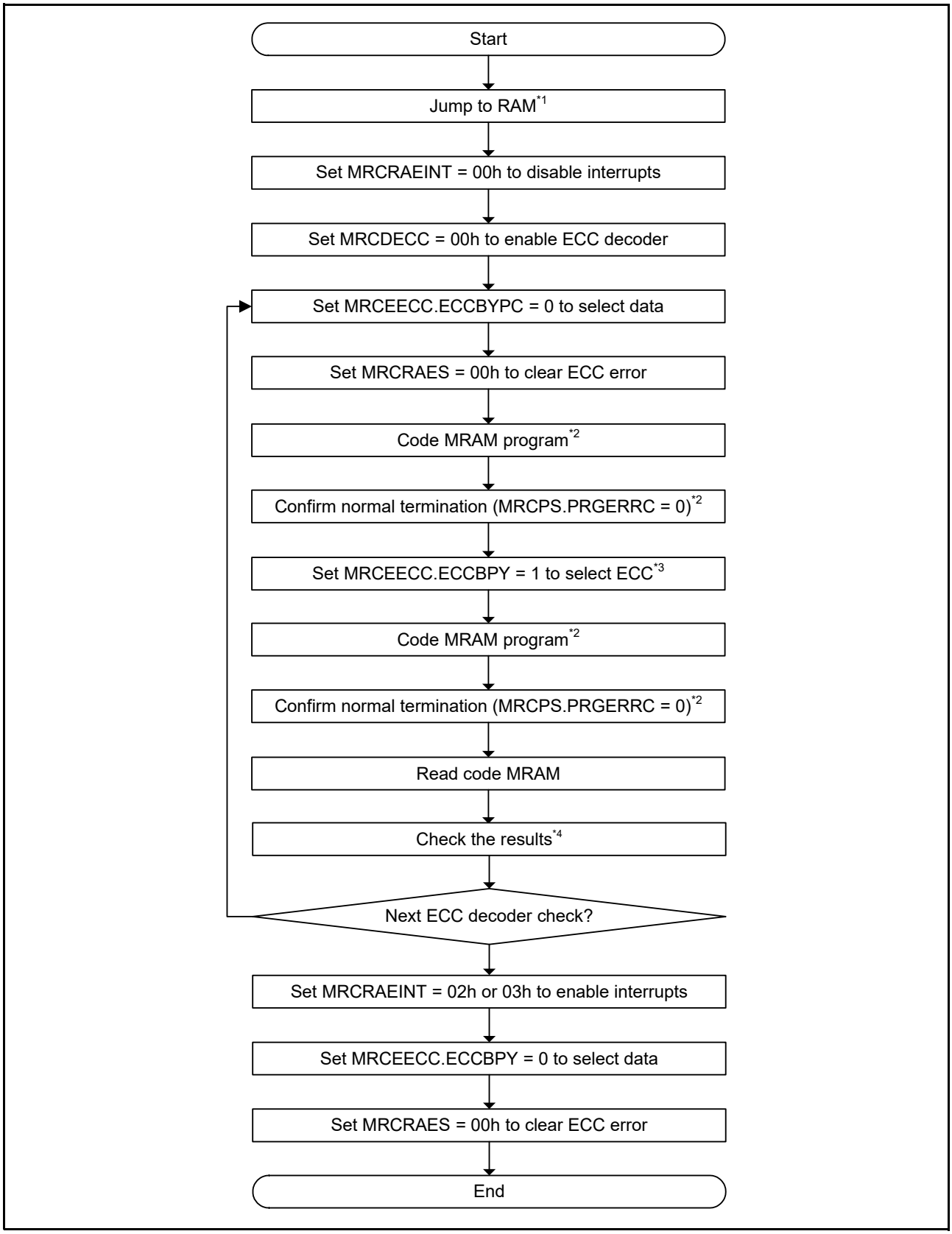


Figure 65.46 Code MRAM ECC Decoder Test Flow

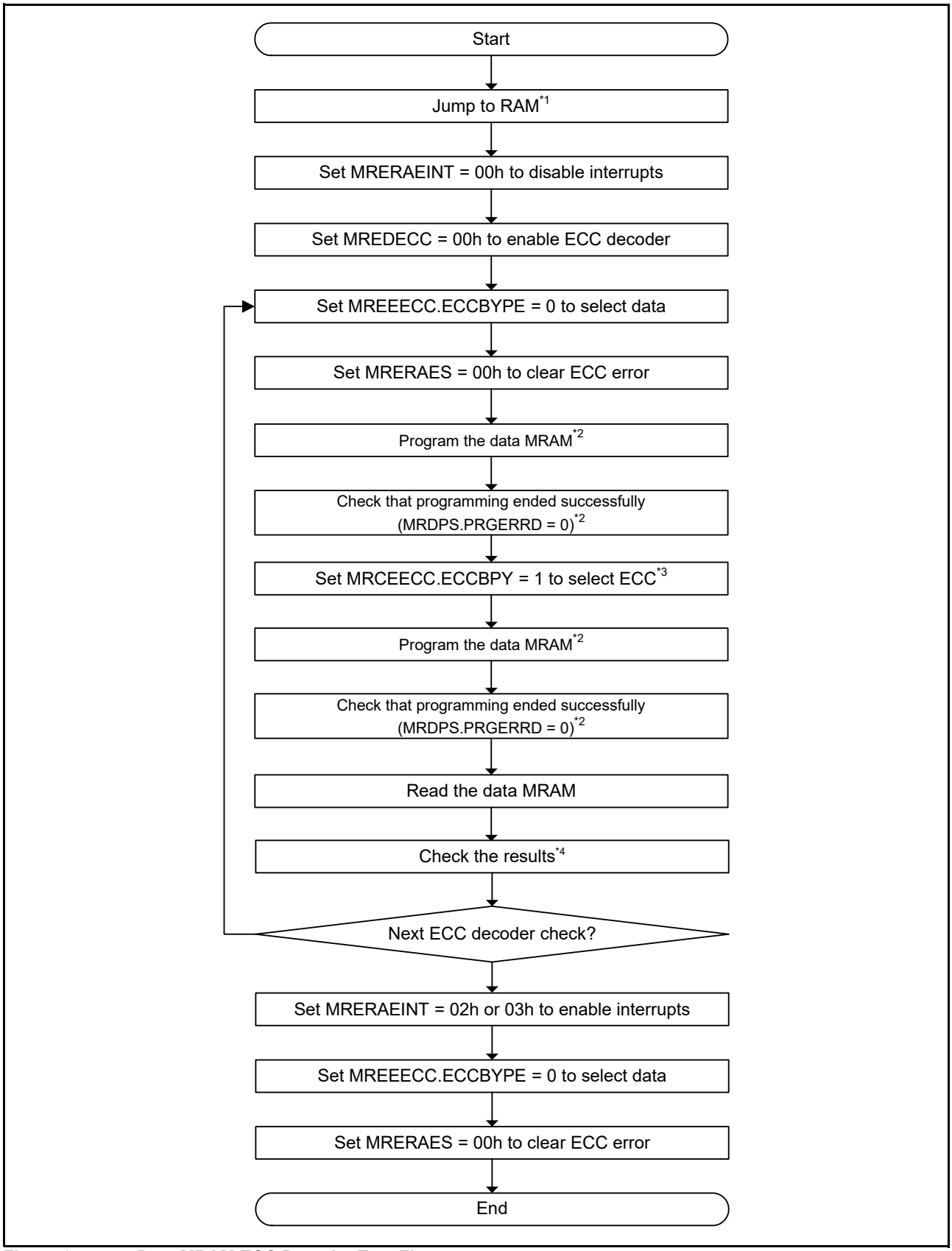


Figure 65.47 Data MRAM ECC Decoder Test Flow

65.10 Configuration Update Transfer

After the program or configuration set command operation, the system status can be updated by configuration update transfer without system reset. When 1 is written to the MCTRCNTR.TRTRG bit in the read mode (the MENTRYR register is 0000h and the WHUKEXE flag in MREZS register is 0), MACI transfers data from extra MRAM. The MCTRSTATR register can monitor to configuration update transfer processing status. Extra MRAM cannot be read during configuration data transfer.

Figure 65.48 shows configuration update transfer usage.

If the WHUKEXE flag in the MREZS register is set to 1 in the configuration update transfer flow, the configuration update transfer must be re-executed after the W-HUK zeroing is finished.

See section 65.5.23, MRAM Configuration Update Transfer Control Register (MCTRCNTR), and section 65.5.24, MRAM Configuration Update Transfer Status Register (MCTRSTATR) in details of the associated registers specifications (the MCTRCNTR and MCTRSTATR registers).

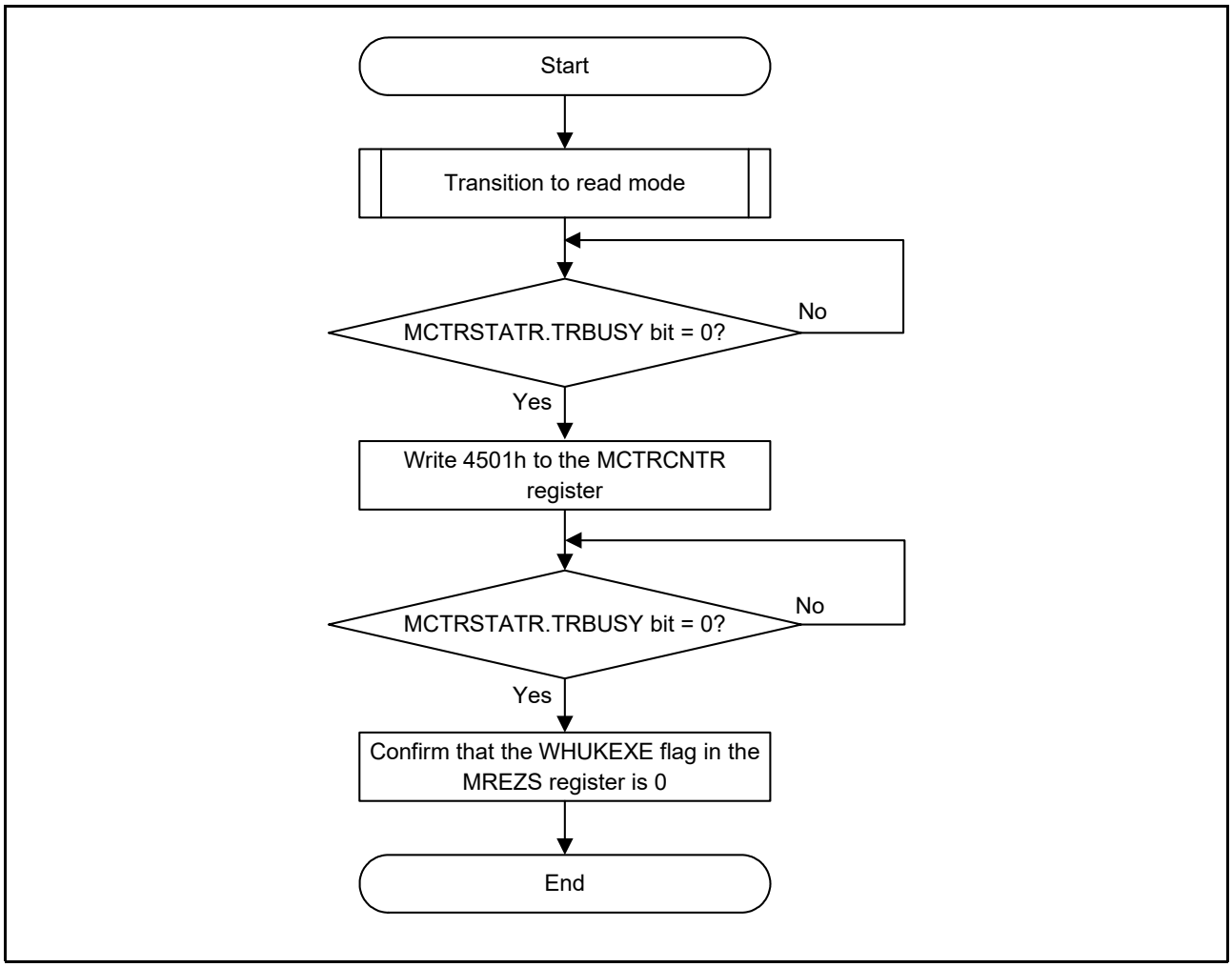


Figure 65.48 Configuration Update Transfer Usage

65.11 Boot Mode

There are four types of boot mode; boot mode (SCI interface), boot mode (USB interface), boot mode (JTAG/SWD interface), and boot mode (FINE interface).
Table 65.47 lists the I/O pins used in boot mode. Table 65.48 lists the available communication interface used in the boot mode.

Table 65.47 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD/FINED	Input	Boot mode (SCI interface) Boot mode (USB interface)	Selection of operating mode
P303/RXD1	Input	Boot mode (SCI interface)	For host communication (to receive data through SCI)
P302/TXD1	Output		For host communication (to transmit data through SCI)
USB0_DP, USB0_DM	I/O	Boot mode (USB interface)	Data input/output of USB
P101/USB0_VBUS	Input		Detection of connection and disconnection of USB cables
P301/TCK [JTAG]	Input	Boot mode (JTAG/SWD interface)	On-chip emulator or boundary scan pins.
P300/TMS [JTAG]	Input		
P302/TDO [JTAG]	Output		
P303/TDI [JTAG]	Input		
P301/SWCLK [SWD]	Input		Serial wire clock
P300/SWDIO [SWD]	I/O		Serial wire debug data input/output
MD/FINED	I/O	Boot mode (FINE interface)	FINE data input/output

Table 65.48 Available Communication Interface Used in Boot Mode

Main clock oscillator or external clock is connected	Yes	No	No
Sub clock oscillator is connected*1	Yes or No	Yes	No
Available interface	SCI or USB	SCI or USB	SCI
Tool connection time*2	Up to 1 second	Up to 2 seconds	Up to 3 seconds

Note 1. The drive capability of the sub clock oscillator is set to standard by SOMCR.SODRV bit. Note that if you use the crystal corresponding the low drive capability on your board, the crystal may not oscillate in the boot mode.
Note 2. See the boot firmware manual for the detail connection time.

65.11.1 Boot Mode (SCI Interface)

In boot mode (SCI interface), the host sends control commands and data for programming, and the MRAM is programmed accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.
When this MCU is activated in boot mode (SCI interface), the program on the dedicated area the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming by receiving control commands from the host.
Figure 65.49 shows the system configuration for operations in boot mode (SCI interface).

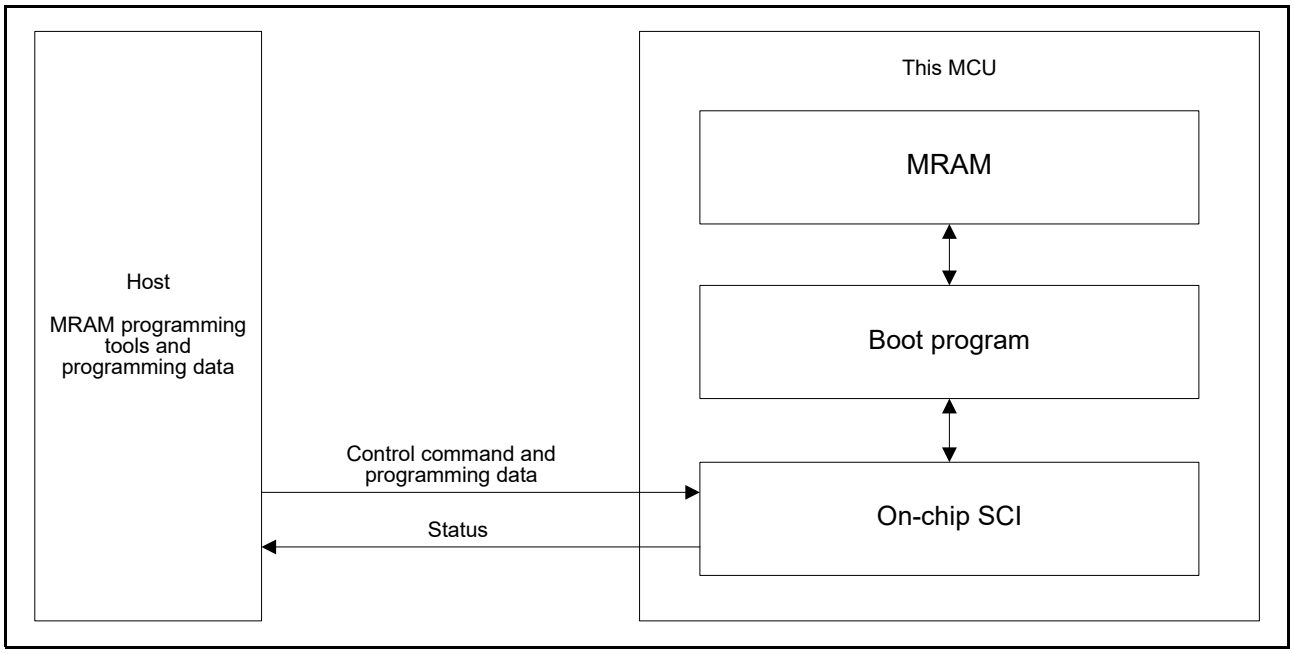


Figure 65.49 System Configuration for Operations in Boot Mode (SCI Interface)

65.11.2 Boot Mode (USB Interface)

In boot mode (USB interface), the MRAM can be programmed by sending control commands and program data from the host. An on-chip USB is used for communications between the host and this MCU. The host requires tools for sending control commands and data for programming. Figure 65.50 shows the configuration of a system for use in boot mode (USB interface). The USB cable must be connected on reset release.

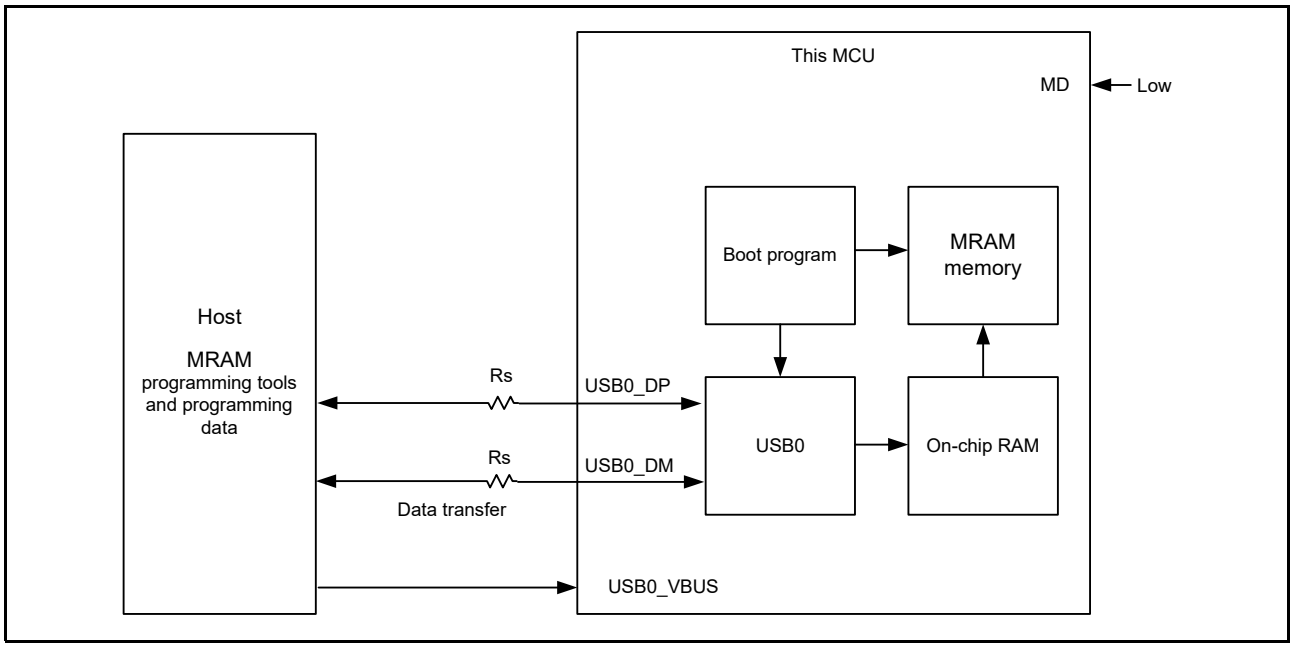


Figure 65.50 System Configuration in Boot Mode (USB Interface)

65.11.3 Boot Mode (FINE Interface)

The MRAM can be programmed using the FINE in boot mode (FINE interface). The MRAM can be rewritten.

65.11.3.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).
Figure 65.51 shows an Example of Pin Connections in Boot Mode (FINE Interface). Table 65.49 lists Pin Handling in Boot Mode (FINE Interface).
The example of pin connections shown in Figure 65.51 is a simplified circuit. Operations are not guaranteed in all systems.

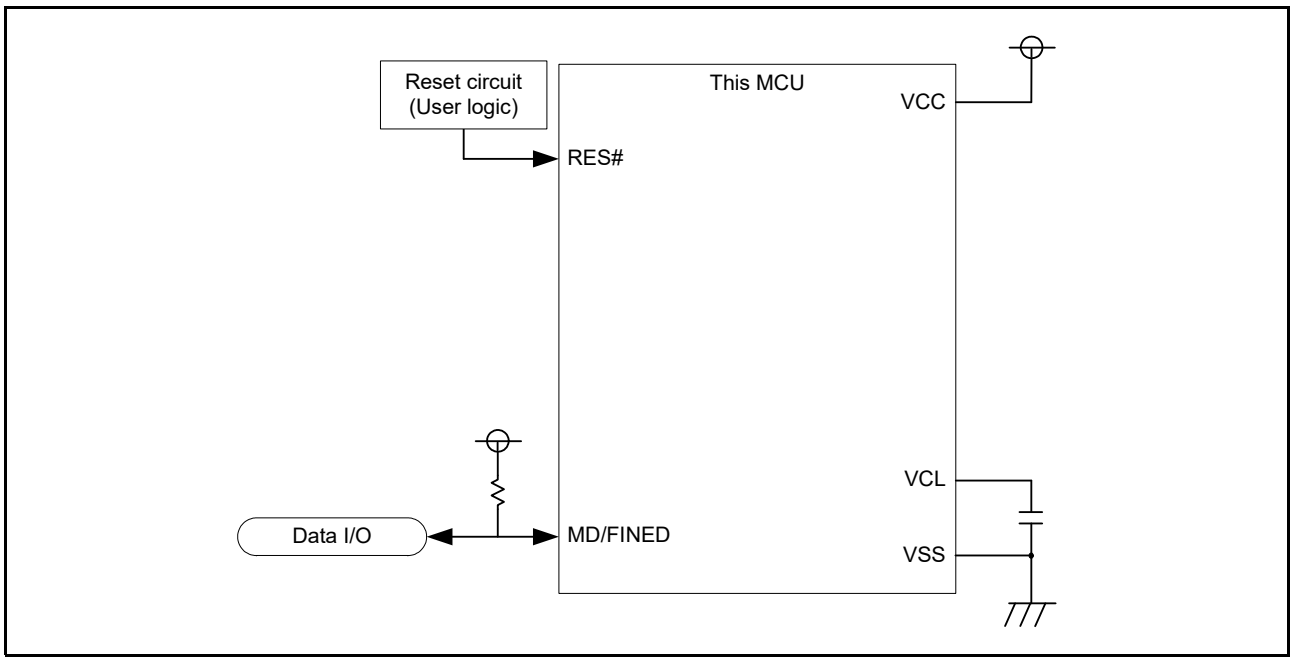


Figure 65.51 Example of Pin Connections in Boot Mode (FINE Interface)

Table 65.49 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input 1.62 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a 0.22-μF multilayer ceramic capacitor for stabilizing the internal voltage.
MD	Operating mode control/ data I/O	I/O	Connect the VCC pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

65.12 Using the Serial Programmer for Rewriting

A serial programmer can be used to rewrite MRAM in boot mode.

(1) Serial Programming

This MCU is mounted on the system board at the time of serial programming. Providing a connector to the board enables rewriting of this MCU by the serial programmer to proceed.

65.12.1 Environments for Serial Programming

The recommended environments for rewriting the MRAM of the MCU with data are described below.

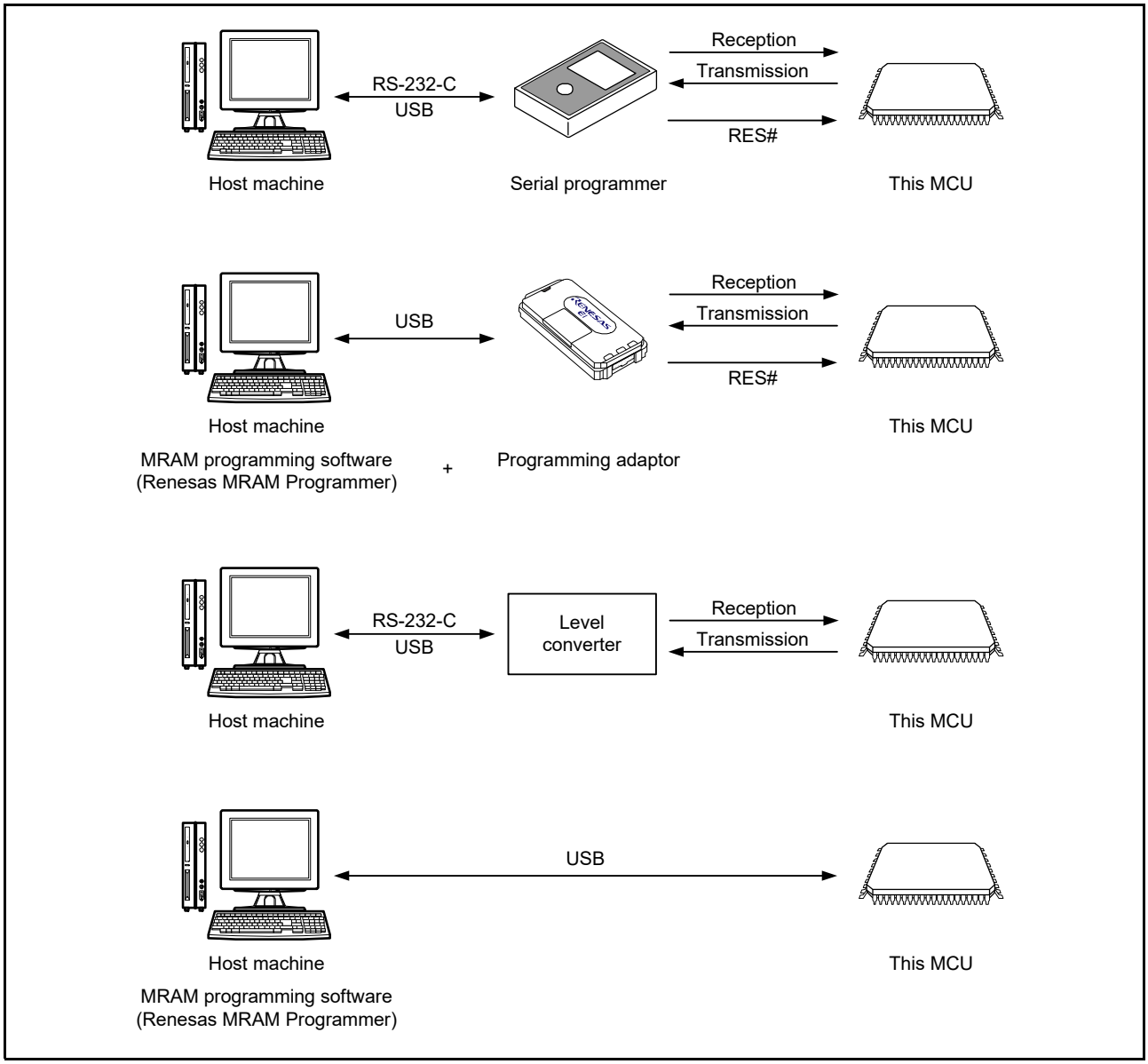


Figure 65.52 Environments for Rewriting the MRAM

65.13 Programming through Self-Programming

65.13.1 Overview

This MCU supports programming of the MRAM by the user program itself. The write page buffer or the MACI commands can be used with user programs for writing to the MRAM. This allows upgrading of user programs and rewriting of constant data fields.
The program for rewriting must be transferred to the internal RAM or external memory in advance when the BGO is not available or when rewriting the option-setting memory.

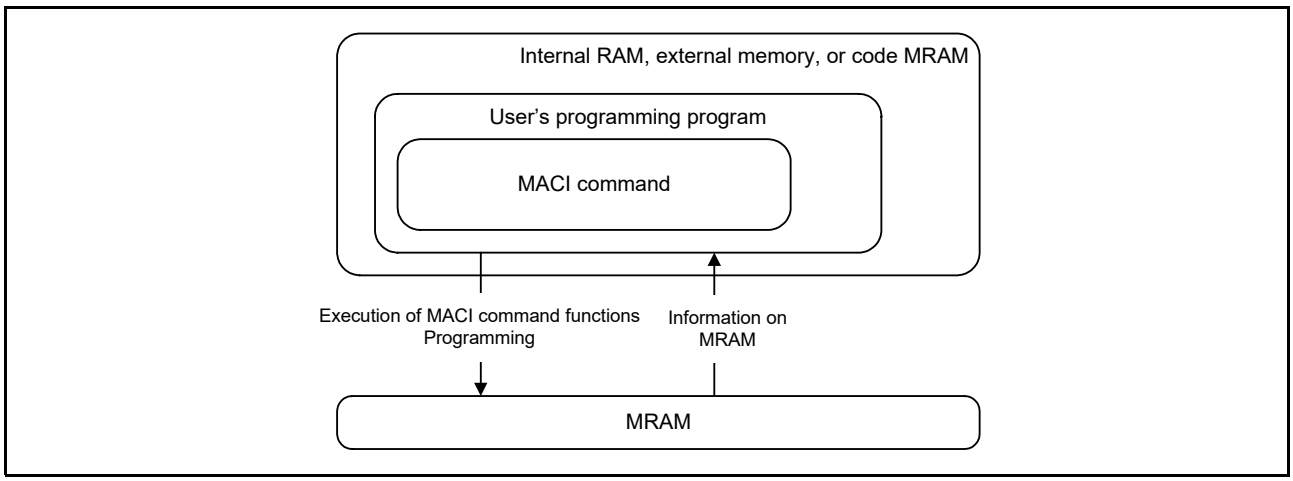


Figure 65.53 Schematic View of Self-Programming

For comprehensive information on the self-programming, see section 65.7, MACI Commands.

65.13.2 Parallel Accessibility

Parallel read and program accesses are possible in conditions shown in Table 65.50.
Read while program (background operation: BGO) is possible between different MRAM banks. Parallel read to different MRAM banks is possible. Only one MRAM bank can be programmed at the same time. W-HUK zeroing has highest priority among all MRAM programming requests. (See section 65.8.6, HUK Protection).
Programming of the code MRAM and extra MRAM cannot be done in parallel. A programming request issued later is kept waiting until previous programming is completed. If a request for programming of the code MRAM is issued while the MSTATR.MRDY flag is 0, programming begins after the flag becomes 1. If a request for programming of the extra MRAM (a program, configuration set, or increment counter command) is issued while the MRCPS.PRGBSYC flag is 1, programming begins after the flag becomes 0. Therefore, the programming time in these cases is the sum of the times of code MRAM programming and extra MRAM programming.
Reading from and programming of a given MRAM area cannot be done in parallel. The values shown in section 67.15, MRAM Characteristics, are the individual operation times in which no arbitration is performed between reading and programming operations.

Table 65.50 Parallel Accessibility

		Code MRAM Bank 0 (MRC0)		Code MRAM Bank 1 (MRC1)		Extra MRAM		
		Read	Program	Read	Program	Read	Program (Data MRAM)	MACI Command
Code MRAM Bank 0 (MRC0)	Read	X	X	O	X	O*1	O*1	O
	Program	X	X	O	X	O*1	X	X
Code MRAM Bank 1 (MRC1)	Read	O	X	X	X	O*1	O*1	O
	Program	O	X	X	X	O*1	X	X
Extra MRAM	Read	O*1	O*1	O*1	O*1	X	X	X
	Program (Data MRAM)	O*1	X	O*1	X	X	X	O*2
	MACI Command	O	X	O	X	X	X*2	X

O: Executable
X: Not executable
Note 1. The bus master cannot read or program the extra MRAM during configuration update transfer (see section 65.10, Configuration Update Transfer).
Note 2. When the MENTRYR.MENTRY bit is set to 1, a slave bus error always occurs and programming of the data MRAM fails.

65.13.3 Reading MRAM

Special settings are not required to read MRAM after release from the reset state. Data can simply be read out through access to addresses in the MRAM.
When reading MRAM that has not yet been programming (i.e. that is in the nonprogrammed state), all bits are read as 1.

65.14 Usage Notes

65.14.1 Area where Programming was Interrupted by W-HUK Zeroing

The data stored in the area where programming was suspended due to zeroing of the W-HUK is undefined. To prevent malfunctions due to reading undefined data, care should be taken not to fetch instructions or read data from the area where writing has been suspended.

65.14.2 Actions Prohibited during Programming

Do not reduce the power to less than operating voltage.

65.14.3 Prohibition of Additional Programming

An OTP area that uses ECC cannot be programmed twice with different data.

65.14.4 Resets During Programming

For a reset due to the signal on the RES# pin during programming of the MRAM, wait for at least t_{RESW} (see section 67, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

65.14.5 Allocation of Vectors for Interrupts and Other Exceptions During Programming

Generation of an interrupt or other exception during programming may lead to fetching of the vector from the code MRAM.
The vectors cannot be read from code MRAM until the program is finished. Code MRAM requires up t_{PMC} (see section 67, Electrical Characteristics) of time to program. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the code MRAM. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming.

65.14.6 Synchronization of Code MRAM Program Completion and Subsequent Instruction Generation

When CPU writes to code MRAM, there is a time lag from execution of MRAM programming by code MRAM program sequencer. Therefore, after programming is finished, appropriate processing for synchronization is required for executing subsequent instructions. The processing for synchronization is shown as follow:

1. Code MRAM programming.
2. Confirm that programming of the code MRAM is finished.
3. Flush the Code MRAM programming data (see Figure 65.2 and Figure 65.3).
4. Synchronization (wait for flushing process to finish).
5. Subsequent instruction.

If it is necessary to avoid CPU stall in step 4., completion of code MRAM programming should be checked by the PRGBSYC bit in the MRCPS register instead of dummy reading in step 3.

65.14.7 Constraints During Programming of MRAM

Frequency change, transition to software standby, and transition to deep standby are prohibited while MRAM is being programmed.

65.14.8 Power Supply in Any Type of Boot Mode

We recommend against rewriting MRAM while power is being supplied from a host*1 as a process that is part of mass production.
If power is to be supplied from a host, proceed with sufficient evaluation in the user environment before putting this into practice.

Note 1. “Power is being supplied from a host” means that power is supplied by a device such as an emulator or by a USB host through a USB cable (USB bus power).

66. Internal Voltage Regulator

66.1 Overview

This MCU supports three types of power supply for the core voltage.

- Switching regulator (DCDC)
- External power supply from an external power device (external VDD)
- Linear regulator (LDO); however, this is only usable with R5K574xxE4xx and R5K574xxF4xx products.

Table 66.1 shows the specifications of the internal voltage regulator.
One of the power supply modes is selected as follows in accord with the connection of the VCC_DCDC pin and the setting of the UMRECR.DCDCCTL bit in the option-setting memory.

- The DCDC mode is selected when VCC is connected to the VCC_DCDC pin and the setting of the UMRECR.DCDCCTL bit is 1.
- The external VDD mode is selected when VCC is connected to the VCC_DCDC pin and the setting of the UMRECR.DCDCCTL bit is 0.
- The LDO mode is selected when VSS is connected to the VCC_DCDC pin, regardless of the setting of the UMRECR.DCDCCTL bit.

Table 66.1 Selection of the Power Supply

VCC_DCDC Pin Connection	UMRECR.DCDCCTL Bit	Power Mode
VCC	1	DCDC mode
VCC	0	External VDD mode
VSS	Don't care	LDO mode

66.2 Operation

66.2.1 DCDC Mode

Table 66.2 lists the DCDC mode pin settings, and Figure 66.1 shows the DCDC mode settings.
In DCDC mode, VDD is supplied from VCL through VLO output and an external inductor and capacitor.

Table 66.2 DCDC Mode Pin Settings

Pin	Settings
All VCC and VCC2 pins	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1-μF multilayer ceramic capacitor. Place the capacitor close to the pins.
VCC_DCDC pins	• Connect the pins to the system power supply. • Connect the pins to VSS_DCDC through a 22-μF and 0.1-μF multilayer ceramic capacitors in parallel. Place the capacitor close to the pins.
VCL pins	• Connect the each pin to VSS through a 0.22-μF*1 multilayer ceramic capacitor. Place the capacitor close to the pins. • Connect each pin to an external inductor and capacitor. Place the inductor and capacitor close to the pins (see Figure 66.1).
All VLO pins	Connect each pin to an external inductor and capacitor. Place the 2.2-μH inductor and 47-μF capacitor close to the pins. Connect the capacitor to VSS_DCDC.

Note 1. 0.47 μF for 100-pin products only

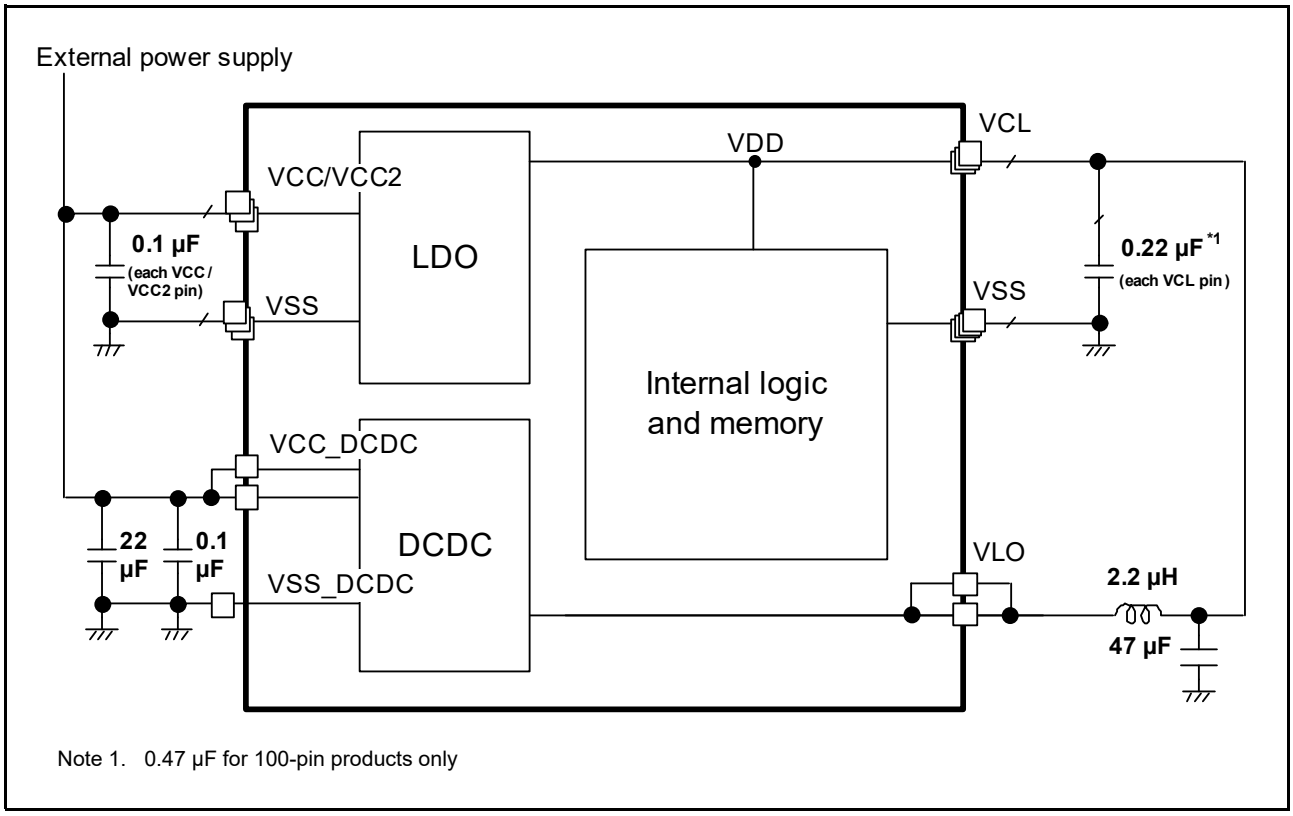


Figure 66.1 DCDC Mode Setting

66.2.2 External VDD Mode

Table 66.3 lists the external VDD mode pin settings, and Figure 66.2 shows the external VDD mode settings. VDD is supplied from the VCL pin.

Note: Software standby mode 1, 2, deep software standby mode 1, 2, 3, battery backup function, and voltage scaling control are not supported in this mode.

Table 66.3 External VDD Mode Settings

Pin	Settings
All VCC and VCC2 pins	• Connect each pin to the system power supply device for VCC. • Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pins.
VCC_DCDC pins	• Connect the pins to the system power supply.
VCL pins	• Connect each pin to the system power supply device for VDD. • Connect each pin to VSS through a 0.22-µF*1 multilayer ceramic capacitor. Place the capacitor close to the pins.
All VLO pins	Keep the pins open.

Note 1. 0.47 µF for 100-pin products only

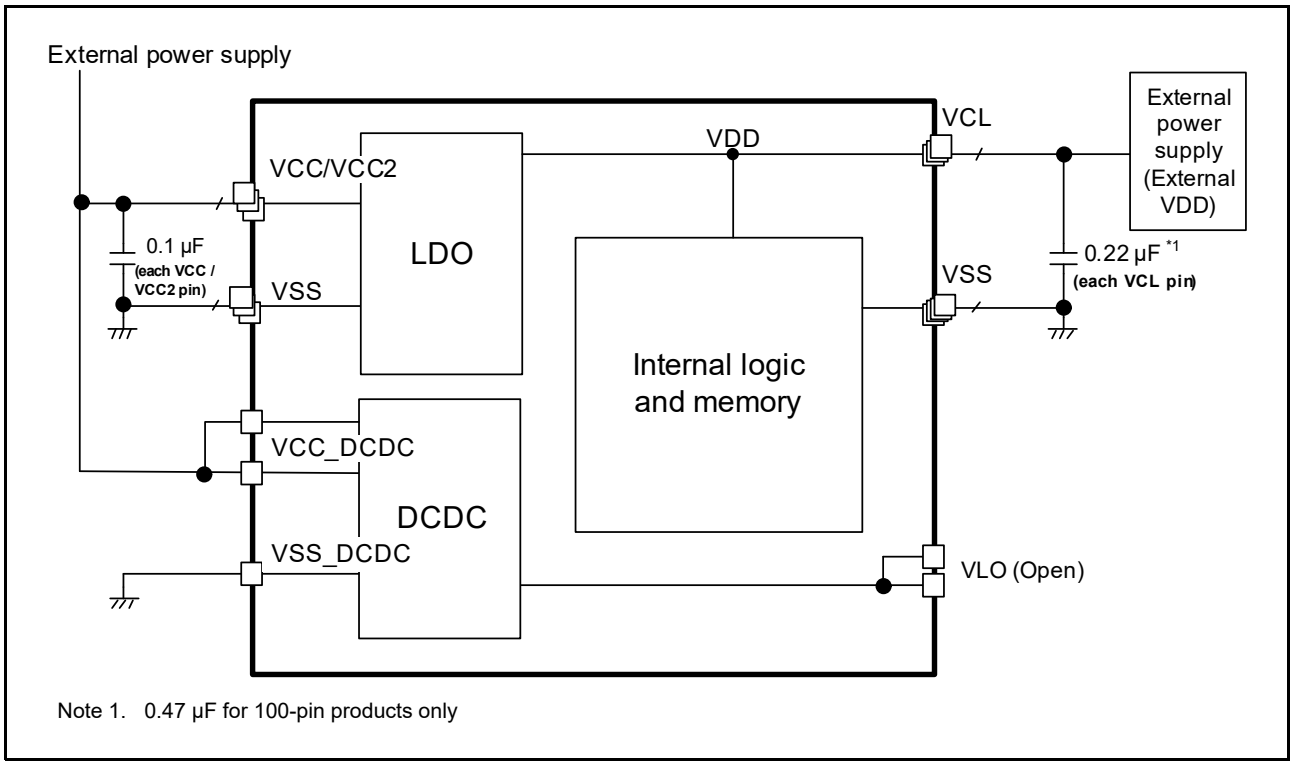


Figure 66.2 External VDD Mode Setting

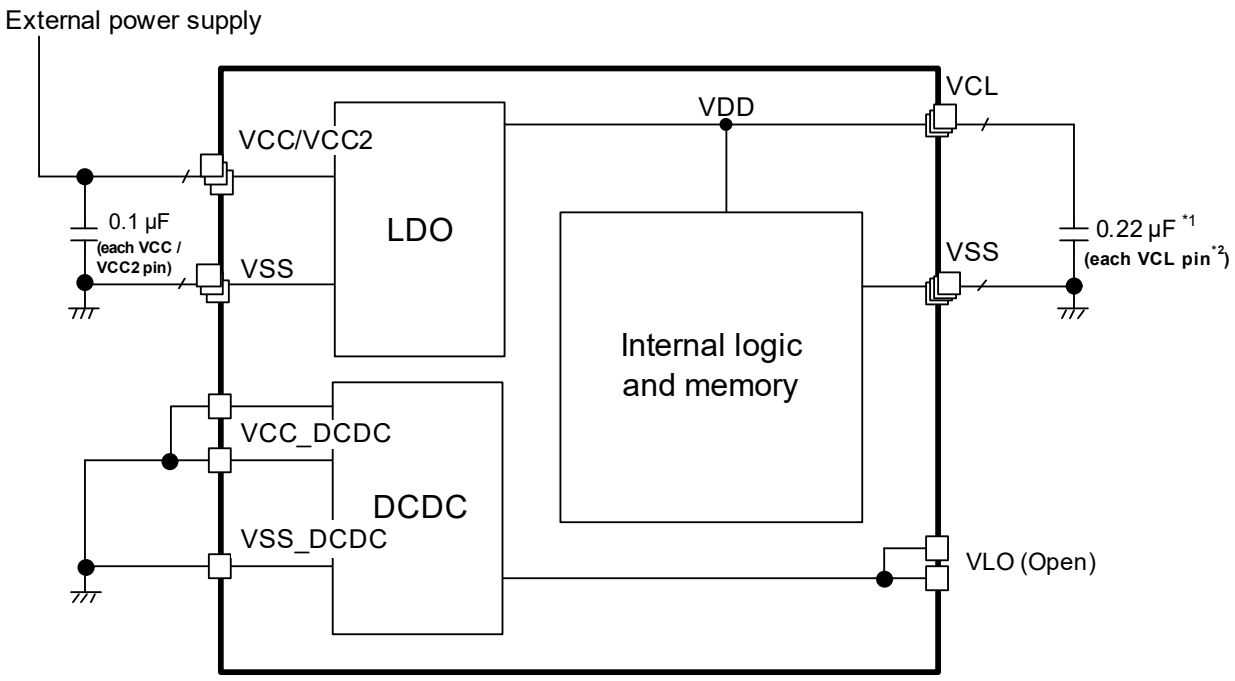
66.2.3 LDO Mode

Table 66.4 lists the LDO mode pin settings, and Figure 66.3 shows the LDO mode settings.
VDD is supplied from the LDO in LDO mode.

Table 66.4 LDO Mode Pin Settings

Pin	Settings
All VCC and VCC2 pins	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pins.
VCC_DCDC pins	• Connect the pins to VSS.
VCL pins	• Connect the pins*1 to VSS through a 0.22-µF*2 multilayer ceramic capacitor. Place the capacitor close to the pins.
All VLO pins	Keep the pins open.

Note 1. For products with more than 10 VCL pins, connect capacitors to no more than 10 VCL pins and leave the remaining pins open.
Note 2. 0.47 µF for 100-pin products only



Note 1. 0.47 µF for 100-pin products only
Note 2. For products with more than 10 VCL pins, connect capacitors to no more than 10 VCL pins and leave the remaining pins open.

Figure 66.3 LDO Mode Setting

66.3 Usage Notes

In DCDC mode, to minimize power loss and maximize efficiency, a 2.2-µH inductor with a DC resistance of 100 mΩ or less is recommended. Short VCC and VCC_DCDC.
In external VDD mode, the voltage of VDD should always be below the voltage of VCC including the power-on and power-off sequence. See section 67, Electrical Characteristics for more information.

67. Electrical Characteristics

67.1 Test Conditions

The supported peripheral functions and pins vary depending on the product model.
Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- VCC = VCC_DCDC = VBATT = 1.62 to 3.63 V
- VCC2 = 1.62 to 3.63 V
- AVCC0 = 1.62 to 3.63 V
- VCC_USB = VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
- VREFH00/VREFH01/VREFH02/VREFH0 = 1.62 V to AVCC0
- VSS = VSS_DCDC = AVSS0 = VREFL00/VREFL01/VREFL02/VREFL0 = VSS_USB = VSS_USBHS = AVSS_USBHS = 0 V
- VCC voltage is lower than 2.7 V : LVOCR.LVO0E = 1, otherwise LVOCR.LVO0E = 0
- Tj = Topj
- V_{OH} = 0.7 × VCC/VCC2, V_{OL} = 0.3 ×VCC/VCC2
- V_{IH} = 0.7 × VCC/VCC2, V_{IL} = 0.3 ×VCC/VCC2
- Capacitive load C = 30 pF

67.1.1 Maximum and Minimum Values (Max./Min.)

Unless otherwise specified, minimum and maximum values are guaranteed by either design simulation, characterization results or test in production.

67.1.2 Typical Values (Typ.)

Unless otherwise specified, the following conditions were used in obtaining the typical (Typ.) values:

- Room temperature of 25°C
- VCC/VCC2 = VCC_DCDC = VCC_USB = VBATT = VCC_USBHS = AVCC_USBHS = AVCC0 = VREFH00/VREFH01/VREFH02/VREFH0 = 3.3 V

67.2 Absolute Maximum Ratings

Table 67.1 Absolute Maximum Rating

Item			Symbol	Value	Unit
Power supply voltage			VCC, VCC2, VCC_DCDC	–0.3 to +4.0	V
External power supply voltage			VCL	–0.3 to +1.2	V
VBATT power supply voltage			V_{BATT}	–0.3 to +4.0	V
USB power supply voltage			VCC_USB	–0.3 to +4.0	V
USBHS power supply voltage			VCC_USBHS	–0.3 to +4.0	V
USBHS analog power supply voltage			AVCC_USBHS	–0.3 to +4.0	V
Analog power supply voltage			AVCC0	–0.3 to +4.0	V
Reference power supply voltage			VREFH00, VREFH01, VREFH02, VREFH0	–0.3 to AVCC0 + 0.3	V
Input voltage for port pins that are not 5-V tolerant	P000 to P015, P108 to P115, P208 to P215, P300 to P307, P315, P500 to P515, P700 to P715, P800 to P807, PA00 to PA07, PB00 to PB07, PC00 to PC07, PE00 to PE07, PF02 to PF09, PG00 to PG15		V_{in}	–0.3 to VCC + 0.3	V
	P309	EXCIN (when the VCC power supply is selected)		–0.3 to VCC + 0.3	
		EXCIN (when the VBATT power supply is selected)		–0.3 to VBATT + 0.3	
		Other than above		–0.3 to VCC + 0.3	
	P310, P311, P312	RTCICn (n = 0 to 2)		–0.3 to +3.9	
		Other than above		–0.3 to VCC + 0.3 (up to 3.9)	
	P600 to P607, P900 to P907, PD00 to PD07			–0.3 to VCC2 + 0.3	
	P400 to P415			–0.3 to AVCC0 + 0.3	
	PF00, PF01			–0.3 to VCC_USB + 0.3	
	USBHS_DP, USBHS_DM			–0.3 to VCC_USBHS + 0.3	
Input voltage for port pins that are 5-V tolerant	P100 to P107, P200 to P207		V_{in}	–0.3 to VCC + 3.6 (up to 5.8)	V
Analog input voltage	P400 to P415		V_{AN}	–0.3 to AVCC0 + 0.3	V
	P000 to P007	AVCC0 ≤ VCC		–0.3 to AVCC0 + 0.3	
		VCC < AVCC0		–0.3 to VCC + 0.3	
	P900 to P903, PD00 to PD07	AVCC0 ≤ VCC2		–0.3 to AVCC0 + 0.3	
		VCC2 < AVCC0		–0.3 to VCC2 + 0.3	
Junction temperature			T_j	–40 to +125	°C
Storage temperature			T_{stg}	–55 to +125	°C

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

67.3 Recommended Operating Conditions

Table 67.2 Recommended Operating Conditions (1)

Item			Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	When ESWM and SDRAM are not in use		VCC, VCC_DCDC	1.62	—	3.63	V
	When ESWM is in use			2.30	—	3.63	
	When SDRAM is in use			3.00	—	3.63	
	When ESWM and SDRAM are not in use		VCC2	1.62	—	3.63	
	When ESWM is in use*1			2.30	—	3.63	
	When SDRAM is in use*2			3.00	—	3.63	
	When external VDD is in use*3	Voltage range 1	VCL	0.920	0.950	0.990	
		Voltage range 2		0.870	0.925	0.990	
	When DCDC is in use	VSCR_1		—	0.950	—	
		VSCR_2		—	0.925	—	
		SVSCR_1		—	0.950	—	
		SVSCR_2		—	0.925	—	
		SVSCR_3		—	0.825	—	
		SVSCR_4		—	0.765		
		SVSCR_5		—	0.715	—	
			VSS, VSS_DCDC	—	0	—	
VBATT power supply voltage			V_{BATT}	1.62	—	3.63	V
USB power supply voltage	When USB is not in use		VCC_USB, VCC_USBHS, AVCC_USBHS	1.62	—	3.63	V
	When USB is in use			3.00	—	3.60	
			VSS_USB, VSS_USBHS, AVSS_USBHS	—	0	—	
Analog power supply voltage	Other than the following		AVCC0*4	1.62	—	3.63	V
	When sample-and-hold circuit is in use			2.70	—	3.63	
			AVSS0	—	0	—	

Note 1. When the pins for the ESWM with the VCC2 power supply are not in use, the voltage conditions differ from those when the ESWM is in use.
Note 2. When the pins for the SDRAM with the VCC2 power supply are not in use, the voltage conditions differ from those when the SDRAM is in use.
Note 3. Ensure that the VCL voltage is no higher than the VCC voltage.
Note 4. When the A/D converter, the D/A converter and the comparator are not in use, connect the AVCC0 and VREFH0 pins to VCC, and the AVSS0 and VREFL pins to VSS, respectively.

Table 67.3 Recommended Operating Conditions (2)

Item	Package	Symbol	Value	Required number of capacitors
Capacitance of a decoupling capacitor for stabilizing the internal voltage	BGA320	C_{VCL}	0.22 µF ± 30%	Nine or more
	BGA224			
	BGA176			
	LQFP176			Nine
	LQFP144			Eight
	LQFP100		0.47 µF ± 30%	Four

Note: Use a multilayer ceramic capacitor with a stated capacitance of 0.22 µF or 0.47µF and a capacitance tolerance within the range ±30%.

67.4 DC Characteristics

67.4.1 Characteristics of the Input Voltage of the I/O Pins

Table 67.4 Characteristics of the Input Voltage of the I/O Pins (V_{IH}/V_{IL}) (1/2)

Item		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min.	Typ.	Max.	Unit
High- and low-level input voltage (except for Schmitt trigger input pins)	EXTAL (excluding usage with external clock input)	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	V
			V_{IL}	—	—	0.2 × VCC	
	RSPI (excluding usage with RSPCK)*1	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	
			V_{IL}	—	—	0.2 × VCC	
			V_{IH}	0.8 × VCC2	—	—	
			V_{IL}	—	—	0.2 × VCC2	
	XSPI*2	2.70 V or higher	V_{IH}	0.8 × VCC	—	—	
			V_{IL}	—	—	0.2 × VCC	
			V_{IH}	0.8 × VCC2	—	—	
			V_{IL}	—	—	0.2 × VCC2	
		1.70 V to 1.95 V	V_{IH}	0.7 × VCC	—	—	
			V_{IL}	—	—	0.3 × VCC	
			V_{IH}	0.7 × VCC2	—	—	
			V_{IL}	—	—	0.3 × VCC2	
	SDHI	2.70 V or higher	V_{IH}	0.625 × VCC	—	—	
			V_{IL}	—	—	0.25 × VCC	
			V_{IH}	0.625 × VCC2	—	—	
			V_{IL}	—	—	0.25 × VCC2	
		1.70 V to 1.95 V	V_{IH}	1.27	—	2	
			V_{IL}	—	—	0.58	
	MMC	2.70 V or higher	V_{IH}	0.625 × VCC	—	—	
			V_{IL}	—	—	0.25 × VCC	
			V_{IH}	0.625 × VCC2	—	—	
			V_{IL}	—	—	0.25 × VCC2	
		1.70 V to 1.95 V	V_{IH}	0.65 × VCC	—	—	
			V_{IL}	—	—	0.35 × VCC	
			V_{IH}	0.65 × VCC2	—	—	
			V_{IL}	—	—	0.35 × VCC2	
	D0 to D31, BREQ#, WAIT#	3.00 V or higher	V_{IH}	0.7 × VCC	—	—	
			V_{IL}	—	—	0.3 × VCC	
			V_{IH}	0.7 × VCC2	—	—	
			V_{IL}	—	—	0.3 × VCC2	
	RIICHS (SMBus)*3	2.70 V or higher	V_{IH}	2.3	—	—	
			V_{IL}	—	—	0.8	
	RI3C (SMBus)*4	2.70 V or higher	V_{IH}	2.3	—	—	
			V_{IL}	—	—	0.8	
	ESWM(MII*5), ESWM(RMII*6)	2.70 V to 3.60 V	V_{IH}	2.3	—	—	
			V_{IL}	—	—	0.2 × VCC	
			V_{IH}	2.3	—	—	
			V_{IL}	—	—	0.2 × VCC2	
	ESWM(GMII*7), ESWM(RGMII*8)	3.00 V to 3.60 V	V_{IH}	2	—	—	
			V_{IL}	—	—	0.8	
		2.30 V to 2.70 V	V_{IH}	1.7	—	—	
			V_{IL}	—	—	0.7	

Table 67.4 Characteristics of the Input Voltage of the I/O Pins (V_{IH}/V_{IL}) (2/2)

Item		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min.	Typ.	Max.	Unit
High- and low-level input voltage (except for Schmitt trigger input pins)	ESWM*9	2.30 V to 3.60 V	V_{IH}	0.7 × VCC	—	—	V
			V_{IL}	—	—	0.3 × VCC	
			V_{IH}	0.7 × VCC2	—	—	
			V_{IL}	—	—	0.3 × VCC2	
	RTCIC0, RTCIC1, RTCIC2 (when the VCC power supply is selected)	1.62 V or higher	V_{IH}	0.9	—	—	
			V_{IL}	—	—	0.3	
	RTCIC0, RTCIC1, RTCIC2 (when the VBATT power supply is selected)		V_{IH}	0.9	—	—	
			V_{IL}	—	—	0.3	
	EXCIN (when the VCC power supply is selected)	1.62 V or higher	V_{IH}	0.9	—	—	
			V_{IL}	—	—	0.3	
	EXCIN (when the VBATT power supply is selected)		V_{IH}	0.9	—	—	
			V_{IL}	—	—	0.3	
Schmitt trigger input voltage (for the peripheral function pins)	RIICHS input pins (except for SMBus)*3	1.62 V or higher	V_{IH}	0.7 × VCC	—	—	V
			V_{IL}	—	—	0.3 × VCC	
			ΔV_T	0.05 × VCC	—	—	
	RI3C input pins (I3C mode)*4	1.65 V to 1.95 V 2.97 V to 3.63 V	V_{IH}	0.7 × VCC	—	—	
			V_{IL}	—	—	0.3 × VCC	
			ΔV_T	0.1 × VCC	—	—	
	RI3C input pins (I^2C mode excluding SMBus operation)*4	1.62 V or higher	V_{IH}	0.7 × VCC	—	—	
			V_{IL}	—	—	0.3 × VCC	
			ΔV_T	0.1 × VCC	—	—	
	5-V tolerant port pins*10	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	
			V_{IL}	—	—	0.2 × VCC	
			ΔV_T	0.05 × VCC	—	—	
	Other pins in the VCC power domain	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	
			V_{IL}	—	—	0.2 × VCC	
			ΔV_T	0.05 × VCC	—	—	
	Other pins in the VCC2 power domain	1.62 V or higher	V_{IH}	0.8 × VCC2	—	—	
			V_{IL}	—	—	0.2 × VCC2	
			ΔV_T	0.05 × VCC2	—	—	
	Other pins in the AVCC0 power domain	1.62 V or higher	V_{IH}	0.8 × AVCC0	—	—	
			V_{IL}	—	—	0.2 × AVCC0	
			ΔV_T	0.05 × AVCC0	—	—	
	Other pins in the VCC_USB power domain	1.62 V or higher	V_{IH}	0.8 × VCC_USB	—	—	
			V_{IL}	—	—	0.2 × VCC_USB	
			ΔV_T	0.05 × VCC_USB	—	—	
Schmitt trigger input voltage (for the port pins)	5-V tolerant port pins*10	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	V
			V_{IL}	—	—	0.2 × VCC	
	Other pins in the VCC power domain	1.62 V or higher	V_{IH}	0.8 × VCC	—	—	
			V_{IL}	—	—	0.2 × VCC	
	Other pins in the VCC2 power domain	1.62 V or higher	V_{IH}	0.8 × VCC2	—	—	
			V_{IL}	—	—	0.2 × VCC2	
	Other pins in the AVCC0 power domain	1.62 V or higher	V_{IH}	0.8 × AVCC0	—	—	
			V_{IL}	—	—	0.2 × AVCC0	
	Other pins in the VCC_USB power domain	1.62 V or higher	V_{IH}	0.8 × VCC_USB	—	—	
			V_{IL}	—	—	0.2 × VCC_USB	
	Other pins in the VBATT power domain	1.62 V or higher	V_{IH}	0.8 × VCC	—	VCC	
			V_{IL}	—	—	0.2 × VCC	

Note: The items in this table are classified by the power supplied to the individual ports. For details on the power supplied to the ports, see section 22, I/O Ports.
Note 1. MOSIn, MISOn, SSLn0 (n = A, B, C)
Note 2. XSPIn_IOx, XSPIn_DS, XSPIn_CK, XSPIn_CS0# (n = 0, 1; x = 0 to 7)
Note 3. SDAHSn-A, SCLHSn-A (n = 1, 2)
Note 4. SDA00-A, SCL00-A
Note 5. ET0_RX_CLK, ET0_RX_DV, ET0_RXD0, ET0_RXD1, ET0_RXD2, ET0_RXD3, ET0_RX_ER, ET0_TX_CLK, ET0_CRS, ET0_COL
Note 6. RMII0_REF50CK, RMII0_CRS_DV, RMII0_RXD0, RMII0_RXD1, RMII0_RX_ER
Note 7. ET0_RX_CLK-A, ET0_RX_DV-A, ET0_RX_ER-A, ET0_RXD0-A, ET0_RXD1-A, ET0_RXD2-A, ET0_RXD3-A, ET0_RXD4-A, ET0_RXD5-A, ET0_RXD6-A, ET0_RXD7-A, ET0_TX_CLK-A
Note 8. RGMII0_RX_CTL-A, RGMII0_RXC-A, RGMII0_RXD0-A, RGMII0_RXD1-A, RGMII0_RXD2-A, RGMII0_RXD3-A
Note 9. GPTP_CAPTURE0, ET0_LINKSTA, ET0_MDIO, ET0_INT
Note 10. P100 to P107 and P200 to P207 are 5-V tolerant.

67.4.2 Characteristics of the Output Voltage of the I/O Pins

Table 67.5 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (1) (1/3)

Item		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	RI3C*1	2.97 V to 3.63 V	V_{OH}	VCC – 0.27	—	—	V	I_{OH} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
		1.65 V to 1.95 V		VCC – 0.27	—	—		
	ESWM	2.70 V to 3.60 V	V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –1.0 mA
				VCC2 – 0.5	—	—		
		2.30 V to 2.70 V		2	—	—		
	SDHI	2.70 V or higher	V_{OH}	0.75 × VCC	—	—	V	I_{OH} = –2.0 mA
				0.75 × VCC2	—	—		
		1.70 V to 1.95 V		1.4	—	—		
	MMC	2.70 V or higher	V_{OH}	0.75 × VCC	—	—	V	I_{OH} = –0.1 mA
				0.75 × VCC2	—	—		
		1.70 V to 1.95 V		VCC – 0.45	—	—		I_{OH} = –2.0 mA
				VCC2 – 0.45	—	—		
	Other output pins		V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –1.0 mA
				VCC2 – 0.5	—	—		
				AVCC0 – 0.5	—	—		
				VCC_USB – 0.5	—	—		
Low-level output voltage	RIICHS output pins*2	2.70 V or higher	V_{OL}	—	—	0.4	V	I_{OL} = 3 mA (ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.6		I_{OL} = 6 mA (ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
				—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
		1.62 V to 1.95 V		—	—	0.2 × VCC		I_{OL} = 2 mA (ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 3 mA (ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1, ICFER.HSME = 0)
				—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1, ICFER.HSME = 0)
				—	—	0.2 × VCC		I_{OL} = 3 mA (ICFER.FMPE = 0, ICFER.HSME = 1)

Table 67.5 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (1) (2/3)

Item		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Low-level output voltage	RI3C*1	2.97 V or higher	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 1)
				—	—	0.27		I_{OL} = 3.0 mA (ICMR.OMS = 0, ICFER.FMPE = 0, ICFER.HSME = 0)
		2.70 V or higher		—	—	0.4	V	I_{OL} = 3.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.6		I_{OL} = 6.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 15.0 mA (ICMR.OMS = 1, ICFER.FMPE = 1, ICFER.HSME = 0)
				—	0.4	—		I_{OL} = 20.0 mA (ICMR.OMS = 1, ICFER.FMPE = 1, ICFER.HSME = 0)
		1.62 V to 1.95 V		—	—	0.2 × VCC	V	I_{OL} = 2.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 3.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.6		I_{OL} = 6.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 0)
				—	—	0.2 × VCC		I_{OL} = 2.0 mA (ICMR.OMS = 1, ICFER.FMPE = 1, ICFER.HSME = 0)
				—	—	0.4		I_{OL} = 15.0 mA (ICMR.OMS = 1, ICFER.FMPE = 1, ICFER.HSME = 0)
				—	0.4	—		I_{OL} = 20.0 mA (ICMR.OMS = 1, ICFER.FMPE = 1, ICFER.HSME = 0)
				—	—	0.2 × VCC		I_{OL} = 3.0 mA (ICMR.OMS = 1, ICFER.FMPE = 0, ICFER.HSME = 1)
		1.65 V to 1.95 V		—	—	0.27		I_{OL} = 3.0 mA (ICMR.OMS = 0, ICFER.FMPE = 0, ICFER.HSME = 0)
	ESWM		V_{OL}	—	—	0.4	V	I_{OL} = 1.0 mA

Table 67.5 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (1) (3/3)

Item		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Low-level output voltage	SDHI	2.70 V or higher	V_{OL}	—	—	0.125 × VCC	V	I_{OL} = 2.0 mA
				—	—	0.125 × VCC2		
		1.70 V to 1.95 V		—	—	0.45		I_{OL} = 2.0 mA
	MMC	2.70 V or higher	V_{OL}	—	—	0.125 × VCC	V	I_{OL} = 0.1 mA
				—	—	0.125 × VCC2		I_{OL} = 0.1 mA
		1.70 V to 1.95 V		—	—	0.45		I_{OL} = 2.0 mA
	Other output pins		V_{OL}	—	—	0.5	V	I_{OL} = 1.0 mA
				—	—	0.5		
				—	—	0.5		
				—	—	0.5		

Note 1. This specification applies to SDA00-A and SCL00-A.
Note 2. This specification applies to SCLHS1-A, SDAHS1-A, SCLHS2-A, and SDAHS2-A.

Table 67.6 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (2) (1/3)

Item		Power supply voltage	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	DRV1*1	3.3 V	V_{OH}	—	3.15	—	V	I_{OH} = –1 mA
				—	3.00	—		I_{OH} = –2 mA
				—	2.70	—		I_{OH} = –4 mA
				—	2.10	—		I_{OH} = –8 mA
				—	1.05	—		I_{OH} = –15 mA
	DRV2*1	3.3 V	V_{OH}	—	3.23	—	V	I_{OH} = –1 mA
				—	3.15	—		I_{OH} = –2 mA
				—	3.00	—		I_{OH} = –4 mA
				—	2.70	—		I_{OH} = –8 mA
				—	2.18	—		I_{OH} = –15 mA
	DRV3*1	3.3 V	V_{OH}	—	3.25	—	V	I_{OH} = –1 mA
				—	3.20	—		I_{OH} = –2 mA
				—	3.10	—		I_{OH} = –4 mA
				—	2.90	—		I_{OH} = –8 mA
				—	2.55	—		I_{OH} = –15 mA
	DRV4*1	3.3 V	V_{OH}	—	3.27	—	V	I_{OH} = –1 mA
				—	3.24	—		I_{OH} = –2 mA
				—	3.18	—		I_{OH} = –4 mA
				—	3.06	—		I_{OH} = –8 mA
				—	2.85	—		I_{OH} = –15 mA
		1.8 V		—	1.75	—		I_{OH} = –1 mA
				—	1.70	—		I_{OH} = –2 mA
				—	1.60	—		I_{OH} = –4 mA
				—	1.40	—		I_{OH} = –8 mA
				—	1.05	—		I_{OH} = –15 mA
	DRV5*1	3.3 V	V_{OH}	—	3.26	—	V	I_{OH} = –1 mA
				—	3.23	—		I_{OH} = –2 mA
				—	3.16	—		I_{OH} = –4 mA
				—	3.01	—		I_{OH} = –8 mA
				—	2.76	—		I_{OH} = –15 mA

Table 67.6 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (2) (2/3)

Item		Power supply voltage	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	DRV6*1	3.3 V	V_{OH}	—	3.28	—	V	I_{OH} = –1 mA
				—	3.25	—		I_{OH} = –2 mA
				—	3.20	—		I_{OH} = –4 mA
				—	3.11	—		I_{OH} = –8 mA
				—	2.94	—		I_{OH} = –15 mA
	DRV7*1	3.3 V	V_{OH}	—	3.28	—	V	I_{OH} = –1 mA
				—	3.26	—		I_{OH} = –2 mA
				—	3.22	—		I_{OH} = –4 mA
				—	3.14	—		I_{OH} = –8 mA
				—	3.00	—		I_{OH} = –15 mA
		2.5 V		—	2.45	—		I_{OH} = –1 mA
				—	2.40	—		I_{OH} = –2 mA
				—	2.30	—		I_{OH} = –4 mA
				—	2.10	—		I_{OH} = –8 mA
				—	1.75	—		I_{OH} = –15 mA
	DRV8*1	3.3 V	V_{OH}	—	3.28	—	V	I_{OH} = –1 mA
				—	3.25	—		I_{OH} = –2 mA
				—	3.20	—		I_{OH} = –4 mA
				—	3.11	—		I_{OH} = –8 mA
				—	2.94	—		I_{OH} = –15 mA
		1.8 V		—	1.79	—		I_{OH} = –1 mA
				—	1.77	—		I_{OH} = –2 mA
				—	1.75	—		I_{OH} = –4 mA
				—	1.70	—		I_{OH} = –8 mA
				—	1.61	—		I_{OH} = –15 mA
Low-level output voltage	DRV1*1	3.3 V	V_{OL}	—	0.15	—	V	I_{OL} = 1 mA
				—	0.30	—		I_{OL} = 2 mA
				—	0.60	—		I_{OL} = 4 mA
				—	1.20	—		I_{OL} = 8 mA
				—	2.25	—		I_{OL} = 15 mA
	DRV2*1	3.3 V	V_{OL}	—	0.08	—	V	I_{OL} = 1 mA
				—	0.15	—		I_{OL} = 2 mA
				—	0.30	—		I_{OL} = 4 mA
				—	0.60	—		I_{OL} = 8 mA
				—	1.13	—		I_{OL} = 15 mA
	DRV3*1	3.3 V	V_{OL}	—	0.05	—	V	I_{OL} = 1 mA
				—	0.10	—		I_{OL} = 2 mA
				—	0.20	—		I_{OL} = 4 mA
				—	0.40	—		I_{OL} = 8 mA
				—	0.75	—		I_{OL} = 15 mA
	DRV4*1	3.3 V	V_{OL}	—	0.03	—	V	I_{OL} = 1 mA
				—	0.06	—		I_{OL} = 2 mA
				—	0.12	—		I_{OL} = 4 mA
				—	0.24	—		I_{OL} = 8 mA
				—	0.45	—		I_{OL} = 15 mA
		1.8 V		—	0.05	—		I_{OL} = 1 mA
				—	0.10	—		I_{OL} = 2 mA
				—	0.20	—		I_{OL} = 4 mA
				—	0.40	—		I_{OL} = 8 mA
				—	0.75	—		I_{OL} = 15 mA

Table 67.6 Characteristics of the Output Voltage of the I/O Pins (V_{OH}/V_{OL}) (2) (3/3)

Item		Power supply voltage	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Low-level output voltage	DRV5*1	3.3 V	V_{OL}	—	0.04	—	V	I_{OL} = 1 mA
				—	0.07	—		I_{OL} = 2 mA
				—	0.14	—		I_{OL} = 4 mA
				—	0.29	—		I_{OL} = 8 mA
				—	0.54	—		I_{OL} = 15 mA
	DRV6*1	3.3 V	V_{OL}	—	0.02	—	V	I_{OL} = 1 mA
				—	0.05	—		I_{OL} = 2 mA
				—	0.10	—		I_{OL} = 4 mA
				—	0.19	—		I_{OL} = 8 mA
				—	0.36	—		I_{OL} = 15 mA
	DRV7*1	3.3 V	V_{OL}	—	0.02	—	V	I_{OL} = 1 mA
				—	0.04	—		I_{OL} = 2 mA
				—	0.08	—		I_{OL} = 4 mA
				—	0.16	—		I_{OL} = 8 mA
				—	0.30	—		I_{OL} = 15 mA
		2.5 V		—	0.05	—		I_{OL} = 1 mA
				—	0.10	—		I_{OL} = 2 mA
				—	0.20	—		I_{OL} = 4 mA
				—	0.40	—		I_{OL} = 8 mA
				—	0.75	—		I_{OL} = 15 mA
	DRV8*1	3.3 V	V_{OL}	—	0.02	—	V	I_{OL} = 1 mA
				—	0.05	—		I_{OL} = 2 mA
				—	0.10	—		I_{OL} = 4 mA
				—	0.19	—		I_{OL} = 8 mA
				—	0.36	—		I_{OL} = 15 mA
		1.8 V		—	0.01	—		I_{OL} = 1 mA
				—	0.03	—		I_{OL} = 2 mA
				—	0.05	—		I_{OL} = 4 mA
				—	0.10	—		I_{OL} = 8 mA
				—	0.20	—		I_{OL} = 15 mA

Note 1. This is the driving ability as selected by the setting of the driving ability select (PmnPFS.DSCR[3:0]) bits.

67.4.3 Characteristics of the Other I/O Pins

Table 67.7 Characteristics of the Other I/O Pins

Item		Power supply voltage	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES# pin		$\lvert I_{in} \rvert$	—	—	5	μA	V_{in} = 0 V, V_{in} = 5.5 V
	NMI, XCIN, and XCOUT pins			—	—	1		V_{in} = 0 V, V_{in} = VCC
Three-state leakage current (off state)	Other than ports for 5-V tolerant		$\lvert I_{TSI} \rvert$	—	—	1	μA	V_{in} = 0 V, V_{in} = VCC, VCC2, AVCC0, VCC_USB
	Ports for 5-V tolerant*1			—	—	5		V_{in} = 0 V, V_{in} = 5.5 V
Input pull-up resistor current*2		2.70 V or higher	I_p	–300	—	–10	μA	VCC, VCC2, AVCC0, VCC_USB = 2.7 to 3.63 V V_{in} = 0 V
		1.62 V or higher		–300	—	–5		VCC, VCC2, AVCC0, VCC_USB = 1.62 to 3.63 V V_{in} = 0 V
Input capacitance	P414, P415		C_{in}	—	—	16	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz T_a = 25°C
	PF00, PF01, USB_DP, USB_DM			—	—	12		
	P102, P103, P106, P107, P200, P201			—	—	10		
	USBHS_DP, USBHS_DM			—	—	10		
	Other input pins			—	—	8		

Note 1. P100 to P107 and P200 to P207 are 5-V tolerant.
Note 2. The values are for pins excluding USB_DP, USB_DM, USBHS_DP, USBHS_DM, SDA00-A, SCL00-A, SDAHSn-A, and SCLHSn-A (n = 1, 2).

67.4.4 Operating and Standby Currents

The operating current and standby current depend on the product part number.
For the current characteristics of the products with part numbers of the form R5K574Nxy4xx (y = A, B, C, D), see section 67.4.4.1, Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = A, B, C, D).
For the current characteristics of the products with part numbers of the form R5K574Nxy4xx (y = E, F), see section 67.4.4.2, Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = E, F).
Unless otherwise specified, the supply current values were measured with all output pins load-free and all input pull-up resistors disabled.
The values for current given or used in this section do not include the current supplied for background operation (BGO).
The VCC current includes the VCC2 power supply current.

The currents were measured with the operating frequencies specified as shown in Table 67.8.

Table 67.8 Frequency Settings for the Individual Clocks

Item		ICLK	MRICLK	MRPCLK	BCLK	PCLKA	PCLKB, PCLKC	PCLKD	PCLKE	Unit
When the peripheral clock supplies are stopped*1	Frequency setting 1 (CPUCLK0 = 600 MHz)	300	300	150	Stopped	Stopped	Stopped	Stopped	Stopped	MHz
	Frequency setting 2 (CPUCLK0 = 480 MHz)	240	240	120	Stopped	Stopped	Stopped	Stopped	Stopped	
	Frequency setting 3 (CPUCLK0 = 300 MHz)	300	300	150	Stopped	Stopped	Stopped	Stopped	Stopped	
Other cases	Frequency setting 1 (CPUCLK0 = 600 MHz)	300	300	150	150	150	75	150	300	
	Frequency setting 2 (CPUCLK0 = 480 MHz)	240	240	120	120	120	60	120	240	
	Frequency setting 3 (CPUCLK0 = 300 MHz)	300	300	150	150	150	75	150	300	

Note 1. The frequencies of the clocks in the table are set for division by 64 in the SCKDIVCR register when the peripheral clock supplies are stopped.

The symbols shown in Figure 67.1, Supply Current Measurement Diagram (DCDC Mode), and Figure 67.2, Supply Current Measurement Diagram (LDO Mode), are used in the tables that follow the figures.

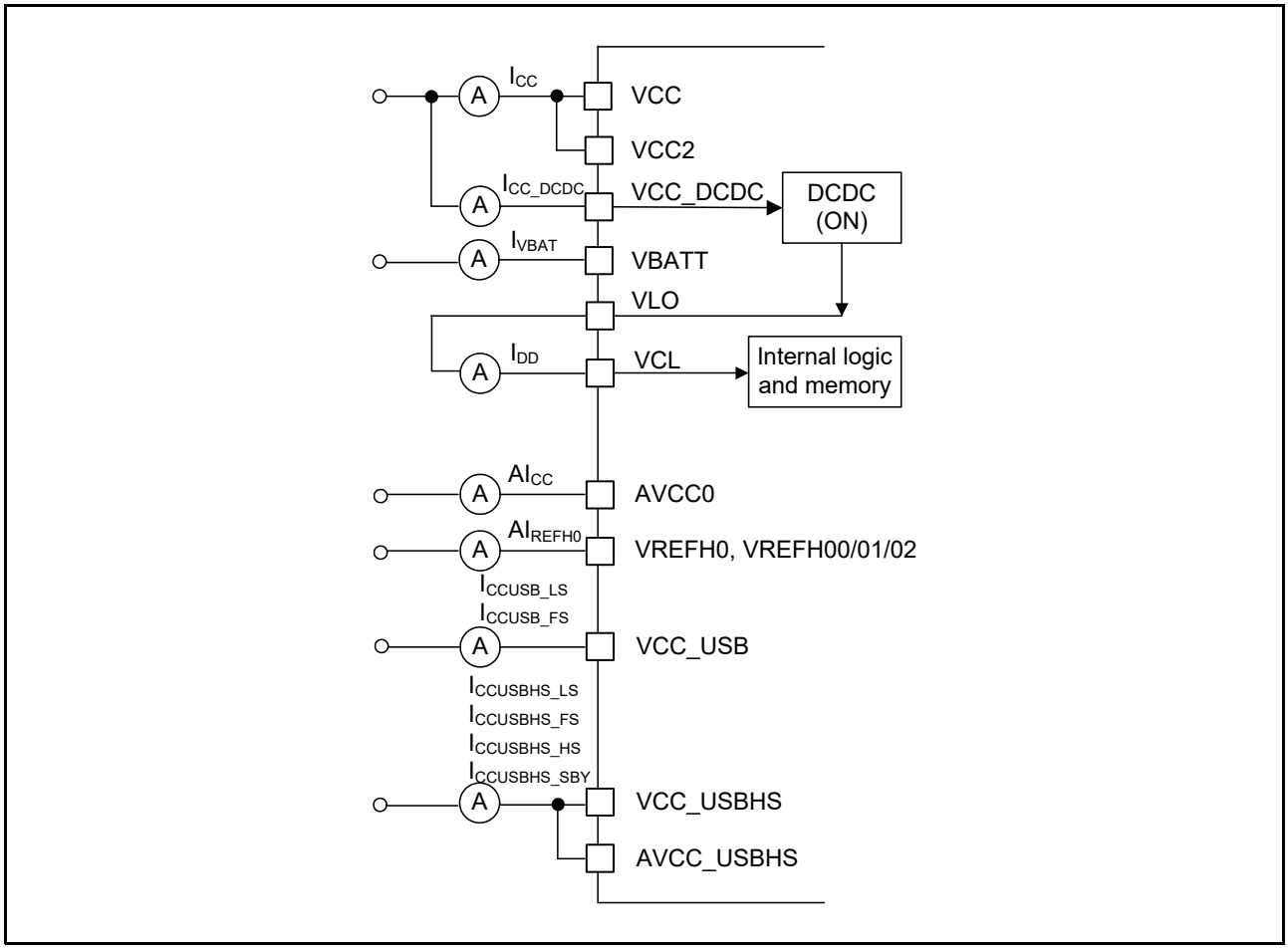


Figure 67.1 Supply Current Measurement Diagram (DCDC Mode)

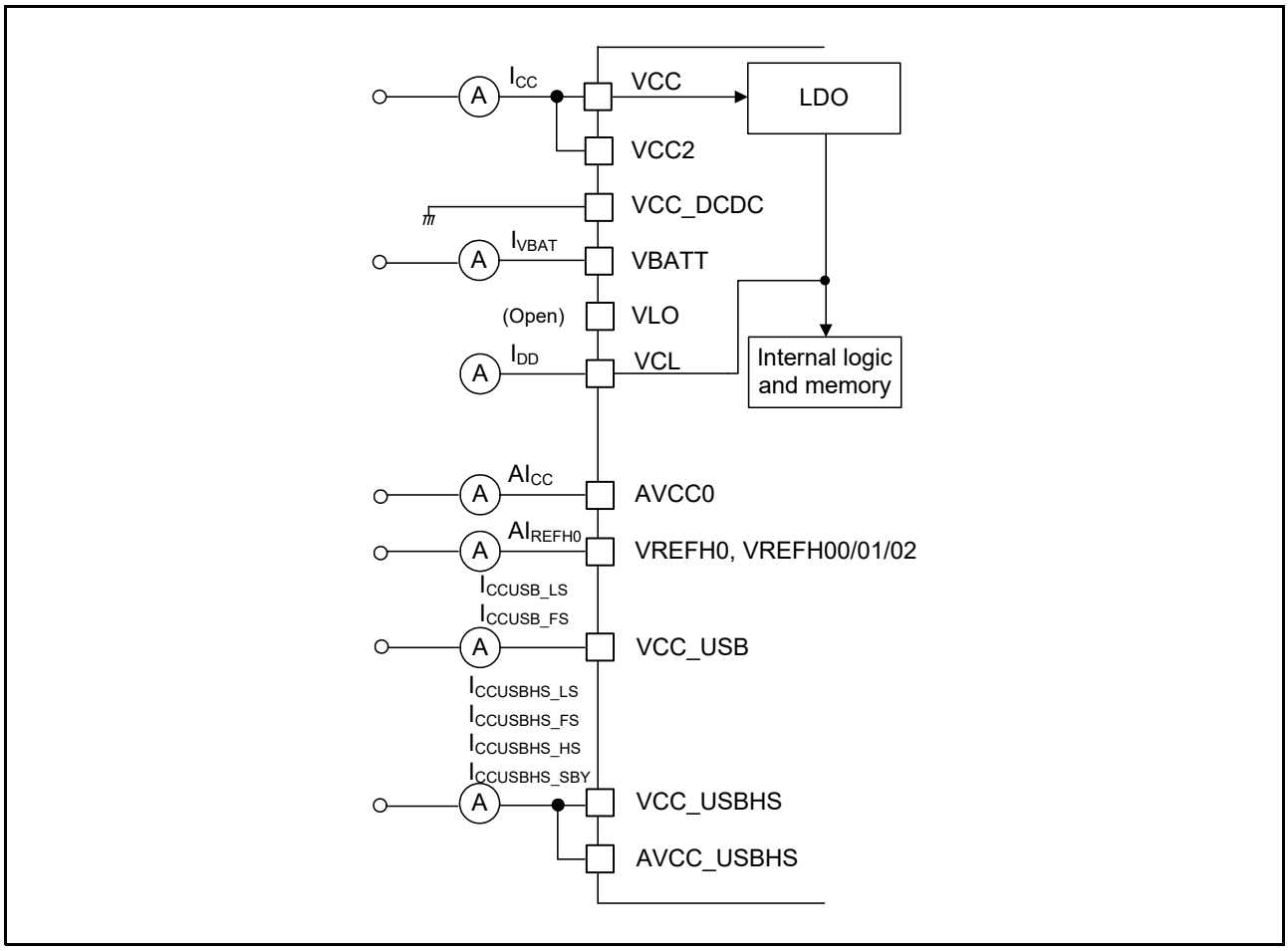


Figure 67.2 Supply Current Measurement Diagram (LDO Mode)

When the DCDC is in use, the efficiency of power conversion depends on the load currents, temperature, and VCC voltage. For reference, the results of conversion efficiency simulation are shown in Figure 67.3 to Figure 67.5.

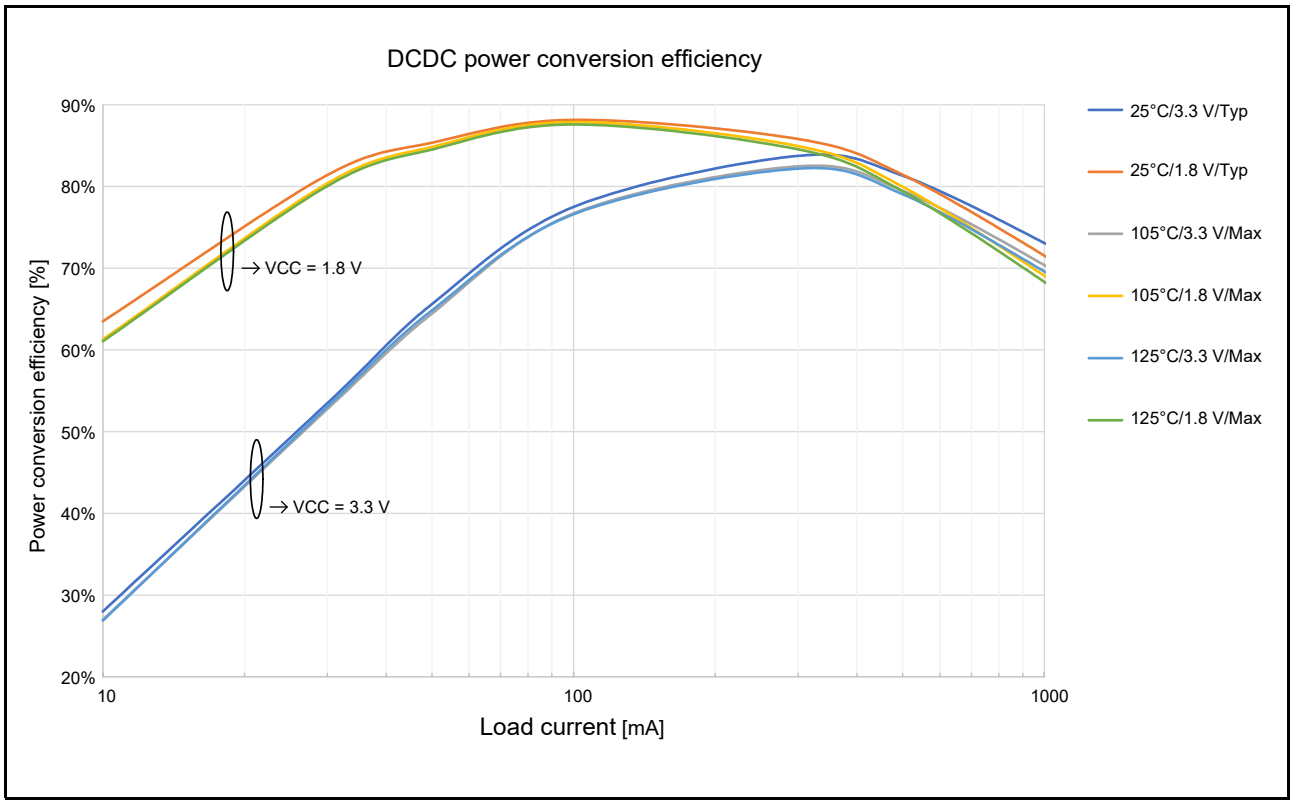


Figure 67.3 DCDC Power Conversion Efficiency in Normal Mode and Software Standby Mode (SSCR1.SS2LP = SS2LP_0) (Reference: Simulated Values)

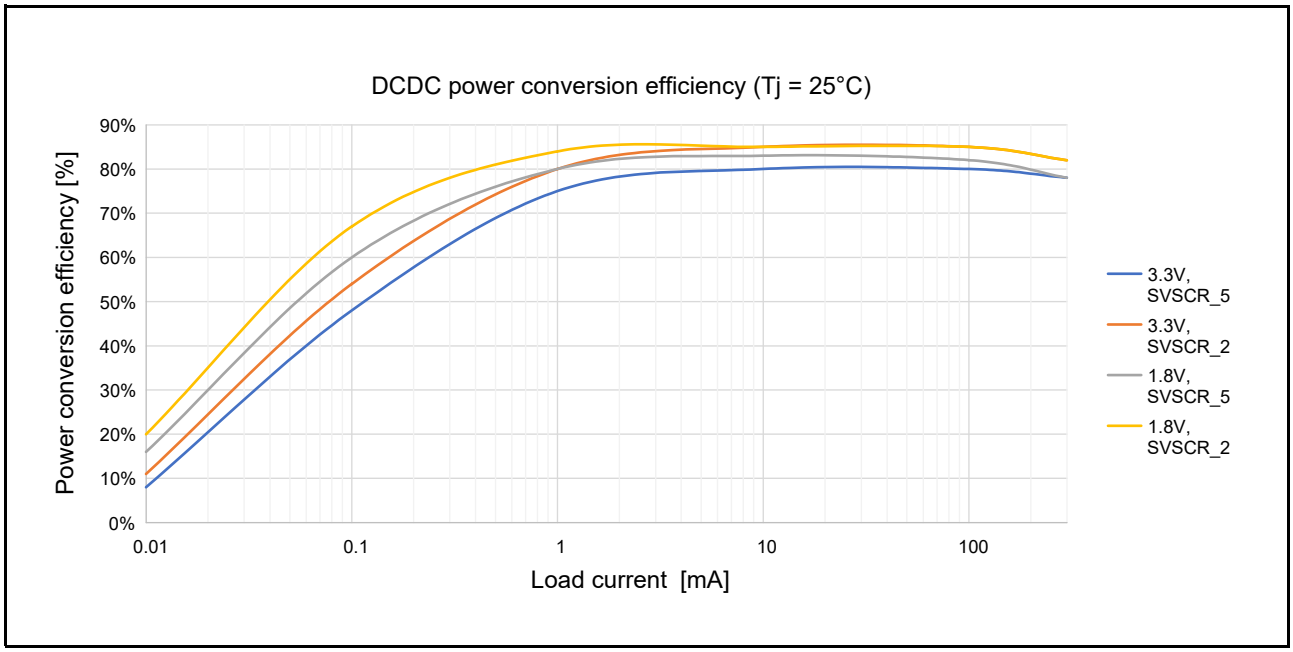


Figure 67.4 DCDC Power Conversion Efficiency in Software Standby Mode (SSCR1.SS2LP = SS2LP_1) at Tj = 25°C (Reference: Simulated Values)

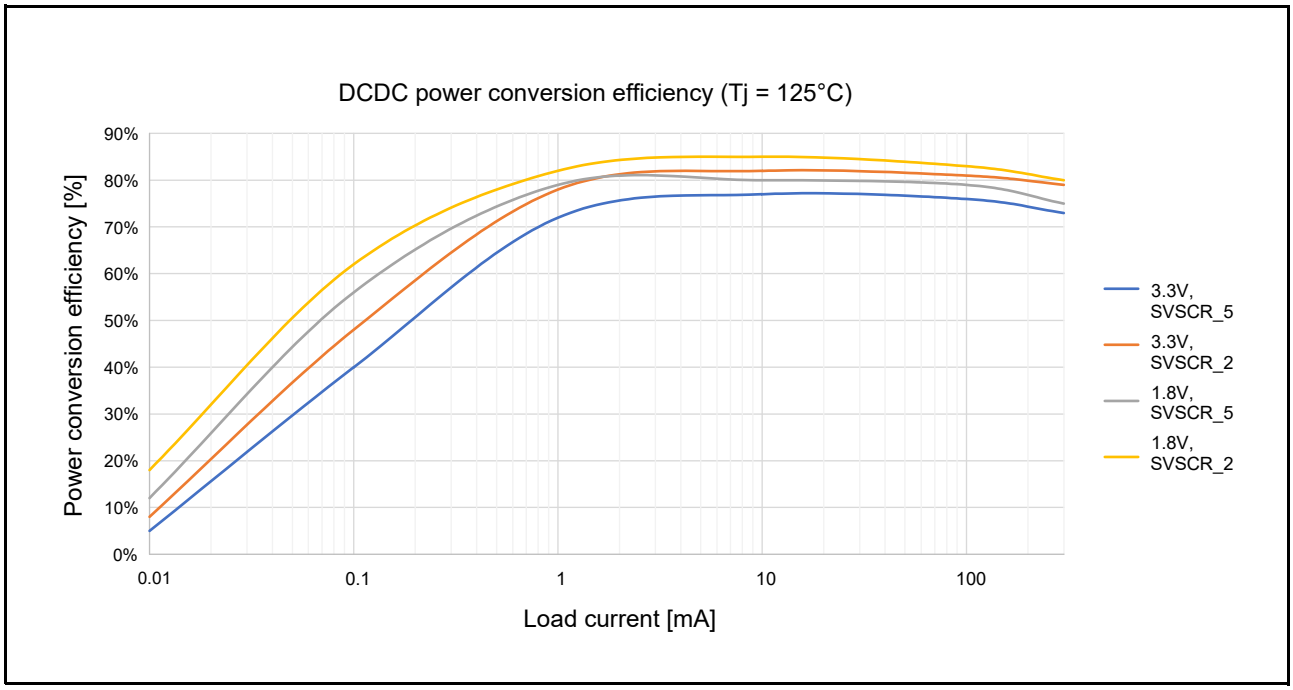


Figure 67.5 DCDC Power Conversion Efficiency in Software Standby Mode (SSCR1.SS2LP = SS2LP_1) at Tj = 125°C (Reference: Simulated Values)

67.4.4.1 Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = A, B, C, D)

Table 67.9 Supply Currents for Full Operation (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max.*2 (Tj = 125°C)	Unit	Test Conditions
Supply current (full operation)	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC} *1	88	217	307	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		68	181	262		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		68	180	261		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		153	396	568		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		116	329	482		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		115	329	482		
	VCC current		I_{CC}	11	15	17		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- The peripheral functions are supplied with clocks and kept as active as is possible.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).

[When VCC_DCDC = 3.3 V]

I_{CC_DCDC} Typ.(VSCR_1) = 0.0498 × $f_{CPUCLK0}$ + 0.0895 × f_{ICLK} + 30.84
I_{CC_DCDC} Typ.(VSCR_2) = 0.0512 × $f_{CPUCLK0}$ + 0.0633 × f_{ICLK} + 28.49
I_{CC_DCDC} Max.(VSCR_1, 105°C) = 0.0654 × $f_{CPUCLK0}$ + 0.1276 × f_{ICLK} + 139.00
I_{CC_DCDC} Max.(VSCR_1, 125°C) = 0.0716 × $f_{CPUCLK0}$ + 0.1305 × f_{ICLK} + 225.16
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0604 × $f_{CPUCLK0}$ + 0.0810 × f_{ICLK} + 132.18
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0679 × $f_{CPUCLK0}$ + 0.0907 × f_{ICLK} + 207.32

[When VCC_DCDC = 1.8 V]

I_{CC_DCDC} Typ.(VSCR_1) = 0.0974 × $f_{CPUCLK0}$ + 0.1756 × f_{ICLK} + 42.07
I_{CC_DCDC} Typ.(VSCR_2) = 0.0927 × $f_{CPUCLK0}$ + 0.1180 × f_{ICLK} + 43.29
I_{CC_DCDC} Max.(VSCR_1, 105°C) = 0.1223 × $f_{CPUCLK0}$ + 0.2385 × f_{ICLK} + 251.34
I_{CC_DCDC} Max.(VSCR_1, 125°C) = 0.1363 × $f_{CPUCLK0}$ + 0.2484 × f_{ICLK} + 411.75
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.1157 × $f_{CPUCLK0}$ + 0.1521 × f_{ICLK} + 237.43
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.1280 × $f_{CPUCLK0}$ + 0.1714 × f_{ICLK} + 379.37

However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. For the LQFP100 package (products with part numbers of the form R5K574NxxxFP), control the number of peripheral functions to ensure that the maximum current does not exceed 710 mA (see section 67.4.5.2, Supply Currents for the Peripheral Functions).

Table 67.10 Supply Currents for Full Operation (External VDD Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max.*2 (Tj = 125°C)	Unit	Test Conditions
Supply current (full operation)	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}*1	256	597	800	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		202	524	719		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		201	523	718		
	VCC current		I_{CC}	11	15	17		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.92 to 0.99 V) = 0.1584 × $f_{CPUCLK0}$ + 0.2852 × f_{ICLK} + 75.36
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.1588 × $f_{CPUCLK0}$ + 0.2007 × f_{ICLK} + 77.55
I_{DD} Max.(VCL = 0.92 to 0.99 V, 105°C) = 0.1600 × $f_{CPUCLK0}$ + 0.3120 × f_{ICLK} + 407.46
I_{DD} Max.(VCL = 0.92 to 0.99 V, 125°C) = 0.1565 × $f_{CPUCLK0}$ + 0.2849 × f_{ICLK} + 620.60
I_{DD} Max.(VCL = 0.87 to 0.99 V, 105°C) = 0.1622 × $f_{CPUCLK0}$ + 0.2122 × f_{ICLK} + 394.73
I_{DD} Max.(VCL = 0.87 to 0.99 V, 125°C) = 0.1615 × $f_{CPUCLK0}$ + 0.2126 × f_{ICLK} + 590.17
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. For the LQFP100 package (products with part numbers of the form R5K574NxxxFP), control the number of peripheral functions to ensure that the maximum current does not exceed 718 mA (see section 67.4.5.2, Supply Currents for the Peripheral Functions).

Table 67.11 Supply Currents for Maximum Data Processing with the Peripheral Clock Supplies Stopped (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current (maximum data processing with the peripheral clock supplies stopped)	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC}*1	41	165	254	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		33	144	225		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		28	138	218		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		66	299	467		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		52	260	412		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		42	248	399		
	VCC current		I_{CC}	3.9	7.1	9.7		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- The clock-signal supplies to the peripheral functions are stopped.
- Power to the ESWM domain is turned off.

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0352 × $f_{CPUCLK0}$ + 0.0163 × f_{ICLK} + 14.69
I_{CC_DCDC} Typ.(VSCR_2) = 0.0369 × $f_{CPUCLK0}$ + 0.0119 × f_{ICLK} + 12.17
I_{CC_DCDC} Max.(VSCR_1, 105°C) = 0.0454 × $f_{CPUCLK0}$ + 0.0180 × f_{ICLK} + 132.14
I_{CC_DCDC} Max.(VSCR_1, 125°C) = 0.0496 × $f_{CPUCLK0}$ + 0.0190 × f_{ICLK} + 218.27
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0443 × $f_{CPUCLK0}$ + 0.0162 × f_{ICLK} + 119.62
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0479 × $f_{CPUCLK0}$ + 0.0183 × f_{ICLK} + 197.45
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0674 × $f_{CPUCLK0}$ + 0.0279 × f_{ICLK} + 17.22
I_{CC_DCDC} Typ.(VSCR_2) = 0.0646 × $f_{CPUCLK0}$ + 0.0223 × f_{ICLK} + 15.14
I_{CC_DCDC} Max.(VSCR_1, 105°C) = 0.0863 × $f_{CPUCLK0}$ + 0.0343 × f_{ICLK} + 237.02
I_{CC_DCDC} Max.(VSCR_1, 125°C) = 0.0935 × $f_{CPUCLK0}$ + 0.0359 × f_{ICLK} + 399.63
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0806 × $f_{CPUCLK0}$ + 0.0305 × f_{ICLK} + 214.33
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0897 × $f_{CPUCLK0}$ + 0.0343 × f_{ICLK} + 361.19
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.12 Supply Currents for Maximum Data Processing with the Peripheral Clock Supplies Stopped (External VDD Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current (maximum data processing with the peripheral clock supplies stopped)	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}*1	112	468	681	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		90	426	631		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		73	408	614		
	VCC current		I_{CC}	3.9	7.1	9.7		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- The clock-signal supplies to the peripheral functions are stopped.
- Power to the ESWM domain is turned off.
- The values given were measured with all output pins load-free and all input pull-up resistors disabled.
- The values given do not include the supply current for background operation (BGO).

Note 1. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.92 to 0.99 V) = 0.1181 × $f_{CPUCLK0}$ + 0.0469 × f_{ICLK} + 27.55
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.1153 × $f_{CPUCLK0}$ + 0.0405 × f_{ICLK} + 25.17
I_{DD} Max.(VCL = 0.92 to 0.99 V, 105°C) = 0.1192 × $f_{CPUCLK0}$ + 0.0469 × f_{ICLK} + 381.95
I_{DD} Max.(VCL = 0.92 to 0.99 V, 125°C) = 0.1148 × $f_{CPUCLK0}$ + 0.0437 × f_{ICLK} + 598.85
I_{DD} Max.(VCL = 0.87 to 0.99 V, 105°C) = 0.1192 × $f_{CPUCLK0}$ + 0.0448 × f_{ICLK} + 357.95
I_{DD} Max.(VCL = 0.87 to 0.99 V, 125°C) = 0.1182 × $f_{CPUCLK0}$ + 0.0449 × f_{ICLK} + 563.65
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.13 Supply Currents for while(1) Operation by the CPU with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Supply current (DCDC mode)	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC} *1	28	—	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		23	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		22	—		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		41	—		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		33	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		30	—		
Supply current (external VDD mode)	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD} *2	68	—	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		56	—		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		52	—		

Note: This table lists the values for current under the following operating condition.
- The program is placed in the LRAM, the cache is enabled, and the CPU executes a while(1) loop.

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0145 × $f_{CPUCLK0}$ + 0.0149 × f_{ICLK} + 14.35
I_{CC_DCDC} Typ.(VSCR_2) = 0.0113 × $f_{CPUCLK0}$ + 0.0109 × f_{ICLK} + 14.76
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0259 × $f_{CPUCLK0}$ + 0.0269 × f_{ICLK} + 17.25
I_{CC_DCDC} Typ.(VSCR_2) = 0.0236 × $f_{CPUCLK0}$ + 0.0218 × f_{ICLK} + 16.14
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.92 to 0.99 V) = 0.0448 × $f_{CPUCLK0}$ + 0.0469 × f_{ICLK} + 27.37
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0440 × $f_{CPUCLK0}$ + 0.0406 × f_{ICLK} + 25.19
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.14 Supply Currents for Coremark Execution with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off (DCDC Mode)

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Cache enabled	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC}	36	—	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		29	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		25	—		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		58	—		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		46	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		38	—		
Cache disabled, code executed in LRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC}	29	—	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		24	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		22	—		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		45	—		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		36	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		31	—		
Cache disabled, code executed in SRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC}	23	—	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		21	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		20	—		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		34	—		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		29	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		27	—		
Cache disabled, code executed in MRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC}	26	—	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		22	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		22	—		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		40	—		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		32	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		32	—		

Table 67.15 Supply Currents for Coremark Execution with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off (External VDD Mode)

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Cache enabled	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}	97	—	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		78	—		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		63	—		
Cache disabled, code executed in LRAM	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}	75	—	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		60	—		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		52	—		
Cache disabled, code executed in SRAM	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}	55	—	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		45	—		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		45	—		
Cache disabled, code executed in MRAM	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}	65	—	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		53	—		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		53	—		

Table 67.16 Supply Currents in CPU Sleep Mode with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Supply current (DCDC Mode)	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC} *1	22	227	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		19	205		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		19	205		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		31	416		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		26	375		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		26	375		
Supply current (External VDD Mode)	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}*2	52	618	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		43	582		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		44	582		

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0049 × $f_{CPUCLK0}$ + 0.0128 × f_{ICLK} + 15.37
I_{CC_DCDC} Typ.(VSCR_2) = 0.0051 × $f_{CPUCLK0}$ + 0.0109 × f_{ICLK} + 14.14
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ.(VSCR_1) = 0.0098 × $f_{CPUCLK0}$ + 0.0260 × f_{ICLK} + 17.72

I_{CC_DCDC} Typ.(VSCR_2) = 0.0096 × $f_{CPUCLK0}$ + 0.0221 × f_{ICLK} + 15.88

However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).

I_{DD} Typ.(VCL = 0.92 to 0.99 V) = 0.0176 × $f_{CPUCLK0}$ + 0.0471 × f_{ICLK} + 27.25

I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0175 × $f_{CPUCLK0}$ + 0.0409 × f_{ICLK} + 25.14

However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.17 Supply Currents in CPU Deep Sleep Mode with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Supply current (DCDC Mode)	VCC_DCDC ≥ 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{CC_DCDC} *1	20	186	mA	VSCR_1 (VCL = 0.95 V) VCC_DCDC = 3.3 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		17	168		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		18	169		
	VCC_DCDC < 2.5 V	Frequency setting 1 (CPUCLK0 = 600 MHz)		27	340		VSCR_1 (VCL = 0.95 V) VCC_DCDC = 1.8 V
		Frequency setting 2 (CPUCLK0 = 480 MHz)		22	307		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
		Frequency setting 3 (CPUCLK0 = 300 MHz)		23	308		
Supply current (External VDD Mode)	VCL = 0.92 to 0.99 V	Frequency setting 1 (CPUCLK0 = 600 MHz)	I_{DD}*2	43	520	mA	Voltage range 1 (VCL = 0.95 V)
	VCL = 0.87 to 0.99 V	Frequency setting 2 (CPUCLK0 = 480 MHz)		36	489		Voltage range 2 (VCL = 0.925 V)
		Frequency setting 3 (CPUCLK0 = 300 MHz)		38	491		

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).

[When VCC_DCDC = 3.3 V]

I_{CC_DCDC} Typ.(VSCR_1) = 0.0024 × $f_{CPUCLK0}$ + 0.0134 × f_{ICLK} + 14.33

I_{CC_DCDC} Typ.(VSCR_2) = 0.0029 × $f_{CPUCLK0}$ + 0.0122 × f_{ICLK} + 12.87

[When VCC_DCDC = 1.8 V]

I_{CC_DCDC} Typ.(VSCR_1) = 0.0049 × $f_{CPUCLK0}$ + 0.0265 × f_{ICLK} + 15.67

I_{CC_DCDC} Typ.(VSCR_2) = 0.0050 × $f_{CPUCLK0}$ + 0.0226 × f_{ICLK} + 13.98

However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).

I_{DD} Typ.(VCL = 0.92 to 0.99 V) = 0.0089 × $f_{CPUCLK0}$ + 0.0472 × f_{ICLK} + 23.68

I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0089 × $f_{CPUCLK0}$ + 0.0406 × f_{ICLK} + 22.00

However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.18 Increases in Supply Currents during BGO Programming of MRAM-OTP in the Case of Maximal Operation

Item		Symbol	Typ.	Max.	Unit	Test Conditions
Supply current	Normal speed write mode	I_{CC}	—	20	mA	VCC ≥ 1.62 V
		I_{DD}	—	0.5		
	High speed write mode 0	I_{CC}	—	25		VCC ≥ 2.5 V
		I_{DD}	—	5		
	High speed write mode 1	I_{CC}	—	80		VCC ≥ 3.0 V
		I_{DD}	—	0.5		

Note: The MWMCR register is used to select the mode for writing data to MRAM-OTP.

Table 67.19 Currents in Software Standby Mode 1 (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VCC current			I_{CC}	0.20	0.41	0.62	mA	VCC = 3.3 V
VCC_DCDC current	The SRAM and LRAM data are retained.*1, *2	VSCR_1 (VCL = 0.95 V)	I_{CC_DCDC}	10.5	86.3	135	mA	VCC_DCDC = 3.3 V
		VSCR_2 (VCL = 0.925 V)		9.76	79.2	124		
	The SRAM and LRAM data are not retained. *1, *2	VSCR_1 (VCL = 0.95 V)		10.3	85.7	134		
		VSCR_2 (VCL = 0.925 V)		9.63	78.7	123		

Note 1. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 2. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

Table 67.20 Currents in Software Standby Mode 2 (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VCC current	For SS2LP_0		I_{CC}	0.20	0.41	0.62	mA	VCC = 3.3 V
	For SS2LP_1			0.04	0.27	0.48		
SS2LP_0*1	SVSCR_1*2	The SRAM and LRAM data are retained.*3, *4	I_{CC_DCDC}	8.95	26.8	39.2	mA	VCC_DCDC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		8.88	26.0	37.8		
	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4		8.89	24.7	36.0		
		The SRAM and LRAM data are not retained.*3, *4		8.74	24.2	35.1		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		8.69	18.7	27.9		
		The SRAM and LRAM data are not retained.*3, *4		8.68	18.4	27.3		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		8.62	16.6	24.7		
		The SRAM and LRAM data are not retained.*3, *4		8.59	16.1	23.6		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		8.39	14.6	21.5		
		The SRAM and LRAM data are not retained.*3, *4		8.37	14.2	20.6		
SS2LP_1*1	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4	I_{CC_DCDC}	1.06	20.3	34.1	mA	VCC_DCDC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		0.96	19.8	33.0		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		0.69	14.9	25.5		
		The SRAM and LRAM data are not retained.*3, *4		0.65	14.6	24.8		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		0.56	13.0	22.6		
		The SRAM and LRAM data are not retained.*3, *4		0.52	12.4	21.3		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		0.46	11.2	19.6		
		The SRAM and LRAM data are not retained.*3, *4		0.43	10.7	18.5		

Note 1. Setting of the SS2LP[1:0] bits in the SSCR1 register: SS2LP_0 = default mode, SS2LP_1 = low power setting 1

Note 2. SVSCR_n (n = 1 to 5) is the target VDD voltage specified in the SVSCR register.
Note 3. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 4. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

67.4.4.2 Operating and Standby Currents for Product Part No. R5K574Nxy4xx (y = E, F)

Table 67.21 Supply Currents for Full Operation (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current (full operation	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC} *1	62	144	210	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		104	260	384		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
	VCC current		I_{CC}	10	13	15		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- The peripheral functions are supplied with clocks and kept as active as is possible.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ. = 0.0460 × $f_{CPUCLK0}$ + 0.0622 × f_{ICLK} + 29.28
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0456 × $f_{CPUCLK0}$ + 0.0702 × f_{ICLK} + 109.07
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0477 × $f_{CPUCLK0}$ + 0.0723 × f_{ICLK} + 173.67
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ. = 0.0818 × $f_{CPUCLK0}$ + 0.1136 × f_{ICLK} + 45.27
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0854 × $f_{CPUCLK0}$ + 0.1322 × f_{ICLK} + 194.42
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0890 × $f_{CPUCLK0}$ + 0.1352 × f_{ICLK} + 316.79
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.22 Supply Currents for Full Operation (External VDD Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current (full operation	VCL = 0.87 to 0.99 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{DD}*1	181	425	594	mA	Voltage range 2 (VCL = 0.925 V)
	VCC current		I_{CC}	10	13	15		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- The peripheral functions are supplied with clocks and kept as active as is possible.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ. = 0.1408 × $f_{CPUCLK0}$ + 0.1942 × f_{ICLK} + 80.70
I_{DD} Max.(VCL = 0.87 to 0.99 V, 105°C) = 0.1267 × $f_{CPUCLK0}$ + 0.1933 × f_{ICLK} + 329.00
I_{DD} Max.(VCL = 0.87 to 0.99 V, 125°C) = 0.1200 × $f_{CPUCLK0}$ + 0.1800 × f_{ICLK} + 504.00
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.23 Supply Currents for Full Operation (LDO Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current (full operation)	LDO efficiency improvement function is disabled*2	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC}*1	192	432	600	mA	VCC = 3.3 V
	LDO efficiency improvement function is enabled*2, *3	Frequency setting 3 (CPUCLK0 = 300 MHz)		175	352*4	477*4		
				230	487*4	666*4		VCC = 1.8 V

Note: This table lists the values for current under the following operating conditions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- The peripheral functions are supplied with clocks and kept as active as is possible.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{CC} can be estimated by using the following formulae (units: I_{CC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When LDO efficiency improvement function is disabled (LDOECR.EFUPE = 0)]
I_{CC} Typ. = 0.1431 × $f_{CPUCLK0}$ + 0.2051 × f_{ICLK} + 87.95
I_{CC} Max.(VSCR_2, 105°C) = 0.1333 × $f_{CPUCLK0}$ + 0.2000 × f_{ICLK} + 332.01
I_{CC} Max.(VSCR_2, 125°C) = 0.1400 × $f_{CPUCLK0}$ + 0.2000 × f_{ICLK} + 498.00
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)] (VCC = 3.3 V)
I_{CC} Typ. = (0.0856 × $f_{CPUCLK0}$ + 0.1260 × f_{ICLK} + 95.97) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 105°C) = (0.1133 × $f_{CPUCLK0}$ + 0.1333 × f_{ICLK} + 246.02) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 125°C) = (0.0867 × $f_{CPUCLK0}$ + 0.1400 × f_{ICLK} + 365.99) × (correction coefficients 1.1)
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)] (VCC = 1.8 V)
I_{CC} Typ. = (0.1456 × $f_{CPUCLK0}$ + 0.2042 × f_{ICLK} + 103.76) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 105°C) = (0.1267 × $f_{CPUCLK0}$ + 0.1867 × f_{ICLK} + 348.98) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 125°C) = (0.1000 × $f_{CPUCLK0}$ + 0.1733 × f_{ICLK} + 523.01) × (correction coefficients 1.1)
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. LDOECR.EFUPE = 0 (disabled) or LDOECR.EFUPE = 1 (enabled)

Note 3. When the LDO efficiency improvement function is enabled, the efficiency deteriorates and the supply current increases as the VCC voltage decreases. See Figure 67.6.

Note 4. The values are reference values.

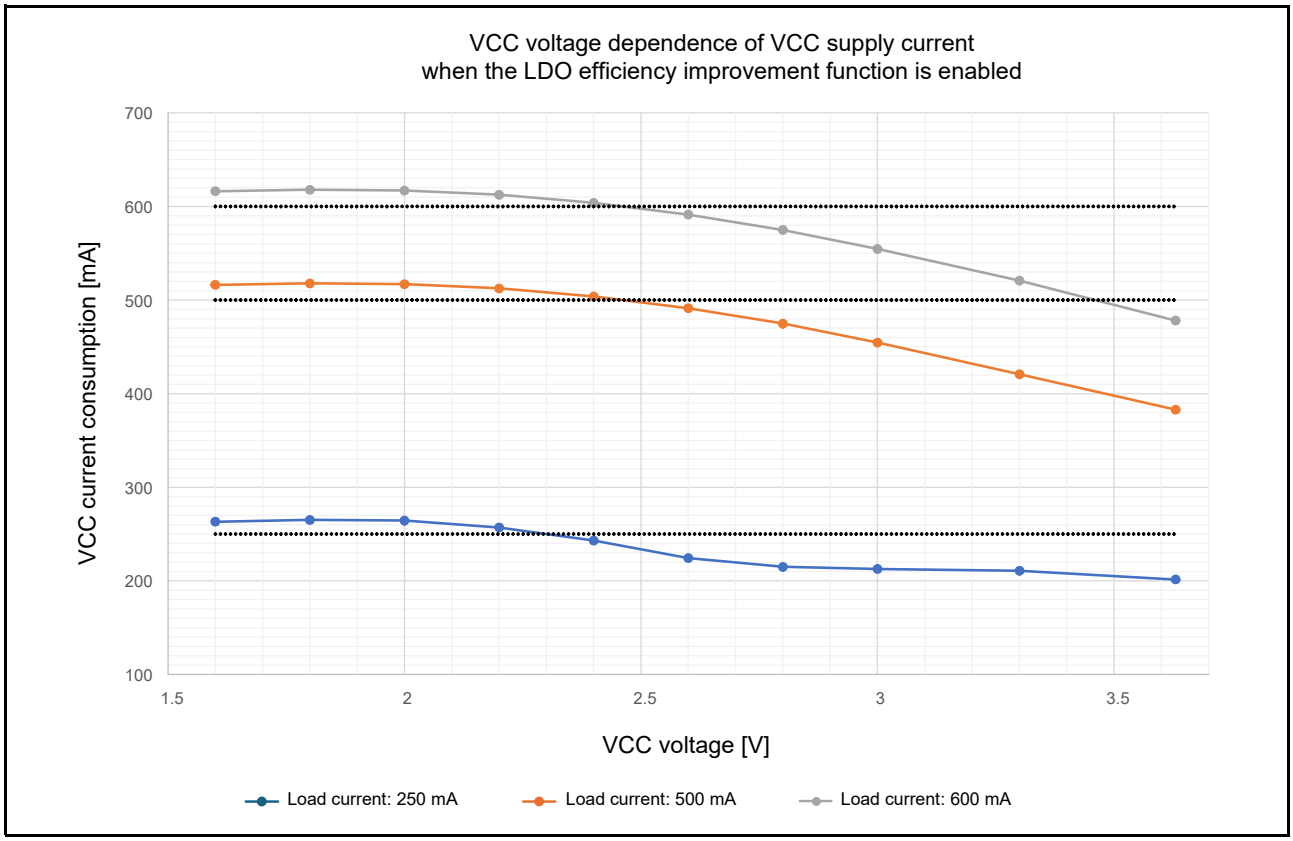


Figure 67.6 VCC Voltage Dependence of VCC Supply Current when the LDO Efficiency Improvement Function is Enabled

Table 67.24 Supply Currents for Maximum Data Processing with the Peripheral Clock Supplies Stopped (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC} *1	25	106	173	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		36	189	315		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
	VCC current		I_{CC}	3.9	6.2	8.3		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- The clock-signal supplies to the peripheral functions are stopped.
- Power to the ESWM domain is turned off.

Note 1. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ. = 0.0318 × $f_{CPUCLK0}$ + 0.0111 × f_{ICLK} + 12.00
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0210 × $f_{CPUCLK0}$ + 0.0070 × f_{ICLK} + 97.82
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0126 × $f_{CPUCLK0}$ + 0.0026 × f_{ICLK} + 168.35
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ. = 0.0612 × $f_{CPUCLK0}$ + 0.0197 × f_{ICLK} + 12.07
I_{CC_DCDC} Max.(VSCR_2, 105°C) = 0.0387 × $f_{CPUCLK0}$ + 0.0129 × f_{ICLK} + 174.02
I_{CC_DCDC} Max.(VSCR_2, 125°C) = 0.0243 × $f_{CPUCLK0}$ + 0.0048 × f_{ICLK} + 306.77
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.25 Supply Currents for Maximum Data Processing with the Peripheral Clock Supplies Stopped (External VDD Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current	VCL = 0.87 to 0.99 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{DD}*1	62.5	320	501	mA	Voltage range 2 (VCL = 0.925 V)
	VCC current		I_{CC}	3.9	6.2	8.3		VCC = 3.3 V

Note: This table lists the values for current under the following operating conditions.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- The clock-signal supplies to the peripheral functions are stopped.
- Power to the ESWM domain is turned off.
- The values given were measured with all output pins load-free and all input pull-up resistors disabled.
- The values given do not include the supply current for background operation (BGO).

Note 1. Value for I_{DD} can be estimated by using the following formulae (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ. = 0.1124 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 18.30
I_{DD} Max.(VCL = 0.87 to 0.99 V, 105°C) = 0.0600 × $f_{CPUCLK0}$ + 0.0200 × f_{ICLK} + 296.00
I_{DD} Max.(VCL = 0.87 to 0.99 V, 125°C) = 0.0333 × $f_{CPUCLK0}$ + 0.0067 × f_{ICLK} + 489.00
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.26 Supply Currents for Maximum Data Processing with the Peripheral Clock Supplies Stopped (LDO Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Supply current	LDO efficiency improvement function is disabled*2	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC}*1	68.1	323	503	mA	VCC = 3.3 V
	LDO efficiency improvement function is enabled*2	Frequency setting 3 (CPUCLK0 = 300 MHz)		68.0	256*3	424*3		
				68.2	373*3	554*3		VCC = 1.8 V

Note: This table lists the values for current under the following operating conditions.
- The internal memory (SRAM and MRAM) is configured with wait settings that suit the operating frequency and kept as active as is possible through data transfer by the DMAC.
- The peripheral functions are supplied with clocks and kept as active as is possible.
- Unless otherwise specified, the clocks are set to the states that maximize the supply currents within the limits imposed by the specifications.
- The program is placed in the LRAM, the cache is enabled, and the CPU repeatedly executes multiply-and-accumulate instructions.
- Power to the ESWM domain is turned on.

Note 1. Value for I_{CC} can be estimated by using the following formulae (units: I_{CC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When LDO efficiency improvement function is disabled (LDOECR.EFUPE = 0)]
I_{CC} Typ. = 0.1115 × $f_{CPUCLK0}$ + 0.0349 × f_{ICLK} + 24.16
I_{CC} Max.(VSCR_2, 105°C) = 0.0800 × $f_{CPUCLK0}$ + 0.0333 × f_{ICLK} + 289.01
I_{CC} Max.(VSCR_2, 125°C) = 0.0800 × $f_{CPUCLK0}$ + 0.0400 × f_{ICLK} + 467.00
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)] (VCC = 3.3 V)
I_{CC} Typ. = 0.1115 × $f_{CPUCLK0}$ + 0.0349 × f_{ICLK} + 24.08
I_{CC} Max.(VSCR_2, 105°C) = (0.0400 × $f_{CPUCLK0}$ + 0.0200 × f_{ICLK} + 215.00) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 125°C) = (0.0733 × $f_{CPUCLK0}$ + 0.2200 × f_{ICLK} + 297.01) × (correction coefficients 1.1)
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)] (VCC = 1.8 V)
I_{CC} Typ. = 0.1115 × $f_{CPUCLK0}$ + 0.0349 × f_{ICLK} + 24.25
I_{CC} Max.(VSCR_2, 105°C) = (0.0327 × $f_{CPUCLK0}$ + 0.1029 × f_{ICLK} + 298.32) × (correction coefficients 1.1)
I_{CC} Max.(VSCR_2, 125°C) = (0.0333 × $f_{CPUCLK0}$ + 0.0200 × f_{ICLK} + 488.01) × (correction coefficients 1.1)
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 2. LDOECR.EFUPE = 0 (disabled) or LDOECR.EFUPE = 1 (enabled)

Note 3. The values are reference values.

Table 67.27 Supply Currents for while(1) Operation by the CPU with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item			Symbol	Typ.	Max.	Unit	Test Conditions
DCDC mode	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC}*2	16	—	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		24	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
External VDD mode	VCL = 0.87 to 0.99 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{DD}*3	41	—	mA	Voltage range 2 (VCL = 0.925 V)
LDO mode	LDO efficiency improvement function is disabled*1	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC}*4	47	—	mA	VCC = 3.3 V
	LDO efficiency improvement function is enabled*1	Frequency setting 3 (CPUCLK0 = 300 MHz)		47	—		

Note 1. LDOECR.EFUPE = 0 (disabled) or LDOECR.EFUPE = 1 (enabled)

Note 2. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ.(VSCR_2) = 0.0069 × $f_{CPUCLK0}$ + 0.0122 × f_{ICLK} + 10.71
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ.(VSCR_2) = 0.0213 × $f_{CPUCLK0}$ + 0.0200 × f_{ICLK} + 11.63
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 3. Value for I_{DD} can be estimated by using the following formula (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0420 × $f_{CPUCLK0}$ + 0.0353 × f_{ICLK} + 18.16
However, note that the above formula is only for use in estimation and values from it do not guarantee the actual supply currents.

Note 4. Value for I_{CC} can be estimated by using the following formulae (units: I_{CC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When LDO efficiency improvement function is disabled (LDOECR.EFUPE = 0)]
I_{CC} Typ. = 0.0425 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 24.05
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)]
I_{CC} Typ. = 0.0423 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 23.78
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Table 67.28 Supply Currents for Coremark Execution with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off (DCDC Mode)

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Cache enabled	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC}	23	—	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		34	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
Cache disabled, code executed in LRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC}	20	—	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		27	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
Cache disabled, code executed in SRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC}	18	—	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		23	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
Cache disabled, code executed in MRAM	VCC_DCDC ≥ 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{CC_DCDC}	20	—	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V	Frequency setting 3 (CPUCLK0 = 300 MHz)		28	—		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V

Table 67.29 Supply Currents for Coremark Execution with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off (External VDD Mode)

Item			Symbol	Typ.	Max.	Unit	Test Conditions
Cache enabled	VCL = 0.87 to 0.99 V	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{DD}	56	—	mA	Voltage range 2 (VCL = 0.925 V)
Cache disabled, code executed in LRAM			I_{DD}	45	—	mA	
Cache disabled, code executed in SRAM			I_{DD}	38	—	mA	
Cache disabled, code executed in MRAM			I_{DD}	45	—	mA	

Table 67.30 Supply Currents for Coremark Execution with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off (LDO Mode)

Item		Symbol	Typ.	Max.	Unit	Test Conditions
Cache enabled	Frequency setting 3 (CPUCLK0 = 300 MHz)	I_{DD}	62	—	mA	VCC = 3.3 V
Cache disabled, code executed in LRAM		I_{DD}	51	—	mA	
Cache disabled, code executed in SRAM		I_{DD}	44	—	mA	
Cache disabled, code executed in MRAM		I_{DD}	58	—	mA	

Table 67.31 Supply Currents in CPU Sleep Mode with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item		Symbol	Typ.	Max.	Unit	Test Conditions
DCDC mode	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *2	17	188	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V		21	344		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
External VDD mode	VCL = 0.87 to 0.99 V	I_{DD}*3	34	540	mA	Voltage range 2 (VCL = 0.925 V)
LDO mode	LDO efficiency improvement function is disabled*1	I_{CC}*4	40	542	mA	VCC = 3.3 V
	LDO efficiency improvement function is enabled*1		40	406*5		
			40	595*5		VCC = 1.8 V

Note 1. LDOECR.EFUPE = 0 (disabled) or LDOECR.EFUPE = 1 (enabled)

Note 2. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ. = 0.0028 × $f_{CPUCLK0}$ + 0.0096 × f_{ICLK} + 12.80
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ. = 0.0074 × $f_{CPUCLK0}$ + 0.0184 × f_{ICLK} + 12.82
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 3. Value for I_{DD} can be estimated by using the following formula (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0169 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 18.18
However, note that the above formula is only for use in estimation and values from it do not guarantee the actual supply currents.

Note 4. Value for I_{CC} can be estimated by using the following formulae (units: I_{CC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When LDO efficiency improvement function is disabled (LDOECR.EFUPE = 0)]
I_{CC} Typ. = 0.0171 × $f_{CPUCLK0}$ + 0.0352 × f_{ICLK} + 24.02
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)]
I_{CC} Typ. = 0.0170 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 24.50
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 5. The values are reference values.

Table 67.32 Supply Currents in CPU Deep Sleep Mode with the Peripheral Clock Supplies Stopped and the ESWM Domain Turned Off

Item		Symbol	Typ.	Max.	Unit	Test Conditions
DCDC mode	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *2	15	154	mA	VSCR_2 (VCL = 0.925 V) VCC_DCDC = 3.3 V
	VCC_DCDC < 2.5 V		19	279		VSCR_2 (VCL = 0.925 V) VCC_DCDC = 1.8 V
External VDD mode	VCL = 0.87 to 0.99 V	I_{DD} *3	30	450	mA	Voltage range 2 (VCL = 0.925 V)
LDO mode	LDO efficiency improvement function is disabled*1	I_{CC} *4	36	453	mA	VCC = 3.3 V
	LDO efficiency improvement function is enabled*1		36	347*5		
			36	506*5		VCC = 1.8 V

Note 1. LDOECR.EFUPE = 0 (disabled) or LDOECR.EFUPE = 1 (enabled)

Note 2. Value for I_{CC_DCDC} can be estimated by using the following formulae (units: I_{CC_DCDC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When VCC_DCDC = 3.3 V]
I_{CC_DCDC} Typ.(VSCR_2) = 0.0016 × $f_{CPUCLK0}$ + 0.0062 × f_{ICLK} + 13.10
[When VCC_DCDC = 1.8 V]
I_{CC_DCDC} Typ.(VSCR_2) = 0.0039 × $f_{CPUCLK0}$ + 0.0156 × f_{ICLK} + 12.82
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 3. Value for I_{DD} can be estimated by using the following formula (units: I_{DD} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
I_{DD} Typ.(VCL = 0.87 to 0.99 V) = 0.0085 × $f_{CPUCLK0}$ + 0.0350 × f_{ICLK} + 17.37
However, note that the above formula is only for use in estimation and values from it do not guarantee the actual supply currents.

Note 4. Value for I_{CC} can be estimated by using the following formulae (units: I_{CC} = mA, $f_{CPUCLK0}$ = MHz, f_{ICLK} = MHz).
[When LDO efficiency improvement function is disabled (LDOECR.EFUPE = 0)]
I_{CC} Typ. = 0.0086 × $f_{CPUCLK0}$ + 0.0351 × f_{ICLK} + 23.25
[When LDO efficiency improvement function is enabled (LDOECR.EFUPE = 1)]
I_{CC} Typ. = 0.0085 × $f_{CPUCLK0}$ + 0.0352 × f_{ICLK} + 22.89
However, note that the above formulae are only for use in estimation and values from them do not guarantee the actual supply currents.

Note 5. The values are reference values.

Table 67.33 Increases in Supply Currents during BGO Programming of MRAM-OTP in the Case of Maximal Operation

Item		Symbol	Typ.	Max.	Unit	Test Conditions
Supply current	Normal speed write mode	I_{CC}	—	20	mA	VCC ≥ 1.62 V
		I_{DD}	—	0.5		
	High speed write mode 0	I_{CC}	—	25		VCC ≥ 2.5 V
		I_{DD}	—	5		
	High speed write mode 1	I_{CC}	—	80		VCC ≥ 3.0 V
		I_{DD}	—	0.5		

Note: The MWMCR register is used to select the mode for writing data to MRAM-OTP.

Table 67.34 Currents in Software Standby Mode 1 (DCDC Mode)

Item		Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VCC current		I_{CC}	0.18	0.39	0.62	mA	VCC = 3.3 V
VCC_DCDC current	The SRAM and LRAM data are retained. *1, *2	I_{CC_DCDC}	9.76	77.4	122	mA	VCC_DCDC = 3.3 V
	The SRAM and LRAM data are not retained.*1, *2		9.63	76.7	121		

Note 1. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 2. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

Table 67.35 Currents in Software Standby Mode 1 (LDO Mode)

Item		Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VCC current	The SRAM and LRAM data are retained.*1, *2	I_{CC}	9.94	223	353	mA	VCC = 3.3 V
	The SRAM and LRAM data are not retained. *1, *2		9.81	223	353		

Note: In software standby mode 1, disable the LDO efficiency improvement function (LDOECR.EFUPE = 0).
Note 1. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 2. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

Table 67.36 Currents in Software Standby Mode 2 (DCDC Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VCC current	For SS2LP_0		I_{CC}	0.18	0.39	0.62	mA	VCC = 3.3 V
	For SS2LP_1			0.04	0.26	0.49		
SS2LP_0*1	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4	I_{CC_DCDC}	8.34	23.2	34.1	mA	VCC_DCDC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		8.25	22.8	33.3		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		7.99	17.8	26.2		
		The SRAM and LRAM data are not retained.*3, *4		7.94	17.6	25.7		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		7.87	15.6	22.9		
		The SRAM and LRAM data are not retained.*3, *4		7.84	15.3	22.0		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		7.79	13.6	19.8		
		The SRAM and LRAM data are not retained.*3, *4		7.77	13.4	19.1		
SS2LP_1*1	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4	I_{CC_DCDC}	0.84	18.7	31.5	mA	VCC_DCDC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		0.73	18.3	30.5		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		0.53	13.6	23.3		
		The SRAM and LRAM data are not retained.*3, *4		0.48	13.4	22.7		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		0.42	11.7	20.3		
		The SRAM and LRAM data are not retained.*3, *4		0.39	11.3	19.4		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		0.34	10.0	17.6		
		The SRAM and LRAM data are not retained.*3, *4		0.32	9.70	16.8		

Note 1. Setting of the SS2LP[1:0] bits in the SSCR1 register: SS2LP_0 = default mode, SS2LP_1 = low power setting 1
Note 2. SVSCR_n (n = 2 to 5) is the target VDD voltage specified in the SVSCR register.
Note 3. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 4. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

Table 67.37 Currents in Software Standby Mode 2 (LDO Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
SS2LP_0*1	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4	I_{CC}	4.46	57.2	95.9	mA	VCC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		4.11	56.3	94.0		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		3.66	47.2	80.3		
		The SRAM and LRAM data are not retained.*3, *4		3.49	46.5	78.7		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		3.39	42.7	73.8		
		The SRAM and LRAM data are not retained.*3, *4		3.26	41.2	70.6		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		3.14	38.6	67.3		
		The SRAM and LRAM data are not retained.*3, *4		3.05	37.3	64.4		
SS2LP_1*1	SVSCR_2*2	The SRAM and LRAM data are retained.*3, *4	I_{CC}	2.85	56.4	94.9	mA	VCC = 3.3 V
		The SRAM and LRAM data are not retained.*3, *4		2.51	55.6	93.1		
	SVSCR_3*2	The SRAM and LRAM data are retained.*3, *4		2.09	46.3	79.3		
		The SRAM and LRAM data are not retained.*3, *4		1.92	45.6	77.8		
	SVSCR_4*2	The SRAM and LRAM data are retained.*3, *4		1.78	41.6	72.6		
		The SRAM and LRAM data are not retained.*3, *4		1.65	40.4	69.6		
	SVSCR_5*2	The SRAM and LRAM data are retained.*3, *4		1.54	37.5	66.1		
		The SRAM and LRAM data are not retained.*3, *4		1.45	36.4	63.3		

Note: In software standby mode 2, disable the LDO efficiency improvement function (LDOECR.EFUPE = 0).
Note 1. Setting of the SS2LP[1:0] bits in the SSCR1 register: SS2LP_0 = default mode, SS2LP_1 = low power setting 1
Note 2. SVSCR_n (n = 2 to 5) is the target VDD voltage specified in the SVSCR register.
Note 3. When the PDRAMSCR0 register is set to 7FFFh, data in the SRAM are retained; when the register is set to 7000h, data in the SRAM are not retained.
Note 4. When the PDRAMSCR2.RKEEP0 bit is set to 1, data in the LRAM are retained; when the bit is set to 0, data in the LRAM are not retained.

67.4.4.3 Currents in Deep Software Standby Mode

Table 67.38 Currents in Deep Software Standby Mode (DCDC Mode or LDO Mode)

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
DSTBY1	VCC_DCDC current		I_{CC_DCDC}	0.16	1.24	2.45	µA	Condition for the Max. values: VCC = VCC_DCDC = 3.63 V Condition for the Typ. values: VCC = VCC_DCDC = 3.3 V
	VCC current		I_{CC}	7.10	185	332		
		Increase in the current when PVDn or VBATT power switching is enabled*1		See Table 67.40				
		Increase in the current when the LOCO is in use*2		3.42	4.50	5.20		
		Increase in the current during SOSC and RTC operation		See Table 67.41				
		Increase in the current during IWDT operation		0.01	0.02	0.02		
DSTBY2	VCC_DCDC current		I_{CC_DCDC}	0.16	1.24	2.45	µA	
	VCC current		I_{CC}	4.90	108	195		
		Increase in the current when PVDn or VBATT power switching is enabled*1		See Table 67.40				
		Increase in the current during SOSC and RTC operation		See Table 67.41				
DSTBY3	VCC_DCDC current		I_{CC_DCDC}	0.16	1.24	2.45	µA	
	VCC current		I_{CC}	4.40	106	193		
		Increase in the current during SOSC and RTC operation		See Table 67.41				
Increase in the current when the LDO regulator is in use			I_{CC}	0.10	7.42	26.1	µA	VCC = 3.3 V

DSTBY1: Deep software standby mode 1
DSTBY2: Deep software standby mode 2
DSTBY3: Deep software standby mode 3
PVDn: Programmable voltage detection module (n = 0, 1, 2, 4, 5)
LOCO: Low-speed on-chip oscillator
SOSC: Sub-clock oscillator
RTC: Realtime clock
Note 1. When the VBATTCR1.PSWDIS bit = 0 (initial value), VBATT power switching is enabled.
Note 2. The maximum value for this item is the result measured at VCC = VCC_DCDC = 3.3 V.

Table 67.39 RTC Operating Current with VCC Off (VCC = 0 V) and Only the RTC Operating through the Battery Backup Function

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
VBATT supply current	When a crystal resonator is in use	Low-power mode 2 (7 pF) *1	I_{VBAT}	0.57	—	—	µA	VBATT = 1.8 V
				0.98	—	—		VBATT = 3.3 V
		Low-power mode 1 (9 pF) *1		0.67	—	—		VBATT = 1.8 V
				1.08	—	—		VBATT = 3.3 V
		Standard mode (12.5 pF) *1		0.92	—	—		VBATT = 1.8 V
				1.34	—	—		VBATT = 3.3 V
	When the external sub-clock input pin (EXCIN) is in use			0.23	—	—		VBATT = 1.8 V
				0.57	—	—		VBATT = 3.3 V
	Increased in the current per channel when the RTCICn (n = 0 to 2) input is in use			0.01	—	—		VBATT = 1.8 V
				0.01	—	—		VBATT = 3.3 V

Note 1. This mode is selected by the SOMCR.SODRV[1:0] bits.

Table 67.40 Increases in the Current when the PVDn or VBATT Function is Enabled in DSTBY1 and DSTBY2

Item		Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Increase in VCC current	Current in the common circuit of PVDn in DSTBY1*1	I_{CC}	3.60	—	—	µA	VCC = VCC_DCDC = 3.3 V
	Current in the common circuit of PVDn in DSTBY2*1		3.60	—	—		
	PVD1 is enabled		1.70	—	—		
	PVD2 is enabled		1.70	—	—		
	PVD4 is enabled		1.70	—	—		
	PVD5 is enabled		1.70	—	—		
	When VBATT power switching is enabled*2		1.70	—	—		

n = 0, 1, 2, 4, 5

Note 1. When PVDn is enabled and the low power consumption function of PVD0 is disabled (OFS1.DSBYLPSEL = 1)

Note 2. When the VBATTCR1.PSWDIS bit = 0 and PVD0 is not in use for VBATT voltage detection (OFS1.DSBYLPSEL = 1)

Table 67.41 Increase in the Current during Sub-Clock Oscillator and RTC Operation in DSTBY Mode

Item			Symbol	Typ.	Max. (Tj = 105°C)	Max. (Tj = 125°C)	Unit	Test Conditions
Increase in VCC current	When a crystal resonator is in use	Low-power mode 2 (7 pF) *1	I_{CC}	0.47	—	—	µA	VCC = VCC_DCDC = 3.3 V
		Low-power mode 1 (9 pF) *1		0.57	—	—		
		Standard mode (12.5 pF) *1		0.83	—	—		
	During RTC Operation			0.05	—	—		

Note 1. This mode is selected by the SOMCR.SODRV[1:0] bits.

Table 67.42 Inrush Currents

Item		Product Part No. (R5K574Nxy4zz)	DPSBYCR Register	Symbol	Typ.	Max.	Unit	Test Conditions
Inrush current of VCC_DCDC	Inrush current on power-on	y = A, B, C, D	—	I_{RUSH}	—	1200	mA	—
		y = E, F	—		—	1050		
	Inrush current on return from deep software standby mode	y = A, B, C, D	DCSSMODE[1:0] = 01b		—	1150		
			DCSSMODE[1:0] = 10b		—	1050		
			DCSSMODE[1:0] = 11b		—	1050		
		y = E, F	DCSSMODE[1:0] = 01b		—	1000		
			DCSSMODE[1:0] = 10b		—	900		
			DCSSMODE[1:0] = 11b		—	800		
Inrush current of VCC	Inrush current on power-on	zz = FP (LQFP100)		I_{RUSH}	—	495	mA	—
		zz = FB (LQFP144)			—	490		
		zz = FC (LQFP176)			—	500		
		zz = BG (BGA176)			—	500		
		zz = BD (BGA224)			—	500		
		zz = AE (BGA320)			—	511		*1
	Inrush current on return from deep software standby mode	zz = FP (LQFP100)	LDOSSMODE[1:0] = 01b		—	615		—
		zz = FB (LQFP144)			—	602		
		zz = FC (LQFP176)			—	626		
		zz = BG (BGA176)			—	626		
		zz = BD (BGA224)			—	626		
		zz = AE (BGA320)			—	650		*1
		zz = FP (LQFP100)	LDOSSMODE[1:0] = 10b		—	541		—
		zz = FB (LQFP144)			—	532		
		zz = FC (LQFP176)			—	548		
		zz = BG (BGA176)			—	548		
		zz = BD (BGA224)			—	548		
		zz = AE (BGA320)			—	563		*1
		zz = FP (LQFP100)	LDOSSMODE[1:0] = 11b		—	495		—
		zz = FB (LQFP144)			—	490		
		zz = FC (LQFP176)			—	500		
		zz = BG (BGA176)			—	500		
		zz = BD (BGA224)			—	500		
		zz = AE (BGA320)			—	511		*1

Note: The values are reference values.

Note 1. The VCL pins must be connected to the VSS pins via multilayer ceramic capacitors and no more than ten multilayer ceramic capacitors should be connected in LDO mode.
The values for the BGA320 package are for the case with ten multilayer ceramic capacitors connected.

67.4.4.4 Other Operating Currents

Table 67.43 Main Clock Oscillator Operating Current

Item		Symbol	Typ.	Max.	Unit
Main clock oscillator operating current	MOMCR.MODRV0[2:0] = 000b	I_{CC}	0.65	—	mA
	MOMCR.MODRV0[2:0] = 011b		0.76	—	
	MOMCR.MODRV0[2:0] = 101b		0.88	—	

Table 67.44 Analog Operating Currents

Item								Symbol	Typ.	Max.	Unit
	A/D Unit 0	S&H (3ch.)	A/D Unit 1	A/D Unit 2	D/A*1 AMP ON	D/A*2 AMP OFF	TEMPS				
Analog power supply current	Converting	Sleep	Sleep	Sleep	Stopped	Stopped	Stopped	AI_{CC}	2.4	3.2	mA
	Converting	Converting	Sleep	Sleep	Stopped	Stopped	Stopped		6.6	8.7	
	Waiting	Sleep	Sleep	Sleep	Stopped	Stopped	Stopped		1.6	2.0	
	Sleep	Sleep	Converting	Sleep	Stopped	Stopped	Stopped		2.4	3.2	
	Sleep	Sleep	Waiting	Sleep	Stopped	Stopped	Stopped		1.6	2.0	
	Sleep	Sleep	Sleep	Converting	Stopped	Stopped	Stopped		2.4	3.2	
	Sleep	Sleep	Sleep	Waiting	Stopped	Stopped	Stopped		1.6	2.0	
	Stopped	Stopped	Stopped	Stopped	Converting	Stopped	Stopped		0.5	1.5	
	Stopped	Stopped	Stopped	Stopped	Stopped	Converting	Stopped		0.1	0.2	
	Stopped	Stopped	Stopped	Stopped	Stopped	Stopped	Converting		0.1	0.2	
	Standby mode								0.3	7.3	µA
Reference power supply current	A/D conversion is in progress in all ADC16H units.			SAR mode				AI_{REFH0}	220	370	µA
				Oversampling mode or hybrid mode					580	880	
	All ADC16H units waiting for the start of A/D conversion								8.2	14	
	All ADC16H units in standby mode								0.1	0.5	

Stopped: The module is stopped.
Waiting: The module is operating but not performing A/D conversion.
Sleep: The module is in sleep mode. (ADUSLPCRn.ADUSLPn = 1 (n = 0, 1, or 2))
S&H: Channel-dedicated sample-and-hold circuit
Note 1. The load resistance is 5 kΩ when the amplifier is on.
Note 2. The load resistance is 2 MΩ when the amplifier is off.

Table 67.45 USB Operating Currents

Item				Symbol	Typ.	Max.	Unit
USB operating current	Low-speed operation	USBf		I_{CCUSB_LS}	2.90	4.00	mA
		USBHS	Classic-only mode is disabled	$I_{CCUSBHS_LS}$	11.5	14.6	
			Classic-only mode is enabled		5.04	6.80	
	Full-speed operation	USBf		I_{CCUSB_FS}	4.00	4.70	
		USBHS	Classic-only mode is disabled	$I_{CCUSBHS_FS}$	12.5	14.7	
			Classic-only mode is enabled		5.98	6.90	
	High-speed operation	USBHS		$I_{CCUSBHS_HS}$	45.7	55.3	
	Standby mode	USBHS		$I_{CCUSBHS_SBY}$	0.89	11.4	µA

Note: The power supplies to the USBHS are set as VCC_USBHS = AVCC_USBHS.
Note: Classic-only mode is disabled when the PHYSET.HSEB bit is 0 and enabled when the PHYSET.HSEB bit is 1.

67.4.4.5 Permissible Output Currents

Table 67.46 Permissible Output Currents (1/6)

Item				Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (average value per pin)	I3C*1	I3C protocol mode*5	LVO0E = 0	I_{OL}	—	—	20	mA
			LVO0E = 1		—	—	25	
		I2C (Sm)			—	—	25	
		I2C (Fm)			—	—	25	
		I2C (Fm+)			—	—	25	
		I2C (Hs)			—	—	25	
	RIICHS*2	I2C (Sm)			—	—	25	
		I2C (Fm)			—	—	25	
		I2C (Fm+)			—	—	25	
		I2C (Hs)			—	—	25	
	Other than above*3, *4	DRV1			—	—	7.5	
		DRV2			—	—	15	
		DRV3			—	—	20	
		DRV4			—	—	25	
		DRV5			—	—	25	
		DRV6			—	—	25	
		DRV7			—	—	25	
		DRV8			—	—	25	
Permissible output low current (max. value per pin)	I3C*1	I3C protocol mode*5	LVO0E = 0	I_{OL}	—	—	40	mA
			LVO0E = 1		—	—	50	
		I2C (Sm)			—	—	50	
		I2C (Fm)			—	—	50	
		I2C (Fm+)			—	—	50	
		I2C (Hs)			—	—	50	
	RIICHS*2	I2C (Sm)			—	—	50	
		I2C (Fm)			—	—	50	
		I2C (Fm+)			—	—	50	
		I2C (Hs)			—	—	50	
	Other than above*3, *4	DRV1			—	—	15	
		DRV2			—	—	30	
		DRV3			—	—	40	
		DRV4			—	—	50	
		DRV5			—	—	50	
		DRV6			—	—	50	
		DRV7			—	—	50	
		DRV8			—	—	50	
Permissible output low current (total)	VCC IO (LQFP100)	P001 to P007, P306, P307, P310 to P312, P315 (Total of 13 pins)		ΣI_{OL}	—	—	80	mA
		P101 to P107, P200, P201, P300 to P304 (Total of 14 pins)			—	—	80	
		PA04 to PA07, PB00 to PB06, PC00 to PC06 (Total of 18 pins)			—	—	80	
		PA00 to PA03, PE00 to PE06 (Total of 11 pins)			—	—	80	

Table 67.46 Permissible Output Currents (2/6)

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (total)	VCC IO (LQFP144)	P000 to P007, P306, P307, P310 to P312, P315 (Total of 14 pins)	ΣI_{OL}	—	—	80	mA
		P202 to P207, P302 to P304 (Total of 9 pins)		—	—	80	
		P100 to P107, P200, P201, P300, P301 (Total of 12 pins)		—	—	80	
		P800 to P803, PC00 to PC07 (Total of 12 pins)		—	—	80	
		PB00 to PB07 (Total of 8 pins)		—	—	80	
		PA00 to PA07 (Total of 8 pins)		—	—	80	
		P700 to P702, PE00 to PE07 (Total of 11 pins)		—	—	80	
	VCC IO (LQFP176, BGA176)	P000 to P007, P306, P307, P310 to P312, P315 (Total of 14 pins)	ΣI_{OL}	—	—	80	mA
		P202 to P207, P302 to P304 (Total of 9 pins)		—	—	80	
		P100 to P107, P200, P201, P300, P301, P804, P805 (Total of 14 pins)		—	—	80	
		P500 to P503, P800 to P803, PC00 to PC07 (Total of 16 pins)		—	—	80	
		PA04 to PA07, PB00 to PB07, PG00 to PG02 (Total of 15 pins)		—	—	80	
		P700 to P707, PA00 to PA03 (Total of 12 pins)		—	—	80	
		PE00 to PE07 (Total of 8 pins)		—	—	80	
	VCC IO (BGA224)	P000 to P007, P214, P215 (Total of 10 pins)	ΣI_{OL}	—	—	80	mA
		P306, P307, P310 to P312, P315, P208 to P213 (Total of 12 pins)		—	—	80	
		P112 to P115, P202 to P207, PF02 to PF05 (Total of 14 pins)		—	—	80	
		P108 to P111, P300 to P304, P804 to P807 (Total of 13 pins)		—	—	80	
		P100 to P107, P200, P201, P504 to P507 (Total of 14 pins)		—	—	80	
		P500 to P503, P800 to P803, PC05 to PC07 (Total of 11 pins)		—	—	80	
		PB04 to PB07, PC00 to PC04 (Total of 9 pins)		—	—	80	
		PB00 to PB03, PG04 to PG07 (Total of 8 pins)		—	—	80	
		PA04 to PA07, PG00 to PG03 (Total of 8 pins)		—	—	80	
		P700 to P707, PA00 to PA03 (Total of 12 pins)		—	—	80	
		PE00 to PE07, PF06 to PF09 (Total of 12 pins)		—	—	80	

Table 67.46 Permissible Output Currents (3/6)

Item				Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (total)	VCC IO (BGA320)	P000 to P015, P214, P215 (Total of 18 pins)		ΣI_{OL}	—	—	80	mA
		P306, P307, P310 to P312, P315, P208 to P213 (Total of 12 pins)			—	—	80	
		P112 to P115, P202 to P207, PF02 to PF05 (Total of 14 pins)			—	—	80	
		P108 to P111, P300 to P304, P804 to P807 (Total of 13 pins)			—	—	80	
		P100 to P107, P200, P201, P504 to P507 (Total of 14 pins)			—	—	80	
		P500 to P503, P800 to P803, PC05 to PC07(Total of 11 pins)			—	—	80	
		P512 to P515, PB04 to PB07, PC00 to PC04 (Total of 13 pins)			—	—	80	
		P510, P511, PB00 to PB03, PG04 to PG07 (Total of 10 pins)			—	—	80	
		P508, P509, PA04 to PA07, PG00 to PG03 (Total of 10 pins)			—	—	80	
		P704 to P707, PA00 to PA03, PG12 to PG15 (Total of 12 pins)			—	—	80	
		P700 to P703, P712 to P715, PG08 to PG11 (Total of 12 pins)			—	—	80	
		P708 to P711, PE00 to PE07, PF06 to PF09 (Total of 16 pins)			—	—	80	
	VCC2 IO	P600 to P607, P900 to P907, PD00 to PD07 (Total of 24 pins)		ΣI_{OL_VCC2}	—	—	80	mA
	AVCC0 IO	P400 to P415 (Total of 16 pins)		ΣI_{OL_AVCC0}	—	—	33	mA
	VCC_USB IO	PF00, PF01 (Total of 2 pins)*6		ΣI_{OL_VCCUSB}	—	—	33	mA
Permissible output high current (average value per pin)	I3C*1	I3C protocol mode*5, *7	LVO0E = 0	I_{OH}	—	—	–20	mA
			LVO0E = 1		—	—	–25	
	Other than above*3, *4	DRV1			—	—	–7.5	
		DRV2			—	—	–15	
		DRV3			—	—	–20	
		DRV4			—	—	–25	
		DRV5			—	—	–25	
		DRV6			—	—	–25	
		DRV7			—	—	–25	
		DRV8			—	—	–25	

Table 67.46 Permissible Output Currents (4/6)

Item				Symbol	Min.	Typ.	Max.	Unit
Permissible output high current (max. value per pin)	I3C*1	I3C protocol mode*5, *7	LVO0E = 0	I_{OH}	—	—	–40	mA
			LVO0E = 1		—	—	–50	
	Other than above*3, *4	DRV1			—	—	–15	
		DRV2			—	—	–30	
		DRV3			—	—	–40	
		DRV4			—	—	–50	
		DRV5			—	—	–50	
		DRV6			—	—	–50	
		DRV7			—	—	–50	
		DRV8			—	—	–50	
Permissible output high current (total)	VCC IO (LQFP100)	P001 to P007, P306, P307, P310 to P312, P315 (Total of 13 pins)		ΣI_{OH}	—	—	–80	mA
		P101 to P107, P200, P201, P300 to P304 (Total of 14 pins)			—	—	–80	
		PA04 to PA07, PB00 to PB06, PC00 to PC06 (Total of 18 pins)			—	—	–80	
		PA00 to PA03, PE00 to PE06 (Total of 11 pins)			—	—	–80	
	VCC IO (LQFP144)	P000 to P007, P306, P307, P310 to P312, P315 (Total of 14 pins)		ΣI_{OH}	—	—	–80	mA
		P202 to P207, P302 to P304 (Total of 9 pins)			—	—	–80	
		P100 to P107, P200, P201, P300, P301 (Total of 12 pins)			—	—	–80	
		P800 to P803, PC00 to PC07 (Total of 12 pins)			—	—	–80	
		PB00 to PB07 (Total of 8 pins)			—	—	–80	
		PA00 to PA07 (Total of 8 pins)			—	—	–80	
		P700 to P702, PE00 to PE07 (Total of 11 pins)			—	—	–80	
	VCC IO (LQFP176, BGA176)	P000 to P007, P306, P307, P310 to P312, P315 (Total of 14 pins)		ΣI_{OH}	—	—	–80	mA
		P202 to P207, P302 to P304 (Total of 9 pins)			—	—	–80	
		P100 to P107, P200, P201, P300, P301, P804, P805 (Total of 14 pins)			—	—	–80	
		P500 to P503, P800 to P803, PC00 to PC07 (Total of 16 pins)			—	—	–80	
		PA04 to PA07, PB00 to PB07, PG00 to PG02 (Total of 15 pins)			—	—	–80	
		P700 to P707, PA00 to PA03 (Total of 12 pins)			—	—	–80	
		PE00 to PE07 (Total of 8 pins)			—	—	–80	

Table 67.46 Permissible Output Currents (5/6)

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output high current (total)	VCC IO (BGA224)	P000 to P007, P214, P215 (Total of 10 pins)	ΣI_{OH}	—	—	–80	mA
		P306, P307, P310 to P312, P315, P208 to P213 (Total of 12 pins)		—	—	–80	
		P112 to P115, P202 to P207, PF02 to PF05 (Total of 14 pins)		—	—	–80	
		P108 to P111, P300 to P304, P804 to P807 (Total of 13 pins)		—	—	–80	
		P100 to P107, P200, P201, P504 to P507 (Total of 14 pins)		—	—	–80	
		P500 to P503, P800 to P803, PC05 to PC07 (Total of 11 pins)		—	—	–80	
		PB04 to PB07, PC00 to PC04 (Total of 9 pins)		—	—	–80	
		PB00 to PB03, PG04 to PG07 (Total of 8 pins)		—	—	–80	
		PA04 to PA07, PG00 to PG03 (Total of 8 pins)		—	—	–80	
		P700 to P707, PA00 to PA03 (Total of 12 pins)		—	—	–80	
		PE00 to PE07, PF06 to PF09 (Total of 12 pins)		—	—	–80	
	VCC IO (BGA320)	P000 to P015, P214, P215 (Total of 18 pins)	ΣI_{OH}	—	—	–80	mA
		P306, P307, P310 to P312, P315, P208 to P213 (Total of 12 pins)		—	—	–80	
		P112 to P115, P202 to P207, PF02 to PF05 (Total of 14 pins)		—	—	–80	
		P108 to P111, P300 to P304, P804 to P807 (Total of 13 pins)		—	—	–80	
		P100 to P107, P200, P201, P504 to P507 (Total of 14 pins)		—	—	–80	
		P500 to P503, P800 to P803, PC05 to PC07(Total of 11 pins)		—	—	–80	
		P512 to P515, PB04 to PB07, PC00 to PC04 (Total of 13 pins)		—	—	–80	
		P510, P511, PB00 to PB03, PG04 to PG07 (Total of 10 pins)		—	—	–80	
		P508, P509, PA04 to PA07, PG00 to PG03 (Total of 10 pins)		—	—	–80	
		P704 to P707, PA00 to PA03, PG12 to PG15 (Total of 12 pins)		—	—	–80	
		P700 to P703, P712 to P715, PG08 to PG11 (Total of 12 pins)		—	—	–80	
		P708 to P711, PE00 to PE07, PF06 to PF09 (Total of 16 pins)		—	—	–80	

Table 67.46 Permissible Output Currents (6/6)

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output high current (total)	VCC2 IO	P600 to P607, P900 to P907, PD00 to PD07 (Total of 24 pins)	ΣI_{OH_VCC2}	—	—	–80	mA
	AVCC0 IO	P400 to P415 (Total of 16 pins)	ΣI_{OH_AVCC0}	—	—	–33	mA
	VCC_USB IO	PF00, PF01 (Total of 2 pins)*6	ΣI_{OH_VCCUSB}	—	—	–33	mA

Caution: To protect the MCU's reliability, the output current values should not exceed the values in Table 67.46.
The average output current is obtained from the values for current measured over a period of 100 µs.
Note 1. When SDA00-A and SCL00-A are selected for the P102 and P103 pins.
Note 2. When SDAHSn-A and SCLHSn-A (n = 1, 2) are selected for the P106, P107, P200, and P201 pins.
Note 3. This is the driving ability as selected by the setting of the driving ability select (PmnPFS.DSCR[3:0]) bits.
Note 4. The values are for pins excluding USB0_DP, USB0_DM, USBHS_DP, and USBHS_DM.
Note 5. This characteristics depends on the setting of the LVO0E bit in the low-voltage operating characteristics control register (LVOCR).
Note 6. When a function other than USB0_DP or USB0_DM is selected.
Note 7. When the module operates in Push-Pull mode

67.4.5 Thermal Characteristics

67.4.5.1 Thermal Resistance Value

Table 67.47 Thermal Resistance Value (Reference)

Item	Package	Symbol	Max.	Unit	Test Conditions
Thermal resistance	176-pin LQFP (PLQP0176KJ-A)	θ_{ja}	29.6	°C/W	JESD51-2 and JESD51-7 compliant
	144-pin LQFP (PLQP0144KA-B)		34.3		
	100-pin LQFP (PLQP0100KP-A)		31.6		
	320-pin BGA (PLBG0320GD-A)		20.3		JESD51-2 and JESD51-9 compliant
	224-pin BGA (PLBG0224GE-A)		20.4		
	176-pin BGA (PLBG0176GG-A)		24.7		
	176-pin LQFP (PLQP0176KJ-A)	Ψ_{jt}	0.45	°C/W	JESD51-2 and JESD51-7 compliant
	144-pin LQFP (PLQP0144KA-B)		0.36		
	100-pin LQFP (PLQP0100KP-A)		0.45		
	320-pin BGA (PLBG0320GD-A)		0.17		JESD51-2 and JESD51-9 compliant
	224-pin BGA (PLBG0224GE-A)		0.19		
	176-pin BGA (PLBG0176GG-A)		0.21		

Note: The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, refer to the JEDEC standards.

67.4.5.2 Supply Currents for the Peripheral Functions

Table 67.48 Operating Currents for the Peripheral Functions (Load Currents inside the Chip after Release from Module Stop State)

Item		Symbol	Max.	Unit	Register
SRAM1		I_{DD}	23.1	mA	MSTPCRA.MSTPA1
DMAC, DTC	DMACAa (8 channels), DTC	I_{DD}	7.04	mA	MSTPCRA.MSTPA22
	EXDMACAa (2 channels)		6.56		MSTPCRA.MSTPA29
Timers	GPTW (16 channels)	I_{DD}	5.54	mA	MSTPCRE.MSTPE31
	HRPWMb		0.09		MSTPCRD.MSTPD6
	TMRa (2 units)		0.07		MSTPCRF.MSTPF7, MSTPCRF.MSTPF6
	CMT (2 units)		0.06		MSTPCRF.MSTPF15, MSTPCRF.MSTPF14
	CMTW (2 units)		0.05		MSTPCRF.MSTPF13, MSTPCRF.MSTPF12
Communication interface	ESWM	I_{DD}	24.5	mA	MSTPCRC.MSTPC30
	USB		0.25		MSTPCRB.MSTPB11
	USBHS		0.35		MSTPCRB.MSTPB12
	RSCI (13 channels)		5.34		MSTPCRB.MSTPB31 to MSTPCRB.MSTPB20, MSTPCRB.MSTPB0
	RI3C		0.21		MSTPCRB.MSTPB4
	RIICHS (2 channels)		0.40		MSTPCRB.MSTPB8, MSTPCRB.MSTPB7
	CANFDG (2 channels)		0.30		MSTPCRC.MSTPC25
	CANFD		0.14		MSTPCRC.MSTPC27
	RSPIA (3 channels)		0.67		MSTPCRB.MSTPB19, MSTPCRB.MSTPB18, MSTPCRC.MSTPC23
	SDHIc		0.08		MSTPCRC.MSTPC12
	PDMIF		0.03		MSTPCRC.MSTPC24
	XSPI (2 units), DOTF		9.37		MSTPCRB.MSTPB17, MSTPCRB.MSTPB16
	PCIF (2 units)		0.44		MSTPCRF.MSTPF31, MSTPCRF.MSTPF30
	SSIE (2 channels)		0.08		MSTPCRC.MSTPC8, MSTPCRC.MSTPC7
Analog	ADC16H	I_{DD}	4.73	mA	MSTPCRD.MSTPD21
	R12DAB		0.02		MSTPCRD.MSTPD20, MSTPCRD.MSTPD19
	Temperature sensor		0.01		MSTPCRD.MSTPD22
Safety, security function	CRCA	I_{DD}	0.01	mA	MSTPCRC.MSTPC1
	DOCA		0.01		MSTPCRC.MSTPC13
	RSIP		0.67		MSTPCRC.MSTPC31
	POEGd (4 channels)		0.22		MSTPCRD.MSTPD14, MSTPCRD.MSTPD13, MSTPCRD.MSTPD12, MSTPCRD.MSTPD11
Image processing	PDC	I_{DD}	0.04	mA	MSTPCRC.MSTPC2

Note: This table lists the internal load currents flowing inside the chip; they are not the I_{CC_DCDC} current supplied to the DCDC.
Note: The values listed are the increases in the currents after release from the module stop state.

Table 67.49 Leakage Current of the Peripheral Functions (Load Current inside the Chip)

Item	Tj = 105°C	Tj = 125°C	Unit	Control register
ESWM	9.48	18.3	mA	PDCTRESWM.PDDE

Note: The value given is the leakage current when the ESWM power domain is turned on.

67.4.5.3 Example of Calculation of T_a

The estimated maximum operating temperature (i.e., maximum ambient temperature) can be calculated from the maximum junction temperature and the thermal resistance value by using the following formula.

Tamax = Tjmax – (Iccmax × VCC × θja)

- Tamax: Maximum ambient temperature (maximum operating temperature) [°C]
- Tjmax: Maximum junction temperature [°C]
- Iccmax: Maximum operating current [A]
- θja: Thermal resistance value

For example, for the BGA320 package, when the LDO efficiency improvement function is enabled and the ICC current in maximal operation is 211 mA (typ) with VCC = 3.3 V, the calculation can proceed as shown below. In this case, the MCU can operate at ambient temperatures up to 111°C.

Tamax = 125°C – (0.211 A × 3.3 V × 20.3°C/W) = Approximately 111°C

Figure 67.7 shows the VCC voltage dependence of the maximum ambient temperature with a load current of 250 mA when the LDO efficiency improvement function is disabled.

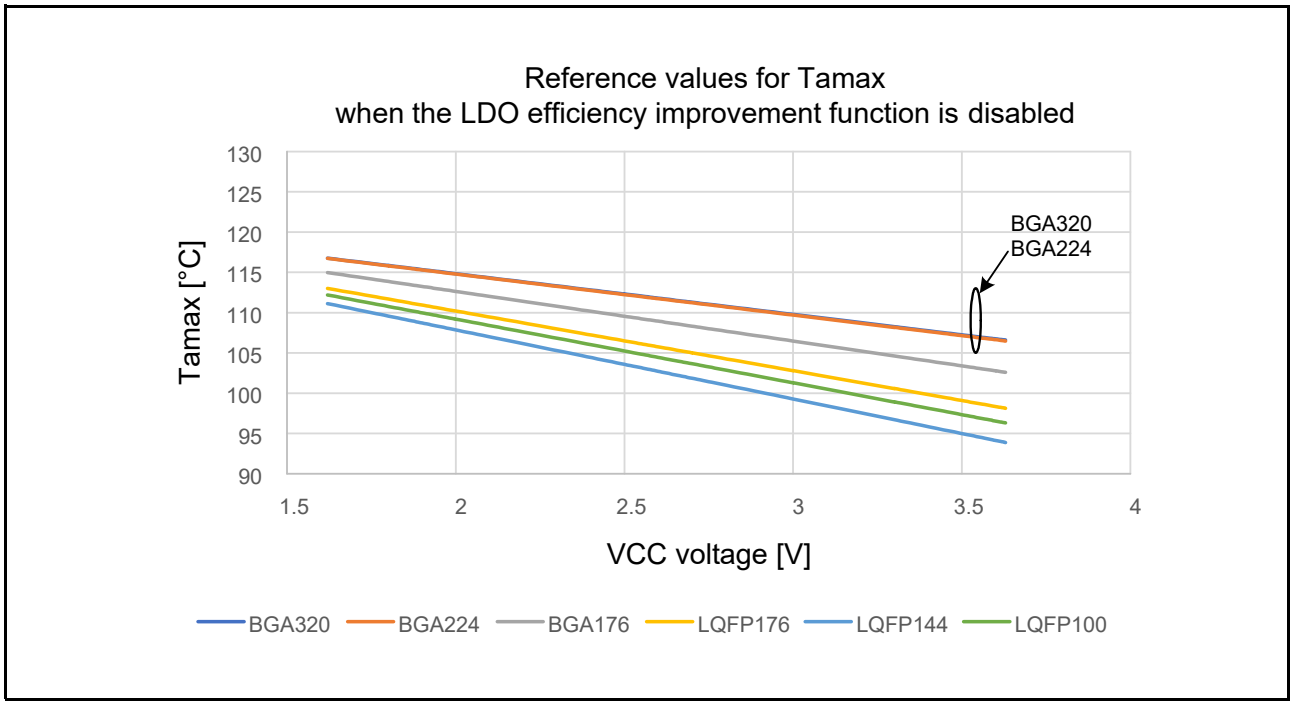


Figure 67.7 VCC Voltage Dependence of the Maximum Ambient Temperature with a Load Current of 250 mA When the LDO Efficiency Improvement Function Is Disabled

Figure 67.8 shows the VCC voltage dependence of the maximum ambient temperature with a load current of 250 mA when the LDO efficiency improvement function is enabled.

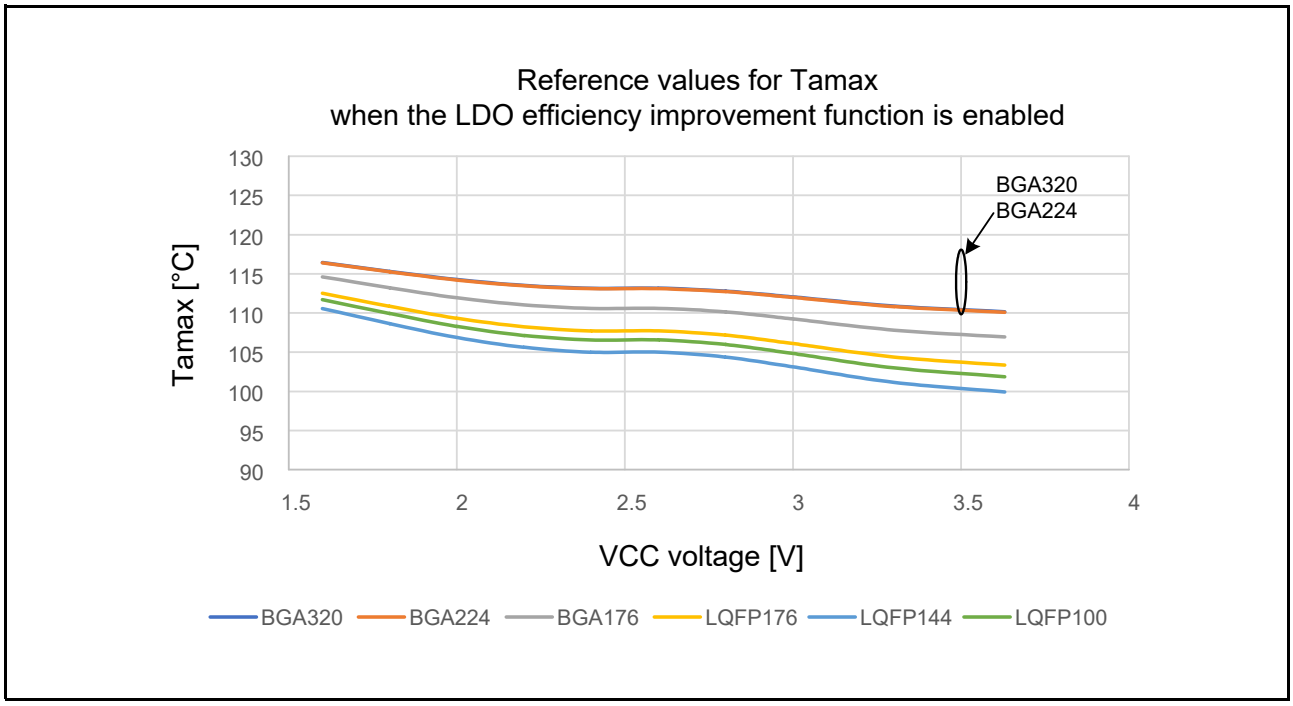


Figure 67.8 VCC Voltage Dependence of the Maximum Ambient Temperature with a Load Current of 250 mA When the LDO Efficiency Improvement Function Is Enabled

Figure 67.9 shows the VCC voltage dependence of the maximum ambient temperature with a load current of 200 mA when the DCDC supply is in use. Since the ICC current and power conversion efficiency of the DCDC supply vary with the VCC voltage, the values shown are only for reference and were calculated assuming a power conversion efficiency of approximately 70% at higher VCC voltages and approximately 85% at lower VCC voltages.

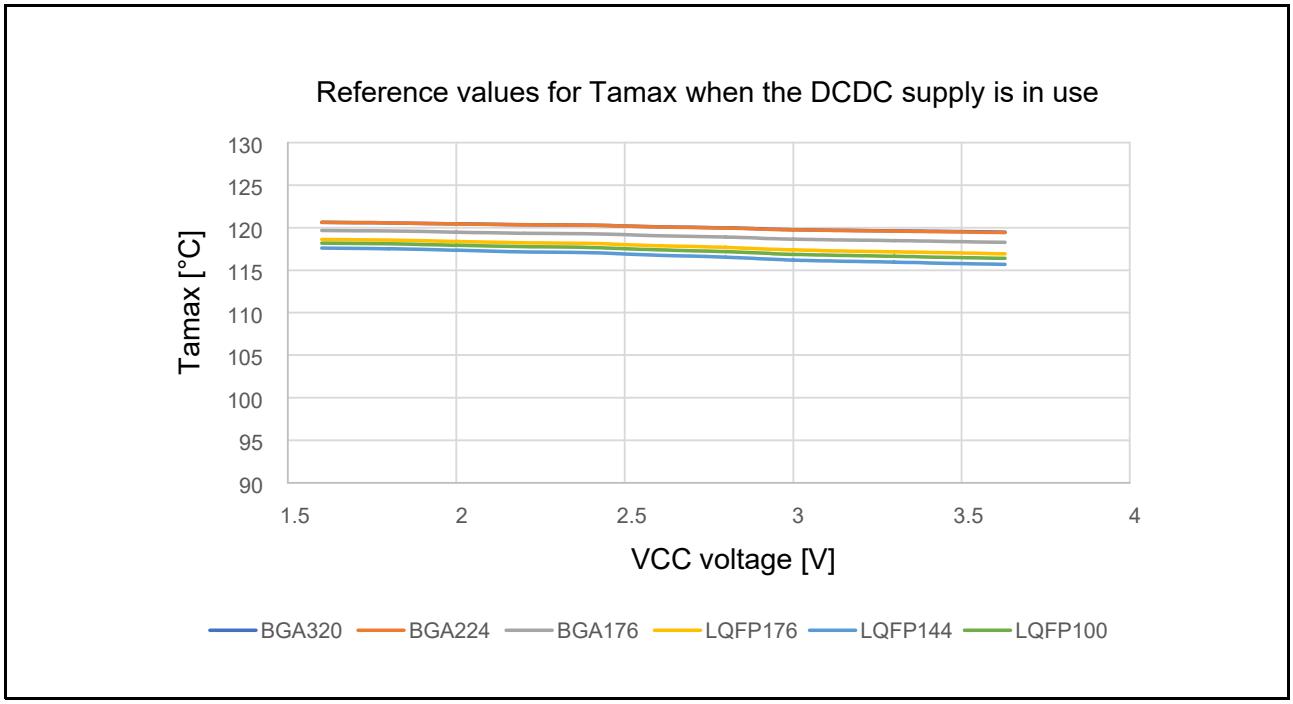


Figure 67.9 VCC Voltage Dependence of the Maximum Ambient Temperature with a Load Current of 200 mA When the DCDC Supply Is in Use

67.4.6 VCC Rising and Falling Slopes and Ripple Frequency

Table 67.50 Characteristics of VCC Rising and Falling Slopes When Power is Switched On and Off

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
VCC rising slope at power on*1		SrVCC	0.0084	—	4	ms/V	Figure 67.10
VCC falling slope at power off*2	VBATT function is disabled	SfVCC1	0.0084	—	—		
	VBATT function is enabled	SfVCC2	1.0000	—	—		

Note 1. When the VCC voltage changes from below V_{POR1} to above V_{POR1}
Note 2. When the VCC voltage changes from above V_{POR1} to below V_{POR1}

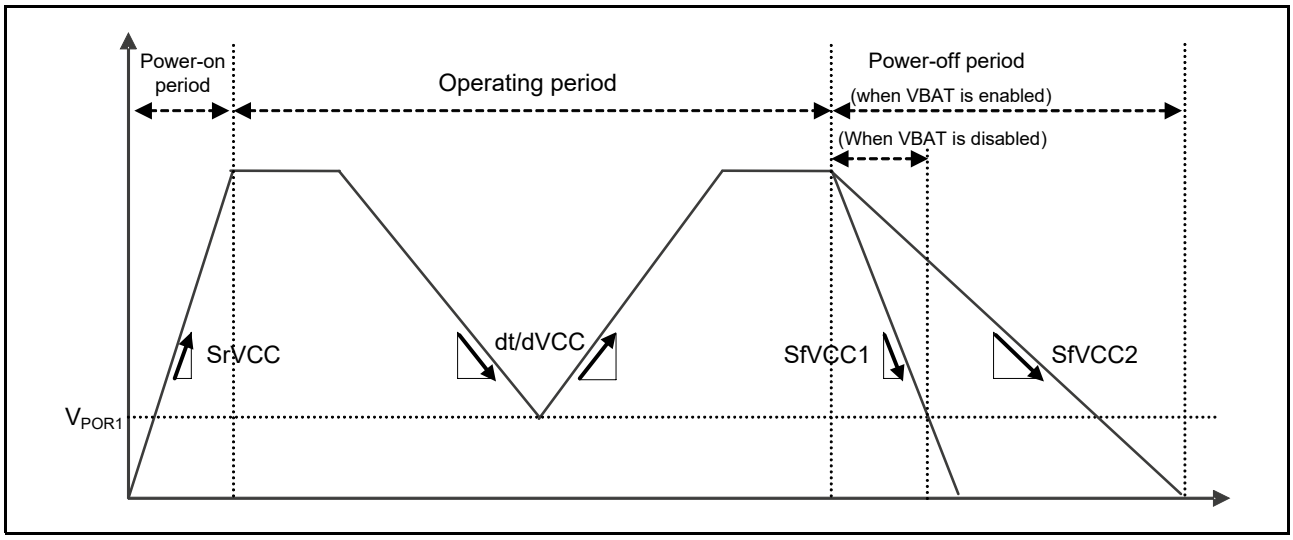


Figure 67.10 VCC Rising and Falling Slopes

Table 67.51 Characteristics of VCC Ripple Frequency and Rising and Falling Slope during Operation

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10	kHz	Figure 67.11 $V_{r(VCC)} \leq VCC \times 0.2$
		—	—	1	MHz	Figure 67.11 $V_{r(VCC)} \leq VCC \times 0.08$
		—	—	10		Figure 67.11 $V_{r(VCC)} \leq VCC \times 0.06$
Allowable voltage fluctuation rising and falling slope*1	dt/dVCC	1	—	—	ms/V	When the fluctuation in VCC exceeds VCC ± 10%

Note: The ripple voltage must be within the range between the upper (3.63 V) and lower (1.62 V) limits of VCC and satisfy the condition for allowable ripple frequency $f_{r(VCC)}$.
Note: When the fluctuation in VCC exceeds VCC ± 10%, the rate of change must be within the allowable voltage fluctuation rising and falling slope dt/dVCC.
Note 1. When the VCC voltage does not drop below V_{POR1}.

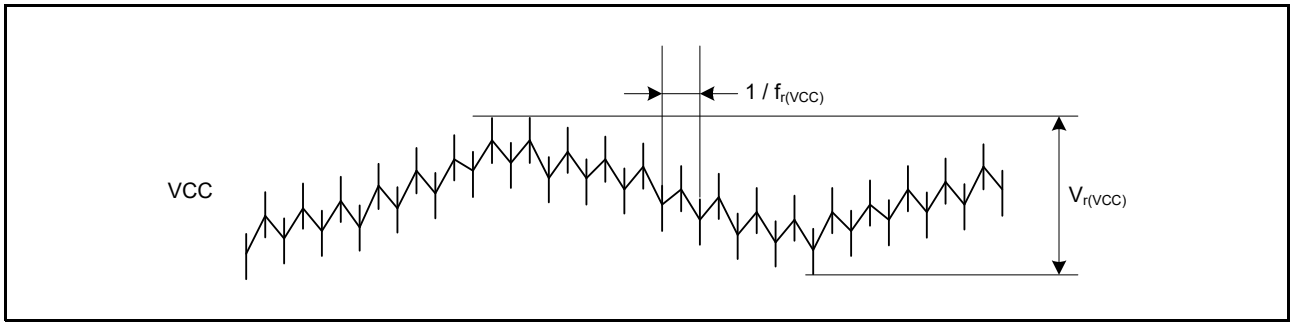


Figure 67.11 Ripple Waveform

67.5 AC Characteristics

67.5.1 Operating Frequency

Table 67.52 Operating Frequency

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Operating frequency	CPU clock (CPUCLK0)*1	For product part no. R5K574Nxy4xx (y = A, B)	f	—	—	600	MHz	VSCR_1 (DCDC mode) Voltage range 1 (external VDD mode)
				—	—	480		VSCR_2 (DCDC mode) Voltage range 2 (external VDD mode)
		For product part no. R5K574Nxy4xx (y = C, D)		—	—	480		VSCR_2 (DCDC mode) Voltage range 2 (external VDD mode)
		For product part no. R5K574Nxy4xx (y = E, F)		—	—	300		
	System clock (ICLK)*1			—	—	300		
	Peripheral module clock (PCLKA)*1, *2			—	—	150		
	Peripheral module clock (PCLKB)*1			—	—	75		
	Peripheral module clock (PCLKC)*1			—	—	75		
	Peripheral module clock (PCLKD)*1			—	—	150		
	Peripheral module clock (PCLKE)*1			—	—	300		
	MRAM code bus clock (MRICLK)			—	—	300		
	MRAM peripheral bus clock (MRPCLK)*1			—	—	150		
	External bus clock	Synchronous clock (BCLK)		—	—	150		
		Asynchronous clock (BCLKA)		—	—	133		
	EBCLK pin output*3	3.0 V ≤ VCC ≤ 3.6 V		—	—	100		
		2.7 V ≤ VCC < 3.0 V		—	—	60		
	SDCLK pin output*3	3.0 V ≤ VCC ≤ 3.6 V		—	—	133		
		2.7 V ≤ VCC < 3.0 V		—	—	60		
	SCI clock (SCICLK)			—	—	120		
	SPI clock (SPICLK)			—	—	100		
	XSPI clock (XSPICLK)			—	—	266		
	CANFD core clock (CANFDCLK)			—	—	80		
	ADC clock (ADCCLK)			25	—	60		
	GPT clock (GPTCLK)*2			—	—	300		
	USB clock (USBCLK)			—	—	48		
	USBHS-PHY clock (USB60CLK)			—	—	60		
	I3C clock (I3CCLK)			—	—	100		
	ESWM clock (ESWCLK)			—	—	250		
	ESWM-PHY clock (ESWPHYCLK)			—	—	500		
	Ethernet-PHY clock (ETHPHYCLK)			—	—	50		

Note 1. For the restriction among the CPUCLK0, ICLK, MRICLK, MRPCLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, and BCLK frequencies, see section 9, Clock Generation Circuit.
Note 2. When using PCLKA or GPTCLK at frequencies of and below 25 MHz or of and above 120 MHz, select the ADC clock (ADCCLK) as the A/D conversion clock (ADCLK).
Note 3. This is the case when the setting of the port drive capability bits in the PmnPFS register is for DRV7.

67.5.2 Clock Timing

Table 67.53 Timing of the EBCLK and SDCLK Pins Output

Conditions: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV7.

Item	VCC	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
ECLK pin output cycle time	3.0 V or higher	t_{Bcyc}	10.0	—	—	ns	Figure 67.12
	2.7 V or higher		16.6	—	—		
EBCLK pin output high pulse width	3.0 V or higher	t_{CH}	2.0	—	—	ns	
	2.7 V or higher		3.3	—	—		
EBCLK pin output low pulse width	3.0 V or higher	t_{CL}	2.0	—	—	ns	
	2.7 V or higher		3.3	—	—		
EBCLK pin output rising time	3.0 V or higher	t_{Cr}	—	—	3.0	ns	
	2.7 V or higher		—	—	5.0		
EBCLK pin output falling time	3.0 V or higher	t_{Cf}	—	—	3.0	ns	
	2.7 V or higher		—	—	5.0		
SDCLK pin output cycle time	3.0 V or higher	t_{SDcyc}	7.52	—	—	ns	
	2.7 V or higher		16.6	—	—		
SDCLK pin output high pulse width	3.0 V or higher	t_{CH}	1.0	—	—	ns	
	2.7 V or higher		2.2	—	—		
SDCLK pin output low pulse width	3.0 V or higher	t_{CL}	1.0	—	—	ns	
	2.7 V or higher		2.2	—	—		
SDCLK pin output rising time	3.0 V or higher	t_{Cr}	—	—	2.7	ns	
	2.7 V or higher		—	—	2.7		
SDCLK pin output falling time	3.0 V or higher	t_{Cf}	—	—	2.7	ns	
	2.7 V or higher		—	—	2.7		

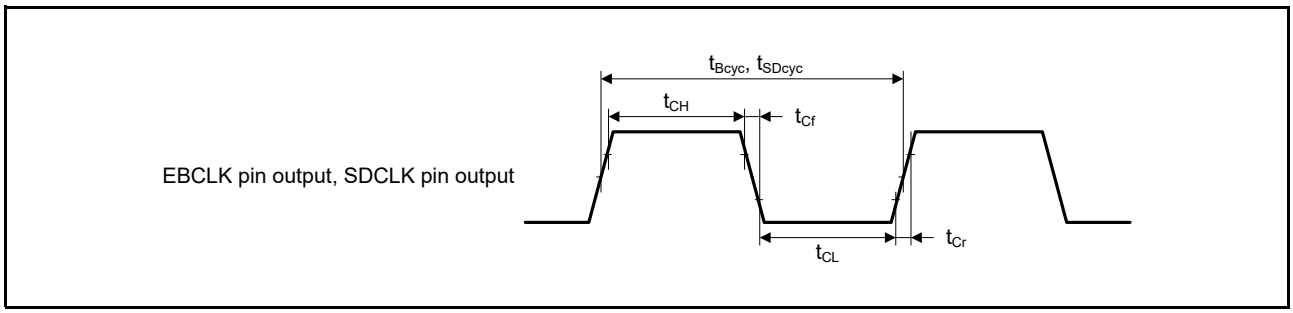


Figure 67.12 EBCLK and SDCLK Pins Output Timing

Table 67.54 EXTAL Clock Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{EXcyc}	20.80	—	—	ns	Figure 67.13
EXTAL external clock input high pulse width	t_{EXH}	5.30	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	5.30	—	—	ns	
EXTAL external clock rising time	t_{EXr}	—	—	3.0	ns	
EXTAL external clock falling time	t_{EXf}	—	—	3.0	ns	

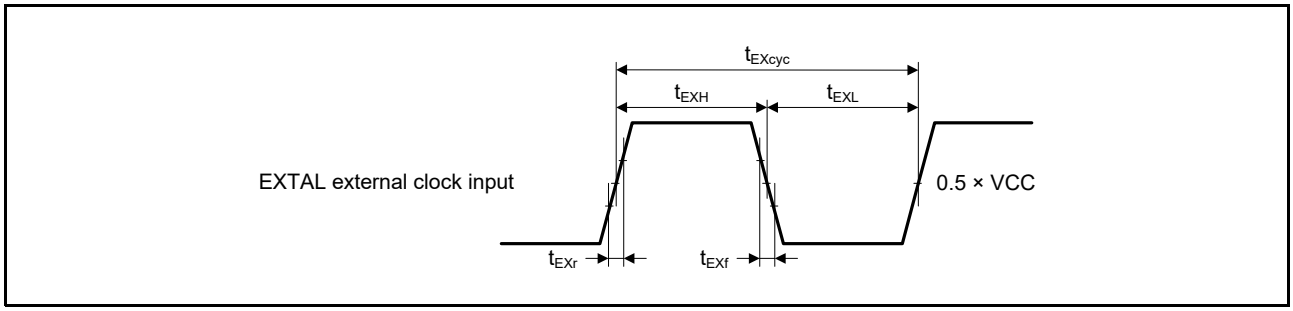


Figure 67.13 EXTAL External Clock Input Timing

Table 67.55 Main Clock Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Main clock oscillation frequency	f_{MAIN}	8	—	48	MHz	
Main clock oscillation stabilization wait time (crystal)*1	$t_{MAINOSCWT}$	—	—	—	ms	Figure 67.14

Note 1. When using the main clock, request evaluation by the manufacturer of the oscillator. For the time until the oscillation is stabilized, confirm the evaluation result by the manufacturer and set a value in the MOSCWTCR register that exceeds the manufacturer's recommended value. After having set the MOSCCR.MOSTP bit to start the main clock operation, wait until the oscillation is stabilized before using the main clock.

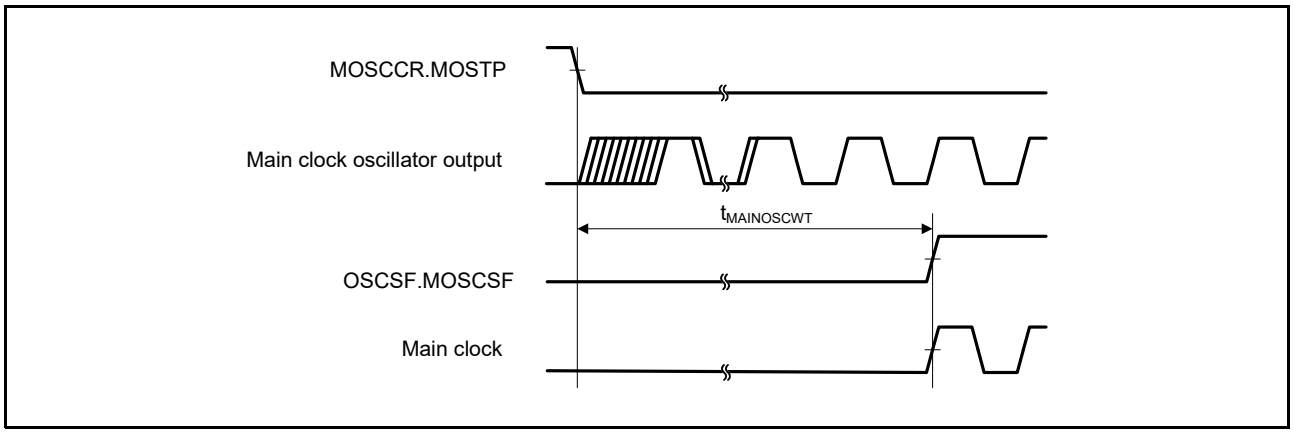


Figure 67.14 Main Clock Oscillator Start Timing

Table 67.56 LOCO and MOCO Clock Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LOCO clock oscillation frequency	f_{LOCO}	29.4912	32.768	36.0448	kHz	
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	26.0	µs	Figure 67.15
MOCO clock oscillation frequency	f_{MOCO}	7.2	8.0	8.8	MHz	
MOCO clock oscillation stabilization wait time	t_{MOCOWT}	—	—	3.5	µs	Figure 67.16

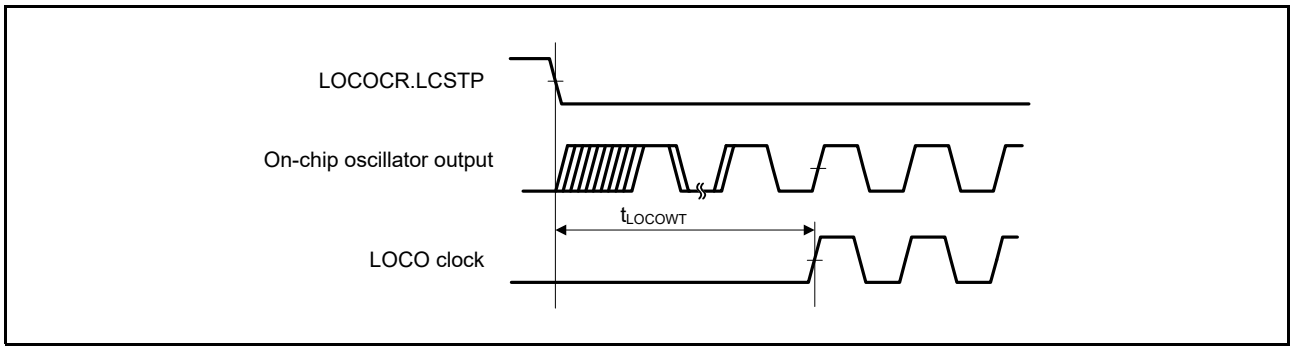


Figure 67.15 LOCO Clock Oscillation Start Timing

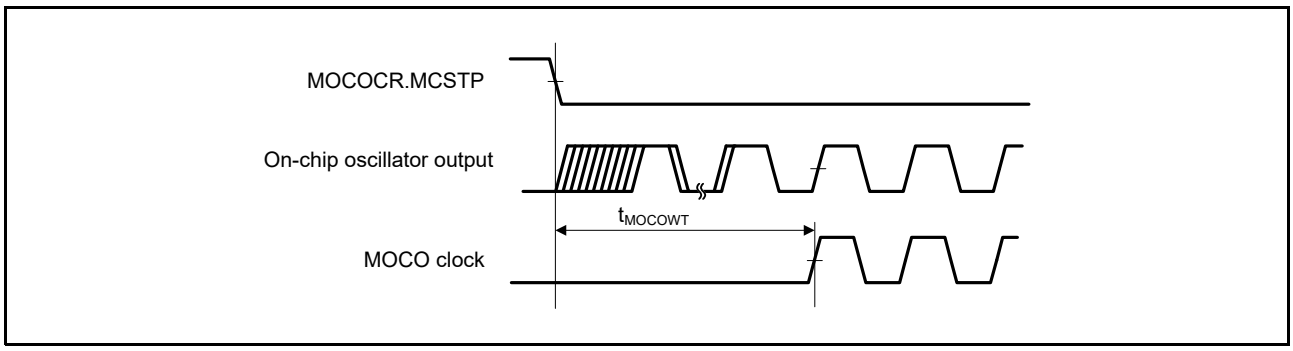


Figure 67.16 MOCO Clock Oscillation Start Timing

Table 67.57 HOCO Clock Timing

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	Without FLL	f_{HOCO16}	15.776	16.000	16.224	MHz	–20°C ≤ T_j ≤ 105°C
		f_{HOCO18}	17.748	18.000	18.252		
		f_{HOCO20}	19.720	20.000	20.280		
		f_{HOCO32}	31.552	32.000	32.448		
		f_{HOCO48}	47.328	48.000	48.672		
		f_{HOCO16}	15.712	16.000	16.288		–40°C ≤ T_j ≤ 125°C
		f_{HOCO18}	17.676	18.000	18.324		
		f_{HOCO20}	19.640	20.000	20.360		
		f_{HOCO32}	31.424	32.000	32.576		
		f_{HOCO48}	47.136	48.000	48.864		
	With FLL	f_{HOCO16}	15.960	16.000	16.040		–40°C ≤ T_j ≤ 125°C Sub-clock frequency accuracy is ±50 ppm.
		f_{HOCO18}	17.955	18.000	18.045		
		f_{HOCO20}	19.950	20.000	20.050		
		f_{HOCO32}	31.920	32.000	32.080		
		f_{HOCO48}	47.880	48.000	48.120		
HOCO clock oscillation stabilization wait time*1		t_{HOCOWT}	—	—	15.0	μs	Figure 67.17
HOCO stop width time		$t_{HOCOSTP}$	1.0	—	—	μs	Figure 67.18
HOCO period jitter		—	–3.0	—	3.0	ps	
FLL stabilization wait time		t_{FLLWT}	—	—	1.92	ms	

Note 1. This value indicates the time that is required for the HOCO oscillation frequency (f_{HOCO}) to fall within the corresponding guaranteed range of error in the above table after release from the reset state.

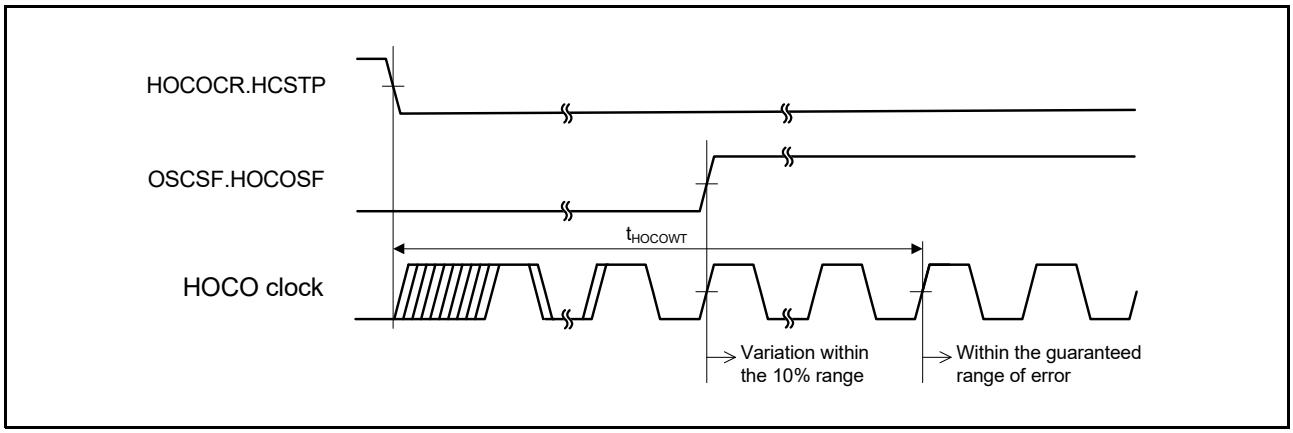


Figure 67.17 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

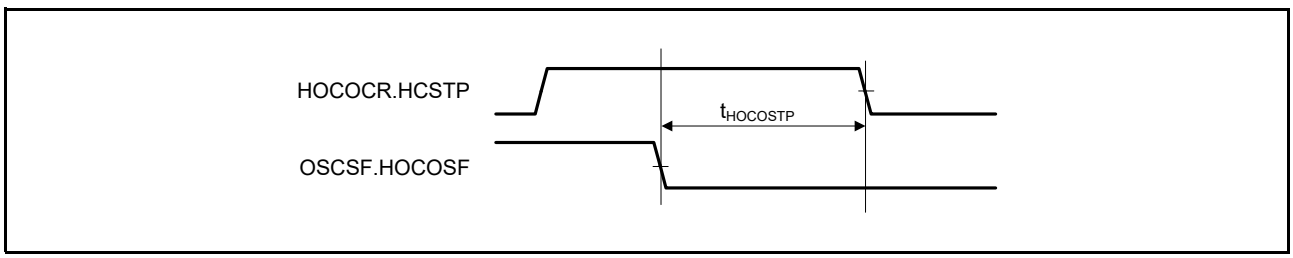
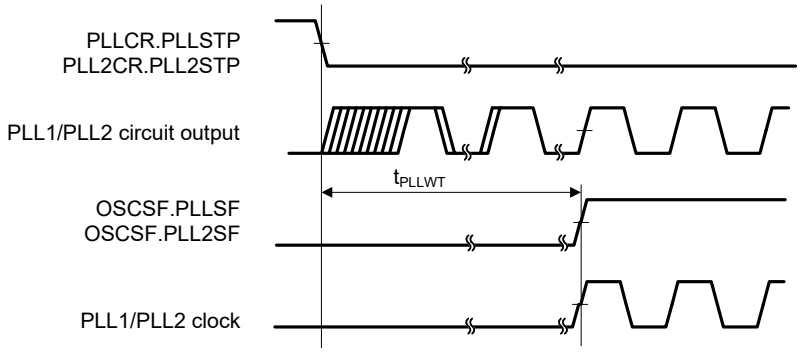


Figure 67.18 HOCO Stop Width Time

Table 67.58 PLL Clock Timing

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL1 clock oscillation frequency	VCO frequency		f_{PLL_VCO}	960	—	2400	MHz	—
	PLL output clock P	Part no. type 1	f_{PLL_P}	60	—	600		VSCR_1 (DCDC mode) Voltage range 1 (external VDD mode)
				60	—	480		VSCR_2 (DCDC mode) Voltage range 2 (external VDD mode)
		Part no. type 2		60	—	480		VSCR_2 (DCDC mode) Voltage range 2 (external VDD mode)
		Part no. type 3		60	—	300		
	PLL output clock Q	Part no. types 1 and 2	f_{PLL_Q}	106.6	—	500		—
		Part no. type 3		106.6	—	300		—
	PLL output clock R		f_{PLL_R}	106.6	—	500		—
PLL2 clock oscillation frequency	VCO frequency		f_{PLL_VCO}	960	—	2400	MHz	—
	PLL output clock P	Part no. types 1 and 2	f_{PLL_P}	60	—	500		—
		Part no. type 3		60	—	300		—
	PLL output clock Q	Part no. types 1 and 2	f_{PLL_Q}	106.6	—	500		—
		Part no. type 3		106.6	—	300		—
	PLL output clock R		f_{PLL_R}	106.6	—	500		—
PLL1/PLL2 clock oscillation stabilization wait time			t_{PLLWT}	—	—	50	µs	Figure 67.19
PLL1/PLL2 period jitter			—	—	±52	—	ps	—
PLL1/PLL2 long term jitter			—	—	±300	—	ps	Term: 1 µs, 10 µs

Part no. type 1: Products with part numbers of the form R5K574Nxy4xx (y = A, B)
Part no. type 2: Products with part numbers of the form R5K574Nxy4xx (y = C, D)
Part no. type 3: Products with part numbers of the form R5K574Nxy4xx (y = E, F)



Note 1. The PLL only starts operation after the main clock oscillation is stabilized.

Figure 67.19 PLL1/PLL2 Clock Oscillation Start Timing

Table 67.59 Sub-Clock Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Sub-clock oscillation frequency	f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization wait time*1	$t_{SUBOSCWT}$	—	—	—	s	Figure 67.20

Note 1. When using the sub-clock, request evaluation by the manufacturer of the oscillator. Regarding the oscillation stabilization wait time, refer to the resonator manufacturer's evaluation results and wait in the program for a time longer than the value recommended by the resonator manufacturer. After having set the SOSCCR.SOSTP bit to start the sub-clock operation, wait until the oscillation is stabilized before using the sub-clock.

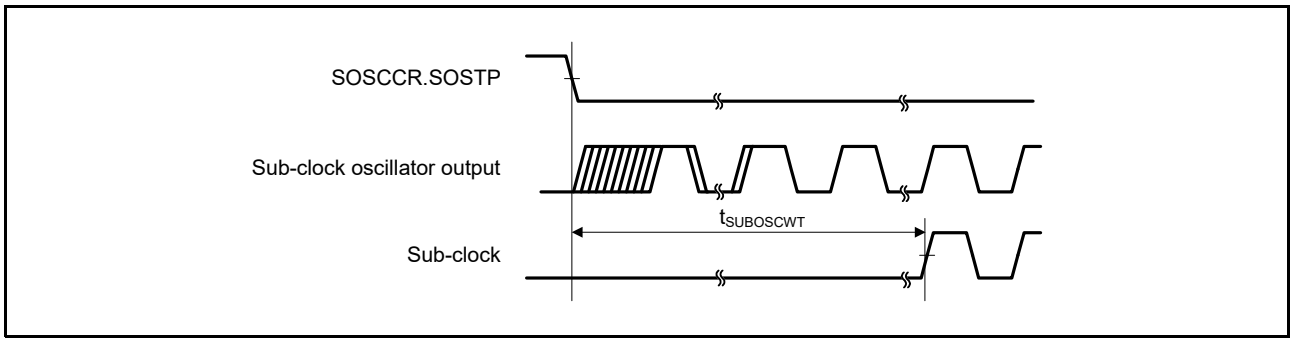


Figure 67.20 Sub-Clock Oscillation Start Timing

Table 67.60 EXCIN External Clock Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXCIN external clock input cycle time	t_{EXcyc}	27.7	—	—	µs	Figure 67.21
EXCIN external clock input frequency	t_{EXCIN}	—	—	36	kHz	
EXCIN external clock input high pulse width	t_{EXH}	11.08	—	—	µs	
EXCIN external clock input low pulse width	t_{EXL}	11.08	—	—	µs	
EXCIN external clock rising time	t_{EXr}	—	—	200	ns/V	
EXCIN external clock falling time	t_{EXf}	—	—	200	ns/V	

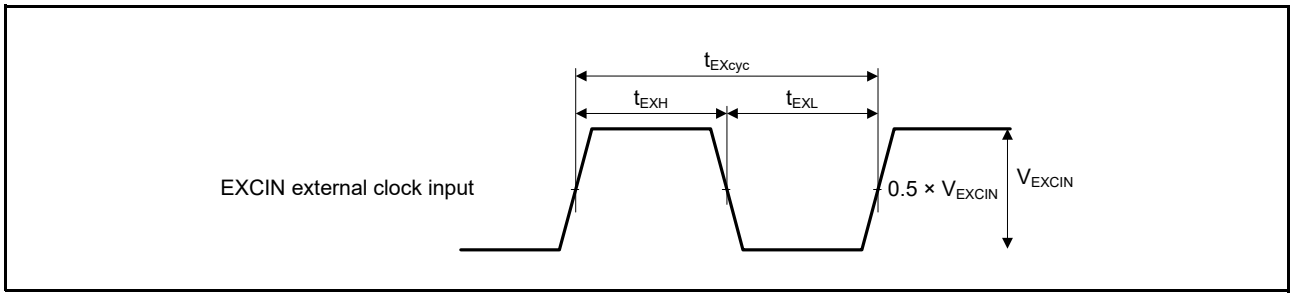
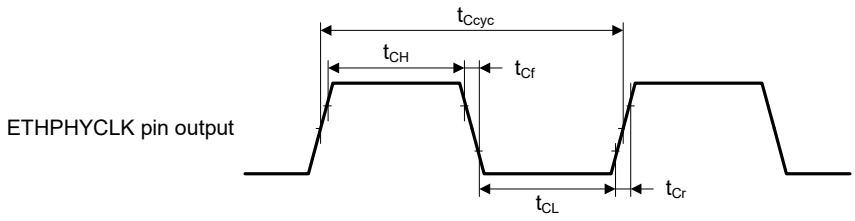


Figure 67.21 EXCIN External Clock Input Timing

Table 67.61 ETHPHYCLK Pin Output Timing

Conditions: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.
VCC = 2.70 to 3.63 V
Capacitive load C_L = 30 pF (during operation at 25 MHz), C_L = 15 pF (during operation at 50 MHz)

Item	Output Frequency	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
ETHPHYCLK pin output cycle time	25 MHz	t_{Ccyc}	—	40	—	ns	Figure 67.22
	50 MHz		—	20	—		
ETHPHYCLK pin high pulse width	25 MHz	t_{CH}	13	—	—	ns	
	50 MHz		6.5	—	—		
ETHPHYCLK pin low pulse width	25 MHz	t_{CL}	13	—	—	ns	
	50 MHz		6.5	—	—		
ETHPHYCLK pin rising time	25 MHz	t_{Cr}	—	—	3.3	ns	
	50 MHz		—	—	1.66		
ETHPHYCLK pin falling time	25 MHz	t_{Cf}	—	—	3.3	ns	
	50 MHz		—	—	1.66		



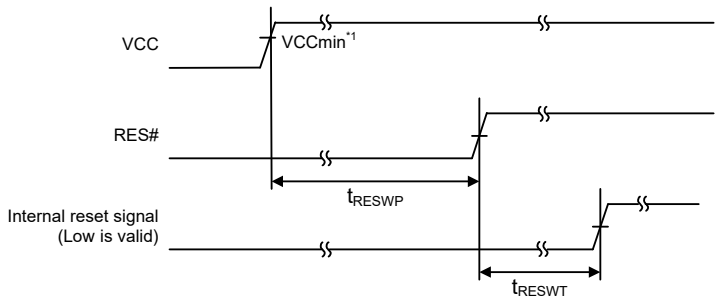
Test conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C_L = 30 pF or 15 pF

Figure 67.22 ETHPHYCLK Pin Output Timing

67.5.3 Reset Timing

Table 67.62 Reset Timing

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on		t_{RESWP}	2.40	—	—	ms	Figure 67.23
	Deep software standby mode 1	DPSBYCR.DCSSMODE = 01b	t_{RESWD}	0.51	—	—		Figure 67.24
		DPSBYCR.DCSSMODE = 10b		0.67	—	—		
		DPSBYCR.DCSSMODE = 11b		1.00	—	—		
		DPSBYCR.LDOSSMODE = 01b		0.51	—	—		
		DPSBYCR.LDOSSMODE = 10b		0.67	—	—		
		DPSBYCR.LDOSSMODE = 11b		1.00	—	—		
	Deep software standby mode 2	DPSBYCR.DCSSMODE = 01b		0.51	—	—		
		DPSBYCR.DCSSMODE = 10b		0.67	—	—		
		DPSBYCR.DCSSMODE = 11b		1.00	—	—		
		DPSBYCR.LDOSSMODE = 01b		0.51	—	—		
		DPSBYCR.LDOSSMODE = 10b		0.67	—	—		
		DPSBYCR.LDOSSMODE = 11b		1.00	—	—		
	Deep software standby mode 3	DPSBYCR.DCSSMODE = 01b		0.68	—	—		
		DPSBYCR.DCSSMODE = 10b		0.84	—	—		
		DPSBYCR.DCSSMODE = 11b		1.20	—	—		
		DPSBYCR.LDOSSMODE = 01b		0.68	—	—		
		DPSBYCR.LDOSSMODE = 10b		0.84	—	—		
		DPSBYCR.LDOSSMODE = 11b		1.20	—	—		
	Software standby mode		t_{RESWS}	0.55	—	—		
	CPU deep sleep mode (SCKSCR.CKSEL[2:0] = 100b)		$t_{RESWSODS}$	0.16	—	—		
	CPU deep sleep mode (Any setting for SCKSCR.CKSEL[2:0] other than 100b)		t_{RESWDS}	0.04	—	—		
Waiting time after release from the RES# pin reset			t_{RESWT}	—	78.7	79.1	µs	Figure 67.23
Waiting time after release from internal reset time (independent watchdog timer reset, watchdog timer reset, bus error reset, common memory error reset, software reset, local memory error reset)			t_{RESWT2}	—	78.7	79.1	µs	
RESOUT# output pulse width		OFS1.ROWSEL[2:0] = 000b	t_{ROUT}	0.25	0.28	0.31	ms	Figure 67.25
		OFS1.ROWSEL[2:0] = 100b		0.50	0.55	0.62		
		OFS1.ROWSEL[2:0] = 101b		1.00	1.10	1.23		
		OFS1.ROWSEL[2:0] = 110b		5.00	5.50	6.12		
		OFS1.ROWSEL[2:0] = 111b		10.0	11.0	12.2		



Note 1. VCCmin is the minimum guaranteed voltage of the MCU.

Figure 67.23 Reset Input Timing at Power-On

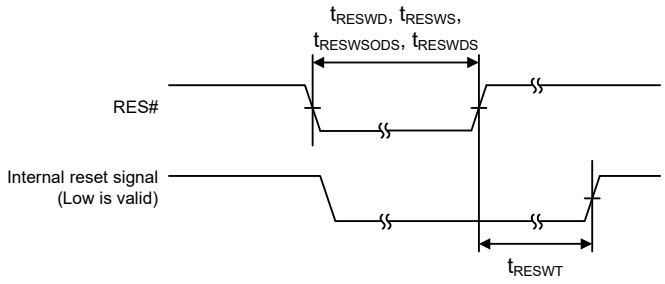
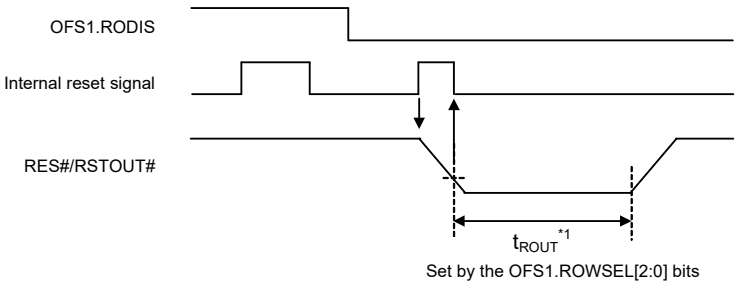


Figure 67.24 Reset Input Timing



Note 1. t_{ROUT} is the value counted for the time set by OFS1.ROWSEL[2:0] after the internal reset is released. After the count ends, the RSTOUT# output is released.

Figure 67.25 RSTOUT# Output Operation

67.5.4 Timing of Return from Low Power Consumption Modes

The following symbols are used in this section to represent the frequencies of the individual clocks.

f_{MOCO}: MOCO clock (8 MHz)
$f_{MOCOmin}$: Minimum frequency of the MOCO clock (7.2 MHz)
f_{SRCCLK}: System clock shown in Table 67.64 to Table 67.67
f_{PLL}: PLL1 circuit clock
f_{ICLK}: System clock ICLK
f_{MOSC}: MOSC clock
f_{HOCO}: HOCO clock
f_{SOSC}: SOSC clock (32.768 kHz)

Table 67.63 Time to Return from CPU Deep Sleep Mode

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Time to return from CPU deep sleep mode	t_{DSLP}	—	6.13	9.44	µs	ICLK = 300 MHz

Note: To calculate the time to return at a frequency that differs from the condition in testing, use the following formula.
t_{DSLP} (typ.) = 44 / f_{MOCO} + 9 / f_{ICLK} + 0.6 [µs], t_{DSLP} (max.) = 44 / $f_{MOCOmin}$ + 9 / f_{ICLK} + 3.3 [µs]
(f_{MOCO}: MOCO frequency [MHz], $f_{MOCOmin}$: Minimum frequency of MOCO [MHz], f_{ICLK}: ICLK frequency [MHz])

Table 67.64 Time to Return from Software Standby Mode 1

Item	Input Clock Source		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL1 output clock P	MOSC (Crystal resonator is connected.)	Oscillation continues.*1	t_{SBYPC}	—	132	194	µs	PLL1P = 300 MHz ICLK = MRICLK = PCLKE = 300 MHz PCLKA = PCLKD = MRPCLK = BCLK = 150 MHz PCLKB = 75 MHz (divide-by-4)
		Oscillation stops.*1		—	2209	2499		
	MOSC (External clock is connected.)		t_{SBYPE}	—	132.5	194		
	HOCO	Oscillation continues.*2	t_{SBYPH}	—	132.5	194		
		Oscillation stops.*2		—	142.7	205		
Main clock oscillator (MOSC)	Crystal resonator	Oscillation continues.*1	t_{SBYMC}	—	24.3	62.4	µs	Main clock = 48 MHz ICLK = 48 MHz
		Oscillation stops.*1		—	2091	2368		
	External clock		t_{SBYEX}	—	24.3	62.4		
High-speed on-chip oscillator (HOCO)		Oscillation continues.*2	t_{SBYHO}	—	22.9	25.7	µs	High-speed clock = 48 MHz ICLK = 48 MHz
		Oscillation stops.*2		—	25.4	70.1		
Sub-clock oscillator (SOSC)			t_{SBYSC}	—	569	569	µs	Sub-clock = 32.768 kHz ICLK = 32.768 kHz
Middle-speed on-chip oscillator (MOCO)			t_{SBYMO}	—	23.9	26.7	µs	Middle-speed clock = 8 MHz ICLK = 8 MHz

Note 1. The setting of the MOSCSOKP bit in the main clock oscillator standby control register (MOSCSCR) determines this. When MOSCSOKP = 1, oscillation will be continuing. When MOSCSOKP = 0, oscillation will be stopped.
Note 2. The setting of the HCSTP bit in the high-speed on-chip oscillator control register (HOCOCR) determines this. When HCSTP = 0, oscillation will be continuing. When HCSTP = 1, oscillation will be stopped.

Table 67.65 Formula for Calculating the Time to Return from Software Standby Mode 1

Item	Input Clock Source		Symbol	Formula (Typ.)	Formula (Max.)
PLL1 output clock P	MOSC (Crystal resonator is connected.)	Oscillation continues.*1	t_{SBYPC}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
		Oscillation stops.*1		FG + MAX (($t_{MAINOSCWT}$ + 118.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (($t_{MAINOSCWT}$ + 178), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	MOSC (External clock is connected.)		t_{SBYPE}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	HOCO	Oscillation continues.*2	t_{SBYPH}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
		Oscillation stops.*2		FG + MAX (140, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (202, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
Main clock oscillator (MOSC)	Crystal resonator	Oscillation continues.*1	t_{SBYMC}	FG + MAX (oscillation stabilization time (11.5), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))	FG + MAX (oscillation stabilization time (59.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))
		Oscillation stops.*1		FG + MAX ($t_{MAINOSCWT}$, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (($t_{MAINOSCWT}$ + 46.6), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	External clock		t_{SBYEX}	FG + MAX (oscillation stabilization time (11.5), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))	FG + MAX (oscillation stabilization time (59.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))
High-speed on-chip oscillator (HOCO)		Oscillation continues.*2	t_{SBYHO}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{HOCO}$))	
		Oscillation stops.*2		FG + MAX (20, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (67, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
Sub-clock oscillator (SOSC)			t_{SBYSC}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{SOSC}$))	
Middle-speed on-chip oscillator (MOCO)			t_{SBYMO}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOCO}$))	

The symbols used in the table above represent the following value and calculation results.

n: The larger of the frequency division values set by the SCKDIVCR and SCKDIVCR2 registers.
Typ. value: FG = $14/f_{MOCO}$ + $2/f_{ICLK}$
Max. value: FG = $14/f_{MOCOmin}$ + $2/f_{ICLK}$
Typ. value: B = Oscillation stabilization time/f_{MOCO} + $158/f_{MOCO}$ + 0.3
Max. value: B = Oscillation stabilization time/$f_{MOCOmin}$ + $158/f_{MOCO}$ + 0.59
Typ. value: C = (10.5 + 2.5n)/f_{MOCO} + $2.5/f_{SRCCLK}$ + $2/f_{ICLK}$
Max. value: C is obtained by the same formula as for the typ. value.
Typ. value: D = $7/f_{MOCO}$ + $8/f_{ICLK}$
Max. value: D = $7/f_{MOCOmin}$ + $8/f_{ICLK}$
Typ. value: H = $2/f_{ICLK}$
Max. value: H is obtained by the same formula as for the typ. value.

$t_{MAINOSCWT}$: Main clock oscillation stabilization wait time

Note: In the formulae above, when no distinction is made between Typ. and Max., use the same formula for the calculation.

Note 1. The setting of the MOSCSOKP bit in the main clock oscillator standby control register (MOSCSCR) determines this. When MOSCSOKP = 1, oscillation will be continuing. When MOSCSOKP = 0, oscillation will be stopped.

Note 2. The setting of the HCSTP bit in the high-speed on-chip oscillator control register (HOCOCR) determines this. When HCSTP = 0, oscillation will be continuing. When HCSTP = 1, oscillation will be stopped.

Table 67.66 Time to Return from Software Standby Mode 2

Conditions: The SS2FR bit in software standby control register 1 (SSCR1) is set to 1 (fast return function is enabled) and the SS2LP[1:0] bits are set to SS2LP_1 (low power setting 1).
The VSCM[2:0] bits in the voltage scaling control register (VSCR) are set to 010b (0.90 V).
The SVMCM[2:0] bits in the SSTBY voltage scaling control register (SVSCR) are set to 010b (0.90 V).
The LDOSTP bits in both of the PLLn-LDO control registers (PLLnLDOCR, n = 1, 2) are set to 1 (LDO is stopped).

Item	Input Clock Source		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL1 output clock P	MOSC (Crystal resonator is connected.)	Oscillation continues.*1	t_{SBYPC}	—	294	355	µs	PLL1P = 300 MHz ICLK = MRICLK = PCLKE = 300 MHz PCLKA = PCLKD = MRPCLK = BCLK = 150 MHz PCLKB = 75 MHz (divide-by-4)
		Oscillation stops.*1		—	2370	2660		
	MOSC (External clock is connected.)		t_{SBYPE}	—	294	355		
	HOCO	Oscillation continues.*2	t_{SBYPH}	—	294	355		
		Oscillation stops.*2		—	304	367		
Main clock oscillator (MOSC)	Crystal resonator	Oscillation continues.*1	t_{SBYMC}	—	202	251	µs	Main clock = 48 MHz ICLK = 48 MHz
		Oscillation stops.*1		—	2252	2529		
	External clock		t_{SBYEX}	—	202	251		
High-speed on-chip oscillator (HOCO)		Oscillation continues.*2	t_{SBYHO}	—	200	243	µs	High-speed clock = 48 MHz ICLK = 48 MHz
		Oscillation stops.*2		—	203	252		
Sub-clock oscillator (SOSC)			t_{SBYSC}	—	729	729	µs	Sub-clock = 32.768 kHz ICLK = 32.768 kHz
Middle-speed on-chip oscillator (MOCO)			t_{SBYMO}	—	201	244	µs	Middle-speed clock = 8 MHz ICLK = 8 MHz

Note 1. The setting of the MOSCSOKP bit in the main clock oscillator standby control register (MOSCSCR) determines this. When MOSCSOKP = 1, oscillation will be continuing. When MOSCSOKP = 0, oscillation will be stopped.
Note 2. The setting of the HCSTP bit in the high-speed on-chip oscillator control register (HOCOCR) determines this. When HCSTP = 0, oscillation will be continuing. When HCSTP = 1, oscillation will be stopped.

Table 67.67 Formula for Calculating the Time to Return from Software Standby Mode 2

Item	Input Clock Source		Symbol	Formula (Typ.)	Formula (Max.)
PLL1 output clock P	MOSC (Crystal resonator is connected.)	Oscillation continues.*1	t_{SBYPC}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
		Oscillation stops.*1		FG + MAX (($t_{MAINOSCWT}$ + 118.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (($t_{MAINOSCWT}$ + 178), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	MOSC (External clock is connected.)		t_{SBYPE}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	HOCO	Oscillation continues.*2	t_{SBYPH}	FG + MAX (oscillation stabilization time (129.8), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (oscillation stabilization time (190.7), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
		Oscillation stops.*2		FG + MAX (140, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (202, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
Main clock oscillator (MOSC)	Crystal resonator	Oscillation continues.*1	t_{SBYMC}	FG + MAX (oscillation stabilization time (11.5), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))	FG + MAX (oscillation stabilization time (59.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))
		Oscillation stops.*1		FG + MAX ($t_{MAINOSCWT}$, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (($t_{MAINOSCWT}$ + 46.6), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
	External clock		t_{SBYEX}	FG + MAX (oscillation stabilization time (11.5), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))	FG + MAX (oscillation stabilization time (59.3), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOSC}$))
High-speed on-chip oscillator (HOCO)		Oscillation continues.*2	t_{SBYHO}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{HOCO}$))	
		Oscillation stops.*2		FG + MAX (20, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))	FG + MAX (67, B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{PLL}$))
Sub-clock oscillator (SOSC)			t_{SBYSC}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{SOSC}$))	
Middle-speed on-chip oscillator (MOCO)			t_{SBYMO}	FG + MAX (oscillation stabilization time (0), B, C) + MAX (D, H + ($2/f_{ICLK}$ + $2n/f_{MOCO}$))	

The symbols used in the table above represent the following value and calculation results.

n: The larger of the frequency division values set by the SCKDIVCR and SCKDIVCR2 registers.
Typ. value: FG = CYC + $2/f_{ICLK}$
Max. value: FG = CYC + $2/f_{ICLK}$ (See Table 67.68 for CYC values.)
Typ. value: B = Oscillation stabilization time/f_{MOCO} + $193/f_{MOCO}$ + 11.6
Max. value: B = Oscillation stabilization time/$f_{MOCOmin}$ + $193/f_{MOCO}$ + 51
Typ. value: C = $(10.5 + 2.5n)/f_{MOCO}$ + $2.5/f_{SRCCLK}$ + $2/f_{ICLK}$
Max. value: C is obtained by the same formula as for the typ. value.
Typ. value: D = $18/f_{MOCO}$ + $9/f_{ICLK}$
Max. value: D = $18/f_{MOCOmin}$ + $9/f_{ICLK}$
Typ. value: H = $2/f_{ICLK}$
Max. value: H is obtained by the same formula as for the typ. value.

$t_{MAINOSCWT}$: Main clock oscillation stabilization wait time

Note: In the formulae above, when no distinction is made between Typ. and Max., use the same formula for the calculation.

Note 1. The setting of the MOSCSOKP bit in the main clock oscillator standby control register (MOSCSCR) determines this. When MOSCSOKP = 1, oscillation will be continuing. When MOSCSOKP = 0, oscillation will be stopped.

Note 2. The setting of the HCSTP bit in the high-speed on-chip oscillator control register (HOCOCR) determines this. When HCSTP = 0, oscillation will be continuing. When HCSTP = 1, oscillation will be stopped.

Table 67.68 CYC Value

SSCR1.SS2LP[1:0]	VSCR.VSCM[2:0]	SVSCR.SVSCM[2:0]	CYC	Unit	Remarks
SS2LP_0	VSCR_1	SVSCR_1	237	Cycles	*1
		SVSCR_2	379		—
		SVSCR_3	591		
		SVSCR_4	696		
		SVSCR_5	802		
	VSCR_2	SVSCR_1	379		—
		SVSCR_2	237		*1
		SVSCR_3	538		—
		SVSCR_4	643		
		SVSCR_5	749		
SS2LP_1	VSCR_1	SVSCR_2	514	Cycles	—
		SVSCR_3	726		
		SVSCR_4	831		
		SVSCR_5	937		
	VSCR_2	SVSCR_2	459		*2
		SVSCR_3	673		—
		SVSCR_4	778		
		SVSCR_5	884		

Note 1. When the LDOSTP bits in both of the PLLn-LDO control registers (PLLnLDOCR, n = 1, 2) are set to 1 (LDO is stopped) or the SKEEP bits in both registers are set to 1 (the PLLn-LDO state before software standby mode 2 is retained), CYC is 56 if both the LDOSTP bit and SKEEP bit in the HOCO/LCO control register (HOCOLDOCR) are set to 0; CYC is 27 if either of the bits is set to1.

Note 2. When the SS2FR bit in software standby control register 1 (SSCR1) is set to 1 (fast return function is enabled), CYC is 162 if the LDOSTP bits in both of the PLLn-LDO control registers (PLLnLDOCR, n = 1, 2) are set to 1 (LDO is stopped) or the SKEEP bits in both registers are set to 1 (the PLLn-LDO state before software standby mode 2 is retained); otherwise, CYC is 327.

Table 67.69 Time to Return from Deep Software Standby Mode

Item		DPSBYCR Setting	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
DCDC mode	DSTBY1	DCSSMODE = 01b	t_{DBSY}	—	296	346	μs	Figure 67.27
		DCSSMODE = 10b		—	456	506		
		DCSSMODE = 11b		—	776	826		
	DSTBY2	DCSSMODE = 01b		—	296	346		
		DCSSMODE = 10b		—	456	506		
		DCSSMODE = 11b		—	776	826		
	DSTBY3	DCSSMODE = 01b		—	483	604		
		DCSSMODE = 10b		—	643	764		
		DCSSMODE = 11b		—	963	1084		
LDO mode	DSTBY1	LDOSSMODE = 01b	t_{DBSY}	—	208	268	μs	Figure 67.27
		LDOSSMODE = 10b		—	226	307		
		LDOSSMODE = 11b		—	250	353		
	DSTBY2	LDOSSMODE = 01b		—	262	291		
		LDOSSMODE = 10b		—	262	307		
		LDOSSMODE = 11b		—	262	353		
	DSTBY3	LDOSSMODE = 01b		—	449	549		
		LDOSSMODE = 10b		—	449	565		
		LDOSSMODE = 11b		—	449	611		
Waiting time after release from the deep software standby mode			t_{DBSYWT}	22.2	—	33.6	μs	

DSTBY1: Deep software standby mode 1
DSTBY2: Deep software standby mode 2
DSTBY3: Deep software standby mode 3

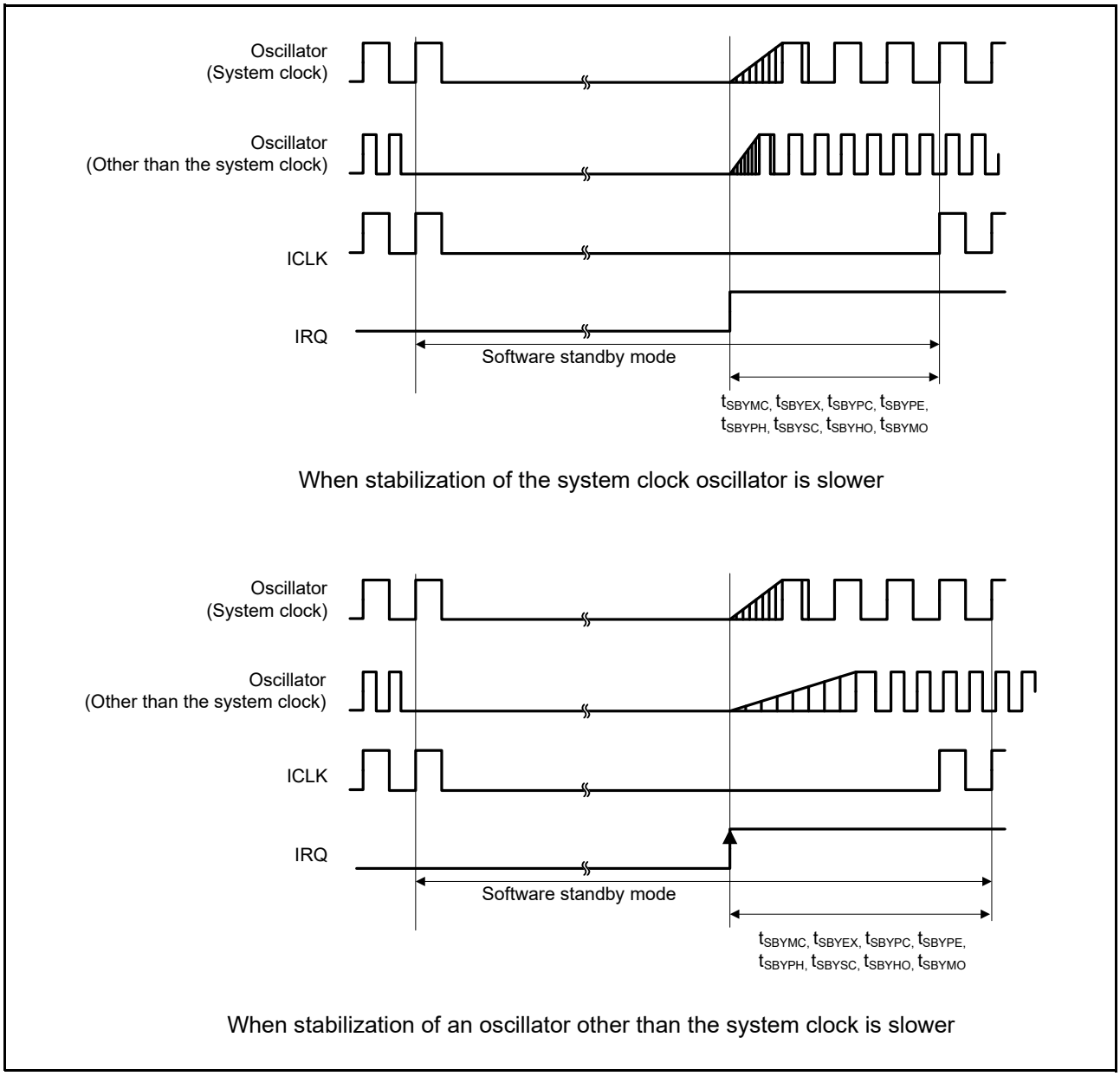


Figure 67.26 Timing of Return from Software Standby Mode

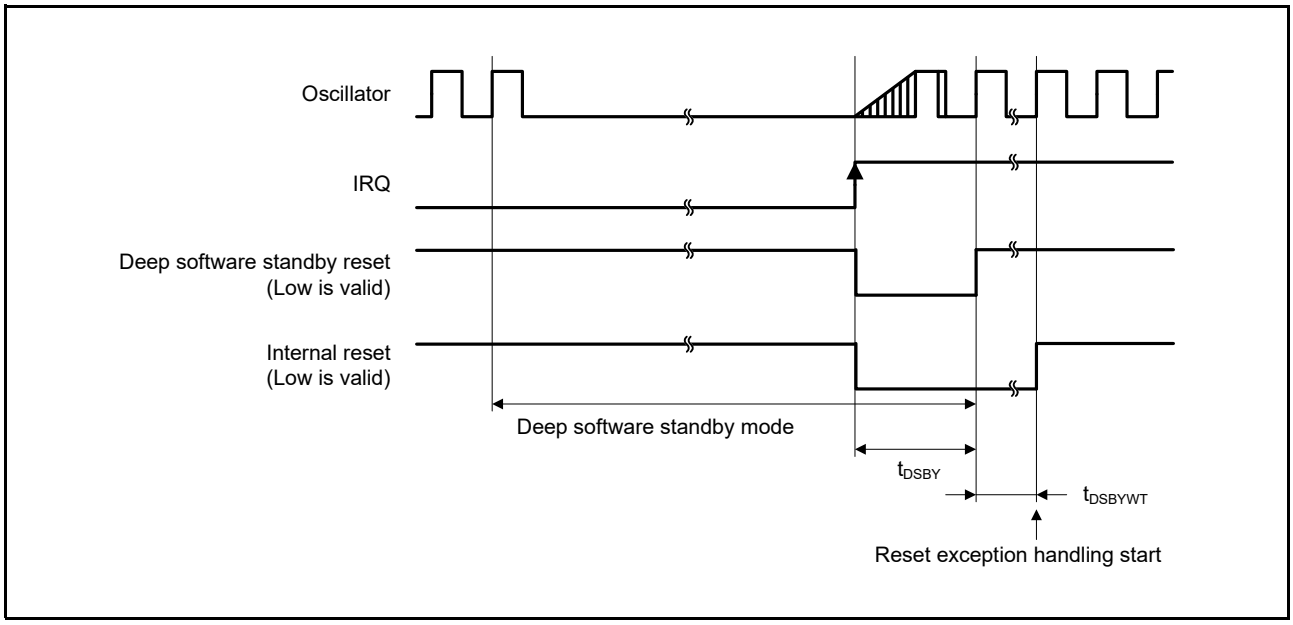


Figure 67.27 Timing of Return from Deep Software Standby Mode

67.5.5 Control Signal Timing

Table 67.70 Control Signal Timing

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
NMI pulse width	NMI digital filter disabled	$t_{PBcyc} \times 2 \leq 200$ ns	t_{NMIW}	200	—	—	ns	Figure 67.28
		$t_{PBcyc} \times 2 > 200$ ns		$t_{PBcyc} \times 2$*1	—	—		
	NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns		200	—	—		
		$t_{NMICK} \times 3 > 200$ ns		$t_{NMICK} \times 3.5$*2	—	—		
IRQ pulse width	IRQ digital filter disabled	$t_{PBcyc} \times 2 \leq 200$ ns	t_{IRQW}	200	—	—	ns	Figure 67.29
		$t_{PBcyc} \times 2 > 200$ ns		$t_{PBcyc} \times 2$*1	—	—		
	IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns		200	—	—		
		$t_{IRQCK} \times 3 > 200$ ns		$t_{IRQCK} \times 3.5$*3	—	—		

Note: In software standby mode, the minimum value is 200 ns.
Note: When switching system clock sources, add 4 clock cycles of the clock source you are switching to.
Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock.

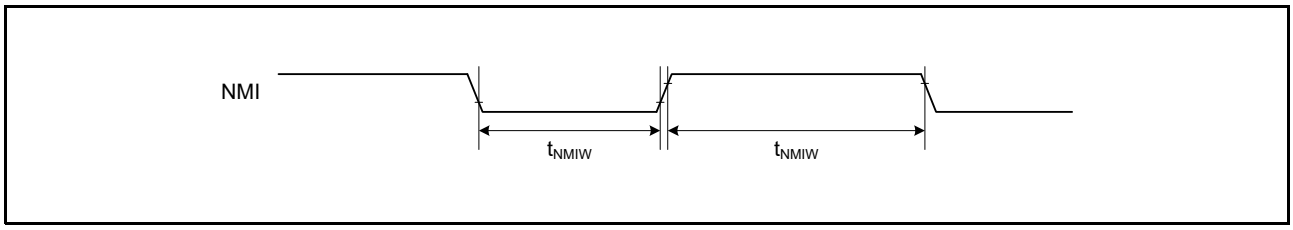


Figure 67.28 NMI Interrupt Input Timing

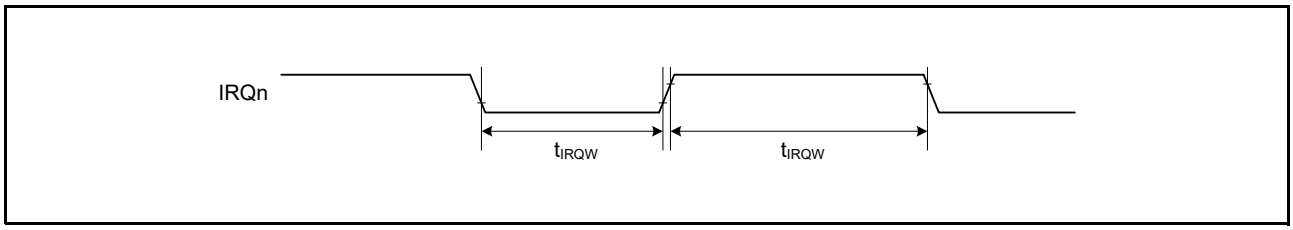


Figure 67.29 IRQ Interrupt Input Timing

67.5.6 Bus Timing

Table 67.71 Bus Timing (1)

Conditions: With the CS area controller in use
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the EBCLK pin are set for DRV7.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than the EBCLK pin are set for DRV3.
EBCLK = 100 MHz,
VCC = VCC2 = 3.0 to 3.63 V, C = 15 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	0.5	10.8	ns	Figure 67.30 to Figure 67.37
Byte control delay time	t_{BCD}	0.5	10.8	ns	
CS# delay time	t_{CSD}	0.5	10.8	ns	
ALE delay time	t_{ALED}	0.5	10.8	ns	
RD# delay time	t_{RSD}	0.5	10.8	ns	
Read data setup time	t_{RDS}	10.8	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	0.5	10.8	ns	
Write data delay time	t_{WDD}	0.5	10.8	ns	
Write data hold time	t_{WDH}	0.5	—	ns	
WAIT# setup time	t_{WTS}	10.8	—	ns	
WAIT# hold time	t_{WTH}	0	—	ns	
BREQ# setup time	t_{BRS}	12.5	—	ns	
BREQ# hold time	t_{BRH}	0	—	ns	
BACK# delay time	t_{BAC}	0.5	12.5	ns	
Bus release time	t_{BRL}	—	12.5	ns	
Bus acquisition time	t_{BAQ}	—	12.5	ns	

Note: This table only applies when pins with the suffix -A, indicating the group name, as part of the pin name are in use.
Note: The above characteristics are only guaranteed for products with part numbers of the form R5K574Nxx4yy (yy = BG, BD, AE).

Table 67.72 Bus Timing (2)

Conditions: With the CS area controller in use
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the EBCLK pin are set for DRV7.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than the EBCLK pin are set for DRV3.
EBCLK = 60 MHz,
VCC = VCC2 = 2.7 to 3.63 V, C = 30 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	0.5	12.5	ns	Figure 67.30 to Figure 67.37
Byte control delay time	t_{BCD}	0.5	12.5	ns	
CS# delay time	t_{CSD}	0.5	12.5	ns	
ALE delay time	t_{ALED}	0.5	12.5	ns	
RD# delay time	t_{RSD}	0.5	12.5	ns	
Read data setup time	t_{RDS}	12.5	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	0.5	12.5	ns	
Write data delay time	t_{WDD}	0.5	12.5	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	12.5	—	ns	
WAIT# hold time	t_{WTH}	0	—	ns	
BREQ# setup time	t_{BRS}	12.5	—	ns	
BREQ# hold time	t_{BRH}	0	—	ns	
BACK# delay time	t_{BAC}	—	12.5	ns	
Bus release time	t_{BRL}	—	12.5	ns	
Bus acquisition time	t_{BAQ}	—	12.5	ns	

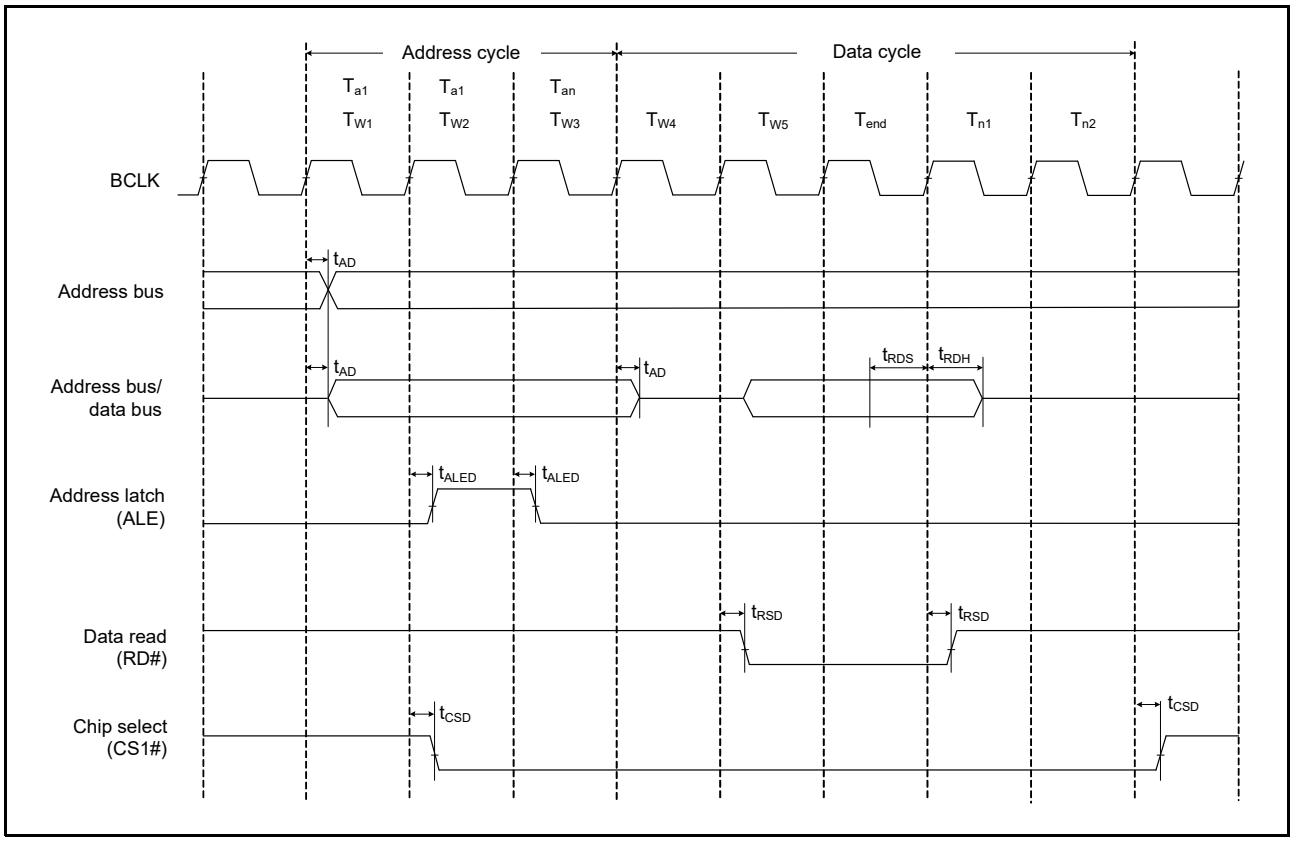


Figure 67.30 Address/Data Multiplexed Bus Read Access Timing

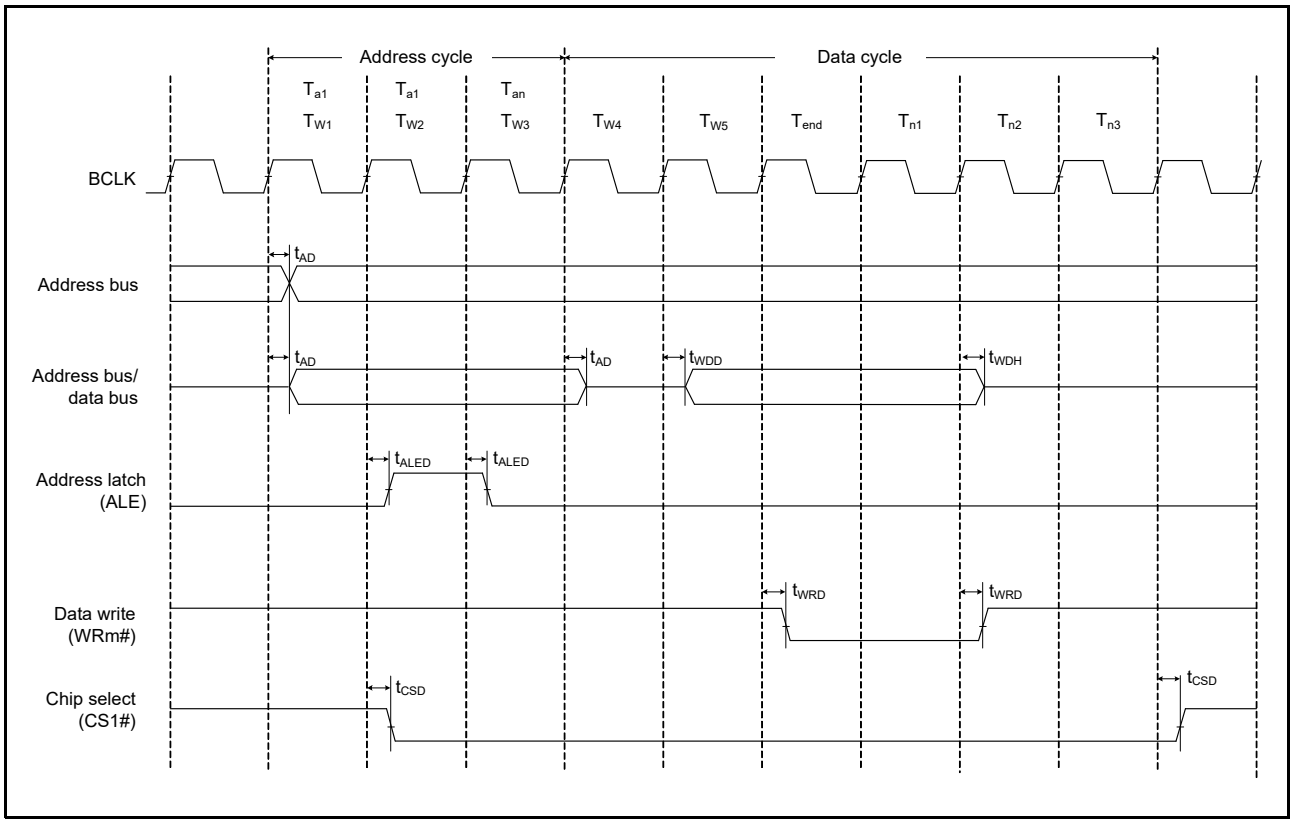


Figure 67.31 Address/Data Multiplexed Bus Write Access Timing

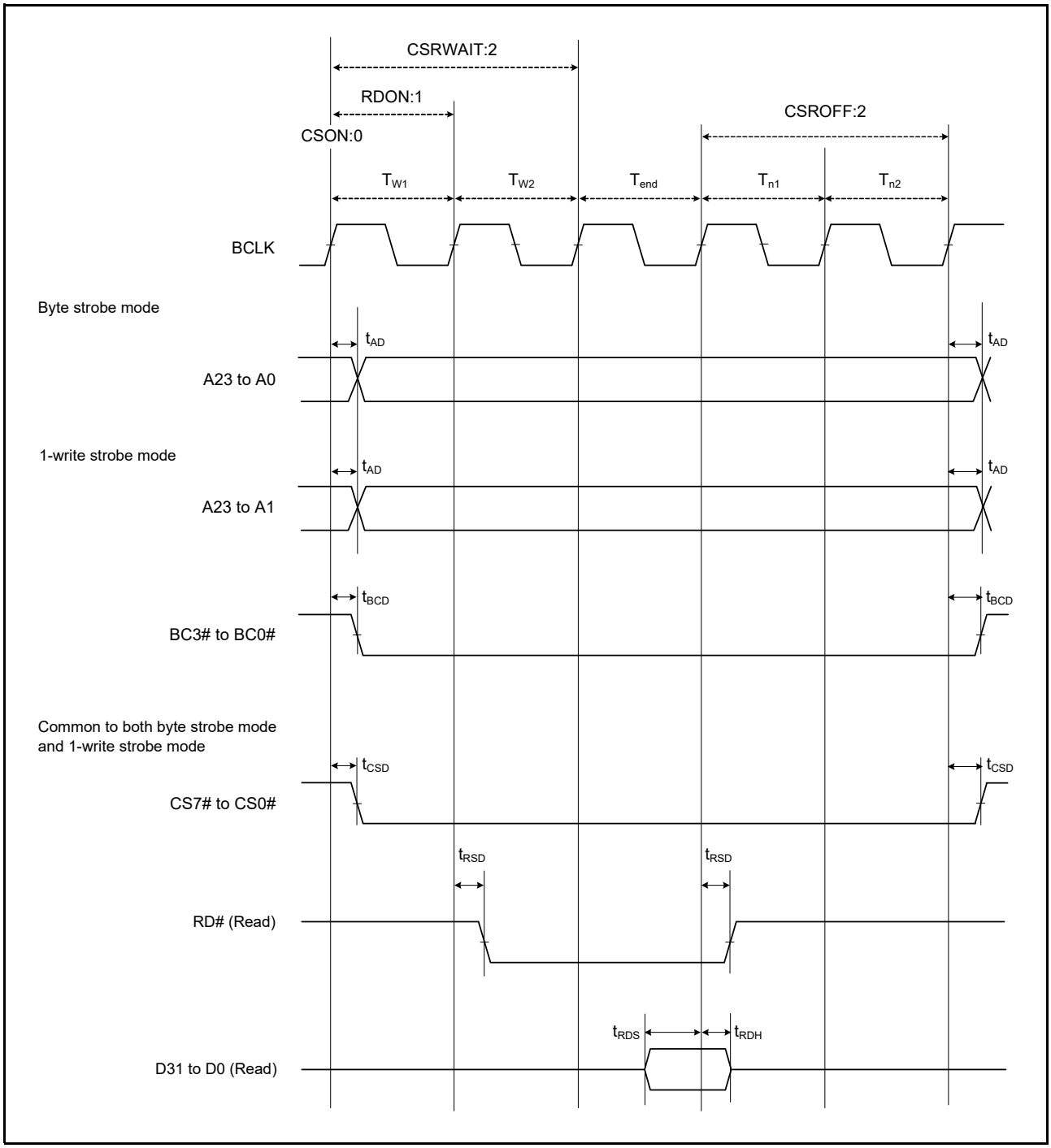
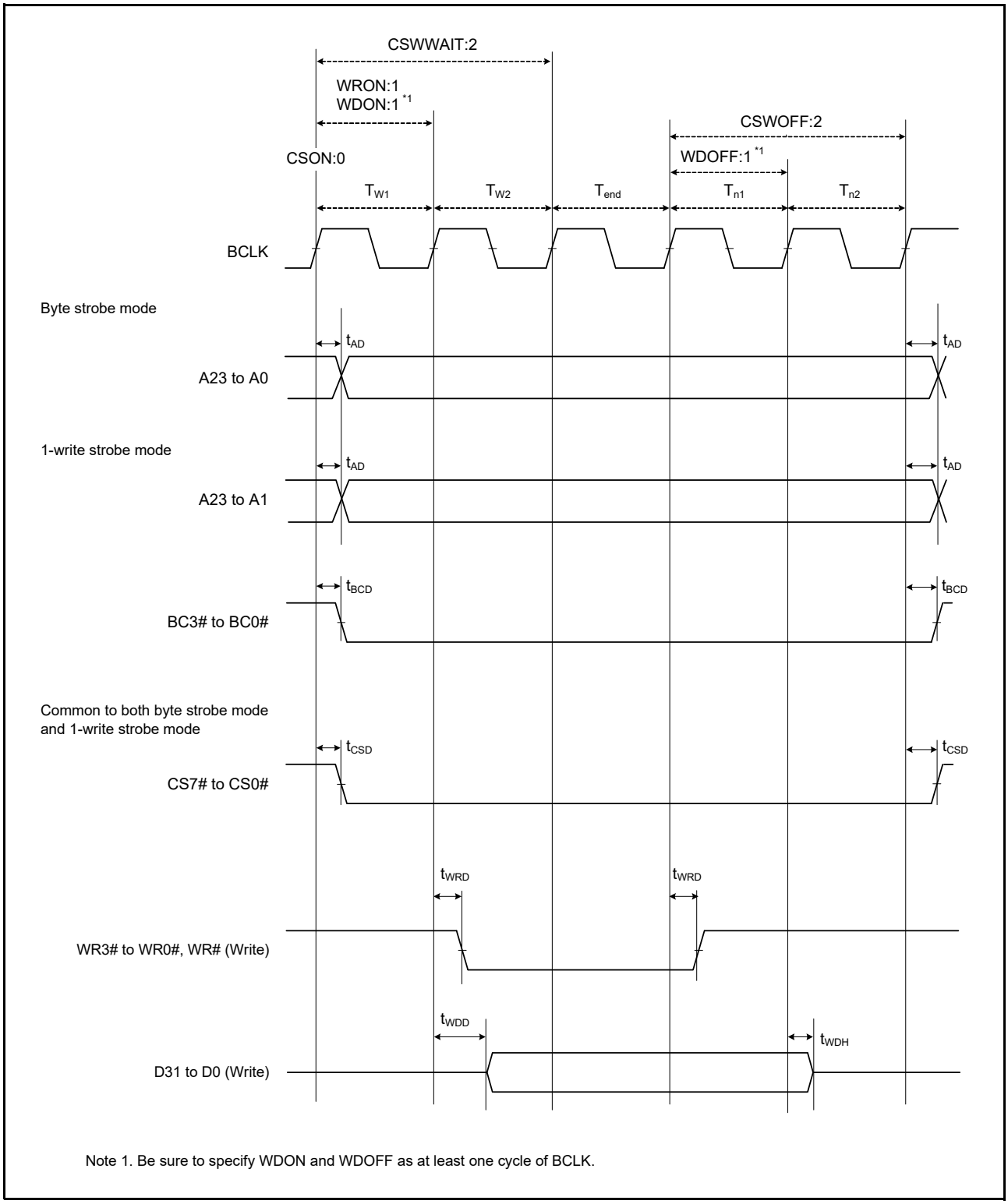


Figure 67.32 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of BCLK.

Figure 67.33 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

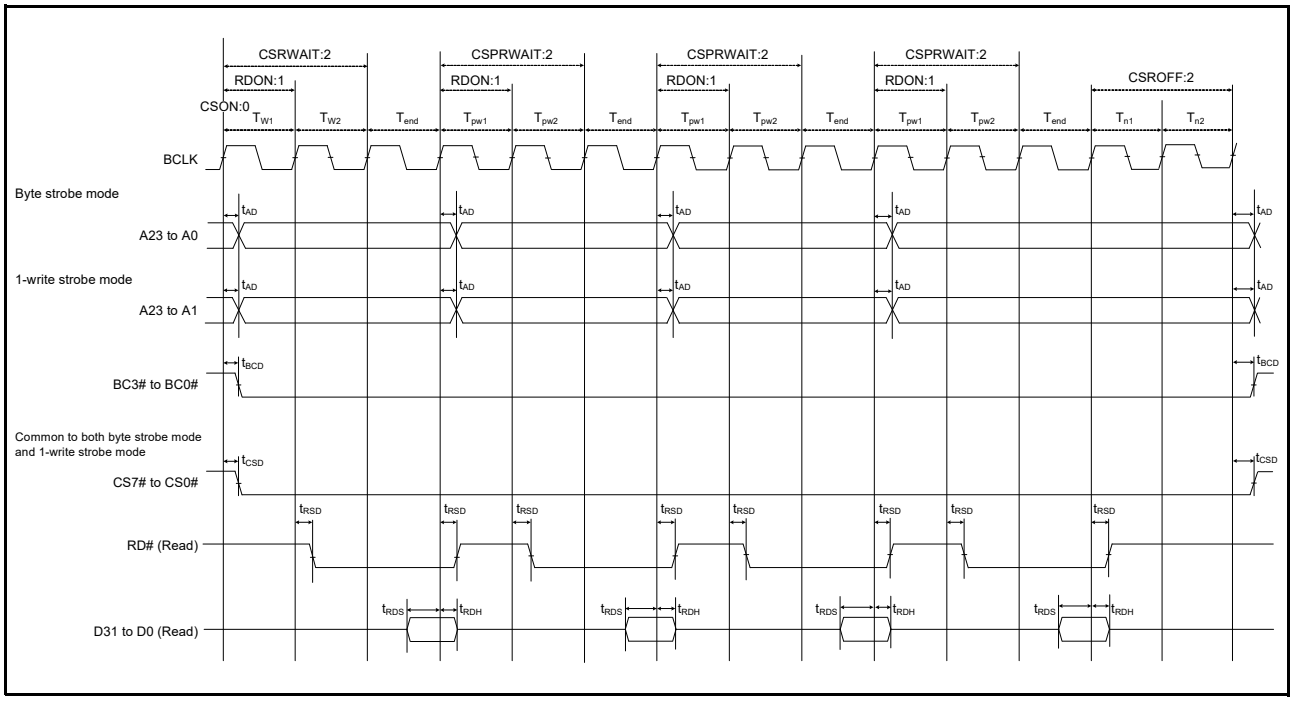


Figure 67.34 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)

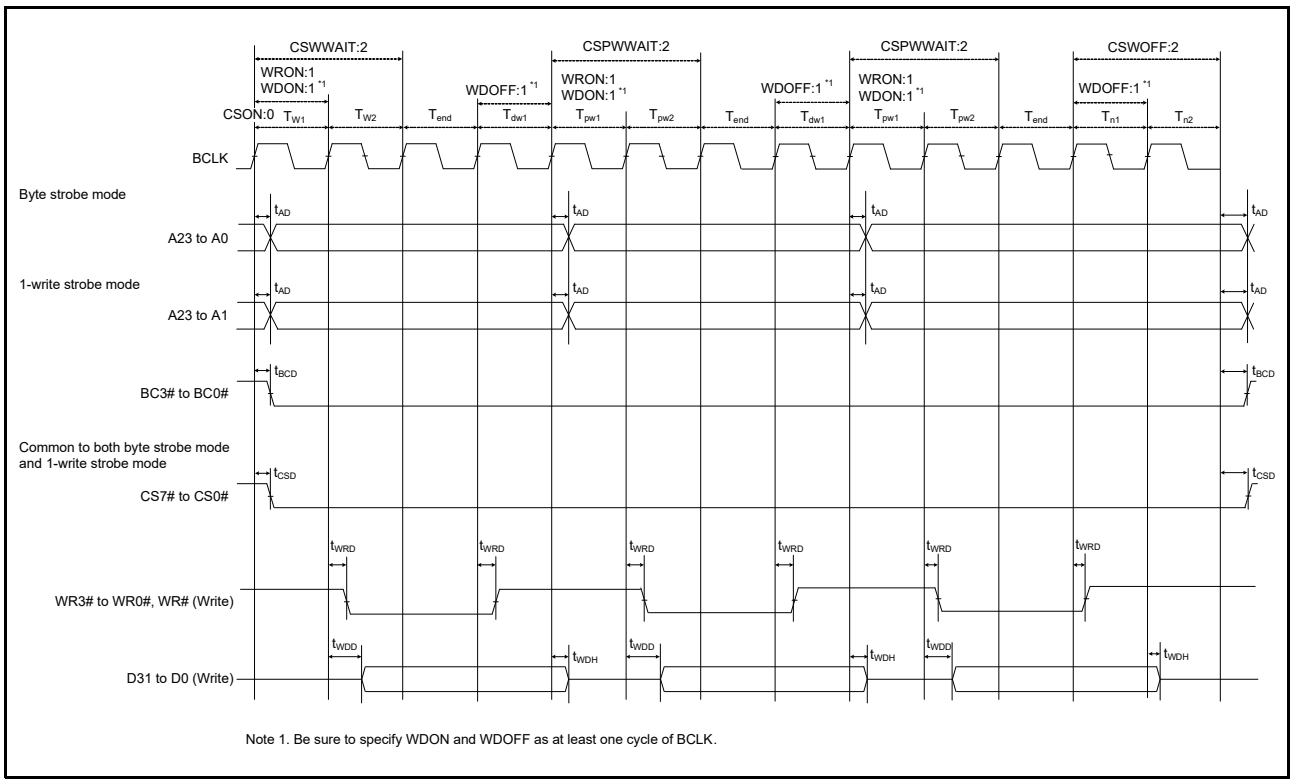


Figure 67.35 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

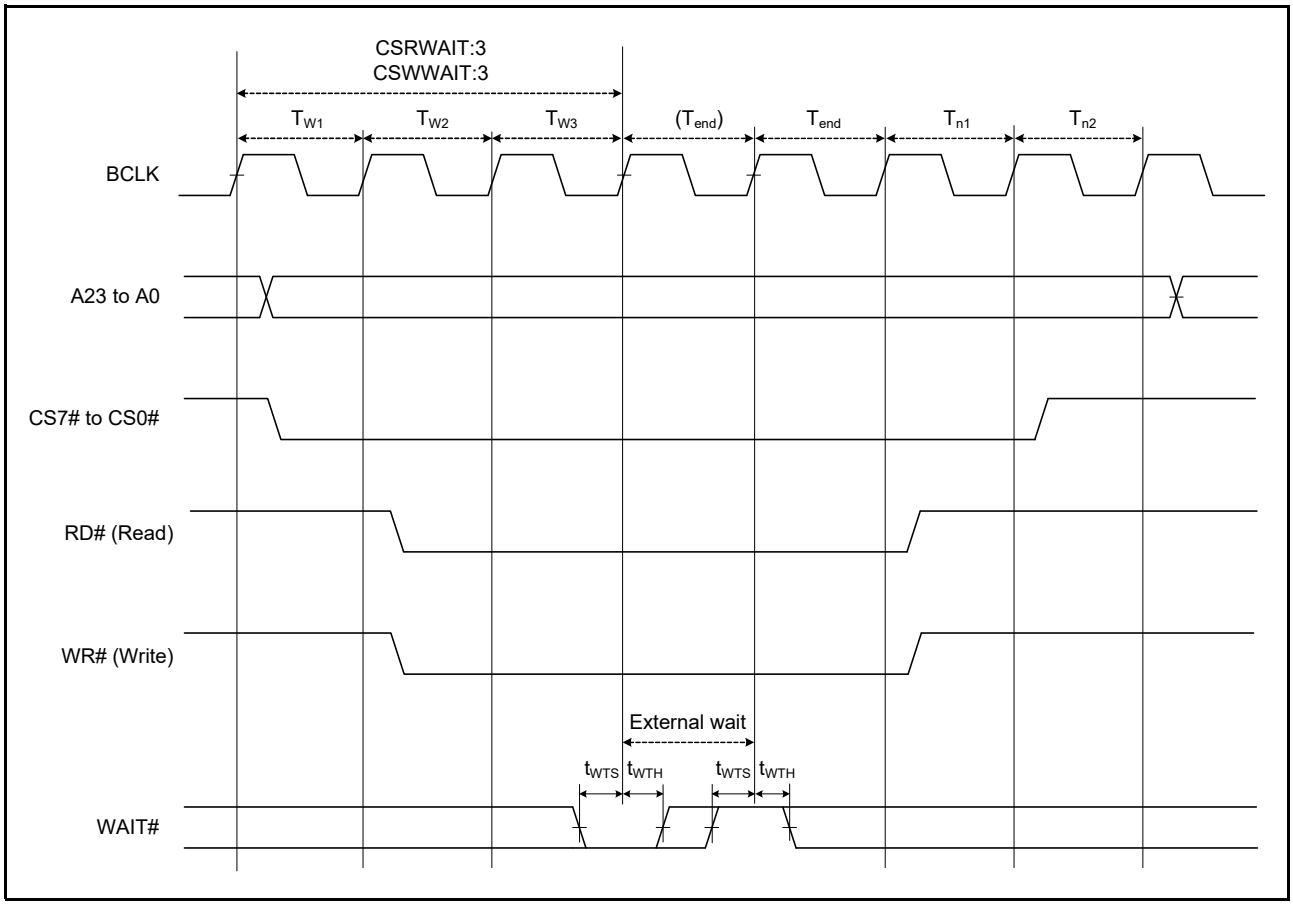


Figure 67.36 External Bus Timing/External Wait Control

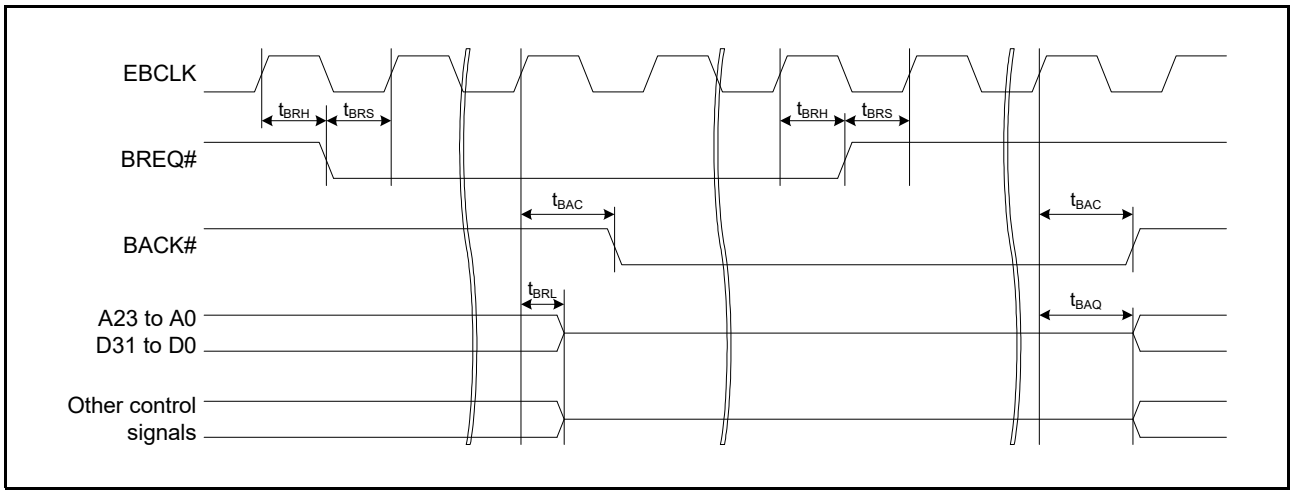


Figure 67.37 External Bus Timing/BREQ, BACK

Table 67.73 Bus Timing (3)

Conditions: With the SDRAM area controller in use
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the SDCLK pin are set for DRV7.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than SDCLK pin are set for DRV3.
SDCLK = 133 MHz,
VCC = VCC2 = 3.0 to 3.63 V, C = 15 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time 2 (SDRAM)	t_{AD2}	0.8	6	ns	Figure 67.38 to Figure 67.45
CS# delay time 2 (SDRAM)	t_{CSD2}	0.8	6	ns	
DQM delay time (SDRAM)	t_{DQMD}	0.8	6	ns	
CKE delay time (SDRAM)	t_{CKED}	0.8	6	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	2.1	—	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t_{WDD2}	—	6	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	0.8	—	ns	
WE# delay time (SDRAM)	t_{WED}	0.8	6	ns	
RAS# delay time (SDRAM)	t_{RASD}	0.8	6	ns	
CAS# delay time (SDRAM)	t_{CASD}	0.8	6	ns	
BREQ# setup time 2 (SDRAM)	t_{BRS2}	12.5	—	ns	
BREQ# hold time 2 (SDRAM)	t_{BRH2}	0.8	—	ns	
BACK# delay time 2 (SDRAM)	t_{BAC2}	0.8	12.5	ns	
Bus release time 2 (SDRAM)	t_{BRL2}	0.8	12.5	ns	
Bus acquisition time 2 (SDRAM)	t_{BAQ2}	0	12.5	ns	

Note: This table only applies when pins with the suffix -A, indicating the group name, as part of the pin name are in use.
Note: The above characteristics are only guaranteed for products with part numbers of the form R5K574Nxx4yy (yy = BG, BD, AE).

Table 67.74 Bus Timing (4)

Conditions: With the SDRAM area controller in use
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the SDCLK pin are set for DRV7.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than SDCLK pin are set for DRV3.
SDCLK = 100 MHz,
VCC = VCC2 = 3.0 to 3.63 V, C = 15 pF

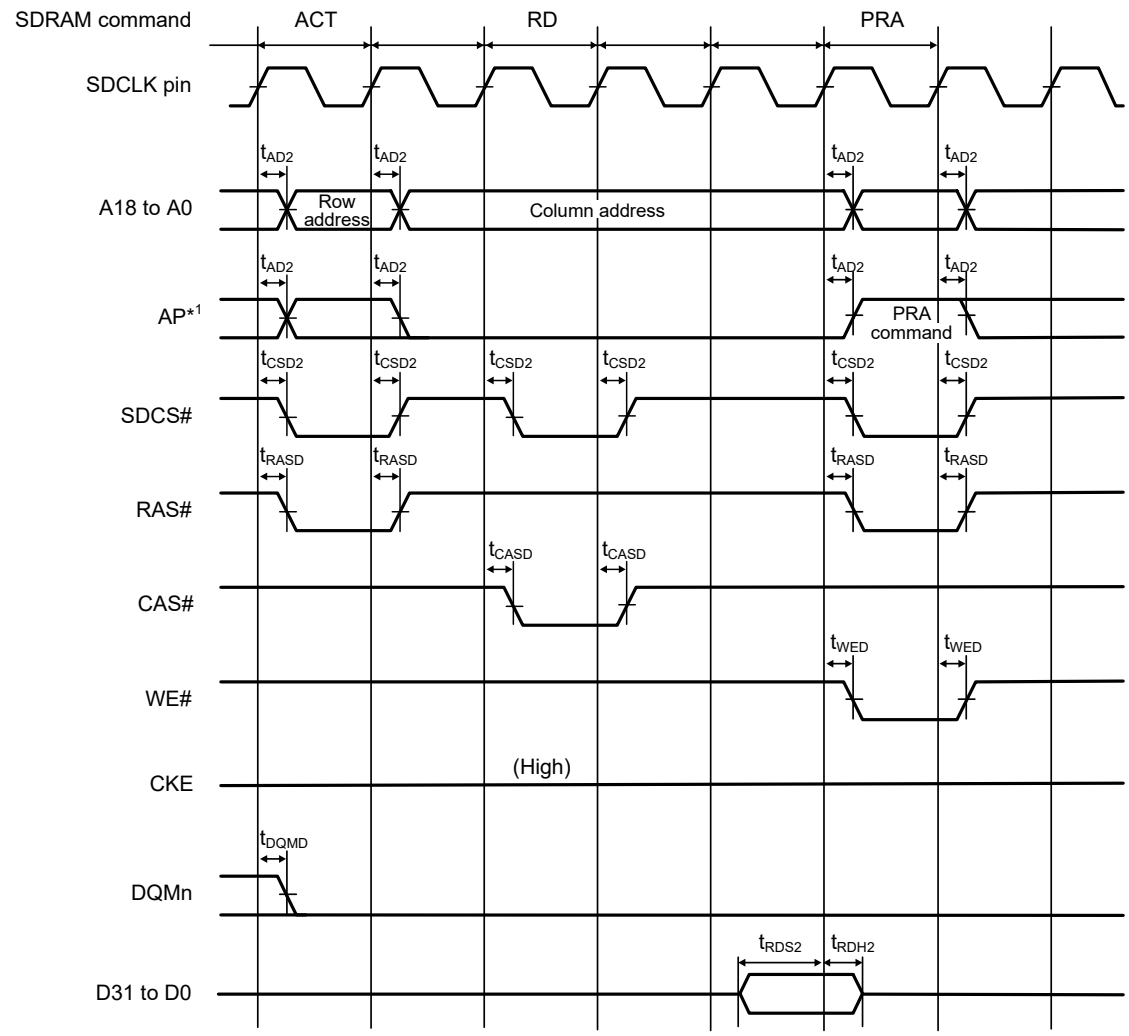
Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time 2 (SDRAM)	t_{AD2}	0.8	8.5	ns	Figure 67.38 to Figure 67.45
CS# delay time 2 (SDRAM)	t_{CSD2}	0.8	8.5	ns	
DQM delay time (SDRAM)	t_{DQMD}	0.8	8.5	ns	
CKE delay time (SDRAM)	t_{CKED}	0.8	8.5	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	4	—	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t_{WDD2}	—	8.5	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	0.8	—	ns	
WE# delay time (SDRAM)	t_{WED}	0.8	8.5	ns	
RAS# delay time (SDRAM)	t_{RASD}	0.8	8.5	ns	
CAS# delay time (SDRAM)	t_{CASD}	0.8	8.5	ns	
BREQ# setup time 2 (SDRAM)	t_{BRS2}	12.5	—	ns	
BREQ# hold time 2 (SDRAM)	t_{BRH2}	0.8	—	ns	
BACK# delay time 2 (SDRAM)	t_{BAC2}	0.8	12.5	ns	
Bus release time 2 (SDRAM)	t_{BRL2}	0.8	12.5	ns	
Bus acquisition time 2 (SDRAM)	t_{BAQ2}	0	12.5	ns	

Note: The above characteristics are only guaranteed for products with part numbers of the form R5K574Nxx4yy (yy = BG, BD, AE).

Table 67.75 Bus Timing (5)

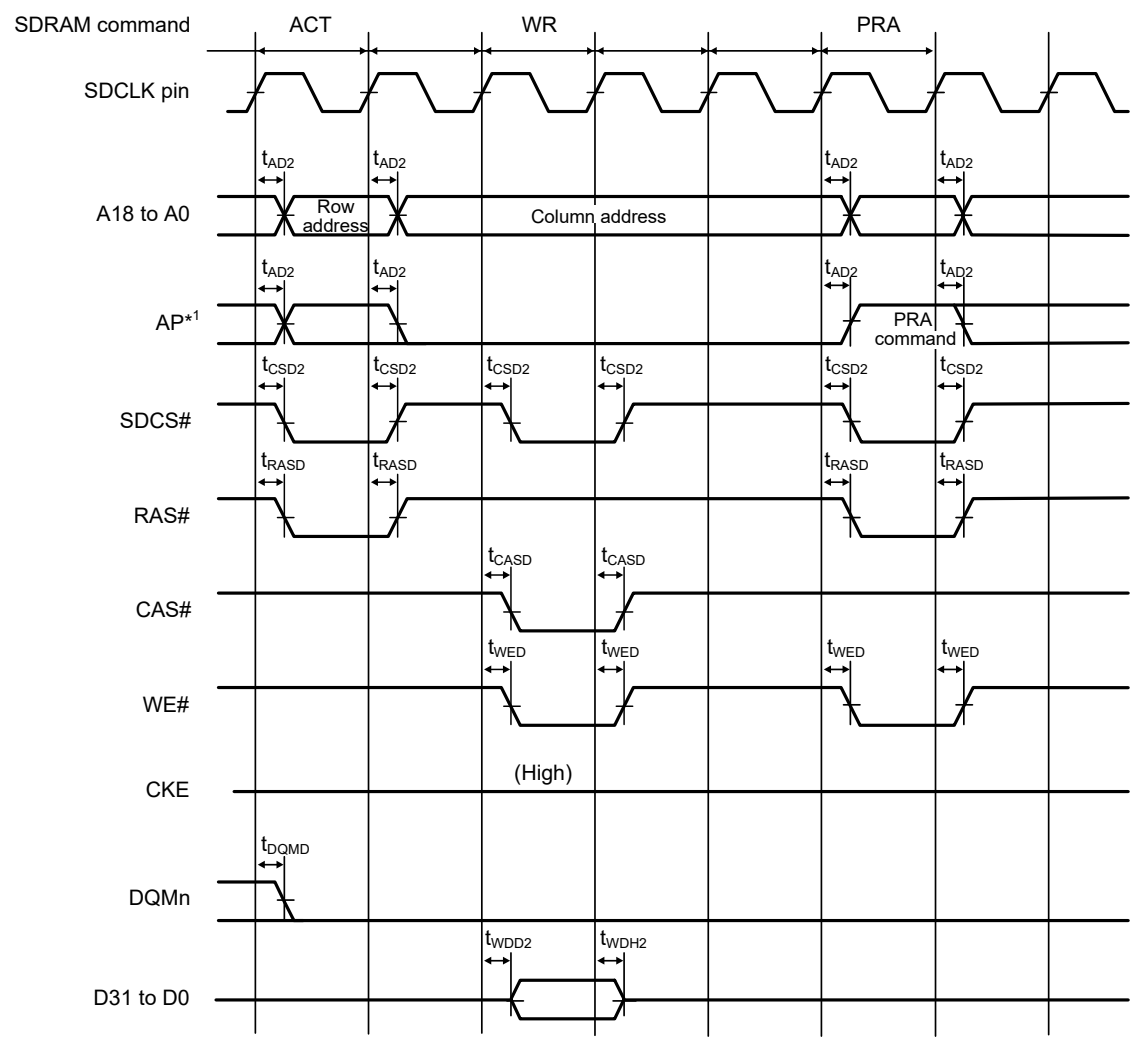
Conditions: With the SDRAM area controller in use
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the SDCLK pin are set for DRV7.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than SDCLK pin are set for DRV3.
SDCLK = 60 MHz,
VCC = VCC2 = 2.70 to 3.63 V, C = 30 pF

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time 2 (SDRAM)	t_{AD2}	0.8	12.5	ns	Figure 67.38 to Figure 67.45
CS# delay time 2 (SDRAM)	t_{CSD2}	0.8	12.5	ns	
DQM delay time (SDRAM)	t_{DQMD}	0.8	12.5	ns	
CKE delay time (SDRAM)	t_{CKED}	0.8	12.5	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	10	—	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t_{WDD2}	—	12.5	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	0.8	—	ns	
WE# delay time (SDRAM)	t_{WED}	0.8	12.5	ns	
RAS# delay time (SDRAM)	t_{RASD}	0.8	12.5	ns	
CAS# delay time (SDRAM)	t_{CASD}	0.8	12.5	ns	
BREQ# setup time 2 (SDRAM)	t_{BRS2}	12.5	—	ns	
BREQ# hold time 2 (SDRAM)	t_{BRH2}	0.8	—	ns	
BACK# delay time 2 (SDRAM)	t_{BAC2}	0.8	12.5	ns	
Bus release time 2 (SDRAM)	t_{BRL2}	0.8	12.5	ns	
Bus acquisition time 2 (SDRAM)	t_{BAQ2}	0	12.5	ns	



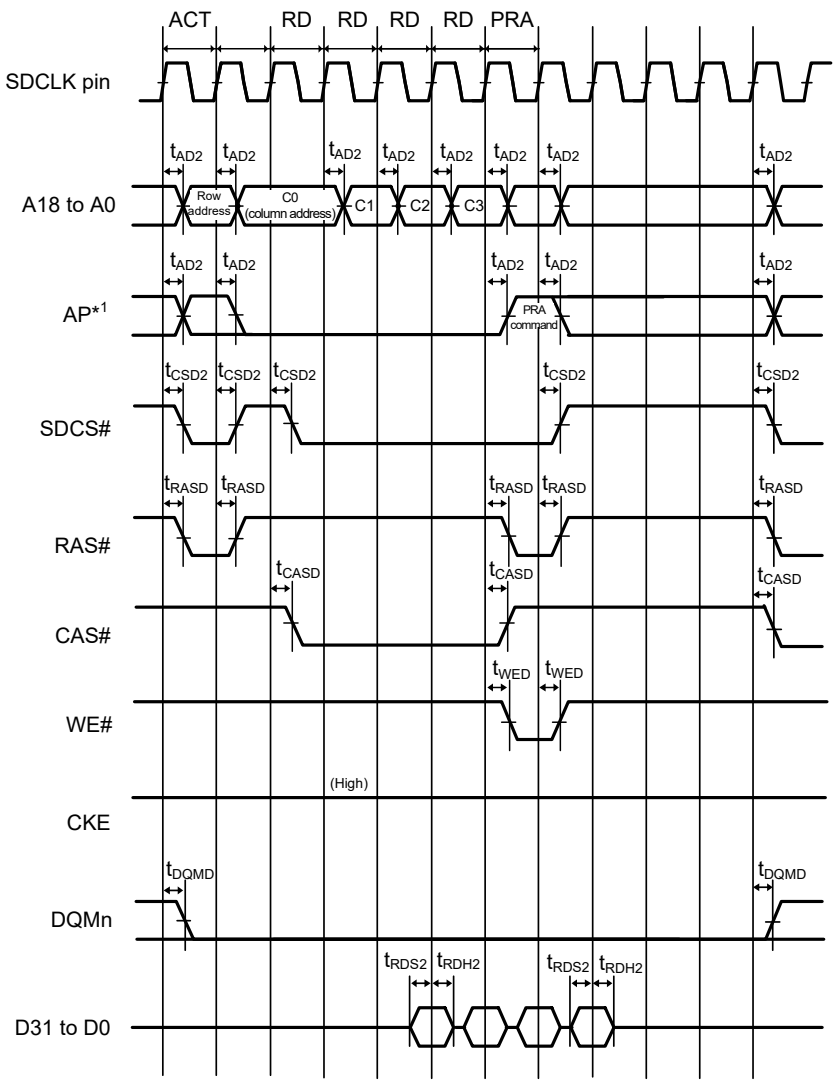
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.38 SDRAM Space Single Read Bus Timing



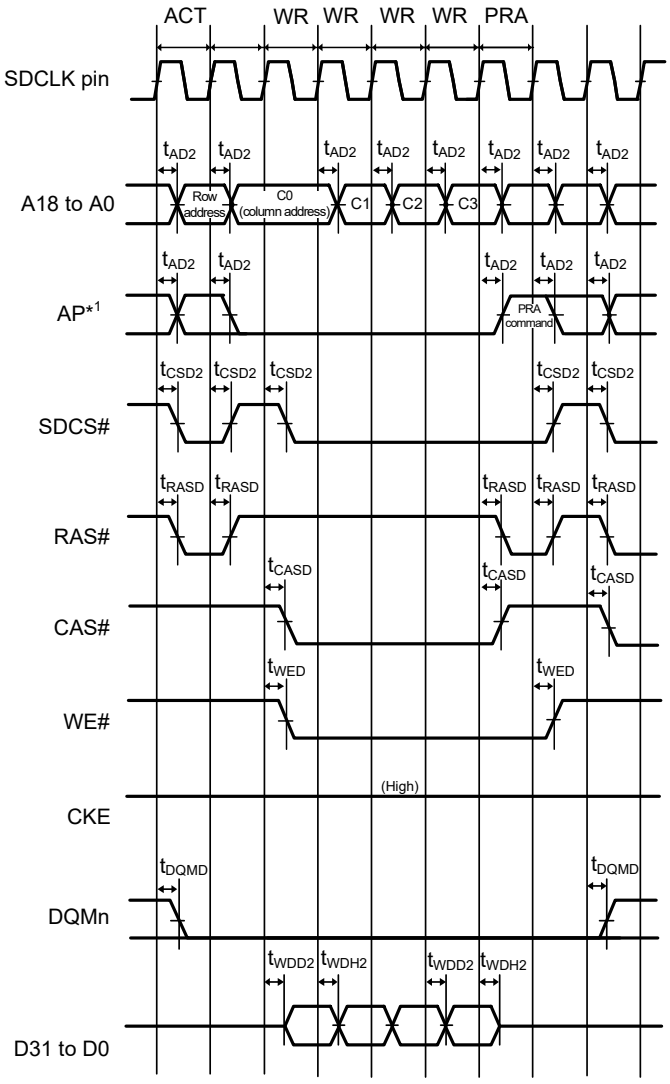
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.39 SDRAM Space Single Write Bus Timing



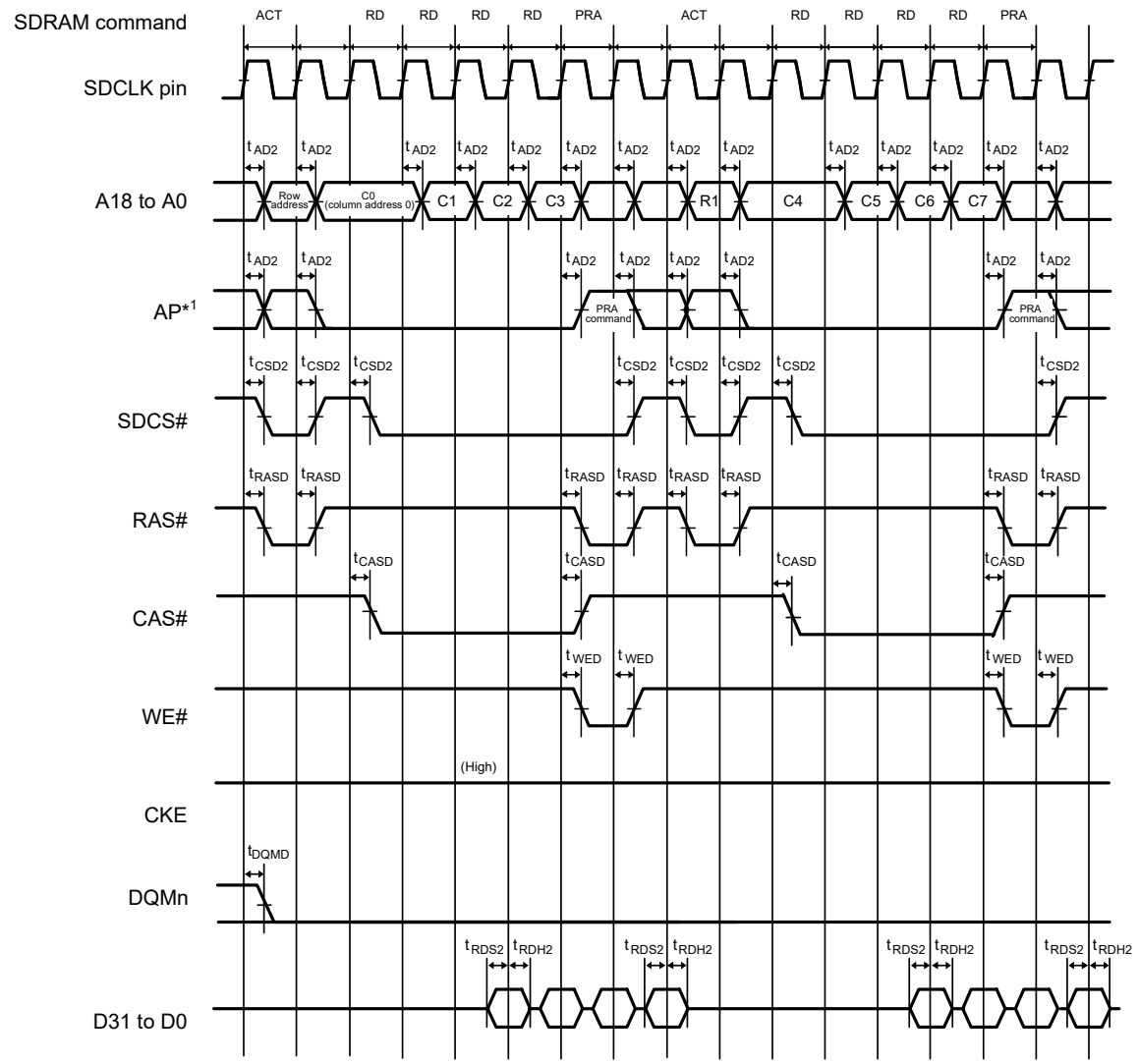
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.40 SDRAM Space Multiple Read Bus Timing



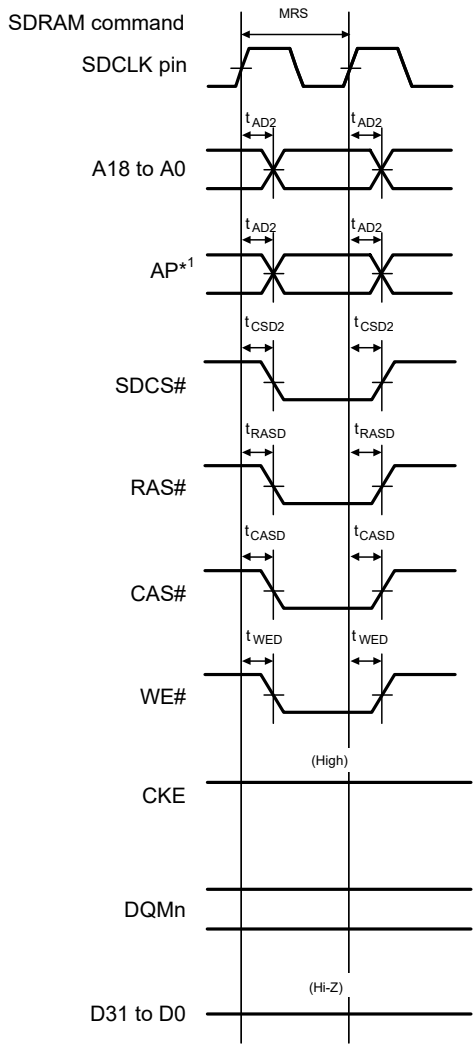
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.41 SDRAM Space Multiple Write Bus Timing



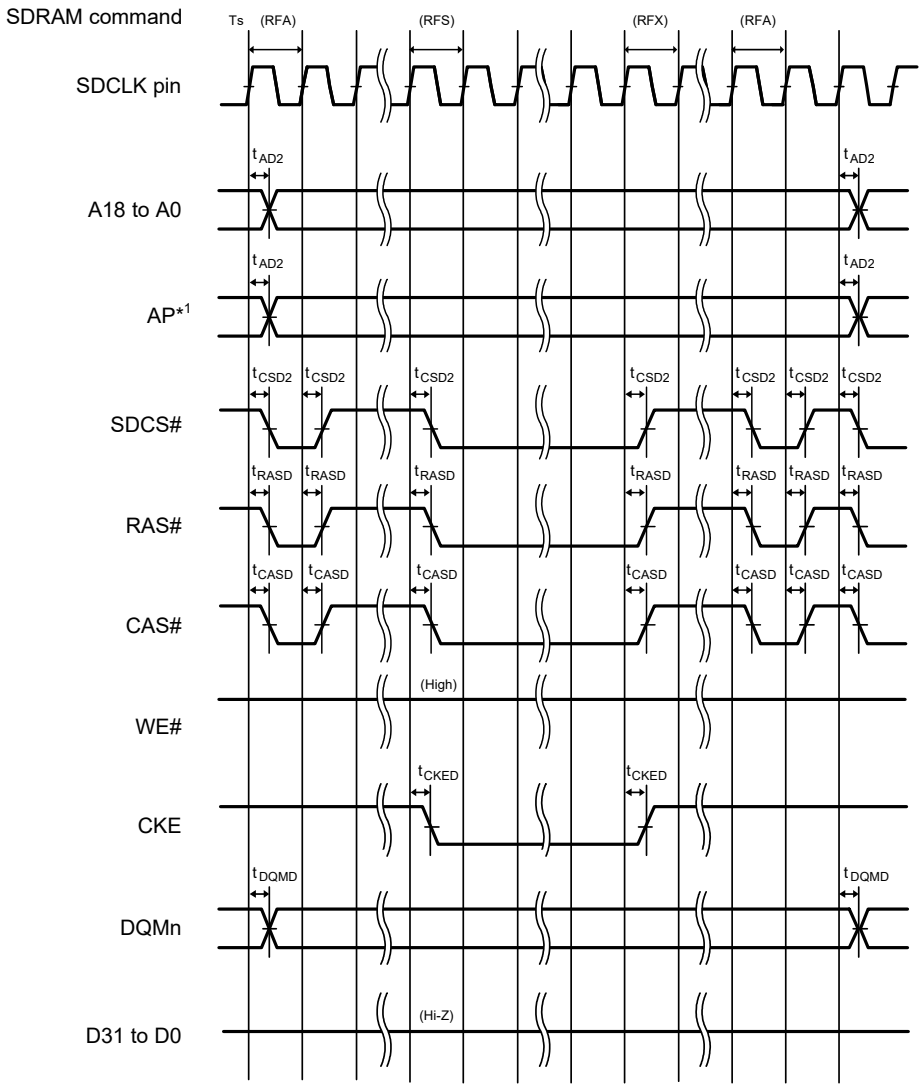
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.42 SDRAM Space Multiple Read Line Stride Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.43 SDRAM Space Mode Register Set Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 67.44 SDRAM Space Self-Refresh Bus Timing

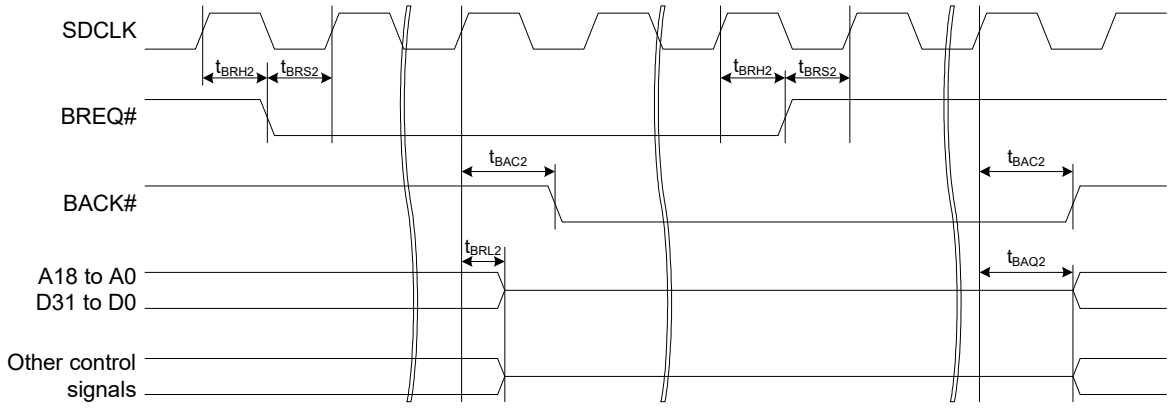


Figure 67.45 External Bus Timing/BREQ, BACK

67.5.7 Timing of On-Chip Peripheral Modules

67.5.7.1 I/O Port

Table 67.76 I/O Port Timing

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	7.5	—	t_{Icyc}	Figure 67.46

Note 1. t_{Icyc}: ICLK cycle

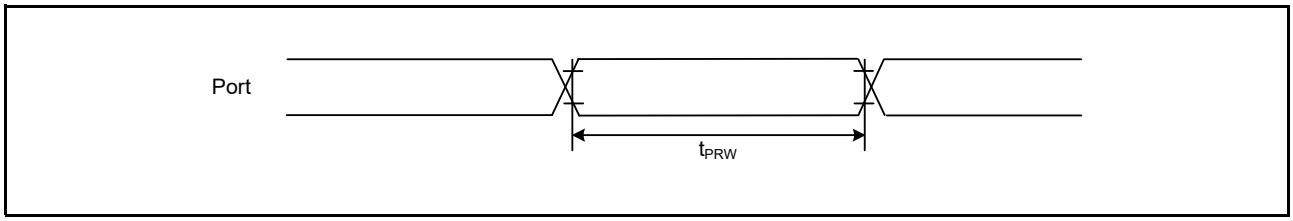


Figure 67.46 I/O Port Input Timing

67.5.7.2 TMR

Table 67.77 TMR Timing

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{PBcyc}	Figure 67.47
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

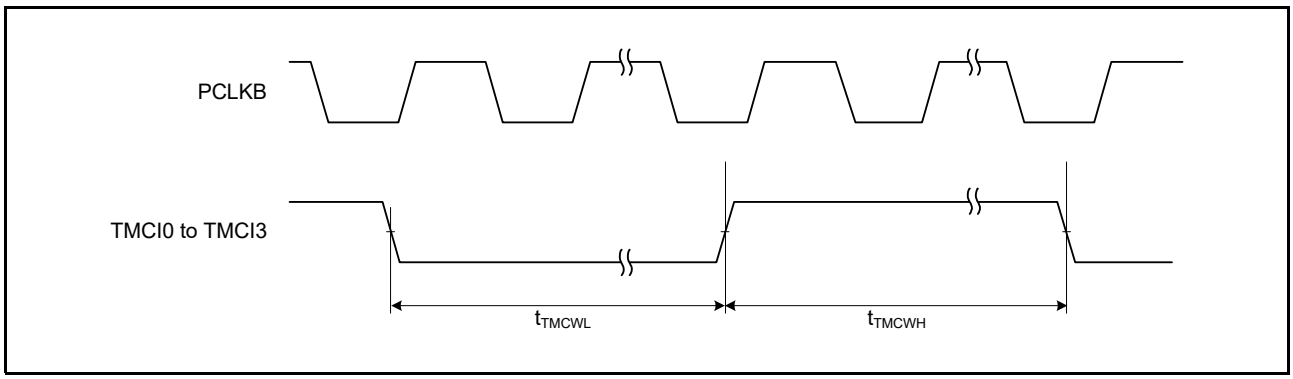


Figure 67.47 TMR Clock Input Timing

67.5.7.3 CMTW

Table 67.78 CMTW Timing

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
CMTW	Input capture input pulse width	Single-edge setting	$t_{CMTWTICW}$	1.5	—	t_{PBcyc}	Figure 67.48
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

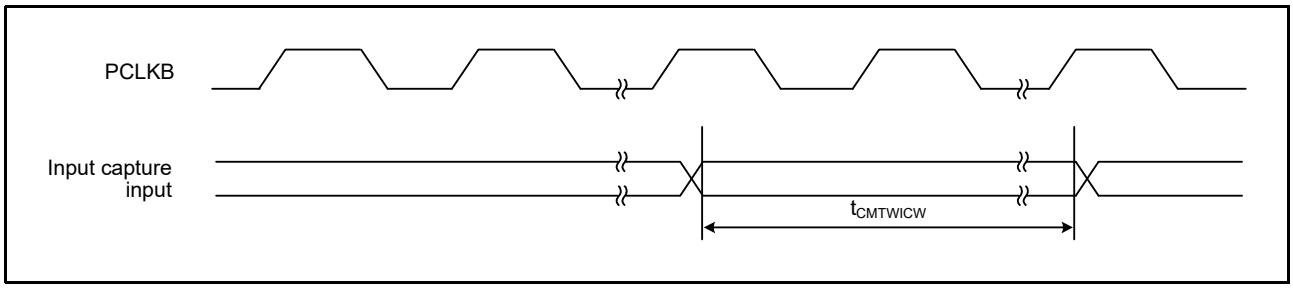


Figure 67.48 CMTW Input Capture Input Timing

67.5.7.4 POEG

Table 67.79 POEG Timing

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
POEG	GTETRGn input pulse width (n = A to D)		t_{POEGW}	1.5	—	t_{PAcyc}	
	POEGn input pulse width (n = A to D)		t_{POEGW2}	1.5	—	t_{PAcyc}	
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	3 PCLKA + 0.34	μs	When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Input level detection of the POEGn pin (via flag)	$t_{POEGDI2}$	—	3 PCLKA + 0.34		When the digital noise filter is not in use (POEGG2n.NFEN = 0 (n = A to D))
		Change of the POEGn signal level	$t_{POEGDI3}$	—	4 PCLKA + 0.34		Detection of a rising or falling edge (POEGG2n.PIDM[1:0] = 01b or 10b)
		Detection of the output stopping signal from GPTW (deadtime error, simultaneous high output, or simultaneous low output)	t_{POEGDE}	—	0.5		
		Register setting	t_{POEGDS}	—	1 PCLKA + 0.3		Time for access to the register is not included.
		Detection of the main clock oscillation stop	$t_{POEGDOS}$	—	2		

Note 1. t_{PAcyc}: PCLKA cycle

67.5.7.5 GPTW

Table 67.80 GPTW Timing

Item			Symbol	Min.	Max.	Unit*1, *2	Test Conditions
GPTW	Input capture input pulse width (cycle)	Single-edge setting	t_{GTICW}	1.5	—	t_{GTcyc}	
		Both-edge setting		2.5	—		
	External trigger input pulse width (cycle)	Single-edge setting	t_{GTEW}	1.5	—	t_{PAcyc}	
		Both-edge setting		2.5	—		
	Timer clock pulse width		t_{GTCKWH}, t_{GTCKWL}	1.5	—	t_{PAcyc}	

Note 1. t_{GTcyc}: GTCLK cycle
Note 2. t_{PAcyc}: PCLKA cycle

67.5.7.6 RSCI

Table 67.81 RSCI Timing (Asynchronous Mode)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item		VCC/VCC2	Symbol	Min.	Max.	Unit*1	Test Conditions
RSCI (Asynchronous)	Input clock cycle		t_{Scyc}	4	—	t_{PAcyc}	Figure 67.49
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	0.1	ns	
	Input clock fall time		t_{SCKf}	—	0.1	ns	
	Output clock cycle		t_{Scyc}	6	—	t_{PAcyc}	
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time	2.70 V or higher	t_{SCKr}	—	3.3	ns	
		1.62 V or higher		—	6.6		
	Output clock fall time	2.70 V or higher	t_{SCKf}	—	3.3	ns	
		1.62 V or higher		—	6.6		

Note 1. t_{PAcyc}: PCLKA cycle

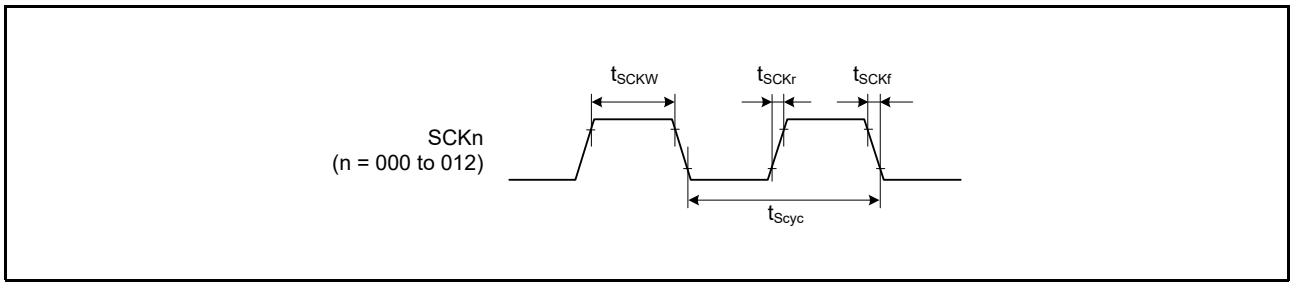


Figure 67.49 SCK Clock Input Timing

Table 67.82 RSCI Timing (Clock Synchronous Mode)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item			VCC/VCC2	Symbol	Min.	Max.	Unit*1	Test Conditions
RSCI (clock synchronous)	Output clock cycle	Master	2.70 V or higher	t_{Scyc}	2 (TCLK ≤ 120 MHz)	—	t_{Tcyc}	Figure 67.50
					4 (TCLK > 120 MHz)	—		
			1.62 V or higher		2 (TCLK ≤ 60 MHz)	—		
					4 (TCLK ≤ 120 MHz)	—		
					8 (TCLK > 120 MHz)	—		
	Input clock cycle	Slave	2.70 V or higher		2	—	t_{Tcyc}	
			1.62 V or higher		2 (TCLK ≤ 100 MHz)	—		
					4 (TCLK > 100 MHz)	—		
	Clock high pulse width	Master		t_{SCKWH}	0.4	0.6	t_{Scyc}	
		Slave			0.4	0.6		
	Clock low pulse width	Master		t_{SCKWL}	0.4	0.6	t_{Scyc}	
		Slave			0.4	0.6		
	Clock rise/fall time	Output	2.70 V or higher	t_{SCKr}, t_{SCKf}	—	3.3	ns	
			1.62 V or higher		—	6.6		
		Input	1.62 V or higher		—	1.0	μs	
	Data input setup time	Master (high-speed pins)*2		t_{SU}	2.6	—	ns	
		Master			2.8	—		
		Slave	2.70 V or higher		3.3	—		
			1.62 V or higher		5.3	—		
	Data input hold time	Master	2.70 V or higher	t_{H}	7.5	—	ns	
			1.62 V or higher		9.5	—		
		Slave	2.70 V or higher		3.0	—		
			1.62 V or higher		5.0	—		
	Data output delay time	Master (high-speed pins)*2		t_{OD}	—	5.0	ns	
		Master			—	5.0		
		Slave (high speed pins)*2	2.70 V or higher		—	15		
			1.62 V or higher		—	23		
		Slave	2.70 V or higher		—	21		
			1.62 V or higher		—	29		
	Data rise/fall time	Output	2.70 V or higher	t_{Dr}, t_{Df}	—	3.3	ns	
			1.62 V or higher		—	6.6		
		Input			—	1.0	μs	

Note 1. t_{Tcyc}: TCLK (PCLKA or SCICLK) cycle
Note 2. The listed value applies when the pins with -A appended to their names are in use.

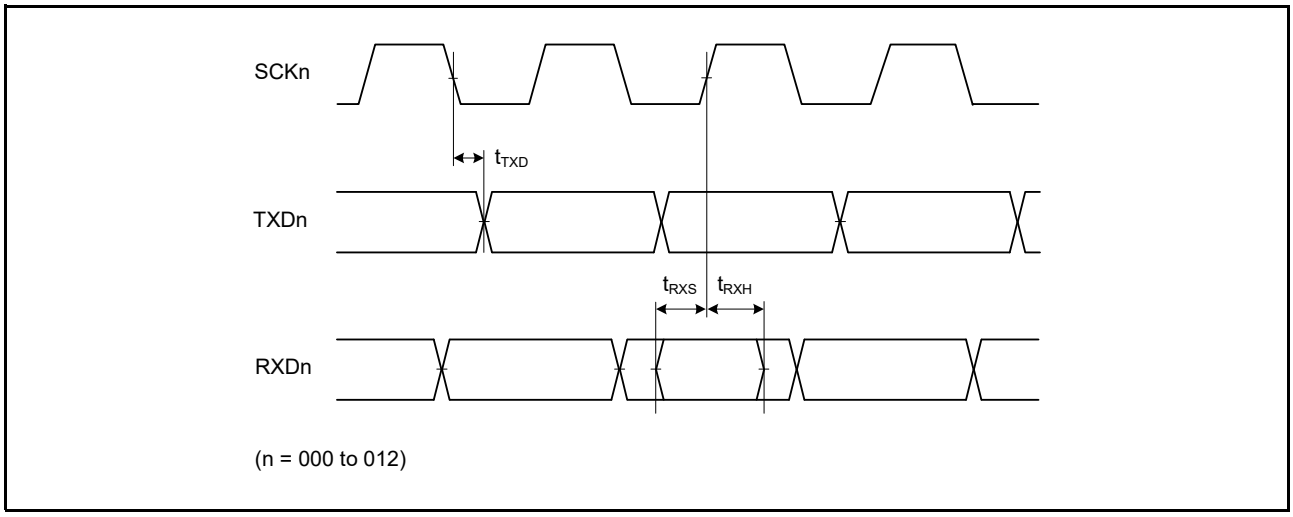


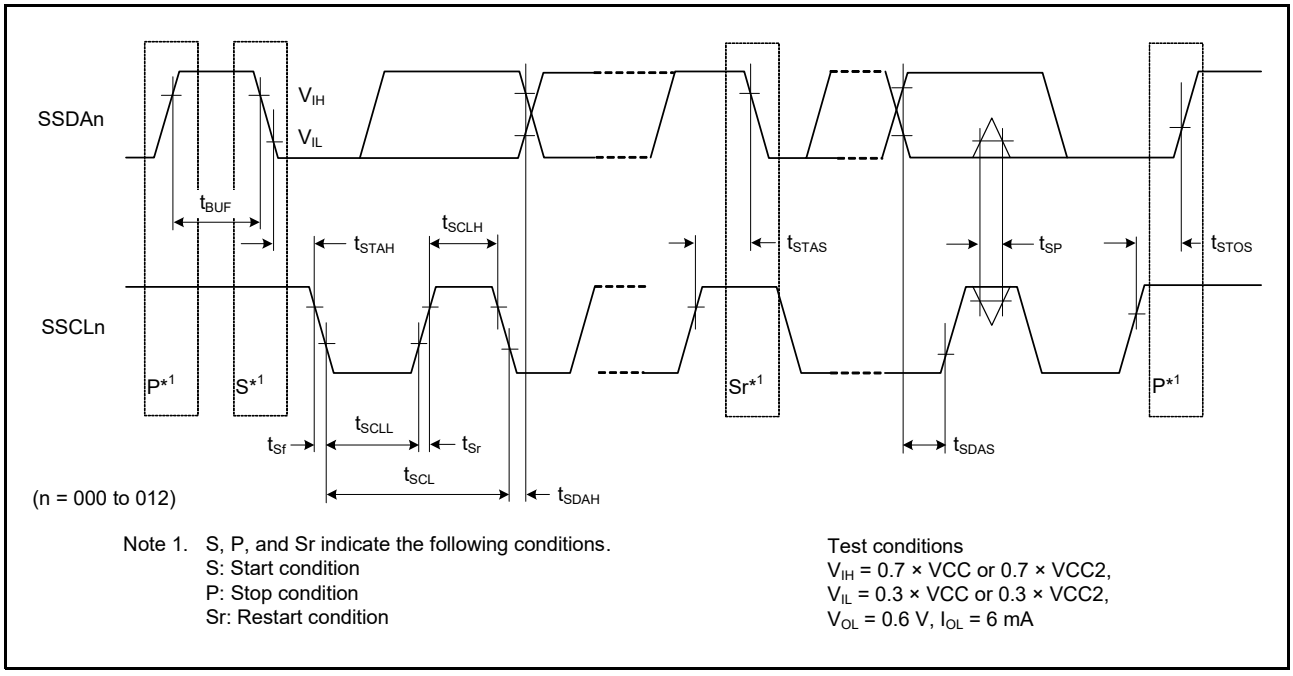
Figure 67.50 RSCI Input/Output Timing: Clock Synchronous Mode

Table 67.83 Simple IIC Timing

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item		Symbol	Min.	Max.*1	Unit	Test Conditions
Simple IIC (Standard-mode)	SSCL and SSDA input rise time	t_{Sr}	—	1000	ns	Figure 67.51
	SSCL and SSDA input fall time	t_{Sf}	—	300		
	SSCL and SSDA input spike pulse removal time	t_{SP}	0	4 × t_{PAcyc}		
	Data input setup time	t_{SDAS}	250	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL and SSDA capacitive load	C_b*2	—	400	pF	
Simple IIC (Fast-mode)	SSCL and SSDA input rise time	t_{Sr}	—	300	ns	
	SSCL and SSDA input fall time	t_{Sf}	—	300		
	SSCL and SSDA input spike pulse removal time	t_{SP}	0	4 × t_{PAcyc}		
	Data input setup time	t_{SDAS}	100	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL and SSDA capacitive load	C_b*2	—	400	pF	

Note 1. t_{PAcyc}: PCLKA cycle
Note 2. Cb is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC or 0.7 × VCC2,
V_{IL} = 0.3 × VCC or 0.3 × VCC2,
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 67.51 Simple IIC Bus Interface Input/Output Timing

Table 67.84 Simple SPI Timing (1/2)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item			VCC/VCC2	Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	Output clock cycle	Master	2.70 V or higher	t_{Scyc}	2 (TCLK ≤ 120 MHz)	65536	t_{Tcyc}	Figure 67.52
					4 (TCLK > 120 MHz)	65536		
			1.62 V or higher		2 (TCLK ≤ 60 MHz)	65536		
					4 (TCLK ≤ 120 MHz)	65536		
					8 (TCLK > 120 MHz)	65536		
	Input clock cycle	Slave	2.70 V or higher		2	—	t_{Tcyc}	
			1.62 V or higher		2 (TCLK ≤ 100 MHz)	—		
					4 (TCLK > 100 MHz)	—		
	Clock high pulse width			t_{SCKWH}	0.4	0.6	t_{SPcyc}	
	Clock low pulse width			t_{SCKWL}	0.4	0.6		
	Clock rise/fall time	Output	2.70 V or higher	t_{SCKr}, t_{SCKf}	—	3.3	ns	
			1.62 V or higher		—	6.6		
		Input			—	1.0	µs	
	Data input setup time	Master (high-speed pins)*2		t_{SU}	–1.5	—	ns	Figure 67.53, Figure 67.54
		Master			2.0	—		
		Slave	2.70 V or higher		2.5	—		
			1.62 V or higher		4.5	—		
	Data input hold time	Master	2.70 V or higher	t_{H}	7.5	—	ns	
			1.62 V or higher		9.5	—		
		Slave	2.70 V or higher		2.5	—		
			1.62 V or higher		4.5	—		
	Data output delay time	Master (high-speed pins)*2	2.70 V or higher	t_{OD}	—	3.0	ns	
			1.62 V or higher		—	4.5		
		Master	2.70 V or higher		—	3.5		
			1.62 V or higher		—	5.5		
		Slave (high-speed pins)*2	2.70 V or higher		—	15		
			1.62 V or higher		—	23		
		Slave	2.70 V or higher		—	21		
			1.62 V or higher		—	29		
	Data output hold time	Master (high-speed pins)*2	2.70 V or higher	t_{OH}	–3.0	—	ns	
			1.62 V or higher		–4.5	—		
		Master	2.70 V or higher		–3.5	—		
			1.62 V or higher		–5.5	—		
		Slave			0	—		
	Data rise/fall time	Output	2.70 V or higher	t_{Dr}, t_{Df}	—	3.3	ns	
			1.62 V or higher		—	6.6		
		Input			—	1.0		
	SS input setup time			t_{LEAD}	1.0	—	t_{SPcyc}	Figure 67.55, Figure 67.56
	SS input hold time			t_{LAG}	1.0	—		
	SS data rise/fall time			t_{SSLr}, t_{SSLf}	—	1.0	µs	

Table 67.84 Simple SPI Timing (2/2)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item		VCC/VCC2	Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	Slave access time	2.70 V or higher	t_{SA}	—	$3 \times t_{Tcyc} + 25$*1	ns	Figure 67.55, Figure 67.56
		1.62 V or higher		—	$3 \times t_{Tcyc} + 32$*1		
	Slave output release time	2.70 V or higher	t_{REL}	—	$3 \times t_{Tcyc} + 25$*1	ns	
		1.62 V or higher		—	$3 \times t_{Tcyc} + 32$*1		

Note 1. t_{Tcyc}: TCLK (PCLKA or SCICLK) cycle
Note 2. The listed value applies when the pins with -A appended to their names are in use.

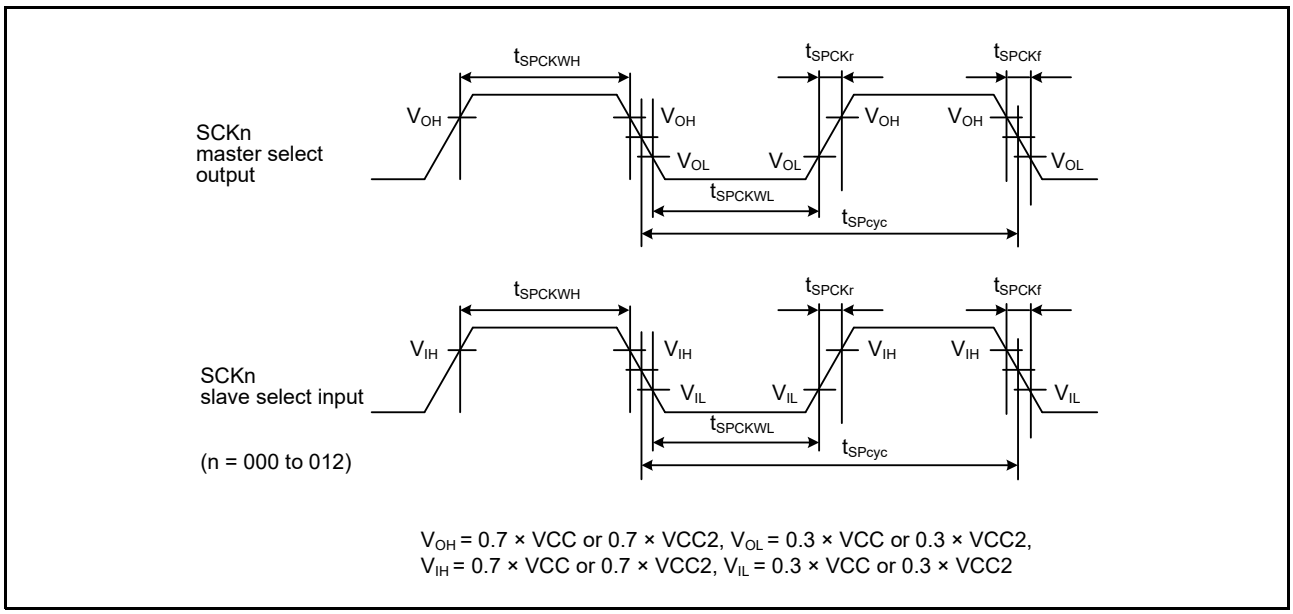


Figure 67.52 Simple SPI Clock Timing

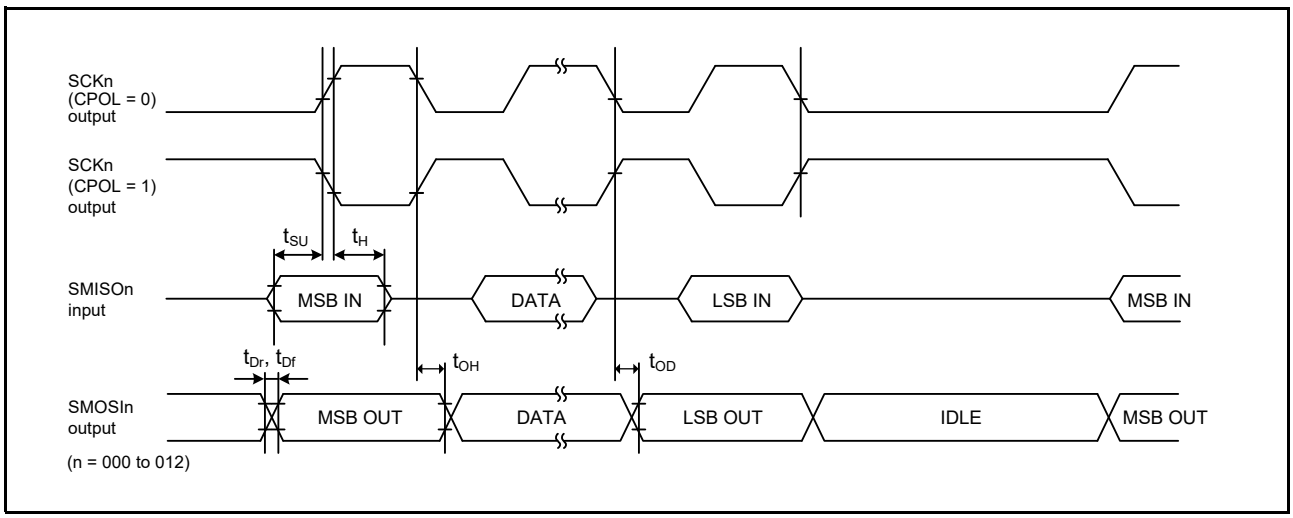


Figure 67.53 Simple SPI Timing (Master, CPHA = 0)

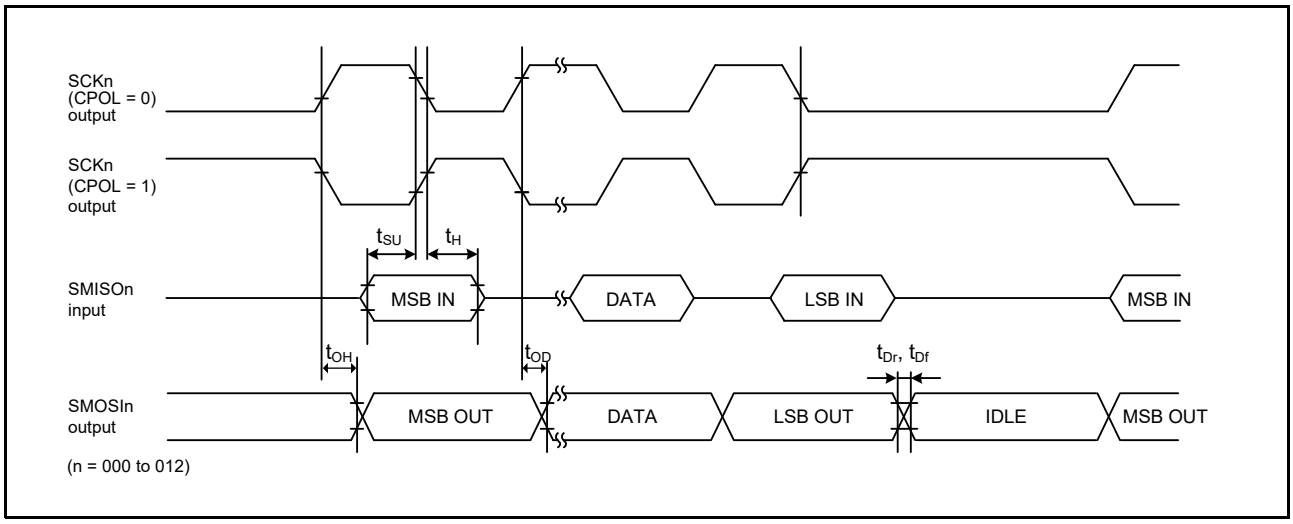


Figure 67.54 Simple SPI Timing (Master, CPHA = 1)

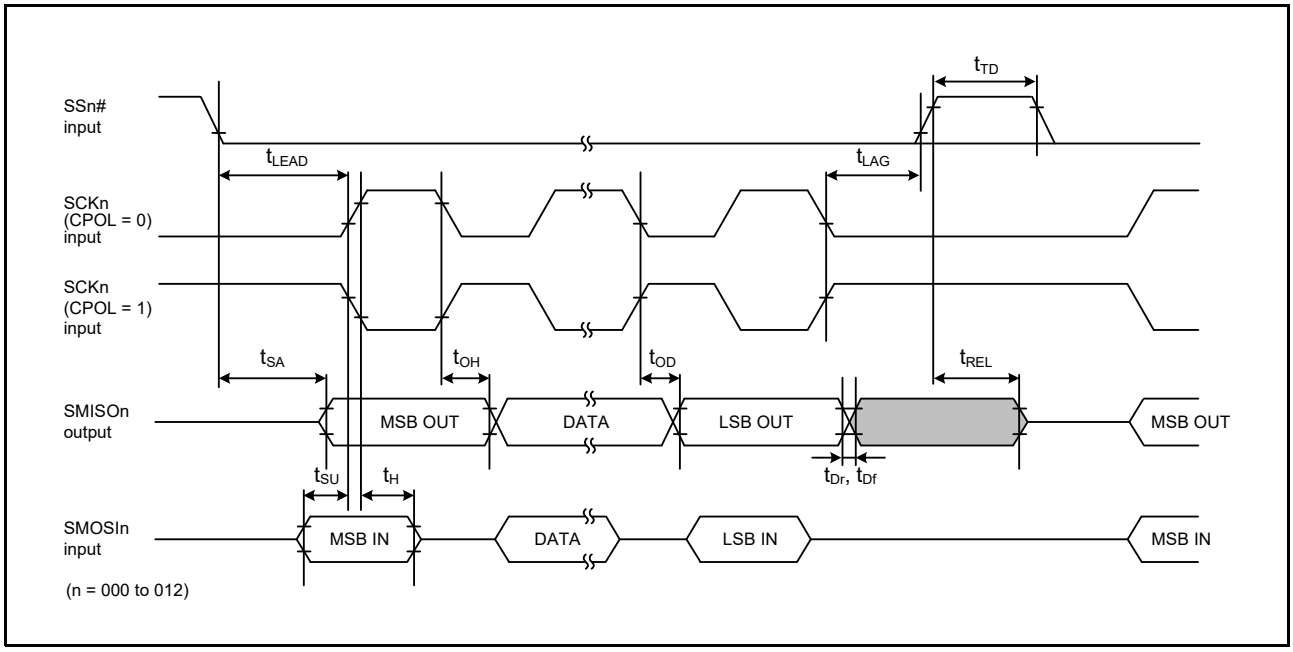


Figure 67.55 Simple SPI Timing (Slave, CPHA = 0)

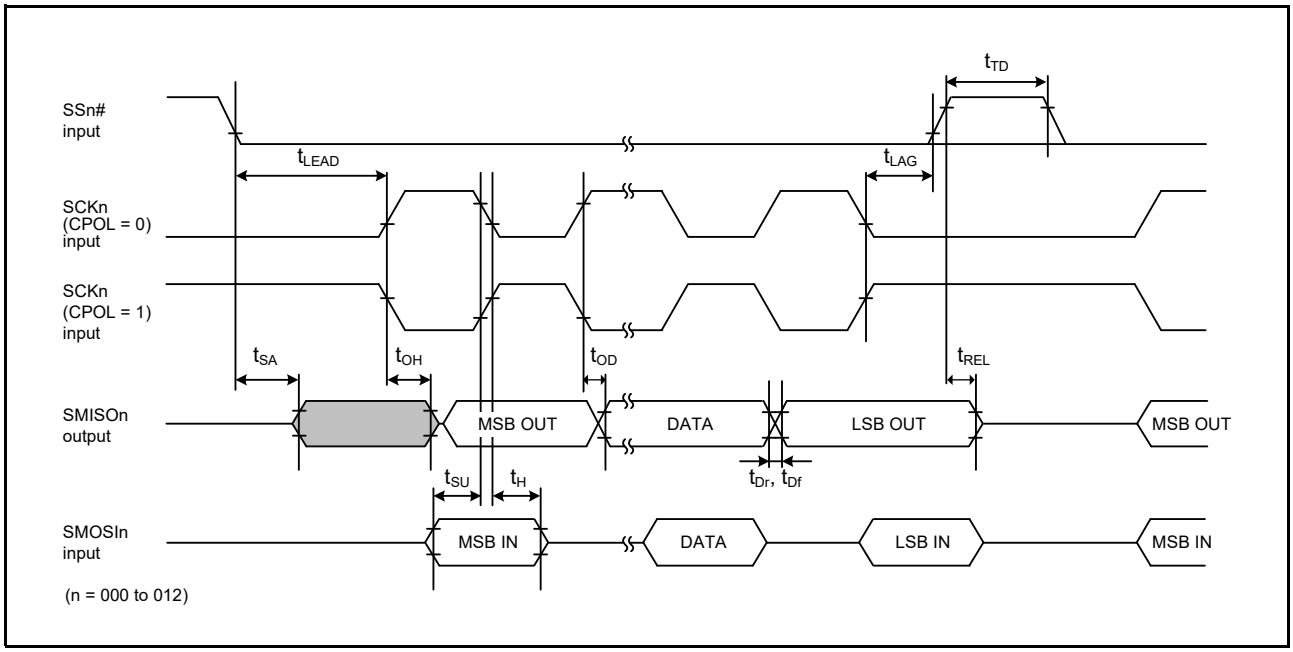


Figure 67.56 Simple SPI Timing (Slave, CPHA = 1)

67.5.7.7 RSPIA

Table 67.85 RSPIA Timing (1/2)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item			VCC/VCC2	Symbol	Min.*1	Max.*1	Unit*1	Test Conditions
RSPIA	Clock cycle	Master	3.00 V or higher	t_{SPcyc}	2	4096	t_{Tcyc}	Figure 67.57
			2.70 V or higher		2	4096		
			1.62 V or higher		2 (TCLK ≤ 60 MHz)	4096		
					4 (TCLK > 60 MHz)	4096		
		Slave	3.00 V or higher		2	—		
			2.70 V or higher		2	—		
			1.62 V or higher		2 (TCLK ≤ 60 MHz)	—		
					4 (TCLK > 60 MHz)	—		
	Clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			0.4	—	t_{SPcyc}	
	Clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			0.4	—	t_{SPcyc}	
	Clock rise/fall time	Output	3.00 V or higher	t_{SPCKr}, t_{SPCKf}	—	1.66	ns	
			2.70 V or higher		—	3.30		
			1.62 V or higher		—	6.60		
		Input			—	0.1	μs	
	Data input setup time	Master	3.00 V or higher	t_{SU}	–1.5	—	ns	Figure 67.58, Figure 67.59
			2.70 V or higher		0.0	—		
			1.62 V or higher		0.0	—		
		Slave			2.5	—		
	Data input hold time	Master	3.00 V or higher	t_H	7.5	—	ns	
			2.70 V or higher		7.5	—		
			1.62 V or higher		9.5	—		
		Slave	3.00 V or higher		2.5	—		
			2.70 V or higher		2.5	—		
			1.62 V or higher		5.5	—		
	SSL setup time	Master		t_{LEAD}	$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$	ns	
		Slave			5.0	—	t_{Tcyc}	
	SSL hold time	Master		t_{LAG}	$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$	ns	
		Slave			5.0	—	t_{Tcyc}	
	TI SSP SS input setup time	Slave		t_{TISS}	2.5	—	ns	Figure 67.63, Figure 67.64
	TI SSP SS input hold time	Slave	3.00 V or higher	t_{TISH}	2.5	—	ns	
			2.70 V or higher		2.5	—		
			1.62 V or higher		5.5	—		
	TI SSP next-access time	Slave		t_{TIND}	$2 \times t_{Tcyc} + S_{LNDL} \times t_{Tcyc}$	—	ns	
	TI SSP master SS output delay time	Master	3.00 V or higher	t_{TISSOD}	—	4.0	ns	Figure 67.60
			2.70 V or higher		—	8.0		
			1.62 V or higher		—	8.0		

Table 67.85 RSPIA Timing (2/2)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.

Item			VCC/VCC2	Symbol	Min.*1	Max.*1	Unit*1	Test Conditions
RSPIA	Data output delay time	Master	3.00 V or higher	t_{OD1}	—	2.5	ns	Figure 67.58, Figure 67.59
			2.70 V or higher		—	3.0		
			1.62 V or higher		—	6.0		
		Master	3.00 V or higher	t_{OD2}	—	2.5		
			2.70 V or higher		—	2.5		
			1.62 V or higher		—	5.0		
		Slave	3.00 V or higher	t_{OD}	—	11.5*2 13.0*3		
			2.70 V or higher		—	16		
			1.62 V or higher		—	24		
	Data output hold time	Master	3.00 V or higher	t_{OH}	–2.5	—	ns	
			2.70 V or higher		–2.5	—		
			1.62 V or higher		–4.5	—		
		Slave			0.0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$	ns	
		Slave			t_{Tcyc}	—		
	MOSI and MISO rise/fall time	Output	3.00 V or higher	t_{Dr}, t_{Df}	—	1.66	ns	
			2.70 V or higher		—	3.30		
			1.62 V or higher		—	6.60		
		Input			—	1.00	µs	
	SSL rise/fall time	Output	3.00 V or higher	t_{SSLr}, t_{SSLf}	—	1.66	ns	
			2.70 V or higher		—	3.30		
			1.62 V or higher		—	6.60		
		Input			—	1.00	µs	
	Slave access time	Slave	3.00 V or higher	t_{SA}	—	20.0	ns	Figure 67.61, Figure 67.62
			2.70 V or higher		—	20.0		
			1.62 V or higher		—	25.0		
	Slave output release time	Slave	3.00 V or higher	t_{REL}	—	20.0	ns	
			2.70 V or higher		—	20.0		
			1.62 V or higher		—	25.0		

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for a given RSPIA interface are from the same group.

Note 1. t_{Tcyc}: TCLK (PCLKA or SPICLK) cycle

Note 2. When the pin in use is a group-A pin

Note 3. When the pin in use is not a group-A pin

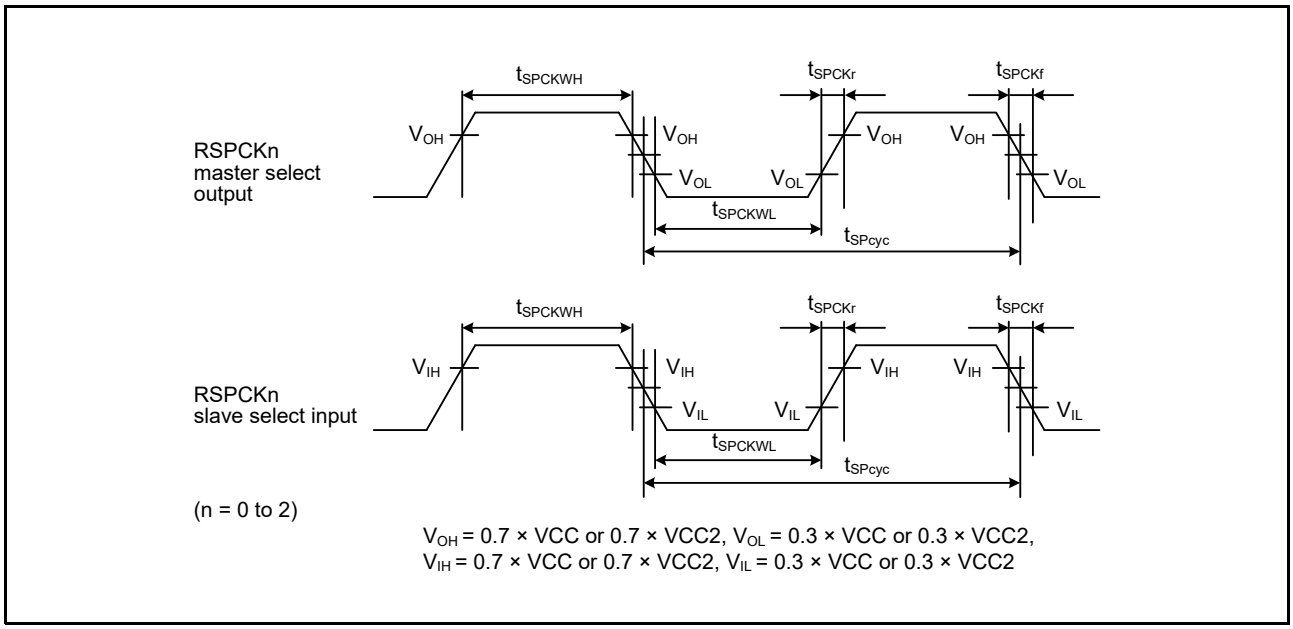


Figure 67.57 RSPIA Clock Timing

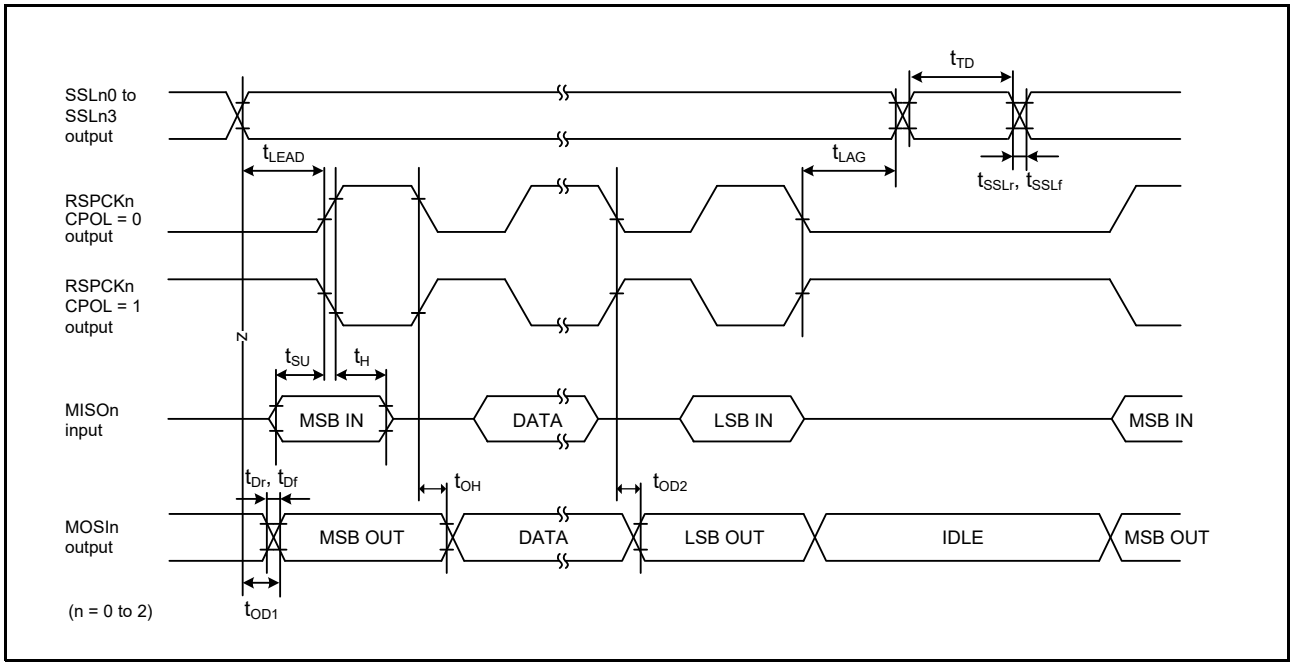


Figure 67.58 RSPIA Timing (Master, Motorola SPI, CPHA = 0)

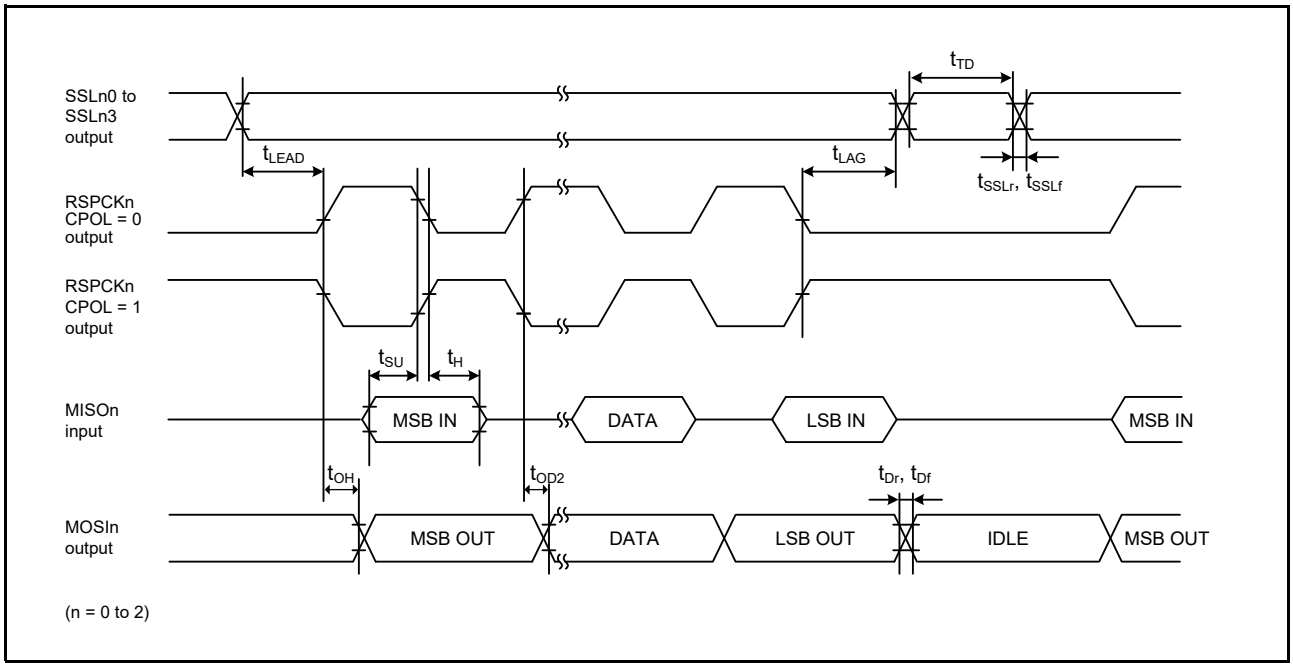


Figure 67.59 RSPIA Timing (Master, Motorola SPI, CPHA = 1)

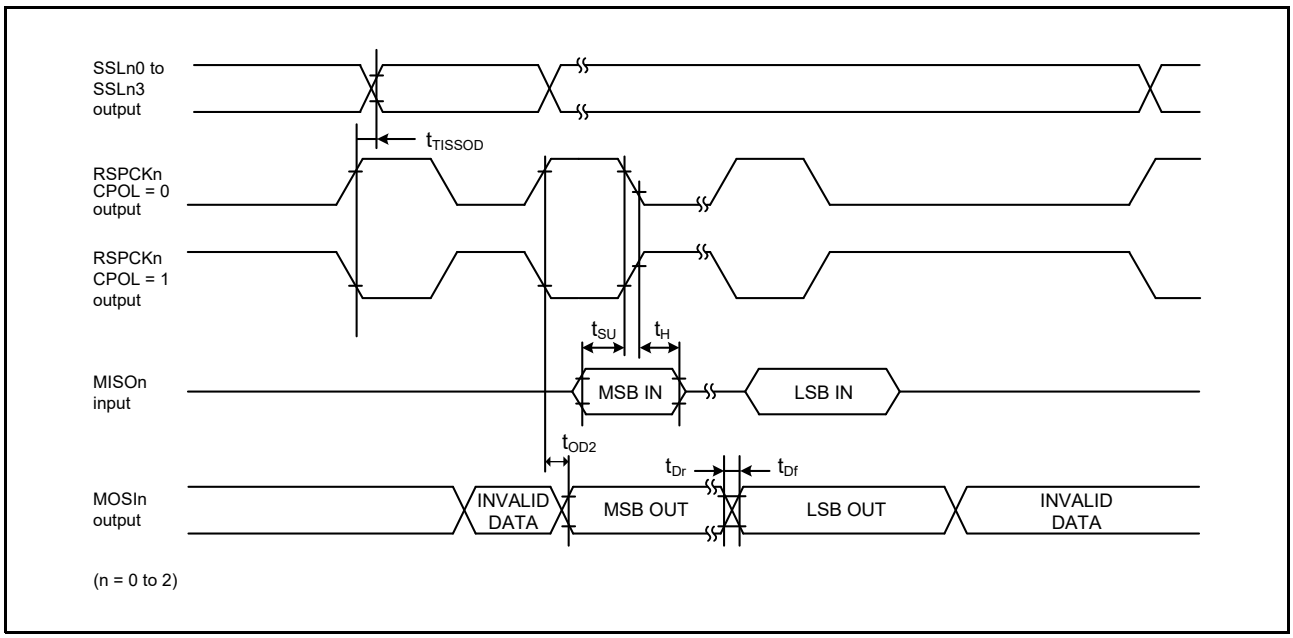


Figure 67.60 RSPIA Timing (Master, TI SSP)

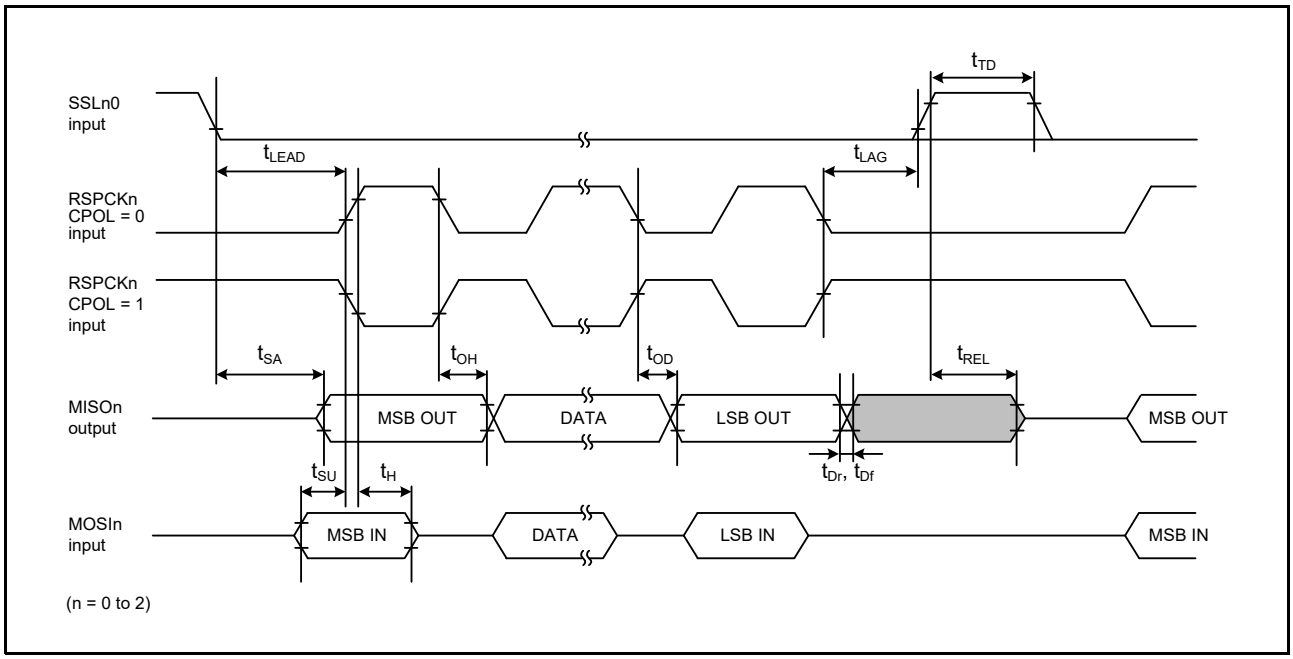


Figure 67.61 RSPIA Timing (Slave, Motorola SPI, CPHA = 0)

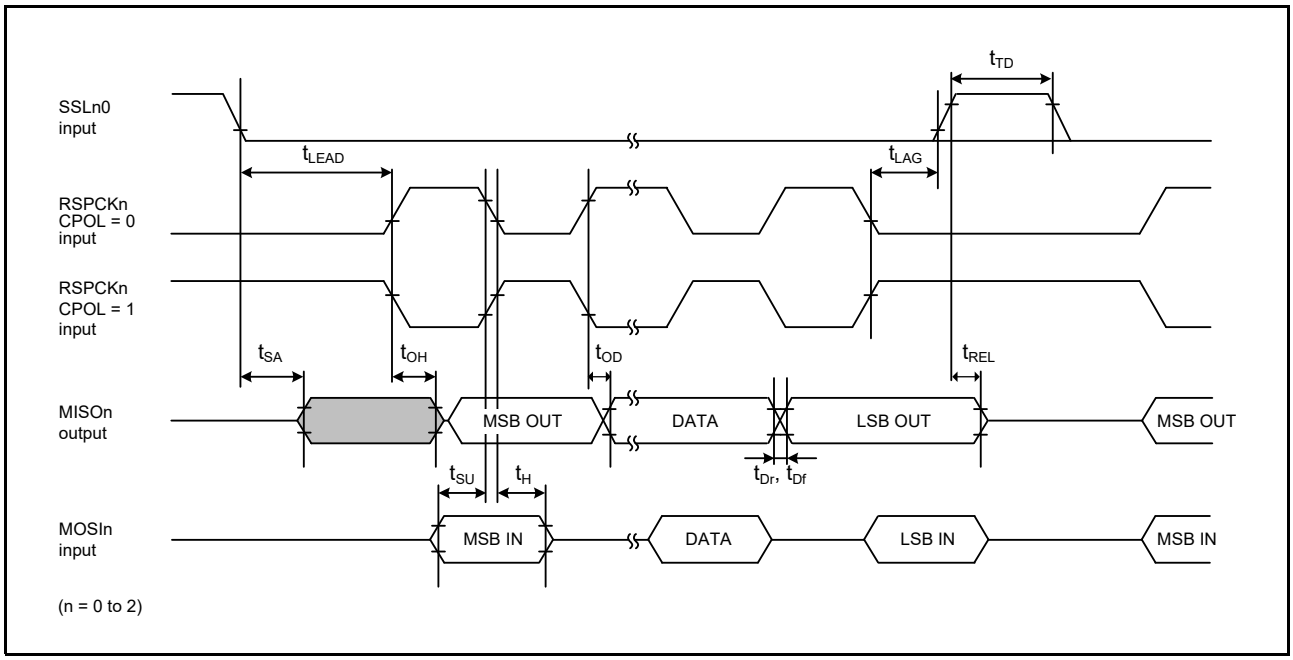


Figure 67.62 RSPIA Timing (Slave, Motorola SPI, CPHA = 1)

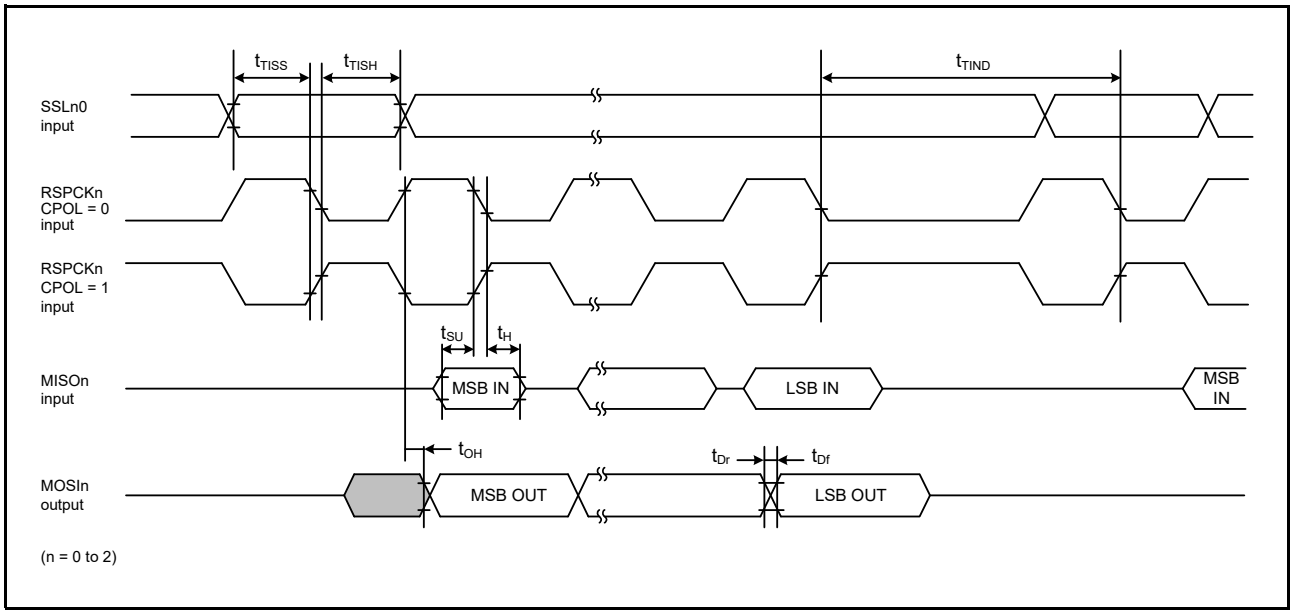


Figure 67.63 RSPIA Timing (Slave, TI SSP, with Inter-Frame Delay)

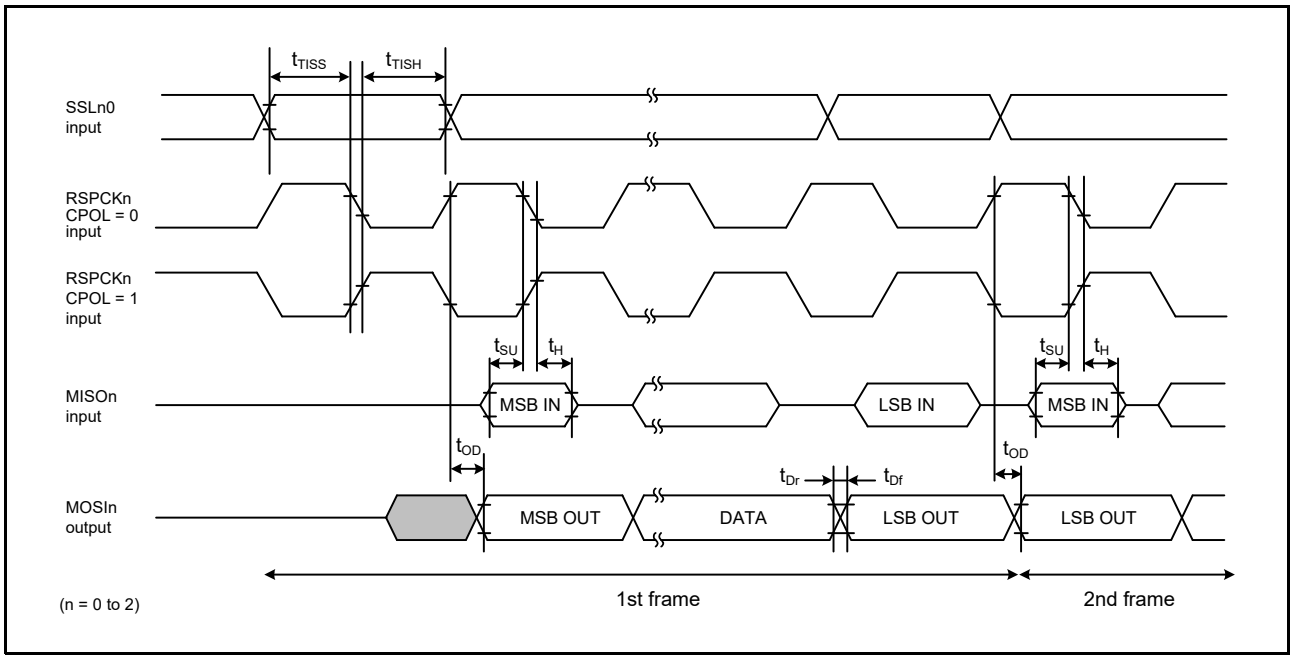


Figure 67.64 RSPIA Timing (Slave, TI SSP, with No Inter-Frame Delay)

67.5.7.8 XSPI

Table 67.86 XSPI Master Timing

Conditions: Capacitive load C = 15 pF,
When VCC/VCC2 ≥ 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the XSPIn_CK, XSPIn_DS, and XSPIn_IO0 to XSPIn_IO7 pins are set for DRV6.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the XSPIn_CS0# and XSPIn_CS1# pins are set for DRV3.
When VCC/VCC2 < 2.0 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the XSPIn_CK, XSPIn_DS, and XSPIn_IO0 to XSPIn_IO7 pins are set for DRV8.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the XSPIn_CS0# and XSPIn_CS1# pins are set for DRV4.

Item			VCC/VCC2	Symbol	Min.	Max.	Unit	Test Conditions
XSPI master	Cycle time	SDR without the XSPIn_DS pin	2.70 V or higher	t_{PERIOD}	16.67	—	ns	Figure 67.65
			Less than 2.00 V		20.00	—		
		DDR with the XSPIn_DS pin*1			7.5	—		
	Clock output slew rate			t_{SRck}	0.75	—	V/ns	
	Clock duty cycle-distortion			t_{CKDCD}	0	0.375	ns	
	Clock minimum pulse width			t_{CKMPW}	3.375	—	ns	
	DS Duty cycle distortion			t_{DSDCD}	0	0.3	ns	
	DS minimum pulse width			t_{DSMPW}	3.075	—	ns	
	Data input/output slew rate			t_{SR}	1.37	—	ns	
	Data input setup time	SDR without the XSPIn_DS pin	2.70 V or higher	t_{SU}	8.17	—	ns	Figure 67.66
			Less than 2.00 V		13.0	—		
	Data input hold time	SDR without the XSPIn_DS pin		t_{H}	0.5	—	ns	
	Data output valid time*2	SDR without the XSPIn_DS pin		t_{OV}	—	5.4	ns	
	Data output hold time	SDR without the XSPIn_DS pin		t_{OH}	–5.4	—	ns	
	Data output buffer off time	SDR without the XSPIn_DS pin		t_{BOFF}	–5.4	—	ns	
	Data input setup time	DDR with the XSPIn_DS pin*1		t_{SU}	–0.7	—	ns	Figure 67.67, Figure 67.68
	Data input hold time	DDR with the XSPIn_DS pin*1		t_{H}	2.375	—	ns	
	Data output valid time*2	DDR with the XSPIn_DS pin*1		t_{OV}	—	$t_{PERIOD}/4 + 0.6$	ns	
	Data output hold time	DDR with the XSPIn_DS pin*1		t_{OH}	0.9	—	ns	
	Data output buffer off time	DDR with the XSPIn_DS pin*1		t_{BOFF}	0.9	—	ns	
	Time from clock low to XSPIn_CSi# low			t_{CKLCSL}	6	—	ns	Figure 67.66, Figure 67.67, Figure 67.68
	Time from XSPIn_CSi# low to clock high*3			t_{CSLCKH}	6	—	ns	
	Time from clock low to XSPIn_CSi# high			t_{CKLCSH}	6	—	ns	
	Time from XSPIn_CSi# high to clock high			t_{CSHCKH}	6	—	ns	
	Time from XSPIn_DS low to XSPIn_CSi# high			t_{DSLCSH}	$0.8 \times t_{PERIOD}$	—	ns	Figure 67.69
	Time from XSPIn_CSi# high to XSPIn_DS tristate			t_{CSHDST}	—	t_{PERIOD}	ns	
	Time when XSPIn_CSi# low to XSPIn_DS low input *3, *4			t_{CSLDSL}	0	12.5	ns	
	Time from XSPIn_DS tristate to XSPIn_CSi# low			t_{DSTCSL}	0	—	ns	

n = 0, 1; i = 0, 1

Note 1. For products with part numbers of the form R5K574Nxx4Fx, only use this value for reference.
Note 2. COMCFG.OEASTEX = 1
Note 3. LIOCFGCSx.CSASTEX = 1
Note 4. When using a JESD251 Profile 1.0 memory with an external pull-down connected to the XSPIn_DS pin, you do not need to follow this restriction.

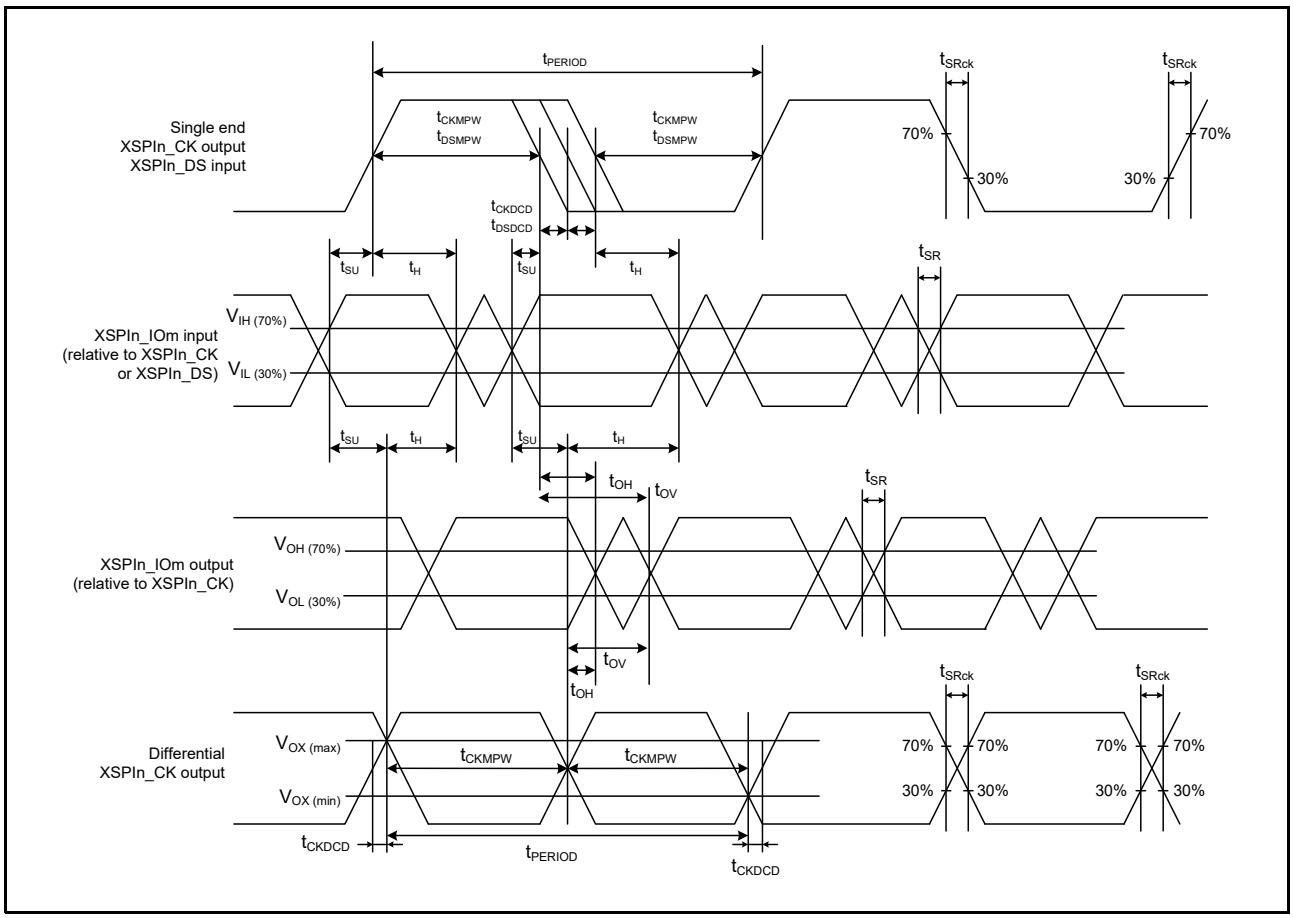


Figure 67.65 XSPI Clock/DS Timing (n = 0, 1; m = 0 to 7)

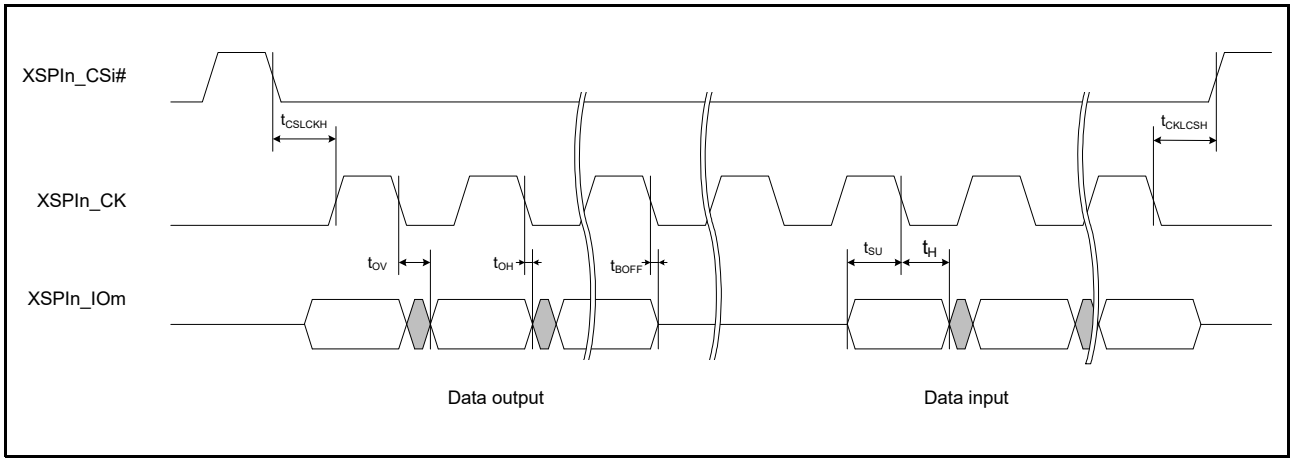


Figure 67.66 SDR Transmit/Receive Timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) (n = 0, 1; m = 0 to 7; i = 0, 1)

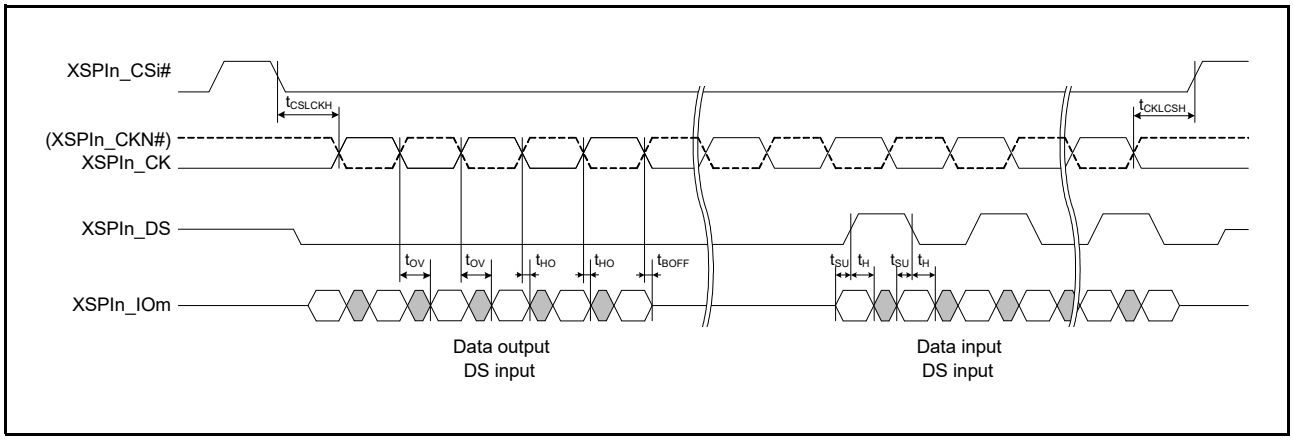


Figure 67.67 DDR Transmit/Receive Timing (4S-4D-4D, 8D-8D-8D) (n = 0, 1; m = 0 to 7; i = 0, 1)

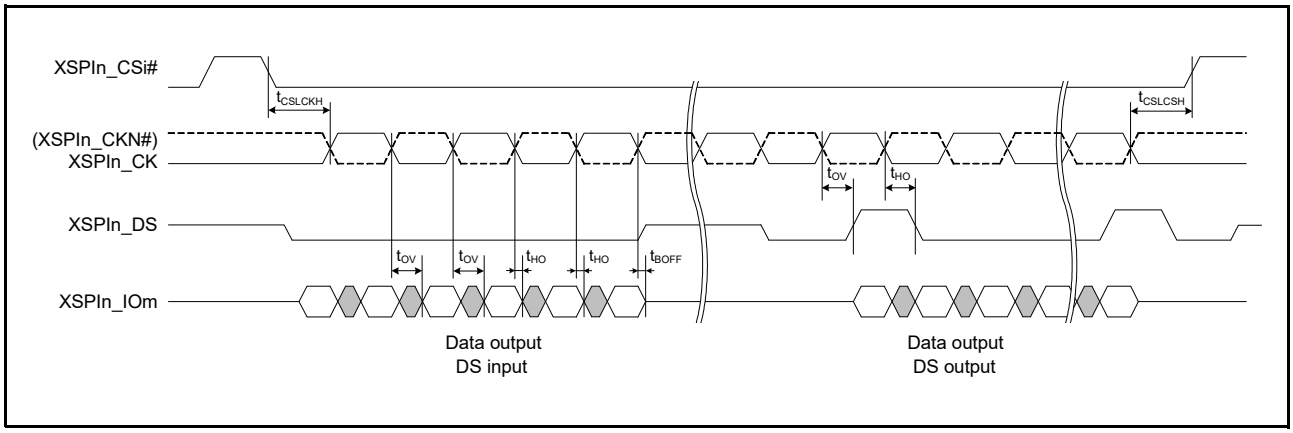


Figure 67.68 DDR Transmit/Receive Timing (HyperRAM write) (n = 0, 1; m = 0 to 7; i = 0, 1)

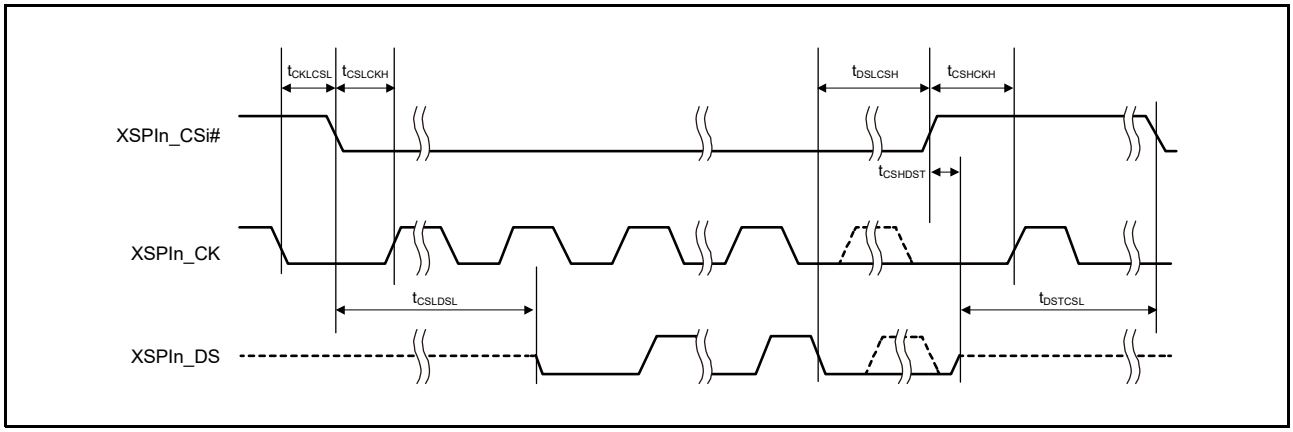


Figure 67.69 DS to CS Signal Timing (n = 0, 1; i = 0, 1)

Table 67.87 XSPI Slave Timing

Conditions: Capacitive load C = 15 pF,
When VCC/VCC2 ≥ 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV6.
When VCC/VCC2 < 2.0 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV8.

Item			VCC/VCC2	Symbol	Min.	Max.	Unit	Test Conditions
XSPI slave	Cycle time	SDR without the XSPIn_DS pin		t_{PERIOD}	40	—	ns	Figure 67.70
		DDR with the XSPIn_DS pin*1			10	—		
	Clock output slew rate			t_{SRck}	0.56	—	V/ns	
	Clock duty cycle-distortion			t_{CKDCD}	0	0.5	ns	
	Clock minimum pulse width			t_{CKMPW}	4.5	—	ns	
	DS Duty cycle distortion			t_{DSDCD}	0	0.4	ns	
	DS minimum pulse width			t_{DSMPW}	4.1	—	ns	
	Data input/output slew rate		2.70 V or higher	t_{SR}	1.03	—	ns	
			Less than 2.00 V		0.56	—		
	Data input setup time	SDR without the XSPIn_DS pin		t_{SU}	2.5	—	ns	Figure 67.71
	Data input hold time	SDR without the XSPIn_DS pin		t_{H}	2.5	—	ns	
	Data output valid time	SDR without the XSPIn_DS pin		t_{OV}	—	14	ns	
	Data output hold time	SDR without the XSPIn_DS pin		t_{OH}	0	—	ns	
	Data output buffer off time	SDR without the XSPIn_DS pin		t_{BOFF}	—	2 × t_{PERIOD}	ns	
	Data input setup time	DDR with the XSPIn_DS pin*1		t_{SU}	1	—	ns	Figure 67.72, Figure 67.73
	Data input hold time	DDR with the XSPIn_DS pin*1		t_{H}	1	—	ns	
	Data output skew time*2	DDR with the XSPIn_DS pin*1		t_{RQ}	0.8	—	ns	
	Data output skew hold time	DDR with the XSPIn_DS pin*1		t_{RQH}	0.8	—	ns	
	Data output buffer off time	DDR with the XSPIn_DS pin*1		t_{BOFF}	—	2 × t_{PERIOD}	ns	
	Time from clock low to XSPIn_CSi# low			t_{CKLCSL}	6	—	ns	Figure 67.71, Figure 67.72, Figure 67.73
	Time from XSPIn_CSi# low to clock high			t_{CSLCKH}	6	—	ns	
	Time from clock low to XSPIn_CSi# high			t_{CKLCSH}	6	—	ns	
	Time from XSPIn_CSi# high to clock high			t_{CSHCKH}	6	—	ns	
	Time from XSPIn_DS low to XSPIn_CSi# high			t_{DSLCSH}	0.8 × t_{PERIOD}	—	ns	Figure 67.74
	Time from XSPIn_CSi# high to XSPIn_DS tristate			t_{CSHDST}	—	2 × t_{PERIOD}	ns	
	Time when XSPIn_CSi# low to XSPIn_DS low input*3		2.70 V or higher	t_{CSLDSL}	0	15	ns	
			Less than 2.00 V		0	19		
	Time from XSPIn_DS tristate to XSPIn_CSi# low			t_{DSTCSL}	0	—	ns	

n = 0, 1; i = 0, 1

Note 1. For products with part numbers of the form R5K574Nxx4Fx, only use this value for reference.
Note 2. L15CFG0.OEASTEX = 1
Note 3. When using a JESD251 Profile 1.0 memory with an external pull-down connected to the XSPIn_DS pin, you do not need to follow this restriction.

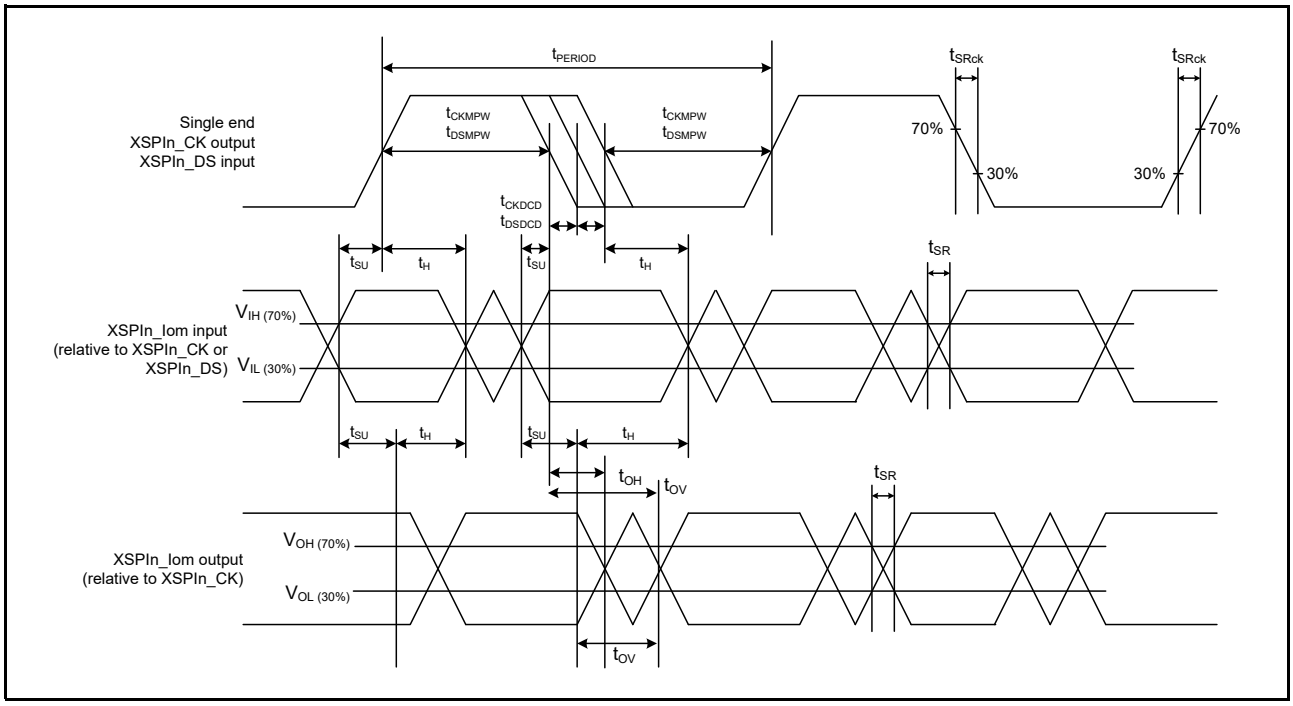


Figure 67.70 XSPI Clock/DS Timing (n = 0, 1; m = 0 to 7)

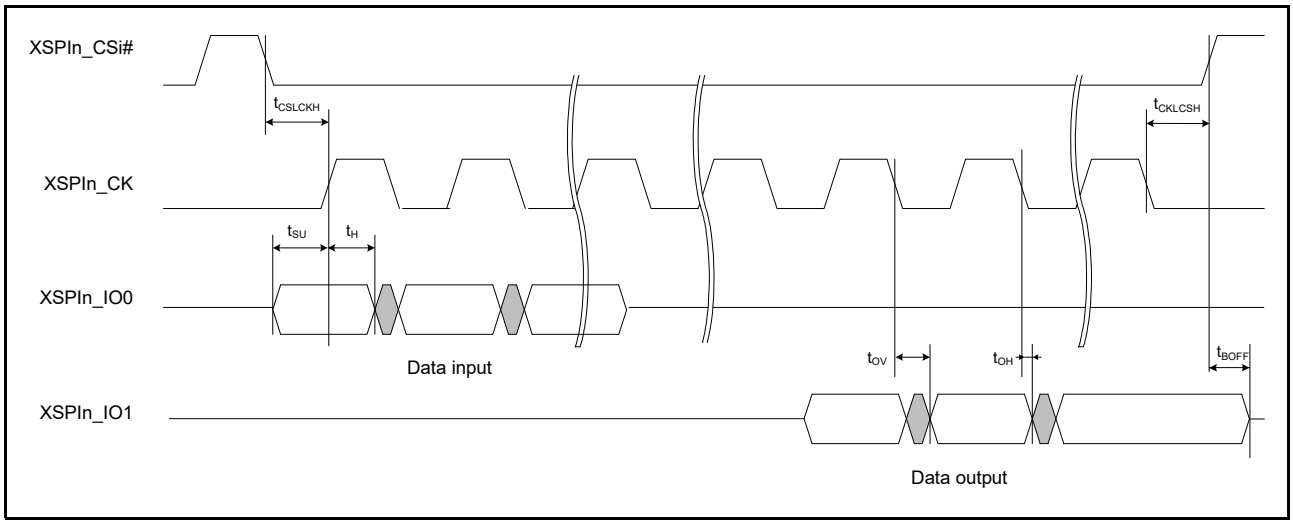


Figure 67.71 SDR Transmit/Receive Timing (1S-1S-1S) (n = 0, 1; i = 0, 1)

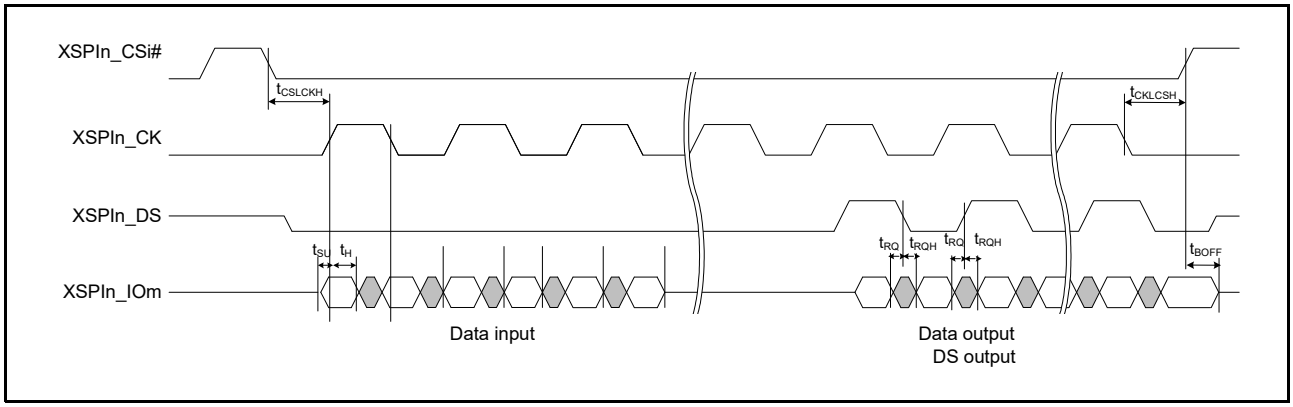


Figure 67.72 DDR Transmit/Receive Timing (4S-4D-4D, 8D-8D-8D) (n = 0, 1; m = 0 to 7; i = 0, 1)

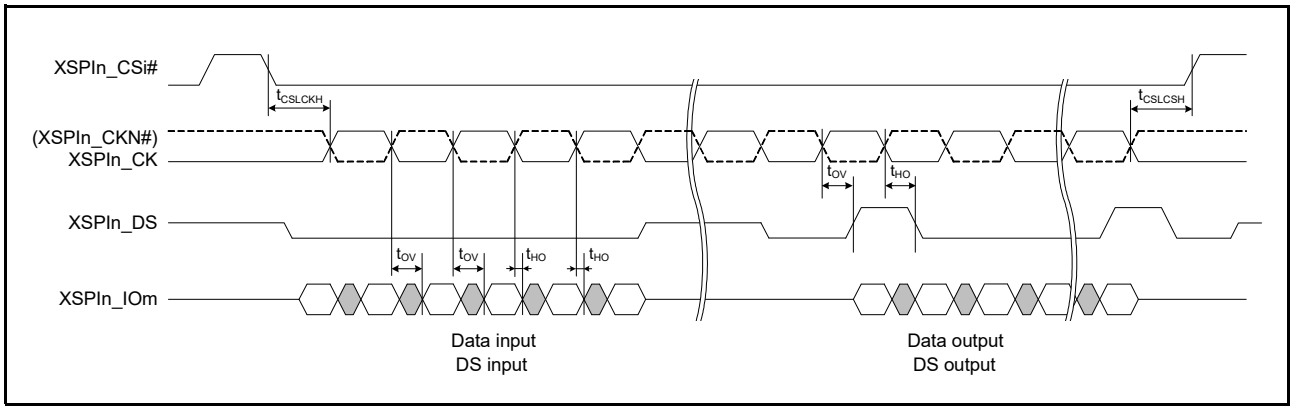


Figure 67.73 DDR Transmit/Receive Timing (HyperRAM write) (n = 0, 1; m = 0 to 7; i = 0, 1)

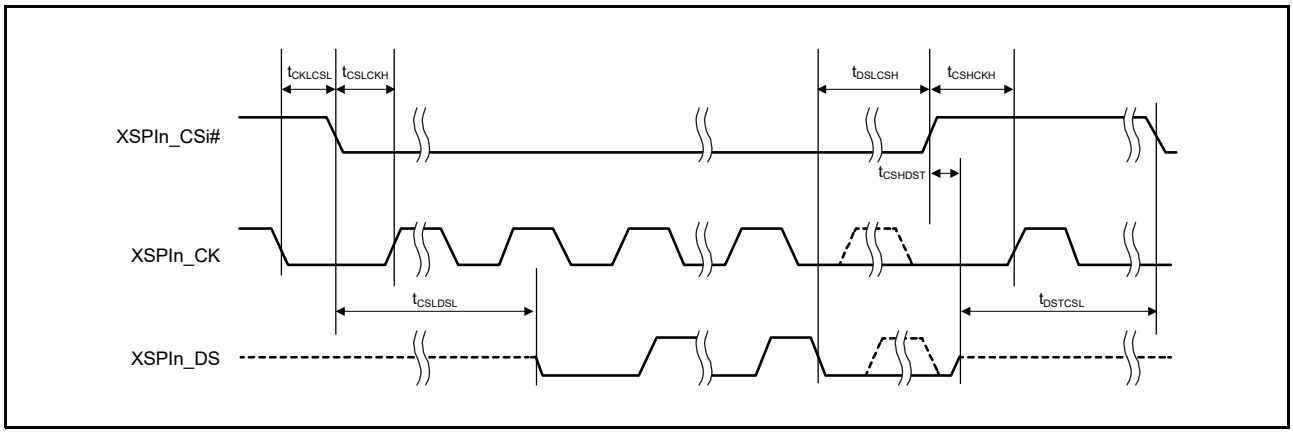


Figure 67.74 DS to CS Signal Timing (n = 0, 1; i = 0, 1)

67.5.7.9 RIICHS

Table 67.88 RIICHS Timing (1) (1/2)

Conditions: VCC = 1.62 to 1.98 V or VCC = 2.70 to 3.63 V,
Evaluation was with either of the following driving ability settings for the target pins, depending on the VCC voltage.
When VCC = 1.62 to 1.98 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.
When VCC = 2.70 to 3.63 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		Wakeup Function	Symbol	Min.*1	Max.*1	Unit	Test Conditions
RIICHS (Standard-mode, SMBus) ICFER.FMPE = 0	SCLHS input cycle time		t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 67.75
	SCLHS input high pulse width		t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCLHS input low pulse width		t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCLHS, SDAHS input rise time		t_{Sr}	—	1000	ns	
	SCLHS, SDAHS input fall time		t_{Sf}	—	300	ns	
	SCLHS, SDAHS input spike pulse removal time		t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDAHS input bus free time	Enabled	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
		Disabled		3(6) × t_{IICcyc} + 4 × t_{PAcyc} + 300	—		
	Start condition input hold time	Enabled	t_{STAH}	t_{IICcyc} + 300	—	ns	
		Disabled		1(5) × t_{IICcyc} + t_{PAcyc} + 300	—		
	Restart condition input setup time		t_{STAS}	1000	—	ns	
	Stop condition input setup time		t_{STOS}	1000	—	ns	
	Data input setup time		t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load		C_b*2	—	400	pF	
RIICHS (Fast-mode)*3 ICFER.FMPE = 0	SCLHS input cycle time		t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	Figure 67.75
	SCLHS input high pulse width		t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCLHS input low pulse width		t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCLHS, SDAHS input rise time		t_{Sr}	20 × (External pull-up voltage/5.5 V)	300	ns	
	SCLHS, SDAHS input fall time		t_{Sf}	20 × (External pull-up voltage/5.5 V)	300	ns	
	SCLHS, SDAHS input spike pulse removal time		t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDAHS input bus free time	Enabled	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
		Disabled		3(6) × t_{IICcyc} + 4 × t_{PAcyc} + 300	—		
	Start condition input hold time	Enabled	t_{STAH}	t_{IICcyc} + 300	—	ns	
		Disabled		1(5) × t_{IICcyc} + t_{PAcyc} + 300	—		
	Restart condition input setup time		t_{STAS}	300	—	ns	
	Stop condition input setup time		t_{STOS}	300	—	ns	
	Data input setup time		t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load		C_b*2	—	400	pF	

Table 67.88 RIICHS Timing (1) (2/2)

Conditions: VCC = 1.62 to 1.98 V or VCC = 2.70 to 3.63 V,
Evaluation was with either of the following driving ability settings for the target pins, depending on the VCC voltage.
When VCC = 1.62 to 1.98 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.
When VCC = 2.70 to 3.63 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		Wakeup Function	Symbol	Min.*1	Max.*1	Unit	Test Conditions
RIICHS (Fast-mode Plus) *3 ICFER.FMPE = 1	SCLHS input cycle time		t_{SCL}	$6(12) \times t_{IICcyc} + 240$	—	ns	Figure 67.75
	SCLHS input high pulse width		t_{SCLH}	$3(6) \times t_{IICcyc} + 120$	—	ns	
	SCLHS input low pulse width		t_{SCLL}	$3(6) \times t_{IICcyc} + 120$	—	ns	
	SCLHS, SDAHS input rise time		t_{Sr}	—	120	ns	
	SCLHS, SDAHS input fall time		t_{Sf}	—	120	ns	
	SCLHS, SDAHS input spike pulse removal time		t_{SP}	0	$1(4) \times t_{IICcyc}$	ns	
	SDAHS input bus free time	Enabled	t_{BUF}	$3(6) \times t_{IICcyc} + 120$	—	ns	
		Disabled		$3(6) \times t_{IICcyc} + 4 \times t_{PAcyc} + 120$	—		
	Start condition input hold time	Enabled	t_{STAH}	$t_{IICcyc} + 120$	—	ns	
		Disabled		$1(5) \times t_{IICcyc} + t_{PAcyc} + 120$	—		
	Restart condition input setup time		t_{STAS}	120	—	ns	
	Stop condition input setup time		t_{STOS}	120	—	ns	
	Data input setup time		t_{SDAS}	$t_{IICcyc} + 30$	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load		C_b*2	—	550	pF	

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for a given RIICHS interface are from the same group.
Note: t_{IICcyc}: RIICHS internal reference clock (IICφ) cycle; t_{PAcyc}: PCLKA cycle
Note 1. Numbers enclosed in parentheses are for the case where the ICFER.NFE bit is set to 1, enabling the digital filter, and the ICMR3.NF[1:0] bits are set to 11b.
Note 2. C_b is the total capacitance of the bus lines.
Note 3. These electrical characteristics are only guaranteed when pins with the suffix -A, indicating the group name, as part of the pin name are in use.

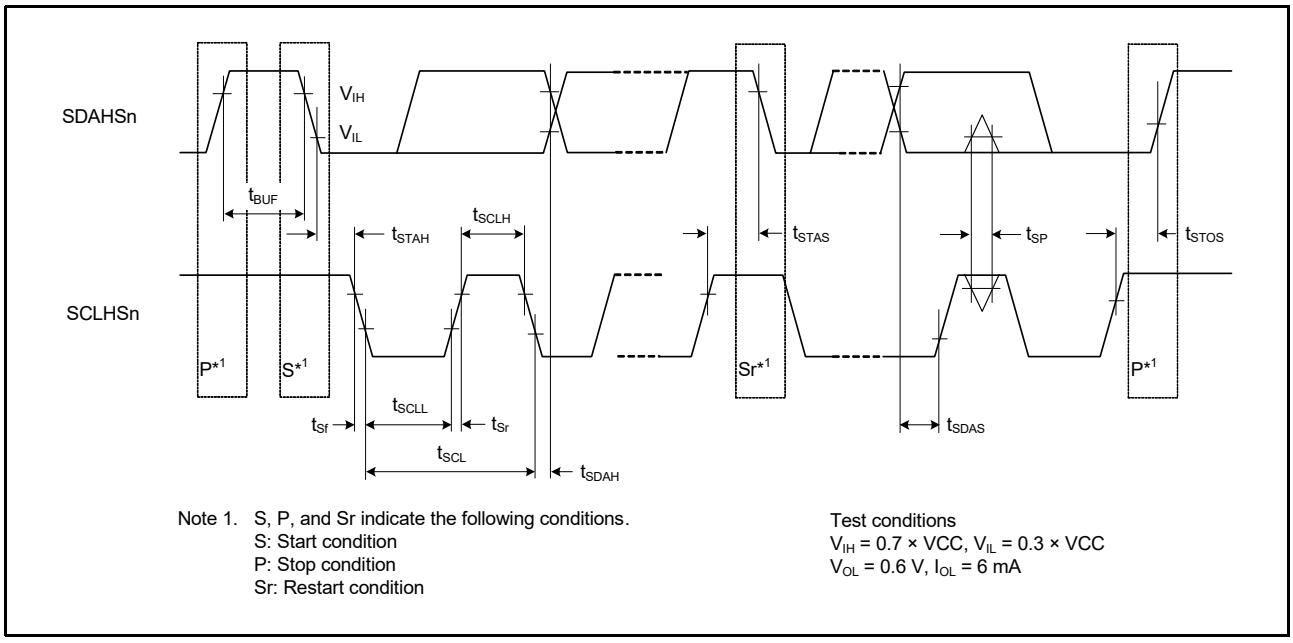


Figure 67.75 RIICHS Bus Interface Input/Output Timing (n = 1, 2)

Table 67.89 RIICHS Timing (2)

Conditions: VCC = 1.65 to 1.95 V or VCC = 2.97 to 3.63 V,
Evaluation was with either of the following driving ability settings for the target pins, depending on the VCC voltage.
When VCC = 1.65 to 1.95 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV3.
When VCC = 2.97 to 3.63 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		Load capacitance*2	Symbol	Min.*1	Max.*1	Unit	Test Conditions
RIICHS (Hs-mode) ICFER.HSME = 1	SCLHS input cycle time		t_{SCL}	46(48) × t_{IICcyc}	—	ns	Figure 67.76
	SCLHS input high pulse width	C_b = 400 pF	t_{SCLH}	29(30) × t_{IICcyc}	—	ns	
		C_b = 100 pF		13(14) × t_{IICcyc}	—		
	SCLHS input low pulse width	C_b = 400 pF	t_{SCLL}	69(70) × t_{IICcyc}	—	ns	
		C_b = 100 pF		33(34) × t_{IICcyc}	—		
	SCLHS input rise time	C_b = 400 pF	t_{SrCL}	—	80	ns	
		C_b = 100 pF		—	40		
	SDAHS input rise time	C_b = 400 pF	t_{SrDA}	—	160	ns	
		C_b = 100 pF		—	80		
	SCLHS input fall time	C_b = 400 pF	t_{SfCL}	—	80	ns	
		C_b = 100 pF		—	40		
	SDAHS input fall time	C_b = 400 pF	t_{SfDA}	—	160	ns	
		C_b = 100 pF		—	80		
	SCLHS, SDAHS input spike pulse removal time		t_{SP}	0	1(1) × t_{IICcyc}	ns	
	Restart condition input setup time		t_{STAS}	40	—	ns	
	Stop condition input setup time		t_{STOS}	40	—	ns	
	Data input setup time		t_{SDAS}	10	—	ns	
	Data input hold time	C_b = 400 pF	t_{SDAH}	0	150	ns	
		C_b = 100 pF		0	70		
	SCLHS, SDAHS capacitive load		C_b*2	—	400	pF	

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for a given RIICHS interface are from the same group.
Note: t_{IICcyc}: RIICHS internal reference clock (IICφ) cycle
Note 1. Numbers enclosed in parentheses are for the case where the ICFER.NFE bit is set to 1, enabling the digital filter, and the ICMR3.NF[1:0] bits are set to 11b.
Note 2. C_b is the total capacitance of the bus lines.

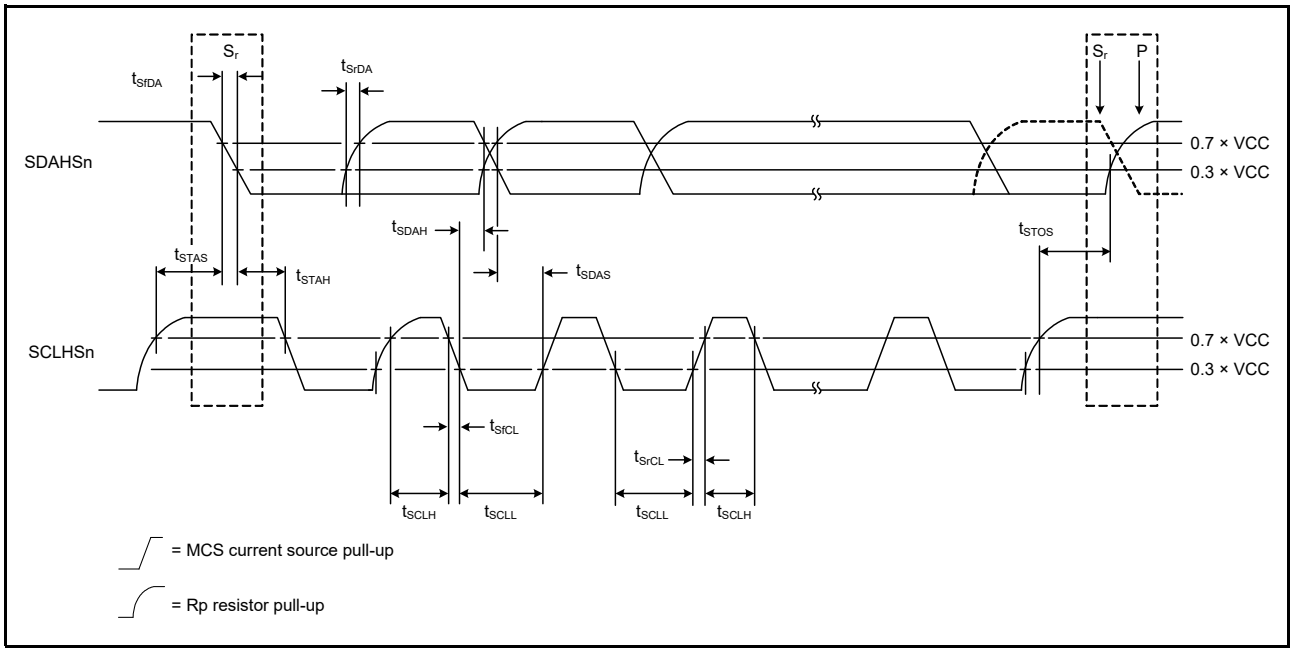


Figure 67.76 RIICHS Bus Interface Input/Output Timing (Hs-mode) (n = 1, 2)

67.5.7.10 RI3C

Table 67.90 RI3C Timing (I²C mode) (1) (1/2)

Conditions: VCC = 1.65 to 1.98 V, or VCC = 2.70 to 3.63 V

Item		Wakeup Function	Symbol	Min.*1	Max.*1	Unit	Test Conditions
I²C mode (Standard-mode, SMBus) ICFER.FMPE = 0	SCL input cycle time		t_{SCL}	10(18) × t_{I3Ccyc} + 1300	—	ns	Figure 67.77
	SCL input high pulse width		t_{SCLH}	5(9) × t_{I3Ccyc} + 300	—	ns	
	SCL input low pulse width		t_{SCLL}	5(9) × t_{I3Ccyc} + 300	—	ns	
	SCL, SDA input rise time		t_{Sr}	—	1000	ns	
	SCL, SDA input fall time		t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time		t_{SP}	0	1(4) × t_{I3Ccyc}	ns	
	SDA input bus free time	Enabled	t_{BUF}	5(9) × t_{I3Ccyc} + 300	—	ns	
		Disabled		5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—		
	Start condition input hold time	Enabled	t_{STAH}	t_{I3Ccyc} + 300	—	ns	
		Disabled		1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—		
	Restart condition input setup time		t_{STAS}	1000	—	ns	
	Stop condition input setup time		t_{STOS}	1000	—	ns	
	Data input setup time		t_{SDAS}	t_{I3Ccyc} + 50	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load		C_b*2	—	400	pF	
I²C mode (Fast-mode)*3 ICFER.FMPE = 0	SCL input cycle time		t_{SCL}	10(18) × t_{I3Ccyc} + 600	—	ns	Figure 67.77
	SCL input high pulse width		t_{SCLH}	5(9) × t_{I3Ccyc} + 300	—	ns	
	SCL input low pulse width		t_{SCLL}	5(9) × t_{I3Ccyc} + 300	—	ns	
	SCL, SDA input rise time		t_{Sr}	20	300	ns	
	SCL, SDA input fall time		t_{Sf}	20 × (External pull-up voltage/3.6 V)	300	ns	
	SCL, SDA input spike pulse removal time		t_{SP}	0	1(4) × t_{I3Ccyc}	ns	
	SDA input bus free time	Enabled	t_{BUF}	5(9) × t_{I3Ccyc} + 300	—	ns	
		Disabled		5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—		
	Start condition input hold time	Enabled	t_{STAH}	t_{I3Ccyc} + 300	—	ns	
		Disabled		1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—		
	Restart condition input setup time		t_{STAS}	300	—	ns	
	Stop condition input setup time		t_{STOS}	300	—	ns	
	Data input setup time		t_{SDAS}	t_{I3Ccyc} + 50	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load		C_b*2	—	400	pF	

Table 67.90 RI3C Timing (I²C mode) (1) (2/2)

Conditions: VCC = 1.65 to 1.98 V, or VCC = 2.70 to 3.63 V

Item		Wakeup Function	Symbol	Min.*1	Max.*1	Unit	Test Conditions
I²C mode (Fast-mode Plus) *3 ICFER.FMPE = 1	SCL input cycle time		t_{SCL}	10(18) × t_{I3Ccyc} + 240	—	ns	Figure 67.77
	SCL input high pulse width		t_{SCLH}	5(9) × t_{I3Ccyc} + 120	—	ns	
	SCL input low pulse width		t_{SCLL}	5(9) × t_{I3Ccyc} + 120	—	ns	
	SCL, SDA input rise time		t_{Sr}	—	120	ns	
	SCL, SDA input fall time		t_{Sf}	20 × (External pull-up voltage/3.6 V)	120	ns	
	SCL, SDA input spike pulse removal time		t_{SP}	0	1(4) × t_{I3Ccyc}	ns	
	SDA input bus free time	Enabled	t_{BUF}	5(9) × t_{I3Ccyc} + 120	—	ns	
		Disabled		5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 120	—		
	Start condition input hold time	Enabled	t_{STAH}	t_{I3Ccyc} + 120	—	ns	
		Disabled		1(5) × t_{I3Ccyc} + t_{Tcyc} + 120	—		
	Restart condition input setup time		t_{STAS}	120	—	ns	
	Stop condition input setup time		t_{STOS}	120	—	ns	
	Data input setup time		t_{SDAS}	t_{I3Ccyc} + 30	—	ns	
	Data input hold time		t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load		C_b*2	—	550	pF	

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the RI3C interface are from the same group.

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle; t_{Tcyc}: TCLK cycle

Note 1. Numbers enclosed in parentheses are for the case where the INTCL.DNFE bit is set to 1, enabling the digital filter, and the INTCL.DNFS[3:0] bits are set to 0011b.

Note 2. C_b is the total capacitance of the bus lines.

Note 3. These electrical characteristics are only guaranteed when pins with the suffix -A, indicating the group name, as part of the pin name are in use.

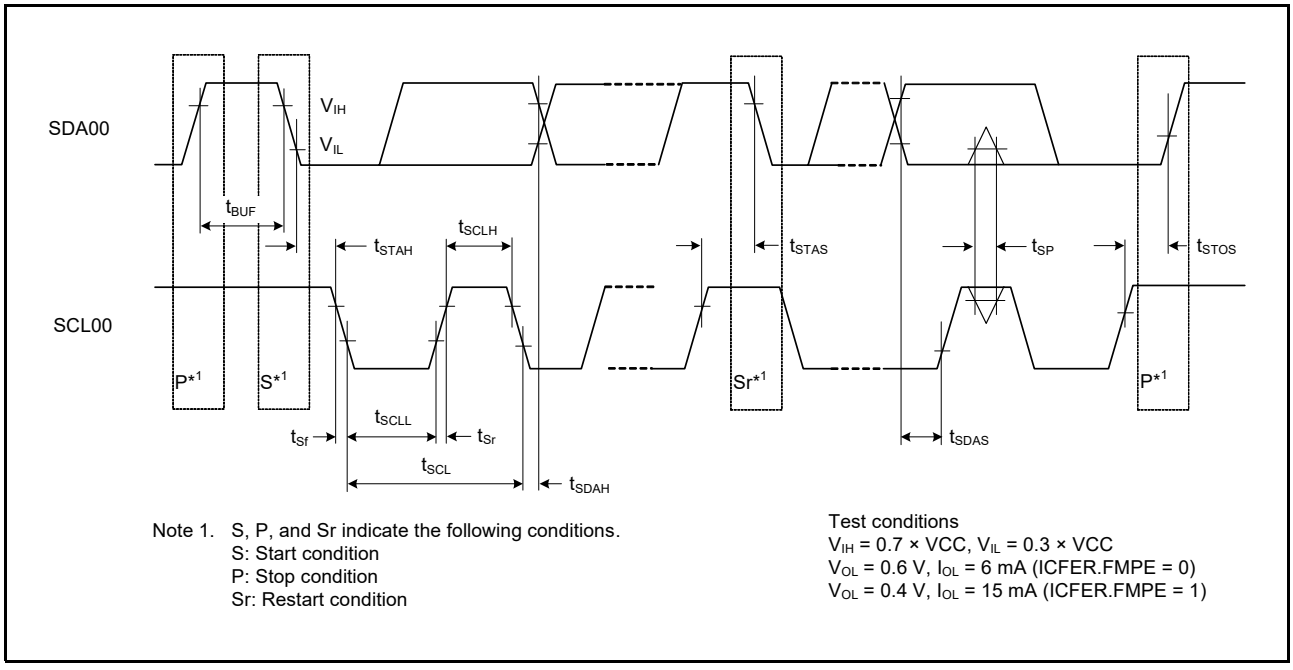


Figure 67.77 I²C Bus Interface Input/Output Timing

Table 67.91 RI3C Timing (I^2C mode) (2)

Conditions: VCC = 1.65 to 1.95 V, or VCC = 2.97 to 3.63 V

Item		Load capacitance*2	Symbol	Min.*1	Max.*1	Unit	Test Conditions
I^2C mode (Hs-mode)*3 ICFER.HSME = 1	SCL input cycle time		t_{SCL}	46(48) × t_{I3Ccyc}	—	ns	Figure 67.78
	SCL input high pulse width	C_b = 400 pF	t_{SCLH}	29(30) × t_{I3Ccyc}	—	ns	
		C_b = 100 pF		13(14) × t_{I3Ccyc}	—		
	SCL input low pulse width	C_b = 400 pF	t_{SCLL}	69(70) × t_{I3Ccyc}	—	ns	
		C_b = 100 pF		33(34) × t_{I3Ccyc}	—		
	SCL input rise time	C_b = 400 pF	t_{SrCL}	—	80	ns	
		C_b = 100 pF		—	40		
	SDA input rise time	C_b = 400 pF	t_{SrDA}	—	160	ns	
		C_b = 100 pF		—	80		
	SCL input fall time	C_b = 400 pF	t_{SfCL}	—	80	ns	
		C_b = 100 pF		—	40		
	SDA input fall time	C_b = 400 pF	t_{SfDA}	—	160	ns	
		C_b = 100 pF		—	80		
	SCL, SDA input spike pulse removal time		t_{SP}	0	1(1) × t_{I3Ccyc}	ns	
	Restart condition input setup time		t_{STAS}	40	—	ns	
	Stop condition input setup time		t_{STOS}	40	—	ns	
	Data input setup time		t_{SDAS}	10	—	ns	
	Data input hold time	C_b = 400 pF	t_{SDAH}	0	150	ns	
		C_b = 100 pF		0	70		
	SCL, SDA capacitive load		C_b*2	—	400	pF	

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle

Note 1. Numbers enclosed in parentheses are for the case where the INTCL.DNFE bit is set to 1, enabling the digital filter, and the INTCL.DNFS[3:0] bits are set to 0011b.

Note 2. C_b is the total capacitance of the bus lines.

Note 3. These electrical characteristics are only guaranteed when pins with the suffix -A, indicating the group name, as part of the pin name are in use.

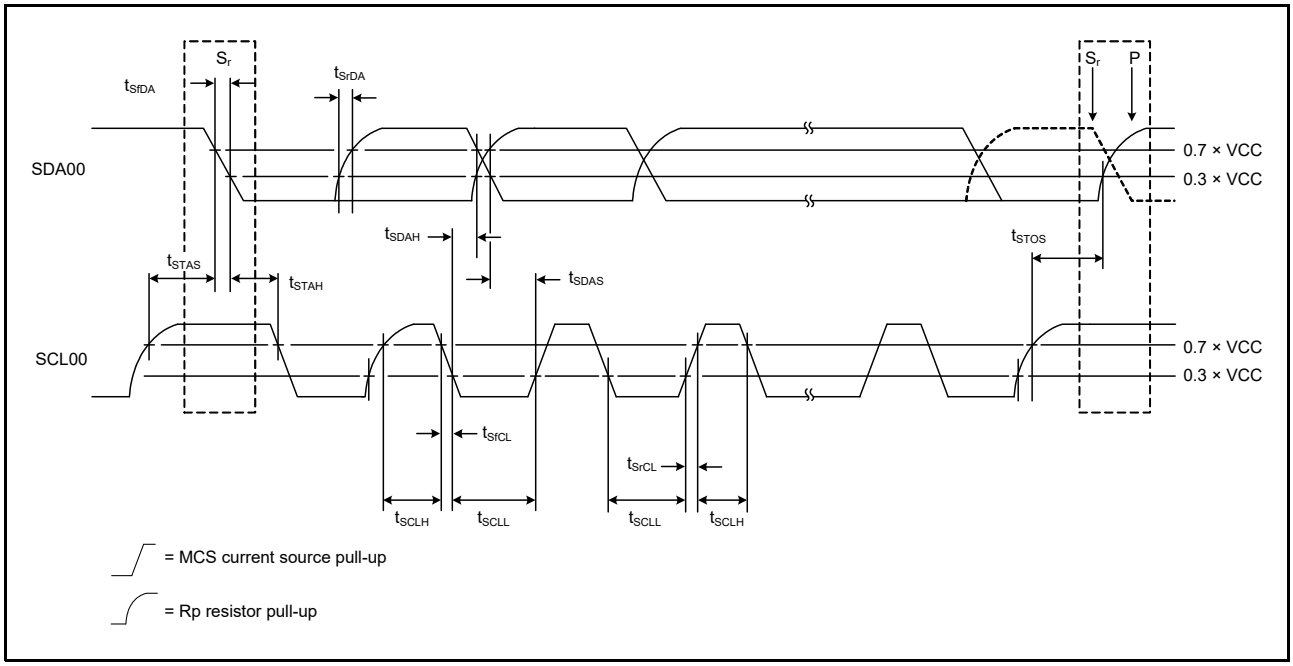


Figure 67.78 I^2C Bus Interface Input/Output Timing (Hs-mode)

Table 67.92 RI3C Timing (Open-Drain Timing Parameters)

Conditions: VCC = 1.65 to 1.95 V, or VCC = 2.97 to 3.63 V

Item	Item	VCC	Symbol	Min.	Max.	Unit	Test Conditions
I3C mode (Open-drain timing parameter)	SCL clock low period		t_{LOW_OD}	200*1, *2	—	ns	Figure 67.80
			$t_{DIG_OD_L}$	t_{LOW_ODmin} + t_{fDA_ODmin}	—		
	SCL clock high period		t_{HIGH}	—	41*3, *4	ns	Figure 67.79
			t_{DIG_H}	—	t_{HIGH} + t_{CF}		
	SDA signal fall time		t_{fDA_OD}	t_{CF}	12	ns	Figure 67.80
	SDA data setup time open drain mode	3.00 V or higher	t_{SU_OD}	12	—	ns	Figure 67.80, Figure 67.81
		1.65 to 1.95 V		18	—		
	Clock after START (S) condition		t_{CAS}	38.4 ns*5	ENAS0: 1 µs	—	Figure 67.80
					ENAS1: 100 µs		
					ENAS2: 2 ms		
					ENAS3: 50 ms*6		
	Clock before STOP (P) condition		t_{CBP}	t_{CASmin}/2	—	s	Figure 67.82
	Current controller to secondary controller overlap time during handoff		$t_{CRHPOverlap}$	$t_{DIG_OD_Lmin}$	—	ns	Figure 67.83
	Bus available condition		t_{AVAL}	1*7	—	µs	—
	Bus idle condition		t_{IDLE}	1	—	ms	—
	Time internal where new controller not driving SDA low		$t_{NEWCRLock}$	$t_{AVALmin}$	—	µs	Figure 67.83

Note: This table only applies when pins with the suffix -A, indicating the group name, as part of the pin name are in use.
Note 1. This is approximately equal to t_{LOWmin} + t_{DS_ODmin} + t_{rDA_ODtyp} + t_{SU_ODmin}.
Note 2. The controller may use a shorter low period if it knows that this is safe, i.e., that SDA is already above V_{IH}.
Note 3. This value is based on t_{SPIKE}, rise and fall times, and interconnects.
Note 4. This maximum high period may be exceeded if the signal can be safely verified by legacy I^2C devices or if interconnect limitations are taken into consideration (such as short bus lengths). If this maximum value cannot be guaranteed by the product specifications, change this value and specify that it cannot be used on mixed buses.
Note 5. On a legacy bus where I^2C devices need to see start, the t_{CAS} Min value is further constrained.
Note 6. Targets that do not support the optional ENTASx CCCs shall use the t_{CAS} Max value shown for ENTAS3
Note 7. On a mixed bus with Fm legacy I^2C devices, t_{AVAL} is 300 ns shorter than the Fm bus free condition time (t_{BUF})

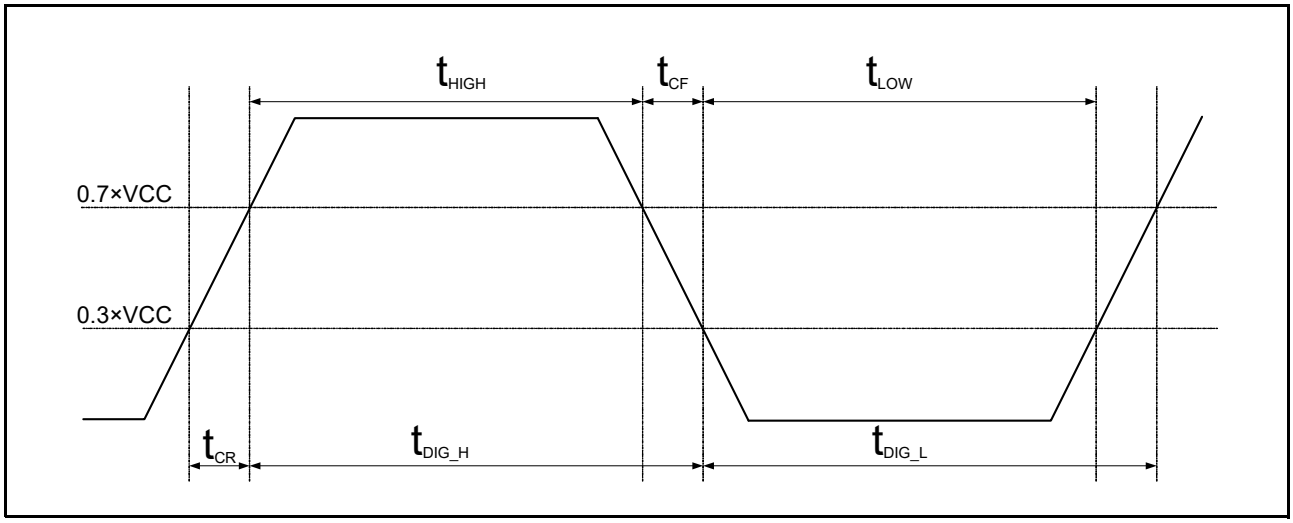


Figure 67.79 t_{DIG_H} and t_{DIG_L}

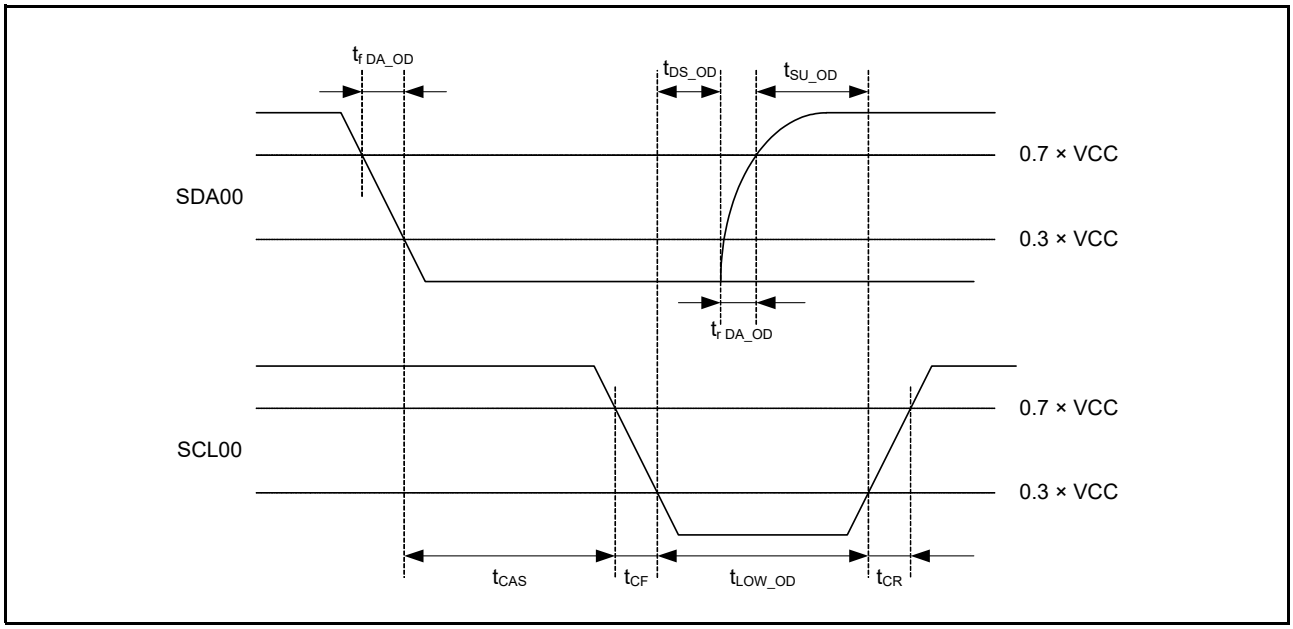


Figure 67.80 RI3C Start Condition Timing

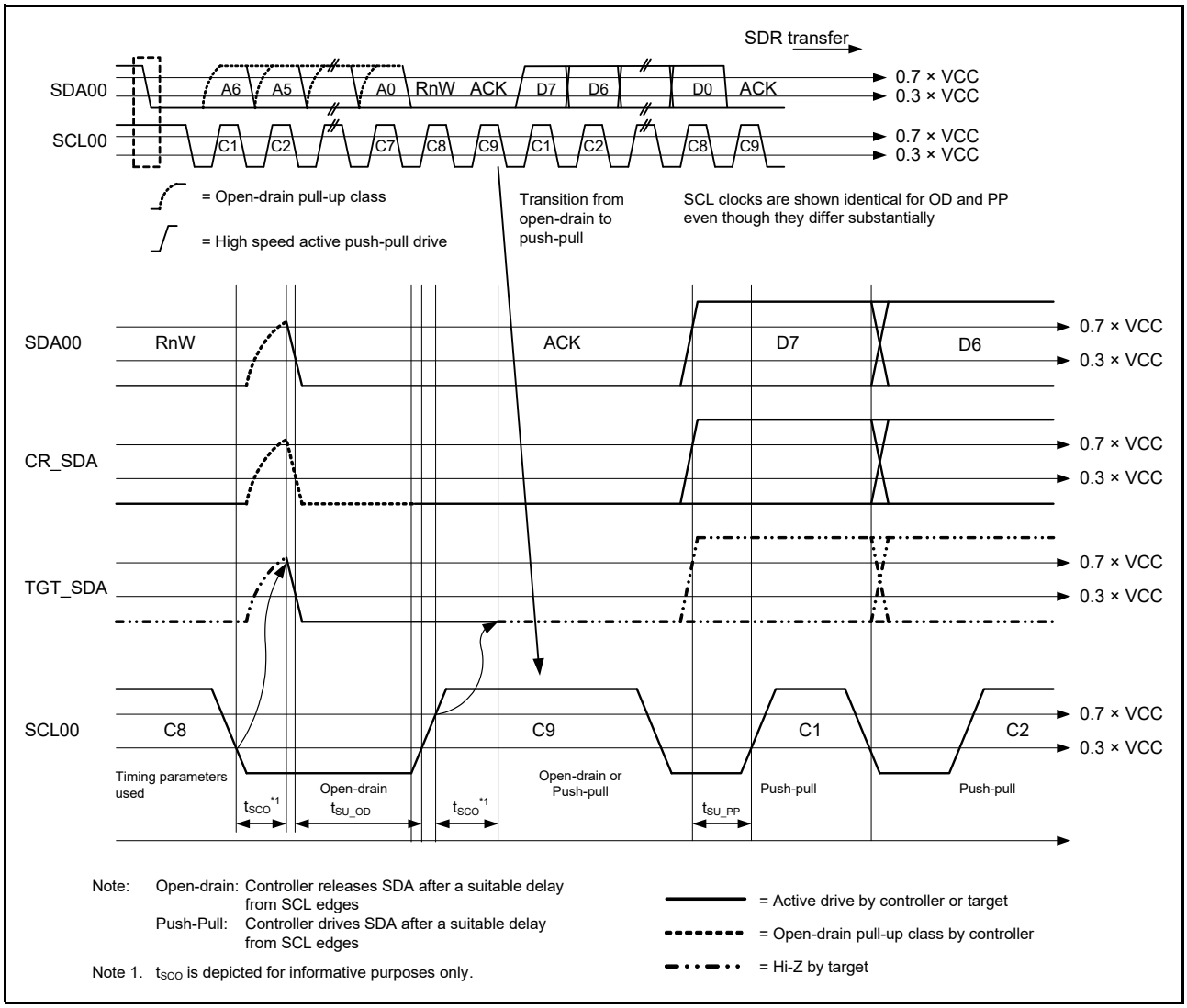


Figure 67.81 RI3C Data Transfer — ACK by Target

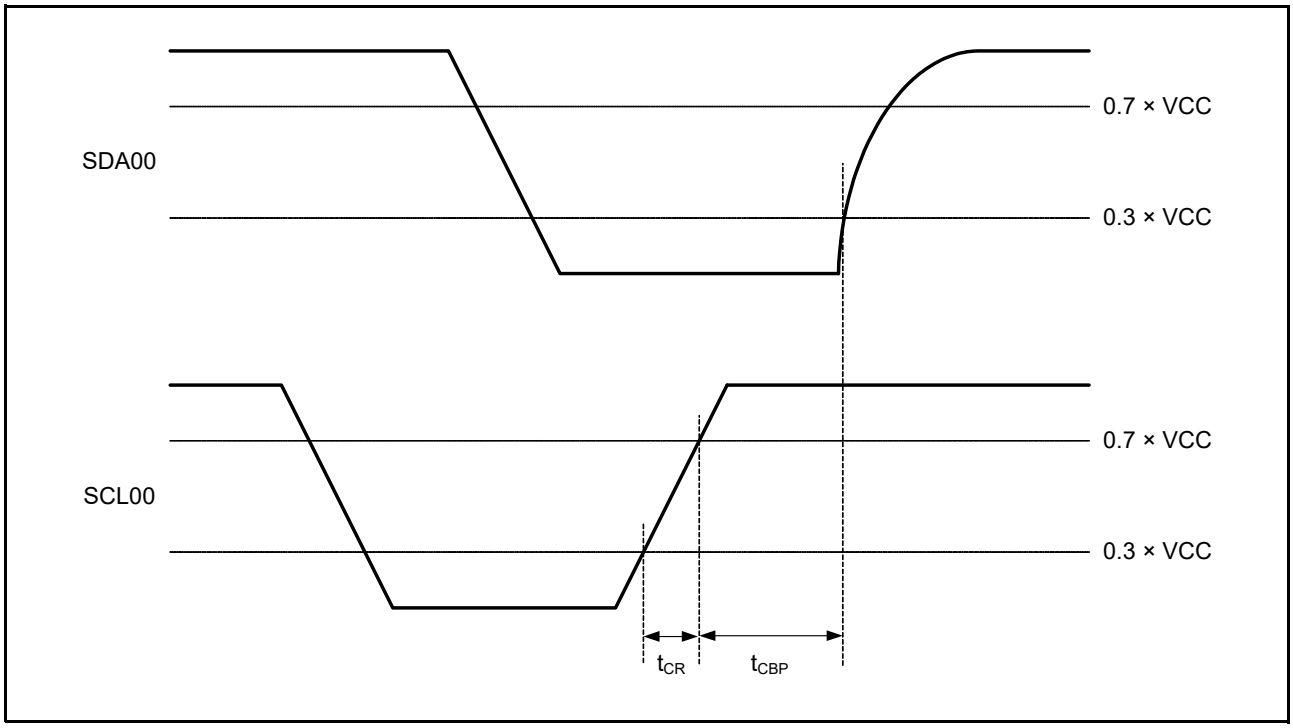


Figure 67.82 RI3C Stop Condition Timing

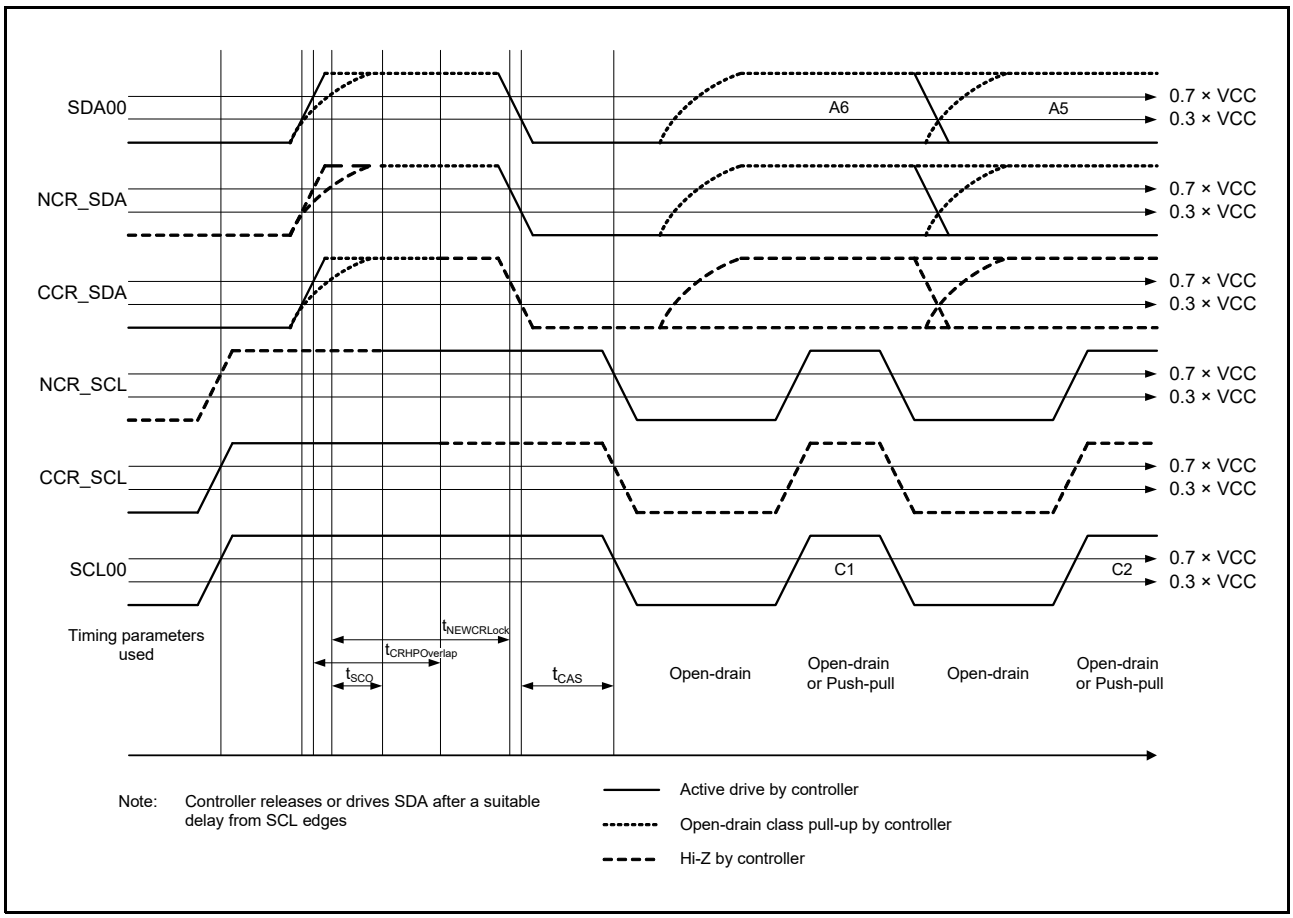


Figure 67.83 RI3C Output Timing

Table 67.93 RI3C Timing (Push-Pull Timing Parameters for SDR)

Conditions: VCC = 1.65 to 1.95 V, or VCC = 2.97 to 3.63 V

Item	Item	VCC	Symbol	Min.	Max.	Unit	Test Conditions
I3C mode (Push-pull timing parameters for SDR)	SCL clock frequency		f_{SCL}*1	0.01	12.5	MHz	—
	SCL clock low period	3.00 V or higher	t_{LOW}	24	—	ns	Figure 67.80
		1.65 to 1.95 V		24	—		
		3.00 V or higher	t_{DIG_L} *2, *4	32	—		
		1.65 to 1.95 V		32	—		
	SCL clock high period for mixed bus*3	3.00 V or higher	t_{HIGH_MIXED}	24	—	ns	Figure 67.80
		1.65 to 1.95 V		24	—		
		3.00 V or higher	$t_{DIG_H_MIXED}$	32	45		
		1.65 to 1.95 V		32	45		
	SCL clock high period	3.00 V or higher	t_{HIGH}	24	—	ns	Figure 67.80
		1.65 to 1.95 V		24	—		
		3.00 V or higher	t_{DIG_H}*2	32	—		
		1.65 to 1.95 V		32	—		
	Clock in to data out for target		t_{SCO}	—	12	ns	Figure 67.84
	SCL clock rise time		t_{CR}	—	150 × 1 /f_{SCL}*5	ns	Figure 67.80
	SCL clock fall time		t_{CF}	—	150 × 1 /f_{SCL}*5	ns	
	SDA signal data hold in push-pull mode (Controller)		t_{HD_PP}*6	t_{CR} + 3*4, t_{CF} + 3*4	—	ns	Figure 67.84
	SDA signal data hold in push-pull mode (Target)			0	—		Figure 67.85
	SDA signal data setup in push-pull mode	3.00 V or higher	t_{SU_PP}	12	N/A	ns	Figure 67.86
		1.65 to 1.95 V		18	N/A		
	Clock after repeated start (Sr)		t_{CASr}	t_{CASmin}	N/A	ns	Figure 67.87
	Clock before repeated start (Sr)		t_{CBSr}	t_{CASmin}/2	N/A	ns	
	Capacitive load per bus line (SDA/SCL)		C_b*7	—	50	pF	—

Note: This table only applies when pins with the suffix -A, indicating the group name, as part of the pin name are in use.
Note 1. $f_{SCL} = 1 / (t_{DIG_L} + t_{DIG_H})$
Note 2. t_{DIG_L} and t_{DIG_H} are the clock low and high periods as seen at the receiver end of the I3C bus using V_{IL} and V_{IH}.
Note 3. In communication with an I3C device on a mixed bus, the $t_{DIG_H_MIXED}$ period must be constrained to ensure that I^2C devices do not interpret I3C signals as valid I^2C signals.
Note 4. As both edges are used, the hold time must be satisfied for the respective edges, for example, t_{CF} + 3 for falling edge clocks, and t_{CR} + 3 for rising edge clocks.
Note 5. The upper limit is 60 ns.
Note 6. The hold time parameter is referred to as t_{HD_SDR} in SDR mode.
Note 7. C_b is the total capacitance of the bus lines.

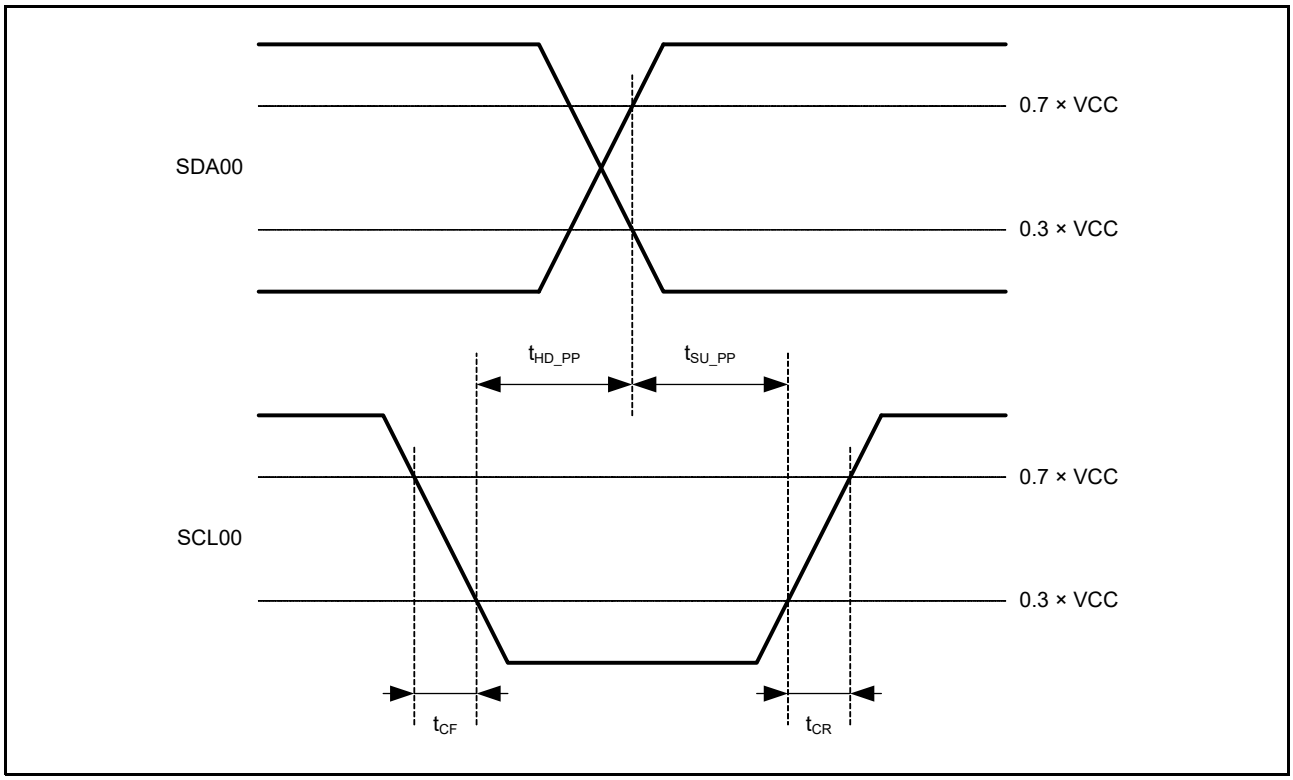


Figure 67.84 I3C Current Controller Output Timing

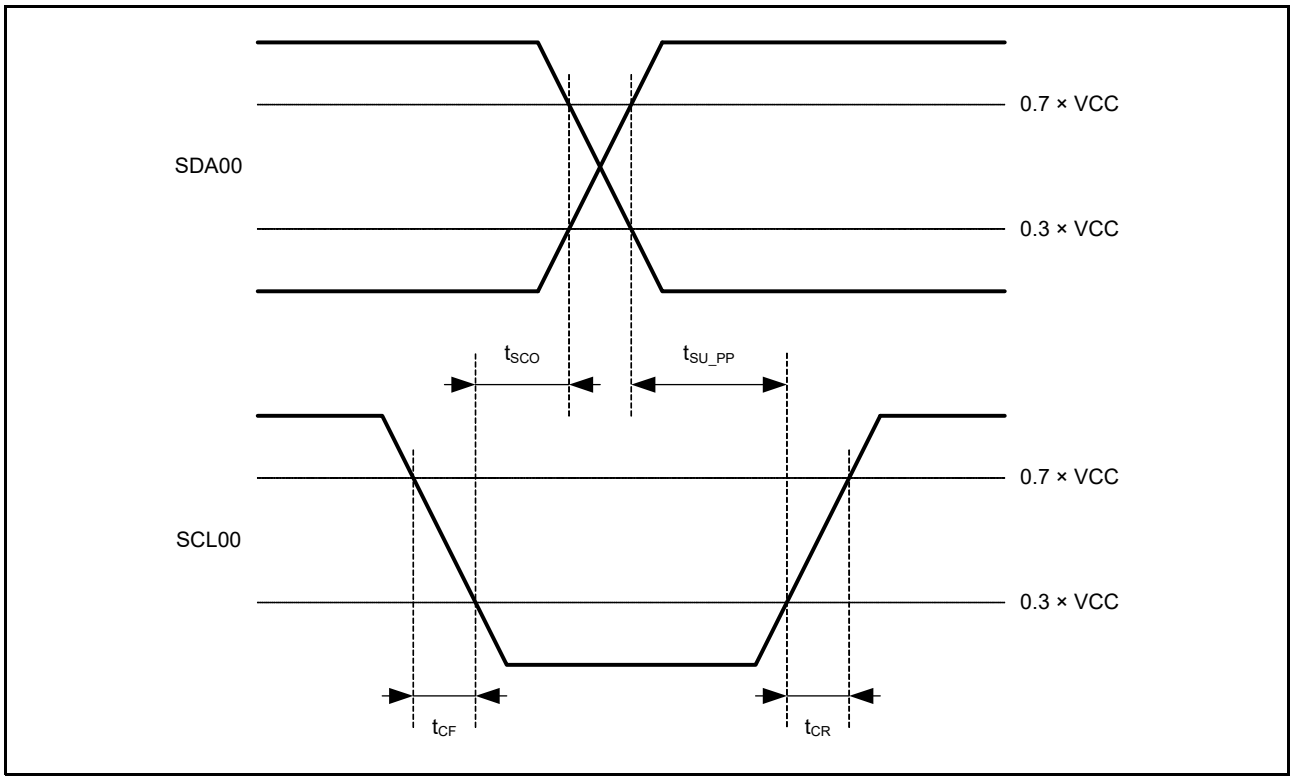


Figure 67.85 I3C Secondary Controller Output Timing

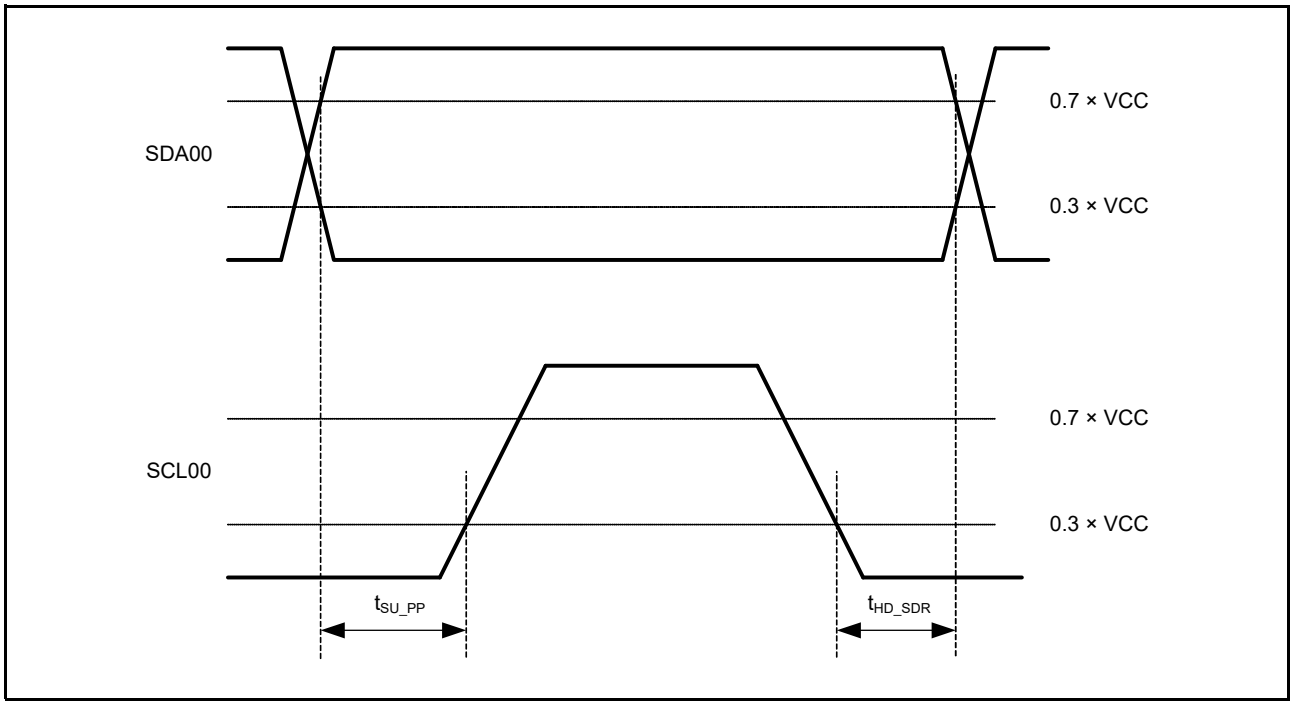


Figure 67.86 Current Controller SDR Timing

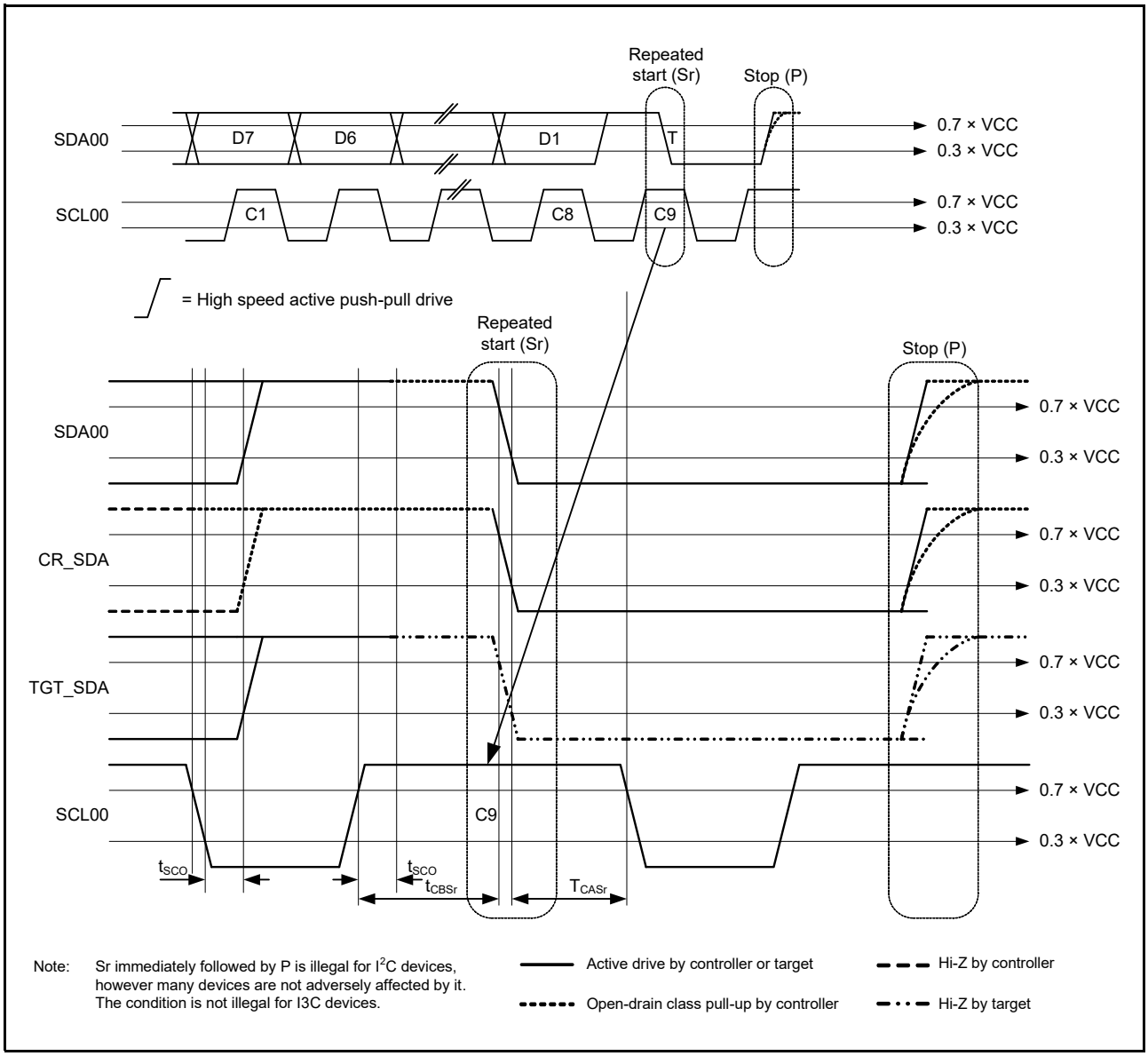


Figure 67.87 T-Bit When Controller Ends Read with Repeated Start and Stop

67.5.7.11 SSIE

Table 67.94 Expansion Serial Sound Interface Timing

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item			VCC/VCC2	Symbol	Min.	Max.	Unit	Test Conditions
AUDIO_CLK, GTIOC14A	Cycle		2.70 V or higher	t_{EXcyc}	20	—	ns	Figure 67.88
			1.65 V or higher		40	—		
	High/low level			t_{EXL}/t_{EXH}	0.4	0.6	t_{EXcyc}	
SSIBCKn	Cycle	Master		t_O	80	—	ns	Figure 67.89
		Slave		t_I	80	—		
	Output clock high level	Master		t_{HC}	0.35	—	t_O	
	Output clock low level			t_{LC}	0.35	—		
	Input clock high level	Slave		t_{HC}	0.35	—	t_I	
	Input clock low level			t_{LC}	0.35	—		
	Output clock rise time	Master		t_{RC}	—	0.15	t_O	
	Output clock fall time			t_{FC}	—	0.15		
	Input clock rise time	Slave		t_{RC}	—	0.15	t_I	
	Input clock fall time			t_{FC}	—	0.15		
SSILRCKn, SSITXDn, SSIRXDn	Input setup time	Master	2.70 V or higher	t_{SR}	12	—	ns	Figure 67.90, Figure 67.91
			1.65 V or higher		24	—		
		Slave			12	—		
	Input hold time	Master		t_{HR}	8	—	ns	
		Slave			15	—		
	Output delay time	Master	2.70 V or higher	t_{DTR}	–10	5	ns	
			1.65 V or higher		–10	7		
		Slave	2.70 V or higher		0	20		
			1.65 V or higher		0	25		
	Output delay time from when an SSILRCK0 signal is changed*1	Slave	2.70 V or higher	t_{DTRW}	—	20	ns	Figure 67.92
			1.65 V or higher		—	25		

n = 0, 1

Note 1. The SSIE has a single path for transmission in slave mode. To generate the data for transmission, the signals input through the SSILRCKn pin through the abovementioned path are used. After that, the data for transmission proceed to be used as the logical outputs to the SSITXDn pin.

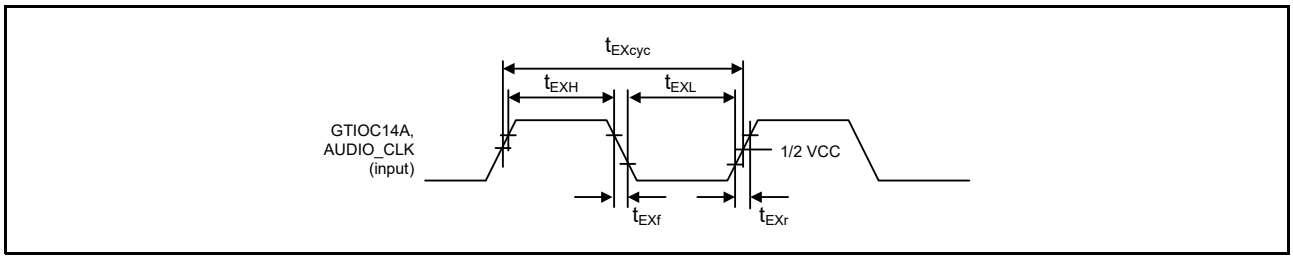


Figure 67.88 Clock Input Timing

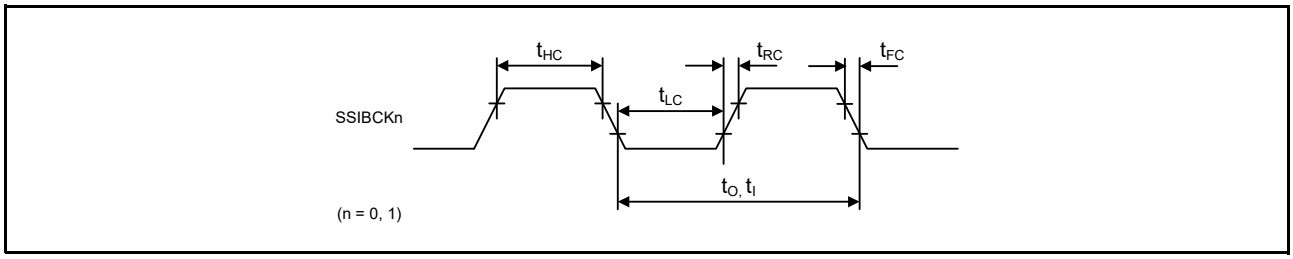


Figure 67.89 SSIE Clock Input/Output Timing

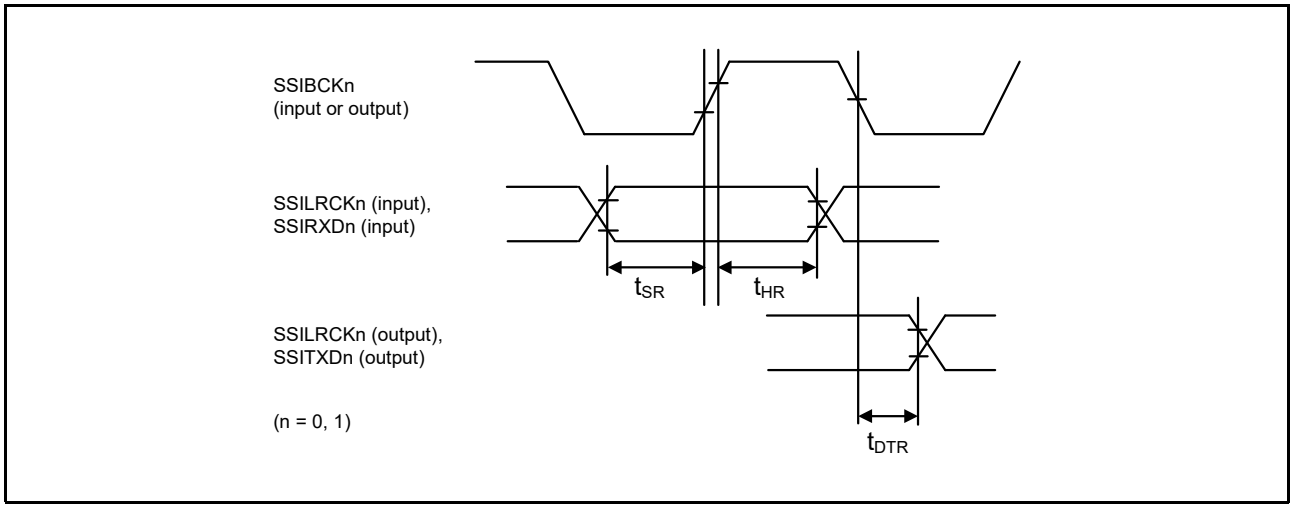


Figure 67.90 Transmission and Reception Timing for the SSIE Data When the SSICR.BCKP Bit is 0

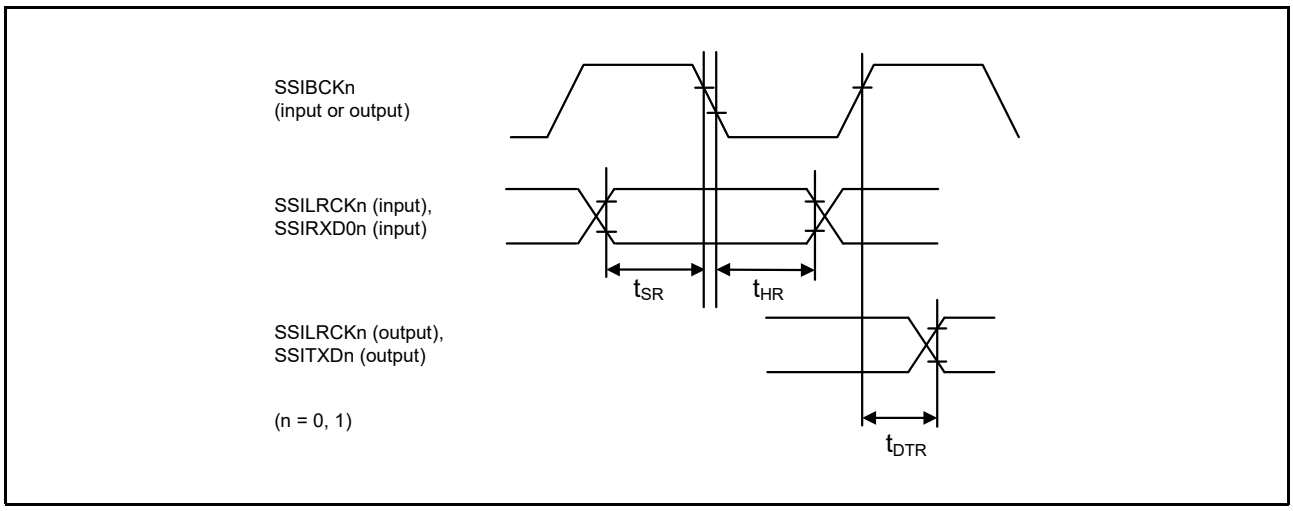


Figure 67.91 Transmission and Reception Timing for the SSIE Data When the SSICR.BCKP Bit is 1

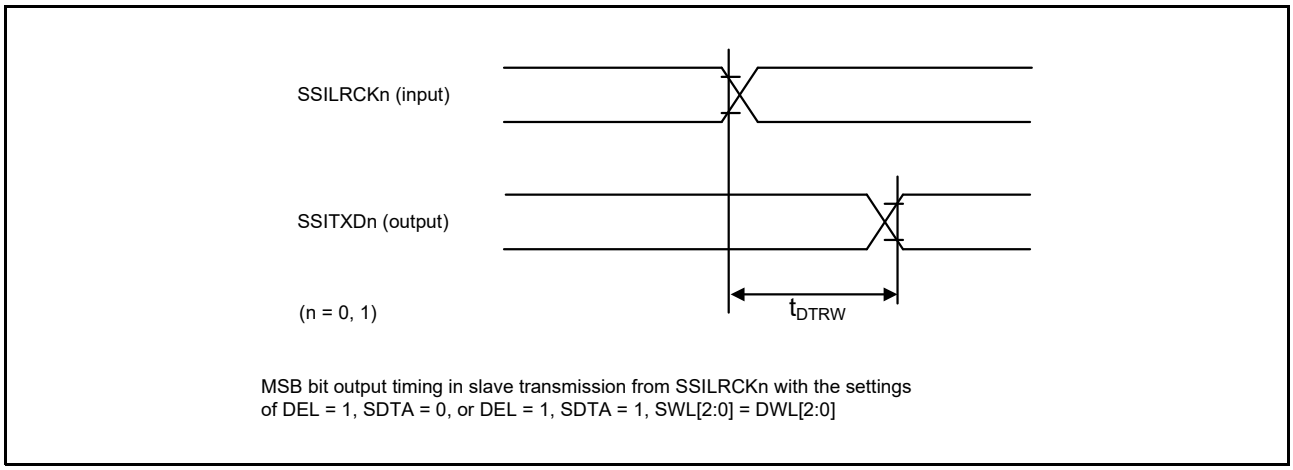


Figure 67.92 Output Delay of the SSIE Data from When an SSILRCKn Signal is Changed

67.5.7.12 SD/MMC Host Interface

Table 67.95 SD/MMC Host Interface Timing

Conditions: VCC = 1.70 to 1.95 V, or VCC = 2.70 to 3.63 V
Evaluation was with either of the following driving ability settings for the target pins, depending on the VCC voltage.
When VCC = 1.70 to 1.95 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV6.
When VCC = 2.70 to 3.63 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV3.

Item		VCC/VCC2	Symbol	Min.	Max.	Unit	Test Conditions
SDHI	SDCLK clock cycle		t_{SDcyc}	20	—	ns	Figure 67.93
	SDCLK clock high level width		t_{SDWH}	6.5	—	ns	
	SDCLK clock low level width		t_{SDWL}	6.5	—	ns	
	SDCLK clock rising time	2.70 V or higher	t_{SDLH}	—	3	ns	
		1.70 V to 1.95 V		—	4		
	SDCLK clock falling time	2.70 V or higher	t_{SDHL}	—	3	ns	
		1.70 V to 1.95 V		—	4		
	SDCMD/SDDAT output data delay	2.70 V or higher	t_{SDODLY}	–7	4	ns	
		1.70 V to 1.95 V		–7	7		
	SDCMD/SDDAT input data setup time		t_{SDIS}	4.5	—	ns	
	SDCMD/SDDAT input data hold time		t_{SDIH}	1.5	—	ns	

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the SD/MMC host interface are from the same group.
Note: If SD0_DAT4-B to SD0_DAT7-B are used, characteristics above is guaranteed only when VCC = VCC2.

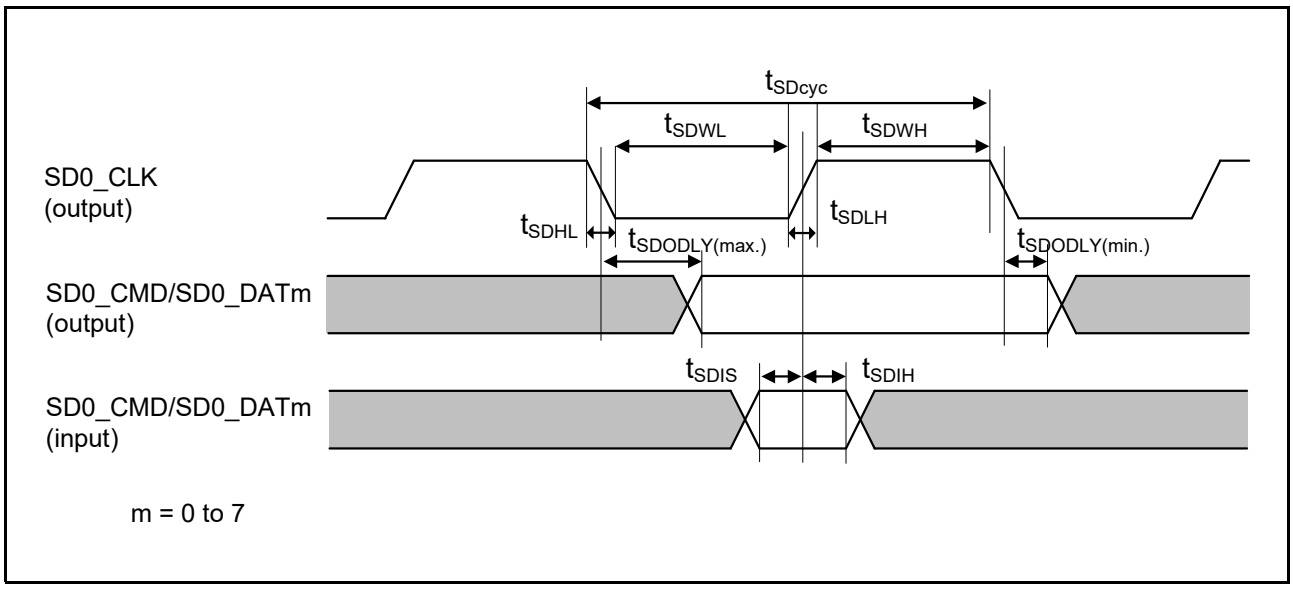


Figure 67.93 SD/MMC Interface Input/Output Signal Timing

67.5.7.13 PDC

Table 67.96 PDC Timing

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
PDC	PIXCLK input cycle time	t_{PIXcyc}	37	—	ns	Figure 67.94
	PIXCLK input high pulse width	t_{PIXH}	10	—	ns	
	PIXCLK input low pulse width	t_{PIXL}	10	—	ns	
	PIXCLK rising time	t_{PIXr}	—	5	ns	
	PIXCLK falling time	t_{PIXf}	—	5	ns	
	PCKO output cycle time	t_{PCKcyc}	$2 \times t_{PBcyc}$	—	ns	Figure 67.95
	PCKO output high pulse width	t_{PCKH}	$(t_{PCKcyc} - t_{PCKr} - t_{PCKf})/2 - 3$	—	ns	
	PCKO output low pulse width	t_{PCKL}	$(t_{PCKcyc} - t_{PCKr} - t_{PCKf})/2 - 3$	—	ns	
	PCKO rising time	t_{PCKr}	—	5	ns	
	PCKO falling time	t_{PCKf}	—	5	ns	
	VSYNC/HSYNC input setup time	t_{SYNCS}	10	—	ns	Figure 67.96
	VSYNC/HSYNC input hold time	t_{SYNCH}	5	—	ns	
	PIXD input setup time	t_{PIXDS}	10	—	ns	
	PIXD input hold time	t_{PIXDH}	5	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle

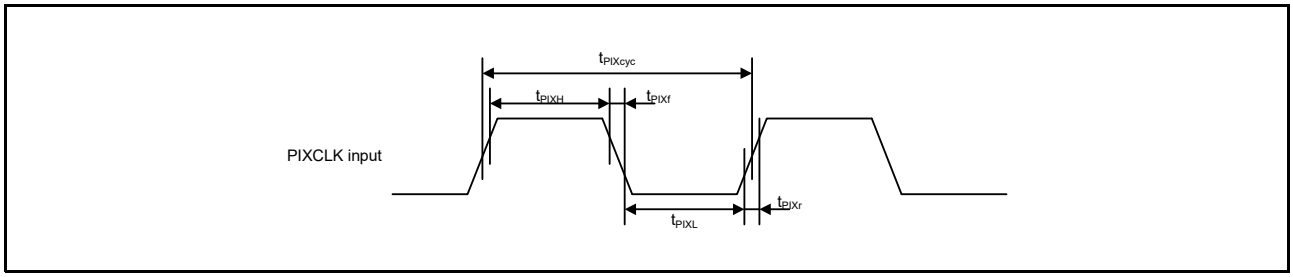


Figure 67.94 PDC Input Clock Timing

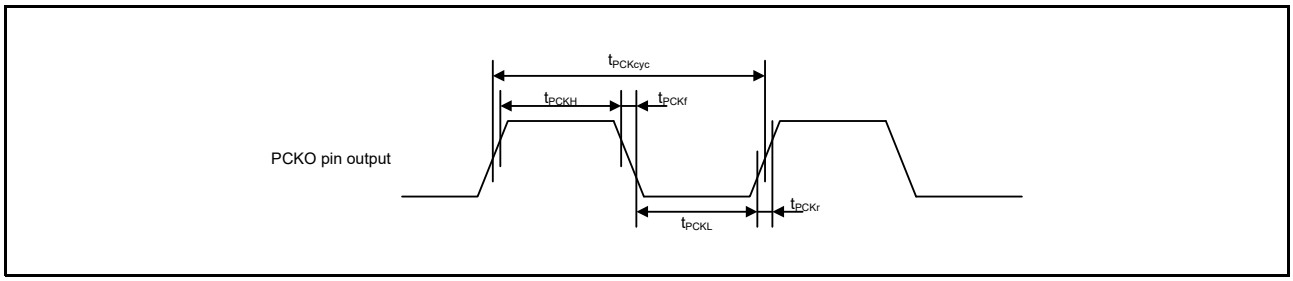


Figure 67.95 PDC Output Clock Timing

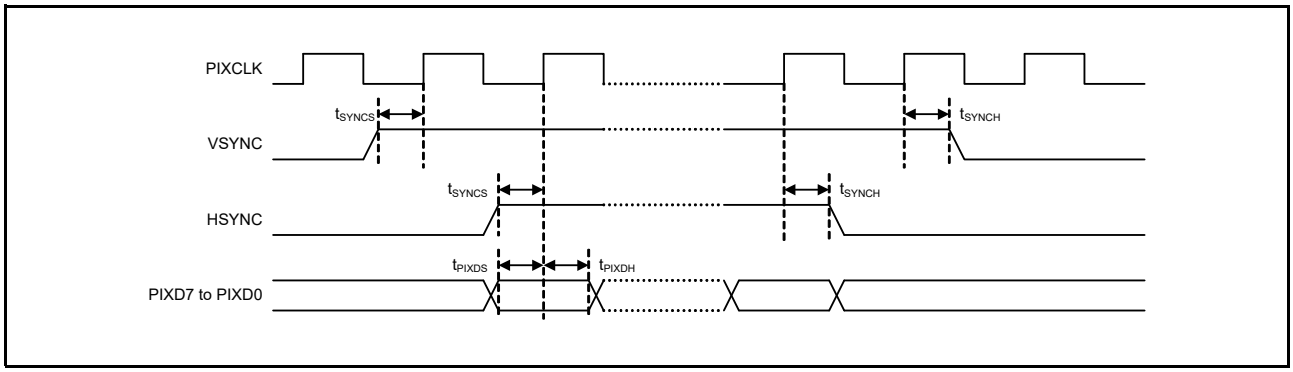


Figure 67.96 PDC AC Timing

67.5.7.14 HRPWM

Table 67.97 HRPWM Timing

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HRPWM	Input frequency (f_{IN})	t_{cyc}	80	—	200	MHz	
	Resolution	—	1/64	—	—	t_{cyc}	f_{IN} > 160 MHz
			1/128	—	—		f_{IN} ≤ 160 MHz
	DNL*1	—	—	±2.0	—	LSB	

Note 1. The value is that difference from code to code normalized by the resolution (1 LSB).

67.5.7.15 CANFD/CANFDG

Table 67.98 CANFD/CANFDG Timing

Conditions: When VCC/VCC2 ≥ 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV1.
When VCC/VCC2 ≥ 1.62 V: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item	Symbol	Min.	Max.	Unit	Test Conditions
Internal delay time	t_{NODE}	—	50	ns	Figure 67.97

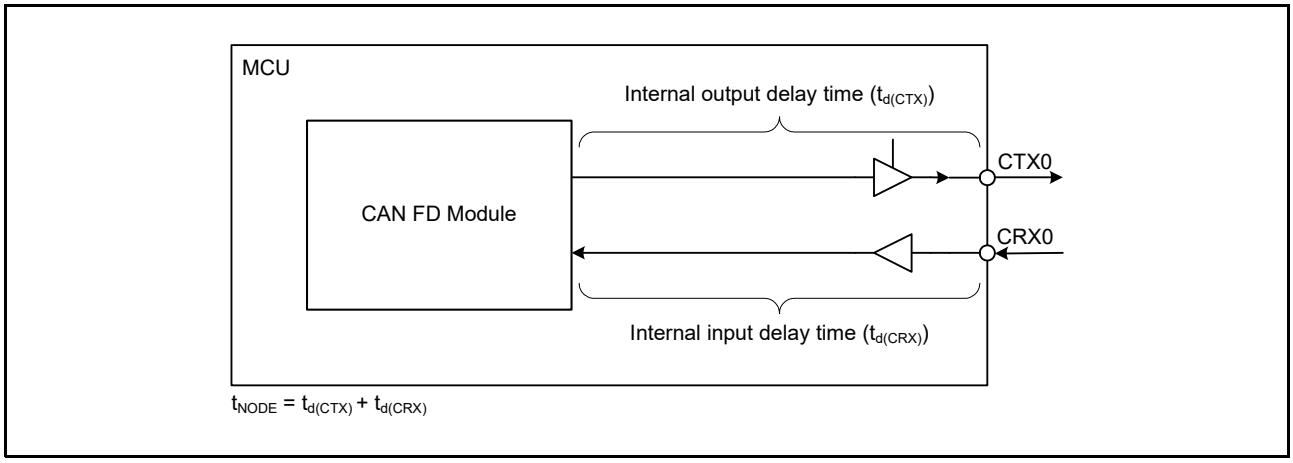


Figure 67.97 Definition of Internal Delay Time

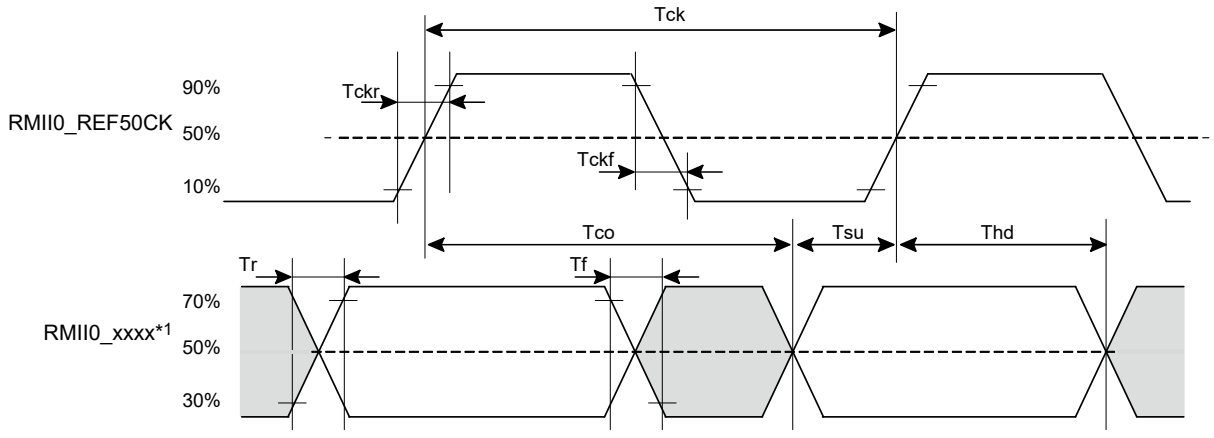
67.5.7.16 ESWM

Table 67.99 ESWM Timing (RMII)

Conditions: VCC = 2.7 to 3.6 V
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the ET0_MDC and ET0_MDIO pins are set for DRV2.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than ET0_MDC and ET0_MDIO are set for DRV3.

Item		Symbol	Min.	Max.	Unit	Test Conditions
ESWM (RMII)	RMII0_REF50CK cycle time	T_{ck}	20	—	ns	Figure 67.98
	RMII0_REF50CK frequency	—	—	50 + 100 ppm	MHz	
	RMII0_REF50CK duty	—	35	65	%	
	RMII0_REF50CK rise/fall time	T_{ckr} / T_{ckf}	0.5	3.5	ns	
	RMII0_xxxx*1 output delay time	T_{co}	2.5	15	ns	
	RMII0_xxxx*2 setup time	T_{su}	3	—	ns	
	RMII0_xxxx*2 hold time	T_{hd}	1	—	ns	
	RMII0_xxxx*1, *2 rise/fall time	T_r / T_f	0.5	4	ns	

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the ESWM are from the same group.
Note 1. RMII0_TX_EN, RMII0_TXD1, RMII0_TXD0
Note 2. RMII0_CRS_DV, RMII0_RXD1, RMII0_RXD0, RMII0_RX_ER



Note 1. RMII0_TX_EN, RMII0_TXD1, RMII0_TXD0, RMII0_CRS_DV, RMII0_RXD1, RMII0_RXD0, RMII0_RX_ER

Figure 67.98 Timing with the RMII0_REF50CK and RMII Signals

Table 67.100 ESWM Timing (MII)

Conditions: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the ET0_MDC and ET0_MDIO pins are set for DRV2.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than ET0_MDC and ET0_MDIO are set as follows.
With only the MII in use:
The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV2 and VCC is in the range from 2.7 to 3.6 V.
With both the MII and GMII in use:
When VCC = 2.3 to 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV7.
When VCC = 3.0 to 3.6 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV6.

Item		Symbol	Min.	Max.	Unit	Test Conditions
ESWM (MII)	ET0_TX_CLK cycle time	t_{Tcyc}	40	—	ns	—
	ET0_TX_EN output delay time	t_{TENd}	1	20		Figure 67.99
	ET0_ETXD0 to ET0_ETXD3 output delay time	t_{MTDd}	1	20		
	ET0_TX_ER output delay time	t_{TERd}	1	20		Figure 67.100
	ET0_RX_CLK cycle time	t_{TRcyc}	40	—		—
	ET0_RX_DV setup time	t_{RDVs}	10	—		Figure 67.101
	ET0_RX_DV hold time	t_{RDVh}	10	—		
	ET0_ERXD0 to ET0_ERXD3 setup time	t_{MRDs}	10	—		
	ET0_ERXD0 to ET0_ERXD3 hold time	t_{MRDh}	10	—		
	ET0_RX_ER setup time	t_{RERs}	10	—		
	ET0_RX_ER hold time	t_{RERh}	10	—		
ESWM (MII MAC-to-MAC)	ET0_TX_CLK cycle time	t_{Tcyc}	40	—	ns	—
	ET0_TX_EN output delay time	t_{TENd}	10	20		Figure 67.99
	ET0_ETXD0 to ET0_ETXD3 output delay time	t_{MTDd}	10	20		
	ET0_TX_ER output delay time	t_{TERd}	10	20		Figure 67.100
	ET0_RX_CLK cycle time	t_{TRcyc}	40	—		—
	ET0_RX_DV setup time	t_{RDVs}	13	—		Figure 67.101
	ET0_RX_DV hold time	t_{RDVh}	0	—		
	ET0_ERXD0 to ET0_ERXD3 setup time	t_{MRDs}	13	—		
	ET0_ERXD0 to ET0_ERXD3 hold time	t_{MRDh}	0	—		
	ET0_RX_ER setup time	t_{RERs}	13	—		
	ET0_RX_ER hold time	t_{RERh}	0	—		

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the ESWM are from the same group.

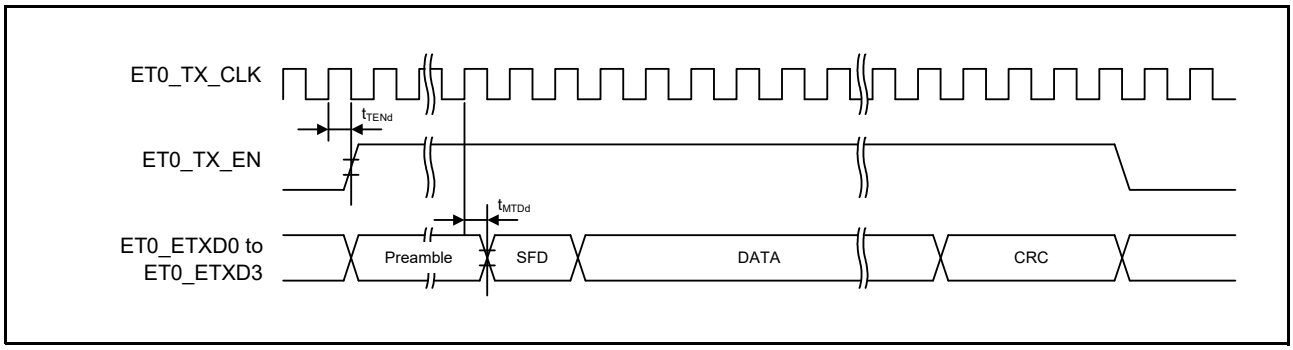


Figure 67.99 MII Transmission Timing (Normal Operation)

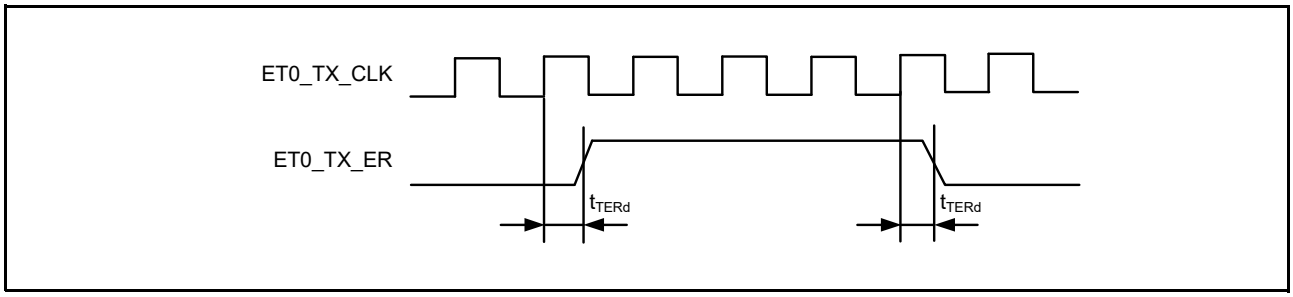


Figure 67.100 ET0_TX_ER Output Timing

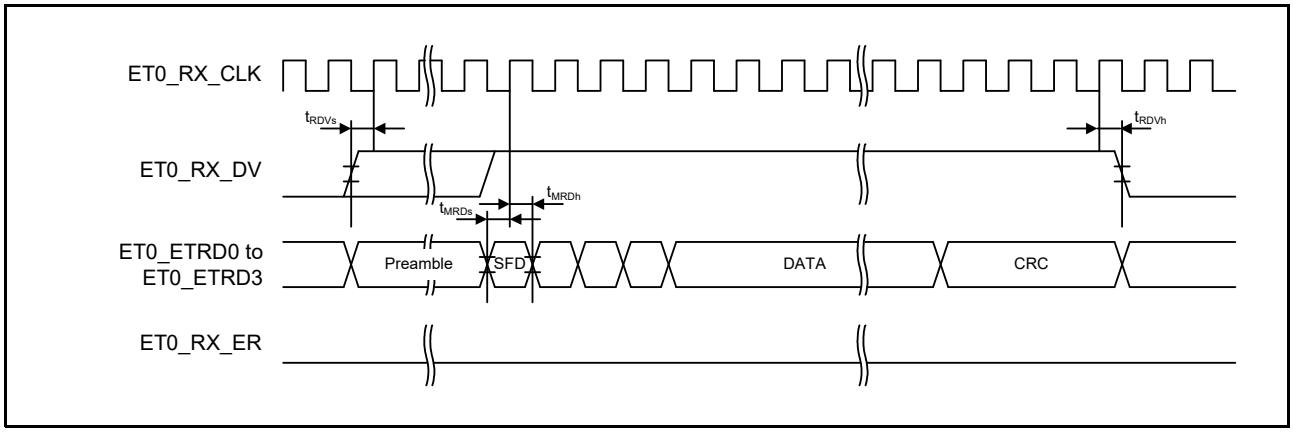


Figure 67.101 MII Reception Timing (Normal Operation)

Table 67.101 ESWM Timing (GMII)

Conditions: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the ET0_MDC and ET0_MDIO pins are set for DRV2.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than ET0_MDC and ET0_MDIO are set as follows.
When VCC = 2.3 to 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV7.
When VCC = 3.0 to 3.6 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV6.

Item		Symbol	Min.	Max.	Unit	Test Conditions
ESWM (GMII)	ET0_GTX_CLK frequency	t_{FREQ}	125 – 100ppm	125 + 100ppm	MHz	Figure 67.102, Figure 67.103
	ET0_GTX_CLK cycle	t_{PERIOD}	7.5	8.5	ns	
	ET0_RX_CLK cycle	t_{PERIOD}	7.5	—	ns	
	ET0_GTX_CLK, ET0_RX_CLK high pulse width	t_{HIGH}	2.5	—	ns	
	ET0_GTX_CLK, ET0_RX_CLK low pulse width	t_{LOW}	2.5	—	ns	
	ET0_GTX_CLK, ET0_RX_CLK rising time	t_R	—	1	ns	
	ET0_GTX_CLK, ET0_RX_CLK falling time	t_F	—	1	ns	
	Slew rate of ET0_GTX_CLK and ET0_RX_CLK (during rising)*1	—	0.6	—	V/ns	
	Slew rate of ET0_GTX_CLK and ET0_RX_CLK (during falling)*1	—	0.6	—	V/ns	
	Setup time of ET0_TXD, ET0_TX_EN, and ET0_TX_ER relative to ET0_GTX_CLK	t_{SETUP}	2.5	—	ns	
	Hold time of ET0_TXD, ET0_TX_EN, and ET0_TX_ER relative to ET0_GTX_CLK	t_{HOLD}	0.5	—	ns	
	Setup time of ET0_RXD, ET0_RX_DV, and ET0_RX_ER relative to ET0_RX_CLK	t_{SETUP}	2	—	ns	
	Hold time of ET0_RXD, ET0_RX_DV, and ET0_RX_ER relative to ET0_RX_CLK	t_{HOLD}	0	—	ns	

Note: The above characteristics are only guaranteed for products with part numbers of the form R5K574Nxx4yy (yy = BG, BD, AE).
Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the ESWM are from the same group.
Note 1. Clock slew rate is the instantaneous rate of change of the clock potential with respect to time (dV/dt), not an average value over the entire rise or fall time interval.

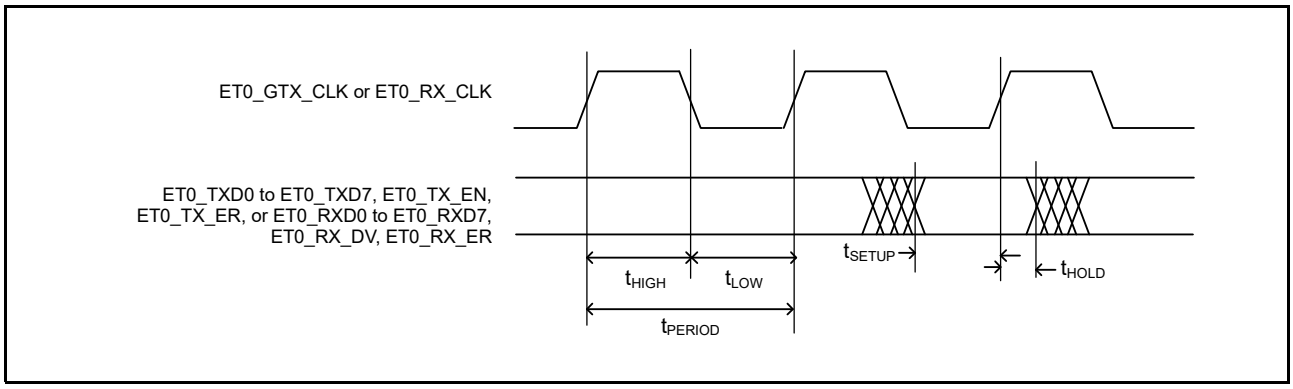


Figure 67.102 GMII Timing

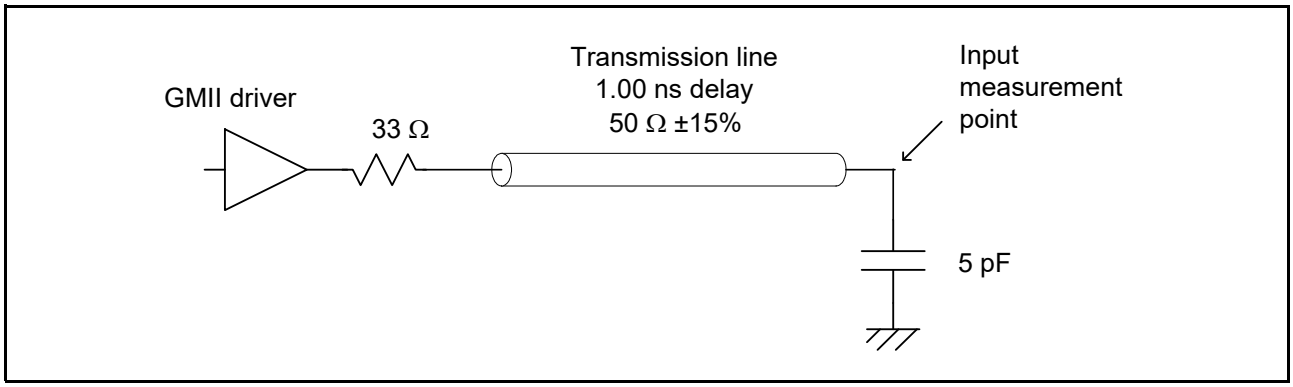


Figure 67.103 GMII Output Timing Measurement Conditions

Table 67.102 ESWM Timing (RGMII)

Conditions: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the ET0_MDC and ET0_MDIO pins are set for DRV2.
The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to pins other than ET0_MDC and ET0_MDIO are set as follows.
When VCC = 2.3 to 2.7 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV7.
When VCC = 3.0 to 3.6 V: The driving ability select bits (PmnPFS.DSCR[3:0]) are set for DRV6.

Item			Symbol	Min.	Max.	Unit	Test Conditions
ESWM (RGMII)	Output data skew relative to RGMII0_TXC*1		T_{skewT}	–500	500	ps	Figure 67.104, Figure 67.106
	Input data skew relative to RGMII0_RXC*1		T_{skewR}	1.0	2.6	ns	
	Output data setup time relative to RGMII0_TXC (integrated delay)		T_{setupT}	1.2	—	ns	Figure 67.105, Figure 67.106
	Output data hold time relative to RGMII0_TXC (integrated delay)		T_{holdT}	1.2	—	ns	
	Input data setup time relative to RGMII0_RXC (integrated delay)		T_{setupR}	1.0	—	ns	
	Input data hold time relative to RGMII0_RXC (integrated delay)		T_{holdR}	1.0	—	ns	
	Clock cycle time	1000 Mbps	—	7.2	8.8	ns	—
		100 Mbps	—	36	44	ns	
		10 Mbps	—	360	440	ns	
	Duty cycle for gigabit*2		—	45	55	ns	
	Duty cycle for 10/100T*2		—	40	60	%	
	Rise/fall time (20 to 80%)		—	—	0.75	%	

Note: The above characteristics are only guaranteed for products with part numbers of the form R5K574Nxx4yy (yy = BG, BD, AE).
Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the ESWM are from the same group.
Note 1. This indicates that the clock signals require routing in PC board design so that the trace delay adds a value longer than 1.5 ns and shorter than 2.0 ns to the corresponding clock signal.
Note 2. The duty cycle can be stretched or reduced during speed changes or transitions to the clock domain for received packets as long as the minimum duty cycle is ensured and the length of the stretching does not exceed three cycles of Tcyc, the minimum transition speed.

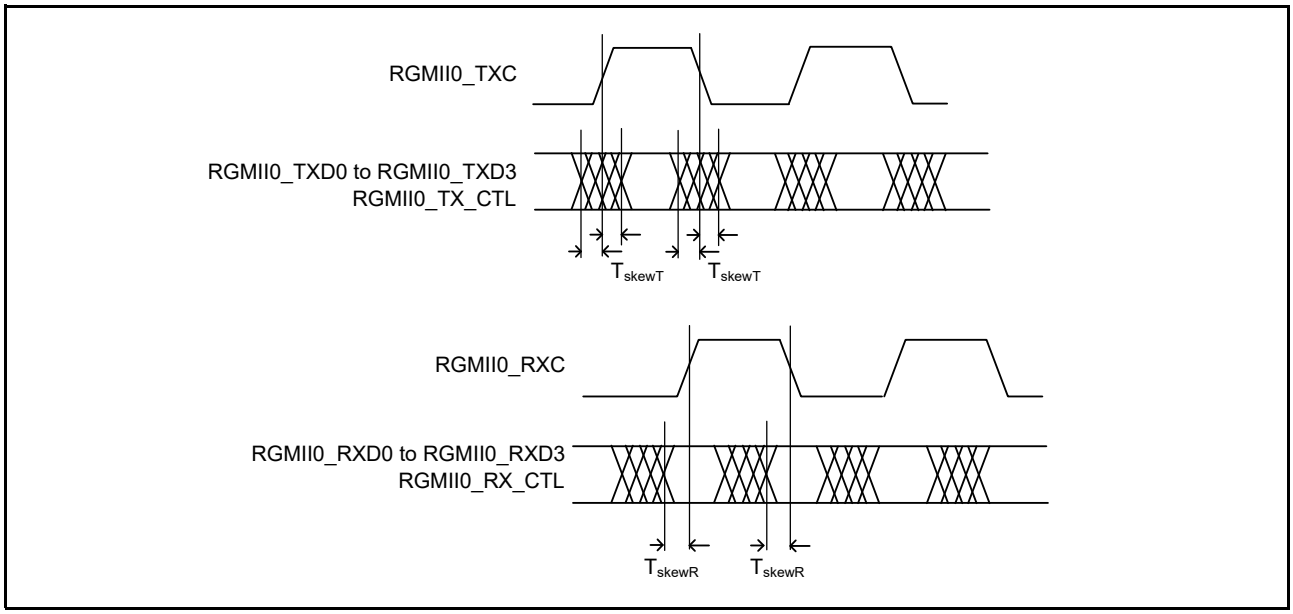


Figure 67.104 RGMII Timing

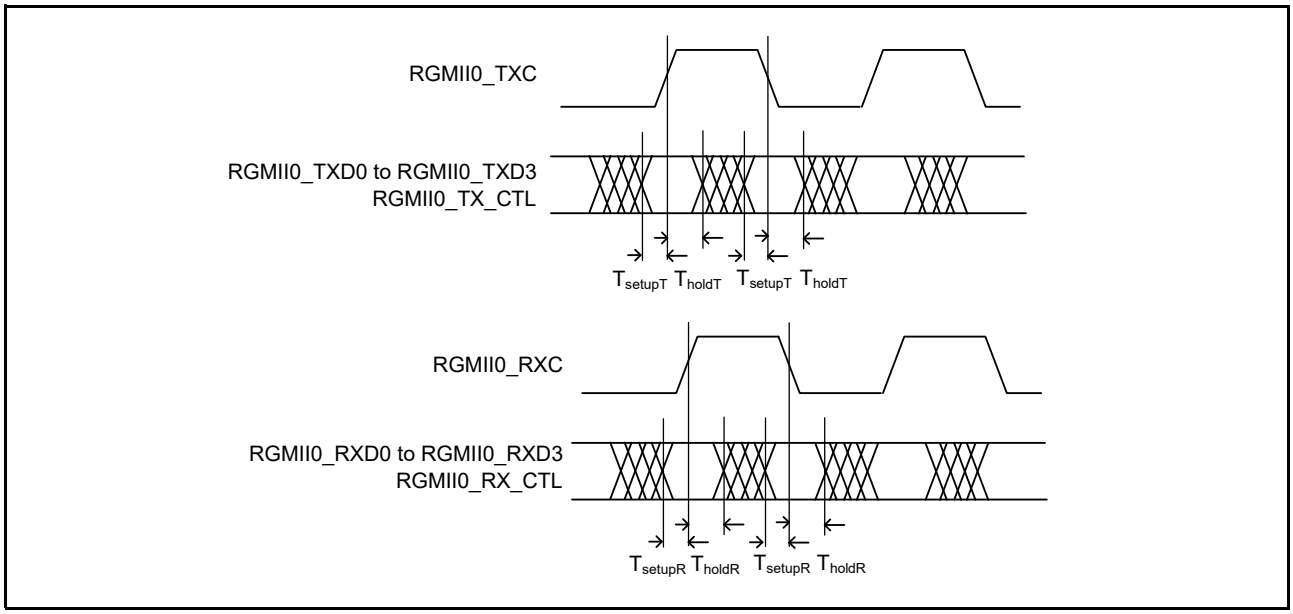


Figure 67.105 RGMII Timing (RGMII-ID)

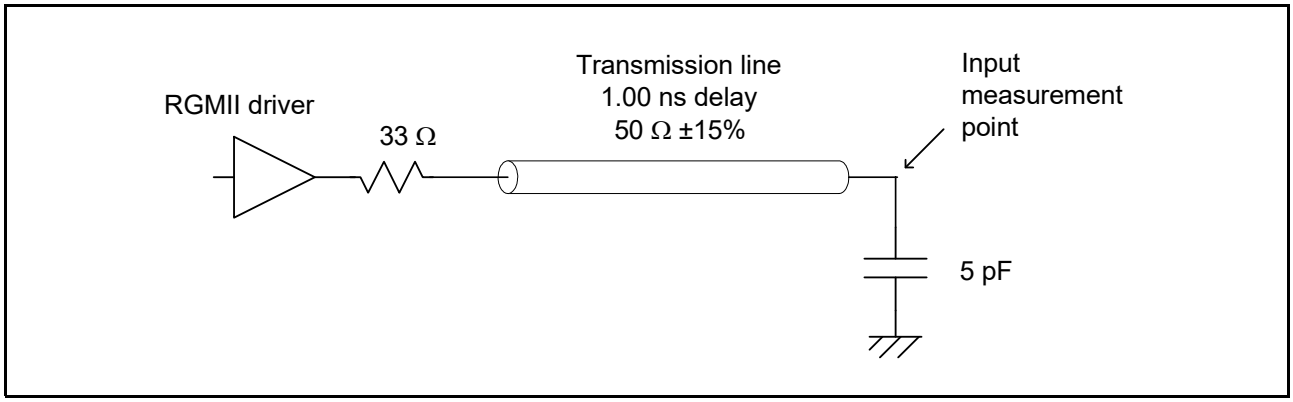


Figure 67.106 RGMII Output Timing Measurement Conditions

Table 67.103 ESWM Timing (ET0_MDIO, ET0_MDC)

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		VCC	Symbol	Min.	Max.	Unit	Test Conditions
ESWM (ET0_MDIO, ET0_MDC)	ETH0_MDC output cycle time	2.7 V or higher	t_{MDC}	80	—	ns	Figure 67.107
		2.3 V or higher		160	—		
	ETH0_MDIO setup time	2.7 V or higher	t_{SMDIO}	20	—	ns	
		2.3 V or higher		40	—		
	ETH0_MDIO hold time	2.7 V or higher	t_{HMDIO}	0	—	ns	
		2.3 V or higher		0	—		
	ETH0_MDIO output delay time	2.7 V or higher	t_{DMDIO}	0	20	ns	
		2.3 V or higher		0	40		

Note: The pin names include either -A or -B as a suffix to indicate the group to which the given pin belongs. Make sure that pins to be used for the ESWM are from the same group.

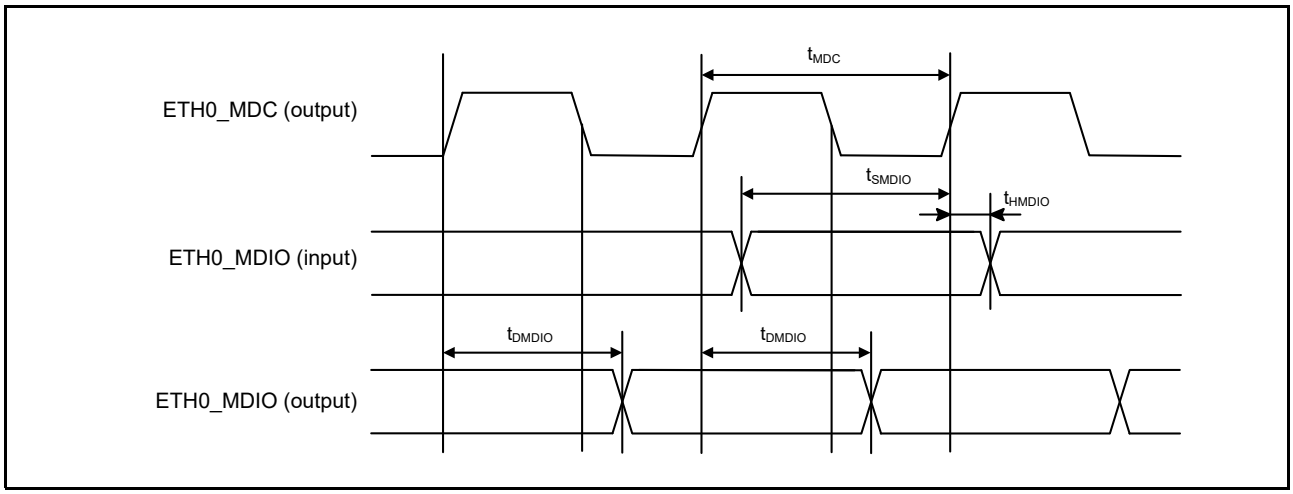


Figure 67.107 ET0_MDIO, ET0_MDC Timing

67.5.7.17 PDMIF

Table 67.104 PDMIF Timing

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item		VCC/VCC2	Symbol	Min.	Max.	Unit	Test Conditions
PDMIF	Clock cycle	2.70 V or higher	t_{Psync}	250	4000	ns	Figure 67.108
		1.62 V or higher		500	4000		
	Clock high pulse width		t_{PDCKWH}	t_{Psync} × 0.45	t_{Psync} × 0.55	ns	
	Clock low pulse width		t_{PDCKWL}	t_{Psync} × 0.45	t_{Psync} × 0.55	ns	
	Setup time	2.70 V or higher	t_{SU}	15	—	ns	Figure 67.109, Figure 67.110
		1.62 V or higher		30	—		
	Hold time		t_H	0	—	ns	

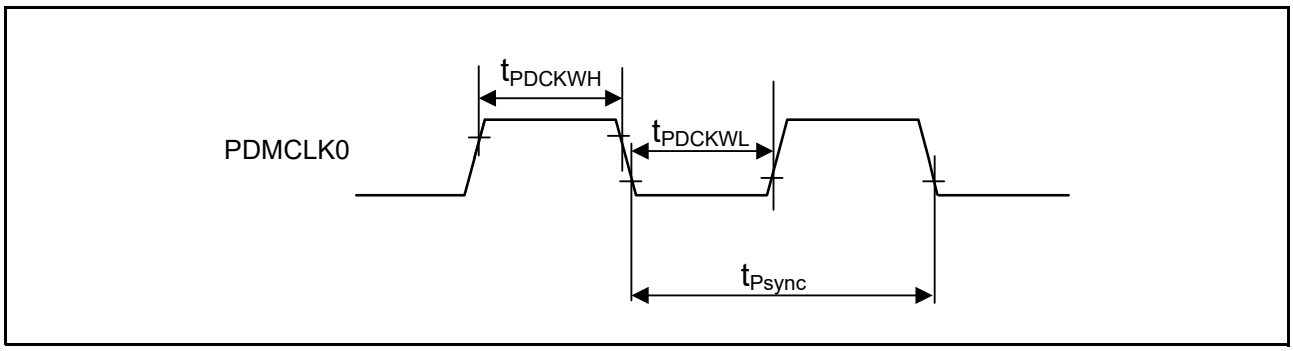


Figure 67.108 Timing of Clock Output (PDMCLK0)

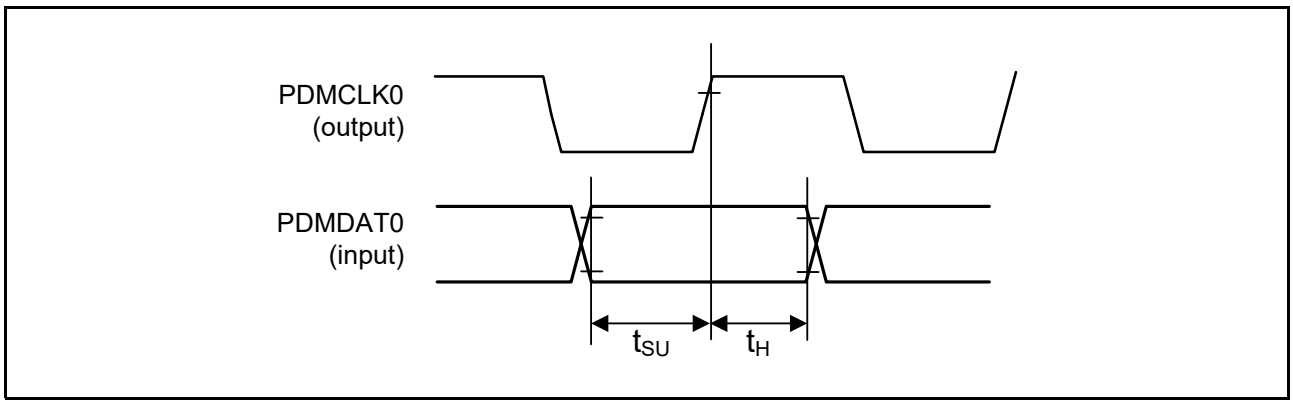


Figure 67.109 Receive Timing (Synchronized with the Rise of PDMCLK0)

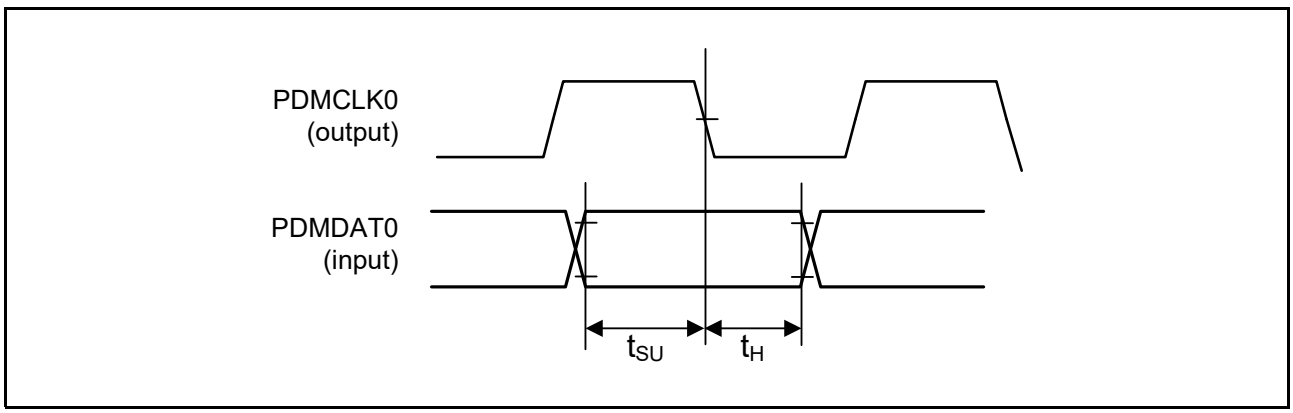


Figure 67.110 Receive Timing (Synchronized with the Fall of PDMCLK0)

67.5.7.18 PCIF

Table 67.105 PCIF Timing

Condition: The driving ability select bits (PmnPFS.DSCR[3:0]) corresponding to the target pins are set for DRV2.

Item				Symbol	Min.*1, *2	Max.	Unit	Test Conditions
Incremental mode (input)	Mode 0 (AB-phase pulses)	Toggle interval for the A- and B-phase inputs		$t_{PCINMD0AB}$	• 2.7 ≤ VCC ≤ 3.6 V $t_{CCLK_DIV} + t_{CCLK} + 5$ • VCC < 2.7 V $t_{CCLK_DIV} + t_{CCLK} + 6$	—	ns	Figure 67.111
		Pulse width of the Z-phase input		$t_{PCINMD0Z}$		—		
	Mode 1 (A-phase pulse) (B-phase rotation direction)	Toggle interval for the A-phase input		$t_{PCINMD1AA}$		—		Figure 67.112
		Toggle interval for the A- and B-phase inputs when the direction of rotation is changed		$t_{PCINMD1CR}$		—		
		Pulse width of the Z-phase input		$t_{PCINMD1Z}$		—		
	Mode 2 (pulse switching)	Toggle intervals for each of the A- and B-phase inputs		$t_{PCINMD2AA}$, $t_{PCINMD2BB}$		—		Figure 67.113
		Pulse width of the Z-phase input		$t_{PCINMD2Z}$		—		
		Toggle intervals for the A- and B-phase inputs when the direction of rotation is changed	Period over which both the A- and B-phase inputs are high before the change	$t_{PCINMD2CRH}$		—		
			Period over which either the A- or B-phase input is low after the change	$t_{PCINMD2CRL}$	• 2.7 ≤ VCC ≤ 3.6 V $2 \times t_{CCLK_DIV} + t_{CCLK} + 5$ • VCC < 2.7 V $2 \times t_{CCLK_DIV} + t_{CCLK} + 6$	—		
Phase counting mode (output) Incremental mode (output)	Mode 0 (AB-phase pulses)	Toggle interval for the A- and B-phase outputs		$t_{PCOUTMD0AB}$	• 2.7 ≤ VCC ≤ 3.6 V $t_{CCLK_DIV} - 5.5$ • VCC < 2.7 V $t_{CCLK_DIV} - 12$	—	ns	Figure 67.114
		Pulse width of the Z-phase output		$t_{PCOUTMD0Z}$		—		
	Mode 1 (A-phase pulse) (B-phase rotation direction)	Toggle interval for the A-phase output		$t_{PCOUTMD1AA}$		—		Figure 67.115
		Toggle interval for the A- and B-phase outputs when the direction of rotation is changed		$t_{PCOUTMD1CR}$		—		
		Pulse width of the Z-phase output		$t_{PCOUTMD1Z}$		—		
	Mode 2 (pulse switching)	Toggle intervals for each of the A- and B-phase outputs		$t_{PCOUTMD2AA}$, $t_{PCOUTMD2BB}$		—		Figure 67.116
		Pulse width of the Z-phase output		$t_{PCOUTMD2Z}$		—		
		Toggle intervals for the A- and B-phase outputs when the direction of rotation is changed	Period over which both the A- and B-phase outputs are high before the change	$t_{PCOUTMD2CRH}$		—		
			Period over which either the A- or B-phase output is low after the change	$t_{PCOUTMD2CRL}$		—		

Note 1. t_{CCLK}: Period of the core clock for use in receiving an event trigger input
Note 2. t_{CCLK_DIV}: Period of the frequency-divided clock with which the ABZ-phase outputs are in synchronization

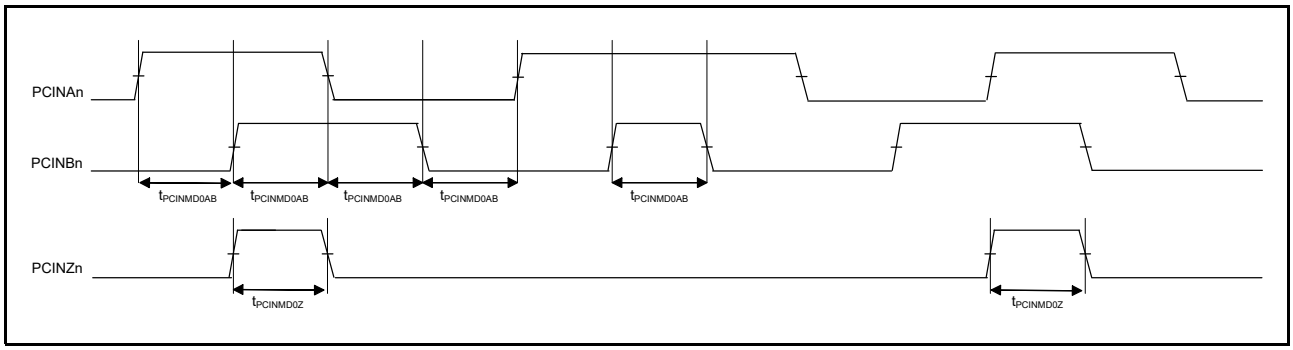


Figure 67.111 Timing of Operation with Mode 0 Selected in Incremental Mode (n = 0, 1)

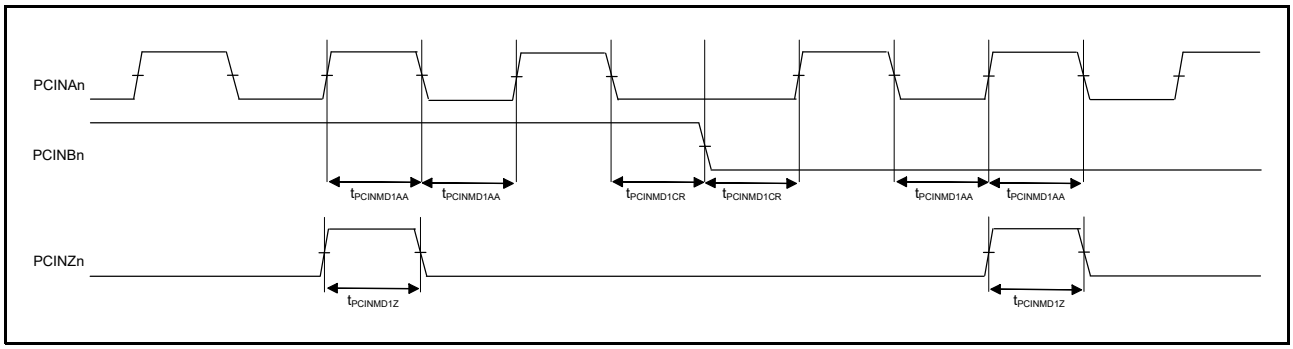


Figure 67.112 Timing of Operation with Mode 1 Selected in Incremental Mode (n = 0, 1)

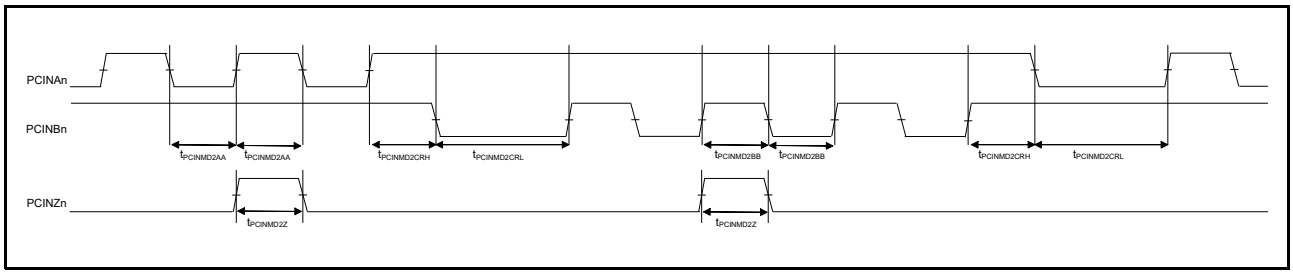


Figure 67.113 Timing of Operation with Mode 2 Selected in Incremental Mode (n = 0, 1)

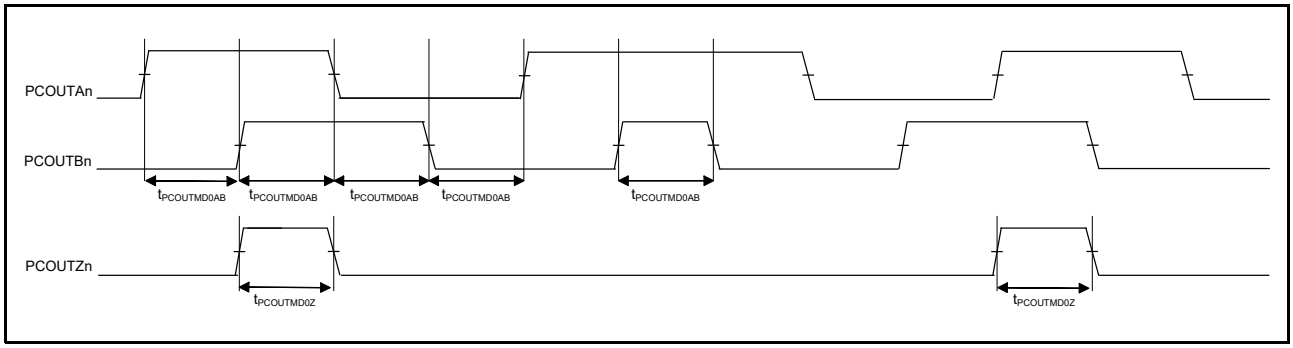


Figure 67.114 Timing of Output Operation in Mode 0 (n = 0, 1)

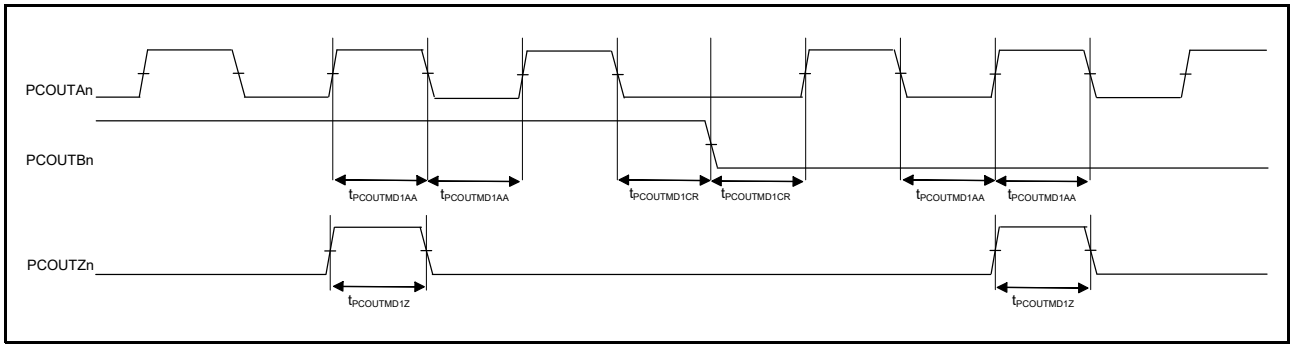


Figure 67.115 Timing of Output Operation in Mode 1 (n = 0, 1)

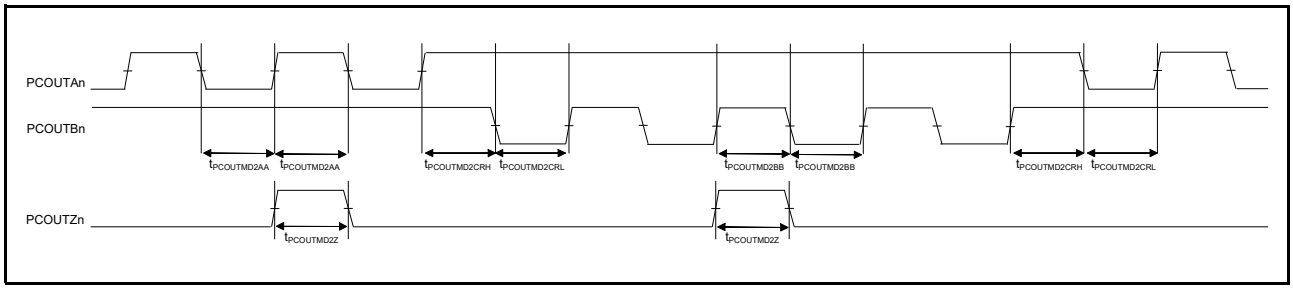


Figure 67.116 Timing of Output Operation in Mode 2 (n = 0, 1)

67.5.7.19 PIO

Table 67.106 PIO Timing

Item	Symbol	Min.	Max.	Unit*1	Test Conditions
PIO_n input delay	t_{PIOID}	—	6	t_{PBcyc}	Figure 67.117
PIO_n output delay	t_{PIOOD}	—	3		Figure 67.118

n = 000 to 063

Note 1. t_{PBcyc}: PCLKB cycle

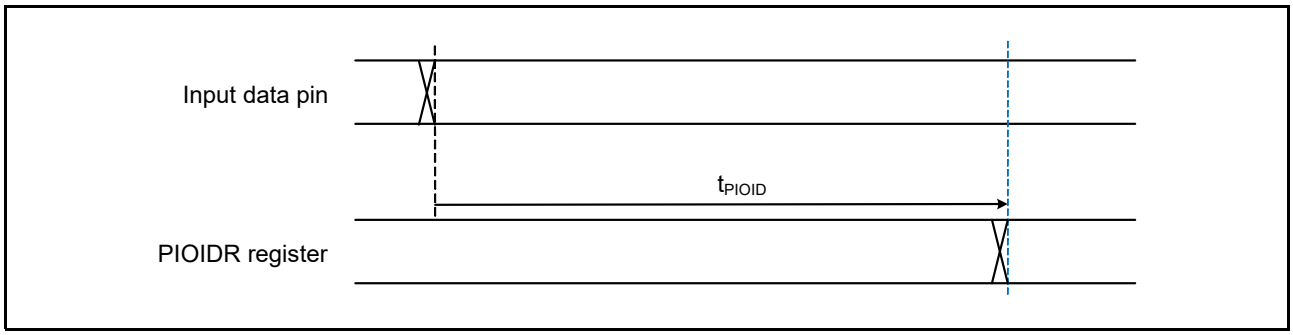


Figure 67.117 Input Delay

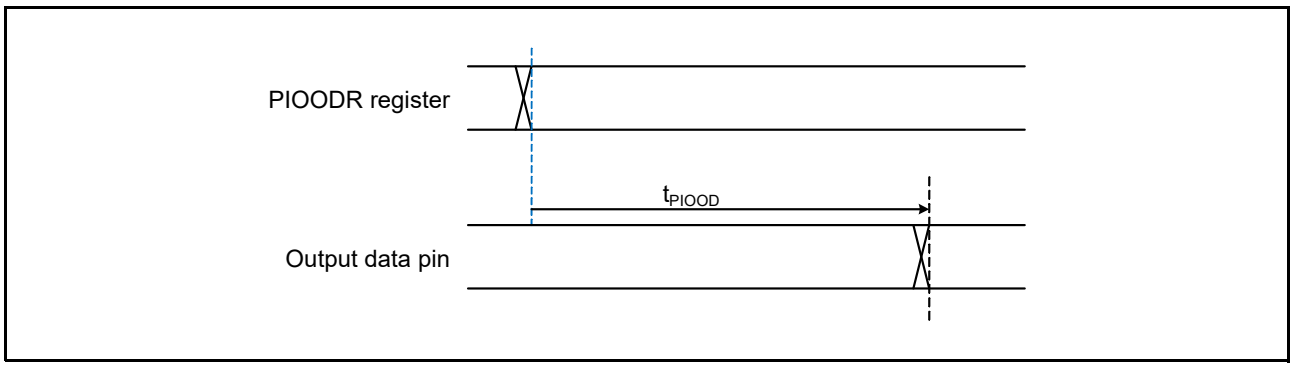


Figure 67.118 Output Delay

67.5.7.20 A/D Converter Trigger

Table 67.107 A/D Converter Trigger Timing

Item		VCC	Symbol	Min.	Max.	Unit*1	Test Conditions
A/D converter	A/D converter trigger input pulse width	2.7 V or higher	t_{TRGW}	1.5	—	t_{ADcyc}	Figure 67.119
		Less than 2.7 V		3.0	—		

Note 1. t_{ADcyc}: ADCLK cycle

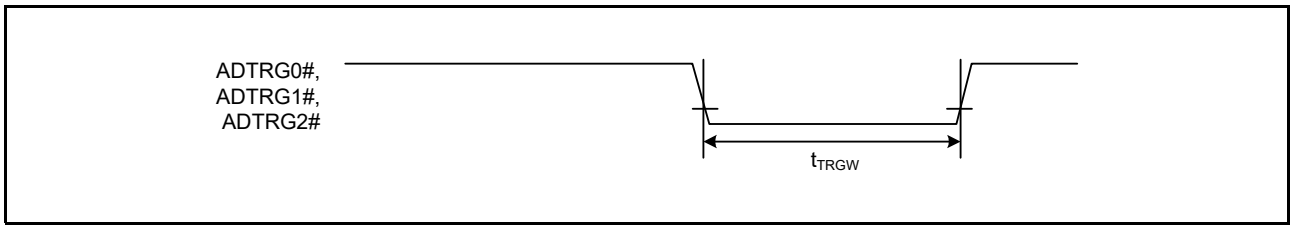


Figure 67.119 A/D Converter Trigger Input Timing

67.5.7.21 CAC

Table 67.108 CAC Timing

Item*1, *2			Symbol	Min.*1, *2	Max.	Unit	Test Conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	—	ns	
		$t_{PBcyc} > t_{cac}$		$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	—		

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{cac}: CAC count clock source cycle

67.6 USB Characteristics

67.6.1 USB Timing

Table 67.109 On-Chip USB Low Speed (Host Only) Characteristics (DP and DM Pin Characteristics)

Conditions: VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8		
	Differential input sensitivity	V_{DI}	0.2	—	—		| USB_DP – USB_DM |
	Differential common mode range	V_{CM}	0.8	—	2.5		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3		I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0		Figure 67.120
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300		
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

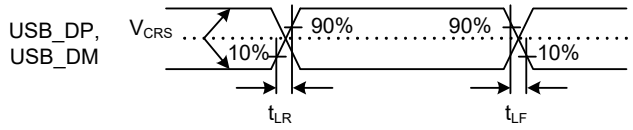


Figure 67.120 USB_DP and USB_DM Output Timing (Low Speed)

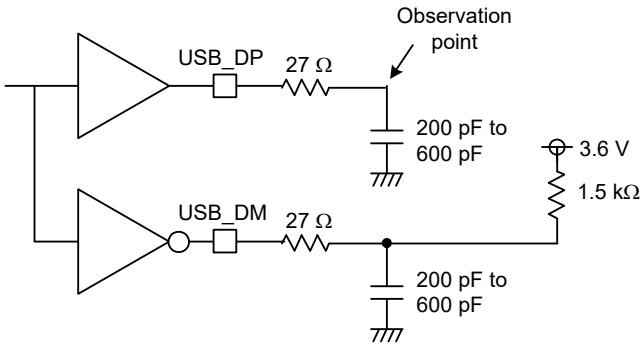


Figure 67.121 Test Circuit (Low Speed)

Table 67.110 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics)

Conditions: VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8		
	Differential input sensitivity	V_{DI}	0.2	—	—		| USB_DP – USB_DM |
	Differential common mode range	V_{CM}	0.8	—	2.5		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3		I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0		Figure 67.122
	Rise time	t_{FR}	4	—	20	ns	
	Fall time	t_{FF}	4	—	20		
	Rise/fall time ratio	t_{FR} / t_{FF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	Rs = 27 Ω included
Pull-up and pull-down characteristics	DP pull-up resistance (when the function controller function is selected)	R_{pu}	0.900	—	1.575	kΩ	Idle state
			1.425	—	3.090		At transmission and reception
	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80		

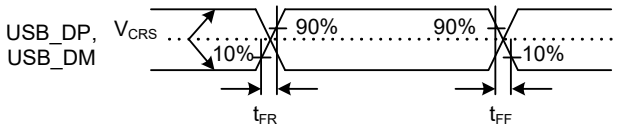


Figure 67.122 USB_DP and USB_DM Output Timing (Full-Speed)

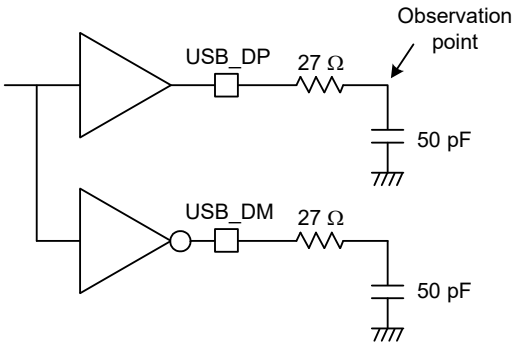


Figure 67.123 Test Circuit (Full-Speed)

67.6.2 USBHS Timing

Table 67.111 On-Chip USB Low-Speed (Host Only) Characteristics (DP and DM Pin Characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ±1%, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, USBMCLK = 12/20/24/48 MHz, USBCLK = 48 MHz, USB60CLK = 60 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8		
	Differential input sensitivity	V_{DI}	0.2	—	—		| USB_DP – USB_DM |
	Differential common mode range	V_{CM}	0.8	—	2.5		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low level voltage	V_{OL}	0.0	—	0.3		I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0		Figure 67.124
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300		
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

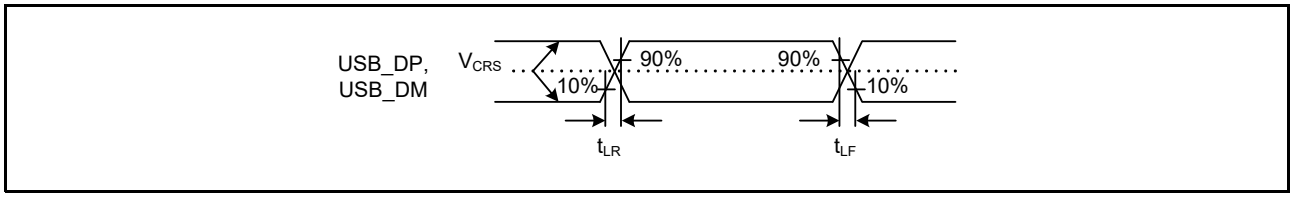


Figure 67.124 USB_DP and USB_DM Output Timing (Low Speed)

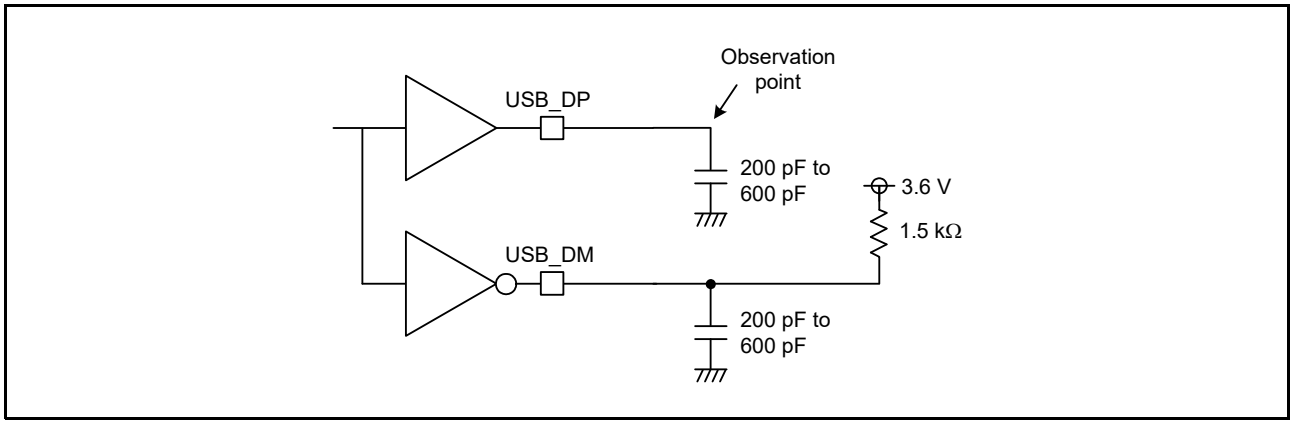


Figure 67.125 Test Circuit (Low Speed)

Table 67.112 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ±1%, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, USBMCLK = 12/20/24/48 MHz, USBCLK = 48 MHz, USB60CLK = 60 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8		
	Differential input sensitivity	V_{DI}	0.2	—	—		| USB_DP – USB_DM |
	Differential common mode range	V_{CM}	0.8	—	2.5		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low level voltage	V_{OL}	0.0	—	0.3		I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0		Figure 67.126
	Rise time	t_{FR}	4	—	20	ns	
	Fall time	t_{FF}	4	—	20		
	Rise/fall time ratio	t_{FR} / t_{FF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	Rs = 27 Ω included
			40.5	—	49.5		Rs not used (PHYSET.REPSEL[1:0] = 01b and PHYSET. HSEB = 0)
Pull-up and pull-down characteristics	DP pull-up resistance (when the function controller function is selected)	R_{pu}	0.900	—	1.575	kΩ	Idle state
			1.425	—	3.090		At transmission and reception
	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80		

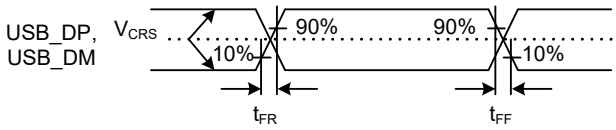


Figure 67.126 USB_DP and USB_DM Output Timing (Full-Speed)

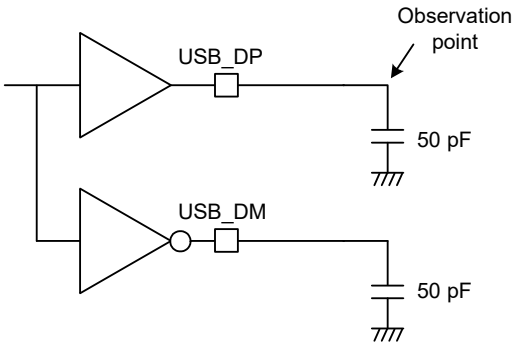


Figure 67.127 Test Circuit (Full-Speed)

Table 67.113 On-Chip USB High-Speed Characteristics (DP and DM Pin Characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ±1%, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, USBMCLK = 12/20/24/48 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Squelch detection threshold voltage (differential voltage)	V_{HSSQ}	100	—	150	mV	Figure 67.128
	Disconnect detect sensitivity	V_{HSDSC}	525	—	625		Figure 67.129
	Common mode voltage range	V_{HSCM}	–50	—	500		
Output characteristics	Idle state	V_{HSOI}	–10	—	10	mV	
	Output high level voltage	V_{HSOH}	360	—	440		
	Output low level voltage	V_{HSOL}	–10	—	10		
	Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100		
	Chirp K output voltage (difference)	V_{CHIRPK}	–900	—	–500		
AC characteristics	Rise time	t_{HSR}	500	—	—	ps	Figure 67.130
	Fall time	t_{HSF}	500	—	—		
	Output resistance	Z_{HSDRV}	40.5	—	49.5	Ω	

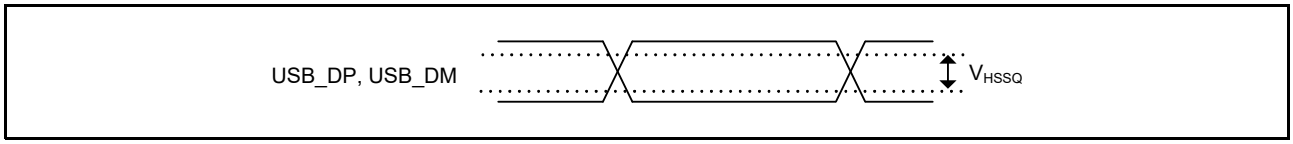


Figure 67.128 USB_DP and USB_DM Squelch Detect Sensitivity (High-Speed)

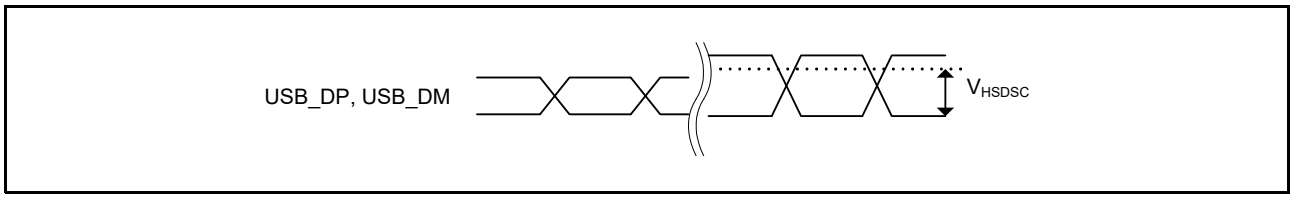


Figure 67.129 USB_DP and USB_DM Disconnect Detect Sensitivity (High-Speed)

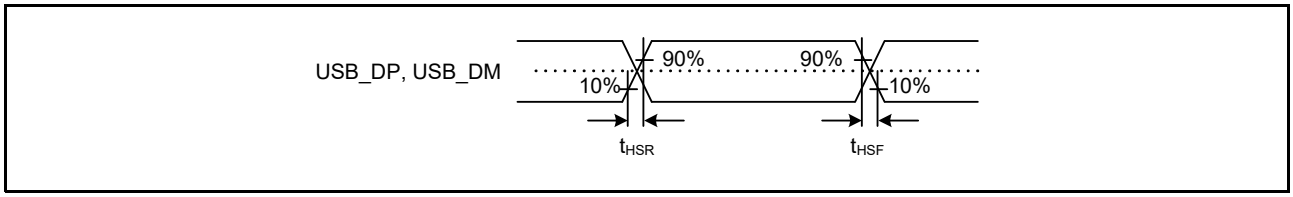


Figure 67.130 USB_DP and USB_DM Output Timing (High-Speed)

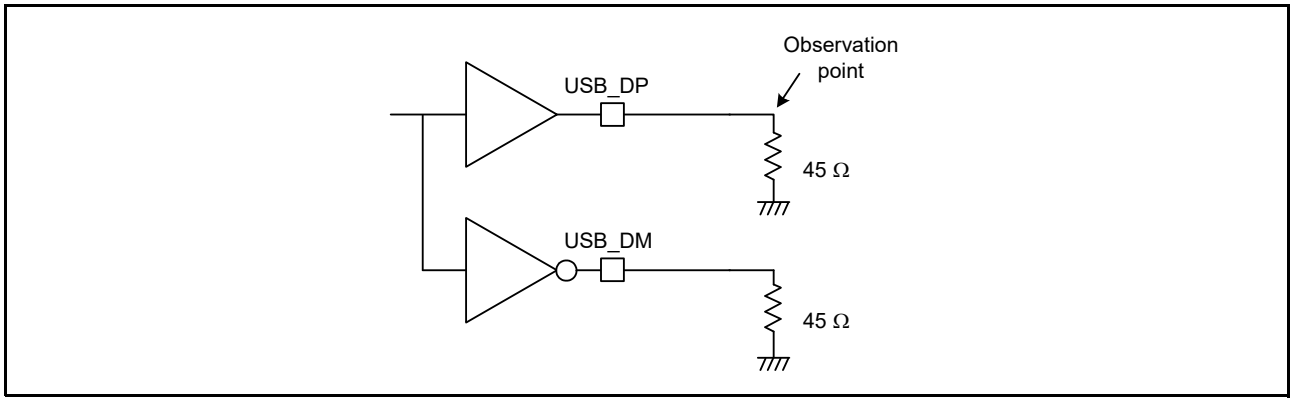


Figure 67.131 Test Circuit (High-Speed)

67.7 A/D Conversion Characteristics

67.7.1 Analog Input Voltage Range

Table 67.114 Analog Input Voltage Range (BGA320)

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input voltage	Unit 0	AN000 to AN007	VREFL00	—	VREFH00	V	—
	Unit 1	AN008 to AN015	VREFL01	—	VREFH01		—
		AN016 to AN023	VREFL01	—	VREFH01		VCC ≥ VREFH01
			VREFL01	—	VCC		VCC ≤ VREFH01
		AN024 to AN035	VREFL01	—	VREFH01		VCC2 ≥ VREFH01
			VREFL01	—	VCC2		VCC2 ≤ VREFH01
	Unit 2	AN008 to AN015	VREFL02	—	VREFH02		—
		AN016 to AN023	VREFL02	—	VREFH02		VCC ≥ VREFH02
			VREFL02	—	VCC		VCC ≤ VREFH02
		AN024 to AN035	VREFL02	—	VREFH02		VCC2 ≥ VREFH02
			VREFL02	—	VCC2		VCC2 ≤ VREFH02
Differential input voltage*1	Unit 0	AN000P to AN006P AN000N to AN006N	–VREFL00	—	+VREFH00	V	—
	Unit 1	AN008P to AN014P AN008N to AN014N	–VREFL01	—	+VREFH01		—
	Unit 2	AN008P to AN014P AN008N to AN014N	–VREFL02	—	+VREFH02		—

Note 1. The differential input voltage is ($A_{INP} – A_{INN}$).
- A_{INP} represents the input voltage of A_{Nx} and VREFL0m ≤ A_{INP} ≤ VREFH0m (m = 0, 1, 2) applies.
- A_{INN} represents the input voltage of A_{Ny} and VREFL0m ≤ A_{INN} ≤ VREFH0m (m = 0, 1, 2) applies.
  (x = 2i, y = 2i + 1, i = 0, 1, 2 ... (a desired integer))

Table 67.115 Analog Input Voltage Range (Except for Products in BGA320)

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input voltage	Unit 0	AN000 to AN007	VREFL0	—	VREFH0	V	—
	Unit 1	AN014, AN015	VREFL0	—	VREFH0		—
		AN016 to AN023	VREFL0	—	VREFH0		VCC ≥ VREFH0
			VREFL0	—	VCC		VCC ≤ VREFH0
		AN024 to AN035	VREFL0	—	VREFH0		VCC2 ≥ VREFH0
			VREFL0	—	VCC2		VCC2 ≤ VREFH0
	Unit 2	AN014, AN015	VREFL0	—	VREFH0		—
		AN016 to AN023	VREFL0	—	VREFH0		VCC ≥ VREFH0
			VREFL0	—	VCC		VCC ≤ VREFH0
		AN024 to AN035	VREFL0	—	VREFH0		VCC2 ≥ VREFH0
			VREFL0	—	VCC2		VCC2 ≤ VREFH0
Differential input voltage*1	Unit 0	AN000P to AN006P AN000N to AN006N	–VREFL0	—	+VREFH0	V	—
	Unit 1	AN008P to AN014P AN008N to AN014N	–VREFL0	—	+VREFH0		—
	Unit 2	AN008P to AN014P AN008N to AN014N	–VREFL0	—	+VREFH0		—

Note 1. The differential input voltage is ($A_{INP} – A_{INN}$).
- A_{INP} represents the input voltage of A_{Nx} and VREFL0 ≤ A_{INP} ≤ VREFH0 applies.
- A_{INN} represents the input voltage of A_{Ny} and VREFL0 ≤ A_{INN} ≤ VREFH0 applies.
  (x = 2i, y = 2i + 1, i = 0, 1, 2 ... (a desired integer))

67.7.2 Common Characteristics of the 16-Bit A/D Converter

Table 67.116 A/D Conversion Characteristics (Common)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0

Item			Min.*1	Typ.	Max.	Unit	Test Conditions*2
A/D conversion clock frequency (ADCLK)			25	50	60	MHz	
Successive approximation time			100	—	200	ns	
A/D sampling time (Normal accuracy)	Self-calibration*3		$1 \times t_{ADcyc} + 40$	—	—	ns	t_{Cmp} = 100 ns
	Self-diagnosis*3		$1 \times t_{ADcyc} + 40$	—	—		
	A/D conversion	AN000 to AN015*3 (S&H circuits not in use)	$1 \times t_{ADcyc} + 40$	—	—		
			40*5	—	—		
		AN000 to AN005*4 (S&H circuits in use)	$1 \times t_{ADcyc} + 160$	—	—		
		AN016 to AN035*3	400	—	—		
A/D sampling time (High-accuracy)	Self-calibration*4		$1 \times t_{ADcyc} + 140$*6	—	—	ns	t_{Cmp} = 100 ns
	Self-diagnosis*4		$1 \times t_{ADcyc} + 140$*6	—	—		
	A/D conversion*4	AN000 to AN015 (S&H circuits not in use)	$1 \times t_{ADcyc} + 140$*6	—	—		
		AN000 to AN005 (S&H circuits in use)	$1 \times t_{ADcyc} + 320$	—	—		
		AN016 to AN035	840	—	—		
Channel-dedicated sample-and-hold circuits	Sampling time	Self-calibration	400	—	—	ns	t_{Cmp} = 100 ns
		A/D conversion	400	—	—		
	Hold mode switching time		40	—	—		
	Hold time		—	—	5	µs	
Operation stabilization time	A/D start-up time		2	—	—	µs	t_{Cmp} = 100 ns
	Channel-dedicated sample-and-hold circuits start-up time		2	—	—		
	A/D shut-down time		1	—	—		

S&H: Channel-dedicated sample-and-hold circuits

Note 1. t_{ADcyc}: ADCLK cycle

Note 2. t_{Cmp}: Successive approximation time

Note 3. SAR mode and oversampling mode are supported.

Note 4. SAR mode and hybrid mode are supported.

Note 5. This value is for conversion in same-channel continuous scan mode.

Note 6. If t_{Cmp} is longer than 100 ns, the A/D sampling time must be longer than the value obtained from the following formula.
$1 \times t_{ADcyc} + 1.6 \times t_{Cmp}$

Table 67.117 A/D Conversion Characteristics (Common)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0

Item			Min.*1	Typ.	Max.	Unit	Test Conditions*2
A/D conversion clock frequency (ADCLK)			25	50	60	MHz	
Successive approximation time			200	—	200	ns	
A/D sampling time (Normal accuracy)	Self-calibration*3		$1 \times t_{ADcyc} + 420$	—	—	ns	t_{Cmp} = 200 ns
	Self-diagnosis*3		$1 \times t_{ADcyc} + 420$	—	—		
	A/D conversion	AN000 to AN015	$1 \times t_{ADcyc} + 420$	—	—		
			440*5	—	—		
		AN016 to AN035	800	—	—		
A/D sampling time (High-accuracy)	Self-calibration*4		$1 \times t_{ADcyc} + 780$	—	—	ns	t_{Cmp} = 200 ns
	Self-diagnosis*4		$1 \times t_{ADcyc} + 780$	—	—		
	A/D conversion*4	AN000 to AN015	$1 \times t_{ADcyc} + 780$	—	—		
		AN016 to AN035	1680	—	—		
Operation stabilization time	A/D start-up time		2	—	—	µs	t_{Cmp} = 200 ns
	A/D shut-down time		1	—	—		

Note 1. t_{ADcyc}: ADCLK cycle
Note 2. t_{Cmp}: Successive approximation time
Note 3. SAR mode and oversampling mode are supported.
Note 4. SAR mode and hybrid mode are supported.
Note 5. This value is for conversion in same-channel continuous scan mode.

Table 67.118 A/D Internal Reference Voltage Characteristics

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	0.77	0.80	0.84	V	
Sampling time	4.15	—	—	µs	

Table 67.119 A/D Conversion Characteristics of D/A Output

Item	Min.	Typ.	Max.	Unit	Test Conditions
Sampling time	2	—	—	µs	

67.7.3 SAR Mode Characteristics

Table 67.120 A/D Conversion Characteristics (SAR mode) (1) (1/2)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Signal source impedance: 50 Ω or less, Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK cycles

Item			Min.	Typ.	Max.	Unit	Test Conditions
Resolution			—	—	12	Bit	—
Single-ended input (normal accuracy)	AN000 to AN015 (S&H circuits not in use)	Conversion time	0.16	—	—	µs	• Sampling time: 3 ADCLK cycles
		Offset error	—	±3.0	±6.5	LSB	—
		Full-scale error	—	±3.0	±6.5		
		Absolute accuracy	—	±4.0	±11		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.0		
	AN000 to AN005 (S&H circuits in use)	Conversion time	1.00	—	—	µs	• Sampling time: 8 ADCLK cycles • Sampling time of S&H circuits: 35 ADCLK cycles
		Offset error	—	±1.5	±6.75	LSB	—
		Full-scale error	—	±1.5	±6.75		
		Absolute accuracy	—	±5.0	±10.5		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.5	±4.5		
	AN016 to AN035	Conversion time	0.50	—	—	µs	• Sampling time: 20 ADCLK cycles
		Offset error	—	±1.5	±6.5	LSB	—
		Full-scale error	—	±1.5	±6.5		
		Absolute accuracy	—	±5.5	±11		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.0		
Single-ended input (high-accuracy)	AN000 to AN015 (S&H circuits not in use)	Conversion time	0.26	—	—	µs	• Sampling time: 8 ADCLK cycles
		Offset error	—	±1.5	±4.5	LSB	—
		Full-scale error	—	±1.5	±4.5		
		Absolute accuracy	—	±4.0	±7.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.0		
	AN000 to AN005 (S&H circuits in use)	Conversion time	1.72	—	—	µs	• Sampling time: 16 ADCLK cycles • Sampling time of S&H circuits: 63 ADCLK cycles
		Offset error	—	±1.5	±6.75	LSB	—
		Full-scale error	—	±1.5	±6.75		
		Absolute accuracy	—	±4.5	±9.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.5	±4.5		

Table 67.120 A/D Conversion Characteristics (SAR mode) (1) (2/2)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Signal source impedance: 50 Ω or less, Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK cycles

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input (high-accuracy)	AN016 to AN035	Conversion time	0.94	—	—	μs	• Sampling time: 42 ADCLK cycles
		Offset error	—	±1.5	±4.5	LSB	—
		Full-scale error	—	±1.5	±4.5		
		Absolute accuracy	—	±5.5	±8.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.0		
Differential input (normal accuracy)	AN000P to AN014P, AN000N to AN014N (S&H circuits not in use)	Conversion time	0.16	—	—	μs	• Sampling time: 3 ADCLK cycles
		Offset error	—	±2.0	±3.5	LSB	—
		Full-scale error	—	±2.0	±3.5		
		Absolute accuracy	—	±3.0	±6.0		
		DNL differential nonlinearity error	—	±0.75	±1.0		
		INL integral nonlinearity error	—	±1.5	±3.5		
	AN000P to AN004P, AN000N to AN004N (S&H circuits in use)	Conversion time	1.00	—	—	μs	• Sampling time: 8 ADCLK cycles • Sampling time of S&H circuits: 35 ADCLK cycles
		Offset error	—	±1.5	±6.75	LSB	—
		Full-scale error	—	±1.5	±6.75		
		Absolute accuracy	—	±3.5	±10.5		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.5	±4.5		
Differential input (high-accuracy)	AN000P to AN014P, AN000N to AN014N (S&H circuits not in use)	Conversion time	0.26	—	—	μs	• Sampling time: 8 ADCLK cycles
		Offset error	—	±1.0	±2.5	LSB	—
		Full-scale error	—	±1.0	±2.5		
		Absolute accuracy	—	±2.0	±4.0		
		DNL differential nonlinearity error	—	±0.75	±1.0		
		INL integral nonlinearity error	—	±1.5	±3.5		
	AN000P to AN004P, AN000N to AN004N (S&H circuits in use)	Conversion time	1.72	—	—	μs	• Sampling time: 16 ADCLK cycles • Sampling time of S&H circuits: 63 ADCLK cycles
		Offset error	—	±1.5	±6.75	LSB	—
		Full-scale error	—	±1.5	±6.75		
		Absolute accuracy	—	±3.5	±9.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.5	±4.5		

S&H: Channel-dedicated sample-and-hold circuits

Note: Since the DNL values were measured by using the histogram method, the lower limit is –1.

Table 67.121 A/D Conversion Characteristics (SAR mode) (2) (1/2)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 10 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Resolution			—	—	12	Bit	—
Single-ended input (normal accuracy)	AN000 to AN015	Conversion time	0.64	—	—	µs	• Sampling time: 22 ADCLK cycles
		Offset error	—	±3.0	±6.5	LSB	—
		Full-scale error	—	±3.0	±6.5		
		Absolute accuracy	—	±5.5	±11		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.5		
	AN016 to AN035	Conversion time	1.00	—	—	µs	• Sampling time: 40 ADCLK cycles
		Offset error	—	±1.5	±6.5	LSB	—
		Full-scale error	—	±1.5	±6.5		
		Absolute accuracy	—	±5.5	±11		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.5		
Single-ended input (high-accuracy)	AN000 to AN015	Conversion time	1.00	—	—	µs	• Sampling time: 40 ADCLK cycles
		Offset error	—	±1.5	±4.5	LSB	—
		Full-scale error	—	±1.5	±4.5		
		Absolute accuracy	—	±5.0	±8.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.5		
	AN016 to AN035	Conversion time	1.88	—	—	µs	• Sampling time: 84 ADCLK cycles
		Offset error	—	±1.5	±4.5	LSB	—
		Full-scale error	—	±1.5	±4.5		
		Absolute accuracy	—	±5.5	±8.0		
		DNL differential nonlinearity error	—	±1.0	–1.0 to +1.5		
		INL integral nonlinearity error	—	±2.0	±4.5		
Differential input (normal accuracy)	AN000P to AN014P, AN000N to AN014N	Conversion time	0.64	—	—	µs	• Sampling time: 22 ADCLK cycles
		Offset error	—	±2.0	±3.5	LSB	—
		Full-scale error	—	±2.0	±3.5		
		Absolute accuracy	—	±4.5	±6.0		
		DNL differential nonlinearity error	—	±0.75	±1.0		
		INL integral nonlinearity error	—	±1.5	±4.0		

Table 67.121 A/D Conversion Characteristics (SAR mode) (2) (2/2)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 10 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Differential input (high-accuracy)	AN000P to AN014P, AN000N to AN014N	Conversion time	1.00	—	—	µs	• Sampling time: 40 ADCLK cycles
		Offset error	—	±1.0	±2.5	LSB	—
		Full-scale error	—	±1.0	±2.5		
		Absolute accuracy	—	±3.5	±4.5		
		DNL differential nonlinearity error	—	±0.75	±1.0		
		INL integral nonlinearity error	—	±1.5	±4.0		

Note: Since the DNL values were measured by using the histogram method, the lower limit is –1.

67.7.4 Common Characteristics of Oversampling Mode and Hybrid Mode

Table 67.122 A/D Conversion Characteristics (Common Characteristics of Oversampling Mode and Hybrid Mode) (1)

Conditions: ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Disconnection detection assistance is not in use, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit*1	Test Conditions
Resolution			—	—	16	Bit	
Oversampling period	Oversampling mode		0.16	—	—	µs	• Sampling time: 3 ADCLK cycles
	Hybrid mode*2		0.18	—	—		• Sampling time: 8 ADCLK cycles
Digital filter characteristics*3	Sinc filter	Initial delay	—	22	—	/Fos	
		Group delay	—	11	—		
		Normalized cutoff frequency	—	0.033	—	Fin/Fos	

Note 1. Fos is oversampling frequency.
Fin is input frequency.
In the hybrid mode, Fos is obtained by 1/ (the sum of the oversampling cycles in each analog channel assigned to the scan group).
Note 2. The listed value applies per channel.
Note 3. See Figure 67.132, Digital Filter Characteristics (Sinc Filter).

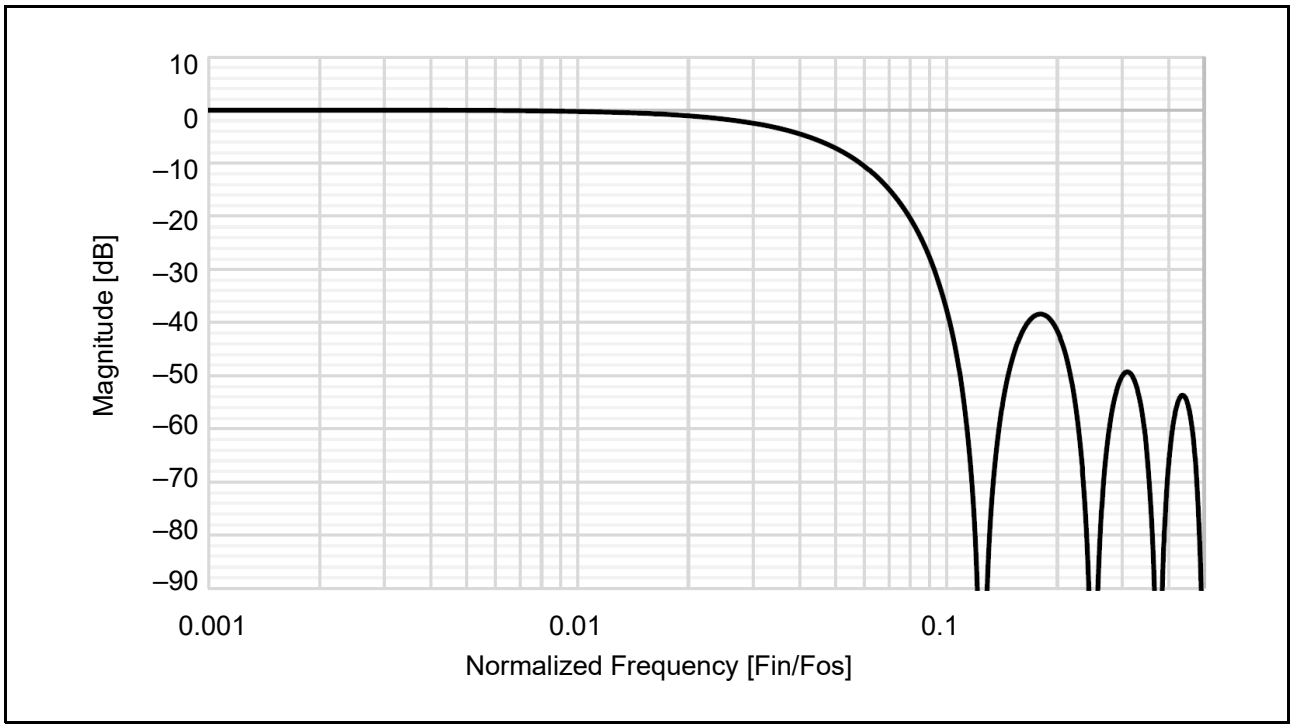


Figure 67.132 Digital Filter Characteristics (Sinc Filter)

Table 67.123 A/D Conversion Characteristics (Common Characteristics of Oversampling Mode and Hybrid Mode) (2)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Signal source impedance: 50 Ω or less,
Input frequency: 5 kHz

Item			Min.	Typ.	Max.	Unit	Test Conditions
Sinc filter	Single-ended input	SNDR: Signal-to-noise distortion ratio	—	80	—	dB	—
		ENOB: Effective number of bits	—	13	—	Bit	
	Differential input	SNDR: Signal-to-noise distortion ratio	—	86	—	dB	
		ENOB: Effective number of bits	—	14	—	Bit	

Note: The sampling time depends on the mode and input channel number.
- Oversampling mode (AN000 to AN015): 3 ADCLK cycles
- Hybrid mode (AN000 to AN015): 8 ADCLK cycles

Table 67.124 A/D Conversion Characteristics (Common Characteristics of Oversampling Mode and Hybrid Mode) (3)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 10 ADCLK cycles, Signal source impedance: 50 Ω or less,
Input frequency: 5 kHz

Item			Min.	Typ.	Max.	Unit	Test Conditions
Sinc filter	Single-ended input	SNDR: Signal-to-noise distortion ratio	—	74	—	dB	—
		ENOB: Effective number of bits	—	12	—	Bit	
	Differential input	SNDR: Signal-to-noise distortion ratio	—	80	—	dB	
		ENOB: Effective number of bits	—	13	—	Bit	

Note: The sampling time depends on the mode and input channel number.
- Oversampling mode (AN000 to AN015): 3 ADCLK cycles
- Hybrid mode (AN000 to AN015): 8 ADCLK cycles

67.7.5 Oversampling Mode Characteristics

Table 67.125 A/D Conversion Characteristics (Oversampling Mode) (1)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input	AN000 to AN015*1	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		Single or continuous scan mode
			—	±1.0	±5.0		One-channel continuous scan mode
		DNL differential nonlinearity error*4, *5	—	–1.0 to +1.5	–1.0 to +2.5		—
		INL integral nonlinearity error*4	—	±4.0	±8.0		
	AN016 to AN035*2	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +4.0		
		INL integral nonlinearity error*4	—	±4.0	±12		
Differential input	AN000 to AN015*1	Offset error*3	—	±0.25	±2.0	LSB	—
		Gain error*3	—	±0.5	±2.0		Single or continuous scan mode
			—	±0.5	±2.5		One-channel continuous scan mode
		DNL differential nonlinearity error*4, *5	—	–1.0 to +1.5	–1.0 to +2.0		—
		INL integral nonlinearity error*4	—	±3.0	±6.0		

Note: These specifications apply when only one 16-bit A/D converter is operating, the 12-bit D/A converter is not operating, and there is no access to the external bus during A/D conversion.
If another 16-bit A/D converter or 12-bit D/A converter is also in operation or if bus access occurs during A/D conversion, the actual values may not fall within the stated ranges.
When using a 16-bit A/D converter, do not use any pin of port 4 as a digital output.
The above characteristics were observed when AVCC0, AVSS0, VREFH0, VREFH00, VREFH01, VREFH02, VREFL0, VREFL00, VREFL01, VREFL02, and the voltages input to the 16-bit A/D converter had become stable.

Note 1. Sampling time: 3 ADCLK cycles
Note 2. Sampling time: 20 ADCLK cycles
Note 3. This value is for conversion with 12-bit resolution.
Note 4. Test condition: 0.2% to 99.8% of the analog input voltage range
Note 5. Since the DNL values were measured by using the histogram method, the lower limit is –1.

Table 67.126 A/D Conversion Characteristics (Oversampling Mode) (2)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 10 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input	AN000 to AN015*1	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		Single or continuous scan mode
			—	±1.0	±5.0		One-channel continuous scan mode
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +2.5		—
		INL integral nonlinearity error*4	—	±4.0	±8.0		
	AN016 to AN035*2	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +4.0		
		INL integral nonlinearity error*4	—	±4.0	±12		
Differential input	AN000 to AN015*1	Offset error*3	—	±0.25	±2.0	LSB	—
		Gain error*3	—	±0.5	±2.0		Single or continuous scan mode
			—	±0.5	±2.5		One-channel continuous scan mode
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +2.0		—
		INL integral nonlinearity error*4	—	±3.0	±6.0		

Note: These specifications apply when only one 16-bit A/D converter is operating, the 12-bit D/A converter is not operating, and there is no access to the external bus during A/D conversion.
If another 16-bit A/D converter or 12-bit D/A converter is also in operation or if bus access occurs during A/D conversion, the actual values may not fall within the stated ranges.
When using a 16-bit A/D converter, do not use any pin of port 4 as a digital output.
The above characteristics were observed when AVCC0, AVSS0, VREFH0, VREFH00, VREFH01, VREFH02, VREFL0, VREFL00, VREFL01, VREFL02, and the voltages input to the 16-bit A/D converter had become stable.

Note 1. Sampling time: 22 ADCLK cycles
Note 2. Sampling time: 40 ADCLK cycles
Note 3. This value is for conversion with 12-bit resolution.
Note 4. Test condition: 0.2% to 99.8% of the analog input voltage range
Note 5. Since the DNL values were measured by using the histogram method, the lower limit is –1.

67.7.6 Hybrid Mode Characteristics

Table 67.127 A/D Conversion Characteristics (Hybrid Mode) (1)

Conditions: AVCC0 = VCC = VCC2 = 2.70 V to 3.63 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 2.70 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 5 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input	AN000 to AN015*1 (S&H circuits not in use)	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +1.5	–1.0 to +2.5		
		INL integral nonlinearity error*4	—	±4.0	±8.0		
	AN000 to AN005*1 (S&H circuits in use)	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±4.0		
		DNL differential nonlinearity error*4, *5	—	±1.0	–1.0 to +2.0		
		INL integral nonlinearity error*4	—	±12	±16		
	AN016 to AN035*2	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±5.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +4.0		
		INL integral nonlinearity error*4	—	±4.0	±12		
Differential input	AN000 to AN015*1 (S&H circuits not in use)	Offset error*3	—	±0.25	±2.0	LSB	—
		Gain error*3	—	±0.5	±2.5		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +1.5	–1.0 to +2.0		
		INL integral nonlinearity error*4	—	±3.0	±6.0		
	AN000 to AN005*2 (S&H circuits in use)	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±0.5	±4.0		
		DNL differential nonlinearity error*4, *5	—	±1.0	–1.0 to +2.0		
		INL integral nonlinearity error*4	—	±4.0	±16		

S&H: Channel-dedicated sample-and-hold circuits

Note: These specifications apply when only one 16-bit A/D converter is operating, the 12-bit D/A converter is not operating, and there is no access to the external bus during A/D conversion.
If another 16-bit A/D converter or 12-bit D/A converter is also in operation or if bus access occurs during A/D conversion, the actual values may not fall within the stated ranges.
When using a 16-bit A/D converter, do not use any pin of port 4 as a digital output.
The above characteristics were observed when AVCC0, AVSS0, VREFH0, VREFH00, VREFH01, VREFH02, VREFL0, VREFL00, VREFL01, VREFL02, and the voltages input to the 16-bit A/D converter had become stable.

Note 1. Sampling time: 8 ADCLK cycles
Note 2. Sampling time: 42 ADCLK cycles
Note 3. This value is for conversion with 12-bit resolution.
Note 4. Test condition: 0.2% to 99.8% of the analog input voltage range
Note 5. Since the DNL values were measured by using the histogram method, the lower limit is –1.

Table 67.128 A/D Conversion Characteristics (Hybrid Mode) (2)

Conditions: AVCC0 = VCC = VCC2 = 1.62 V to 2.70 V, VREFH0 = VREFH00 = VREFH01 = VREFH02 = 1.62 V to AVCC0
ADCLK: 50 MHz, Successive approximation time: 10 ADCLK cycles, Signal source impedance: 50 Ω or less

Item			Min.	Typ.	Max.	Unit	Test Conditions
Single-ended input	AN000 to AN015*1 (S&H circuits not in use)	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±5.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +2.5		
		INL integral nonlinearity error*4	—	±4.0	±8.0		
	AN016 to AN035*2	Offset error*3	—	±0.5	±4.0	LSB	—
		Gain error*3	—	±1.0	±5.0		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +4.0		
		INL integral nonlinearity error*4	—	±4.0	±12		
Differential input	AN000 to AN015*1 (S&H circuits not in use)	Offset error*3	—	±0.25	±2.0	LSB	—
		Gain error*3	—	±0.5	±2.5		
		DNL differential nonlinearity error*4, *5	—	–1.0 to +2.0	–1.0 to +2.5		
		INL integral nonlinearity error*4	—	±3.0	±6.0		

S&H: Channel-dedicated sample-and-hold circuits

Note: These specifications apply when only one 16-bit A/D converter is operating, the 12-bit D/A converter is not operating, and there is no access to the external bus during A/D conversion.
If another 16-bit A/D converter or 12-bit D/A converter is also in operation or if bus access occurs during A/D conversion, the actual values may not fall within the stated ranges.
When using a 16-bit A/D converter, do not use any pin of port 4 as a digital output.
The above characteristics were observed when AVCC0, AVSS0, VREFH0, VREFH00, VREFH01, VREFH02, VREFL0, VREFL00, VREFL01, VREFL02, and the voltages input to the 16-bit A/D converter had become stable.

Note 1. Sampling time: 40 ADCLK cycles
Note 2. Sampling time: 84 ADCLK cycles
Note 3. This value is for conversion with 12-bit resolution.
Note 4. Test condition: 0.2% to 99.8% of the analog input voltage range
Note 5. Since the DNL values were measured by using the histogram method, the lower limit is –1.

67.8 D/A Conversion Characteristics

Table 67.129 D/A Conversion Characteristics

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Resolution			—	12	12	12	Bit	—
Unbuffered output	Absolute accuracy	AVCC0 ≥ 2.7 V	—	—	—	±24	LSB	2-MΩ resistive load
				—	—	±6.0		No load
		AVCC0 < 2.7 V		—	—	±36		2-MΩ resistive load
				—	—	±6.0		No load
	Differential nonlinearity error	AVCC0 ≥ 2.7 V	DNL	—	—	±1.0	LSB	—
		AVCC0 < 2.7 V		—	—	±2.0		
	Output resistance		R_O	—	7.8	—	kΩ	—
	Recovery time	AVCC0 ≥ 2.7 V	$t_{DSPLUP1}$	—	—	3	μs	—
		AVCC0 < 2.7 V		—	—	6		
	Conversion time	AVCC0 ≥ 2.7 V	t_{DCONV1}	—	—	3	μs	2-MΩ resistive load, 20-pF capacitive load
			t_{DCONV3}	—	—	3		No load
		AVCC0 < 2.7 V	t_{DCONV1}	—	—	6		2-MΩ resistive load, 20-pF capacitive load
			t_{DCONV3}	—	—	6		No load
Buffered output	Integral nonlinearity error	AVCC0 ≥ 2.7 V	INL	—	—	±4.0	LSB	—
		AVCC0 < 2.7 V		—	—	±8.0		
	Differential nonlinearity error	AVCC0 ≥ 2.7 V	DNL	—	—	±1.0	LSB	
		AVCC0 < 2.7 V		—	—	±2.0		
	Start time	AVCC0 ≥ 2.7 V	t_{DISOUT}	3	—	—	μs	
		AVCC0 < 2.7 V		6	—	—		
	Recovery time	AVCC0 ≥ 2.7 V	$t_{DSPLUP2}$	—	—	3.5	μs	
		AVCC0 < 2.7 V		—	—	6		
	Conversion time	AVCC0 ≥ 2.7 V	t_{DCONV2}	—	—	3.5	μs	
		AVCC0 < 2.7 V		—	—	6		
	Load resistance		R_L	5	—	—	kΩ	
	Load capacitance		C_L	—	—	50	pF	
	Output voltage	AVCC0 ≥ 2.7 V	V_O	0.2	—	AVCC0 – 0.2	V	
		AVCC0 < 2.7 V		0.34	—	AVCC0 – 0.34		

Note: Values in the table above are for the case where the D/A converter is in stand-alone operation (that is, the A/D converters are stopped).

Note: When using the converter with AVCC0 < 2.7 V, set the OFSSEL bit in D/A control register 2 (DACR2) to 1. When using it with AVCC0 at or above 2.7 V, set the OFSSEL bit to 0 (the default value).

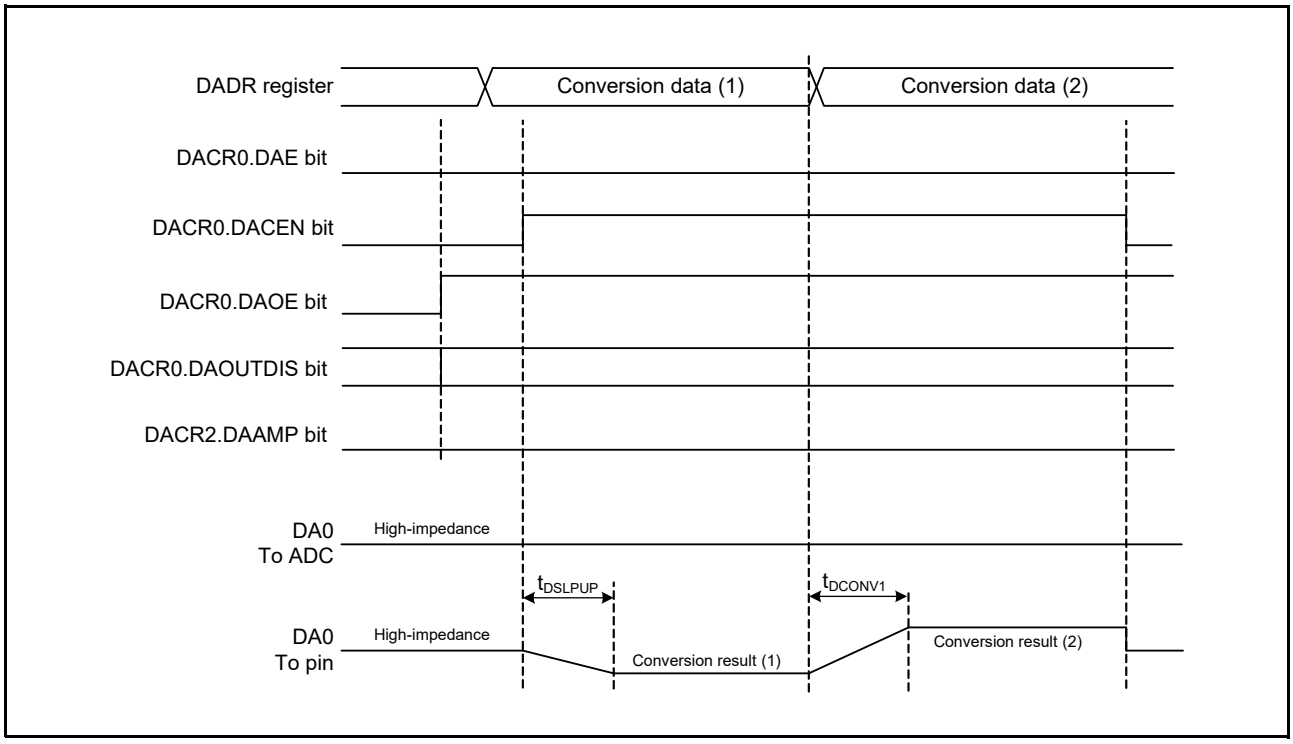


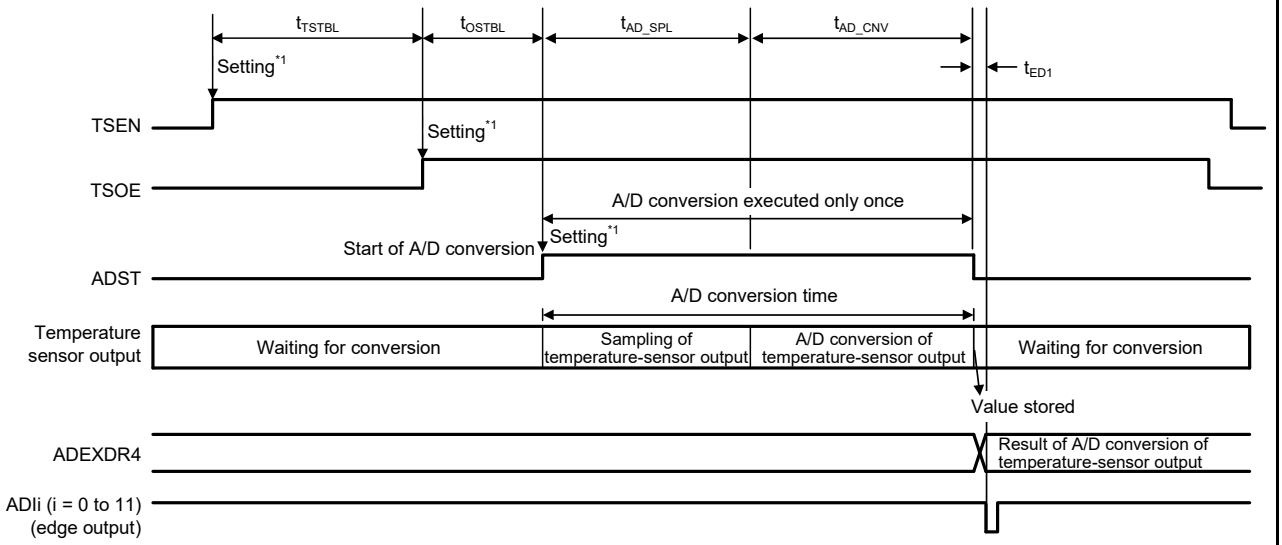
Figure 67.133 12-bit D/A Converter Operation Example (Mode for Outputting to the D/A Pin)

67.9 Temperature Sensor Characteristics

Table 67.130 Temperature Sensor Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy		—	—	±1	—	°C	—
Temperature slope		—	—	2.7	—	mV/°C	—
Output voltage		—	—	0.83	—	V	T_a = 25°C
Temperature sensor start time		t_{TSBL}	—	—	30	µs	Figure 67.134, Figure 67.135, Figure 67.136
Temperature sensor output stabilization wait time		t_{OSBL}	—	—	0	µs	
Stabilization wait time of the abnormal temperature detection circuit in the temperature sensor		t_{RSTBL}	—	—	30	µs	Figure 67.135, Figure 67.136
Sampling time*1		—	4.15	—	—	µs	—
Abnormal detection temperatures	High temperature threshold	T_{jdetH}	105	125	—	°C	Figure 67.137
	Low temperature threshold	T_{jdetL}	—	–40	–20		

Note 1. Set the ADSSTRn register such that the sampling time of the A/D converter satisfies this electrical characteristic.



Note 1. ↓ indicates the time of instruction execution by software.

Figure 67.134 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

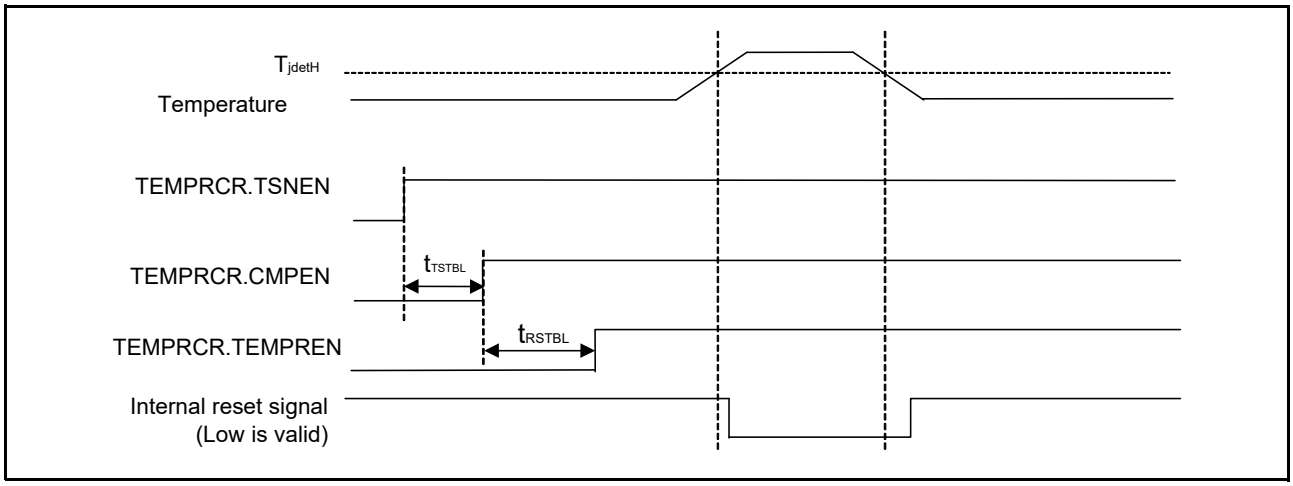


Figure 67.135 Timing of a Temperature Monitoring Reset in Response to High-Temperature Detection

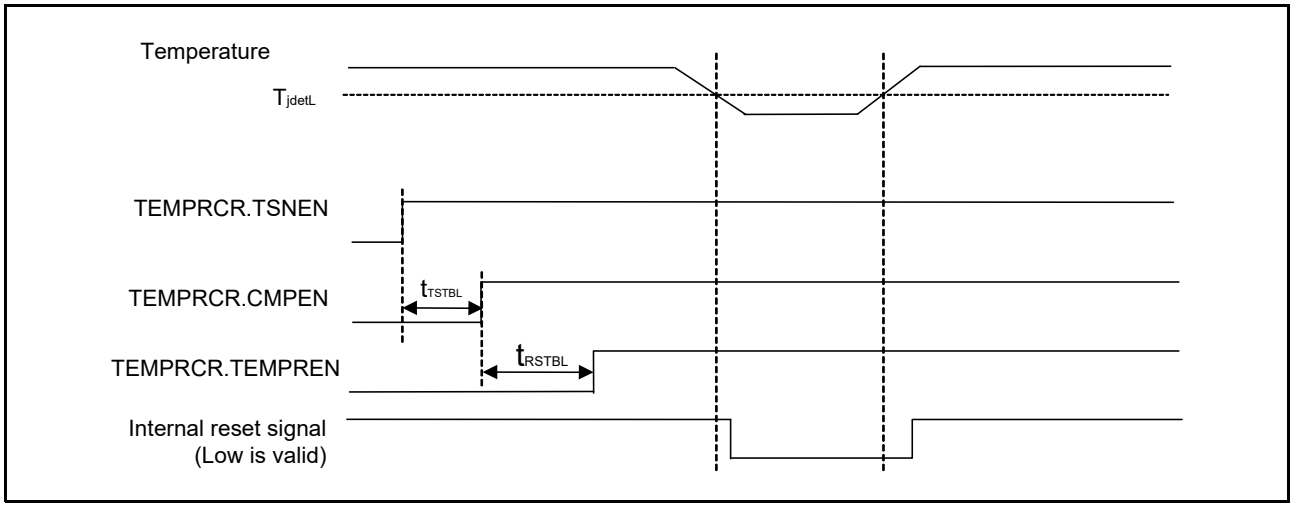


Figure 67.136 Timing of a Temperature Monitoring Reset in Response to Low-Temperature Detection

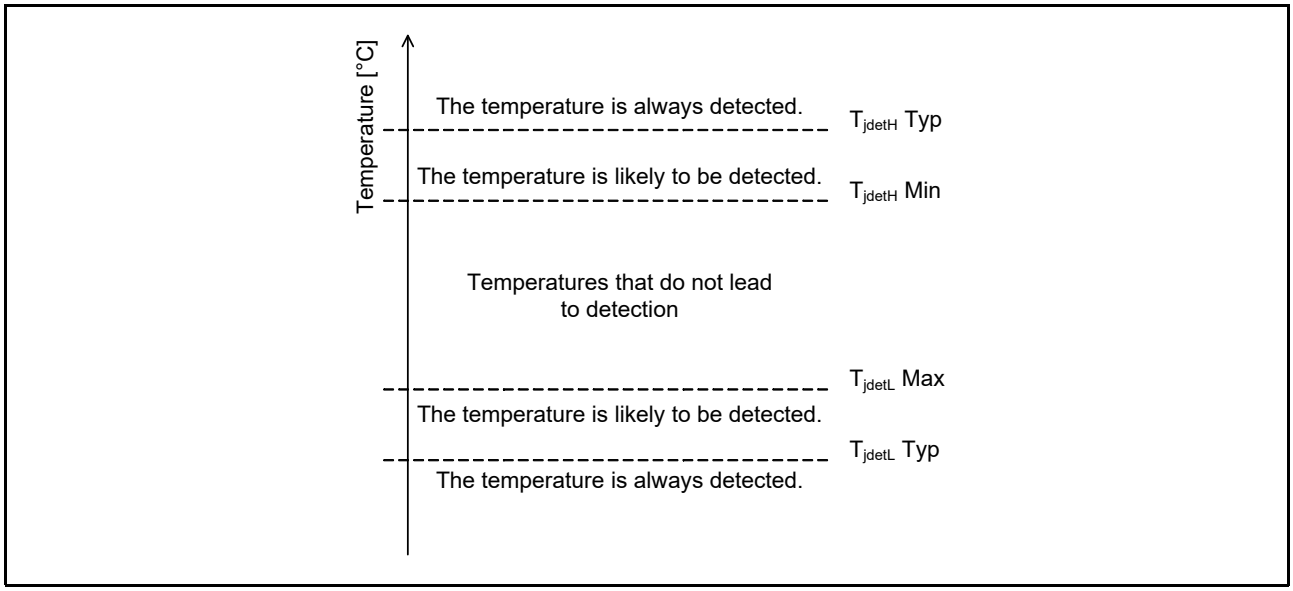


Figure 67.137 Range of Abnormal Temperatures to be Detected by the Temperature Sensor

67.10 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 67.131 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (1/2)

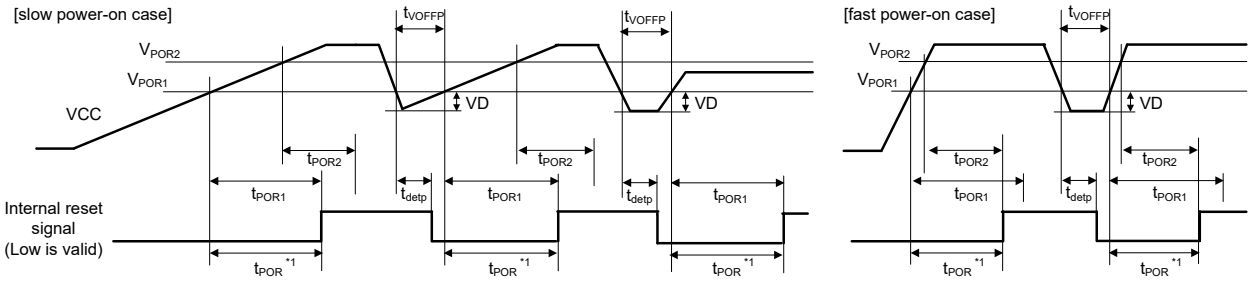
Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR1}	1.52	1.56	1.61	V	Figure 67.138
		V_{POR2}	—	—	1.73		
	Voltage detection circuit (PVD0)	V_{det0_0}	2.76	2.85	2.94		Figure 67.139
		V_{det0_1}	2.50	2.58	2.66		
		V_{det0_2}	2.08	2.15	2.22		
		V_{det0_3}	1.93	2.00	2.07		
		V_{det0_4}	1.84	1.90	1.96		
		V_{det0_5}	1.74	1.80	1.86		
		V_{det0_6}	1.62	1.67	1.73		
	Voltage detection circuit (LVDn) (n = 1, 2, 4, 5)	$V_{detn_4_rise}$	3.09	3.20	3.30		Figure 67.140
		$V_{detn_4_fall}$	3.03	3.14	3.24		
		$V_{detn_5_rise}$	3.05	3.16	3.26		
		$V_{detn_5_fall}$	2.99	3.10	3.20		
		$V_{detn_6_rise}$	3.03	3.14	3.24		
		$V_{detn_6_fall}$	2.97	3.08	3.18		
		$V_{detn_7_rise}$	2.81	2.91	3.00		
		$V_{detn_7_fall}$	2.75	2.85	2.94		
		$V_{detn_8_rise}$	2.79	2.89	2.98		
		$V_{detn_8_fall}$	2.73	2.83	2.92		
		$V_{detn_9_rise}$	2.76	2.86	2.95		
		$V_{detn_9_fall}$	2.70	2.80	2.89		
		$V_{detn_10_rise}$	2.58	2.67	2.75		
		$V_{detn_10_fall}$	2.53	2.62	2.70		
		$V_{detn_11_rise}$	2.30	2.38	2.46		
		$V_{detn_11_fall}$	2.25	2.33	2.41		
		$V_{detn_12_rise}$	1.88	1.94	2.00		
		$V_{detn_12_fall}$	1.84	1.90	1.96		
		$V_{detn_13_rise}$	1.84	1.90	1.96		
		$V_{detn_13_fall}$	1.80	1.86	1.92		
		$V_{detn_14_rise}$	1.72	1.78	1.84		
		$V_{detn_14_fall}$	1.68	1.74	1.80		
		$V_{detn_15_rise}$	1.66	1.72	1.77		
		$V_{detn_15_fall}$	1.62	1.68	1.73		
Internal reset time	Power-on reset time	t_{POR1}	—	—	6.7	ms	Figure 67.138
		t_{POR2}	—	—	1.6		
	PVD0 reset time	t_{PVD0}	—	—	*1		Figure 67.139
	PVD1 reset time	t_{PVD1}	—	—	*1		
	PVD2 reset time	t_{PVD2}	—	—	*1		
	PVD4 reset time	t_{PVD4}	—	—	*1		
	PVD5 reset time	t_{PVD5}	—	—	*1		
Minimum VCC drop time (POR)*2	50 mV < VD	t_{VOFFP}	900	—	—	µs	Figure 67.138
	VD ≤ 50 mV		2000	—	—		

Table 67.131 Power-on Reset Circuit and Voltage Detection Circuit Characteristics (2/2)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Minimum VCC drop time (PVD)*2	PVD0	t_{VOFF}	25	—	—	µs	Figure 67.139
	PVD1, PVD2, PVD4, PVD5		25	—	—		
Response delay time (POR)	50 mV < VD	t_{detp}	—	—	900	µs	Figure 67.138
	VD ≤ 50 mV		—	—	2000		
Response delay time (PVD)	PVD0	t_{det}	—	—	25	µs	Figure 67.139, Figure 67.140
	PVD1, PVD2, PVD4, PVD5		—	—	25		
PVDn operation stabilization time (after PVD is enabled) (n = 1, 2, 4, 5)		$t_{d(E-A)}$	—	—	20	µs	Figure 67.140

Note 1. On recovering from the deep software standby mode, the internal reset time reaches its maximum, so the maximum value of PVDn (n = 0, 1, 2, 4, 5) becomes equal to the value of t_{DSBY}.

Note 2. The minimum VCC drop time indicates the duration of VCC falling below the minimum voltage detection level of POR or PVD.



Note 1. t_{POR} is determined by the earlier point in time when the t_{POR1} has passed after crossing V_{POR1} or when the t_{POR2} has passed after crossing V_{POR2}.

Figure 67.138 Power-on Reset Timing

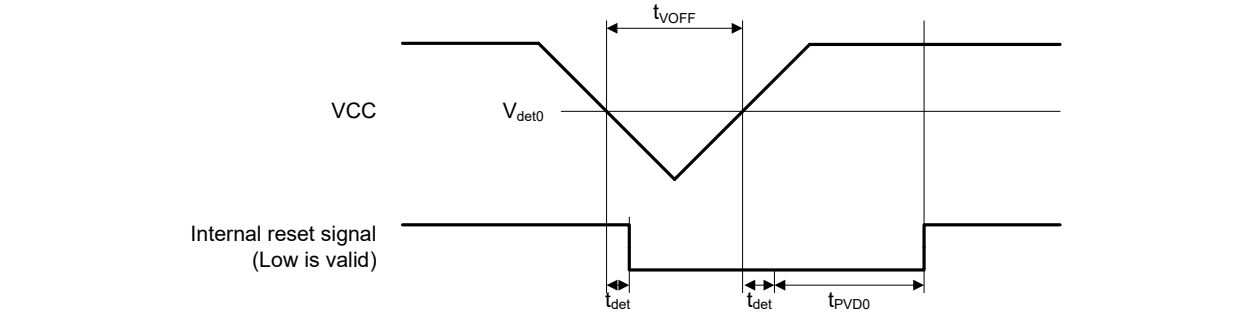


Figure 67.139 Voltage Detection Circuit Timing (V_{det0})

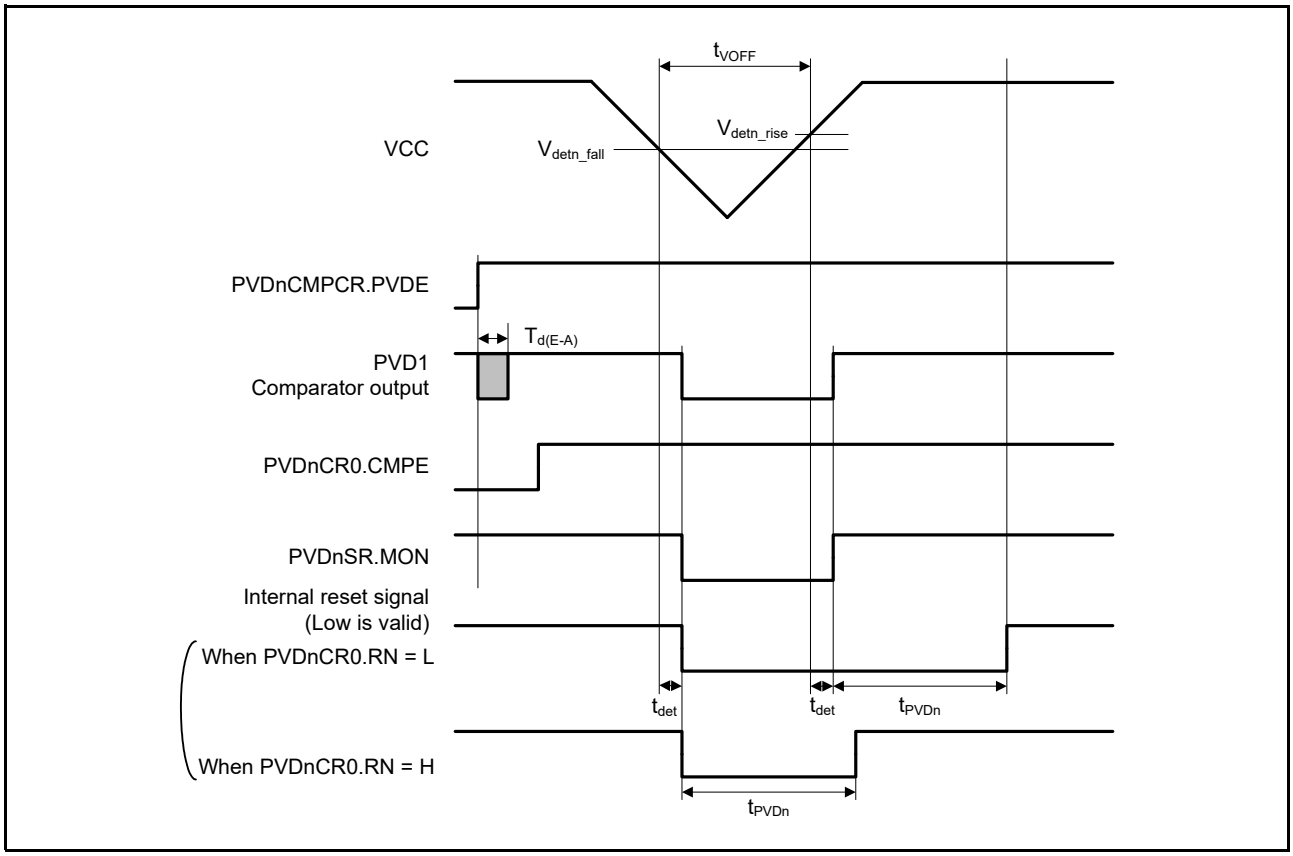


Figure 67.140 Voltage Detection Circuit Timing (V_{detn}) (n = 1, 2, 4, 5)

67.11 Core Voltage Monitoring Reset Characteristics

Table 67.132 Core Voltage Monitoring Reset Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level		V_{det_VDDH}	1.05	1.10	1.15	V	Figure 67.141
		V_{det_VDDL}	0.55	0.58	0.61		
Core voltage monitoring reset time	DCDC mode	t_{CVM}	—	—	0.18	ms	
	External VDD mode		—	—	2.6		
Minimum time for VDD falling and rising		t_{CVMOFF}	45	—	—	µs	
Response delay time		t_{CVMdet}	—	—	45	µs	
Hysteresis width		V_{CVMH}	—	0.225	—	V	

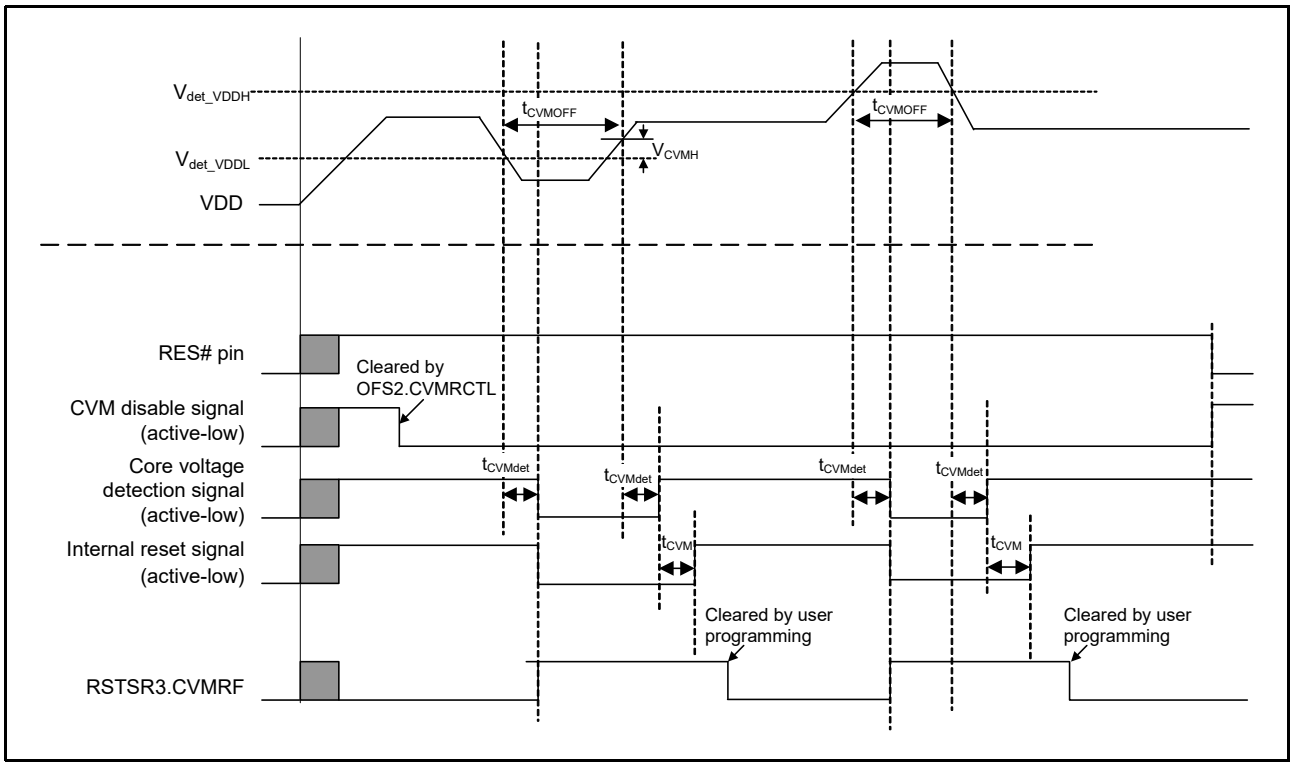


Figure 67.141 Core Voltage Monitoring Reset Time

67.12 Oscillation Stop Detection Timing

Table 67.133 Oscillation Stop Detection Circuit Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 67.142

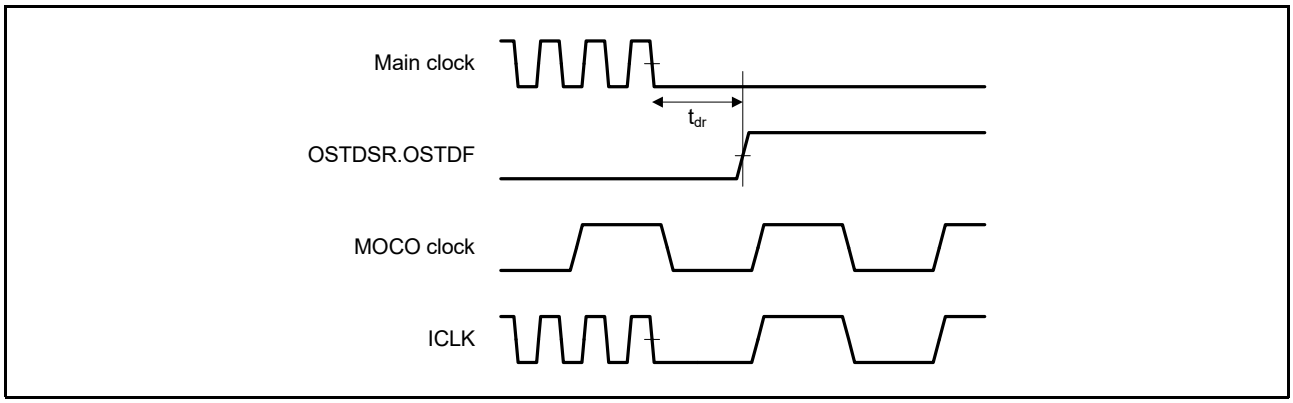


Figure 67.142 Oscillation Stop Detection Timing

Table 67.134 Sub-Clock Oscillation Stop Detection Circuit Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection circuit operation start waiting time	t_{sosdup}	100	—	—	µs	Figure 67.143
Detection time	t_{dr}	—	—	2	ms	Figure 67.144

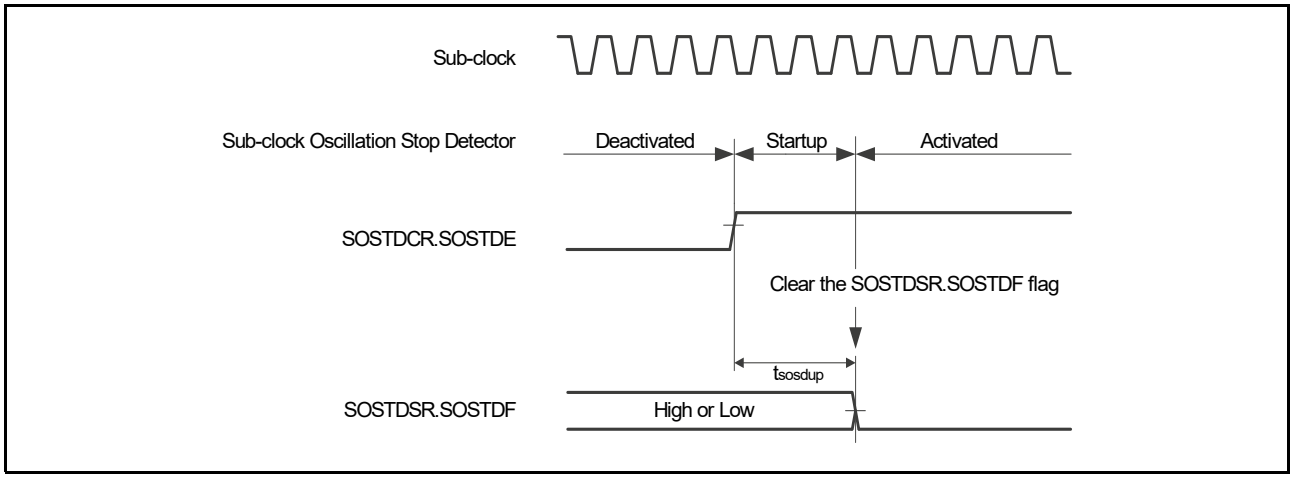


Figure 67.143 Detection Circuit Operation Start Waiting Time

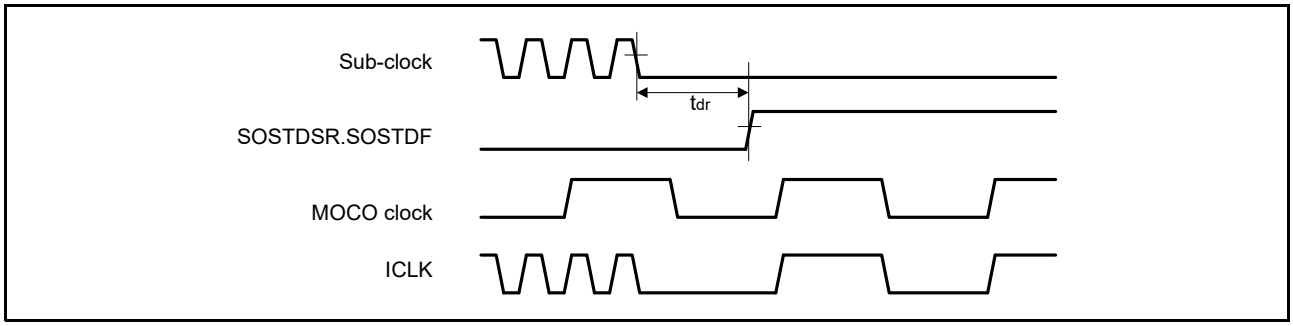


Figure 67.144 Sub-Clock Oscillation Stop Detection Timing

67.13 External VDD Timing Characteristics

Table 67.135 External VDD Timing Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reset hold time at power up of external VDD with using RES# pin	t_{EXTVRH}	600	—	—	µs	Figure 67.145
VDD rise time at power up of external VDD without using RES# pin	t_{EXTVDD}	—	—	550	µs	Figure 67.146

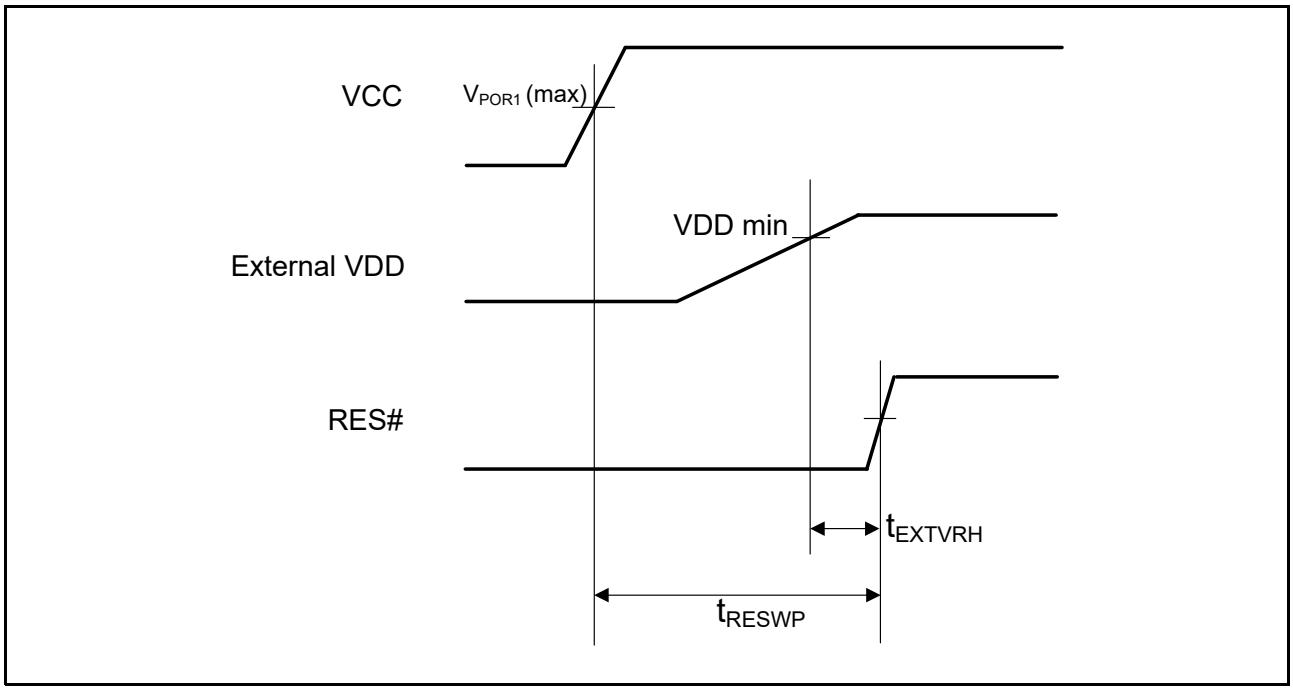


Figure 67.145 Power-up Sequence of External VDD Mode with Using RES Pin

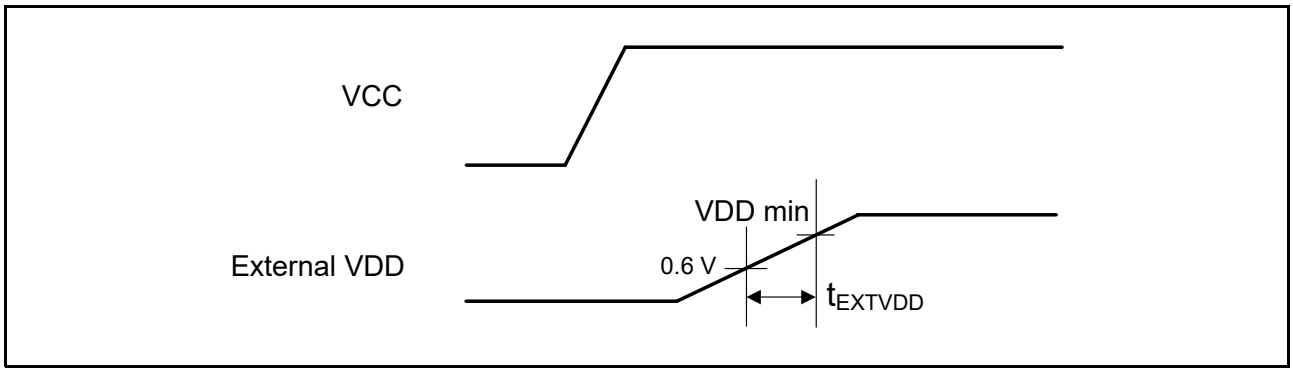


Figure 67.146 Power-up Sequence of External VDD Mode without Using RES Pin

67.14 Battery Backup Function Characteristics

Table 67.136 Battery Backup Function Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage level for switching to battery backup*1	$V_{DETBATT_0}$	2.71	2.80	2.89	V	Figure 67.147
	$V_{DETBATT_1}$	2.45	2.53	2.61		
	$V_{DETBATT_2}$	2.03	2.10	2.17		
	$V_{DETBATT_3}$	1.89	1.95	2.01		
	$V_{DETBATT_4}$	1.79	1.85	1.91		
	$V_{DETBATT_5}$	1.69	1.75	1.81		
VCC voltage drop detection stabilization waiting time*2	V_{DETWT}	—	—	20	µs	—
Lower-limit V_{BATT} voltage for power supply switching due to VCC voltage drop	V_{BATTSW}	1.8	—	—	V	Figure 67.147
VCC-off period for starting power supply switching*1, *3	$t_{VOFFBATT}$	25	—	—	µs	
Detection level for backup domain power-down	$V_{PDR(BKP)}$	1.43	1.47	1.52	V	Figure 67.148
Backup domain reset signal assertion delay time*4	$t_{P(PDRL)}$	—	—	2000	µs	
Backup domain reset signal negation delay time	$t_{P(PDRH)}$	—	—	3000	µs	
VBATT voltage measurement stabilization time*5	t_{MONWT}	—	—	4.2	µs	—

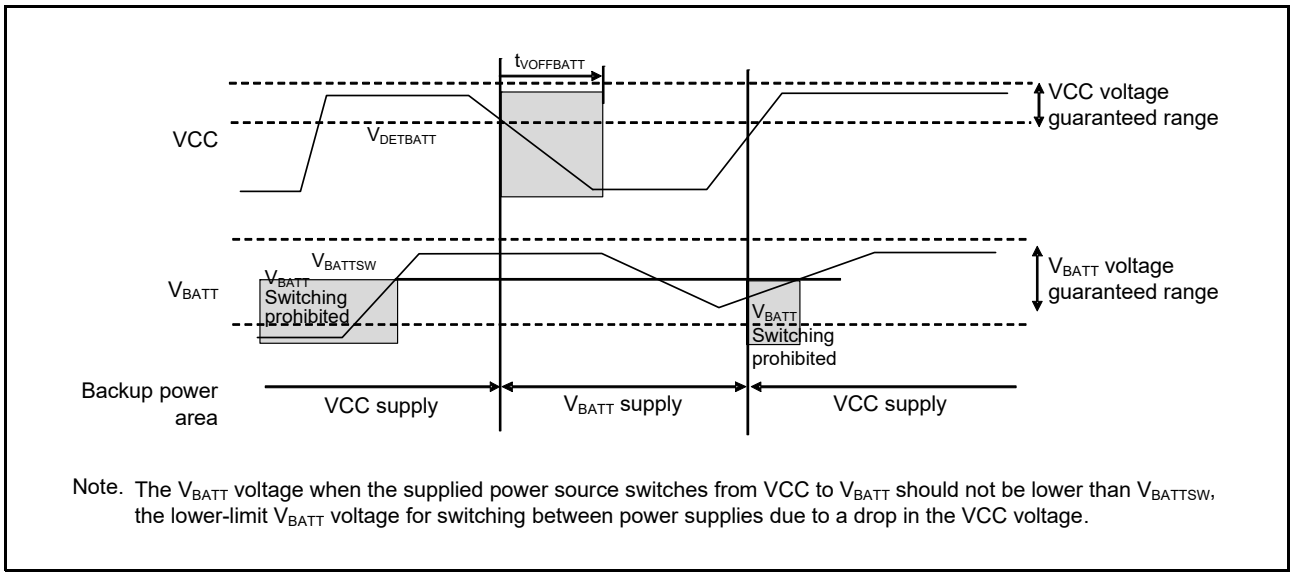
Note 1. In deep software standby modes 1 and 2, if the OFS1.PVDAS bit is 0 (voltage monitoring 0 reset is enabled after a reset) and the OFS1.DSBYLPSEL bit is 0 (PVD0 is selected, low power function is enabled, operation depends on VDSEL[2:0] bit setting), the battery backup switching voltage level is V_{det0}.

Note 2. This indicates the time the voltage detection circuit requires to become stable after the value of the VBATTCR2.VDETSEL[2:0] bits has been changed or the value of the VBATTCR1.PSWDIS bit is changed from 1 to 0.

Note 3. This indicates the time over which VCC is below the minimum level of the battery backup switching voltage ($V_{DETBATT_n}$).

Note 4. If the V_{BKP} voltage recovers, that is, exceeds the $V_{POR(BKP)}$ voltage, within this period, the backup domain reset signal may not be generated.

Note 5. This indicates the time that the output voltage required to become stable after the VBATTMCR.VBATTME bit is set to 1.



Note. The V_{BATT} voltage when the supplied power source switches from VCC to V_{BATT} should not be lower than V_{BATTSW}, the lower-limit V_{BATT} voltage for switching between power supplies due to a drop in the VCC voltage.

Figure 67.147 Battery Backup Function Characteristics

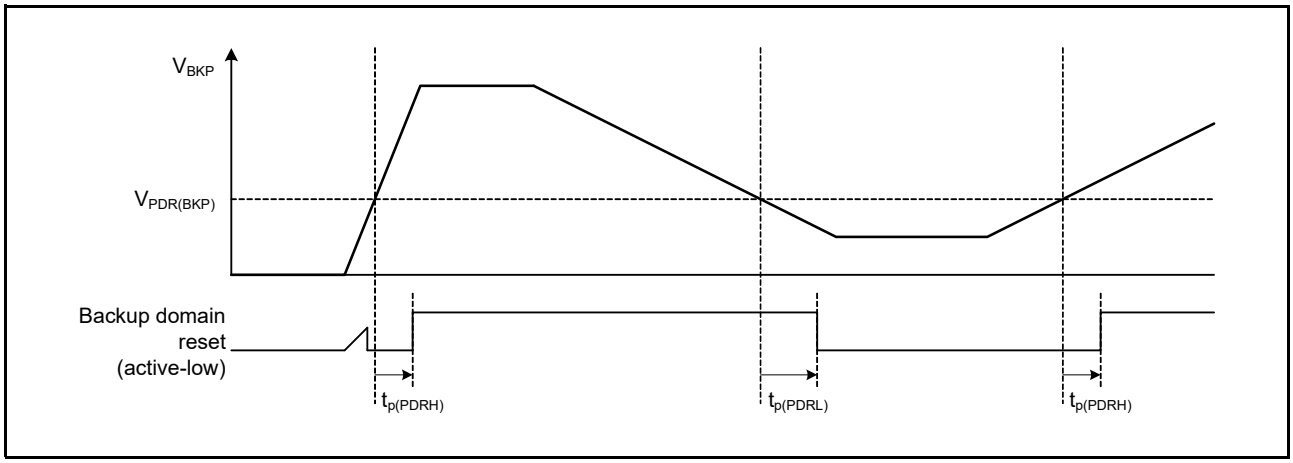


Figure 67.148 Backup Domain Reset Characteristics

67.15 MRAM Characteristics

67.15.1 Code MRAM Characteristics

Table 67.137 Code MRAM Characteristics

Item		Symbol	MRICLK = 300 MHz			MRICLK = 240 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Time for programming 32 bytes*1, *2	Normal program mode*3	t_{PMC}	—	6.53	81.3	—	6.72	83.5	µs
	High speed program mode*3		—	4.53	79.3	—	4.72	81.5	
Number of programming operations*4		N_{PC}	100 k	—	—	100 k	—	—	Cycle
Data retention time*5		t_{DRP}	10	—	—	10	—	—	Year

Note 1. When MRPCLK < 125 MHz, add one MRPCLK cycle in units of µs to the time for programming.
Note 2. Use the following approximation formulae to calculate the time for programming at other frequencies:
- Normal program mode
  t_{PMC} (Typ) = 137.8 / f_{MRICLK} + 6.452 [µs], t_{PMC} (Max) = 1879 / f_{MRICLK} + 78.75 [µs]
- High speed program mode
  t_{PMC} (Typ) = 137.8 / f_{MRICLK} + 4.452 [µs], t_{PMC} (Max) = 1879 / f_{MRICLK} + 76.75 [µs]
  (f_{MRICLK}: MRICLK frequency [MHz])

Note 3. The programming mode is selected by the setting of the MHSPEN bit in the MRPSC register.
0: Normal program mode; 1: High speed program mode
Note 4. This indicates the number of programming operations that can be performed per 32-byte code MRAM address range. Programming a given address with the same data also increases the number of programming operations.
Note 5. Test condition: Tj = 125°C

67.15.2 Option-Setting Memory (MRAM Area) Characteristics

Table 67.138 Option-Setting Memory (MRAM Area) Characteristics

Item		Symbol	MRPCLK = 150 MHz			MRPCLK = 120 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Time to process a configuration set command*1	Normal program mode*2	t_{PCFG}	—	0.343	7.8	—	0.352	8.08	ms
	High speed program mode*2		—	0.0596	7.46	—	0.0683	7.74	
Number of programming operations*3		N_{CUPC}	100 k	—	—	100 k	—	—	Cycle
Data retention time*4		t_{DRP}	10	—	—	10	—	—	Year

Note 1. Use the following approximation formulae to calculate the times for processing at other frequencies:
- Normal program mode
  t_{PCFG} (Typ) = 6.146 / f_{MRPCLK} + 0.3133 [ms], t_{PCFG} (Max) = 266.5 / f_{MRPCLK} + 6.331 [ms]
- High speed program mode
  t_{PCFG} (Typ) = 5.184 / f_{MRPCLK} + 0.02754 [ms], t_{PCFG} (Max) = 267.5 / f_{MRPCLK} + 6.025 [ms]
  (f_{MRPCLK}: MRPCLK frequency [MHz])

Note 2. The programming mode is selected by the setting of the MHSPEN bit in the MRPSC register.
0: Normal program mode; 1: High speed program mode
Note 3. This indicates the number of configuration set commands that can be issued for the MRAM area. Programming a given address with the same data also increases the number of programming operations.
Note 4. Test condition: Tj = 125°C

67.15.3 Option-Setting Memory (OTP Area with ECC) Characteristics

Table 67.139 Option-Setting Memory (OTP Area with ECC) Characteristics

Item		Symbol	MRPCLK = 150 MHz			MRPCLK = 120 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Time for programming 16 bytes*1	MRAM-OTP normal speed write mode*2	t_{OTPE}	—	8.05	113	—	8.05	113	ms
	MRAM-OTP high speed write mode 0*2		—	4.03	56.7	—	4.03	56.7	
	MRAM-OTP high speed write mode 1*2		—	0.229	3.14	—	0.229	3.14	

Note 1. Use the following approximation formulae to calculate the time for programming at other frequencies:
- MRAM-OTP normal speed write mode
  t_{OPTE} (Typ) = 0.5065 / f_{MRPCLK} + 8.123 [ms], t_{OPTE} (Max) = 4.433 / f_{MRPCLK} + 114.1 [ms]
- MRAM-OTP high speed write mode 0
  t_{OPTE} (Typ) = 0.3389 / f_{MRPCLK} + 4.067 [ms], t_{OPTE} (Max) = 2.428 / f_{MRPCLK} + 57.08 [ms]
- MRAM-OTP high speed write mode 1
  t_{OPTE} (Typ) = 0.1458 / f_{MRPCLK} + 0.2312 [ms], t_{OPTE} (Max) = 0.3904 / f_{MRPCLK} + 3.166 [ms]
  (f_{MRPCLK}: MRPCLK frequency [MHz])

Note 2. The MRAM-OTP write mode is selected by the setting of the MWM[1:0] bits in the MWMCR register.
00b: Normal-speed write mode; 01b: High-speed write mode 0; 10b: High-speed write mode 1

67.15.4 Option-Setting Memory (OTP Area without ECC) Characteristics

Table 67.140 Option-Setting Memory (OTP Area without ECC) Characteristics

Item		Symbol	MRPCLK = 150 MHz			MRPCLK = 120 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Time for programming 16 bytes*1	MRAM-OTP normal speed write mode*2	t_{OTPENE}	—	14.1	200	—	14.1	200	ms
	MRAM-OTP high speed write mode 0*2		—	7.07	100	—	7.07	100	
	MRAM-OTP high speed write mode 1*2		—	0.458	6.28	—	0.458	6.28	

Note 1. Use the following approximation formulae to calculate the time for programming at other frequencies:
- MRAM-OTP normal speed write mode
  t_{OPTENE} (Typ) = 0.8486 / f_{MRPCLK} + 14.25 [ms], t_{OPTENE} (Max) = 7.711 / f_{MRPCLK} + 201.5 [ms]
- MRAM-OTP high speed write mode 0
  t_{OPTENE} (Typ) = 0.5479 / f_{MRPCLK} + 7.133 [ms], t_{OPTENE} (Max) = 4.148 / f_{MRPCLK} + 100.8 [ms]
- MRAM-OTP high speed write mode 1
  t_{OPTENE} (Typ) = 0.2571 / f_{MRPCLK} + 0.4627 [ms], t_{OPTENE} (Max) = 0.7401 / f_{MRPCLK} + 6.333 [ms]
  (f_{MRPCLK}: MRPCLK frequency [MHz])

Note 2. The MRAM-OTP write mode is selected by the setting of the MWM[1:0] bits in the MWMCR register.
00b: Normal-speed write mode; 01b: High-speed write mode 0; 10b: High-speed write mode 1

67.15.5 MACI Command Characteristics

Table 67.141 MACI Command Characteristics

Item	Symbol	MRPCLK = 150 MHz			MRPCLK = 120 MHz			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
Time to execute a forced-stop command	t_{FS}	—	—	3.31	—	—	3.36	µs
Time to execute an increment counter command	t_{INCC}	—	0.252	1.61	—	0.252	1.61	ms
Time to execute a read counter command	t_{RDC}	—	—	0.13	—	—	0.163	µs

Note: Use the following approximation formulae to calculate the times to execute the commands at other frequencies:
$t_{FS} = 38.62 / f_{MRPCLK} + 3.155$ [µs]
t_{INCC} (Typ) $= 0.3348 / f_{MRPCLK} + 0.2533$ [ms], t_{INCC} (Max) $= 0.8698 / f_{MRPCLK} + 1.62$ [ms]
$t_{RDC} = 19.13 / f_{MRPCLK} + 0.004099$ [µs]
(f_{MRPCLK}: MRPCLK frequency [MHz])

67.15.6 W-HUK Zeroization Characteristics

Table 67.142 W-HUK Zeroization Characteristics

Item	Symbol	MRPCLK = 150 MHz			MRPCLK = 120 MHz			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
Time for W-HUK zeroization operation	t_{ZWH}	—	—	793	—	—	793	ms

Note: Use the following approximation formula to calculate the times for operation at other frequencies:
$t_{ZWH} = 34.59 / f_{MRPCLK} + 799$ [ms]
(f_{MRPCLK}: MRPCLK frequency [MHz])

67.15.7 MRAM Magnetic Field Resistance Characteristics

Table 67.143 MRAM Magnetic Field Resistance Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Operation	Writing	G_{max_wr}	—	—	200	Gauss	—
	Reading	G_{max_rd}	—	—	200		
	No access in progress	G_{max_noac}	—	—	500		
Storage		G_{max_stg}	—	—	500		

67.15.8 Data MRAM Characteristics

Table 67.144 Data MRAM Characteristics

Item		Symbol	MRPCLK = 150 MHz			MRRCLK = 120 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
16-byte programming time*1	Normal program mode*2	t_{PME}	—	6.63	72.8	—	6.87	74.9	µs
	High speed program mode*2		—	4.63	70.8	—	4.87	72.9	
Number of programming operations*3		N_{PC}	100 k	—	—	100 k	—	—	Cycle
Data retention time*4		t_{DRP}	10	—	—	10	—	—	Year

Note 1. Use the following approximation formulae to calculate the time for programming at other frequencies:
- Normal program mode
  t_{PME} (Typ) = 179.2 / f_{MRPCLK} + 5.725 [µs], t_{PME} (Max) = 1988 / f_{MRPCLK} + 61.9 [µs]
- High speed program mode
  t_{PME} (Typ) = 179.2 / f_{MRPCLK} + 3.725 [µs], t_{PME} (Max) = 1988 / f_{MRPCLK} + 59.9 [µs]
  (f_{MRPCLK}: MRPCLK frequency [MHz])

Note 2. The programming mode is selected by the setting of the MHSPEN bit in the MRPSC register.
0: Normal program mode; 1: High speed program mode

Note 3. This indicates the number of programming operations that can be performed per 16-byte data MRAM address range. Programming a given address with the same data also increases the number of programming operations.

Note 4. Test condition: Tj = 125°C

67.16 Boundary Scan

Table 67.145 Boundary Scan Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
TCK clock cycle time	t_{TCKcyc}	100	—	—	ns	Figure 67.149
TCK clock high pulse width	t_{TCKH}	45	—	—	ns	
TCK clock low pulse width	t_{TCKL}	45	—	—	ns	
TCK clock rise time	t_{TCKr}	—	—	5	ns	
TCK clock fall time	t_{TCKf}	—	—	5	ns	
TMS setup time	t_{TMSS}	20	—	—	ns	Figure 67.150
TMS hold time	t_{TMSH}	20	—	—	ns	
TDI setup time	t_{TDIS}	20	—	—	ns	
TDI hold time	t_{TDIH}	20	—	—	ns	
TDO data delay time	t_{TDOD}	—	—	40	ns	
Boundary scan circuit activation time	t_{BSSTUP}	t_{RESWP}	—	—	—	Figure 67.151

Note 1. Boundary scanning does not function until power-on reset is disabled.

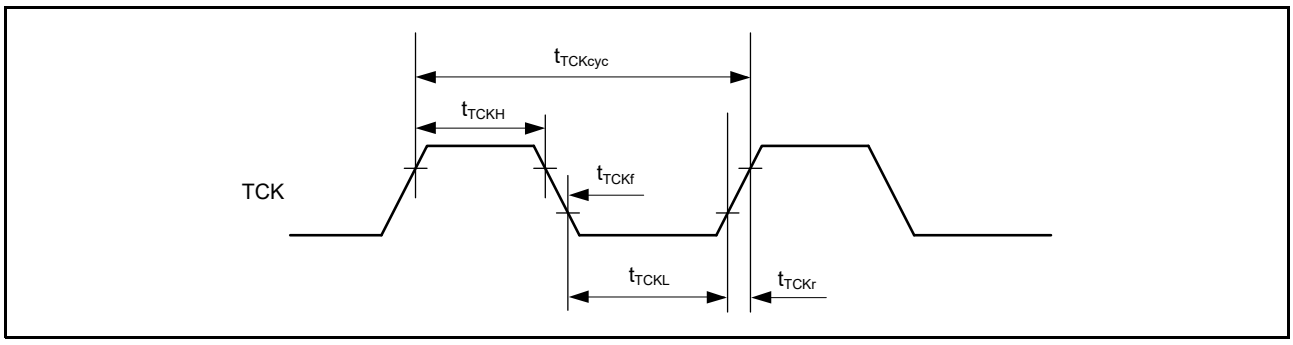


Figure 67.149 Boundary Scan TCK Timing

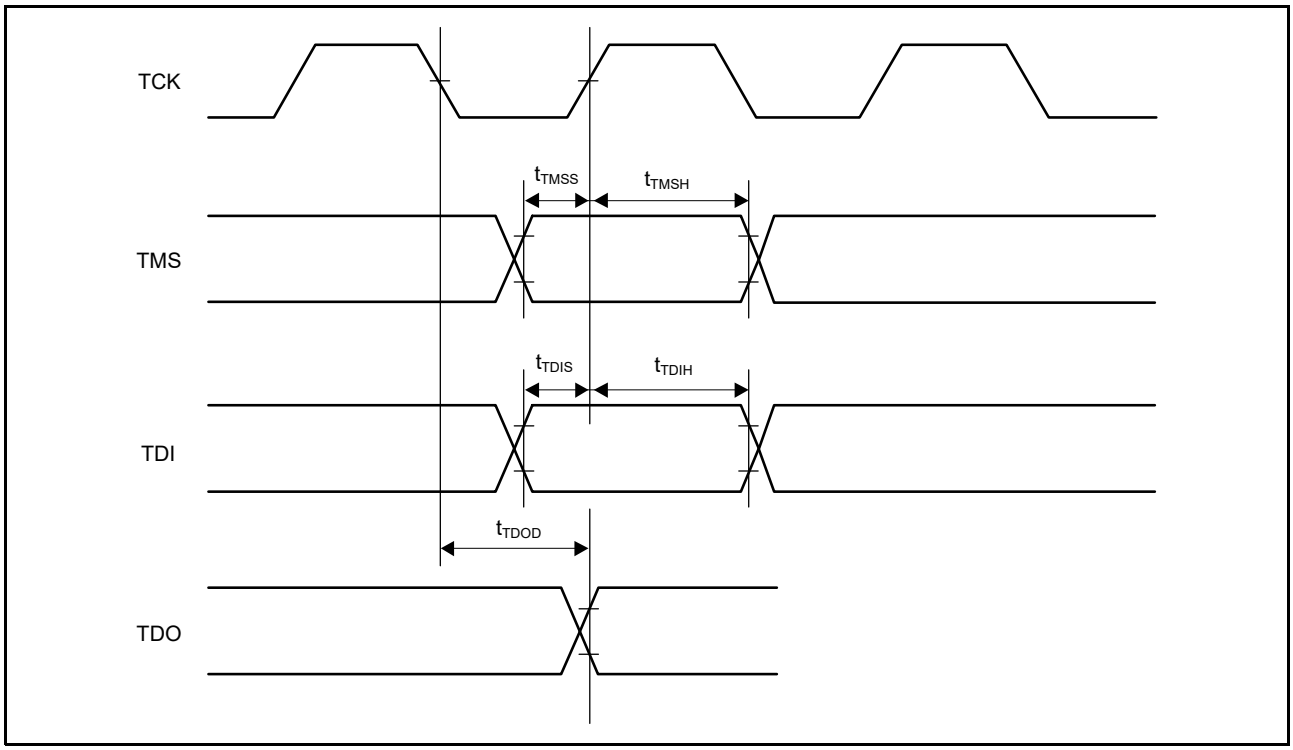


Figure 67.150 Boundary Scan Input/Output Timing

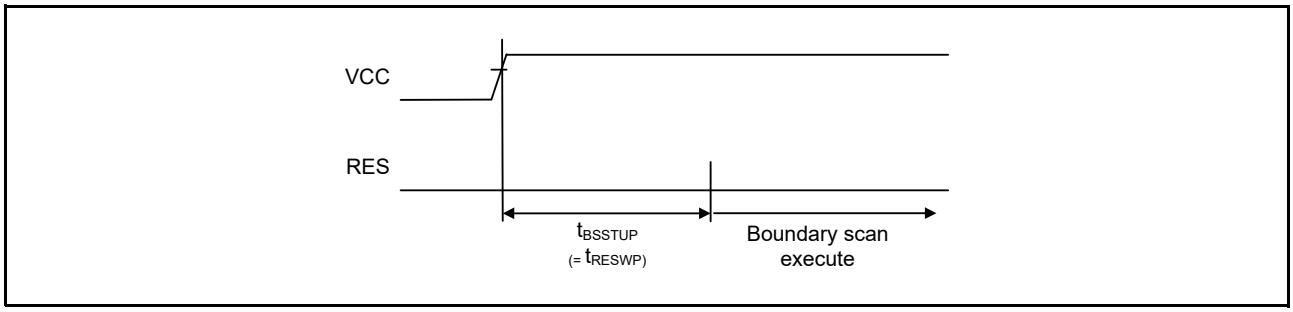


Figure 67.151 Boundary Scan Circuit Activation Time

67.17 Usage Note

67.17.1 Connecting VCL Capacitor and Bypass Capacitors

This MCU integrates an internal voltage-down circuit, which is used for lowering the power supply voltage in the internal MCU to adjust automatically to the optimum level. The number of internal step-down power supplies (VCL pins) differs with the type of package. Connect a capacitor for stabilizing the internal voltage between each VCL pin and a VSS pin. For details on the values of capacitors for each type of package, see Table 67.3, Recommended Operating Conditions (2). Place an external capacitor close to the pins. Do not apply the power supply voltage to the VCL pin.
In addition, place a multilayer ceramic capacitor as a bypass capacitor between each VCC or VCC2 pin and a nearby VSS pin. Implement a bypass capacitor to the MCU power supply pins as close as possible. Use a recommended value of 0.1 µF as the capacitance of the capacitors. For the capacitors related to crystal oscillation, see section 9, Clock Generation Circuit. For the capacitors related to analog modules, also see section 60, 16-Bit A/D Converter (ADC16H).
For notes on designing the printed circuit board, see the descriptions of the application note “Hardware Design Guide” (R01AN1411EJ). The latest version can be downloaded from Renesas Electronics Website.

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1/2)

Functions	Pin Function	Resets	Software Standby Mode 1, 2 (SSTBY1, 2)		Deep Software Standby Mode 1, 2, 3 (DSTBY1, 2, 3) IOKEEP = 1/0		After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 0	OPE = 1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1*1
Operating mode control	MD	Pull-up	Keep-I		Keep		Pull-up	Keep
On-chip emulator	TCK/TMS/TDI/SWCLK	Pull-up	TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input	
	TDO	Output	TDO output		TDO output		TDO output	
	SWDIO	Pull-up	SWDIO input/output		SWDIO input/output		SWDIO input/output	
Interrupts	IRQ0 to IRQ31	Hi-Z	Hi-Z*2		Keep		Hi-Z	Keep
	IRQ0-DS to IRQ31-DS	Hi-Z	Hi-Z*2		Keep*3		Hi-Z	Keep
High-Speed I2C-bus Interface	SCLHS1/SCLHS2/ SDAHS1/SDAHS2	Hi-Z	Keep-O*2		Keep		Hi-Z	Keep
I3C Bus Interface	SCL00/SDA00	Hi-Z	Keep-O*2		Hi-Z		Hi-Z	
USB 2.0 full-speed host/ function module	USB0_OVRCURA/ USB0_OVRCURB/ USB0_VBUS	Hi-Z	Hi-Z*2		Keep		Hi-Z	Keep
	USB0_OVRCURA-DS/ USB0_OVRCURB-DS/ USB0_VBUS-DS	Hi-Z	Hi-Z*2		Keep*3	Keep	Hi-Z	Keep
	USB0_DP/USB0_DM	Hi-Z	Keep-O*4		Keep*3	Keep	Hi-Z	Keep
USB 2.0 high-speed host/function module	USBHS_OVRCURA/ USBHS_OVRCURB/ USBHS_VBUS	Hi-Z	Hi-Z*2		Keep		Hi-Z	Keep
	USBHS_OVRCURA-DS/ USBHS_OVRCURB-DS/ USBHS_VBUS-DS	Hi-Z	Hi-Z*2		Keep*3	Keep	Hi-Z	Keep
	USBHS_DP/ USBHS_DM	Hi-Z	Keep-O*4		Keep*5	Keep	Hi-Z	Keep
Realtime clock	RTCIC0 to RTCIC2	Hi-Z	Hi-Z*2		Keep*3		Hi-Z	Keep
	RTCOUT	Hi-Z	RTCOUT output		Keep		Hi-Z	Keep
Clock	CLKOUT	Hi-Z	CLKOUT output		Keep		Hi-Z	Keep
12-bit D/A converter	DA0, DA1	Hi-Z	D/A output retained		Keep		Hi-Z	Keep
External bus (CS area controller, SDRAM area controller)	EBCLK/SDCLK	Hi-Z	High output		Keep		Hi-Z	Keep
	D0 to D31	Hi-Z	Hi-Z		Hi-Z		Hi-Z	
	A0 to A23/DQM0 to DQM3	Hi-Z	Hi-Z	Keep-O	Keep		Hi-Z	Keep
	BC0# to BC3#/CS0# to CS7#/RD#/WR#/WE#/ BACK#	Hi-Z	Hi-Z	High output	Keep		Hi-Z	Keep
	ALE	Hi-Z	Hi-Z	Low output	Keep		Hi-Z	Keep

Table 1.1 Port States in Each Processing State (2/2)

Functions	Pin Function	Resets	Software Standby Mode 1, 2 (SSTBY1, 2)		Deep Software Standby Mode 1, 2, 3 (DSTBY1, 2, 3) IOKEEP = 1/0		After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 0	OPE = 1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1*1
External bus (CS area controller, SDRAM area controller)	CKE/SDCS#/RAS#/ CAS#	Hi-Z	Hi-Z	SDSELF. SFEN = 0: High output SDSELF. SFEN = 1: Low output	Keep		Hi-Z	Keep
Pulse-Density Modulation Interface (PDMIF)	PDMCLK0	Hi-Z	PDMCLK output		Keep		Hi-Z	Keep
	PDMDAT0	Hi-Z	Keep-O*2		Keep		Hi-Z	Keep
Other	—	Hi-Z	Keep-O		Keep		Hi-Z	Keep

Hi-Z: High-impedance

Keep-O: Output pins retain their previous values, and input pins become high-impedance.

Keep-I: Pin states are retained the same as during periods in Normal mode, Sleep mode, or Deep sleep mode.

Keep: Pin states are retained during periods on software standby.

Note 1. The I/O port state is retained until the DPSBYCR.IOKEEP bit is cleared to 0.

Note 2. Input is enabled while the pin is used as an external interrupt pin.

Note 3. Input is enabled if the pin is specified as the deep software standby cancelling source.

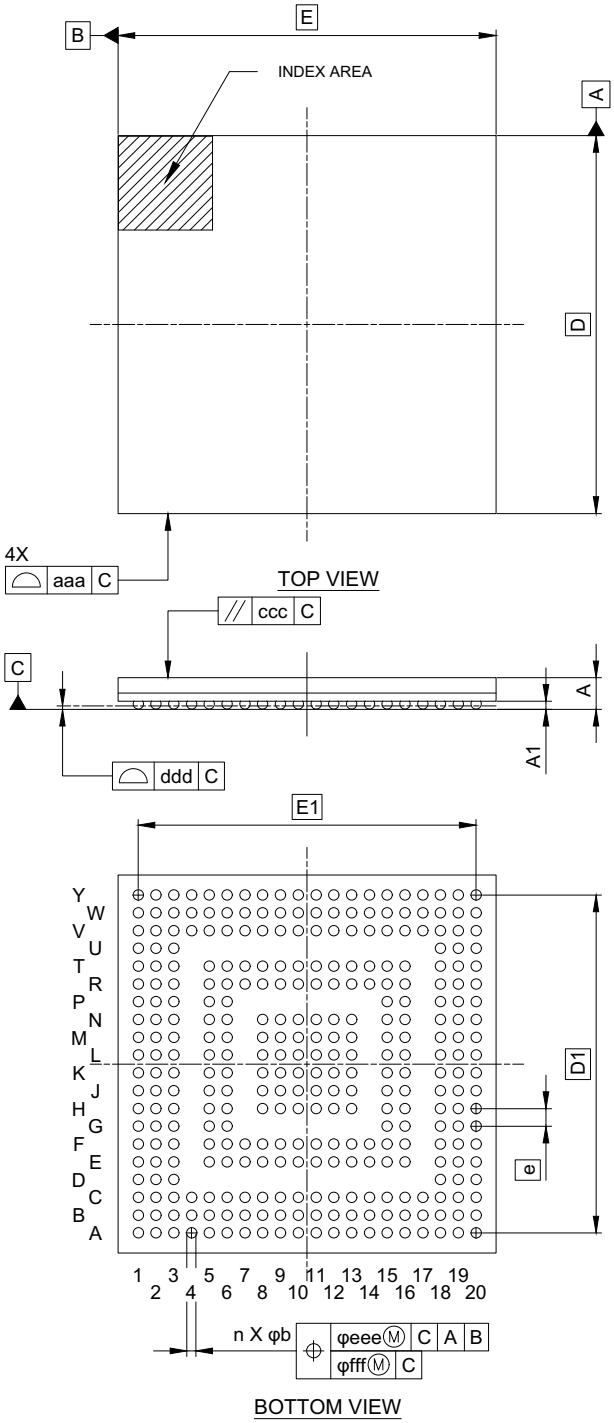
Note 4. Input is enabled while the pin is used as an input pin.

Note 5. For host operation, set the USBHS.SYSCFG.DRPD bit to 1 to enable the USBHS_DP and USBHS_DM pull-down resistors. For device operation, set the USBHS.SYSCFG.DPRPU bit to 1 to enable the DP pull-up resistor.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

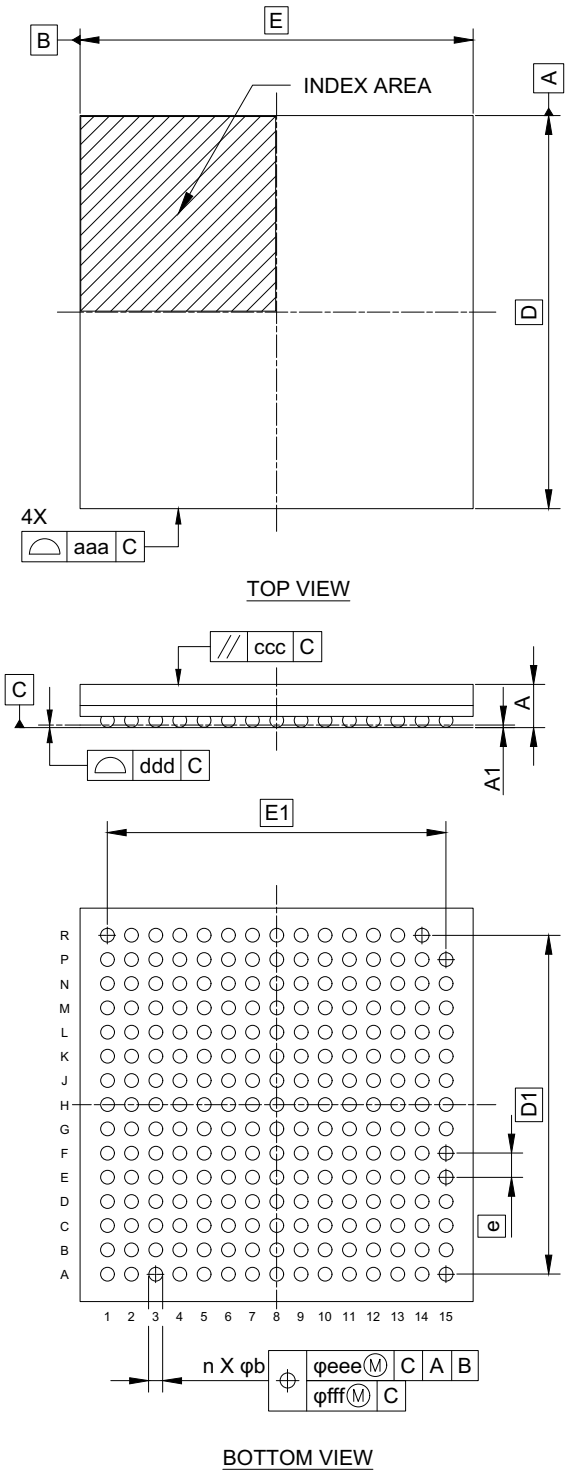
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA320-17x17-0.80	PLBG0320GD-A	-



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	17.00	—
E	—	17.00	—
D1	—	15.20	—
E1	—	15.20	—
A	—	—	1.60
A1	0.31	—	—
b	0.41	0.46	0.51
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.20
eee	—	—	0.15
fff	—	—	0.08
n	—	320	—

Figure A 320-Pin LFBGA (PLBG0320GD-A)

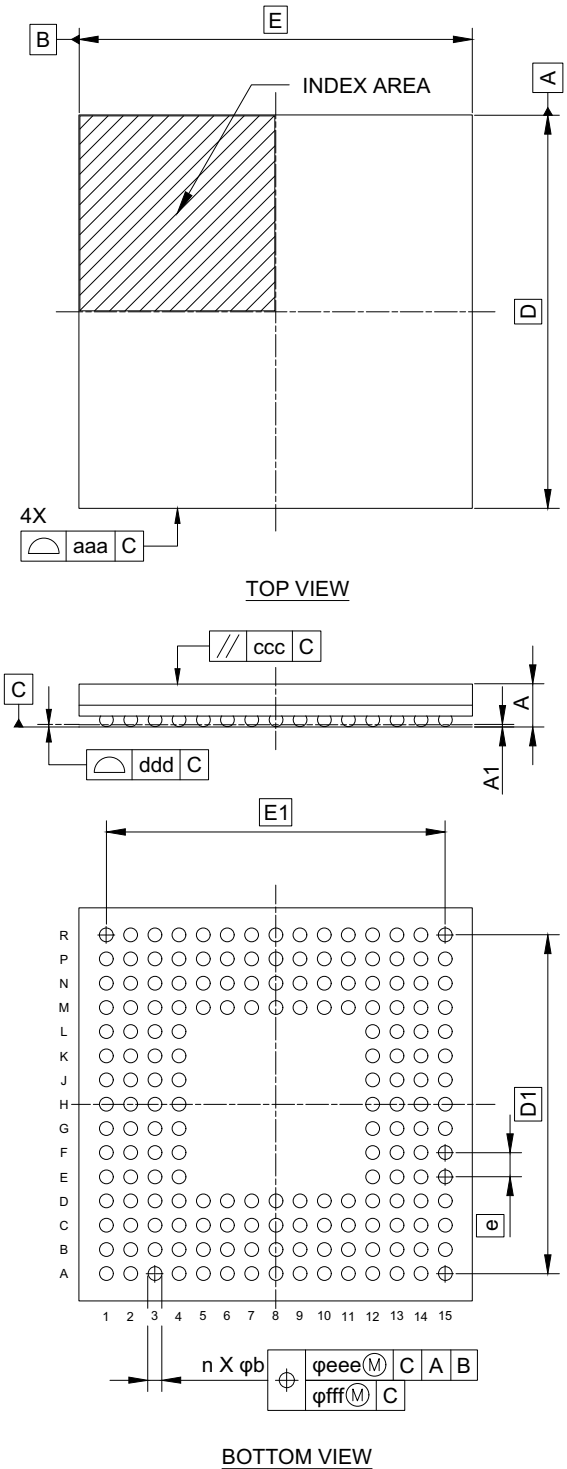
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA224-13x13-0.80	PLBG0224GE-A	-



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	13.00	—
E	—	13.00	—
D1	—	11.20	—
E1	—	11.20	—
A	—	—	1.60
A1	0.31	—	—
b	0.41	0.46	0.51
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.20
eee	—	—	0.15
fff	—	—	0.08
N	—	224	—

Figure B 224-Pin LFBGA (PLBG0224GE-A)

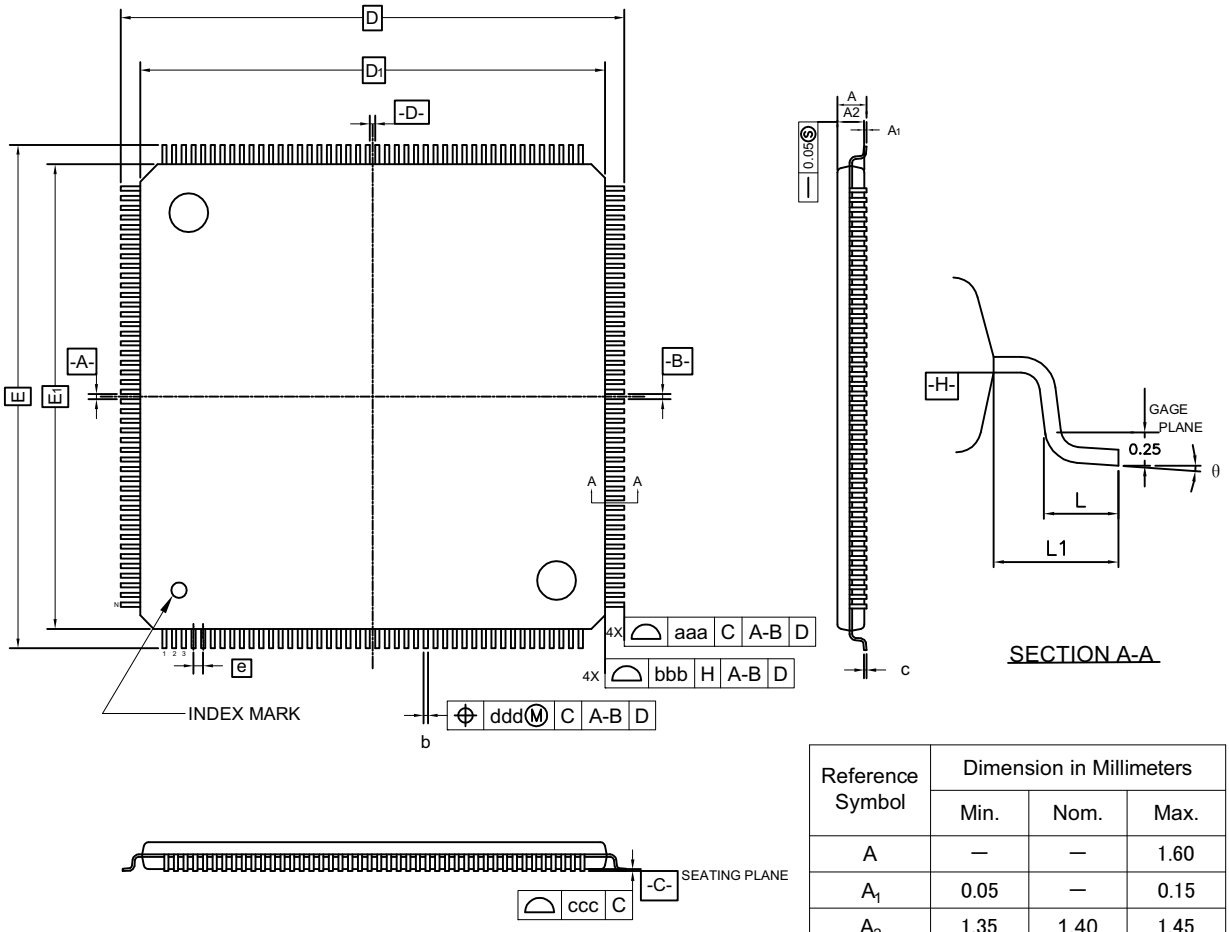
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA176-13x13-0.80	PLBG0176GG-A	-



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	13.00	—
E	—	13.00	—
D1	—	11.20	—
E1	—	11.20	—
A	—	—	1.60
A1	0.31	—	—
b	0.41	0.46	0.51
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.20
eee	—	—	0.15
fff	—	—	0.08
n	—	176	—

Figure C 176-Pin LFBGA (PLBG0176GG-A)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP176-24x24-0.50	PLQP0176KJ-A	1.90

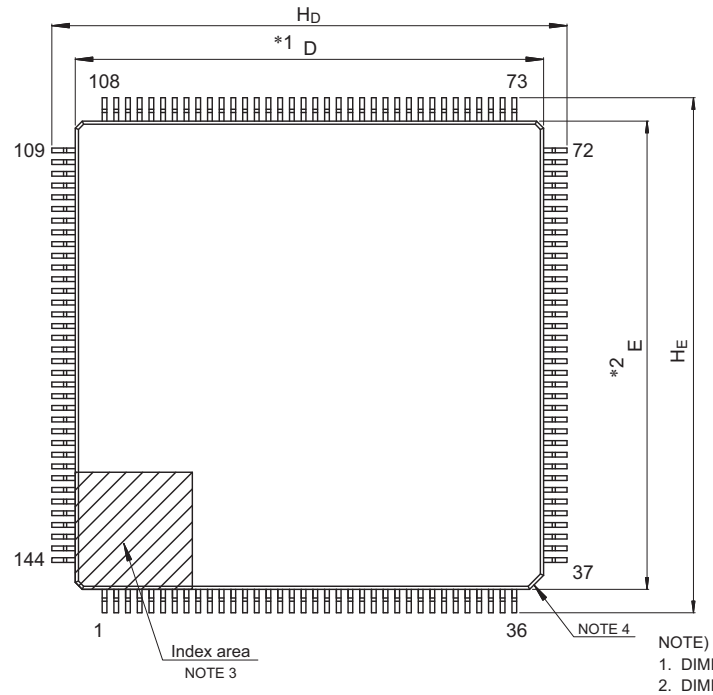


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	1.60
A_1	0.05	—	0.15
A_2	1.35	1.40	1.45
D		26.00 BSC.	
D_1		24.00 BSC.	
E		26.00 BSC.	
E_1		24.00 BSC.	
N	—	176	—
e		0.50 BSC.	
b	0.17	0.22	0.27
c	0.09	—	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1		1.00 REF.	
aaa	—	—	0.20
bbb	—	—	0.20
ccc	—	—	0.08
ddd	—	—	0.08

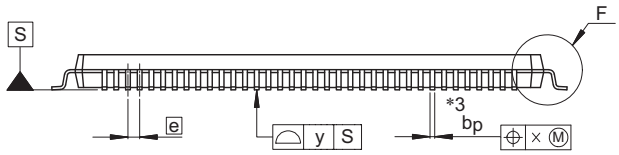
Figure D 176-Pin LFQFP (PLQP0176KJ-A)

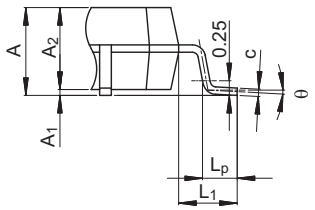
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP144-20x20-0.50	PLQP0144KA-B	—	1.2

Unit: mm



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.



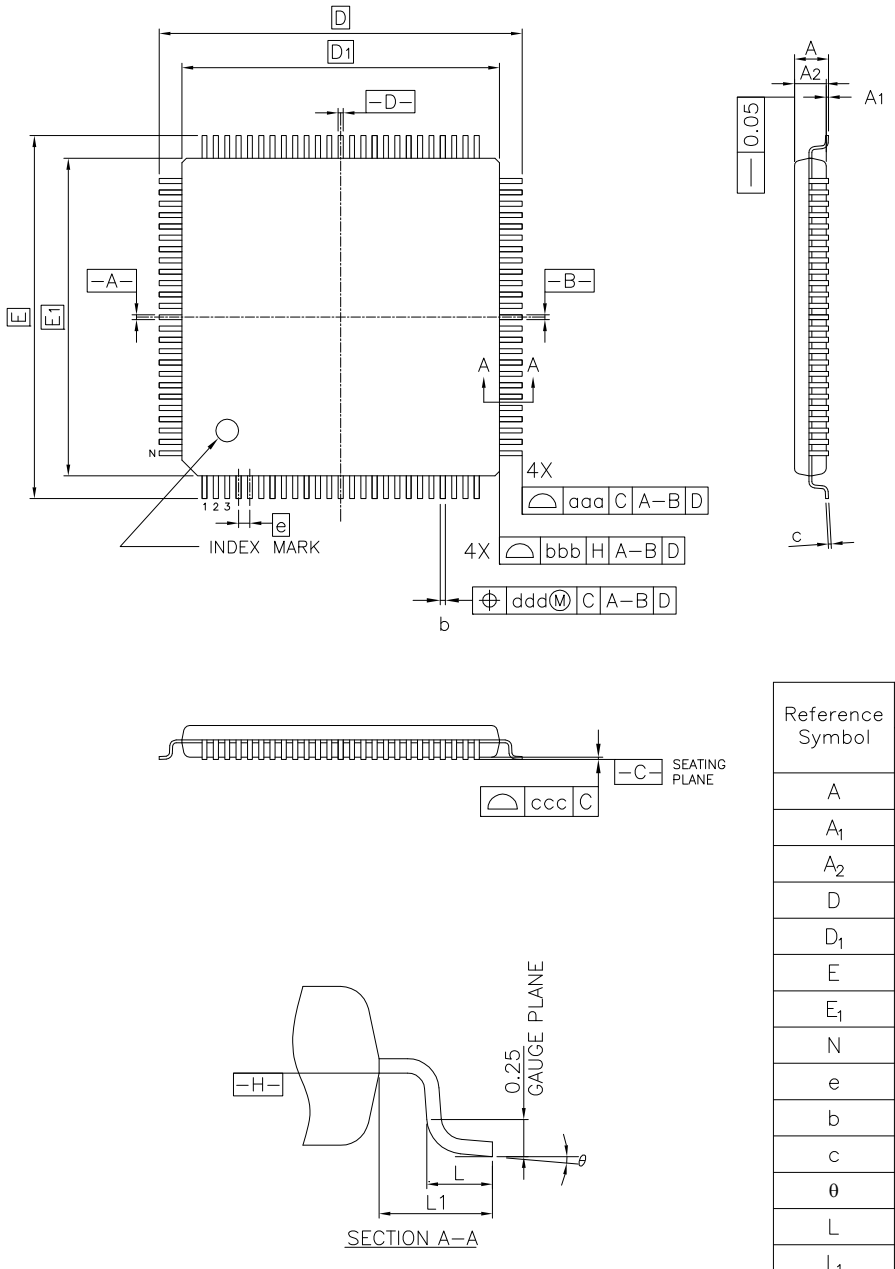


Detail F

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure E 144-Pin LFQFP (PLQP0144KA-B)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP100-14x14-0.50	PLQP0100KP-A	0.67



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	16.00	–
D_1	–	14.00	–
E	–	16.00	–
E_1	–	14.00	–
N	–	100	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

Figure F 100-Pin LFQFP (PLQP0100KP-A)

Appendix 3. Cycles for Access

This appendix gives information on the numbers of clock cycles for access to I/O registers described in this manual.

- Numbers of cycles for access are given per module symbol.
- Numbers of cycles for access are of the stated reference clocks.
- The internal I/O register areas include reserved areas, the addresses of which are not given in the descriptions of registers in the corresponding section. Access to the reserved areas is prohibited. Operation in response to attempted access to these registers and subsequent operation cannot be guaranteed.
- The number of cycles for access to an I/O register depends on the number of clock cycles in bus cycles for on-chip peripherals, cycles for synchronization with a frequency-divided clock, and cycles of waiting for each module.

  The number of cycles for synchronization with a frequency-divided clock depends on the ratio between the frequencies of ICLK and PCLK.
- When the frequencies of ICLK and PCLK are the same, the number of cycles for synchronization with a frequency-divided clock is always fixed.
- When the frequency of ICLK is higher than that of PCLK, at least one cycle of PCLK is added to the number of cycles for synchronization with a frequency-divided clock.
- The listed numbers of cycles for write access are for non-buffered writing.

Note: The numbers of cycles for access given here are for cases where access to registers by the CPU is not in contention with the fetching of instructions to an external memory or with bus access by any other bus master such as the DMAC or DTC.

Table 3.1 List of Cycles for Access (1/4)

Secure Module Symbol	Address (Secure)	Address (Non-secure)	Number of Cycles for Access				Unit	Related Function
			CPUCLK0 = ICLK = PCLK		CPUCLK0 = ICLK > PCLK*1			
			Read	Write	Read	Write		
CPU_CTRL	8100 0000h to 8100 03FFh	0100 0000h to 0100 03FFh	1	1	1	1	CPUCLK0	CPU
SAU	8100 0400h to 8100 07FFh	—	2	2	2	2	CPUCLK0	Security Features
MPU	8100 0800h to 8100 0FFFh	0100 0800h to 0100 0FFFh	2	2	2	2	CPUCLK0	CPU Memory-Protection Unit (MPU)
TFU	8100 1000h to 8100 1FFFh	0100 1000h to 0100 1FFFh	1	1	1	1	CPUCLK0	Arithmetic Unit for Trigonometric Functions
CACHE	8101 0000h to 8101 07FFh	0101 0000h to 0101 07FFh	6	4	6	4	CPUCLK0	Cache Control
LRAM	8101 0800h to 8101 0FFFh	0101 0800h to 0101 0FFFh	6	4	6	4	CPUCLK0	Local SRAM (LRAM) Control
ICU	8700 6000h to 8700 641Fh	0700 6000h to 0700 641Fh	3	2	3	2	ICLK	Interrupt Controller (ICUM)
ICU	8700 6520h to 8700 6521h	0700 6520h to 0700 6521h	3	2	0 to 3	0 to 2	PCLKB	Interrupt Controller (ICUM)
ICU	8700 6540h to 8700 65FFh	0700 6540h to 0700 65FFh	3	2	3	2	ICLK	Interrupt Controller (ICUM)
ICU	8700 6600h to 8700 66FFh	0700 6600h to 0700 66FFh	3	2	0 to 3	0 to 2	PCLKB	Interrupt Controller (ICUM)
ICU	8700 6700h to 8700 69FFh	0700 6700h to 0700 69FFh	3	2	0 to 3	0 to 2	PCLKA	Interrupt Controller (ICUM)
ICU	8700 6A00h to 8700 6FFFh	0700 6A00h to 0700 6FFFh	3	2	3	2	ICLK	Interrupt Controller (ICUM)
LPM	8700 6BFCh to 8700 6BFFh	0700 6BFCh to 0700 6BFFh	3	2	3	2	ICLK	Low Power Consumption
ICU	8700 6C00h to 8700 6FFFh	0700 6C00h to 0700 6FFFh	3	2	3	2	ICLK	Interrupt Controller (ICUM)
DMAC0	8700 7000h to 8700 703Fh	0700 7000h to 0700 703Fh	3	2	3	2	ICLK	DMA Controller 0 (DMAC0)
DMAC1	8700 7040h to 8700 707Fh	0700 7040h to 0700 707Fh	3	2	3	2	ICLK	DMA Controller 1 (DMAC1)
DMAC2	8700 7080h to 8700 70BFh	0700 7080h to 0700 70BFh	3	2	3	2	ICLK	DMA Controller 2 (DMAC2)
DMAC3	8700 70C0h to 8700 70FFh	0700 70C0h to 0700 70FFh	3	2	3	2	ICLK	DMA Controller 3 (DMAC3)
DMAC4	8700 7100h to 8700 713Fh	0700 7100h to 0700 713Fh	3	2	3	2	ICLK	DMA Controller 4 (DMAC4)
DMAC5	8700 7140h to 8700 717Fh	0700 7140h to 0700 717Fh	3	2	3	2	ICLK	DMA Controller 5 (DMAC5)
DMAC6	8700 7180h to 8700 71BFh	0700 7180h to 0700 71BFh	3	2	3	2	ICLK	DMA Controller 6 (DMAC6)
DMAC7	8700 71C0h to 8700 71FFh	0700 71C0h to 0700 71FFh	3	2	3	2	ICLK	DMA Controller 7 (DMAC7)
DMAC	8700 7800h to 8700 7BFFh	0700 7800h to 0700 7BFFh	3	2	3	2	ICLK	DMA Controller (DMAC)

Table 3.1 List of Cycles for Access (2/4)

Secure Module Symbol	Address (Secure)	Address (Non-secure)	Number of Cycles for Access				Unit	Related Function
			CPUCLK0 = ICLK = PCLK		CPUCLK0 = ICLK > PCLK*1			
			Read	Write	Read	Write		
DTC	8700 7C00h to 8700 7FFFh	0700 7C00h to 0700 7FFFh	3	2	3	2	ICLK	Data Transfer Controller (DTCA)
BUSMPU	8701 0000h to 8701 1FFFh	0701 0000h to 0701 1FFFh	3	2	3	2	ICLK	Bus master Memory-Protection Unit (MPU)
CPSCU	8701 8000h to 8701 8FFFh	0701 8000h to 0701 8FFFh	3	2	3	2	ICLK	CPU System Security Attribute Control Unit
SYSC	8701 9000h to 8701 97FFh	0701 9000h to 0701 97FFh	4	3	2 to 4	1 to 3	PCLKB	System Control
SYSC	8701 9800h to 8701 9FFFh	0701 9800h to 0701 9FFFh	7	6	5 to 7	4 to 6	PCLKB	System Control
EXDMAC0	8701 A000h to 8701 A03Fh	0701 A000h to 0701 A03Fh	3	2	3	2	ICLK	EXDMA Controller 0 (EXDMAC0)
EXDMAC1	8701 A040h to 8701 A07Fh	0701 A040h to 0701 A07Fh	3	2	3	2	ICLK	EXDMA Controller 1 (EXDMAC1)
EXDMAC	8701 A800h to 8701 AFFFh	0701 A800h to 0701 AFFFh	3	2	3	2	ICLK	EXDMA Controller (EXDMAC)
SRAM	8701 C000h to 8701 CFFFh	0701 C000h to 0701 CFFFh	3	2	3	2	ICLK	SRAM
BUS	8701 D000h to 8701 DFFFh	0701 D000h to 0701 DFFFh	3	2 to 10	3	2 to 10	ICLK	Buses
BUS	8701 E000h to 8701 EFFFh	0701 E000h to 0701 EFFFh	3	2	3	2	ICLK	Buses
MRAM	8709 C000h to 8709 FFFFh	0709 C000h to 0709 FFFFh	4	3	2 to 4	1 to 3	MRPCLK	MRAM
CRC	8741 0000h to 8741 000Fh	0741 0000h to 0741 000Fh	4	3	2 to 4	1 to 3	PCLKA	CRC Calculator
DOC	8741 1000h to 8741 101Fh	0741 1000h to 0741 101Fh	4	3	2 to 4	1 to 3	PCLKA	Data Operation Circuit
POEG	8741 2000h to 8741 233Fh	0741 2000h to 0741 233Fh	4	3	2 to 4	1 to 3	PCLKA	GPTW Port Output Enable
HRPWM0	8742 4000h to 8742 407Fh	0742 4000h to 0742 407Fh	3	2	1 to 3	0 to 2	PCLKA	High Resolution PWM Waveform Generation Circuit 0
GPTW	8742 BF00h to 8742 BF03h	0742 BF00h to 0742 BF03h	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer (GPTW)
GPTW	8742 BF10h to 8742 BF13h	0742 BF10h to 0742 BF13h	4	3	2 to 4	1 to 3	PCLKA	General PWM Timer (GPTW)
GPTW0	8742 C000h to 8742 C0FFh	0742 C000h to 0742 C0FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 0 (GPTW0)
GPTW1	8742 C100h to 8742 C1FFh	0742 C100h to 0742 C1FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 1 (GPTW1)
GPTW2	8742 C200h to 8742 C2FFh	0742 C200h to 0742 C2FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 2 (GPTW2)
GPTW3	8742 C300h to 8742 C3FFh	0742 C300h to 0742 C3FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 3 (GPTW3)
GPTW4	8742 C400h to 8742 C4FFh	0742 C400h to 0742 C4FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 4 (GPTW4)
GPTW5	8742 C500h to 8742 C5FFh	0742 C500h to 0742 C5FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 5 (GPTW5)
GPTW6	8742 C600h to 8742 C6FFh	0742 C600h to 0742 C6FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 6 (GPTW6)
GPTW7	8742 C700h to 8742 C7FFh	0742 C700h to 0742 C7FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 7 (GPTW7)
GPTW8	8742 C800h to 8742 C8FFh	0742 C800h to 0742 C8FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 8 (GPTW8)
GPTW9	8742 C900h to 8742 C9FFh	0742 C900h to 0742 C9FFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 9 (GPTW9)
GPTW10	8742 CA00h to 8742 CAFFh	0742 CA00h to 0742 CAFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 10 (GPTW10)
GPTW11	8742 CB00h to 8742 CBFFh	0742 CB00h to 0742 CBFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 11 (GPTW11)
GPTW12	8742 CC00h to 8742 CCFFh	0742 CC00h to 0742 CCFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 12 (GPTW12)
GPTW13	8742 CD00h to 8742 CDFFh	0742 CD00h to 0742 CDFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 13 (GPTW13)
GPTW14	8742 CE00h to 8742 CEFFh	0742 CE00h to 0742 CEFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 14 (GPTW14)
GPTW15	8742 CF00h to 8742 CFFFh	0742 CF00h to 0742 CFFFh	9	6	7 to 9	4 to 6	PCLKA	General PWM Timer 15 (GPTW15)
ADC_B	8743 8000h to 8743 9FFFh	0743 8000h to 0743 9FFFh	4	3	2 to 4	1 to 3	PCLKA	16-Bit A/D Converter
ADC_B	8743 A000h to 8743 A3FFh	0743 A000h to 0743 A3FFh	3	2	1 to 3	0 to 2	PCLKA	16-Bit A/D Converter
USBHS*2	8745 1000h to 8745 115Fh	0745 1000h to 0745 115Fh	BWAIT+4	BWAIT+3	BWAIT+2 to BWAIT+4	BWAIT+1 to BWAIT+3	PCLKA	USB 2.0 High-Speed Host/Function Module (USBHS)
USBHS*2	8745 1160h to 8745 1167h	0745 1160h to 0745 1167h	BWAIT+4	130	BWAIT+2 to BWAIT+4	128 to 130	PCLKA	USB 2.0 High-Speed Host/Function Module (USBHS)
USBHS	8745 1168h to 8745 11FFh	0745 1168h to 0745 11FFh	8	130	6 to 8	128 to 130	PCLKA	USB 2.0 High-Speed Host/Function Module (USBHS)
RSCI0	8745 8000h to 8745 807Fh	0745 8000h to 0745 807Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 0 (RSCI0)
RSCI1	8745 8100h to 8745 817Fh	0745 8100h to 0745 817Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 1 (RSCI1)
RSCI2	8745 8200h to 8745 827Fh	0745 8200h to 0745 827Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 2 (RSCI2)
RSCI3	8745 8300h to 8745 837Fh	0745 8300h to 0745 837Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 3 (RSCI3)

Table 3.1 List of Cycles for Access (3/4)

Secure Module Symbol	Address (Secure)	Address (Non-secure)	Number of Cycles for Access				Unit	Related Function
			CPUCLK0 = ICLK = PCLK		CPUCLK0 = ICLK > PCLK*1			
			Read	Write	Read	Write		
RSCI4	8745 8400h to 8745 847Fh	0745 8400h to 0745 847Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 4 (RSCI4)
RSCI5	8745 8500h to 8745 857Fh	0745 8500h to 0745 857Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 5 (RSCI5)
RSCI6	8745 8600h to 8745 867Fh	0745 8600h to 0745 867Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 6 (RSCI6)
RSCI7	8745 8700h to 8745 877Fh	0745 8700h to 0745 877Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 7 (RSCI7)
RSCI8	8745 8800h to 8745 887Fh	0745 8800h to 0745 887Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 8 (RSCI8)
RSCI9	8745 8900h to 8745 897Fh	0745 8900h to 0745 897Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 9 (RSCI9)
RSCI10	8745 8A00h to 8745 8A7Fh	0745 8A00h to 0745 8A7Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 10 (RSCI10)
RSCI11	8745 8B00h to 8745 8B7Fh	0745 8B00h to 0745 8B7Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 11 (RSCI11)
RSCI12	8745 8C00h to 8745 8C7Fh	0745 8C00h to 0745 8C7Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Communications Interface 12 (RSCI12)
RSPIA0	8745 C000h to 8745 C07Fh	0745 C000h to 0745 C07Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Peripheral Interface 0 (RSPIA0)
RSPIA1	8745 C100h to 8745 C17Fh	0745 C100h to 0745 C17Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Peripheral Interface 1 (RSPIA1)
RSPIA2	8745 C200h to 8745 C27Fh	0745 C200h to 0745 C27Fh	4	3	2 to 4	1 to 3	PCLKA	Serial Peripheral Interface 2 (RSPIA2)
RI3C0	8745 F000h to 8745 F3FFh	0745 F000h to 0745 F3FFh	4	3	2 to 4	1 to 3	PCLKA	I3C Bus Interface 0 (RI3C0)
RIICHS1	8745 F400h to 8745 F7FFh	0745 F400h to 0745 F7FFh	4	3	2 to 4	1 to 3	PCLKA	High-Speed I²C-bus Interface 1 (RIICHS1)
RIICHS2	8745 F800h to 8745 FBFFh	0745 F800h to 0745 FBFFh	4	3	2 to 4	1 to 3	PCLKA	High-Speed I²C-bus Interface 2 (RIICHS2)
CANFDG0	8746 E000h to 8746 E21Fh	0746 E000h to 0746 E21Fh	5	4	3 to 5	2 to 4	PCLKA	CAN FD module with gateway functionality
CANFD	8746 F200h to 8746 F21Fh	0746 F200h to 0746 F21Fh	5	4	3 to 5	2 to 4	PCLKA	CAN FD Module (CANFD)
CANFD, CANFD2	8747 0000h to 8747 011Fh	0747 0000h to 0747 011Fh	4	3	2 to 4	1 to 3	PCLKA	CAN FD Module (CANFD), CAN FD Module 2 (CANFD2)
CANFD, CANFD2	8747 0120h to 8747 1FFFh	0747 0120h to 0747 1FFFh	5	4	3 to 5	2 to 4	PCLKA	CAN FD Module (CANFD), CAN FD Module 2 (CANFD2)
CANFDG0, CANFD0, CANFD1	8748 0000h to 8748 17FFh	0748 0000h to 0748 17FFh	4	3	2 to 4	1 to 3	PCLKA	CAN FD module with gateway functionality (CANFDG), CAN FD module with gateway functionality 0 (CANFD0), CAN FD module with gateway functionality 1 (CANFD1)
CANFDG0, CANFD0, CANFD1	8748 1800h to 8749 FFFFh	0748 1800h to 0749 FFFFh	5	4	3 to 5	2 to 4	PCLKA	CAN FD module with gateway functionality (CANFDG), CAN FD module with gateway functionality 0 (CANFD0), CAN FD module with gateway functionality 1 (CANFD1)
MFWD, ESWM, COMA, ETHA0, RMAC0, GWCA0, GPTP*3	874C 0000h to 874E 07FFh	074C 0000h to 074E 07FFh	—	—	9 to 10	3 to 4	PCLKA	Ethernet Module, Ethernet Message Forwarding Engine (MFWD), Ethernet Common Agent (COMA), Ethernet Agent 0 (ETHA0), Ethernet MAC 0 (RMAC0), Ethernet CPU Agent 0 (GWCA0), Ethernet Generic PTP Timer (GPTP)
ESWM, GPTP*3	874E 1000h to 874E 2FFFh	074E 1000h to 074E 2FFFh	—	—	3 to 4	2 to 3	PCLKA	Ethernet Module, Ethernet Generic PTP Timer (GPTP)
ELC	8780 1000h to 8780 13FFh	0780 1000h to 0780 13FFh	4	3	2 to 4	1 to 3	PCLKB	Event Link Controller (ELC)
RTC	8780 2000h to 8780 207Fh	0780 2000h to 0780 207Fh	4	3	2 to 4	1 to 3	PCLKB	Realtime Clock (RTC)
IWDT	8780 2200h to 8780 220Fh	0780 2200h to 0780 220Fh	4	65	2 to 4	63 to 65	PCLKB	Independent Watchdog Timer (IWDT)
CAC0	8780 2400h to 8780 247Fh	0780 2400h to 0780 247Fh	4	3	2 to 4	1 to 3	PCLKB	Clock Frequency Accuracy Measurement Circuit 0 (CAC0)
CAC1	8780 2480h to 8780 24FFh	0780 2480h to 0780 24FFh	4	3	2 to 4	1 to 3	PCLKB	Clock Frequency Accuracy Measurement Circuit 1 (CAC1)
CAC2	8780 2500h to 8780 257Fh	0780 2500h to 0780 257Fh	4	3	2 to 4	1 to 3	PCLKB	Clock Frequency Accuracy Measurement Circuit 2 (CAC2)
WDT0	8780 2600h to 8780 260Fh	0780 2600h to 0780 260Fh	4	3	2 to 4	1 to 3	PCLKB	Watchdog Timer (WDT)
MSTP	8780 3000h to 8780 301Fh	0780 3000h to 0780 301Fh	4	3	2 to 4	1 to 3	PCLKB	Module-stop function

Table 3.1 List of Cycles for Access (4/4)

Secure Module Symbol	Address (Secure)	Address (Non-secure)	Number of Cycles for Access				Unit	Related Function
			CPUCLK0 = ICLK = PCLK		CPUCLK0 = ICLK > PCLK*1			
			Read	Write	Read	Write		
PSCU	8780 4000h to 8780 407Fh	0780 4000h to 0780 407Fh	4	3	2 to 4	1 to 3	PCLKB	Peripheral Module Security Attribution Control
PIO	8780 6000h to 8780 60FFh	0780 6000h to 0780 60FFh	4	3	2 to 4	1 to 3	PCLKB	Software Configurable 32-Bit Programmable I/O Port (PIO)
GPTW11, GPTW13, RSCI2, RSCI3	8780 7000h to 8780 703Fh	0780 7000h to 0780 703Fh	4	3	2 to 4	1 to 3	PCLKB	General PWM Timer 11 (GPTW11), General PWM Timer 13 (GPTW13), Serial Communications Interface 2 (RSCI2), Serial Communications Interface 3 (RSCI3)
TMR0, TMR01, TMR1	8782 5000h to 8782 500Fh	0782 5000h to 0782 500Fh	4	3	2 to 4	1 to 3	PCLKB	8-Bit Timer 0 (TMR0), 8-Bit Timer 0 (TMR0) + 8-Bit Timer 1 (TMR1), 8-Bit Timer 1 (TMR1)
TMR2, TMR23, TMR3	8782 5010h to 8782 501Fh	0782 5010h to 0782 501Fh	4	3	2 to 4	1 to 3	PCLKB	8-Bit Timer 2 (TMR2), 8-Bit Timer 2 (TMR2) + 8-Bit Timer 3 (TMR3), 8-Bit Timer 3 (TMR3)
CMT, CMT0, CMT1, CMT2, CMT3	8782 6000h to 8782 601Fh	0782 6000h to 0782 601Fh	4	3	2 to 4	1 to 3	PCLKB	Compare Match Timer (CMT), Compare Match Timer 0 (CMT0), Compare Match Timer 1 (CMT1), Compare Match Timer 2 (CMT2), Compare Match Timer 3 (CMT3)
CMTW0	8782 6100h to 8782 613Fh	0782 6100h to 0782 613Fh	4	3	2 to 4	1 to 3	PCLKB	Compare Match Timer W 0 (CMTW0)
CMTW1	8782 6140h to 8782 617Fh	0782 6140h to 0782 617Fh	4	3	2 to 4	1 to 3	PCLKB	Compare Match Timer W 1 (CMTW1)
PCIF0	8782 7000h to 8782 707Fh	0782 7000h to 0782 707Fh	4	3	2 to 4	1 to 3	PCLKB	Phase-Counting Interface 0 (PCIF0)
PCIF1	8782 7080h to 8782 70FFh	0782 7080h to 0782 70FFh	4	3	2 to 4	1 to 3	PCLKB	Phase-Counting Interface 1 (PCIF1)
DAC12R0	8783 3000h to 8783 307Fh	0783 3000h to 0783 307Fh	4	3	2 to 4	1 to 3	PCLKB	12-Bit D/A Converter 0 (DAC12R0)
DAC12R1	8783 3100h to 8783 317Fh	0783 3100h to 0783 317Fh	4	3	2 to 4	1 to 3	PCLKB	12-Bit D/A Converter 1 (DAC12R1)
TEMPS	8783 5000h to 8783 507Fh	0783 5000h to 0783 507Fh	4	3	2 to 4	1 to 3	PCLKB	Temperature Sensor (TEMPS)
PDC	8784 1000h to 8784 103Fh	0784 1000h to 0784 103Fh	4	3	2 to 4	1 to 3	PCLKB	Parallel Data Capture Unit (PDC)
USB0	8785 0000h to 8785 00FFh	0785 0000h to 0785 00FFh	5	4	3 to 5	2 to 4	PCLKB	USB 2.0 Full-Speed Host/Function Module 0 (USB0)
USB0	8785 0400h to 8785 040Fh	0785 0400h to 0785 040Fh	4	65	2 to 4	63 to 65	PCLKB	USB 2.0 Full-Speed Host/Function Module 0 (USB0)
SDHI	8785 2000h to 8785 23FFh	0785 2000h to 0785 23FFh	4	3	2 to 4	1 to 3	PCLKB	SD/MMC Host Interface (SDHI)
PDMIF	8785 6000h to 8785 63FFh	0785 6000h to 0785 63FFh	4	3	2 to 4	1 to 3	PCLKB	Pulse-Density Modulation Interface (PDM-IF)
SSIE0	8785 D000h to 8785 D03Fh	0785 D000h to 0785 D03Fh	4	3	2 to 4	1 to 3	PCLKB	Serial Sound Interface 0 (SSIE0)
SSIE1	8785 D100h to 8785 D13Fh	0785 D100h to 0785 D13Fh	4	3	2 to 4	1 to 3	PCLKB	Serial Sound Interface 1 (SSIE1)
XSPI0	8786 7000h to 8786 73FFh	0786 7000h to 0786 73FFh	14	13	7 to 11	6 to 10	PCLKB	Expanded Serial Peripheral Interface 0 (XSPI0)
XSPI1	8786 7400h to 8786 77FFh	0786 7400h to 0786 77FFh	14	13	7 to 11	6 to 10	PCLKB	Expanded Serial Peripheral Interface 1 (XSPI1)
XSPI0	8786 8000h to 8786 83FFh	0786 8000h to 0786 83FFh	4	3	2 to 4	1 to 3	PCLKB	Expanded Serial Peripheral Interface 0 (XSPI0)
XSPI1	8786 8400h to 8786 87FFh	0786 8400h to 0786 87FFh	4	3	2 to 4	1 to 3	PCLKB	Expanded Serial Peripheral Interface 1 (XSPI1)
DOTF0	8786 8800h to 8786 88FFh	0786 8800h to 0786 88FFh	4	3	2 to 4	1 to 3	PCLKB	Decryption On The Fly 0 (DOTF0)
DOTF1	8786 8900h to 8786 89FFh	0786 8900h to 0786 89FFh	4	3	2 to 4	1 to 3	PCLKB	Decryption On The Fly 1 (DOTF1)
PORT0 to 9, PORTA to G	87C0 0000h to 87C0 021Fh	07C0 0000h to 07C0 021Fh	4	2	4	2	ICLK	Port Control (n = 0 to 9, A to G)
PFS	87C0 0800h to 87C0 0FFFh	07C0 0800h to 07C0 0FFFh	10	2	10	2	ICLK	Pmn Pin Function Selection (m = 0 to 9, A to G; n = 00 to 15)

Note 1. When the number of cycles of PCLK or MRPCLK is a non-integer value such as 1.5, the minimum and maximum values are expressed as the results of rounding down and up to the nearest integer, respectively. For example, 1.5 to 2.5 is expressed as 1 to 3.

Note 2. “BWAIT” is the number of cycles of waiting specified by USBHS.BUSWAIT.

Note 3. When the ESWM is to be used, do not set ICLK and PCLKA to run at the same frequency. Regarding the operating clock for the ESWM, refer to section 33.3.1, Operating Clocks.

REVISION HISTORY	RX74N Group User's Manual: Hardware

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Jul 31, 2026	—	First edition, issued	

RX74N Group User’s Manual: Hardware

Publication Date: Rev.1.10 Jul 31, 2026

Published by: Renesas Electronics Corporation

RX74N Group

Renesas Electronics Corporation

R01UH1056EJ0110